深入浅出理解视频编码H264结构（内涵福利） - 简书

原文链接

引言：

在国内直播"欣欣向荣"(ps: 其实大多都亏钱，为的就是炒概念)的年代，相信很多小伙伴也投入了技术的浩瀚大洋当中(ps: 其实就是搬砖)，日复一日，音/视频的神秘面纱开始让更多的小伙伴扯下，而本博主，也只是刚窥探门道，慢慢摸索。好了，废话不扯，我们今天就来说说我们经常在视频编码当中用到的 H.264编码格式 的结构，相信 H.264 这个东西很多小伙伴都不陌生了，也有着自己的理解，但这东西颇为巨大，里面算法千千万万，博主也不会讲太高深的东西，只是让各位小伙伴慢慢理解理解 H.264的主体机构，如果没有兴趣的小伙伴请绕道，如果有技术大牛，请指正本博主那愚钝的脑袋。

首先来一段大家都熟悉的官方话来介绍一下 H.264

> H.264: H.264/AVC项目的目的是为了创建一个比以前的视频压缩标准，在更低的比特率的情况下依然能够提供良好视频质量的标准（如，一半或者更少于MPEG-2,H.263,或者MPEG-4 Part2 ）。同时，还要不会太大的增加设计的复杂性。
> 优势：
> 1）网络亲和性，即可适用于各种传输网络
> 2）高的视频压缩比，当初提出的指标是比 H.263，MPEG-4，约为它们的 2 倍，现在都已基 实现;

那么很明显，什么时候需要到压缩呢？当然是文件体积太大的时候啦，我们想想，所谓的视频，就是像小时候的连环画一样，在一秒内翻过 24 张以上的图片，就感觉图像是连续的了，这就是视频的原理。但是大家有没有想过，一张图片有多大呢？我们的屏幕分辨率按 1280 * 720 算的话，一秒钟的视频大概就 2.64 MB 了，大家想想，我们大部分的小伙伴为了下载个小嗨片省吃俭用才开了个 1M 的网线，然后连个直播都看不了是什么感觉。那肯定不能这样了，所以我们要进行压缩，而 H.264 不仅压缩比比较高，对网络的兼容性也非常好，所以大多数人做直播也就选择了 H.264 作为编码格式了。

编码流程：

那么 H.264 其编解码流程是怎么样的呢？其实可以主要分为 5 部分： 帧间和帧内预测（Estimation）、变换（Transform）和反变换、量化（Quantization）和反量化、环路滤波（Loop Filter）、熵编码（Entropy Coding）。

看起来很高深的样子，实际上也是很高深的样子，因为这里面包含着许许多多的算法和专业知识，这里我们就不做过多的讲解，有兴趣的同学可以上网翻翻，够你看到睡觉的了。H.264详细文档

原理简介

> H.264 原始码流(又称为裸流)，是有一个接一个的 NALU 组成的，而它的功能分为两层:视频编码层(VCL, Video Coding Layer)和网络提取层(NAL, Network Abstraction Layer)。
> VCL 数据即编码处理的输出，它表示被压缩编码后的视频数据 序列。在 VCL 数据传输或存储之前，这些编码的 VCL 数据，先被映射或封装进 NAL 单元(以下简称 NALU，Nal Unit) 中。每个 NALU 包括一个原始字节序列负荷(RBSP, Raw Byte Sequence Payload)、一组 对应于视频编码的 NALU 头部信息。RBSP 的基本结构是:在原始编码数据的后面填加了结尾 比特。一个 bit“1”若干比特“0”，以便字节对齐。

