

音视频终端 SDK(腾讯云视立方)

美颜特效

产品文档



腾讯云

【 版权声明 】

©2013–2022 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

美颜特效

腾讯特效引擎

SDK 功能说明

SDK 集成指引

iOS

Android

美颜特效

腾讯特效引擎

最近更新时间：2022-03-17 16:11:00

腾讯云视立方·腾讯特效引擎（Tencent Effect）SDK 是音视频终端 SDK（腾讯云视立方）的重要组成部分，基于优图精准的 AI 能力和天天 P 图丰富的实时特效处理，为各类视频处理场景提供丰富的产品能力。腾讯特效 SDK 支持与腾讯云直播 SDK、短视频 SDK、音视频通话 SDK 等音视频终端产品集成，高效便捷，优势尤为明显。



SDK 功能说明

最近更新时间：2022-01-20 15:46:27

腾讯特效 SDK 共有 10 个套餐，10 个套餐分为 2 个系列：[A 系列基础套餐](#) 和 [S 系列高级套餐](#)。不同系列的不同套餐对应不同功能，各套餐支持的功能详情如下表。更多下载说明请参见 [SDK 下载](#)。

A 系列基础套餐功能

A 系列基础套餐提供通用美型功能，适用于对脸部美颜调整要求较低的客户。

套餐功能		套餐编号					
		A1-01	A1-02	A1-03	A1-04	A1-05	A1-06
基础功能	基础美颜 美白、磨皮、红润	✓	✓	✓	✓	✓	✓
	画面调整 对比度、饱和度、清晰度	✓	✓	✓	✓	✓	✓
	基础美型 大眼、瘦脸（自然、女神、英俊）	✓	✓	✓	✓	✓	✓
	滤镜 （默认 10 款通用滤镜）	✓	✓	✓	✓	✓	✓
可拓展功能	贴纸 （赠送 10 款可选 2D 通用贴纸）	-	✓	✓	✓	✓	✓
	通用美型 SDK （窄脸/下巴/发际线/瘦鼻）	-	-	✓	-	-	-
	手势识别 （赠送 1 款指定手势贴纸）	-	-	-	✓	-	-
	人像分割 （赠送 3 款指定分割贴纸）	-	-	-	-	✓	-
	美妆 （赠送 3 款指定整妆）	-	-	-	-	-	✓
SDK 下载	iOS	下载	下载	下载	下载	下载	下载
	Android	下载	下载	下载	下载	下载	下载

S 系列高级套餐功能

S 系列高级套餐提供高级美型功能（包括特效贴纸和美妆），适用于对脸部美颜调整需求较高的客户。

套餐功能		套餐编号			
		S1 - 01	S1 - 02	S1 - 03	S1 - 04
基础功能	基础美颜 美白、磨皮、红润	✓	✓	✓	✓
	画面调整 对比度、饱和度、清晰度	✓	✓	✓	✓
	高级美型 大眼、窄脸、瘦脸（自然、女神、英俊）、V脸、下巴、短脸、脸型、发际线、亮眼、眼距、眼角、瘦鼻、鼻翼、瘦颧骨、鼻子位置、白牙、去皱、去法令纹、去眼袋、嘴型、嘴唇厚度、口红、腮红、立体	✓	✓	✓	✓
	滤镜 （默认通用滤镜）	✓	✓	✓	✓
	贴纸 （默认通用 2D 贴纸）	✓	✓	✓	✓
	高级贴纸 （默认通用 3D 贴纸）	✓	✓	✓	✓
	美妆 整妆	✓	✓	✓	✓
	可拓展功能	手势识别 （赠送 1 款指定手势贴纸）	-	✓	-
人像分割 （赠送 3 款指定分割贴纸）		-	-	✓	✓
SDK 下载	iOS	下载	下载	下载	下载
	Android	下载	下载	下载	下载

SDK 集成指引

iOS

最近更新时间：2022-03-04 11:07:02

集成准备

开发者环境要求

- 开发工具 XCode 11 及以上：App Store 或单击 [下载地址](#)。
- 建议运行环境：
 - 设备要求：iPhone 5 及以上；iPhone 6 及以下前置摄像头最多支持到 720p，不支持 1080p。
 - 系统要求：iOS 10.0 及以上。

