

OpenGL ES实践教程（二）摄像头采集数据和渲染



作者 落影loyinglin (/u/815d10a4bdce) + 关注

2016.07.31 00:50 字数 1531 阅读 3104 评论 29 喜欢 23 阅读 3104 评论 29 喜欢 23 (/u/815d10a4bdce)

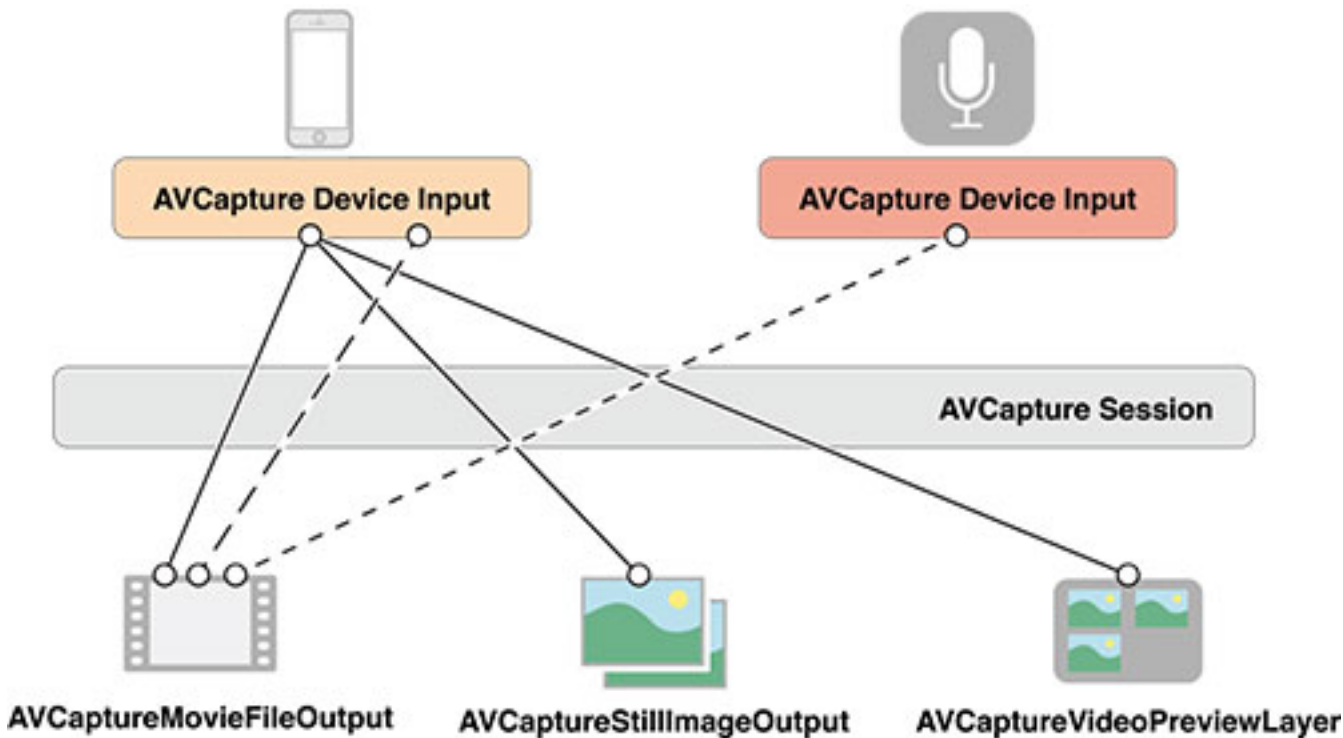
教程

OpenGL ES入门教程1-Tutorial01-GLKit (<http://www.jianshu.com/p/750fde1d8b6a>)
OpenGL ES入门教程2-Tutorial02-shader入门
(<http://www.jianshu.com/p/ee597b2bd399>)
OpenGL ES入门教程3-Tutorial03-三维变换 (<http://www.jianshu.com/p/87c5413c1fc7>)
OpenGL ES入门教程4-Tutorial04-GLKit进阶
(<http://www.jianshu.com/p/ed7fb9555839>)
OpenGL ES进阶教程1-Tutorial05-地球月亮 (<http://www.jianshu.com/p/a82f3f66ddddd>)
OpenGL ES进阶教程2-Tutorial06-光线 (<http://www.jianshu.com/p/4e1a28f23e75>)
OpenGL ES进阶教程3-Tutorial07-粒子效果 (<http://www.jianshu.com/p/b6d2441209f8>)
OpenGL ES进阶教程4-Tutorial08-帧缓存 (<http://www.jianshu.com/p/1193b98634a2>)
OpenGL ES进阶教程5-Tutorial09-碰碰车 (<http://www.jianshu.com/p/3b532f6fcedf>)
OpenGL ES进阶教程6-Tutorial10-平截体优化 (<http://www.jianshu.com/p/bc151ff65cef>)
OpenGL ES进阶教程7-Tutorial11-天空盒效果
(<http://www.jianshu.com/p/502a9c139b37>)
OpenGL ES进阶教程8-Tutorial12-obj文件和mtl文件解析
(<http://www.jianshu.com/p/64d9c58d8344>)
OpenGL ES实践教程1-Demo01-AVPlayer (<http://www.jianshu.com/p/64d9c58d8344>)
这一篇教程是摄像头采集数据和渲染，包括了三部分内容，渲染部分-OpenGL ES
(<http://www.jianshu.com/notebooks/2135411/latest>)，摄像头采集图像部分-
AVFoundation (<http://www.jianshu.com/notebooks/4884717/latest>)和图像数据创建纹理
部分-GPUImage (<http://www.jianshu.com/notebooks/4268718/latest>)。

核心思路

1、摄像头采集

AVFoundation的常用类介绍：
AVCaptureDevice 输入设备，包括摄像头、麦克风。
AVCaptureInput 输入数据源
AVCaptureOutput 输出数据源
AVCaptureSession 会话，协调输入与输出之间的数据流
AVCaptureVideoPreviewLayer 预览效果的layer



采集流程：

- 1、新建会话，设置图像大小；创建处理队列；

```
self.mCaptureSession = [[AVCaptureSession alloc] init];
self.mCaptureSession.sessionPreset = AVCaptureSessionPreset640x480;
mProcessQueue = dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_HIGH, 0);
```

- 2、新建摄像头输入，设置为前置摄像头；创建设备输入，并添加到会话；

```
AVCaptureDevice *inputCamera = nil;
NSArray *devices = [AVCaptureDevice devicesWithMediaType:AVMediaTypeVideo];
for (AVCaptureDevice *device in devices)
{
    if ([device position] == AVCaptureDevicePositionFront)
    {
        inputCamera = device;
    }
}