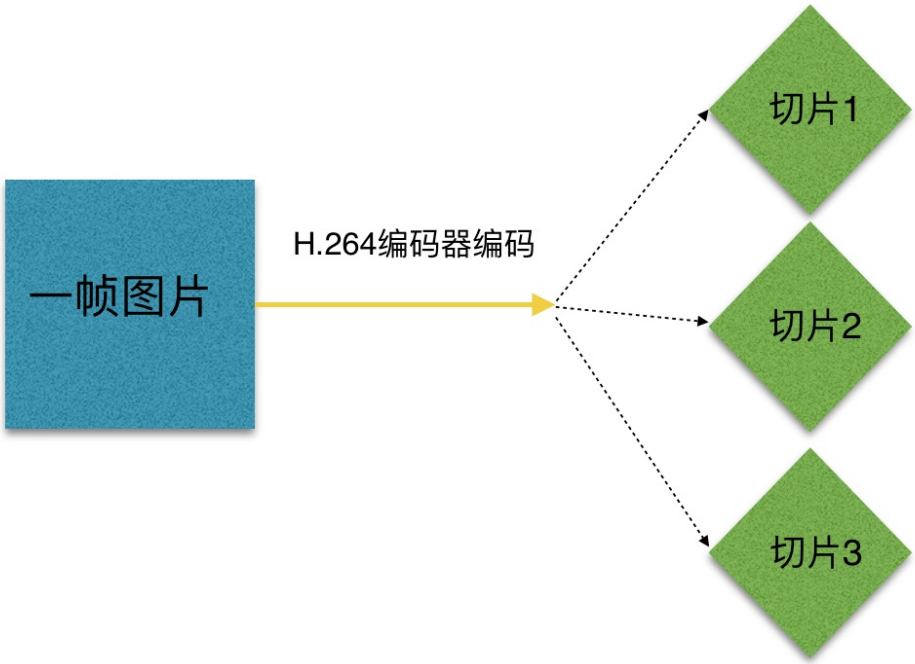
NAL 单元排列

上图中的 NALU头 + RBSP 就相当与一个 NALU (Nal Unit), 每个单元都按独立的 NALU 传送。其实说白了，H.264 中的结构全部都是以 NALU 为主的，理解了 NALU，就理解 H.264 的结构了。

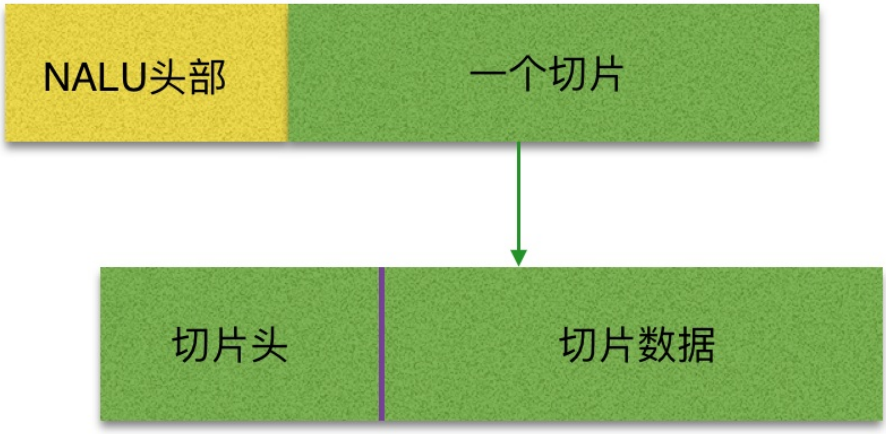
一帧图片跟 NALU 的关联 ：

> 究竟 NALU 是怎么由一帧图片变化而来的呀，H.264究竟为什么这么神奇？

一帧图片经过 H.264 编码器之后，就被编码为一个或多个片（slice），而装载着这些片（slice）的载体，就是 NALU 了，我们可以来看看 NALU 跟片的关系（slice）。