C/C++层开发环境

XCode 默认 C++ 环境。

类型	依赖库
系统依赖库	<ul style="list-style-type: none"> • AVFoundation • Accelerate • AssetsLibrary • CoreML • JavaScriptCore • CoreFoundation • MetalPerformanceShaders • CoreTelephony • libc++.tbd
自带的库	<ul style="list-style-type: none"> • YTCommon (鉴权静态库) • XMagic (美颜静态库) • libpag (视频解码动态库)

导入资源

资源

- 必需资源包：LightCore.bundle
- 分割功能包：LightSegmentPlugin.bundle
- 手势功能包：LightHandPlugin.bundle
- 3D 功能包：Light3DPlugin.bundle

导入方法

- **方法一**：加入到工程资源中即可。
- **方法二**：如果需要指定路径 `initWithRenderSize:assetsDict:` (XMagic) 可通过这里的 `assetsDict` 配置每个资源路径。

配置权限

在 `Info.plist` 文件中添加相应权限的说明，否则程序在 iOS 10 系统上会出现崩溃。请在 `Privacy - Camera Usage Description` 中开启相机权限，允许 App 使用相机。

集成步骤

步骤一：签名准备

framework 签名可以直接在 `General-->Masonry.framework` 和 `libpag.framework` 选 `Embed & Sign`。

步骤二：鉴权

1. 申请授权，得到 `LicenseURL` 和 `LicenseKEY`，请参见 [License 指引](#)。

⚠ 注意：

正常情况下，只要app成功联网一次，就能完成鉴权流程，因此您不需要把 License 文件放到工程的工程目录里。但是如果您的app在从未联网的情况下也需要使用SDK相关功能，那么您可以把license文件下载下来放到工程目录，作为保底方案，此时license文件名必须是 `v_cube.license`。

2. 在相关业务模块的初始化代码中设置 `URL` 和 `KEY`，触发 `license` 下载，避免在使用前才临时去下载。也可以在 `AppDelegate` 的 `didFinishLaunchingWithOptions` 方法里触发下载。其中，`LicenseURL` 和 `LicenseKey` 是控制台绑定 License 时生成的授权信息。

```
[TELICENSECheck setTELICENSE:LicenseURL key:LicenseKey completion:^(NSInteger authresult, NSString * _Nonnull errorMsg) {
    if (authresult == TELICENSECheckOk) {
        NSLog(@"鉴权成功");
    } else {
        NSLog(@"鉴权失败");
    }
}];
```

鉴权 errorCode 说明：

错误码	说明
0	成功。Success
-1	输入参数无效，例如 URL 或 KEY 为空
-3	下载环节失败，请检查网络设置
-4	从本地读取的 TE 授权信息为空，可能是 IO 失败引起
-5	读取 VCUBE TEMP License文件内容为空，可能是 IO 失败引起
-6	v_cube.license 文件 JSON 字段不对。请联系腾讯云团队处理
-7	签名校验失败。请联系腾讯云团队处理
-8	解密失败。请联系腾讯云团队处理
-9	TELicense 字段里的 JSON 字段不对。请联系腾讯云团队处理
-10	从网络解析的TE授权信息为空。请联系腾讯云团队处理
-11	把 TE 授权信息写到本地文件时失败，可能是IO失败引起
-12	下载失败，解析本地 asset 也失败
-13	鉴权失败
其他	请联系腾讯云团队处理

步骤三：加载 SDK (XMagic.framework)

使用腾讯特效 SDK 生命周期大致如下：

1. 加载美颜相关资源。

```
NSDictionary *assetsDict = @{@"core_name":@"LightCore.bundle",
                              @"root_path":[[NSBundle mainBundle] bundlePath]
};
```

2. 初始化腾讯特效 SDK。

```
initWithRenderSize:assetsDict: (XMagic)
self.beautyKit = [[XMagic alloc] initWithRenderSize:previewSize assetsDict:assetsDict];
```

