



文档名称：人员计数 API 参考手册

文档版本号：V1.0

发布日期：2017/8/22

---

**版权所有 © 珠海全志科技股份有限公司 2017。保留一切权利。**

非经本公司书面许可，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。

## **商标声明**



、全志和其他全志商标均为珠海全志科技股份有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## **注意**

您购买的产品、服务或特性等应受全志公司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。除非合同另有约定，全志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

## 概述

本手册将对运动人员计数 API 接口进行介绍，为正确使用运动人员计数算法库请阅读本手册。

## 产品版本

| 产品名称   | 产品版本 |
|--------|------|
| V5-100 | V1.0 |
|        |      |
|        |      |

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 电子产品设计开发人员

## 修订记录

| 版本号  | 修订日期     | 修订内容   |
|------|----------|--------|
| V1.0 | 20170822 | 正式发布版本 |
|      |          |        |
|      |          |        |

# 目 录

|   |    |
|---|----|
| 1. 概述.....                                    | 1  |
| 1.1. 功能描述.....                                | 1  |
| 1.2. API 函数参考域.....                           | 1  |
| 1.3. 数据类型参考域.....                             | 2  |
| 1.4. 相关文档.....                                | 3  |
| 2. API 参考.....                                | 4  |
| 2.1. 人员计数算法 API 参考.....                       | 4  |
| 2.1.1. AW_AI_CVE_HCNT_GetAlgoVersion.....     | 5  |
| 2.1.2. AW_AI_CVE_HCNT_Init.....               | 5  |
| 2.1.3. AW_AI_CVE_HCNT_UnInit.....             | 6  |
| 2.1.4. AW_AI_CVE_HCNT_Process.....            | 7  |
| 2.1.5. AW_AI_CVE_HCNT_GetLastError.....       | 8  |
| 2.1.6. AW_AI_CVE_HCNT_GetSceneVersion.....    | 9  |
| 2.1.7. AW_AI_CVE_HCNT_PacketParamFile.....    | 9  |
| 2.1.8. AW_AI_CVE_HCNT_PacketParamString.....  | 10 |
| 2.1.9. AW_AI_CVE_HCNT_ParseParamFile.....     | 11 |
| 2.1.10. AW_AI_CVE_HCNT_ParseParamString.....  | 12 |
| 2.1.11. AW_AI_CVE_CLBR_GetSceneVersion.....   | 13 |
| 2.1.12. AW_AI_CVE_CLBR_PacketParamFile.....   | 13 |
| 2.1.13. AW_AI_CVE_CLBR_PacketParamString..... | 14 |
| 2.1.14. AW_AI_CVE_CLBR_ParseParamFile.....    | 15 |
| 2.1.15. AW_AI_CVE_CLBR_ParseParamString.....  | 16 |
| 2.2. 数据类型.....                                | 17 |
| 2.2.1. 人员计数算法结果数据类型.....                      | 17 |
| 2.2.2. 景深标定算法配置数据类型.....                      | 28 |
| 2.3. 错误码.....                                 | 30 |
| 3. 开发应用.....                                  | 32 |
| 3.1. 调用流程.....                                | 33 |
| 3.2. 实例介绍.....                                | 33 |
| 4. 算法资源需求指标.....                              | 35 |
| 4.1. 算法资源需求.....                              | 35 |

# 1. 概述

人员计数通过对运动目标检测，提取头部特征等方法来记录出入口人员流动数量。具有功耗低、速度快等特点，针对不同场景提供可配置灵敏度参数。

本章描述内容如下表所示：

| 标题            | 内容          |
|---------------|-------------|
| 1.1 功能描述      | 描述人员计数基本功能  |
| 1.2 API 函数参考域 | 描述 API 参考域。 |
| 1.3 数据类型参考域   | 描述数据类型的参考域。 |
| 1.4 相关文档      | 介绍本指南相关的文档。 |

表 1-1 本章内容说明

## 1.1. 功能描述

- 最大支持 720P 图像输入
- D1 分辨率 25FPS
- 在资源足够的情况下可配置算法的处理帧率
- 可以根据检测效果选择“高”，“中”，“低”三种检测灵敏度
- 单模块运行，硬件功耗小于 100mw

## 1.2. API 函数参考域

本文档对 API 参考信息使用以下域来描述，具体信息如表 1-2 所示。

| 参考域 | 作用                               |
|-----|----------------------------------|
| 描述  | 描述 API 的功能。                      |
| 语法  | 描述 API 的语法样式。                    |
| 参数  | 列出 API 的参数、参数说明及其属性。             |
| 返回值 | 列出 API 的返回值及其返回值说明。              |
| 错误码 | 列出 API 的错误码及其错误码说明。              |
| 需求  | 列出本 API 要包含的头文件和 API 编译时要链接的库文件。 |
| 注意  | 使用 API 时应注意的事项。                  |
| 举例  | 使用 API 的实例                       |

|      |                 |
|------|-----------------|
| 相关主题 | 同本 API 的其他相关信息。 |
|------|-----------------|

表 1-2 API 函数参考域说明

### 1.3. 数据类型参考域

本文档对数据类型参考信息使用以下域来描述，具体信息如表 1-3 所示。

AllwinnerTech

| 参考域       | 作用                               |
|-----------|----------------------------------|
| 说明        | 简单描述结构体所实现的功能。                   |
| 定义        | 列出结构体的定义。                        |
| 成员        | 列出数据结构的成员及含义。                    |
| 注意事项      | 列出使用数据类型时应注意的事项。                 |
| 相关数据类型和接口 | 列出与本数据类型相关联的其他数据类型和接口。           |
| 需求        | 列出本 API 要包含的头文件和 API 编译时要链接的库文件。 |
| 注意        | 使用 API 时应注意的事项。                  |
| 举例        | 使用 API 的实例                       |
| 相关主题      | 同本 API 的其他相关信息。                  |

表 1-3 数据类型参考域说明

## 1.4. 相关文档

与本指南相关的文档有：

- 《人员计数用户手册》

## 2. API 参考

### 关于本章

本章描述内容如下表所示：

| 标题         | 内容                |
|------------|-------------------|
| 2.1 API 参考 | 描述人员计数算法 API 函数参考 |
| 2.2 数据类型   | 描述人员计数算法相关数据类型    |