图片编码后



NALU 结构

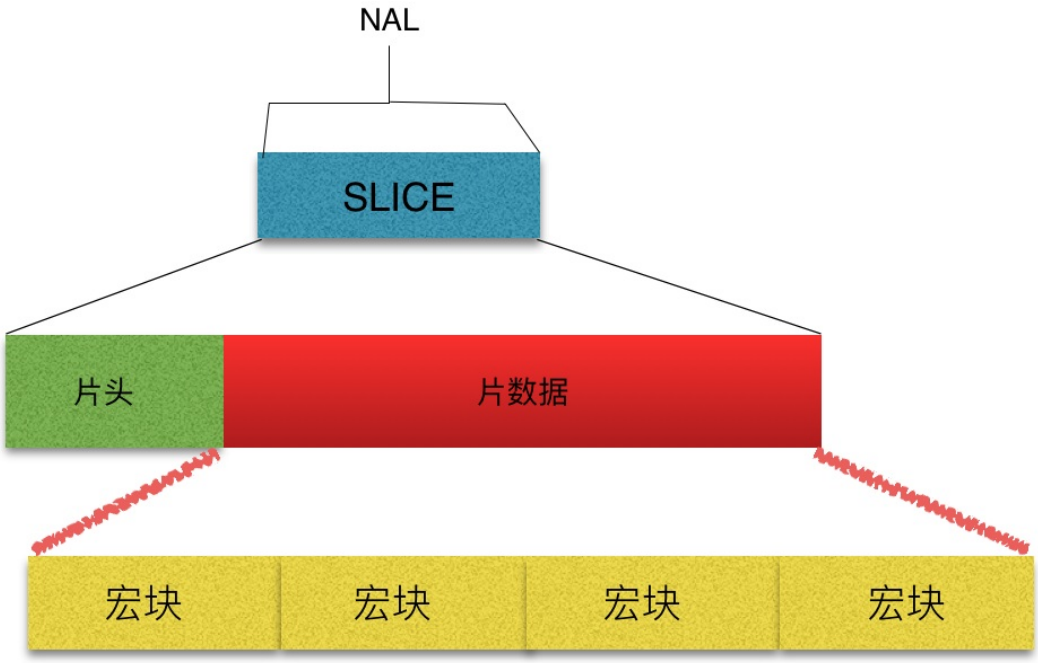
小伙伴们要明白，片（slice）的概念不同与帧（frame），帧（frame）是用作描述一张图片的，一帧（frame）对应一张图片，而片（slice），是 H.264 中提出的新概念，是通过编码图片后切分通过高效的方式整合出来的概念，一张图片至少有一个或多个片（slice）。

上图中可以看出，片（slice）都是又 NALU 装载并进行网络传输的，但是这并不代表 NALU 内就一定是切片，这是充分不必要条件，因为 NALU 还有可能装载着其他用作描述视频的信息。

什么是切片（slice）？

> 片的主要作用是用作宏块（Macroblock）的载体（ps：下面会介绍到宏块的概念）。片之所以被创造出来，主要目的是为限制误码的扩散和传输。
> 如何限制误码的扩散和传输？
> 每个片（slice）都应该是互相独立被传输的，某片的预测（片（slice）内预测和片（slice）间预测）不能以其它片中的宏块（Macroblock）为参考图像。

那么片（slice）的具体结构，我们用一张图来直观说明吧：



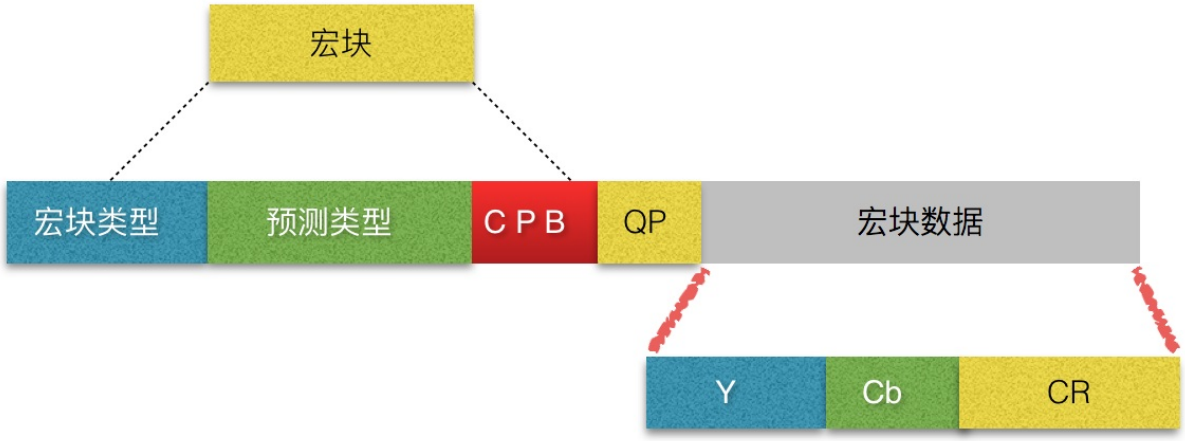
我们可以理解为一 张/帧 图片可以包含一个或多个分片(Slice)，而每一个分片(Slice)包含整数个宏块(Macroblock)，即每片（slice）至少一个 宏块(Macroblock)，最多时每片包 整个图像的宏块。

