作者 落影loyinglin (/users/815d10a4bdce) 2016.09.06 10:45
写了79631字，被1298人关注，获得了692个喜欢
(/users/815d10a4bdce)

H.264学习笔记

H.264组成

- 1、网络提取层 (Network Abstraction Layer，NAL)
- 2、视讯编码层 (Video Coding Layer，VCL)
    - a.**H.264/AVC影像格式阶层架构**
    - b.**Slice的编码模式**

      (1) I -slice: slice的全部MB都采用intra-prediction的方式来编码；

      (2) P-slice: slice中的MB使用intra-prediction和inter-prediction的方式来编码，但每一个inter-prediction block最多只能使用一个移动向量；

      (3) B-slice:与P-slice类似，但每一个inter-prediction block可以使用二个移动向量。B-slice的‘B’是指Bi-predictive（双向预测），除了可由前一张和后一张影像的I(或P、B)-slice外，也能从前二张不同影像的I(或P、B)-slice来做inter-prediction。

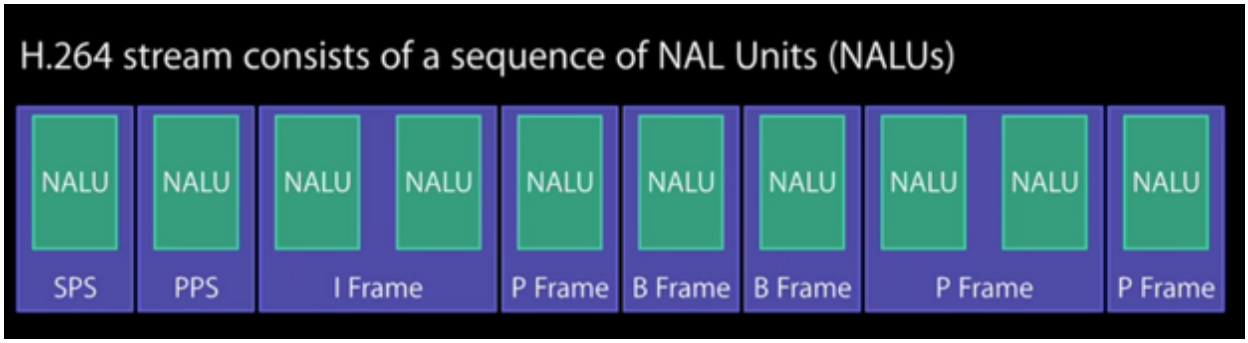
      (4) SP-slice:即所谓的Switching P slice，为P-slice的一种特殊类型，用来串接两个不同bitrate的bitstream；

      (5) SI-slice: 即所谓的Switching I slice，为I-slice的一种特殊类型，除了用来串接两个不同content的bitstream外，也可用来执行随机存取(random access)来达到网络VCR的功能
    - c、**画面内预测技术(Intra-frame Prediction)**
    - d、**画面间预测技术(Inter-frame Prediction)**

H.264介绍 (http://blog.csdn.net/gl1987807/article/details/11945357)

码流结构

H.264的功能分为两层，视频编码层（VCL）和网络提取层（NAL）VCL数据即被压缩编码后的视频数据序列。在VCL数据要封装到NAL单元中之后，才可以用来传输或存储。



NALU （Network Abstraction Layer Unit ）

- SPS：序列参数集，作用于一系列连续的编码图像；
- PSS：图像参数集，作用于编码视频序列中一个或多个独立的图像；

> 参数集是一个独立的数据单位，不依赖于参数集外的其他句法元素。一个参数集不对应某一个特定的图像或序列，同一序列参数集可以被多个图像参数集引用，同理，同一个图像参数集也可以被多个图像引用。只在编码器认为需要更新参数集的内容时，才会发出新的参数集。

NALU根据nal_unit_type的类型，可以分为：VCL的NAL单元和非VCL的NAL单元，详情如下：

```
NSString * const naluTypesStrings[] =
{
    @"0: Unspecified (non-VCL)",
    @"1: Coded slice of a non-IDR picture (VCL)",    // P frame
    @"2: Coded slice data partition A (VCL)",
    @"3: Coded slice data partition B (VCL)",
    @"4: Coded slice data partition C (VCL)",
    @"5: Coded slice of an IDR picture (VCL)",      // I frame
    @"6: Supplemental enhancement information (SEI) (non-VCL)",
    @"7: Sequence parameter set (non-VCL)",         // SPS parameter
    @"8: Picture parameter set (non-VCL)",          // PPS parameter
    @"9: Access unit delimiter (non-VCL)",
    @"10: End of sequence (non-VCL)",
    @"11: End of stream (non-VCL)",
    @"12: Filler data (non-VCL)",
    @"13: Sequence parameter set extension (non-VCL)",
    @"14: Prefix NAL unit (non-VCL)",
    @"15: Subset sequence parameter set (non-VCL)",
    @"16: Reserved (non-VCL)",
    @"17: Reserved (non-VCL)",
    @"18: Reserved (non-VCL)",
    @"19: Coded slice of an auxiliary coded picture without partitioning (non-VCL)",
    @"20: Coded slice extension (non-VCL)",
    @"21: Coded slice extension for depth view components (non-VCL)",
    @"22: Reserved (non-VCL)",
    @"23: Reserved (non-VCL)",
    @"24: STAP-A Single-time aggregation packet (non-VCL)",
    @"25: STAP-B Single-time aggregation packet (non-VCL)",
    @"26: MTAP16 Multi-time aggregation packet (non-VCL)",
    @"27: MTAP24 Multi-time aggregation packet (non-VCL)",
    @"28: FU-A Fragmentation unit (non-VCL)",
    @"29: FU-B Fragmentation unit (non-VCL)",
    @"30: Unspecified (non-VCL)",
    @"31: Unspecified (non-VCL)",
};
```