表 2-1 本章内容说明

### 2.1. 人员计数算法 API 参考

| 人员计数函数列表                         | 功能描述                   |
|----------------------------------|------------------------|
| AW_AI_CVE_HCNT_GetAlgoVersion    | 查询算法版本                 |
| AW_AI_CVE_HCNT_Init              | 创建算法模块并初始化             |
| AW_AI_CVE_HCNT_UnInit            | 终止算法模块并释放内存            |
| AW_AI_CVE_HCNT_Process           | 处理一幅图像，并获取处理结果         |
| AW_AI_CVE_HCNT_GetLastError      | 获取最后的错误码结果             |
| AW_AI_CVE_HCNT_GetSceneVersion   | 查询人员计数算法参数配置版本         |
| AW_AI_CVE_HCNT_PacketParamFile   | 将人员计数算法处理参数打包为 XML 文件  |
| AW_AI_CVE_HCNT_PacketParamString | 将人员计数算法处理参数打包为 XML 字符串 |
| AW_AI_CVE_HCNT_ParseParamFile    | 从 XML 文件解析人员计数算法处理参数   |
| AW_AI_CVE_HCNT_ParseParamString  | 从 XML 字符串解析人员计数算法处理参数  |
| AW_AI_CVE_CLBR_GetSceneVersion   | 查询景深标定算法参数配置版本         |
| AW_AI_CVE_CLBR_PacketParamFile   | 将景深标定算法处理参数打包为 XML 文件  |
| AW_AI_CVE_CLBR_PacketParamString | 将景深标定算法处理参数打包为 XML 字符串 |
| AW_AI_CVE_CLBR_ParseParamFile    | 从 XML 文件解析景深标定算法处理     |



|                                 |                       |
|---------------------------------|-----------------------|
|                                 | 参数                    |
| AW_AI_CVE_CLBR_ParseParamString | 从 XML 字符串解析景深标定算法处理参数 |

表 2-2 人员计数 API 函数列表

### 2.1.1. AW\_AI\_CVE\_HCNT\_GetAlgoVersion

**【描述】**

查询运动人员计数算法版本。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_HCNT\_GetAlgoVersion( char \*pVersion );

**【参数】**

| 参数名称     | 描述     | 输入/输出 |
|----------|--------|-------|
| pVersion | 版本号字符串 | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述 |
|-----------------|----|
| AW_STATUS_OK    | 成功 |
| AW_STATUS_ERROR | 失败 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_interface.h

**【注意】**

无

**【举例】**

无

**【相关主题】**

无

### 2.1.2. AW\_AI\_CVE\_HCNT\_Init

**【描述】**

创建算法模块并初始化。

**【语法】**

AW\_HANDLE AW\_AI\_CVE\_HCNT\_Init( AW\_AI\_CVE\_HCNT\_PARAM \*pstParamHCNT,

版权所有 侵权必究

AW\_AI\_CVE\_CLBR\_PARAM \*pstParamCLBR );

**【参数】**

| 参数名称         | 描述       | 输入/输出 |
|--------------|----------|-------|
| pstParamHCNT | 人员计数算法参数 | 输入    |
| pstParamCLBR | 景深标定算法参数 | 输入    |

**【返回值】**

| 返回值       | 描述     |
|-----------|--------|
| AW_HANDLE | 算法模块句柄 |

**【错误码】**

| 接口返回值           | 描述 |
|-----------------|----|
| AW_STATUS_OK    | 成功 |
| AW_STATUS_ERROR | 失败 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_interface.h

**【注意】**

初始化输入宽度和高度参数需要与 Process 处理图像的宽度和高度保持一致;  
宽度和高度支持 D1、CIF、QCIF 三种分辨率格式, 支持 N 和 P 制式, 建议使用 QCIF;  
支持自定义宽度和高度;  
如返回 NULL, 请调用 AW\_AI\_CVE\_HCNT\_GetLastError 获取错误码

**【举例】**

无

**【相关主题】**

无

### 2.1.3. AW\_AI\_CVE\_HCNT\_UnInit

**【描述】**

终止算法模块并释放内存。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_HCNT\_UnInit ( AW\_HANDLE hModule,  
AW\_AI\_CVE\_HCNT\_RESULT \*pstResult );

**【参数】**

| 参数名称    | 描述     | 输入/输出 |
|---------|--------|-------|
| hModule | 算法模块句柄 | 输入    |

|           |        |    |
|-----------|--------|----|
| pstResult | 算法处理结果 | 输出 |
|-----------|--------|----|

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_interface.h

**【注意】**

必须在 AW\_AI\_CVE\_HCNT\_Init 调用成功后调用。

如返回失败请调用 AW\_AI\_CVE\_HCNT\_GetLastError 获取错误码

**【举例】**

无

**【相关主题】**

无

### 2.1.4. AW\_AI\_CVE\_HCNT\_Process

**【描述】**

处理一幅图像，并获取处理结果。

**【语法】**

```
AW_STATUS_E AW_AI_CVE_HCNT_Process( AW_HANDLE hModule, AW_IMAGE_S
*pstImage, AW_U32 u32TimeStamp, AW_AI_CVE_HCNT_RESULT *pstResult );
```

**【参数】**

| 参数名称         | 描述        | 输入/输出 |
|--------------|-----------|-------|
| hModule      | 算法模块句柄    | 输入    |
| pstImage     | YUV 彩色图像  | 输入    |
| u32TimeStamp | 以帧为单位的时间戳 | 输入    |
| pstResult    | 算法处理结果    | 输出    |

**【返回值】**

| 返回值 | 描述 |
|-----|----|
|-----|----|

|             |      |
|-------------|------|
| AW_STATUS_E | 处理状态 |
|-------------|------|

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_SKIP  | 跳帧。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件：aw\_ai\_cve\_hcnt\_interface.h

**【注意】**

pstImage 需要事先分配好内存空间；

pstImage 的宽度和高度需要与 AW\_AI\_CVE\_HCNT\_Init 的宽度和高度保持一致；

如返回失败请调用 AW\_AI\_CVE\_HCNT\_GetLastError 获取错误码；

**【举例】**

无

**【相关主题】**

无

### 2.1.5. AW\_AI\_CVE\_HCNT\_GetLastError

**【描述】**

获取最后的错误码结果。

**【语法】**

```
AW_AI_CVE_ERROR_CODE_E AW_AI_CVE_HCNT_GetLastError( AW_HANDLE *hModule );
```

**【参数】**

| 参数名称    | 描述     | 输入/输出 |
|---------|--------|-------|
| hModule | 算法模块句柄 | 输入    |

**【返回值】**

| 返回值                    | 描述    |
|------------------------|-------|
| AW_AI_CVE_ERROR_CODE_E | 算法错误码 |