上图结构中，我们不难看出，每个分片也包含着头和数据两部分：
1、分片头中包含着分片类型、分片中的宏块类型、分片帧的数量、分片属于那个图像以及对应的帧的设置和参数等信息。
2、分片数据中则是宏块，这里就是我们要找的存储像素数据的地方。

什么是宏块？

> 宏块是视频信息的主要承载者，因为它包含着每一个像素的亮度和色度信息。视频解码最主要的工作则是提供高效的方式从码流中获得宏块中的像素阵列。
> 组成部分：一个宏块由一个16×16亮度像素和附加的一个8×8 Cb和一个 8×8 Cr 彩色像素块组成。每个图象中，若干宏块被排列成片的形式。

我们先来看看宏块的结构图：



从上图中，可以看到，宏块中包含了宏块类型、预测类型、Coded Block Pattern、Quantization Parameter、像素的亮度和色度数据集等等信息。

切片（slice）类型跟宏块类型的关系

对于切片（slice）来讲，分为以下几种类型：

> 0 P-slice. Consists of P-macroblocks (each macro block is predicted using one reference frame) and / or I-macroblocks.
> 1 B-slice. Consists of B-macroblocks (each macroblock is predicted using one or two reference frames) and / or I-macroblocks.
> 2 I-slice. Contains only I-macroblocks. Each macroblock is predicted from previously coded blocks of the same slice.
> 3 SP-slice. Consists of P and / or I-macroblocks and lets you switch between encoded streams.
> 4 SI-slice. It consists of a special type of SI-macroblocks and lets you switch between encoded streams.

I片：只包 I宏块，I 宏块利用从当前片中已解码的像素作为参考进行帧内预测(不能取其它片中的已解码像素作为参考进行帧内预测)。

P片：可包 P和I宏块，P 宏块利用前面已编码图象作为参考图象进行帧内预测，一个帧内编码的宏块可进一步作宏块的分割:即 16×16、16×8、8×16 或 8×8 亮度像素块(以及附带的彩色像素);如果选了 8×8 的子宏块，则可再分成各种子宏块的分割，其尺寸为 8×8、8×4、4×8 或 4×4 亮度像素块(以及附带的彩色像素)。