nal_unit_type	NAL类型	C
0	未使用	
1	不分区、非 IDR 图像的片	2, 3, 4
2	片分区 A	2
3	片分区 B	3
4	片分区 C	4
5	IDR 图像中的片	2, 3
6	补充增强信息单元（SEI）	5
7	序列参数集	0
8	图像参数集	1
9	分界符	6
10	序列结束	7
11	码流结束	8
12	填充	9
13..23	保留	
24..31	未使用	

官方文档

码流结构 (http://blog.csdn.net/gl1987807/article/details/11946025)

iOS与H.264

1、视频相关的框架

由上到下：

- AVKit
- AVFoundation
- Video Toolbox
- Core Media
- Core Video

其中的AVKit和AVFoudation、VideoToolbox都是使用**硬编码和硬解码**。

2、相关类介绍

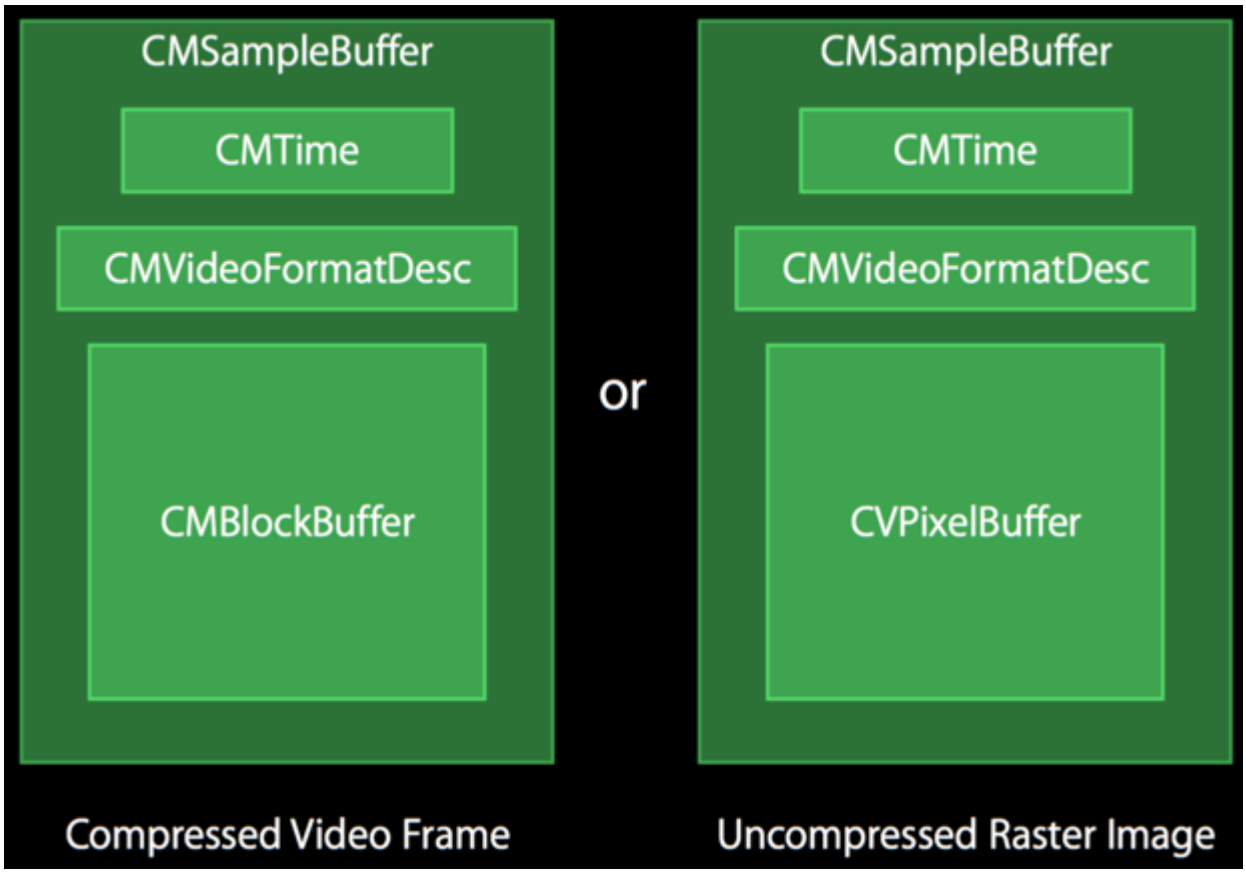
- CVPixelBuffer： 包含未压缩的像素数据，包括图像宽度、高度等；
- CVPixelBufferPool： CVPixelBuffer的缓冲池，因为CVPixelBuffer的创建和销毁代价很大；
- pixelBufferAttributes： CFDictionary包括宽高、像素格式（RGBA、YUV）、使用场景（OpenGL ES、Core Animation）
- CMTime： 64位的value，32位的scale，media的时间格式；
- CMVideoFormatDescription： video的格式，包括宽高、颜色空间、编码格式等；对于H.264的视频，PPS和SPS的数据也在这里；
- CMBlockBuffer：未压缩的图像数据；
- CMSampleBuffer： 存放一个或者多个压缩或未压缩的媒体文件；
- CMClock：时间源

