

# TopGlasses 虚拟眼镜试戴™

## iOS SDK 人脸录制试戴使用文档

版本号：1.0.22

发行日期：2016-11-29

# 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](http://www.topplusvision.com)

技术支持：[tsupport@topplusvision.com](mailto: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

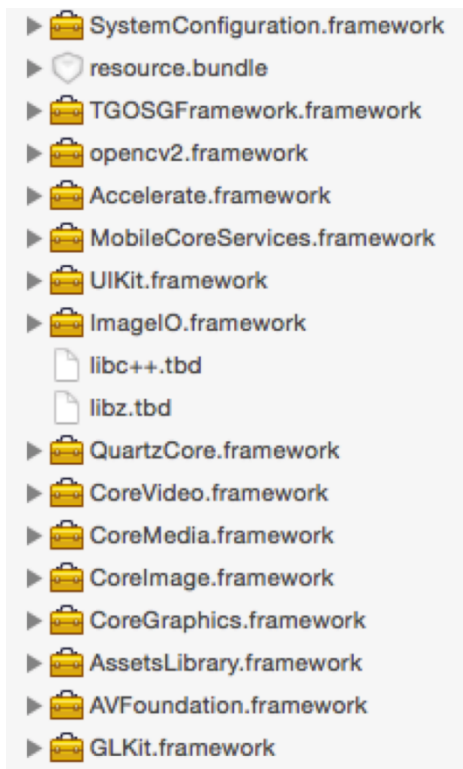
## 目录

人脸录制试戴模块使用方法 .....	4
依赖库导入 .....	4
初始化环境 .....	5
授权认证 .....	6
设置前置摄像头分辨率 .....	6
生成人脸序列图 .....	6
展示试戴效果 .....	7
切换人脸序列图片 .....	7
切换眼镜模型 .....	7
调整眼镜姿态 .....	7
调整人脸序列图效果 .....	8
其他接口 .....	8