3. 腾讯特效 SDK 处理每帧数据并返回相应处理结果。

```
process: (XMagic)
```

```
// 在摄像头回调传入帧数据
- (void)captureOutput:(AVCaptureOutput *)captureOutput didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer fromConnection:(AVCaptureConnection *)connection;

// 获取原始数据，处理每帧的渲染信息
- (void)mycaptureOutput:(AVCaptureOutput *)captureOutput didOutputSampleBuffer:(CMSampleBufferRef)inputSampleBuffer fromConnection:(AVCaptureConnection *)connection originImageProcess:(BOOL)originImageProcess;

// 使用CPU处理数据
- (YTProcessOutput*)processDataWithCpuFuc:(CMSampleBufferRef)inputSampleBuffer;

// 使用GPU处理数据
- (YTProcessOutput*)processDataWithGpuFuc:(CMSampleBufferRef)inputSampleBuffer;

// 腾讯特效 SDK处理数据接口
/// @param input 输入处理数据信息
/// @return 输出处理后的数据信息
- (YTProcessOutput* _Nonnull)process:(YTProcessInput * _Nonnull)input;
```

4. 释放腾讯特效 SDK。

```
deinit (XMagic)
// 在需要释放SDK资源的地方调用
[self.beautyKit deinit]
```

🔗 说明：

完成上述步骤后，用户即可根据自己的实际需求控制展示时机以及其他设备相关环境。

常见问题

问题1: 编译报错: “unexpected service error: build aborted due to an internal error: unable to write manifest to-xxxx-manifest.xcbuild’: mkdir(/data, S_IRWXU | S_IRWXG | S_IRWXO): Read-only file system (30):” ?

1. 前往 **File > Project settings > Build System** 选择 **Legacy Build System**。
2. Xcode 13.0++ 需要在 **File > Workspace Settings** 勾选 **Do not show a diagnostic issue about build system deprecation**。

问题2: iOS 导入资源运行后报错: “Xcode 12.X 版本编译提示 Building for iOS Simulator, but the linked and embedded framework '.framework'...” ?

将 **Buil Settings > Build Options > Validate Workspace** 改为 **Yes**, 再单击运行。

说明:

Validate Workspace 改为 Yes 之后编译完成, 再改回 No, 也可以正常运行, 所这里有这个问题注意下即可。

问题3: 滤镜设置没反应?

检查下设置的值是否正确, 范围为 0-100, 可能值太小了效果不明显。

问题4: iOS Demo 编译, 生成 dSYM 时报错?

```
PhaseScriptExecution CMake\ PostBuild\ Rules build/XMagicDemo.build/Debug-iphoneos/XMagicDemo.build/Script-81731F743E244CF2B089C1BF.sh
cd /Users/zhenli/Downloads/xmagic_s106
/bin/sh -c /Users/zhenli/Downloads/xmagic_s106/build/XMagicDemo.build/Debug-iphoneos/XMagicDemo.build/Script-81731F743E244CF2B089C1BF.sh
```

Command `/bin/sh` failed with `exit code 1`

- 问题原因: libpag.framework和Masonry.framework 重签名失败。
- 解决方法:
 - i. 打开 demo/copy_framework.sh。
 - ii. 用下述命令查看本机 cmake 的路径, 将 \$(which cmake) 改为本地 cmake 绝对路径。

```
which cmake
```

- iii. 用自己的签名替换所有 Apple Development: 。

Android

最近更新时间：2022-03-04 11:07:10

集成准备

文件准备：

文件类型	说明
xmagic-xxx.aar	SDK，必选
../assets/	算法模型、素材资源包，必选
../jniLibs	so 库，必选

导入资源

资源

- 添加上述文件准备的全部 .aar 文件到 app 工程 libs 目录下。
- 将 SDK 包内的 assets/ 目录下的全部资源拷贝到 ../src/main/assets 目录下。
- 将 jniLibs 文件夹拷贝到工程的 ../src/main/jniLibs 目录下。