> A timing source object.

- CMTimebase：时间控制器，可以设置rate和time；

> A timebase represents a timeline that clients can control by setting the rate and time. Each timebase has either a master clock or a master timebase. The rate of the timebase is expressed relative to its master.

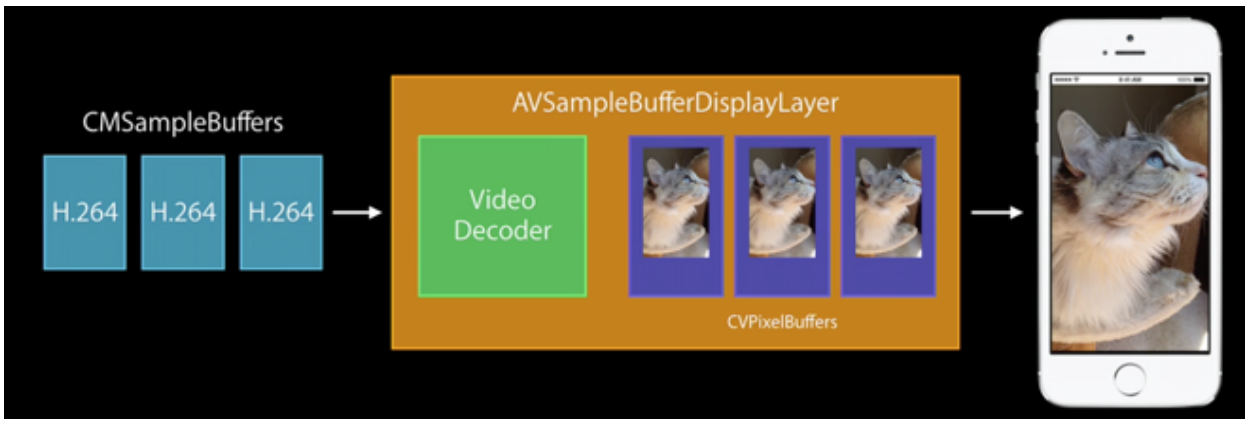
CMSampleBuffer的结构：



可以包含已压缩数据（CMBlockBuffer）或未压缩数据（CVPixelBuffer）及相关描述信息

3、AVKit

使用AVSampleBufferDisplayLayer显示H.264码流



- 初始化

```
self.videoLayer = [[AVSampleBufferDisplayLayer alloc] init];
self.videoLayer.bounds = self.bounds;
self.videoLayer.position = CGPointMake(CGRectGetMidX(self.bounds), CGRectGetMidY(
self.videoLayer.videoGravity = AVLayerVideoGravityResizeAspect;
self.videoLayer.backgroundColor = [[UIColor greenColor] CGColor];
//set Timebase
CMTimebaseRef controlTimebase;
CMTimebaseCreateWithMasterClock( CFAllocatorGetDefault(), CMClockGetHostTimeClock
self.videoLayer.controlTimebase = controlTimebase;
CMTimebaseSetTime(self.videoLayer.controlTimebase, CMTimeMake(5, 1));
CMTimebaseSetRate(self.videoLayer.controlTimebase, 1.0);
// connecting the videolayer with the view
[[self layer] addSublayer:_videoLayer];
```