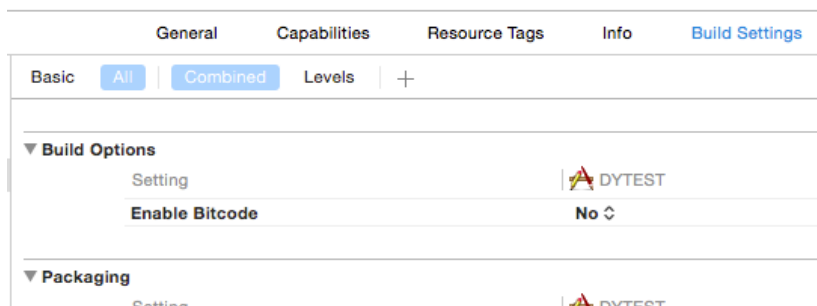
# 人脸录制试戴模块使用方法

## 依赖库导入

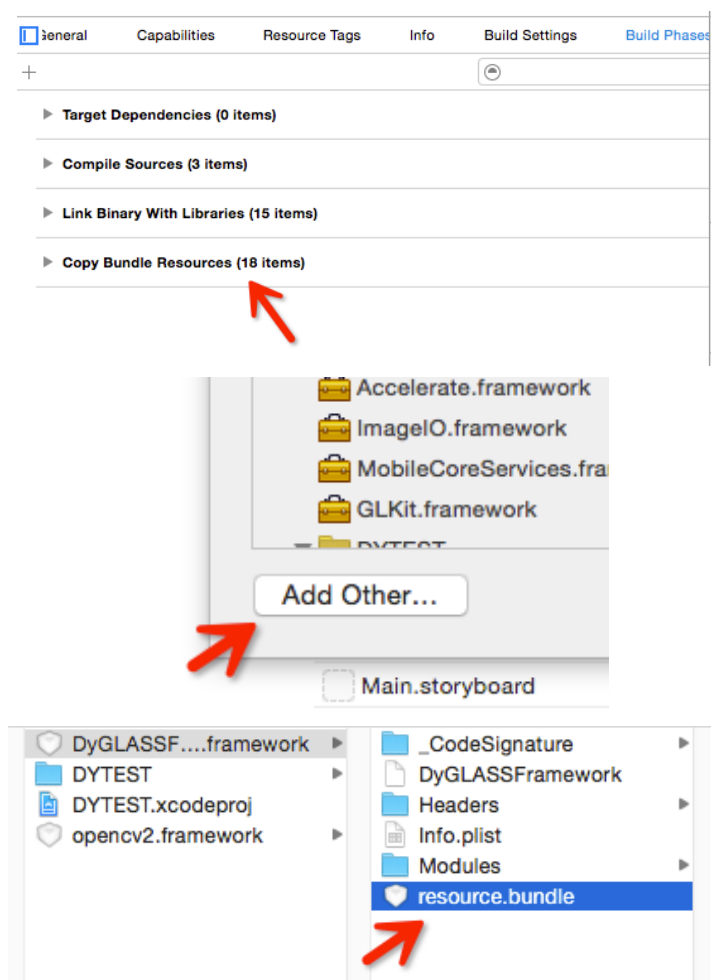
静态试戴模块封装为 TGOSGFramework.framework。所依赖的库如下图，必须都导入工程中。依赖的库中 opencv2.framework，位于 frameworks 文件夹中。模型文件位于 glassModels 文件中。



需要将 bitcode 设置为 NO。



添加 resource.bundle。



## 初始化环境

新建 SDKHandle 对象并对该对象做初始化操作。

SDK 当前支持的试戴视频分辨率是 720p，为目前 iPhone/iPad 前置摄像头的默认最高分辨率。传入 handle 的尺寸比例应为 1:1。

```
width = self.view.bounds.size.width;
```

```
height = width;
```

1. 初始化 SDKHandle

```
handle = [[SDKHandle alloc] initWithFrame:CGRectMake(x, y, width, height)];
```

```
handle.contentMode = UIViewContentModeScaleAspectFit;
```

2. 认证

//传入 client\_id, client\_secret 进行授权认证

```
[handle setLicense:client_id andSecret:client_secret];
```

3. 初始化

```
[handle Engine_Init];
```

```
[handle SetFocuswithFocus: 31];
```

```
handle.multipleTouchEnabled = YES;
```

```
[self.view addSubview:handle];
```

```
[self.view sendSubviewToBack:handle];
```

## 授权认证

```
//设置 ID 和 secret
```

```
- (void)setLicense:(NSString *)Client_id andSecret:(NSString *)Clisent_secret;
```

```
[handle setLicense:client_id andSecret:client_secret];
```

说明：申请 client\_id 和 client\_secret 后调用此函数获得授权。

不调用认证函数会开启试用版，试用版在显示图片上会有水印，正常调用认证函数即可正常显示。

## 设置前置摄像头分辨率

```
imagePicker.videoQuality = UIImagePickerControllerQualityTypeIFrame1280x720;
```

## 生成人脸序列图

在进行静态试戴时，需要把用户录制的试戴视频，生成一组人脸序列图片和相关的其他文件。生成序列图的具体步骤如下：

1. 需要录制一段，用户头部向左转到 30 度左右，然后再转回来，继续向右转 30 度，最后转回中间的视频（缓慢转动）；

```
[handle useVideoWithVideoPath:videoPath andPicturePath:picPath];
```

2. 添加如下监听中心，接收视频处理完成返回的消息：

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(test:)
```

```
name:@"MessageFromUnityFrame" object:nil];
```

添加如下监听中心，接收视频处理过程中未检测到人脸的返回消息：

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(testForDetectedFace:)
```

```
name:@"NodetectedFace" object:nil];
```