**【需求】**

头文件：aw\_ai\_cve\_hcnt\_interface.h

**【注意】**

1、错误码信息详见 3.3 节。

**【举例】**

无

**【相关主题】**

无

### 2.1.6. AW\_AI\_CVE\_HCNT\_GetSceneVersion

**【描述】**

查询人员计数算法参数配置版本。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_HCNT\_GetSceneVersion( AW\_CHAR\*pVersion );

**【参数】**

| 参数名称     | 描述     | 输入/输出 |
|----------|--------|-------|
| pVersion | 版本号字符串 | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_config.h

**【注意】**

无。

**【举例】**

无

**【相关主题】**

无

### 2.1.7. AW\_AI\_CVE\_HCNT\_PacketParamFile

**【描述】**

将人员计数算法处理参数打包为 XML 文件。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_HCNT\_PacketParamFile( AW\_AI\_CVE\_HCNT\_PARAM\_ALGO

\*pParams, AW\_CHAR \*pFilNam, AW\_PVOID \*pMemMgr );

**【参数】**

| 参数名称    | 描述    | 输入/输出 |
|---------|-------|-------|
| pParams | 参数数据  | 输入    |
| pFilNam | 参数文件名 | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_config.h

**【注意】**

无。

**【举例】**

无

**【相关主题】**

无

### 2.1.8. AW\_AI\_CVE\_HCNT\_PacketParamString

**【描述】**

将人员计数算法处理参数打包为 XML 字符串。

**【语法】**

AW\_STATUS\_E

AW\_AI\_CVE\_HCNT\_PacketParamString( AW\_AI\_CVE\_HCNT\_PARAM\_ALGO \*pParams,  
AW\_CHAR \*pBuffer, AW\_S32 nBufLen, AW\_PVOID \*pMemMgr );

**【参数】**

| 参数名称    | 描述   | 输入/输出 |
|---------|------|-------|
| pParams | 参数数据 | 输入    |

|         |        |    |
|---------|--------|----|
| pBuffer | 字符串缓冲区 | 输出 |
| nBufLen | 缓冲区长度  | 输入 |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件：aw\_ai\_cve\_hcnt\_config.h

**【注意】**

无。

**【举例】**

无

**【相关主题】**

无

### 2.1.9. AW\_AI\_CVE\_HCNT\_ParseParamFile

**【描述】**

从 XML 文件解析人员计数算法处理参数。

**【语法】**

```
AW_STATUS_E AW_AI_CVE_HCNT_ParseParamFile(AW_CHAR *pFilNam, AW_AI_CVE_HCNT_PARAM_ALGO *pParams, AW_PVOID *pMemMgr );
```

**【参数】**

| 参数名称    | 描述    | 输入/输出 |
|---------|-------|-------|
| pFilNam | 参数文件名 | 输入    |
| pParams | 参数数据  | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_config.h

**【注意】**

无

**【举例】**

无

**【相关主题】**

无

无

### 2.1.10. AW\_AI\_CVE\_HCNT\_ParseParamString

**【描述】**

从 XML 字符串解析人员计数算法处理参数。

**【语法】**

```
AW_STATUS_E AW_AI_CVE_HCNT_ParseParamString(AW_CHAR *pXmlString,
AW_AI_CVE_HCNT_PARAM_ALGO *pParams, AW_PVOID *pMemMgr );
```

**【参数】**

| 参数名称       | 描述      | 输入/输出 |
|------------|---------|-------|
| pXmlString | XML 字符串 | 输入    |
| pParams    | 参数数据    | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_hcnt\_config.h

**【注意】**



无

**【举例】**

无

**【相关主题】**

无

### 2.1.11. AW\_AI\_CVE\_CLBR\_GetSceneVersion

**【描述】**

查询景深标定算法参数配置版本。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_CLBR\_GetSceneVersion(AW\_CHAR \*pVersion );

**【参数】**

| 参数名称     | 描述     | 输入/输出 |
|----------|--------|-------|
| pVersion | 版本号字符串 | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_clbr\_config.h

**【注意】**

无。

**【举例】**

无

**【相关主题】**

无

### 2.1.12. AW\_AI\_CVE\_CLBR\_PacketParamFile

**【描述】**

将景深标定算法处理参数打包为 XML 文件。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_CLBR\_PacketParamFile( AW\_AI\_CVE\_CLBR\_PARAM \*pParams,  
AW\_CHAR \*pFilNam, AW\_PVOID \*pMemMgr );

**【参数】**

| 参数名称    | 描述    | 输入/输出 |
|---------|-------|-------|
| pParams | 参数数据  | 输入    |
| pFilNam | 参数文件名 | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_clbr\_config.h

**【注意】**

无

**【举例】**

无

**【相关主题】**

无

### 2.1.13. AW\_AI\_CVE\_CLBR\_PacketParamString

**【描述】**

将景深标定算法处理参数打包为 XML 字符串。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_CLBR\_PacketParamString( AW\_AI\_CVE\_CLBR\_PARAM \*pParams, AW\_CHAR \*pBuffer, AW\_S32 nBufLen, AW\_PVOID \*pMemMgr );

**【参数】**

| 参数名称    | 描述     | 输入/输出 |
|---------|--------|-------|
| pParams | 参数数据   | 输入    |
| pBuffer | 字符串缓冲区 | 输出    |

|         |       |    |
|---------|-------|----|
| nBufLen | 缓冲区长度 | 输入 |
|---------|-------|----|

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_clbr\_config.h

**【注意】**

无。

**【举例】**

无

**【相关主题】**

无

### 2.1.14. AW\_AI\_CVE\_CLBR\_ParseParamFile

**【描述】**

从 XML 文件解析景深标定算法处理参数。

**【语法】**

AW\_STATUS\_E AW\_AI\_CVE\_CLBR\_ParseParamFile(AW\_CHAR \*pFilNam, AW\_AI\_CVE\_CLBR\_PARAM \*pParams, AW\_PVOID \*pMemMgr );

**【参数】**

| 参数名称    | 描述    | 输入/输出 |
|---------|-------|-------|
| pFilNam | 参数文件名 | 输入    |
| pParams | 参数数据  | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值 | 描述 |
|-------|----|
|-------|----|

|                 |     |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_clbr\_config.h

**【注意】**

无

**【举例】**

无

**【相关主题】**

无

### 2.1.15. AW\_AI\_CVE\_CLBR\_ParseParamString

**【描述】**

从 XML 字符串解析景深标定算法处理参数。

**【语法】**

```
AW_STATUS_E AW_AI_CVE_CLBR_ParseParamString(AW_CHAR *pXmlString,
AW_AI_CVE_CLBR_PARAM *pParams, AW_PVOID *pMemMgr);
```