- 传入SampleBuffer

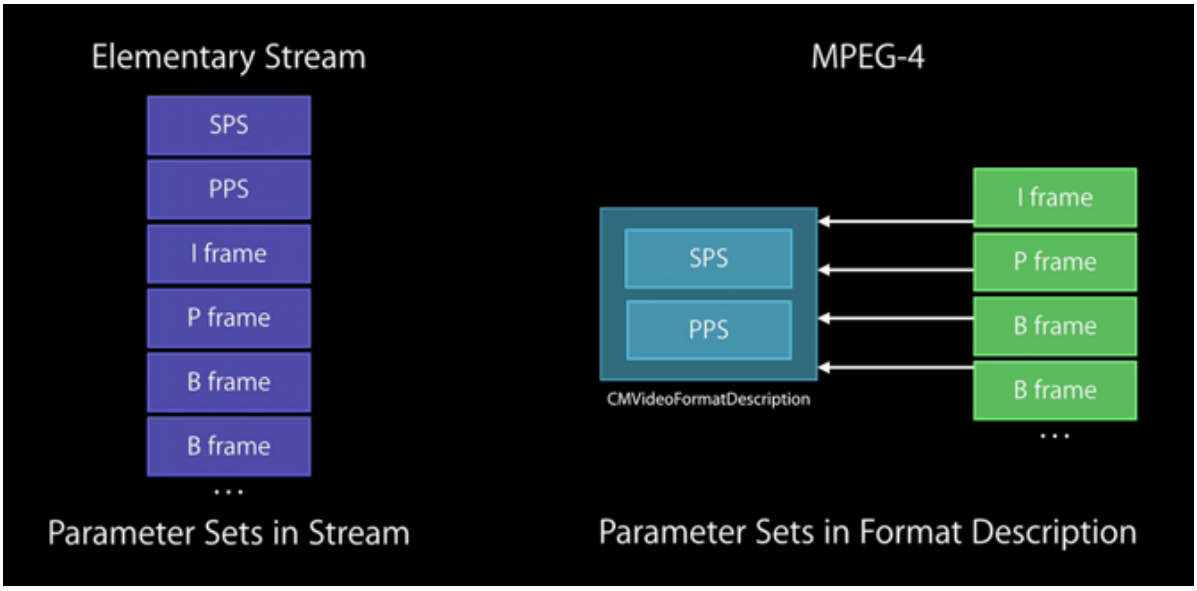
```
__block AVAssetReaderTrackOutput *outVideo = [AVAssetReaderTrackOutput assetReade
if( [assetReaderVideo startReading] )
{
 [_videoLayer requestMediaDataWhenReadyOnQueue: assetQueue usingBlock: ^{
     while( [_videoLayer isReadyForMoreMediaData] )
     {
         CMSampleBufferRef *sampleVideo = [outVideo copyNextSampleBuffer];
         [_videoLayer enqueueSampleBuffer:sampleVideo.data];
     }
 }];
}
```

4、MPEG-4封装的H.264码流格式

H.264的原始码流 与 MPEG-4封装的H.264码流格式不同在于：

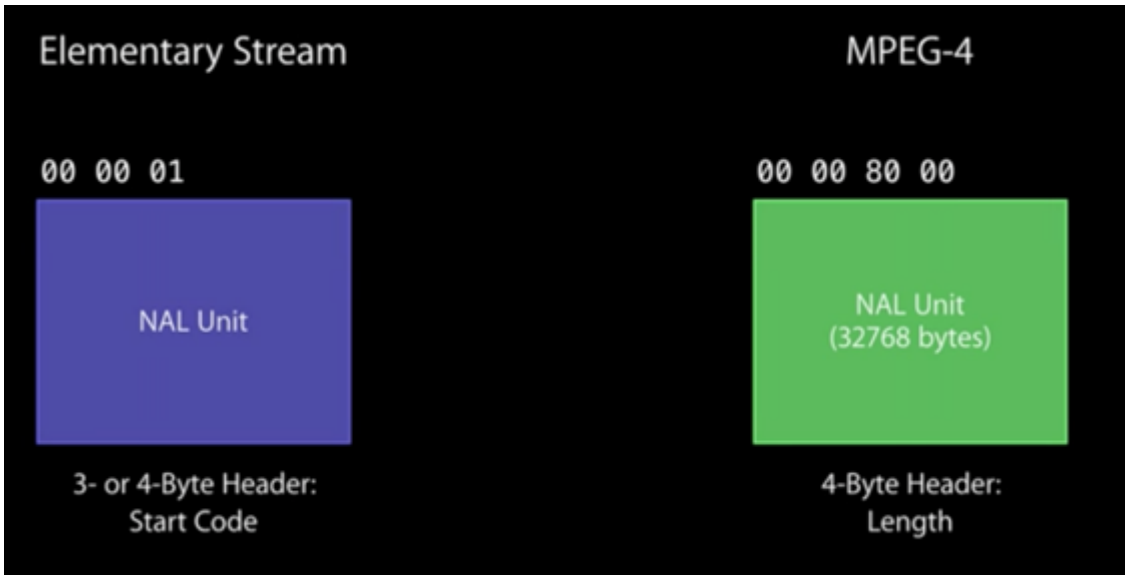
- SPS和PPS被统一
- 需要用 `CMVideoFormatDescriptionCreateFromH264ParameterSets` 方法，统一PPS和SPS

