

# ArcSoft Face Tracking

---

开发指导文档

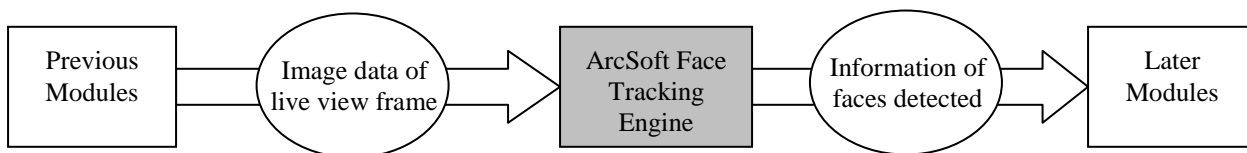
## 目录

<b>ARCSOFT FACE TRACKING .....</b>	<b>1</b>
<b>概述.....</b>	<b>3</b>
1.1. 运行环境 .....	3
1.2. 系统要求 .....	3
1.3. 支持颜色空间格式 .....	3
<b>类描述.....</b>	<b>4</b>
2.1. CLASS AFT_FSDKFACE.....	4
2.1.1. 功能描述.....	4
2.1.2. 构造函数.....	4
2.1.3. 成员函数.....	4
2.2. CLASS AFT_FSDKERROR .....	5
2.2.1. 功能描述.....	5
2.2.2. 构造函数.....	5
2.2.3. 成员函数.....	5
2.2.4. 类常量.....	5
2.3. CLASS AFT_FSDKVERSION.....	6
2.3.1. 功能描述.....	6
2.3.2. 构造函数.....	6
2.3.3. 成员函数.....	6
2.4. CLASS AFD_FSDKENGINE.....	6
2.4.1. 功能描述.....	6
2.4.2. 构造函数.....	6
2.4.3. 成员函数.....	6
2.4.4. 类常量.....	7
<b>示例代码.....</b>	<b>9</b>

## 概述

---

虹软人脸跟踪引擎工作流程图:



---

### 1.1. 运行环境

- Android ARM 32

---

### 1.2. 系统要求

- Android 5.x 或以上

---

### 1.3. 支持颜色空间格式

支持图像的颜色格式: NV21

# 类描述

---

## 2.1. Class AFT\_FSDKFace

### 2.1.1. 功能描述

这个类用来保存检测到的人脸信息

### 2.1.2. 构造函数

#### 2.1.2.1. AFT\_FSDKFace(AFT\_FSDKFace self)

参数:

AFT\_FSDKFace self

类对象，创建的对象和传入的对象数据一致

说明： 在一个 init/unit 过程中， 只支持相同分辨率的图像数据

#### 2.1.2.2. AFT\_FSDKFace()

### 2.1.3. 成员函数

#### 2.1.3.1. Rect getRect()

返回值:

保存的人脸框(Rect(left, top, right, bottom))

#### 2.1.3.2. int getDegree()

返回值:

保存的人脸角度(AFT\_FOC\_0, AFT\_FOC\_90 等)

#### 2.1.3.3. String toString()

返回值:

格式化人脸信息的字符串 (String("Rect(left, top – right, bottom), Degree"))

#### 2.1.3.4. AFT\_FSDKFace clone()

返回值:

本对象的拷贝

## 2.2. Class AFT\_FSDKError

### 2.2.1. 功能描述

这个类用来保存函数执行的错误信息.

### 2.2.2. 构造函数

#### 2.2.2.1. AFD\_FSDKError()

### 2.2.3. 成员函数

#### 2.2.3.1. int getCode()

返回值:

错误码值 (MOK, MERR\_INVALID\_PARAM 等)

### 2.2.4. 类常量

#### 2.2.4.1. 错误码

引擎返回的错误值定义

MOK	0	成功
MERR_BASIC_BASE	0x0001	基础错误起始值
MERR_UNKNOWN	0x0001	未知错误
MERR_INVALID_PARAM	0x0002	参数错误
MERR_UNSUPPORTED	0x0003	输入了引擎不支持的参数或者数据
MERR_NO_MEMORY	0x0004	内存不足
MERR_BAD_STATE	0x0005	状态错误 (未初始化就调用了接口)
MERR_BUFFER_OVERFLOW	0x0009	内存上溢
MERR_BUFFER_UNDERFLOW	0x000a	内存下溢
MERR_FSDK_BASE	0x7000	校验错误起始值
MERR_FSDK_INVALID_APP_ID	0x7001	非法 APPID
MERR_FSDK_INVALID_SDK_ID	0x7002	非法 SDKID
MERR_FSDK_INVALID_ID_PAIR	0x7003	SDKKEY 不是于当前 APPID 名下的

MERR_FSDK_MISMATCH_ID_AND_SDK	0x7004	SDKKEY 不是当前 SDK 所支持的
MERR_FSDK_SYSTEM_VERSION_UNSUPPORTED	0x7005	不支持的系统版本
MERR_FSDK_LICENCE_EXPIRED	0x7006	SDK 过期

---

## 2.3. Class AFT\_FSDKVersion

### 2.3.1. 功能描述

这个类用来保存版本信息

### 2.3.2. 构造函数

#### 2.3.2.1. AFT\_FSDKVersion ()

### 2.3.3. 成员函数

#### 2.3.3.1. String toString()

返回值:

