

在手机视频的场景中，因为手机可以竖着、横着，四个方向随意旋转，为了保证各种设备互通时观看正常，我们提供了一套角度方案。

一、采集角度

角度定义

不同平台、不同系统、不同设备定义采集方向的方法不一致，这里统一为采集角度。iOS和Android的后置摄像头角度一致，所以这里以手机后置摄像头为基准；考虑到和PC互通，PC摄像头采集的图像都是横向的，所以把PC摄像头采集的横向角度定义为0。具体定义如下：

手机采集	视频数据	角度值
		0
		1
		2
		3

其中，手机采集的画面为在手机上preview看到的画面，视频数据为传给SDK的数据。因为无论手机怎么旋转，从摄像头得到的数据其实都是横向的数据，只是其中的内容会由于旋转变得不同，所以视频数据中的图都是横向的。

前置摄像头

iOS和Android在前置摄像头上的定义不一致，具体定义如下：

iOS前置采集	Android前置采集	视频数据	角度值
			0
			1
			2
			3

锁定旋转

iOS和Android手机都有锁定自动旋转的功能。

目前的实现是，当锁定旋转时，Android的角度表现不会发生变化，还是有旋转的效果；iOS的角度会固定为1，无论是竖屏锁定还是横屏锁定，角度都会重置为竖屏状态的1。

锁定的实现，后续会做下优化，争取iOS和Android的表现能统一，并且支持横屏的锁定。

API设置

iOS Capture

iOS的采集角度不用设置，SDK内部实现。

Android Capture

Android需要通过重力感应事件把当前的手机旋转角度传给SDK。

```
AVVideoCtrl avVideoCtrl = AVContextModel.getInstance().getAVContext().getVideoCtrl();
avVideoCtrl.setRotation(rotation);
```

二、视频绘制

视频数据在编码传输、接收解码的整个过程中，不会做旋转处理，一直保持为横向数据。采集端的角度信息，一直透传到接受端的渲染模块。渲染模块在绘制远端视频时，会根据采集角度和当前设备的旋转角度，先对视频画面进行旋转，再绘制出来，保证观看到的是角度正常的视频画面。

视频数据	正常观看	左旋观看	右旋观看	倒置观看
<div></div> <div></div>				
<div></div> <div></div>				

锁定旋转

iOS和Android手机都有锁定自动旋转的功能。当锁定旋转时，也会影响视频绘制的角度。

目前的实现是，当锁定旋转时，Android的视频绘制表现不会发生变化，还是有旋转的效果；iOS的绘制表现会固定为正常观看，无论是竖屏锁定还是横屏锁定，都会重置为正常观看的竖屏观看状态。

锁定的实现，后续会做下优化，争取iOS和Android的表现能统一，并且支持锁定下的横屏观看。

API设置

iOS Render

iOS的渲染需要业务层做好角度的处理，示例代码如下：

```
#pragma mark remoteVideoDelegate
-(void)OnVideoPreview:(QAVVideoFrame*)frameData{
    int peerRotate = frameData.frameDesc.rotate;
    int selfRotate = 0;
    UIInterfaceOrientation currentOri=(UIInterfaceOrientation)[[UIDevice currentDevice] orientation];
    switch (currentOri) {
        case UIDeviceOrientationPortrait:
            selfRotate = 0;
            break;
        case UIDeviceOrientationLandscapeLeft:
            selfRotate = 1;
            break;
        case UIDeviceOrientationLandscapeRight:
            selfRotate = 3;
            break;
        case UIDeviceOrientationPortraitUpsideDown:
            selfRotate = 2;
            break;
        default:
            break;
    }
    frameData.frameDesc.rotate = (selfRotate + peerRotate ) % 4;
    ... ..
}
```

Android Render

Android需要通过重力感应事件把当前的手机旋转角度传给SDK，SDK内部处理采集角度的调整逻辑。

```
AVVideoCtrl avVideoCtrl = AVContextModel.getInstance().getAVContext().getVideoCtrl
();
avVideoCtrl.setRotation(rotation);
```

三、旁路直播和录制功能

旁路直播和录制功能都是在后台实现，而SDK客户端上传给后台server的数据都是横向视频数据，其中的内容可能会发生颠倒。如果后台server不处理，直接直播或录制的话，用户看到的视频画面可能是颠倒的。所以后台server在旁路直播和录制之前，也会根据SDK客户端上传的角度，对视频数据进行解码、旋转、再编码的处理。

当观看旁路直播或录制视频发现角度不对时，首先要确认下，在采集端设置的采集角度是否正确。

四、视频转置（待完善）