

视频外部采集

1 常用场景

当开发者业务中出现以下情况时，我们推荐使用 SDK 的外部采集功能：

1. 普通摄像头的采集无法满足需求。例如，包含了大量的原有业务。
2. 直播过程中，开发者需要使用摄像头完成的额外功能和 SDK 的默认逻辑有冲突，导致摄像头无法正常使用。例如，直播到一半，需要录制短视频。
3. 直播非摄像头数据。例如需要将视频播放的数据推流出去。
4. 开发者可以使用外部采集来做自定义美颜等视频预处理。

2 下载示例源码

互动视频 Demo 源码下载地址：互动视频示例 Demo_iOS

[码云下载地址\(国内推荐\)](#)
[阿里云下载地址\(国内推荐\)](#)
[github下载地址](#)

示例 Demo 包含了基础专题和进阶专题，展示了如何使用 SDK API，开发者可参考其用法来实现自己的业务。示例 Demo 涵盖了 SDK 的大部分功能，开发者也可在示例 Demo 中查找 ZEGO SDK 更多的进阶功能，测试其功能和性能，以实现特定的需求。

视频外部渲染模块源码请查看 [/src/LiveRoomPlayground-iOS/ExternalVideoCaptureUI](#) 目录下的源文件。

3 使用步骤

特别说明

考虑到设备的独占问题，SDK 的视频外部采集采用的是面向对象的设计，帮助用户把原有采集代码封装成外部采集设备。

开发者通过实现 `ZegoVideoCaptureFactory` 和 `ZegoVideoCaptureDevice` 协议，可以把外部采集的数据传给 SDK 进行编码推流：

1. `ZegoVideoCaptureFactory` 是外部采集的入口，定义了创建、销毁 `ZegoVideoCaptureDevice` 的接口，向 SDK 提供管理 `ZegoVideoCaptureDevice` 生命周期的能力。需要调用 `setVideoCaptureFactory` 的地方必须实现该接口。
2. `ZegoVideoCaptureDevice` 定义基本的组件能力，包括 `allocateAndStart`、`stopAndDeAllocate`、`startCapture`、`stopCapture`、`zego_supportBufferType`，方便 SDK 在直播流程中进行交互。

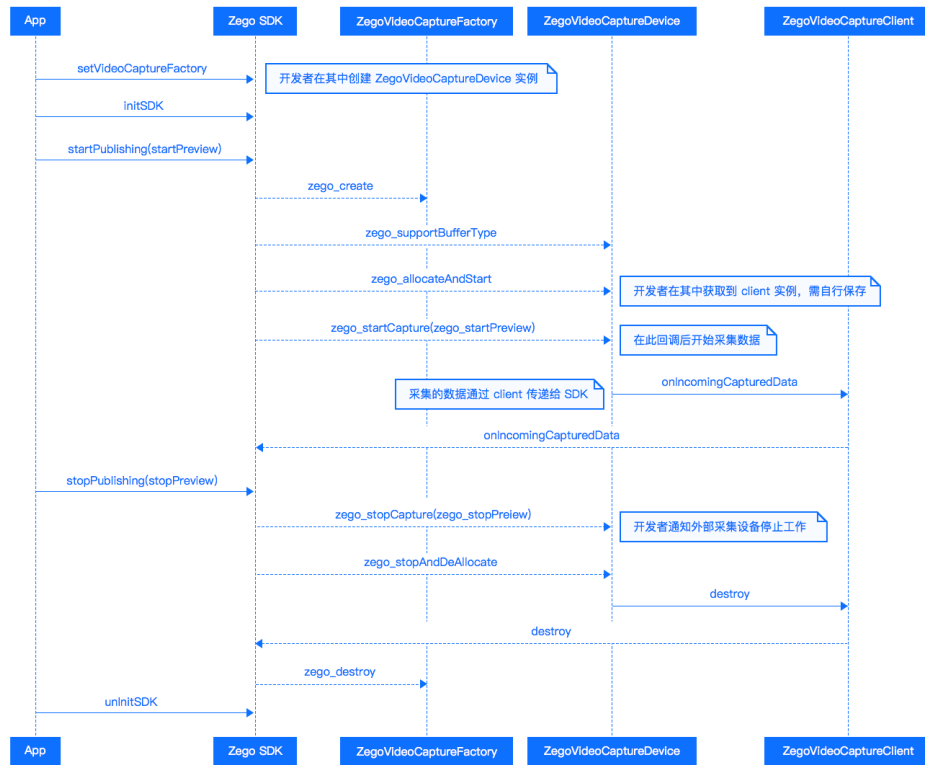
请注意，SDK 会在适当的时机创建和销毁 `ZegoVideoCaptureDevice`，开发者无需担心生命周期不一致的问题。