(/apps)



- 头字节表示帧的长度

  （原来的为00 00 01 或者 00 00 00 01）



当我们需要原始H.264码流包装成CMSampleBuffer时，我们可以按照以下步骤：

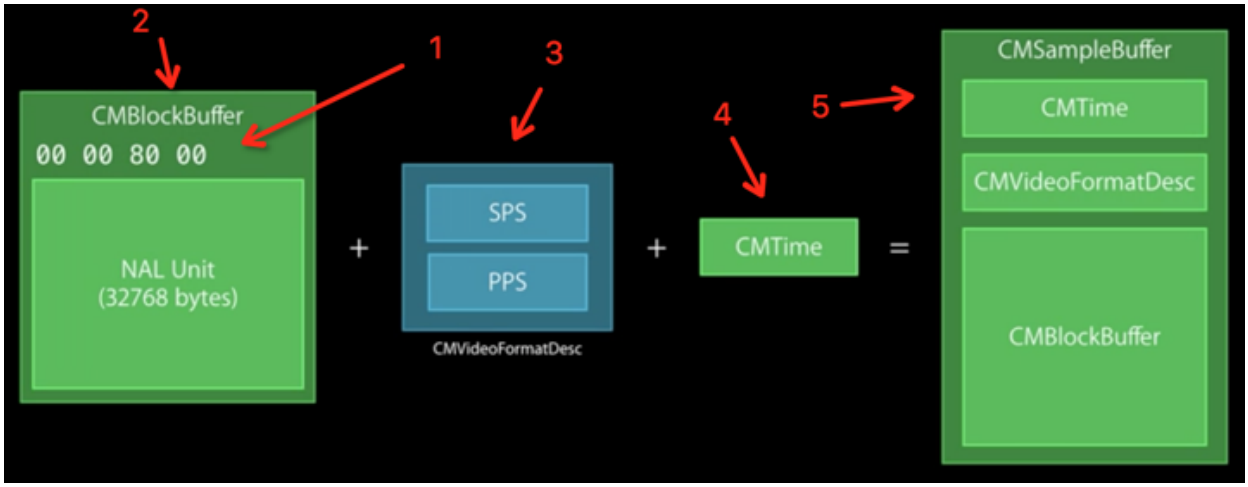
1、替换头字节长度；

2、用CMBlockBuffer把NALUnit包装起来；

3、把SPS和PPS包装成CMVideoFormatDescription；

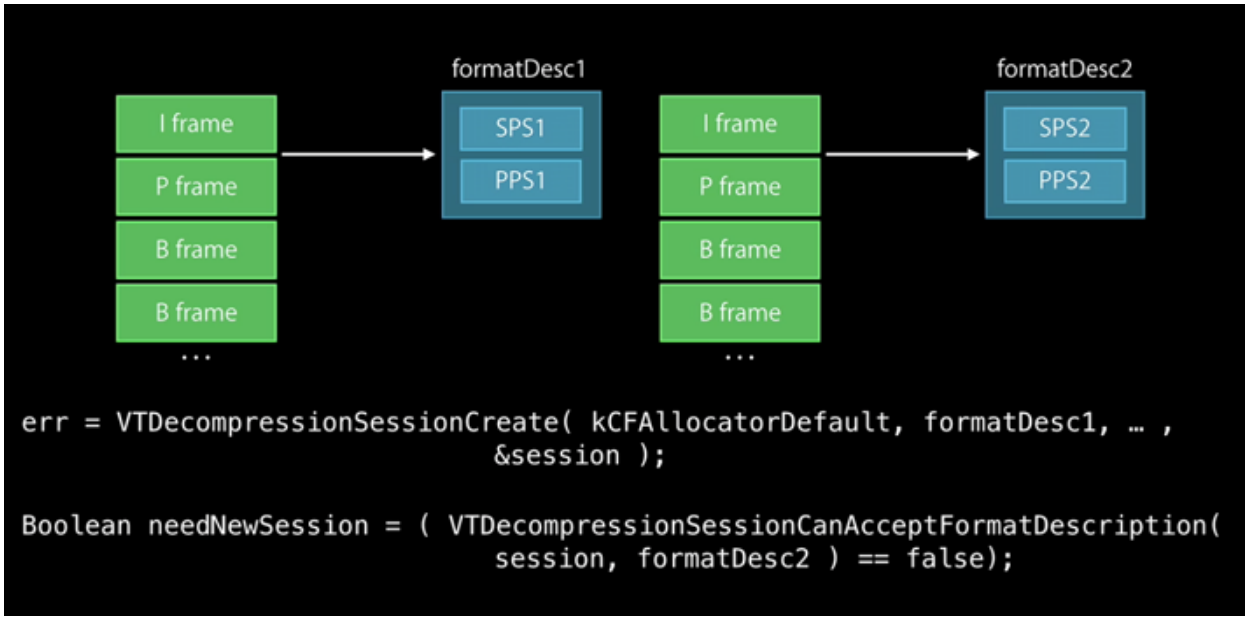
4、添加CMTime时间；

5、创建CMSampleBuffer；



根据H.264原始码流创建CMSampleBuffer

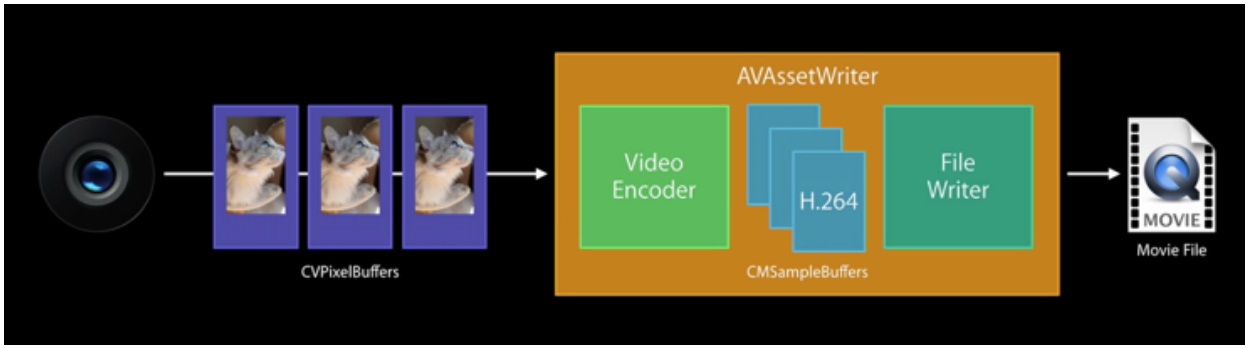
当我们需要更新SPS和PPS的时候，调用 `VTDecompressionSessionCanAcceptFormatDescription` 判断是否能接受新的SPS和PPS；如果不能接受，那么需要新建session来处理frame，注意销毁原来的session；



