

# **VCamera SDK**

# **Android 用户手册**

## **v2.0**

发布日期：2014-09-11

炫一下（北京）科技有限公司  
（版权所有，翻版必究）

## 第1章 概述

VCamera SDK Android 版（短视频拍摄 SDK）是炫一下（北京）科技有限公司官方推出的 Android 平台使用的软件开发工具包，为 Android 开发者提供简单、快捷的接口，帮助开发者实现 Android 平台上的短视频应用开发。

SDK 提供视频录制、后滤镜、炫酷 MV 主题、后期强大的 FFMPEG 命令行支持，可实现水印、音量控制等诸多功能。

本文档面向所有使用该 SDK 的开发人员、测试人员，要求有一定的 Android 编程经验。

## 第2章 SDK 功能说明

SDK 提供拍摄录制、分段回删、后滤镜、静态/动态水印、声音合成、本地图片、视频导入等功能。

## 第3章 SDK 使用说明

### 3.1 运行环境

- Android 4.0 及以上的所有系统（注意 Demo 中使用了 TextureView，要支持更低版本的 Android 请自行修改替换成 SurfaceView）
- 仅支持 ARM v7 的 CPU

### 3.2 初始化

```
VCamera.initialize(this);
```

建议在 Application 的 onCreate 中调用，参考 Demo 中的 VCameraDemoApplication。

## 第4章 开发前准备

如果您新开发一个的项目，我建议您直接在 Demo 上进行修改，修改 AndroidManifest.xml 中的 package 即可。

### 4.1 添加 SDK 到 App 工程

- 创建一个 Android 工程
- 将 VitamioRecorderLibrary 以 Android Library 方式添加工程引用

- 从 Demo 中拷贝需要的类、图片等资源到 App 工程下

#### 4.2 权限声明

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-feature android:name="android.hardware.camera"
    android:required="true" />
<uses-feature android:name="android.hardware.camera.autofocus"
    android:required="false" />
<uses-feature android:glEsVersion="0x00020000"
    android:required="true" />
```

#### 4.3 初始化 SDK

```
//设置拍摄视频缓存路径
if (DeviceUtils.isZte()) {
    if
(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_
DCIM).exists()) {
        VCamera.setVideoCachePath(Environment.getExternalStoragePublicDir
ectory(Environment.DIRECTORY_DCIM) + "/Camera/VCameraDemo/");
    } else {
        VCamera.setVideoCachePath(Environment.getExternalStoragePublicDir
ectory(Environment.DIRECTORY_DCIM).getPath().replace("/sdcard/",
"/sdcard-ext/") + "/Camera/VCameraDemo/");
    }
} else {
    VCamera.setVideoCachePath(Environment.getExternalStoragePublicDir
ectory(Environment.DIRECTORY_DCIM) + "/Camera/VCameraDemo/");
}
//开启 log 输出,ffmpeg 输出到 logcat
VCamera.setDebugMode(true);
//初始化拍摄 SDK, 必须
```