**【参数】**

| 参数名称       | 描述      | 输入/输出 |
|------------|---------|-------|
| pXmlString | XML 字符串 | 输入    |
| pParams    | 参数数据    | 输出    |

**【返回值】**

| 返回值         | 描述   |
|-------------|------|
| AW_STATUS_E | 处理状态 |

**【错误码】**

| 接口返回值           | 描述  |
|-----------------|-----|
| AW_STATUS_OK    | 成功。 |
| AW_STATUS_ERROR | 失败。 |

**【需求】**

头文件: aw\_ai\_cve\_clbr\_config.h

**【注意】**

无

**【举例】**

无

【相关主题】

无

## 2.2. 数据类型

### 2.2.1. 人员计数算法结果数据类型

| 数据结构列表                | 功能描述   |
|-----------------------|--------|
| AW_AI_CVE_HCNT_RESULT | 算法处理结果 |
| AW_TARGET_SET_S       | 目标集合   |
| AW_TARGET_S           | 目标数据   |
| AW_TGT_TYPE_E         | 目标类型   |
| AW_TGT_TRAJECT_S      | 目标轨迹   |
| AW_EVENT_SET_S        | 事件集合   |
| AW_EVENT_S            | 事件数据   |
| AW_EVT_TYPE           | 事件类型   |
| AW_EVT_STATUS         | 事件状态   |

#### AW\_AI\_CVE\_HCNT\_RESULT

| 结构定义   |      |        |                    |
|--|------|--------|--------------------|
| <pre>typedef struct {     AW_TARGET_SET_S stTargetSet;     AW_EVENT_SET_S stEventSet; } AW_AI_CVE_DTCA_RESULT;</pre> |      |        |                    |
| 结构说明   |      |        |                    |
| 变量   | 名称   | 含义     | 取值范围               |
| stTargetSet  | 目标集合 | 目标结果数据 | 参考 AW_TARGET_SET_S |
| stEventSet   | 事件集合 | 事件结果数据 | 参考 AW_EVENT_SET_S  |

#### AW\_TARGET\_SET

| 结构定义           |
|----------------|
| typedef struct |

```

{
    AW_S32 s32TargetNum;
    AW_TARGET_S astTargets[AW_MAX_TGT_NUM];
} AW_TARGET_SET;
    
```

| 结构说明         |      |          |                |
|--------------|------|----------|----------------|
| 变量           | 名称   | 含义       | 取值范围           |
| s32TargetNum | 目标数目 | 有效的目标数目  | 0~64           |
| astTargets   | 目标数组 | 用于存储目标数据 | 参考 AW_TARGET_S |

### AW\_TARGET\_S

| 结构定义  |  |  |  |
|---|--|--|--|
| <pre> typedef struct {     AW_U32 u32ID;     AW_S32 u32Type;     AW_POINT_S stPoint;     AW_RECT_S stRect;     AW_S32 s32AreaPix;     AW_S32 s32Area;     AW_S32 s32Direct;     AW_FLOAT fSpeed;     AW_TGT_TRAJECT_S stTraject } AW_TARGET_S;                 </pre> |  |  |  |

| 结构说明       |          |           |                     |
|------------|----------|-----------|---------------------|
| 变量         | 名称       | 含义        | 取值范围                |
| u32ID      | 目标 ID    | 目标身份标识    | 32 位整数              |
| u32Type    | 目标类型     | 目标的类型     | 参考 AW_TGT_TYPE_E    |
| stPoint    | 目标位置     | 目标质心的图像坐标 | 图像像素坐标              |
| stRect     | 目标矩形框    | 目标的外接矩形框  | 图像像素坐标              |
| s32AreaPix | 目标面积(像素) | 目标的像素面积   | 单位: 像素              |
| s32Area    | 目标面积(实际) | 目标的实际面积   | 单位: 平方厘米            |
| s32Direct  | 运动方向     | 目标的运动方向   | 单位: 角度              |
| fSpeed     | 运动速度     | 目标的运动速度   | 单位: 千米/小时           |
| stTraject  | 目标轨迹     | 目标的运动路径   | 参考 AW_TGT_TRAJECT_S |

### AW\_TGT\_TYPE\_E

| 结构定义  |    |        |      |
|---|----|--------|------|
| <pre>typedef enum {     AW_TGT_TYPE_UNKNOWN    = 0,     AW_TGT_TYPE_HUMAN     = 1,     AW_TGT_TYPE_VEHICLE   = 2 } AW_TGT_TYPE;</pre> |    |        |      |
| 结构说明  |    |        |      |
| 变量  | 名称 | 含义     | 取值范围 |
| AW_TGT_TYPE_HUMAN   | 人  | 目标类型为车 | 1    |
| AW_TGT_TYPE_VEHICLE   | 车  | 目标类型为车 | 2    |
| 注意：此算法模块输出目标类型唯一为人。   |    |        |      |

### AW\_TGT\_TRAJECT

| 结构定义  |       |           |        |
|---|-------|-----------|--------|
| <pre>typedef struct {     AW_S32 length;     AW_POINT_S points[AW_MAX_TRAJECT_LEN]; } AW_TGT_TRAJECT_S;</pre> |       |           |        |
| 结构说明  |       |           |        |
| 变量  | 名称    | 含义        | 取值范围   |
| length  | 轨迹点数目 | 有效的轨迹点数目  | 1~40   |
| points  | 轨迹点数组 | 用于存储轨迹点数组 | 图像像素坐标 |

### AW\_EVENT\_SET\_S

| 结构定义  |      |          |               |
|---|------|----------|---------------|
| <pre>typedef struct {     AW_S32 s32EventNum;     AW_EVENT_S astEvents[AW_MAX_EVT_NUM]; } AW_EVENT_SET;</pre> |      |          |               |
| 结构说明  |      |          |               |
| 变量  | 名称   | 含义       | 取值范围          |
| s32EventNum   | 事件数目 | 有效的事件数目  | 0~128         |
| astEvents   | 事件数组 | 用于存储事件数据 | 参考 AW_EVENT_S |