使用视频外部采集的简要使用步骤：

- 1. 实现外部采集设备
- 2. 实现外部采集工厂
- 3. 设置外部采集工厂
- 4. 初始化 SDK
- 5. 推流

外部采集的接口调用流程如下图所示：

视频外部采集流程图



Demo 中展示了如何通过外部采集工厂和外部采集设备来使用外部采集功能并进行推流、拉流的操作，demo里包含了采用摄像头、图片作为外部采集源的示例。

3.1 外部采集工厂创建外部采集设备

在 SDK 回调 `zego_create` 时，创建并返回 `ZegoVideoCaptureDevice` 实例。在 `zego_destroy` 回调销毁 `ZegoVideoCaptureDevice` 实例。下述代码以 Demo 为例说明，`ZGDemoExternalVideoCaptureFactory.m` 自身实现了 `ZegoVideoCaptureDevice`，所以 `zego_create` 方法返回 `self`。

```

ZGDemoExternalVideoCaptureFactory.m

#pragma mark - ZegoVideoCaptureFactory

- (nonnull id<ZegoVideoCaptureDevice>)zego_create:(nonnull NSString*)deviceId {
    return self;
}
    
```

```

- (void)zego_destroy:(nonnull id<ZegoVideoCaptureDevice>)device {
    [self destroyResource];
}

@end

```

请注意：

1. 大部分情况下，`ZegoVideoCaptureFactory` 会缓存 `ZegoVideoCaptureDevice` 实例，开发者需避免创建新的实例，造成争抢独占设备（例如摄像头）。
2. 开发者必须保证 `ZegoVideoCaptureDevice` 在采集工厂 `zego_create` 和 `zego_destroy` 之间是可用的，请勿直接销毁对象。

3.2 外部采集设备实现

实现 `ZegoVideoCaptureDevice` 协议，协议的方法有：

```

zego-api-external-video-capture-oc.h

/** 视频外部采集设备接口 */
@protocol ZegoVideoCaptureDevice <NSObject, ZegoSupportsVideoCapture>

@required

/**
    初始化采集使用的资源（例如启动线程等）回调

    @param client SDK 实现回调的对象，一定要保存
    @discussion 第一次调用开始预览/推流/拉流时调用
    */
- (void)zego_allocateAndStart:(nonnull id<ZegoVideoCaptureClientDelegate>) client;

/**
    停止并且释放采集占用的资源

    @discussion 在此之后，不能再调用 client 对象的接口
    */
- (void)zego_stopAndDeAllocate;

/**
    启动采集，采集的数据通过 [client -onIncomingCapturedData:withPresentationTimeStamp:]
    通知 SDK

    @return 0 表示成功，其他是错误
    @discussion 一定要实现，不要做丢帧逻辑，SDK内部已经包含了丢帧策略

```

```

    */
    - (int)zego_startCapture;

    /**
     停止采集

     @return 0 表示成功，其它是错误
     @discussion 一定要实现
     */
    - (int)zego_stopCapture;

    @Optional
    /**
     支持的 buffer 类型

     @return 支持的 buffer 类型
     @discussion 如果不实现，则为 ZegoVideoCaptureDeviceOutputBufferTypeCVPixelBuffer
     */
    - (ZegoVideoCaptureDeviceOutputBufferType)zego_supportBufferType;

```

3.2.1 告知 SDK 当前采集数据的类型

由于 iOS 采集的多样性，SDK 支持多种外部采集数据格式，所以开发者需要告知 SDK 当前采集设备使用何种数据类型。目前 SDK 支持的类型有：

类型定义	类型说明
<code>ZegoVideoCaptureDeviceOutputBufferTypeCVPixelBuffer</code>	<code>CVImageBufferRef</code>
<code>ZegoVideoCaptureDeviceOutputBufferTypeGLTexture2D</code>	<code>GLuint</code>
<code>ZegoVideoCaptureDeviceOutputBufferTypeEncodedFrame</code>	<code>EncodedFrame</code> ，编码后的采集数据