更新SPS和PPS

5、采集摄像头数据

从摄像头采集数据，并用AVAssetWriter写入movieFile

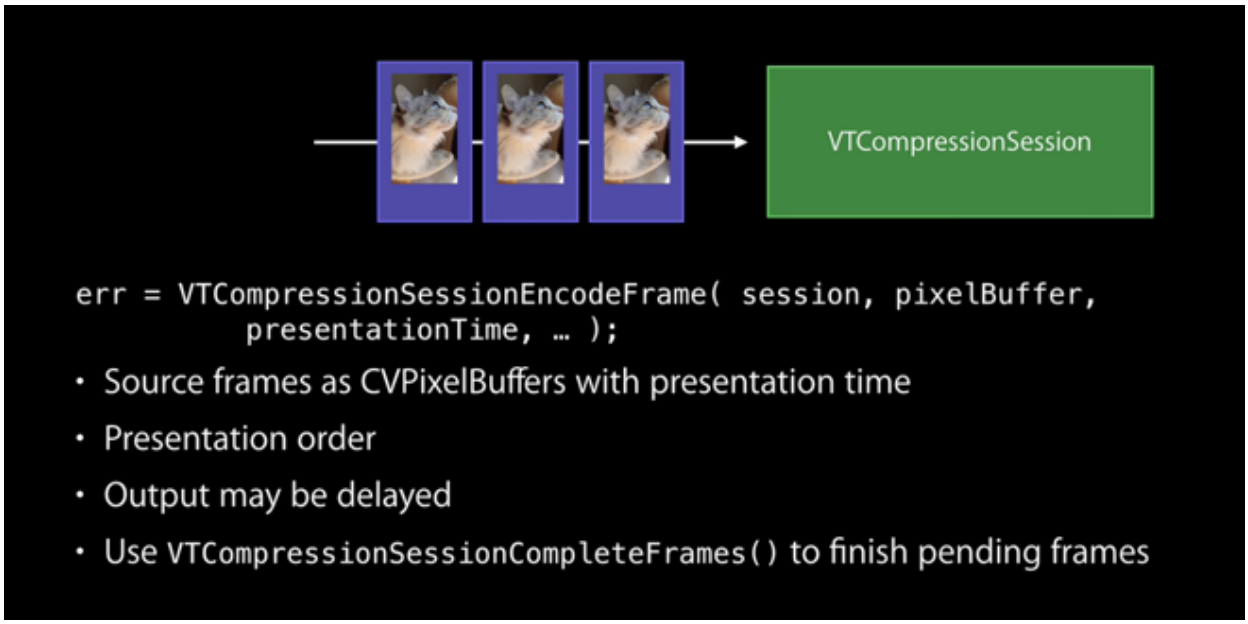


摄像头采集并写入movieFile

从摄像头采集数据，并VideoToolbox硬编码，获取压缩后的码流

- 按照显示顺序来，添加显示时间；
- 时间只能加不能减，不能重复；
- 异步的请求；（H.264的帧间预测）
- 没有帧之后需要调用complete；

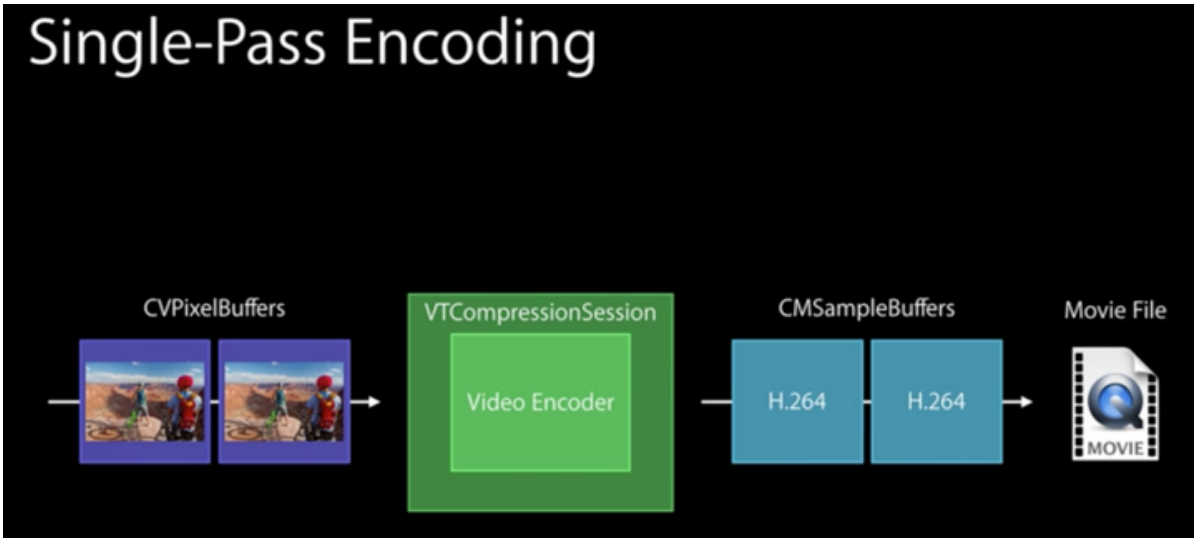
压缩后的码流是MPEG-4封装格式下的码流，要转换成原始码流的格式。