**AW\_EVENT\_S**

| 结构定义   |        |                |   |
|--|--------|----------------|---|
| <pre>typedef struct {     AW_U32 u32Type;     AW_U32 u32ID;     AW_U32 u32Status;     AW_U32 u32Zone;     AW_U32 u32TgtID;     AW_U8 u8Data[16]; } AW_EVENT_S;</pre> |        |                |   |
| 结构说明   |        |                |   |
| 变量   | 名称     | 含义             | 取值范围  |
| u32Type  | 事件类型   | 事件所属类型         | 参考<br>AW_AI_CVE_EVT_TYPE_E  |
| u32ID  | 事件标识   | 事件身份标识         | 32 位整数  |
| u32Status  | 事件状态   | 事件所处状态         | 参考 AW_EVT_STATUS  |
| u32Zone  | 事件发生区域 | 发生此事件的区域<br>编号 | 32 位整数  |
| u32TgtID   | 目标标识   | 关联目标的标识        | 32 位整数  |
| u8Data   | 事件私有数据 | 与类型相关的事件<br>数据 | 当 事 件 类 型 为<br>AW_AI_CVEEVT_TYPE_HUMANCOUNT<br>时 ， 请 参 考 数 据 结 构<br>AW_AI_CVE_HCNTEVT_DATA_HUMAN<br>COUNT |

**AW\_AI\_CVE\_EVT\_TYPE\_E**

| 结构定义  |  |
|---|--|
| <pre>typedef enum {     AW_AI_CVEEVT_TYPE_UNKNOWN           = 0000,     AW_AI_CVEEVT_TYPE_PERIMETER        = 1000,     AW_AI_CVEEVT_TYPE_HUMANCOUNT     = 1202, } AW_AI_CVE_EVT_TYPE_E;</pre> |  |
| 结构说明  |  |



| 变量                           | 名称 | 含义   | 取值范围 |
|------------------------------|----|------|------|
| AW_AI_CVEEVT_TYPE_PERIMETER  |    | 周界入侵 | 1000 |
| AW_AI_CVEEVT_TYPE_HUMANCOUNT |    | 人员计数 | 1202 |

### AW\_EVT\_STATUS\_E

| 结构定义   |     |       |      |
|--|-----|-------|------|
| <pre>typedef enum {     AW_EVT_STATUS_NOSTA = 0,     AW_EVT_STATUS_START = 1,     AW_EVT_STATUS_END = 2 } AW_EVT_STATUS_E;</pre> |     |       |      |
| 结构说明   |     |       |      |
| 变量   | 名称  | 含义    | 取值范围 |
| AW_EVT_STATUS_NOSTA  | 无状态 | 无状态事件 | 0    |
| AW_EVT_STATUS_START  | 开始  | 事件开始  | 1    |
| AW_EVT_STATUS_END  | 结束  | 事件结束  | 2    |

### AW\_AI\_CVE\_HCNTEVT\_DATA\_HUMANCOUNT

| 结构定义  |      |               |         |
|---|------|---------------|---------|
| <pre>typedef struct {     AW_LINE_S line;     AW_S16 angle;     AW_S16 tgtnum; } AW_AI_CVE_HCNTEVT_DATA_HUMANCOUNT;</pre> |      |               |         |
| 结构说明  |      |               |         |
| 变量  | 名称   | 含义            | 取值范围    |
| line  | 线段位置 | 目标穿越的线段。      | 图像坐标    |
| angle   | 方向角度 | 穿越线段是运动方向的角度。 | 0~360 度 |
| tgtnum  | 目标数目 | 目标的数目。        | 固定为 1   |

#### 3.2.2 人员计数算法配置数据类型

| 数据结构列表               | 功能描述     |
|----------------------|----------|
| AW_AI_CVE_HCNT_PARAM | 人员计数算法参数 |

|                            |          |
|----------------------------|----------|
| AW_AI_CVE_HCNT_PARAM_INOUT | 算法 IO 参数 |
| AW_AI_CVE_HCNT_PARAM_ALGO  | 算法处理参数   |
| AW_AI_CVE_HCNT_PARAM_APP   | 应用参数     |
| AW_AI_CVE_HCNT_PARAM_PERF  | 性能参数     |

### AW\_AI\_CVE\_HCNT\_PARAM

| 结构定义   |          |               |                                  |
|--|----------|---------------|----------------------------------|
| <pre>typedef struct {     AW_AI_CVE_HCNT_PARAM_INOUT  cParamInout;     AW_AI_CVE_HCNT_PARAM_ALGO2  cParamAlgo; } AW_AI_CVE_HCNT_PARAM;</pre> |          |               |                                  |
| 结构说明   |          |               |                                  |
| 变量   | 名称       | 含义            | 取值范围                             |
| cParamInout  | 算法 IO 参数 | 与图像输入输出相关的参数。 | 参考<br>AW_AI_CVE_HCNT_PARAM_INOUT |
| cParamAlgo   | 算法处理参数   | 与算法配置相关的参数。   | 参考<br>AW_AI_CVE_HCNT_PARAM_ALGO2 |

### AW\_AI\_CVE\_HCNT\_PARAM\_INOUT

| 结构定义  |         |                 |         |
|---|---------|-----------------|---------|
| <pre>typedef struct {     AW_S32 s32Width;     AW_S32 s32Height;     AW_U32 u32FrameTime; } AW_AI_CVE_HCNT_PARAM_INOUT;</pre> |         |                 |         |
| 结构说明  |         |                 |         |
| 变量  | 名称      | 含义              | 取值范围    |
| s32Width  | 图像宽度    | 所配置视频图像水平分辨率    | 16~1920 |
| s32Height   | 图像高度    | 所配置视频图像垂直分辨率    | 16~1080 |
| u32FrameTime  | 图像帧时间间隔 | 图像帧的时间长度（单位：毫秒） | 33, 40  |

### AW\_AI\_CVE\_HCNT\_PARAM\_ALGO

| 结构定义   |      |                              |                                 |
|--|------|------------------------------|---------------------------------|
| <pre>typedef struct {     AW_AI_CVE_HCNT_PARAM_APP cParamApp;     AW_AI_CVE_HCNT_PARAM_PERF cParamPerf; } AW_AI_CVE_HCNT_PARAM_ALGO;</pre> |      |                              |                                 |
| 结构说明   |      |                              |                                 |
| 变量   | 名称   | 含义                           | 取值范围                            |
| cParamApp  | 应用参数 | 用来设置分析功能、目标输出、视频异常检测、电子稳定等参数 | 参考<br>AW_AI_CVE_HCNT_PARAM_APP  |
| cParamPerf   | 性能参数 | 调整算法相关参数                     | 参考<br>AW_AI_CVE_HCNT_PARAM_PERF |