self.mCaptureDeviceInput = [[AVCaptureDeviceInput alloc] initWithDevice:inputCamera error:nil];
if ([self.mCaptureSession canAddInput:self.mCaptureDeviceInput]) {
    [self.mCaptureSession addInput:self.mCaptureDeviceInput];
}
```

- 3、创建数据输出，设置delegate和输出格式，添加到会话；

```
self.mCaptureDeviceOutput = [[AVCaptureVideoDataOutput alloc] init];
[self.mCaptureDeviceOutput setAlwaysDiscardsLateVideoFrames:NO];
self.mGLView.isFullyYUVRange = NO;
[self.mCaptureDeviceOutput setVideoSettings:[NSDictionary dictionaryWithObject:[NSNumber numberWithInt:kCVPixelFormatType_420YpCbCr8BiPlanarVideoRange] forKey:(id)kCVPixelBufferPixelFormatTypeKey]];
[self.mCaptureDeviceOutput setSampleBufferDelegate:self queue:mProcessQueue];
if ([self.mCaptureSession canAddOutput:self.mCaptureDeviceOutput]) {
    [self.mCaptureSession addOutput:self.mCaptureDeviceOutput];
}
AVCaptureConnection *connection = [self.mCaptureDeviceOutput connectionWithMediaType:AVMediaTypeVideo];
[connection setVideoOrientation:AVCaptureVideoOrientationPortraitUpsideDown];
```

思考1：这里的AVCaptureConnection有何作用？

- 4、开始录制；在delegate中接受图像帧数据；

- 开始会话

```
[self.mCaptureSession startRunning];
```

- 处理图像帧；

```
CVPixelBufferRef pixelBuffer = CMSampleBufferGetImageBuffer(sampleBuffer);
[self.mGLView displayPixelBuffer:pixelBuffer];
```

2、图像纹理数据创建

这一部分的代码参考自GPUImage的GPUImageVideoCamera类，YUV视频帧分为亮度和色度两个纹理，分别用GL_LUMINANCE格式和GL_LUMINANCE_ALPHA格式读取。

- 1、创建亮度纹理；

```
glActiveTexture(GL_TEXTURE0);
err = CVOpenGLTextureCacheCreateTextureFromImage(kCFAAllocatorDefault,
                                                _videoTextureCache,
                                                pixelBuffer,
                                                NULL,
                                                GL_TEXTURE_2D,
                                                GL_LUMINANCE,
                                                frameWidth,
                                                frameHeight,
                                                GL_LUMINANCE,
                                                GL_UNSIGNED_BYTE,
                                                0,
                                                &_lumaTexture);
```