调用 `CMVideoFormatDescriptionGetH264ParameterSetAtIndex`
获取视频的PPS和SPS



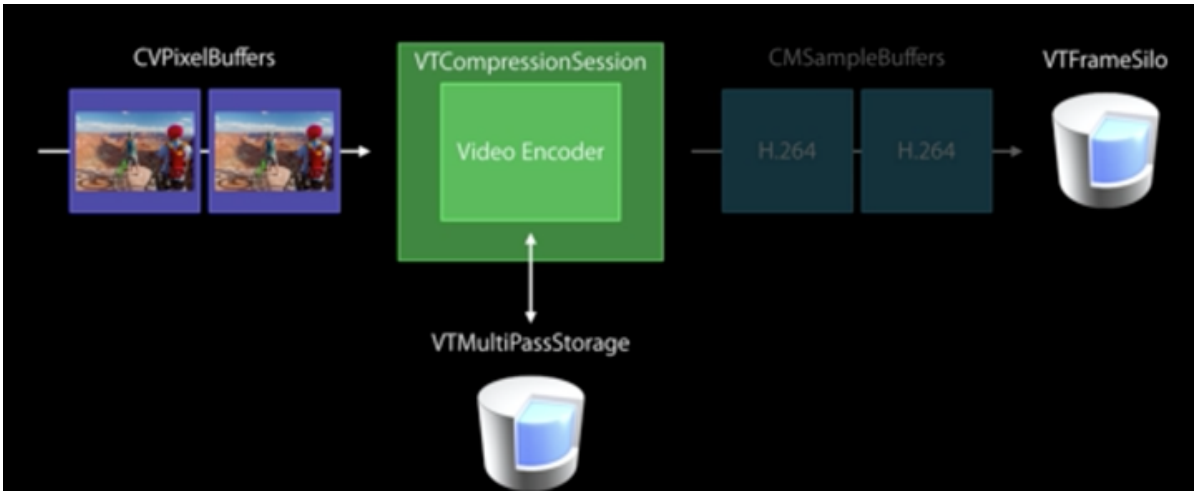
摄像头采集并生成H.264码流

6、Single-Pass和Multi-Pass编码

- Single-Pass编码



- Multi-Pass编码



AVAssetExportSession 优先采用多通道编码，不行再使用单通道编码；
Multi-passes的介绍 (http://www.jianshu.com/p/21291d983953)