SDK 默认为 `ZegoVideoCaptureDeviceOutputBufferTypeCVPixelBuffer`。如果需要使用其他数据类型，需要实现 `ZegoVideoCaptureDevice` 的相应方法，然后在获取视频帧数据后调用 `client` 对应数据类型的接收方法。

```

    /**
     支持的 buffer 类型
     如果不实现，则为 ZegoVideoCaptureDeviceOutputBufferTypeCVPixelBuffer
     */
    - (ZegoVideoCaptureDeviceOutputBufferType)zego_supportBufferType {

```

```

    return ZegoVideoCaptureDeviceOutputBufferTypeCVPixelBuffer;
}

```

Demo 中使用默认的 `ZegoVideoCaptureDeviceOutputBufferTypeCVPixelBuffer` 类型。

3.2.2 初始化资源

外部采集初始化会回调 `zego_allocateAndStart:`，初始化资源需要在此方法中实现。

开发者在 `zego_allocateAndStart:` 中获取到 `client` 对象，`client` 用于向 SDK 传入外部采集的数据。SDK 会在第一次预览 / 推流 / 拉流时回调 `zego_allocateAndStart:`。

请注意，`client` 实例在 `zego_stopAndDeAllocate` 被调用前必须一直保存，否则后续步骤中无法向 SDK 传入外部采集视频。

Demo 示例代码如下：

```

- (void)zego_allocateAndStart:(nonnull id<ZegoVideoCaptureClientDelegate>)client {
    dispatch_async(_clientQueue, ^{
        self.client = client;
        [self.client setFillMode:ZegoVideoFillModeCrop];
    });
}

```

为保证 `client` 对象的线程安全，此处使用了 `_clientQueue` 串行队列来同步操作 `client` 对象。

3.2.3 向 SDK 传入外部采集视频数据

外部采集设备启动后，开发者就可以使用 `client` 的 `-onIncomingCapturedData:withPresentationTimeStamp:` 方法向 SDK 传入外部采集视频数据。

Demo 中在自定义的 `ZGDemoExternalVideoCaptureFactory` 类中封装了 `-postCapturedData:withPresentationTimeStamp:` 方法。

```

- (void)postCapturedData:(CVImageBufferRef)image withPresentationTimeStamp:(CMTime)time {
    if (!image) return;
    CVBufferRetain(image);
    dispatch_async(_clientQueue, ^{

```

```

        if (self.isCapture) {
            [self.client onIncomingCapturedData:image withPresentationTimeStamp:time];
        }
        CVBufferRelease(image);
    });
}

```

为保证 `client` 对象的线程安全，此处也使用了 `_clientQueue` 串行队列来同步操作 `client` 对象。

3.2.4 释放资源

当不再使用外部视频采集或 Uninit SDK 时，会回调 `zego_stopAndDeAllocate`，开发者可以在该回调中释放占用的资源。

Demo 示例代码如下：

```

ZGDEMOExternalVideoCaptureFactory.m

- (void)zego_stopAndDeAllocate {
    [self destoryResource];
}

- (void)destoryResource {
    [self stopCapture];

    // 同步_clientQueue, 在 client destroy 前停止 client 向SDK继续塞数据
    dispatch_sync(_clientQueue, ^{

    });
    [self.client destroy];
    self.client = nil;
}

```

1. 为保证 `client` 对象的线程安全，此处也使用了 `_clientQueue` 串行队列来同步操作 `client` 对象。并在停止采集任务后再清理 `client` 对象，保证 SDK 调用 `zego_stopAndDeAllocate` 后，没有残留的异步任务，否则会导致野指针 crash。2. 请注意，销毁 `client` 需调用 `destroy` 方法，否则会造成内存泄漏。

3.5 设置外部采集工厂

开发者需要使用外部采集功能时，请在初始化 SDK 前设置外部采集工厂对象。

```
zego-api-external-video-capture-oc.h
```

```
@interface ZegoExternalVideoCapture : NSObject
```

```
+ (void)setVideoCaptureFactory:(nullable id<ZegoVideoCaptureFactory>)factory channelId:  
(ZegoAPIPublishChannelIndex)idx;
```

```
@end
```