包含所有版本信息的字符串

---

## 2.4. Class AFD\_FSDKEngine

### 2.4.1. 功能描述

这个类具体实现了人脸跟踪的功能

### 2.4.2. 构造函数

#### 2.4.2.1. AFT\_FSDKEngine()

### 2.4.3. 成员函数

#### 2.4.3.1. AFT\_FSDKError AFT\_FSDK\_InitialFaceEngine(String appid, String sdkkey, int oriensPriority, int scale, int maxFaceNum)

这个函数功能为初始化引擎，设置检测角度、范围，数量。创建对象后，必须先于其他成员函数调用，否则其他成员函数会返回 MERR\_BAD\_STATE。

参数:

String appid	用户申请 SDK 时获取的 App Id
String sdkkey	用户申请 SDK 时获取的 SDK Key
int orientsPriority	指定检测的角度(AFT_OPF_0_ONLY, AFT_OPF_90_ONLY 等)
int scale	指定支持检测的最小人脸尺寸,有效值范围[2,32],推荐值 16
int maxFaceNum	最多能检测到的人脸个数,有效值范围[1,50],推荐值 25

返回值:

返回 AFT\_FSDKError 对象, 具体错误信息参考 Class AFT\_FSDKError

#### 2.4.3.2. AFT\_FSDKError AFT\_FSDK\_FaceFeatureDetect(byte[] data, int width, int height, int format, List<AFT\_FSDKFace> list)

这个函数功能为检测输入的图像中存在的人脸, 输出结果和初始化时设置的参数有密切关系。

参数:

byte[] data	输入的图像数据
int width	图像宽度
int height	图像高度
int format	图像格式
List<AFD_FSDKFace> list	检测到的人脸会 add 到此 list.注意 AFD_FSDKFace 对象引擎内部重复使用,如需保存,请 clone 一份 AFD_FSDKFace 对象或另外保存

返回值:

返回 AFT\_FSDKError 对象, 具体错误信息参考 Class AFT\_FSDKError

#### 2.4.3.3. AFT\_FSDKError AFT\_FSDK\_UninitialFaceEngine()

这个函数功能为销毁引擎, 释放占用的内存资源。

返回值:

返回 AFT\_FSDKError 对象, 具体错误信息参考 Class AFT\_FSDKError

#### 2.4.3.4. AFT\_FSDKError AFT\_FSDK\_GetVersion(AFT\_FSDKVersion version)

这个函数功能为获取版本信息

参数:

AFT_FSDKVersion version	版本信息对象 参考 Class AFT_FSDKVersion
-------------------------	---------------------------------

### 2.4.4. 类常量

#### 2.4.4.1. 颜色格式

支持的颜色格式及其对齐规则

CP_PAF_NV21	0x802	8-bit Y 层, 之后是 8-bit 的 2x2 采样的 U, V 交织层
-------------	-------	-----------------------------------------

#### 2.4.4.2. 检测角度

引擎初始化时输入的目标检测角度

AFT_OPF_0_ONLY	0x1	检测 0 度方向
AFT_OPF_90_ONLY	0x2	检测 90 度方向
AFT_OPF_270_ONLY	0x3	检测 270 度方向
AFT_OPF_180_ONLY	0x4	检测 180 度方向
AFT_OPF_0_HIGHER_EXT	0x5	检测 0, 90, 180, 270 四个方向,0 度更优先

#### 2.4.4.3. 人脸角度

检测结果中的人脸角度

AFT_FOC_0	0x1	0 度
AFT_FOC_90	0x2	90 度
AFT_FOC_270	0x3	270 度
AFT_FOC_180	0x4	180 度



## 示例代码

---

```
AFT_FSDKInterface engine = new AFT_FSDKEngine();

// 用来存放检测到的人脸信息列表
List<AFT_FSDKFace> result = new ArrayList<>();

//初始化人脸跟踪引擎，使用时请替换申请的 APPID 和 SDKKEY
AFT_FSDKError err = engine.AFT_FSDK_InitialFaceEngine("APPID", "SDKKEY",
AFT_FSDKEngine.AFT_OPF_0_HIGHER_EXT, 16, 5);
Log.d("com.arcsoft", "AFT_FSDK_InitialFaceEngine =" + err.getCode());

//输入的 data 数据为 NV21 格式（如 Camera 里 NV21 格式的 preview 数据），其中 height 不能为奇
数，人脸跟踪返回结果保存在 result。
err = engine.AFT_FSDK_FaceFeatureDetect(data, width, height, AFT_FSDKEngine.CP_PAF_NV21,
result);
Log.d("com.arcsoft", "AFT_FSDK_FaceFeatureDetect =" + err.getCode());
Log.d("com.arcsoft", "Face=" + result.size());
for (AFT_FSDKFace face : result) {
    Log.d("com.arcsoft", "Face:" + face.toString());
}

//销毁人脸跟踪引擎
err = engine.AFT_FSDK_UninitialFaceEngine();
Log.d("com.arcsoft", "AFT_FSDK_UninitialFaceEngine =" + err.getCode());
```