

TopGlasses 虚拟眼镜试戴™

Android SDK 美瞳试戴使用文档

版本号：2.0.0

发行日期：2016-11-08

前言

版权声明

TopGlasses 虚拟眼镜试戴™SDK 是成都通甲优博科技有限责任公司（以下简称通甲优博™）独立开发的具有自主知识产权的互联网软件产品。TopGlasses 虚拟眼镜试戴™SDK 文字及徽标为通甲优博™在中国和/或其他国家（地区）的注册商标或商标，受法律保护。通甲优博™拥有以上内容（包括但不限于）的专利、专利申请、软件著作权、商标、版权、商业机密等所有相关知识产权。未经通甲优博™书面许可，任何机构、组织及个人不得以任何方式或理由对该软件及商标的任何部分进行复制、修改、传播、抄录或用作其他目的。通甲优博™保留一切追究侵权行为的权利。

关于我们

通甲优博成立于 2015 年 2 月，在计算机视觉、增强现实、摄像测量领域有超过十五年的技术积累和开发经验。TopGlasses 虚拟眼镜试戴™SDK 是通甲优博研发的一款基于计算机视觉和增强现实技术的软件，它可以帮助消费者在线选购眼镜进行实时试戴，稳定性、实时性好，操作简便，贴合用户习惯。产品可完成尺寸测量、三维重建、数据采集与测量、试戴体验可视化、网购辅助与电商智能推荐，将大大改进眼镜在线试戴体验，增加在线销售。

联系我们

网站：www.topplusvision.com

技术支持：tsupport@topplusvision.com

电话：(+86) 28 8322 7667

地址：中国·成都·天府大道 2039 号天府菁蓉大厦 6F

目录

开发环境说明.....4

支持平台说明.....4

SDK 初始化5

美瞳模块使用方法.....6

 依赖库导入.....6

 获取图片等效焦距.....6

 展示美瞳效果.....6

开发环境说明

使用 android studio 1.4 及以上版本开发，相关工具的版本情况如下：

```
compileSdkVersion 23
```

```
buildToolsVersion "23.0.1"
```

```
defaultConfig {
```

```
    minSdkVersion 14
```

```
    targetSdkVersion 23
```

```
}
```

支持平台说明

目前 sdk 支持的 android 系统是 android4.0 及以上，然后支持的 android 芯片平台有 armeabi-v7a,armeabi,x86,x86_64,arm64-v8a。

SDK 初始化

调用 `topplus.com.commonutils.Library` 的 `init(Context context,String clientId,String clientSecret)` 函数，传入 `clientId` 和 `clientSecret`，进行 SDK 初始化操作。

美瞳模块使用方法

依赖库导入

依赖的库文件为: beauty-pupil-release.aar, commonutils-release.aar，添加依赖库到 Android 项目中。

获取图片等效焦距

获取相机拍摄的 jpg 图片等效焦距

```
exifInterface.readExif(mJpegPath, ExifInterface.Options.OPTION_ALL);
ExifTag focalTag = exifInterface.getTag(ExifInterface.TAG_FOCAL_LENGTH_IN_35_MM_FILE);
float focalLength = 22f;
if (focalTag != null) {
    focalTag.getId();
    focalLength = focalTag.getValueAsInt(22);
}
if (focalLength < 10 || focalLength > 40) {
    mFocusLength = 22f;
} else {
    mFocusLength = focalLength;
}
```

展示美瞳效果

方法一、

```
/**
 * 添加美瞳到指定路径的图片上面
 * @param context 上下文
 * @param image 原图片的绝对路径
 * @param bpPath 美瞳 bp 文件的路径
 * @param focusLength 原图片的焦距
 * @param scale 瞳片的放大倍数
 * @param alpha 瞳片的透明度
 * @return
 */
public static Bitmap addPupilToImage(Context context, String image, String bpPath, float
focusLength, float scale, float alpha) ;
```

方法二、

```
/**
 * 添加美瞳到指定路径的图片上面
 * @param context 上下文
 * @param image 原图片的绝对路径
 * @param bpPath 美瞳 bp 文件的路径
 * @param focusLength 原图片的焦距
 * @param scale 瞳片的放大倍数
 * @param alpha 瞳片的透明度
 * @param isNeedHighLight 是否需要添加高光
 * @param enableHDMI 是否需要获得高清大图
 * @return
 */
public static Bitmap addPupilToImage(Context context, String image, String bpPath, float
focusLength, float scale, float alpha, boolean isNeedHighLight, boolean enableHDMI)
```

方法三、

```
/**
 * 添加指定瞳片到指定图片上面，并根据显示宽度裁剪图片
 * @param context 上下文
 * @param image 原图片的绝对路径
 * @param bpPath 美瞳 bp 文件的路径
 * @param ratio 显示宽度是脸宽的 ratio 倍
 * @param r 显示高度是显示宽度的 r 倍
 * @param isNeedHighLight 是否需要添加高光
 * @param focusLength 原图片的焦距
 * @param scale 瞳片的放大倍数
 * @param alpha 瞳片的透明度
 * @return
 */
public static Bitmap getPupillImage(Context context, String image,
                                     String bpPath,
                                     float ratio,
                                     float r,
                                     boolean isNeedHighLight,
                                     float focusLength,
                                     float scale,
                                     float alpha)
```

获取显示的美瞳片 Icon(有两种方法，建议用异步方法):

1.同步方法（直接调用获取到 icon,但有可能获取失败):

// 获取 bp 文件中的美瞳 icon

```
Bitmap returnBtp = StaticPupilTexture.getOutIcon(getApplicationContext(),bpPath);
```

2.异步方法：初始化资源文件，调用 init 之前必须保证 bp 文件已经 copy 到本地，否则后续获取 icon 失败

```
StaticPupilTexture.init(getApplicationContext(), decodePathList);
```

// 获取当前 bpPath 资源的美瞳 icon

```
Bitmap returnBtp = StaticPupilTexture.getOutIconSync(getApplicationContext(),bpPath);
```

3.同步方法：将 bp 文件解压，并将里面瞳片的 png 图片保存到指定文件 desPath

```
StaticPupilTexture .getOutImage(String bpPath, String desPath);
```