- 2、配置亮度纹理属性

```
glBindTexture(CVOpenGLTextureGetTarget(_lumaTexture), CVOpenGLTexture
GetName(_lumaTexture));
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
```

- 3、创建色度纹理；

```
glActiveTexture(GL_TEXTURE1);
err = CVOpenGLTextureCacheCreateTextureFromImage(kCFAAllocatorDefault,
                                                _videoTextureCache,
                                                pixelBuffer,
                                                NULL,
                                                GL_TEXTURE_2D,
                                                GL_LUMINANCE_ALPHA,
                                                frameWidth / 2,
                                                frameHeight / 2,
                                                GL_LUMINANCE_ALPHA,
                                                GL_UNSIGNED_BYTE,
                                                1,
                                                &_chromaTexture);
```

- 4、配置色度纹理；

```
glBindTexture(CVOpenGLTextureGetTarget(_chromaTexture), CVOpenGLTextu
reGetName(_chromaTexture));
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
```

思考2： 这里为何输出的是YUV帧？ 如何配置输出的视频帧格式？

OpenGL ES渲染

OpenGL ES的渲染流程在前文多有介绍，这里不再赘述。讲讲自己遇到的问题。

- 1、黑屏

黑屏的现象出现多次，首先是

CVOpenGLTextureCacheCreateTextureFromImage failed (error: -6683)

纹理创建失败导致的黑屏，在正确配置好颜色格式，解决；

解决所有报错后，仍常黑屏；

检查纹理代码，正常；

检查颜色缓冲区代码，正常；

检查顶点坐标代码，正常；

检查纹理坐标代码，正常；

采用最后的手段，capture GPU Frame，查看GPU的状态信息。

发现，present的颜色缓冲区无效；惊讶之余，添加下面的代码，断点。

```
if ([EAGLContext currentContext] == _context) {
    [_context presentRenderbuffer:GL_RENDERBUFFER];
}
```

竟然没有调用，发现问题所在。

添加以下代码之后，问题解决。

```
[EAGLContext setCurrentContext:_context];
```

疑惑：为何之前调用过一次设置`context`之后，会需要再次调用`context`？代码其他地方并无设置`context`的地方。

解疑：因为处于新的线程！！

• 2、颜色不对

demo实现过程中遇到颜色不对的情况，图像的亮度没有问题，色度出现偏差，效果如下：



检查了创建纹理的过程，没有发现错误；

修改颜色空间，会导致颜色更加异常；

检查是否顶点着色器的偏移有误差，没有问题；

最后发现图片偏绿，在顶点着色器找到问题代码：

```
yuv.yz = (texture2D(SamplerUV, texCoordVarying).rg - vec2(0.5, 0.5));
```

正确的取值应该是ra，我写成了rg，导致图像偏绿。

总结

旋转图像的的数据是个耗性能的操作，如果是用AVAssetWriter写QuickTime movie文件，更好的做法是设置AVAssetWriterInput的transform属性，而不是修改AVCaptureVideoDataOutput，真正的去修改图像数据。
光看教程是学不会OpenGL ES的，下载教程自己改改代码，自己感兴趣的想法就去实现它。

还有就是，遇到问题多尝试，多查资料。如果绝望，那么就洗洗睡，明天说不定就解决了。

你也喜欢这种脑袋一直有问题在思考的感觉吗？

思考题

思考1: AVCaptureConnection可以使录制出来的图像上下颠倒；

参考GPUImage 的注释：

From the iOS 5.0 release notes:

In previous iOS versions, the front-facing camera would always deliver buffers in VCaptureVideoOrientationLandscapeLeft

and the back-facing camera would always deliver buffers in

AVCaptureVideoOrientationLandscapeRight.

思考2: 在前面的kCVPixelFormatType_420YpCbCr8BiPlanarVideoRange表明了

输出的颜色格式为YUV视频帧，并且颜色空间为(luma=[16,235] chroma=

[16,240])。

iOS通常支持三种格式：

kCVPixelFormatType_420YpCbCr8BiPlanarFullRange

kCVPixelFormatType_420YpCbCr8BiPlanarVideoRange

kCVPixelFormatType_32BGRA


如果遇到了 Failed to create IOSurface image (texture)

CVOpenGLTextureCacheCreateTextureFromImage failed (error: -6683)

这两个错误，一般是配置的颜色输出格与

CVOpenGLTextureCacheCreateTextureFromImage的参数不对应；

代码地址 (<https://github.com/loyinglin/LearnOpenGLS/tree/master/Demo02-%E6%91%84%E5%83%8F%E5%A4%B4%E9%87%87%E9%9B%86%E5%92%8C%E6%B8%B2%E6%9F%93>) - 你的star和fork是我最大的源动力，你的意见能让我走得更远。



落影loyinglin (/u/815d10a4bdce)

写了 171405 字，被 4768 人关注，获得了 个喜欢

(/u/815d10a4bdce) 写了 171405 字，被 4768 人关注，获得了 个喜欢

+ 关注

工程师一枚，喜欢思考，喜欢游戏，喜欢运动。做过什么已经不重要，未来的方向以及当下的准备是生活的...

