

Branch: master    zen / iOS学习：AVFoundation 视频流处理 / iOS学习：AVFoundation 视频流处理 .md     

 100mango on May 27 细微更正

1 contributor

232 lines (151 sloc)    8.323 kB

[Raw](#)    [Blame](#)    [History](#)            

# iOS学习：AVFoundation 视频流处理

## 框架

首先我们从整体对所需框架做个初步了解。

AVFoundation在相关框架栈中的的位置：

Media Player

UIKit

Audio-only classes

AV Foundation

Core Audio

Core Media

Core Animation

为了捕捉视频,我们需要这样几种类（与其它的子类）。

- AVCaptureDevice 代表了输入设备,例如摄像头与麦克风。
- AVCaptureInput 代表了输入数据源
- AVCaptureOutput 代表了输出数据源
- AVCaptureSession 用于协调输入与输出之间的数据流

并且还有AVCaptureVideoPreviewLayer提供摄像头的预览功能

可以用这样一幅图来概述：

<>



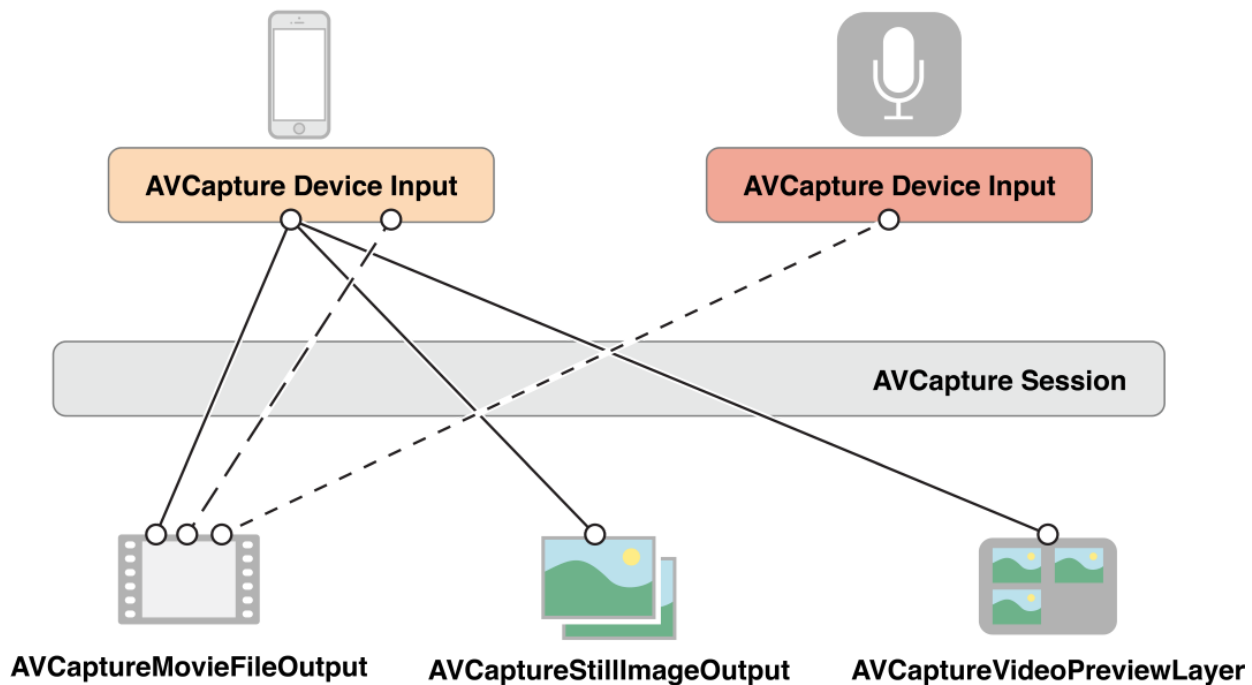








https://github.com/100mango/zen/blob/master/iOS%E5%AD%A6%E4%B9%A0%EF%BC%9AAVFoundation%20%E8%A7%86%E9%A2%91%E6%B5%81... 1/6



## 例子

实际应用AVFoundation来捕捉视频流并不复杂。

Talk is Cheap, Show me the Code.

我们用代码简单地描述用AVFoundation捕捉视频的过程,其他捕捉音频,静态图像的过程也是大同小异的。

### 1. 创建AVCaptureSession。

作为协调输入与输出的中心,我们第一步需要创建一个Session

```
AVCaptureSession *session = [[AVCaptureSession alloc] init];
```

### 2. 创建AVCaptureDevice

创建一个AVCaptureDevice代表代表输入设备。在这里我们制定设备用于摄像。

```
AVCaptureDevice *device = [AVCaptureDevice defaultDeviceWithMediaType:AVMediaTypeVideo];
```

### 3. 创建AVCaptureDeviceInput,并添加到Session中

我们需要使用AVCaptureDeviceInput来让设备添加到session中, AVCaptureDeviceInput负责管理设备端口。我们可以理解它为设备的抽象。一个设备可能可以同时提供视频和音频的捕捉。我们可以分别用AVCaptureDeviceInput来代表视频输入和音频输入。

```
NSError *error;
AVCaptureDeviceInput *input = [AVCaptureDeviceInput deviceInputWithDevice:device error:&error];
[session addInput:input];
```

### 4. 创建AVCaptureOutput

为了从session中取得数据,我们需要创建一个AVCaptureOutput

```
AVCaptureVideoDataOutput *output = [[AVCaptureVideoDataOutput alloc] init];
```