### AW\_AI\_CVE\_HCNT\_PARAM\_PERF

| 结构定义   |          |   |          |     |
|--|----------|---|----------|-----|
| <pre>typedef struct {     AW_S32 proc_interval;     AW_FLOAT head_radevi;     AW_S32 ciruniq_rad;     AW_S32 sensitivity; } AW_AI_CVE_HCNT_PARAM_PERF;</pre> |          |   |          |     |
| 结构说明   |          |   |          |     |
| 参数   | 名称       | 含义  | 取值范围     | 缺省值 |
| proc_interval  | 图像处理时间间隔 | 在资源足够的情况下决定算法的处理帧率，对于公交应用，必须保证至少每两帧能处理一帧。   | 0~200 毫秒 | 100 |
| head_radevi  | 头部半径容差比率 | 头部半径的检测范围为：<br>[head_radius*(1-head_radevi), head_radius*(1+head_radevi)];<br>取值越大，头部半径的检测范围越大， | 0.0~1.0  | 0.3 |

|             |          |                        |                         |    |
|-------------|----------|------------------------|-------------------------|----|
|             |          | 漏检越少，误检越多，同时也会增加算法运算量。 |                         |    |
| ciruniq_rad | 圆单一性半径比率 | 取值越大，则允许检测到的头部能够挨得越近。  | -100~100                | 80 |
| sensitivity | 检测灵敏度    | 值越低，灵敏度越高              | 0: 高,<br>1: 中,<br>2: 低。 | 0  |

**AW\_AI\_CVE\_HCNT\_PARAM\_APP**

| 结构定义  |          |               |                  |
|---|----------|---------------|------------------|
| <pre>typedef struct {     AW_S32 dim_w;     AW_S32 dim_h;     AW_AI_CVE_VANA_FUNCTION functions; } AW_HCNT_PARAM_APP;</pre> |          |               |                  |
| 结构说明  |          |               |                  |
| 变量  | 名称       | 含义            | 取值范围             |
| dim_w   | 配置图像的宽度  | 用户进行场景标定图像的宽度 | 16~1920          |
| dim_h   | 配置图像的宽度  | 用户进行场景标定图像的高度 | 16~1080          |
| functions   | 分析功能参数集合 | 用户设定分析功能相关参数  | 参考 AW_VANA_ZONES |

**AW\_AI\_CVE\_HCNT\_VANA\_FUNCTION**

| 结构定义   |        |    |      |
|--|--------|----|------|
| <pre>typedef struct {     AW_AI_CVE_VANA_PARAM_S defparam;     AW_S32 num_used;     AW_AI_CVE_VANA_FUNC funcs[AW_AI_CVE_VANA_ZONE_MAXNUM]; } AW_AI_CVE_HCNT_VANA_FUNCTION;</pre> |        |    |      |
| 结构说明   |        |    |      |
| 变量   | 名称     | 含义 | 取值范围 |
| defparam   | 缺省功能参数 |    | 参考   |

|          |        |                    |                           |
|----------|--------|--------------------|---------------------------|
|          |        |                    | AW_AI_CVE_VANA_PARAM_S    |
| num_used | 有效功能数目 | 设定有效功能（事件规则）的数目。   | 支持最大 32 个功能               |
| funcs    | 分析功能数组 | 分析功能数组，存储设定相关功能的参数 | 参考<br>AW_AI_CVE_VANA_FUNC |

### AW\_AI\_CVE\_HCNT\_VANA\_FUNC

| 结构定义  |      |                          |                              |
|---|------|--------------------------|------------------------------|
| <pre>typedef struct {     AW_S8 name[32];     AW_U32 enable;     AW_U32 level;     AW_U32 mode;     AW_AI_CVE_VANA_PARAM param; } AW_AI_CVE_HCNT_VANA_FUNC;</pre> |      |                          |                              |
| 结构说明  |      |                          |                              |
| 变量  | 名称   | 含义                       | 取值范围                         |
| name  | 规则名称 | 设定规则功能的名称                | 0(否),<br>1(是);<br>缺省值: 1     |
| enable  | 是否使能 | 规则功能是否开启                 | 0 (关闭);<br>1 (开启);<br>缺省值: 0 |
| level   | 警报级别 | 设定规则警报级别                 | 参考<br>AW_AI_CVE_ALARM_LEVEL  |
| mode  | 分析模式 | 规则启用分析的事件类型, 对应算法具体应用功能。 | 参考<br>AW_AI_CVE_HCNT_FUNCODE |
| param   | 分析参数 | 设定规则功能的分析参数              | 参考<br>AW_AI_CVE_VANA_PARAM   |

### AW\_AI\_CVE\_ALARM\_LEVEL

| 结构定义                      |  |  |  |
|---------------------------|--|--|--|
| <pre>typedef enum {</pre> |  |  |  |

```

        AW_AI_CVEALARM_LEVEL_ALARM      = 0,
        AW_AI_CVEALARM_LEVEL_PREALARM   = 1
    } AW_AI_CVE_ALARM_LEVEL;
    
```

## 结构说明

| 变量                            | 名称 | 含义 | 取值范围 |
|-------------------------------|----|----|------|
| AW_AI_CVEALARM_LEVEL_ALARM    |    | 报警 | 0    |
| AW_AI_CVEALARM_LEVEL_PREALARM |    | 预警 | 1    |

**AW\_AI\_CVE\_HCNT\_FUNCODE**

## 结构定义

```

typedef enum
{
    AW_AI_CVE_HCNT_FUNC_PERIMETER      = 0x00000001,
    AW_AI_CVE_HCNT_FUNC_HUMANCOUNT    = 0x00000002,
} AW_AI_CVE_HCNT_FUNCODE;
    
```

## 结构说明

| 变量                             | 名称 | 含义   | 取值范围       |
|--------------------------------|----|------|------------|
| AW_AI_CVE_HCNT_FUNC_PERIMETER  |    | 周界入侵 | 0x00000001 |
| AW_AI_CVE_HCNT_FUNC_HUMANCOUNT |    | 人员计数 | 0x00000002 |

**AW\_AI\_CVE\_HCNT\_VANA\_PARAM\_S**

## 结构定义

```

typedef struct
{
    AW_AI_CVE_HCNT_VANA_PARAM_PERIMETER      perimeter;
    AW_AI_CVE_HCNT_VANA_PARAM_HUMANCOUNT    humancount;
} AW_AI_CVE_HCNT_VANA_PARAM;
    
```