```
VCamera.initialize(this);
```

#### 4.4 后期处理

UtilityAdaptert 提供 FFmpegRun 方法用于执行 ffmpeg 命令

## 第5章 MV 主题

### 5.1 MV 主题介绍

新版 SDK 提供强大的 MV 主题，支持视频素材、高斯模糊、分屏、滚屏、转场、放大、扭曲、重影、强光、融合、音乐等特效叠加。

SDK 内置 8 个 MV 主题 10 个滤镜。

### 5.2 内置 MV 主题包介绍

内置主题包在 Sample 工程的 assets/MusicVideoAssets.zip，解压即可。  
以 Dreamworld 为例。

文件名	备注
Dreamworld.json	MV 主题的配置文件。
Dreamworld.png	MV 主题的图标
frame.ts	MV 主题叠加的视频
loader	MV 主题的加密脚本
Lover.mp3	MV 主题的音乐
s	MV 主题的加密脚本

### 5.2 替换片尾

替换 Common/ tail.mp4 即可，注意片尾不要太长，最好也是 2 秒内。  
片尾文字在 ThemeHelper.updateVideoAuthorLogo 这个方法，可自行修改。

### 5.3 替换其他元素

MV 主题的 loader 和 s 不能替换，只能替换其他 ts、mp4、png、mp3 等资源，需要定制主题请参考后续增值服务。mp4 转 ts 命令行（15 帧兼容低端机），PC 上转好替换就行：

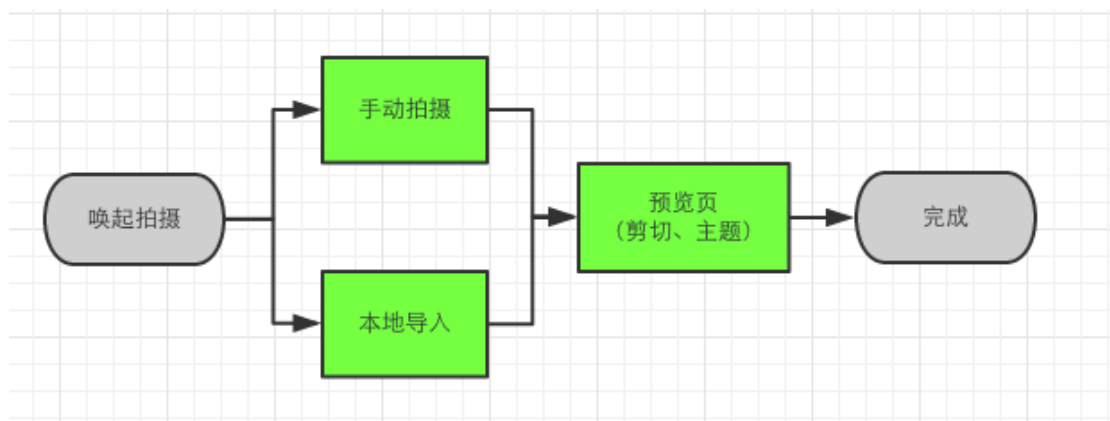
```
ffmpeg -y -i in.mp4 -pix_fmt yuv420p -r 15 -vcodec libx264 -profile:v  
main -f mpegts overlay.ts
```

#### 5.4 旧版兼容

新版的 SDK 用老版本的预览界面来实现功能也没有问题(单图片、多图、gif 等主题)。

## 第6章 开发指南

### 6.1 拍摄流程图



上面是一般的拍摄流程，目前 Demo 只有手动拍摄（后续版本会补上本地导入），开发者可自行用 ffmpeg 命令行实现其他的功能和流程。

### 6.2 拍摄录制模式

SDK 支持三种录制模式：

#### 6.2.1 默认录制模式（MediaRecorderNative）

默认模式（也是秒拍当前使用的模式），生成临时视音频文件（实时旋转和裁剪画面），最后合并 ts 输出 mp4。

#### 6.2.2 系统录制模式（MediaRecorderSystem）

调用了系统的 MediaRecorder 的对象，录制会有滴的一声，拍多段会卡顿一下，适合低端机。

#### 6.2.3 自定义录制模式（MediaRecorderCustom）

未完成。输出的是 640x480 的 yuv 数据，后期处理需要自己增加裁剪和旋转的命令，例如-vf "transpose=1[tran];[tran]crop=480:480:0:0"。

### 6.3 修改录制参数（默认录制模式）

参考 `MediaRecorderNative.startRecord` 源码，支持传入 `ffmpeg` 的 `vf` 参数，默认参数（不传）：`"-vf \"transpose=1,crop=480:480:0:0\" \":\"-vf \"transpose=2,hflip,crop=480:480:0:0\"` 分别是前置和后置。注意保持预览和输出的视频一致。

### 6.4 导入视频

参考 `ImportVideoActivity` 这个类。

## 第7章 ffmpeg 命令行

极力推荐开发者在 PC 上安装 `ffmpeg` 环境，以便测试 `ffmpeg` 命令行，测试通过后再在手机上测试使用。

### 7.1 视频叠加水印（建议添加与视频大小一致并且透明的水印，如 480x480）

//单水印