赞赏是支持别人，也是加深自己的记忆。

赞赏支持



包萌萌 (/u/1ce5d631755c)

2楼 · 2016.08.01 09:16

(/u/1ce5d631755c)

楼主好，看过你的几篇文章，觉得写得很棒👍。想跟您请教一下，我要录制240帧的视频，必须要用到kCVPixelFormatType_420YpCbCr8BiPlanarFullRange和kCVPixelFormatType_420YpCbCr8BiPlanarVideoRange这两种格式，但是使用GPUImageMoiveWriter一直不能获得正确的输出，仔细检查了GPUImageMoiveWriter中的代码，觉得应该是要对createDataFBO中纹理处理的代码进行部分修改，我对这一块不熟悉，但工作又很急，能否指点一下。谢谢啦



赞



回复

落影loyinglin (/u/815d10a4bdce): @包萌萌 (/users/1ce5d631755c) 你可以把问题写在demo上，发给我。我看看。

2016.08.01 10:37 回复



添加新评论



无声的叹息 (/u/8853dde1ffc2)

4楼 · 2016.09.20 22:35

(/u/8853dde1ffc2)

楼主，看了您的博客，我的想法是尝试着获取到视频图像的亮度和灰度纹理ID，通过OpenGL进行处理绘制后，然后用AVAssetWriter写成一个MP4的视频。但是不知道该如何才能拿到绘制之后的pixelbuffer（stackoverflow上说绘制到纹理（opengl最后绘制成的貌似是RGB的），再拿到pixelbuffer，这点有点卡壳，framebuffer好像已经绑定了renderbuffer，不知道怎么绘制到纹理了）



赞



回复

落影loyinglin (/u/815d10a4bdce): @无声的叹息 (/users/8853dde1ffc2) 你如果是想用摄像头录制视频并写入mp4是不需要经过opengl的

2016.09.20 23:26 回复

无声的叹息 (/u/8853dde1ffc2): 嗯，看了您前面的文章有说过，官网也有类似的代码，用到的好像是AVDeviceCapture直接捕获。如果是一个视频文件，不知道您有没有好的处理方法。😁

2016.09.21 07:40 回复

落影loyinglin (/u/815d10a4bdce): @无声的叹息 (/users/8853dde1ffc2) 没懂什么意思

2016.09.21 08:14 回复



添加新评论

还有3条评论，[展开查看](#)



jdbbjd (/u/f74cf849e92c)

5楼 · 2016.10.10 16:04

(/u/f74cf849e92c)

楼主，看你的教程后受益很多，不过关于视频的采集和渲染使用的是yuv格式的，现在我要采集时是BGRA格式，在渲染时我遇到了困难，网上大部分都是渲染yuv格式的视频的，很少有人讲如何渲染BGRA格式的视频，请楼主指点一下，谢谢！



赞



回复

落影loyinglin (/u/815d10a4bdce): @jdbbjd (/users/f74cf849e92c) BGRA倒过来不就是ARGB，用CG重绘有考虑过吗

2016.10.10 19:43 回复



添加新评论



WSGNSLog (/u/ba5f297de9bc)

6楼 · 2016.10.25 14:20

(/u/ba5f297de9bc)

博主，能推荐一下OpenGL的学习资料么

👍 赞 💬 回复

落影loyinglin (/u/815d10a4bdce): @WSGNSLog (/users/ba5f297de9bc) 看看learnopengl
2016.10.25 15:12 💬 回复

WSGNSLog (/u/ba5f297de9bc): @落影loyinglin (/users/815d10a4bdce) 好的，多谢
2016.10.25 17:12 💬 回复

✍️ 添加新评论



o0阿拉斯加的狗0o (/u/679e2f66df65)
7楼 · 2016.12.22 22:57

(/u/679e2f66df65)