导入方法

打开 app 模块的 build.gradle 添加依赖引用：

```
android{
  ...
  defaultConfig {
    applicationId "修改成与授权lic绑定的包名"
    ....
  }
  packagingOptions {
    pickFirst '**/libc++_shared.so'
  }
}

dependencies{
  ...
}
```

```
compile fileTree(dir: 'libs', include: ['*.jar', '*.aar'])//添加 *.aar
}
```

动态下载 assets、so、动效资源指引

为了减少包大小，您可以将 SDK 所需的 assets 资源、so 库、以及动效资源 MotionRes（部分基础版 SDK 无动效资源）改为联网下载。在下载成功后，将上述文件的路径设置给 SDK。

我们建议您复用 Demo 的下载逻辑，当然，也可以使用您已有的下载服务。动态下载的详细指引，请参见 [腾讯特效 SDK 资源动态下载说明（Android）](#)。

整体流程

步骤一：鉴权

1. 申请授权，得到 License URL 和 License KEY，请参见 [License 指引](#)。

⚠ 注意：

正常情况下，只要 App 成功联网一次，就能完成鉴权流程，因此您不需要把 License 文件放到工程的 assets 目录里。但是如果您的 App 在从未联网的情况下也需要使用 SDK 相关功能，那么您可以把 License 文件下载下来放到 assets 目录，作为保底方案，此时 License 文件名必须是 v_cube.license。

2. 在相关业务模块的初始化代码中设置 URL 和 KEY，触发 License 下载，避免在使用前才临时去下载。也可以在 Application 的 onCreate 方法里触发下载，但不建议，因为此时可能没有网络权限或联网失败率较高。

```
//如果仅仅是为了触发下载或更新license，而不关心鉴权结果，则第4个参数传入null。
TELICENSECheck.getInstance().setXMagicLicense(context, URL, KEY, null);
```

3. 然后在真正要使用美颜功能前(例如 Demo 的 LaunchActivity.java)，再去做鉴权：

```
// 如果您的so库是从网络下载的，那么请在调用TELICENSECheck.getInstance().setTELICENSE之前，
// 先设置so的路径，否则鉴权会失败。
// XmagicApi.setLibPathAndLoad(validLibsDirectory);
// 如果您的so内置在apk包内，则无需调用上面的方法。
TELICENSECheck.getInstance().setTELICENSE(context, URL, KEY, new TELICENSECheckListener
() {

@Override
```

```
public void onLicenseCheckFinish(int errorCode, String msg) {
    //注意：此回调不一定在调用线程
    if (errorCode == TELicenseCheck.ERROR_OK) {
        //鉴权成功
    } else {
        //鉴权失败
    }
}
});
```

鉴权 errorCode 说明:

错误码	说明
0	成功。Success
-1	输入参数无效，例如 URL 或 KEY 为空
-3	下载环节失败，请检查网络设置
-4	从本地读取的 TE 授权信息为空，可能是 IO 失败引起
-5	读取 VCUBE TEMP License文件内容为空，可能是 IO 失败引起
-6	v_cube.license 文件 JSON 字段不对。请联系腾讯云团队处理
-7	签名校验失败。请联系腾讯云团队处理
-8	解密失败。请联系腾讯云团队处理
-9	TELicense 字段里的 JSON 字段不对。请联系腾讯云团队处理
-10	从网络解析的 TE 授权信息为空。请联系腾讯云团队处理
-11	把TE授权信息写到本地文件时失败，可能是 IO 失败引起
-12	下载失败，解析本地 asset 也失败
-13	鉴权失败，请检查 so 是否在包里，或者已正确设置 so 路径
3004/3005	无效授权。请联系腾讯云团队处理
3015	Bundle Id / Package Name 不匹配。检查您的 App 使用的 Bundle Id / Package Name 和申请的是否一致，检查是否使用了正确的授权文件
3018	授权文件已过期，需要向腾讯云申请续期

错误码	说明
其他	请联系腾讯云团队处理

步骤二：资源拷贝

- 如果您的资源文件是内置在 assets 目录的，那么使用前需要 copy 到 app 的私有目录。您可以提前 copy 好，或者在上一步鉴权成功的回调里执行拷贝操作。示例代码在 Demo 的 MainActivity.java。

```
XmagicResParser.setResPath(new File(getFilesDir(), "xmagic").getAbsolutePath());
//loading

//copy资源文件到私有目录，只需要做一次
XmagicResParser.copyRes(getApplicationContext());
```