```
ffmpeg -i in.mp4 -i watermark1.png -filter_complex "overlay=x=0:y=0" -f mp4 out.mp4
```

//动态水印，2 张图片间隔 200 毫秒

```
ffmpeg -i in.mp4 -i watermark1.png -i watermark2.png -filter_complex "overlay=x= 'if(eq(floor(mod(t*5,2)),0), 0, -500)':y=0, overlay=x= 'if(eq(floor(mod(t*5,2)),1), 0, -500)':y=0" -f mp4 out.mp4
```

//视频叠加 gif

```
ffmpeg -y -i in.mp4 -i in.gif -filter_complex "overlay=0:0" -t 10 -acodec copy out.mp4
```

### 7.2 视频裁剪

//裁剪 0 秒到 10 秒视频

```
ffmpeg -i in.mp4 -ss 0 -t 10 -vcodec copy -acodec copy out.mp4
```

### 7.3 格式转换

//yuv 转 mp4

```
ffmpeg -y -f rawvideo -pix_fmt yuv420p -s 480x480 -r 15 -i in.yuv -vcodec libx264 -profile:v baseline -preset ultrafast out.mp4
```

//静态图片转 mp4

```
ffmpeg -y -loop 1 -f image2 -i Goddess.png -vcodec mpeg4 -r 15 -t 10
Goddess.mp4
```

```
//ts 转 mp4
```

```
ffmpeg -y -i in.ts -vcodec copy -acodec copy -vbsf h264_mp4toannexb
out.mp4
```

```
//mp4 转 ts
```

```
ffmpeg -y -i in.mp4 -vcodec copy -acodec copy -absf aac_adtstoasc -f mp4
-movflags faststart out.ts
```

```
//gif 转 ts
```

```
ffmpeg -y -i
http://ww4.sinaimg.cn/mw1024/89cfdb1bgw1ek0ig2zx6gg207h05me81.gif
-vf "scale=480:-1,pad=480:480:0:240-ih/2" -pix_fmt yuv420p -vcodec libx264
-vbsf h264_mp4toannexb 0.ts
```

可以先将多个 gif 先转成 ts，然后用 `-i "concat:0.ts|1.ts"` 合并，注意转出来的视频是不带音频的，这个需要自己加一个空音频或者音乐，否则可能无法播放该视频。`pad` 参数可以让视频上下留黑，类似电影的效果。

#### 7.4 其他

```
//加速视频
```

```
ffmpeg -i input.avi -vf "setpts=0.5*PTS" output.avi
```

```
//放慢视频
```

```
ffmpeg -i input.avi -vf "setpts=2.0*PTS" output.avi
```

```
//删除视频文件的旋转信息
```

```
-metadata:s:v rotate="\\"
```

```
//提取视频的 yuv 和 pcm 数据
```

```
ffmpeg -i in.mp4 -ss 0 -t 10 -vn -acodec pcm_s16le -f s16le -ar 44100
-ac 1 a.pcm -ss 0 -t 10 -an -vcodec rawvideo -f rawvideo -s 640x480 -pix_fmt
nv21 a.yuv
```

```
//下载视频
```

```
ffmpeg -y -i "http://xxx.xxx.com/test.mp4" -vcodec copy -acodec copy
out.mp4
```

还可以加 `-ss -t` 参数截取视频，如果无法 `copy`，可以重新转码

```
-vcodec libx264 -profile:v main -acodec libfdk_aac -ar 44100 -ac 1 -b:a 64k
```

//转码参数

-profile:v baseline -preset ultrafast 加这个转码速度快但质量差

-profile:v main 加这个转码速度慢但质量很好

## 第8章 注意

### a) 如果查看 ffmpeg 执行的 log

加-d stdout -loglevel verbose 即可在 logcat 中看到执行的命令行输出，例如：