您好，这个例子的主要目的是什么啊？因为不用OpenGL ES，只用AVFoundation也能达到摄像头实时输出的效果啊。

👍 赞 💬 回复

落影loyinglin (/u/815d10a4bdce): @o0一路向北0o (/users/679e2f66df65) 采集不是重点，opengl es本来就不负责采集。重点是如何渲染
2016.12.23 08:23 💬 回复

o0阿拉斯加的狗0o (/u/679e2f66df65): @落影loyinglin (/users/815d10a4bdce) 博主，您好，有点困惑，能解答一下吗？谢谢了
2016.12.23 08:35 💬 回复

o0阿拉斯加的狗0o (/u/679e2f66df65): @落影loyinglin (/users/815d10a4bdce) 嗯，关于OpenGL ES的我看您的博客学习了很多，我说的是这个例子，这篇文章的例子我运行了一下，看着好像没有别的渲染特效，就是采集实时输出的效果啊，如果只是实时采集输出，为什么还要用OpenGL ES呢，您能解惑一下吗？
2016.12.23 09:01 💬 回复

✍️ 添加新评论 | 还有3条评论， 展开查看



沐有墨鱼丸 (/u/96196dae980d)
8楼 · 2017.01.17 11:47

(/u/96196dae980d)

你好 你的文章对我很有帮助，有个问题请教一下，绘制后置相机的图像是左右颠倒的，请问如何修正，谢谢

👍 赞 💬 回复

落影loyinglin (/u/815d10a4bdce): @沐有墨鱼丸 (/users/96196dae980d) 可以添加左右翻转的矩阵。
2017.01.17 14:56 💬 回复

沐有墨鱼丸 (/u/96196dae980d): @落影loyinglin (/users/815d10a4bdce) 非常感谢，解决了
2017.01.17 17:43 💬 回复

✍️ 添加新评论



88486969 (/u/45f0c6e690c1)
9楼 · 2017.02.24 08:32

(/u/45f0c6e690c1)

这个教程可以优化一下子，采集成 kCVPixelFormatType_32BGRA 这种格式，

```
CVReturn cvRet =
CVOpenGLTextureCacheCreateTextureFromImage(kCFAllocatorDefault,
_glTextureCache,
pixelBuffer,
NULL,
GL_TEXTURE_2D,
GL_RGBA,
```



```
GLuint glTexture = CVOpenGLESTextureGetName(_glOriginalTexture);
glBindTexture(GL_TEXTURE_2D , _glTextureOriginalIn);
```


这样就直接可以了，不用 单独 Y，UV，然后再 YUV to RGBA
， shader 里面日后可以做 4:3 to 16:9

👍 赞 💬 回复


落影loyinglin (/u/815d10a4bdce): @88486969 (/users/45f0c6e690c1) 谢谢 非常好的建议
2017.02.24 14:59 💬 回复

孙金亮 (/u/b6e7a5a68ab6): 优化好的 deomo, 您能发我一份吗,我项目用,急求
2017.07.18 10:40 💬 回复

✍️ 添加新评论

 青春微凉来时路 (/u/22d32b2cff94)
10楼 · 2017.06.22 12:36
(/u/22d32b2cff94)
您好,请问旋转手机的时候 如何让画面不跟着旋转呢?

👍 赞 💬 回复

 S_L (/u/5fd256ddf79a)
11楼 · 2017.08.08 11:43
(/u/5fd256ddf79a)
有个问题想要请教啊，着色器这里懵的一塌糊涂啊，.fsh 和.vsh两个文件到底是怎么工作的，里面的内容应该写点什么？里面的值是从哪儿获取的？ 还有尤其是.fsh里面的
uniform mat3 colorConversionMatrix; 这个变量是干什么用的？ main 里面的
yuv.x = (texture2D(SamplerY, texCoordVarying).r);
yuv.yz = (texture2D(SamplerUV, texCoordVarying).ra - vec2(0.5, 0.5));
rgb = colorConversionMatrix * yuv;
都代表什么意思？好像网上相关的资料很少啊，希望能够得到解答，谢谢！

👍 赞 💬 回复

被以下专题收入，发现更多相似内容


-  OpenGL ES (/c/71a20298524f?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS Dev... (/c/3233d1a249ca?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS学习 (/c/1332c736fe39?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS 开发 (/c/2ffaa203eb6a?utm_source=desktop&utm_medium=notes-included-collection)
-  OPenGL ... (/c/408442c9c764?utm_source=desktop&utm_medium=notes-included-collection)
-  OpenGL+... (/c/5d2c87603bd3?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS进阶 (/c/1d9c0540ff8e?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

 登录/注册

为你个性化推荐内容

(/sign_in?utm_source=desktop&utm_medium=note-bind)

 下载简书App

随时随地发现和创作内容

(/apps/download?utm_source=desktop&utm_medium=click-note-bottom-bind)