B片：可包 B和I宏块，B 宏块则利用双向的参考图象(当前和 来的已编码图象帧)进行帧内预测。

SP片(切换P)：用于不同编码流之间的切换，包含 P 和/或 I 宏块

SI片：扩展档次中必须具有的切换，它包 了一种特殊类型的编码宏块，叫做 SI 宏块，SI 也是扩展档次中的必备功能。

整体结构

通过剖析了这么多个小零件，是时候个大家一个世界地图了，
那么我们的 NALU 整体结构可以呼之欲出了，以下就引用 H.264 文档当中的一幅图了

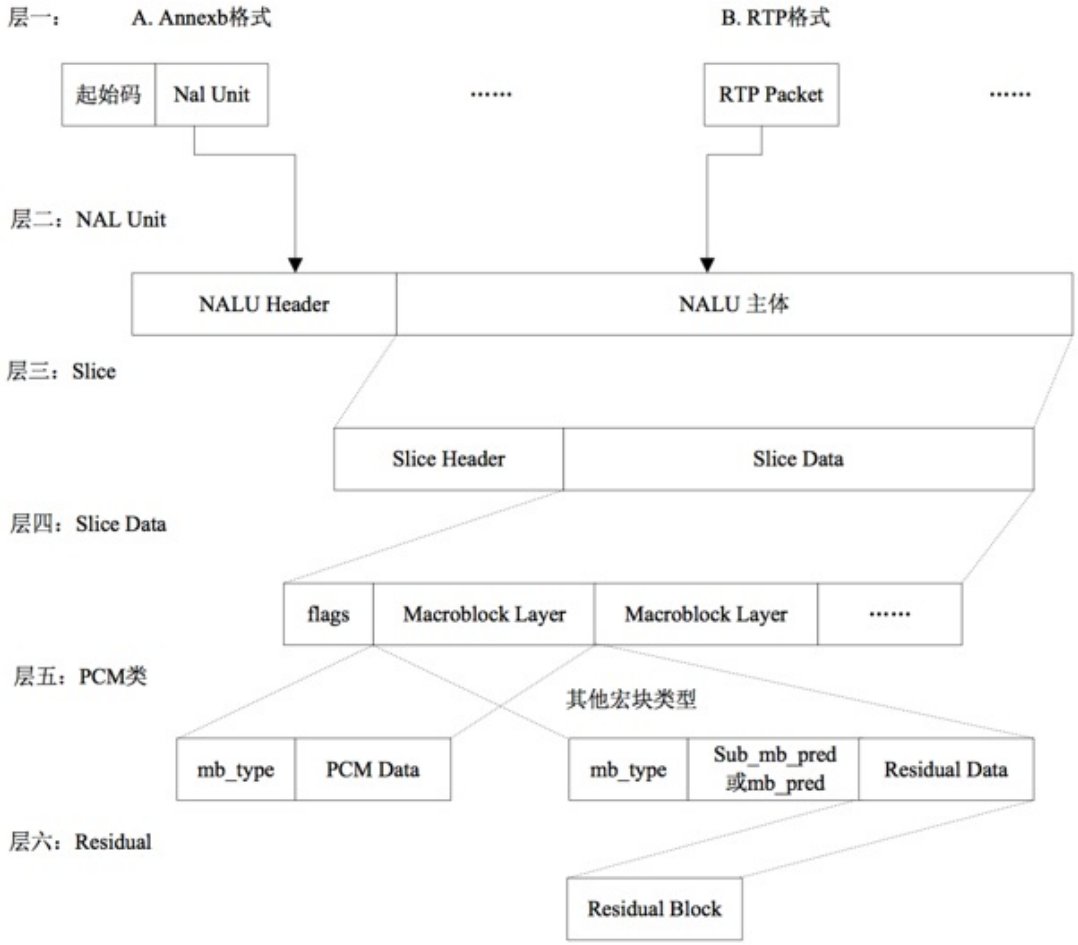
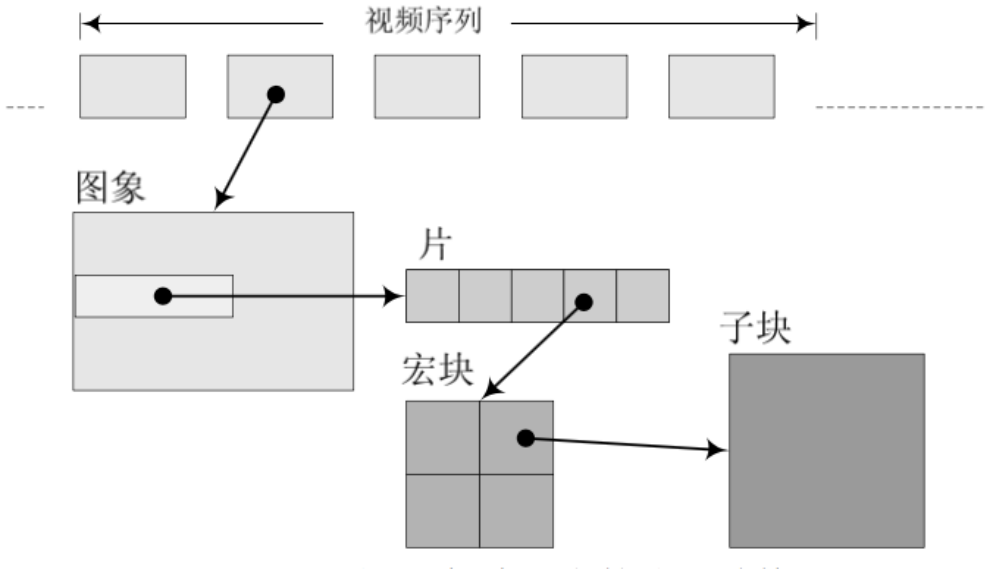