```
ffmpeg -d stdout -loglevel verbose -i in.mp4 -ss 0 -t 10 -vcodec copy  
-acodec copy out.mp4
```

### b) 没有生成输出文件的问题

注意检测 AndroidManifest.xml 是否漏了写磁盘的权限。

注意 ffmpeg 的 log 输出，一般会给出错误提示。

注意文件是否已经生成，需要删除或者加-y 参数。

注意相册里不会马上显示新生成的视频，除非你手动向系统相册新增记录

### c) 视频预览画面变形/压扁的问题

经测试绝大部分设备没有这个问题，已知"GT-N7100", "GT-I9308"这两块设备会出现这种情况。导致出现这种情况的原因是不支持自动对焦参数，请及时和我们联系。此外还要排查一下是不是自己的代码改的有问题，请装一下秒拍看看是不是也有同样的问题，调整预览尺寸大小时必须按比例调整，否则也会变形。

## 第9章 FAQ

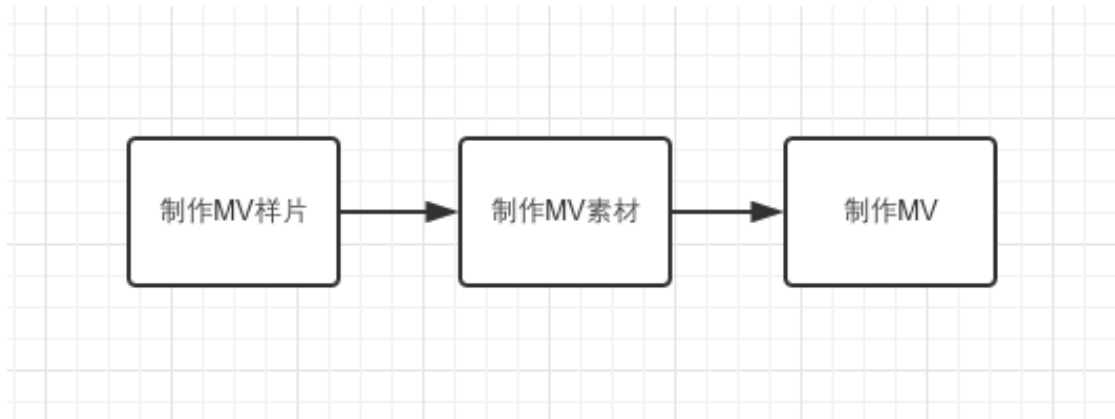
### ● SDK 是否收费？

个人企业均免费（商务 sales@yixia.com，电话座机 010-6482 8682，商务 QQ: 121700313）

### ● 付费增值服务

我们提供定制 MV 炫酷主题的付费服务，根据主题的难易程度收费不同，制作流程如下：





前面两个步骤由设计师完成，这一步可以由您提供，最后制作 MV 主题由我们来实现，费用按 5000RMB/天计算，最少 1 天。例如迷幻这个主题需要 3 天，梦境、风潮、Shinging 这类型的主题需要 2 天时间，Love 主题只需要 1 天，具体细节找商务详谈。

- 最终解释权

最终解释权归炫一下（北京）科技有限公司所有。

- so 的包很大，能优化减少吗？

我们也意识到了这个问题，后续版本会进行优化。

- 拍摄界面能和秒拍做成一样的吗？

能。欢迎为用户提供一致的体验！但仅限于拍摄的相关界面。

- 什么时候推出 IOS 版本的拍摄 SDK？

如果你已经在上线的产品中使用了 Android 版的拍摄上传 SDK，可以联系我们申请 IOS 版的拍摄 SDK 试用。

- 有讨论群吗？

QQ 群：281434005，申请加入写 VCamera，欢迎加入！