## 结构说明

| 变量         | 名称   | 含义                | 取值范围                                       |
|------------|------|-------------------|--|
| perimeter  | 周界入侵 | 人员进入禁止区域后报警。      | 参考<br>AW_AI_CVE_HCNT_VANA_PARAM_PERIMETER  |
| humancount | 人员计数 | 人员穿越设置的线段后输出计数事件。 | 参考<br>AW_AI_CVE_HCNT_VANA_PARAM_HUMANCOUNT |

**AW\_AI\_CVE\_VANA\_PARAM**

| 结构定义  |      |                   |  |
|---|------|-------------------|--|
| <pre>typedef union {     AW_AI_CVE_HCNT_VANA_PARAM_PERIMETER           perimeter;     AW_AI_CVE_HCNT_VANA_PARAM_HUMANCOUNT         humancount; } AW_AI_CVE_HCNT_VANA_PARAM;</pre> |      |                   |  |
| 结构说明  |      |                   |  |
| 变量  | 名称   | 含义                | 取值范围                                       |
| perimeter   | 周界入侵 | 人或者车进入禁止区域后报警。    | 参考<br>AW_AI_CVE_VANA_PARAM_PERIMETER       |
| humancount  | 人员计数 | 人员穿越设置的线段后输出计数事件。 | 参考<br>AW_AI_CVE_HCNT_VANA_PARAM_HUMANCOUNT |

### AW\_AI\_CVE\_HCNT\_VANA\_PARAM\_PERIMETER

| 结构定义  |      |              |        |
|---|------|--------------|--------|
| <pre>typedef struct {     AW_POLYGON_S region; } AW_AI_CVE_HCNT_VANA_PARAM_PERIMETER;</pre> |      |              |        |
| 结构说明  |      |              |        |
| 变量  | 名称   | 含义           | 取值范围   |
| region  | 分析区域 | 在此区域内的目标产生报警 | 图像像素坐标 |

### AW\_AI\_CVE\_HCNT\_VANA\_PARAM\_HUMANCOUNT

| 结构定义   |      |            |        |
|--|------|------------|--------|
| <pre>typedef struct {     AW_AI_CVE_CUT_LINE cutline; } AW_AI_CVE_VANA_PARAM_LOITER;</pre> |      |            |        |
| 结构说明   |      |            |        |
| 变量   | 名称   | 含义         | 取值范围   |
| cutline  | 切面线段 | 用于目标计数的线段。 | 图像像素坐标 |

### AW\_AI\_CVE\_CUT\_LINE

| 结构定义  |  |  |  |
|---|--|--|--|
| <pre>typedef struct {     AW_AI_CVE_CUT_LINE cutline; } AW_AI_CVE_CUT_LINE;</pre> |  |  |  |

|   |      |            |        |
|---|------|------------|--------|
| <pre>typedef struct {     AW_LINE_S line; } AW_AI_CVE_CUT_LINE;</pre> |      |            |        |
| 结构说明  |      |            |        |
| 变量  | 名称   | 含义         | 取值范围   |
| line  | 线段位置 | 用于目标计数的线段。 | 图像像素坐标 |

### 2.2.2. 景深标定算法配置数据类型

| 数据结构列表                | 功能描述   |
|-----------------------|--------|
| AW_AI_CVE_CLBR_PARAM  | 景深标定参数 |
| AW_AI_CVE_FDEPTH_COEF | 景深系数   |
| AW_AI_CVE_FDZ_LINES   | 标定线段集合 |

#### AW\_AI\_CVE\_CLBR\_PARAM

|  |        |           |                             |
|--|--------|-----------|-----------------------------|
| 结构定义   |        |           |                             |
| <pre>typedef struct {     AW_S32 dim_w;     AW_S32 dim_h;     AW_AI_CVE_FDZ_LINES fdlines;     AW_AI_CVE_FDEPTH_COEF coef; } AW_AI_CVE_CLBR_PARAM;</pre> |        |           |                             |
| 结构说明   |        |           |                             |
| 变量   | 名称     | 含义        | 取值范围                        |
| img_w  | 标定图像宽度 | 用户标定图像的宽度 | 16~1920                     |
| img_h  | 标定图像高度 | 用户标定图像的高度 | 16~1080                     |
| fdlines  | 标定数据   | 标定景深线段集合  | 参考<br>AW_AI_CVE_FDZ_LINES   |
| coef   | 景深系数   | 换算的景深系数   | 参考<br>AW_AI_CVE_FDEPTH_COEF |

#### AW\_AI\_CVE\_FDEPTH\_COEF

|                |  |  |  |
|----------------|--|--|--|
| 结构定义           |  |  |  |
| typedef struct |  |  |  |



```

{
    float cx;
    float cy;
    float cf;
} AW_AI_CVE_FDEPTH_COEF;
    
```

## 结构说明

| 变量 | 名称     | 含义                 | 取值范围 |
|----|--------|--------------------|------|
| cx | 水平方向系数 | 水平方向图像像素与实际宽度换算的系数 | 浮点数  |
| cy | 竖直方向系数 | 竖直方向图像像素与实际高度换算的系数 | 浮点数  |
| cf | 偏置系数   | 景深换算的偏置系数          | 浮点数  |

**AW\_AI\_CVE\_FDZ\_LINES**

## 结构定义

```

typedef struct
{
    AW_S32 num_used;
    AW_AI_CVE_FDZ_LINE lines[AW_AI_CVE_MAXNUM_FDEPTH_LINE];
} AW_AI_CVE_FDZ_LINES;
    
```

## 结构说明

| 变量       | 名称     | 含义                 | 取值范围                     |
|----------|--------|--------------------|--------------------------|
| num_used | 有效线段数目 | 用户标定景深时，有效的线段数目    | 3~8                      |
| lines    | 线段数组   | 用于存储用户标定景深，标定线段的数据 | 参考<br>AW_AI_CVE_FDZ_LINE |

**AW\_AI\_CVE\_FDZ\_LINE**

## 结构定义

```

typedef struct
{
    AW_S32 type;
    AW_LINE_S ref_line;
    AW_S32 ref_length;
} AW_AI_CVE_FDZ_LINE;
    
```

## 结构说明

| 变量         | 名称   | 含义               | 取值范围                       |
|------------|------|------------------|----------------------------|
| type       | 线段类型 | 表示标定线段类型，是水平还是竖直 | 参考<br>AW_AI_CVE_FLINE_TYPE |
| ref_line   | 线段坐标 | 用于存储标定线段的坐标      | 图像像素坐标                     |
| ref_length | 线段长度 | 标定线段对应实际的长度      | 单位：厘米                      |