图 3 H.264 码流分层结构

其实 H.264 的码流结构并没有大家想的那么复杂，编码后视频的每一组图像（GOP，图像组）都给予了传输中的序列（PPS）和本身这个帧的图像参数（SPS），所以，我们的整体结构，应该如此：



> GOP （图像组）主要用作形容一个 i 帧 到下一个 i 帧之间的间隔了多少个帧，增大图片组能有效的减少编码后的视频体积，但是也会降低视频质量，至于怎么取舍，得看需求了。

主题外：（未完待续）

那么，NALU 头部中的类型确定着什么信息呢？
我们首先来看看 NALU 中究竟有哪几种类型，我们来看看 H.264 中源码对 nal_unit_type_e 中的定义：

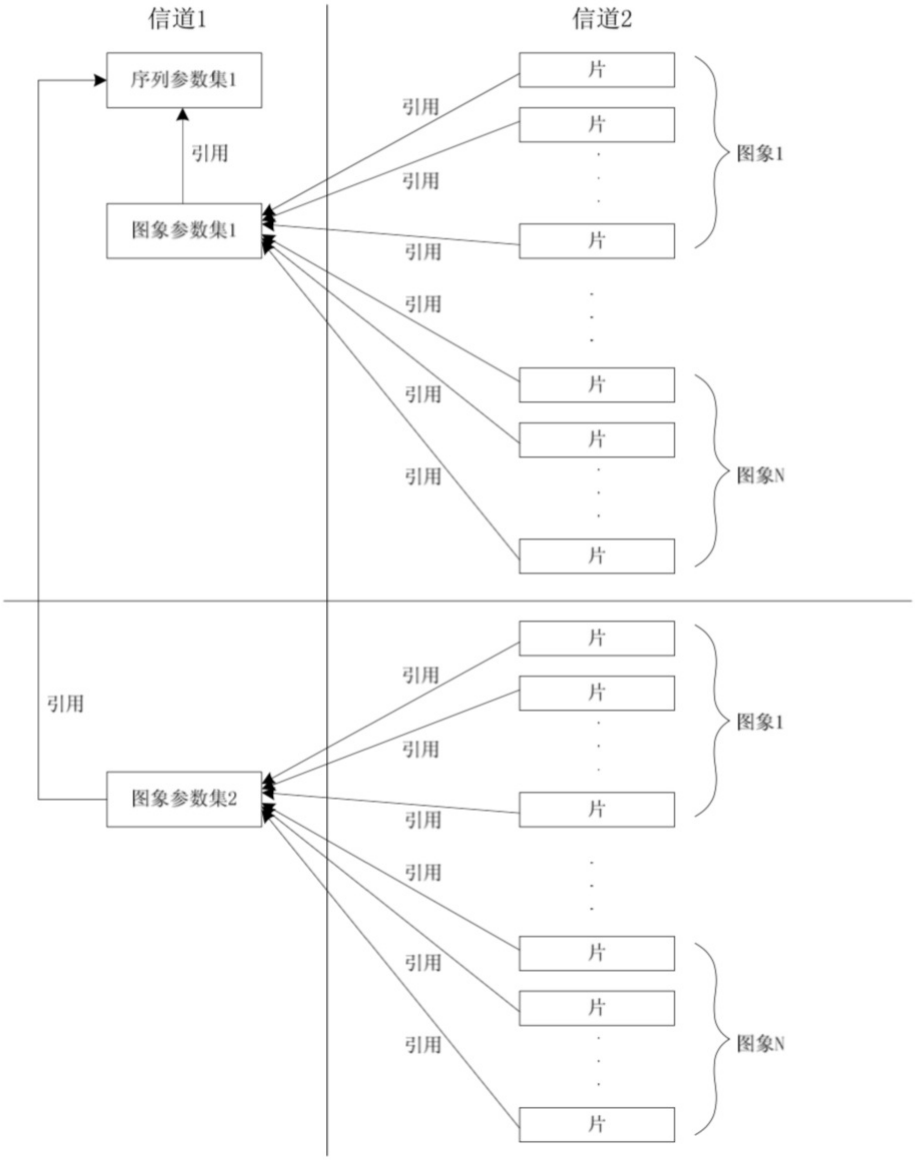
```
enum nal_unit_type_e
{
NAL_UNKNOWN = 0, // 未使用
NAL_SLICE = 1, // 不分区、非 IDR 图像的片（片的头信息和数据）
NAL_SLICE_DPA = 2, // 片分区 A
NAL_SLICE_DPB = 3, // 片分区 B
NAL_SLICE_DPC = 4, // 片分区 C
NAL_SLICE_IDR = 5, /* ref_idc != 0 / // IDR 图像中的片
NAL_SEI = 6, / ref_idc == 0 */ // 补充增强信息单元
```