添加如下监听中心，接收视频处理过程中返回的正脸图片的数据：

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(testForMiddlePic:)
```

```
name:@"MiddlePictureComplete" object:nil];
```

3. 消息处理函数如下：

```
- (void)test:(NSNotification *)aNotification {
    if(handle)
    {
        dispatch_async(dispatch_get_main_queue(), ^{
            [SVProgressHUD dismiss];
            [handle changePicturesIfNeed];
            [handle loadPictureWithOrder:YES andPicturePath:picPath];
            dispatch_async(dispatch_get_main_queue(), ^{
                label.text = [handle GetFaceShape];
            });
        });
    }
}
```

```

        });
    });
}
}
- (void)testForDetectedFace:(NSNotification *)aNotification{
    dispatch_async(dispatch_get_main_queue(), ^{
        [SVProgressHUD dismiss];
    });
}
- (void)testForMiddlePic:(NSNotification *)aNotification{
    NSData *data = (NSData *)aNotification.object;
    dispatch_async(dispatch_get_main_queue(), ^{
        if (!middleImageView) {
            middleImageView = [[UIImageView alloc] initWithFrame:CGRectMake(0, 70, WIDTH,
WIDTH)];
            [self.view addSubview:middleImageView];
        }
        middleImageView.image = [UIImage imageWithData:data];
    });
}
}

```

## 展示试戴效果

接收到消息中心的视频处理完成消息后，使用下列 SDK 接口展示试戴效果：

## 切换人脸序列图片

//根据序列图路径加载单张图片，第一个属性:order 值为 YES 时升序，值为 NO 时降序 第二个属性 PicturePath: 提取序列图的路径

```
- (void) loadPictureWithOrder:(BOOL)order andPicturePath:(NSString *)picturePath;
```

## 切换眼镜模型

//从本地加载眼镜模型到指定窗口。filePath: 本地眼镜模型的路径

```
-(void)loadGlassesModel: (NSString*)filePath;
```

## 调整眼镜姿态

//设置鼻托在鼻梁上的位置，参数范围是 0-1,设置为 0 鼻托在原位置，设置为 1 鼻托在鼻尖上

```
- (void)SetNosePadPos:(float) glassHeight;
```

//设置眼镜在垂直角度绕鼻托旋转的角度，参数范围是 0-1,设置为 0 为向下转 5 度，设置为 1 是向

上转 5 度

- (void) SetGlassVerticalAngle:(float) verticalAngle;

//设置镜腿虚化的位置，参数范围是 0-1,设置为 0 是虚化位置在镜腿根部，设置为 1 是在尾部

- (void) SetFeatherDistance:(float) featherDistance;

//新增调整眼镜大小接口:参数 glassModelScale: 取值范围: 0~1, 大于 0.5 眼镜增大, 小于 0.5 眼镜减小:

[handle setGlassModelScale:glassModelScale];

注意：在使用过程中，需要将以下值初始化设置为：

glassHeight = 0.2;

verticalAngle = 0.5;

featherDistance = 0.9;

glassModelScale = 0.5;

//用户手动调整眼镜姿态完成后调用，保存对应的姿态数据

- (void)saveAdjustedAttitude;

## 调整人脸序列图效果

/\*\*获得美颜效果

type:

1 美白

2 磨皮

3 对比度

value:

0 ~ 1

\*/

- (void)enhancePics:(int)type andValue:(float)value;

## 其他接口

//隐藏正在显示的眼镜模型

- (void)hideGlassModel:(BOOL)isVisible;

//获取脸型分类 -1: 没有人脸存在 0:圆脸 1:椭圆脸 2:方脸 3:梨形脸 4:瓜子脸

- (int)getFaceShape;

//返回正中图片在序列图路径下的文件名称

- (NSString \*)middlePath;



```
//是否镜像序列图
- (void)setImageMirroredStatus:(BOOL)Mirrored;

//保存日志,
- (void)saveLogToFile:(NSString *)filePath;
//设置所有眼镜为统一大小
- (void)SetNormalizedScale:(BOOL)isNormal;
//返回当前的渲染图片
- (UIImage *)snapshotImage;
参数
//设置显示序列图的窗口区域
@property (nonatomic, assign, readwrite)CGRect screenFrame;
```