### 5. 设置output delegate,将output添加至session,在代理方法中分析视频流

为了分析视频流,我们需要为output设置delegate,并且指定delegate方法在哪个线程被调用。需要主要的是,线程必须是

串行的,确保视频帧按序到达。

```
videoDataOutputQueue = dispatch_queue_create("VideoDataOutputQueue", DISPATCH_QUEUE_SERIAL);

[videoDataOutput setSampleBufferDelegate:self queue:videoDataOutputQueue];

[session addOutput:videoDataOutput];
```

我们可以在delegate方法中分析视频流。

```
captureOutput:didOutputSampleBuffer:fromConnection:,
```

## 6. 开始捕捉

```
[session startRunning];
```

通过上面的简单例子,我么可以看出使用AVFoundation来捕捉视频流并不是十分复杂。重点是使用的过程需要了解配置的细节,还有性能问题。

## 实战

学习基础知识过后,让我们用个具体例子来进行阐明。

我们来做一个基于AVFoundation二维码识别应用: QRCatcher



应用已经上架AppStore 并且完整开源

项目架构:

```
| - Model
    | - URLEntity
| - View
    | - QRURLTableViewCell
    | - QRTabBar
| - Controller
    | - QRCatchViewController
    | - QRURLViewController
| - Tools
    | - NSString+Tools
    | - NSObject+Macro
```

项目并不复杂。典型的MVC架构。

- Model层只有一个URLEntity用于存储捕捉到的URL信息。这次项目也顺便学习了一下CoreData。感觉良好,配合NSFetchedResultsController工作很幸福。
- View层则是一个TableViewCell和Tabbar,继承Tabbar主要用于改变tabbar高度。
- Controller层中QRCatchViewController负责捕捉与存储二维码信息, QRURLViewController负责展示与管理收集到的URL信息。
- Tools则是一些辅助方便开发的类。出自我自己平时使用收集编写维护的一个工具库 ([开源链接](#)) 在这个项目中主要用以检查URL是否合法,判断设备类型等。



```
[self insertURLEntityWithURL:metadata.stringValue];
self.stringLabel.text = metadata.stringValue;
}
else
{
    self.stringLabel.text = metadata.stringValue;
}
}
}
```

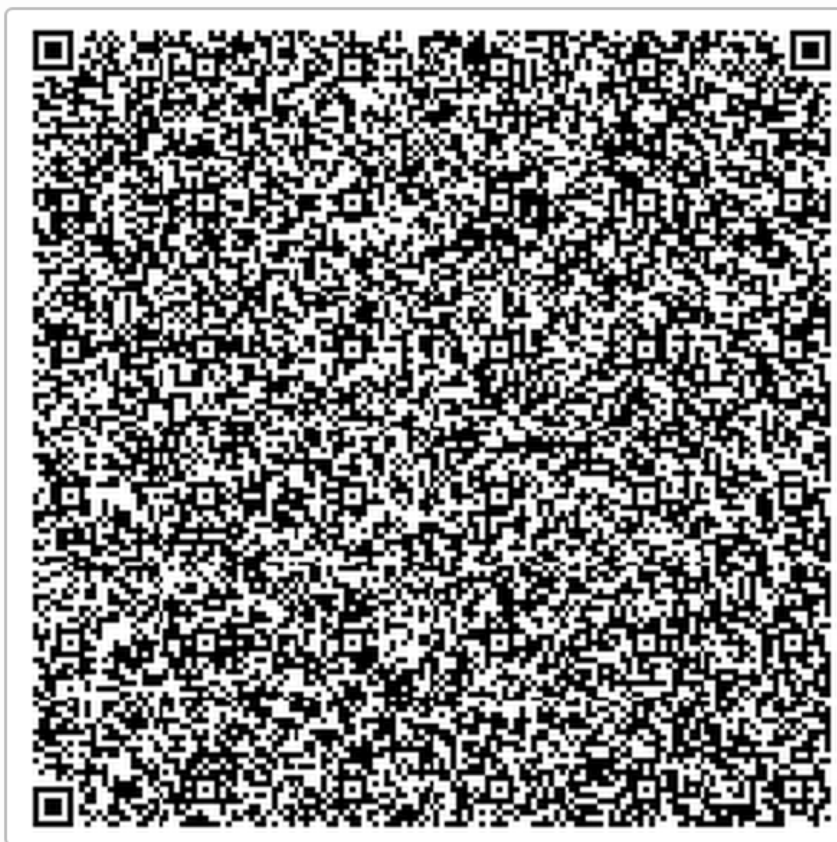
我们需要在代理方法里面接收数据,并根据自己的需求进行处理。在这里我简单地进行了URL的测试,如果是的话则打开safari进行浏览。

## 总结

在这里仅仅是通过一个二维码的应用来展示AVFoundation处理视频流能力。事实上, AVFoundation能够做得更多。能够进行剪辑,处理音轨等功能。如果我们需要对视频与音频相关的事务进行处理,不妨在着手处理,寻找第三方解决方案前,看看这个苹果公司为我们带来的强大模块。

PS:最后来点好玩的东西:

大家可以用手机上的微信扫一扫这张二维码(莫慌,虽然稍微有点密集):



自己用iOS上微信的最新版本进行扫描(版本6.2.0 完成这篇文章的时间是2015.5.27)

是无法完成扫描解析的。自己随机输入一些文字生成二维码进行测试,发现生成二维码的文字如果复杂到一定程度,微信的扫一扫模块是无法扫描解析的。而用QRMatcher是可以准确扫描成功的:)。欢迎下载试用。(请不要嫌我为什么要生成这些密集的二维码:)