### AW\_AI\_CVE\_FDZ\_TYPE

| 结构定义   |    |      |      |
|--|----|------|------|
| <pre>typedef enum {     AW_AI_CVE_FLINE_HORI = 0,     AW_AI_CVE_FLINE_VERT } AW_AI_CVE_FLINE_TYPE;</pre> |    |      |      |
| 结构说明   |    |      |      |
| 变量   | 名称 | 含义   | 取值范围 |
| AW_AI_CVE_FLINE_HORI   |    | 水平线段 | 0    |
| AW_AI_CVE_FLINE_VERT   |    | 竖直线段 | 1    |

## 2.3. 错误码

算法库 API 错误码如下表 2-1 所示：

| 错误代码       | 宏定义                         | 描述         |
|------------|-----------------------------|------------|
| 0x00000001 | AW_AI_CVE_ERR_OK            | 算法运行正常，无错误 |
| 0xA00A0001 | AW_AI_CVE_ERR_NULL_PTR      | 函数参数有空指针   |
| 0xA00A0002 | AW_AI_CVE_ERR_ILLEGAL_PARAM | 参数超出合法范围   |
| 0xA00A0003 | AW_AI_CVE_ERR_INVALID_ADDR  | 传递非法地址     |
| 0xA00A0004 | AW_AI_CVE_ERR_NOT_CONFIG    | 未进行参数配置    |
| 0xA00A0005 | AW_AI_CVE_ERR_NO_LISENCE    | 授权失败       |
| 0xA00A0006 | AW_AI_CVE_ERR_INVALID_IMAGE | 输入了无效的图像   |

表 2-1 算法库 API 错误

Allwinertech

## 3. 开发应用

### 关于本章

本章描述内容如下表所示。

| 标题       | 内容                |
|----------|-------------------|
| 3.1 调用流程 | 介绍运动人员计数算法库的调用流程。 |
| 3.2 实例介绍 | 介绍运动人员计数算法的开发实例。  |

表 3-1 本章内容说明

AllwinnerTech

### 3.1. 调用流程

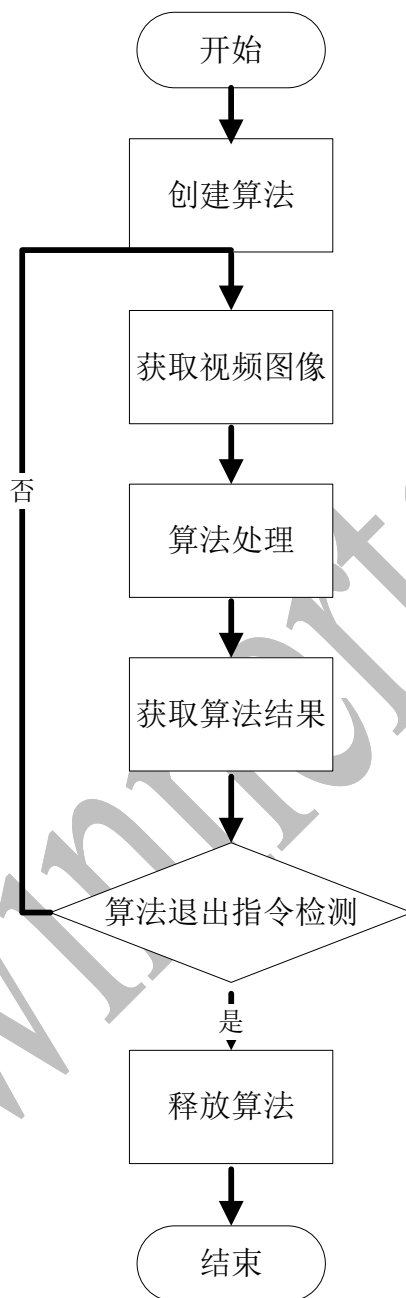


图 3-1 运动人员计数算法库的调用流程

### 3.2. 实例介绍

本实例完成运动人员计数算法库的参数配置、接口 API 调用。

| 文件类型 | 文件列表 | 备注 |
|------|------|----|
|------|------|----|

|       |   |  |
|-------|---|--|
| 接口头文件 | aw_ai_common_type.h<br>aw_ai_cve_base_type.h<br>aw_ai_cve_clbr_config.h<br>aw_ai_cve_clbr_interface.h<br>aw_ai_cve_hcnt_config.h<br>aw_ai_cve_hcnt_result.h<br>aw_ai_cve_hcnt_interface.h |  |
| 算法库文件 | Linux ARM   | libzcvcore_cve.a<br>libzcvxml_cve.a<br>libzvacore_cve.a<br>libzvaclbr_cve.a<br>libzvahcnts_cve.a<br>libzvahcnt_cve.a<br>libclbr_cve.a<br>libhcnts_cve.a<br>libhcnt_cve.a |
| 参数文件  | params_cvehcnt_perf.xml,<br>params_cveclbr.xml  |  |
| 样例代码  | sample_cve_HCNT.c<br>sample_vi2HCNT.c   |  |

表 3-1 运动人员计数算法实例相关文件列表及说明

【参考代码】详见“sample\_cve\_HCNT”

注意：应用时需要将 sample\_cve\_HCNT 目录下的 HCNTParam 文件夹内的 xml 文件与生成可执行文件放置同一文件夹下，文件夹要有读写权限；同时在编译该例程时需要修改 middleware/config/mpp\_config.mk 文件：MPPCFG\_HCNT = Y

## 4. 算法资源需求指标

### 关于本章

本章描述内容如下表所示。

| 标题         | 内容                         |
|------------|----------------------------|
| 4.1 算法资源需求 | 介绍算法所需的内存需求，代码空间需求，内存耗时指标。 |

表 4-1 本章内容说明

### 4.1. 算法资源需求

下表表示算法库资源需求说明。

| 算法资源需求 | 资源占用大小  |      |      | 说明               |
|--------|---|------|------|------------------|
| 内存需求   | 4.5M（按 QCIF 评估）   |      |      | 人员计数算法内存需求       |
| 代码空间需求 | ibzcvcore_cve.a<br>libzcvxml_cve.a<br>libzvacore_cve.a<br>libzvaclbr_cve.a<br>libzvahcnts_cve.a<br>libzvahcnt_cve.a<br>libclbr_cve.a<br>libhcnts_cve.a<br>libhcnt_cve.a | 416k | 416k | linuxARM<br>代码空间 |

表 4-2 人员计数算法资源需求说明