参数集是 H.264 标准的一个新概念，是一种通过改进视频码流结构增强错误恢复能力的方法。
NAL_SPS = 7, // 序列参数集 （包括一个图像序列的所有信息，即两个 IDR 图像间的所有图像信息，如图像尺寸、视频格式等）
NAL_PPS = 8, // 图像参数集 （包括一个图像的所有分片的所有相关信息， 包括图像类型、序列号等，解码时某些序列号的丢失可用来检验信息包的丢失与否）

NAL_AUD = 9, // 分界符
NAL_FILLER = 12, // 填充（哑元数据，用于填充字节）
/* ref_idc == 0 for 6,9, 10 (表明下一图像为 IDR 图像）,11（表明该流中已没有图像）,12 */
};
ps: 以上括号（）中的为类型描述

上面NALU类型当中，分片/切片（slice）的概念我们都已经很清楚了，但是用 NALU 作载体的还有 SEI、SPS、PPS 等等。

今天我们不一一聚述这些类型对整个流程的作用了，我们挑出两个符合我们今天主题的类型来讲，PPS 和 SPS。



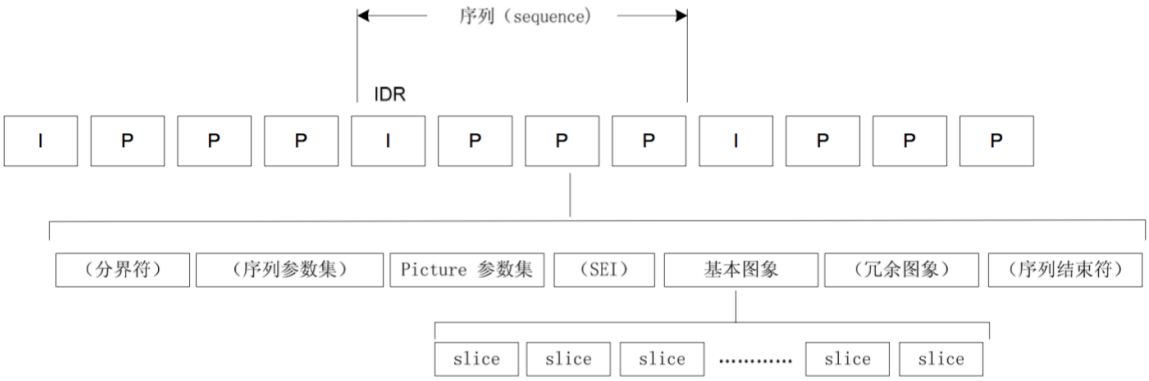


图 7.4　H.264 码流中的数据单位

那么今天我们讲的 H.264 的码流结构相信大家都有个大概轮廓的了解了，总结的一句话就是：

> H.264 中，句法元素共被组织成 序列、图像、片、宏块、子宏块五个层次。

希望大家用心体会，毕竟手工打字和作图不易，大家能关注的关注，能有闲钱的打赏一个，能有赞的赞一下嘛

最后福利：
文章的最后为了方便小伙伴学习，本博主用 ffmpeg 在 iOS 设备写了一个实例代码，主要功能是利用摄像头实时录制 yuv 格式文件并编码 H.264 后，实时分析 H.264 中每个帧（frame）中的 NALU 的数据，希望对小伙伴们有所帮助，找到入门的方法。
（本博主将所有代码只放在个人博客末尾当中：代码地址『所有代码只用做学习只用，并不提供商业用途，如果违反，必究。』）

心如止水，奋力前行

Abson在简书立志成为一名软件开发工程师！实践理论者！知识深入研究强迫症！ 代码仓库：https:/...
总资产23 (约1.70元)共写了4.3W字获得1,348个赞共1,927个粉丝