其他零碎的知识

视频码率是视频数据（视频色彩量、亮度量、像素量）每秒输出的位数。一般用的单位是kbps。
由于不同的系统会有不同的模式，为了统一，规定在网络传输中使用大端模式，这就是**网络字节序**。
RTP协议：实时传送协议（Real-time Transport Protocol或简写RTP，也可以写成RTTP）是一个网络传输协议。RTP协议详细说明了在互联网上传递音频和视频的标准数据包格式。
RTCP协议：实时传输控制协议（Real-time Transport Control Protocol或RTP Control

Protocol或简写RTCP）是实时传输协议（RTP）的一个姐妹协议。
RTSP协议：RTSP（Real Time Streaming Protocol）是用来控制声音或影像的多媒体串流协议。

> RTSP发起/终结流媒体、RTP传输流媒体数据 、RTCP对RTP进行控制，同步。

RTMP协议：RTMP（the Real-time Messaging Protocol）协议作为客户端和服务器端的传输协议，这是一个专门为高效传输视频、音频和数据而设计的 TCP/IP 协议。
HLS协议： HTTP Live Streaming（HLS）是苹果公司(Apple Inc.)实现的基于HTTP的流媒体传输协议。

RTP封包H.264码流 (http://blog.csdn.net/gl1987807/article/details/11937595)
各种协议 (http://blog.csdn.net/tttyd/article/details/12032357/)

总结

如果想更深入学习，可以看H.264标准中文版的文档。

WWDC 2014 - Videos - Apple Developer — AirPrint is widely supported by all the well-known printer companies. Learn how quick and easy it is to support AirPrint in your app. Discover what's new and improved in the iOS printing system. (https://developer.apple.com/videos/wwdc2014/#513)
落影loyinglin (/users/815d10a4bdce)· developer.apple.com →
(https://developer.apple.com/videos/wwdc2014/#513)

objective c - How to use VideoToolbox to decompress H.264 video stream - Stack Overflow
— I had a lot of trouble figuring out how to use Apple's Hardware accelerated video framework to decompress an H.264 video stream. After a few weeks I figured it out and wanted to share an extensive example since I couldn't find one. (http://stackoverflow.com/questions/29525000/how-to-use-videotoolbox-to-decompress-h-264-video-stream)
落影loyinglin (/users/815d10a4bdce)· stackoverflow.com →
(http://stackoverflow.com/questions/29525000/how-to-use-videotoolbox-to-decompress-h-264-video-stream)

> **如果你觉得我的文章有所帮助，可以请我喝瓶可乐，Or点个喜欢或关注。**

¥ 打赏支持

♡ 喜欢 | 7

分享到微博 分享到微信
更多分享

4条评论 （ 按时间正序 · 按时间倒序 · 按喜欢排序 ） 添加新评论 (/sign_in)

落影loyinglin (/users/815d10a4bdce)
2楼 · 2016.09.06 10:46 (/p/8de09a551a66/comments/4040388#comment-4040388)
(/users/815d10a4bdce)
官方文档网上有。WWDC看扩展。

♡ 喜欢(0) 回复

酷走天涯 (/users/1c37e216663b)
3楼 · 2016.09.06 12:53 (/p/8de09a551a66/comments/4042740#comment-4042740)
(/users/1c37e216663b)
很详细😄

♡ 喜欢(0) 回复

> 落影loyinglin (/users/815d10a4bdce)： @酷走天涯 (/users/1c37e216663b) 官方文档给力
> 2016.09.06 13:28 (/p/8de09a551a66/comments/4043361#comment-4043361) 回复
>
> 添加新回复

AlexueQ (/users/f3320ae1a6e6)
5楼 · 2016.09.09 09:11 (/p/8de09a551a66/comments/4091848#comment-4091848)
(/users/f3320ae1a6e6)
喜欢

♡ 喜欢(0) 回复

登录后发表评论 (/sign_in)

被以下专题收入，发现更多相似内容：

程序员 (/collection/NEt52a)

如果你是程序员，或者有一颗喜欢写程序的心，喜欢分享技术干货、项目经验、程序员日常趣事等等，欢迎投稿《程序员》专题。 专题主编：小...

24781篇文章 (/collection/NEt52a) · 185858人关注

添加关注 (/sign_in)

(/collection/NEt52a)

iOS Developer (/collection/3233d1a249ca)

分享 iOS 开发的知识，解决大家遇到的问题，讨论iOS开发的前沿，欢迎大家投稿

13738篇文章 (/collection/3233d1a249ca) · 26997人关注

添加关注 (/sign_in)

(/collection/3233d1a249ca)

iOS 开发 (/collection/2ffaa203eb6a)

8228篇文章 (/collection/2ffaa203eb6a) · 7094人关注

添加关注 (/sign_in)

(/collection/2ffaa203eb6a)