

Live Sticker & Filter SDK Developer Guide(iOS)



Mobile & AI Leader

copyright(c) 2018 All rights reserved. JunSoft., Co.Ltd.

1. iOS Development Overview

1) Project Configuration Overview

- SDK file: Refer to sample program (JFace) for project import related matters.

libjFaceLib.a, jFaceLib.h, Scene.h, Scene.mm, model, bf2, live_sticker.json

- Other files: Files for UI implementation and other processing in the sample program

Downloader, m, Downloader.h, ItemDelegate.m, ItemDelegate.h,
ThumbCell.h,
ThumbCell.m

- Precautions:

The internal SDK core content is written in C ++, so the source that imports jFaceLib.h should be implemented as an .mm file (See ViewController.mm)

If the main project you want to port is in "swift", you need to create a wrapper class that calls the SDK in the bridge file format.

2. Initializing the SDK

1) Scene Interface

- GPUImageFilter Wrapper Class for the video recording function of GPUImage
- Includes a core library (jFaceLib)

```
@interface Scene : GPUImageFilter

@property (nonatomic, retain) jFaceLib *lib;
...

@end
```

2) Initialization of the SDK

- Initialization based on the camera's frame rate and field of view value
- See the viewDidLoad section of ViewController.m for details.

Ex)

```
scene = [[Scene alloc] init];

[scene initWithSDKFrameRate:frameRate fov:fov];
```

- **(BOOL)**initWithSDKFrameRate:(float)framerate fov:(float)fov

(Arguments)

framerate : Camera frame rate

fov : Field of view value of the
camera

(Return)

"YES" when SDK initialization successful

3. Frame Buffer Processing

- To use Live Sticker and Filter, set willOutputSampleBuffer to
Must override willOutputSampleBuffer

-In order for willOutputSampleBuffer to be called, the camera's delegate must be self corresponding to the class.

ex)

-

(Camera initialization)

```
frontCamera = [[GPUImageVideoCamera alloc]
initWithSessionPreset:AVCaptureSessionPreset1280x720
cameraPosition:AVCaptureDevicePositionFront];
    backCamera = [[GPUImageVideoCamera alloc]
initWithSessionPreset:AVCaptureSessionPreset1280x720
cameraPosition:AVCaptureDevicePositionBack];
```

```
camera = frontCamera;
```

```
camera.horizontallyMirrorFrontFacingCamera = (camera == frontCamera) ? YES : NO;
```

```
camera.outputImageOrientation = UIInterfaceOrientationPortrait;
```

```
camera.delegate = (id)self;
```

(Frame buffer processing)

```
- (void)willOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer {
```

```
    UIDeviceOrientation deviceOrientation = [[UIDevice currentDevice] orientation];
```

```
    CFAbsoluteTime startTime = CFAbsoluteTimeGetCurrent();
```

```
    [scene processBuffer:sampleBuffer orientation:(int)deviceOrientation front:camera ==
frontCamera];
```

```
    ..
```

```
    }
```

-The processBuffer function is a function that takes a video frame buffer from the core SDK and processes the Live Sticker and filter.

- **(void)processBuffer:(CMSampleBufferRef)sampleBuffer orientation:(int)deviceOrientation front:(BOOL)bFont**

(Arguments)

sampleBuffer : Sample buffer reference

device Orientation : The orientation value of the device

4. Live Sticker, Filter Initialization

1) Initialization of Live Sticker / Filter

- To initialize the Live Sticker, set the name of the scene to "LiveSticker"
- The name of the Live Sticker corresponds to the name live_sticker.json.
- Live_sticker.json specification details are covered in the Live Sticker Creation Guide.
- To initialize the filter, set the name of the scene to "Filter"

Ex)

1)

```
filter = [scene initWithScene:@"LiveSticker" filterName:name mirrored:(camera == frontCamera) ?  
1 : 0 resType:resType];
```

**-(id)initWithScene:(NSString*)sceneName filterName:(NSString*)name
mirrored:(BOOL)mirrored resType:(int)resType**

(Arguments)

sceneName : @"LiveSticker" for the Live Sticker, @"Filter" for the filter
filterName: Corresponds to the name of json for the Live Sticker, and to the lookup file name for the filter
mirrored : Whether to output a mirror image for the front camera
resType : One can set these values depending on whether you want to initialize local resources or network resources.

```
typedef enum {  
    eNetworkResource = 0,  
    eLocalResource = 1  
}eResType;
```

(Return)

GPUImageFilter 인스턴스 리턴

2) Using Live Sticker / Filter

- The GPUImageFilter is an inherited class, and the remaining class items are similar to the usage method of GPUImage. Please refer to the selectScene function of ViewController.m in the sample project for details.

Ex)

```
filter = [scene initWithScene:sceneName filterName:name mirrored:(camera == frontCamera) ?
1 : 0 resType:resType];
[filter addTarget:primaryView];
[camera addTarget:filter];
```

3) Concerning Live Sticker / Filter video recording

- Recording is started with a long touch of the sample project, and recording is ended when this touch is released.

- For video recording, this SDK is not relevant. For reference, use GPUImageMovieWriter and refer to the sample project's ViewController.m source