- 如果您的资源文件是从 [网络动态下载](#) 的，下载成功后，需要设置资源文件路径。示例代码在 Demo 的 MainActivity.java。

```
XmagicResParser.setResPath(下载的资源文件本地路径);
```

步骤三：SDK 初始化及使用方法

使用腾讯特效 SDK 生命周期大致如下：

- 构造美颜 UI 数据，可参考 Demo 工程的。

XmagicResParser.java, XmagicUIProperty.java, XmagicPanelDataManager.java 代码。

- 预览布局中添加 GLSurfaceView。

```
<android.opengl.GLSurfaceView
    android:id="@+id/camera_gl_surface_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

- (可选) 快速实现相机。

将 Demo 工程中的 com.tencent.demo.camera 目录拷贝到工程中。利用 PreviewMgr 类快速实现相机功能。详细实现可参考 Demo 工程的 MainActivity.java。

```
//初始化相机
mPreviewMgr = new PreviewMgr();

//将布局的GLSurfaceView示例传入相机工具类
```

```

mPreviewMgr.onCreate(mGlsurfaceView,false);
//注册预览纹理数据回调函数
mPreviewMgr.setCustomTextureProcessor((textureId, textureWidth, textureHeight) -> {
if (mXmagicApi == null) {
return textureId;
}
//调用美颜sdk进行渲染
int outTexture = mXmagicApi.process(textureId, textureWidth, textureHeight);
return outTexture;
});

//在Activity中的onResume方法中启动相机
mPreviewMgr.onResume(this, 1280, 720);
    
```

4. 初始化美颜 SDK，建议放在 Activity 的 onResume()方法中。

```

mXmagicApi = new XmagicApi(this, XmagicResParser.getResPath(),new XmagicApi.OnXmagicPropertyErrorListener());
    
```

5. 参数

参数	含义
Context context	上下文
String resDir	资源文件目录，详见请参见 步骤二
OnXmagicPropertyErrorListener errorListener	回调函数实现类

6. 返回

错误码含义对照表：

错误码	含义
-1	未知错误
-100	3D 引擎资源初始化失败
-200	不支持 GAN 素材
-300	设备不支持此素材组件

-400	模板 JSON 内容为空
-500	SDK版本过低
-600	不支持分割
-700	不支持 OpenGL
-800	不支持脚本
5000	分割背景图片分辨率超过2160*3840
5001	分割背景图片所需内存不足
5002	分割背景视频解析失败
5003	分割背景视频超过200秒
5004	分割背景视频格式不支持

7. 添加素材提示语回调函数（方法回调有可能运行在子线程），部分素材会提示用户：点点头、伸出手掌、比心，这个回调就是用于展示类似的提示语。

```

mXmagicApi.setTipsListener(new XmagicTipsListener() {
    final XmagicToast mToast = new XmagicToast();
    @Override
    public void tipsNeedShow(String tips, String tipsIcon, int type, int duration) {
        mToast.show(MainActivity.this, tips, duration);
    }

    @Override
    public void tipsNeedHide(String tips, String tipsIcon, int type) {
        mToast.dismiss();
    }
});
    
```

8. 美颜 SDK 处理每帧数据并返回相应处理结果。

```
int outTexture = mXmagicApi.process(textureId, textureWidth, textureHeight);
```

9. 更新指定类型的美颜特效数值。

```
// 可用的入参属性可以从 XmagicResParser.parseRes() 获得  
mXmagicApi.updateProperty(XmagicProperty<?> p);
```

0. Pause美颜 SDK，建议与 Activity 的 onPause() 生命周期绑定。

```
//在 Activity 的 onPause 时调用, 需要在 OpenGL 线程调用  
mXmagicApi.onPause();
```

1. 释放美颜 SDK，建议与 Activity 的 onDestroy()生命周期绑定。

```
//注意，此方法需要在GL线程中调用  
mXmagicApi.onDestroy();
```