

[翻译]H.264 探索 第二部分 H264码流格式

h264

cppprimer 2016年08月24日发布

H.264 探索 第二部分 H264码流格式

[原文地址](#)

前言

[先前的话题](#)是关于色彩模型，以及如何存储像素色彩数据的。现在是时候去更深了解H.264码流格式，并找到这些像素的数据。



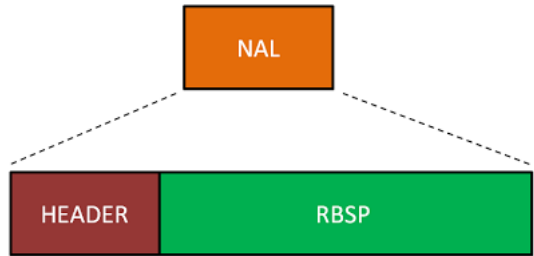
NAL

显然，解码器接收一个特定的格式的比特序列，并进行操作。二进制流是结构化的，由packets组成。在上层，流由NAL-packets组成，并具有以下形式：



NAL表示Network Abstraction Layer(网络抽象层)的缩写。

packet结构如下图



NAL-packet的第一个字节是一个包含关于packet的类型信息的报头。

表1中描述了所有可能的数据包类型。

Type	Definition
0	Undefined
1	Slice layer without partitioning non IDR

Type	Definition
3	Slice data partition B layer
4	Slice data partition C layer
5	Slice layer without partitioning IDR
6	Additional information (SEI)
7	Sequence parameter set
8	Picture parameter set
9	Access unit delimiter
10	End of sequence
11	End of stream
12	Filler data
13..23	Reserved
24..31	Undefined

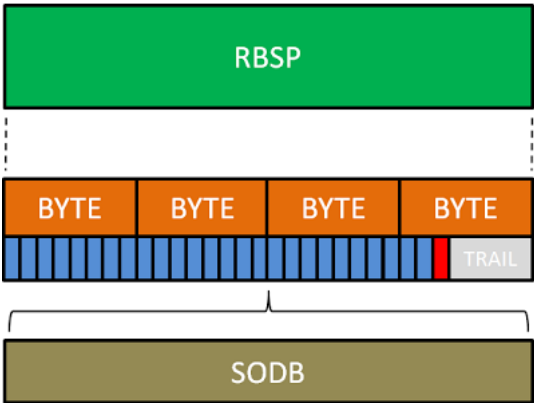
NAL类型定义了当前NAL-packet数据结构。它可以是slice，parameter set，filler等等。

如从图中可以看出，NAL-packet的有效载荷被定义为RBSP(Raw Byte Sequence Payload)。

RBSP描述了一行被定义为SODB(String Of Data Bits)的字节集。所以RBSP包含SODB。

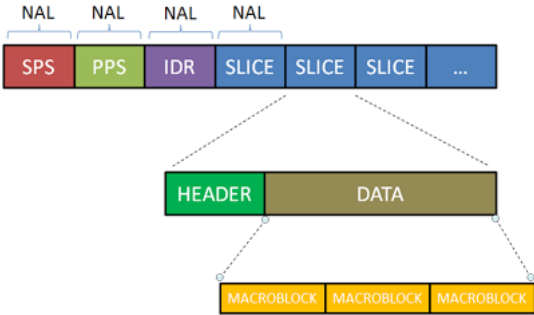
根据ITU-T规范，如果SODB为空（0比特长度），RBSP也为空。

RBSP的第一字节（最显著，最左边）包含八位SODB; RBSP的下一个字节应该也包括以下八个SODB等等，直到有少于8位SODB。接着是一个停止位和均衡位（如下图）



## SLICE

现在，让我们更进一步查看我们的比特流：

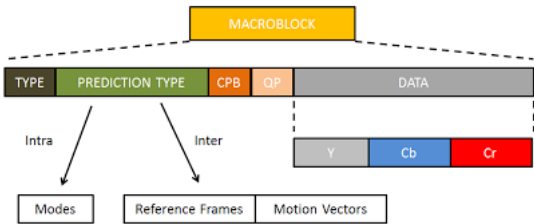


任何已编码图片都包含slice(切片)，slice包含macroblocks(宏块)。大多数情况下，一个已编码图像对应于一个切片。此外，一个图像可以有多个切片。所述切片被分为以下类型：

Type	Description
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.
5	P-slice.
6	B-slice.
7	I-slice.
8	SP-slice.
9	SI-slice.

看起来像上表包含了一些冗余数据，但事实并非如此：

类型5 - 9意味着当前图像的所有其他切片将是同一类型。正如你所看到的每个切片都由header和data组成。切片头包含了关于切片类型，切片宏块的类型，切片帧的数量的信息。当然切片头也包含了参照帧的设置（the reference frame settings）和量化参数（quantification parameters）的信息。最好，我们来查看切片data，这是储存像素数据的地方。宏块是信息的主要载体，因为它们包含将对应于单个像素的亮度和色度分量的集合。无需进入细节可以得出结论，不考虑细节可以得出的结论是:视频解码最终可以简化为从比特流中对宏块的搜索和提取，以及后续借助亮度和色度分量对像素颜色的恢复。宏块组成如下图：



在这里，我们有宏块类型，预测类型（这是下一文章的主题），编码块模式（Coded Block Pattern），量化参数（Quantization Parameter）（如果我们有CPB），最后数据（data）：亮度和色度分量的集合。

2016年08月24日发布 ...

赞 | 0

收藏 | 1

你可能感兴趣的文章

- [PHP+FFMPEG自动转码H264标准Mp4文件](#) 24 收藏, 5.6k 浏览
- [RTP打包h264码流](#) 1.6k 浏览
- [h264 hardware mediacodec+surfacetexture编码解码在某些mtk机型上会出现问题预警](#) 77 浏览

1 条评论

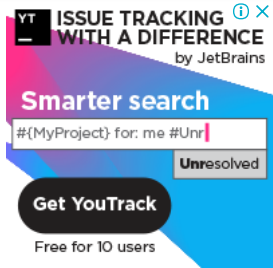
默认排序 | 时间排序

烟雨江南 · 2016年12月08日

不错

👍 赞    回复

文明社会，理性评论



cppprimer

166 声望

关注作者

发布于专栏

扯淡

扯淡

1 人关注

关注专栏

系列文章

[翻译]H.264 探索 第一部分 色彩模型 488 浏览