请注意 1. 2020-05-12 及之后版本，`setVideoCaptureFactory` 必须在引擎启动之前调用，2020-05-12 之前版本必须在 `InitSDK` 前调用，需要使用外部采集功能时，该工厂对象不能为空。 2. 不再使用外部采集时，请调用本接口 `setVideoCaptureFactory` 设置为 `nil`，否则下次引擎启动后（2020-05-12 及之后版本）或者 `Init SDK` 后（2020-05-12 之前版本）仍然还会使用视频外部采集

引擎启动的时机有：1、登录房间；2、未登录房间启动预览功能；3、未登录房间使用媒体播放器；4、未登录房间使用音效播放器。引擎关闭的时机有：1、退出房间；2、未登录房间并且预览、媒体播放器和引擎播放器都停止工作。

4 外部采集具体使用示例

4.1 相机视频源

示例展示了用摄像头作为外部采集数据源。[查看示例代码](#)

4.2 图片视频源

示例展示了读取图片文件数据来作为外部采集数据源。[查看示例代码](#)

5 Q&A

- Q1：如何访问 `CVPixelBufferRef` 持有的图像内存？
- A1：请参考 `ZegoLiveRoomApi` 的 `copyPixelBufferFrom:to:` 方法，然后对照苹果官方头文件。
- Q2：如何使用 `ZegoVideoCaptureDeviceOutputBufferTypeGLTexture2D` 方式传递采集数据？

- A2: 选择 `ZegoVideoCaptureDeviceOutputBufferTypeGLTexture2D` 方式时, 开发者实现 `zego_supportBufferType` 返回 `ZegoVideoCaptureDeviceOutputBufferTypeGLTexture2D`, 然后调用 `client` 的 `onIncomingCapturedData:width:height:withPresentationTimeStamp:` 传递数据。
- Q3: 如何使用 `ZegoVideoCaptureDeviceOutputBufferTypeEncodedFrame` 方式传递编码数据?
- A3: 选择 `ZegoVideoCaptureDeviceOutputBufferTypeEncodedFrame` 方式时, 开发者实现 `zego_supportBufferType` 返回 `ZegoVideoCaptureDeviceOutputBufferTypeEncodedFrame`, 可以调用 `client` 的 `onEncodedFrame:config:height:bKeyframe:withPresentationTimeStamp` 传递数据。目前只支持H.264码流, 关键帧间隔推荐2秒。

请注意, 每个IDR帧都必须携带SPS和PPS, 放在最前面。

- Q4: `ZegoSupportsVideoCapture` 的方法需要实现么?
- A4: 对于 `ZegoSupportsVideoCapture` 所包含的方法, 不要求开发者实现, 仅仅只是为了方便开发者在自己的采集实现和 SDK 默认的采集实现间切换, 避免业务层写两套逻辑代码。

每个方法都有对应的 SDK 接口, 调用 SDK 接口时, 会透传调用对应的方法。比如调用 SDK 的 `startPreview` 时, 会透传调用 `ZegoVideoCaptureDevice` 的 `startPreview` 方法。

- Q5: `ZegoVideoCaptureFactory` 的子类什么时候释放?
- A5: 我们推荐把工厂的实例保存为单例, 仅作为 SDK 管理外部采集设备生命周期的通道, 开发者可以为工厂子类添加 setter 和 getter, 一起管理采集类的生命周期。
- Q6: 使用外部采集, 本地预览的画面正常, 推出去观众端看到的画面变形了?
- A6: 外部采集进来的图像比例和 SDK 默认设置的分辨率的比例不一致 (比如外部采集进来的是 4:3, SDK 默认推流分辨率 360*640 是 16:9)。解决方案:
 1. 开发者将外部采集的视频分辨率比例修改为 16:9
 2. 开发者调用 `setAvConfig`, 将 SDK 的推流分辨率自定义为 4:3 (比如480*640)
- Q7: 开启外部采集后, `setPreviewView`、`startPreview` 无效?

- A7: [SDK开启外部采集, 外部采集的数据, 目前需要用户自己渲染]。
- Q8: 开启外部采集后, 外部采集的帧率和拉流播放帧率不一致?
- A8: 通过设置SDK的帧率 `setVideoFPS` 和外部采集调用 `OnIncomingCapturedData` 的频率一致。
- Q9: SDK 接收视频帧数据方法内部对传入的数据是同步处理还是异步处理?
- A9: SDK 接收视频帧数据后, 会先同步拷贝数据, 然后再异步执行编码等操作, 所以在将数据传入 SDK 后即可立即释放。