语法及结构

# 一、名词解释

场和帧：视频的一场或一帧可用来产生一个编码图像。在电视中，为减少大面积闪烁现象，把一帧分成两个隔行的场。

片：每个图像中，若干宏块被排列成片的形式。片分为I片、B片、P片和其他一些片。

I片只包含I宏块，P片可包含P和I宏块，而B片可包含B和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。

P宏块利用前面已编码图像作为参考图像进行帧内预测。

B宏块则利用双向的参考图像（前一帧和后一帧）进行帧内预测。

片的目的是为了限制误码的扩散和传输，使编码片相互间是独立的。

某片的预测不能以其它片中的宏块为参考图像，这样某一片中的预测误差才不会传播到其它片中去。

宏块：一个编码图像通常划分成若干宏块组成，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。

# 二、数据之间的关系

H.264结构中，一个视频图像编码后的数据叫做一帧，一帧由一个片（slice）或多个片组成，一个片由一个或多个宏块（MB）组成，一个宏块由16x16的yuv数据组成，宏块作为H.264编码的基本单位。

# 三、H264编码过程中的三种不同的数据形式

1、SODB（数据比特串）：最原始的编码数据，即VCL数据；

2、RBSP（原始字节序列载荷）：在SODB的后面填加了结尾比特（RBSP trailing bits　一个bit“1”）若干比特“0”,以便字节对齐；

3、EBSP（扩展字节序列载荷）：在RBSP基础上填加了仿校验字节（0x03），它的原因是：在NALU加到Annexb上时，需要添加每组NALU之前的开始码StartCodePrefix，如果该NALU对应的slice为一帧的开始则用4个字节表示，0x00000001,否则用3个字节表示0x000001（是一帧的一部分）。另外，为了使NALU主体中不包括与开始码相冲突的，在编码时，每遇到两个字节连续为0，就插入一个字节的0x03。解码时将0x03去掉，也称为脱壳操作。

# 四、H264/AVC 的分层结构

1、H.264的主要目标是：

1）高的视频压缩比；

2）良好的网络亲和性；

2、为了完成这些目标H264的解决方案是：

1）VCL（video coding layer）：视频编码层；

2）NAL（network abstraction layer）：网络抽象层；

其中，VCL层是对核心算法引擎，块，宏块及片的语法级别的定义，它最终输出编码完的数据 SODB；NAL层定义片级以上的语法级别（如序列参数集和图像参数集，针对网络传输），同时支持以下功能：独立片解码，起始码唯一保证，SEI以及流格式编码数据传送，NAL层将SODB打包成RBSP然后加上NAL头，组成一个NALU（NAL单元）；

3、H264网络传输的结构

H264在网络传输的是NALU，NALU的结构是：NAL头+RBSP，实际传输中的数据流如图所示：



NALU头用来标识后面的RBSP是什么类型的数据，他是否会被其它帧参考以及网络传输是否有错误。

4、NALU头结构

长度：1byte

forbidden\_bit(1bit) + nal\_reference\_bit(2bit) + nal\_unit\_type(5bit)

1）forbidden\_bit：禁止位，初始为0，当网络发现NAL单元有比特错误时可设置该比特为1，以便接收方纠错或丢掉该单元。

2）nal\_reference\_bit：nal重要性指示，标志该NAL单元的重要性，值越大，越重要，解码器在解码处理不过来的时候，可以丢掉重要性为0的NALU。

不同类型的NALU的重要性指示如下表所示：

|  |  |  |
| --- | --- | --- |
| nal\_unit\_type | NAL类型 | nal\_reference\_bit |
| 0 | 未使用 | 0 |
| 1 | 非IDR的片 | 此片属于参考帧，则不等于0，  不属于参考帧，则等与0 |
| 2 | 片数据A分区 | 同上 |
| 3 | 片数据B分区 | 同上 |
| 4 | 片数据C分区 | 同上 |
| 5 | IDR图像的片 | 5 |
| 6 | 补充增强信息单元（SEI） | 0 |
| 7 | 序列参数集 | 非0 |
| 8 | 图像参数集 | 非0 |
| 9 | 分界符 | 0 |
| 10 | 序列结束 | 0 |
| 11 | 码流结束 | 0 |
| 12 | 填充 | 0 |
| 13..23 | 保留 | 0 |
| 24...31 | 不保留 | 0 |

  所谓参考帧，就是在其他帧解码时需要参照的帧。比如一个I帧可能被一个或多个p帧参考，一个B帧可能被某个P帧参考。

从这个表我们也可以看出来，DIR的I帧是非常重要的，它一丢，那么这个序列的所有帧都没办法解码了；

序列参数集和图像参数集也很重要，没有序列参数集，这个序列的帧就没法解；没有图像参数集，那用到这个图像参数集的帧都没法解。

3）nal\_unit\_type：NALU类型取值如下表所示：

|  |  |  |
| --- | --- | --- |
| nal\_unit\_type | NAL类型 | C |
| 0 | 未使用 |  |
| 1 | 非IDR图像中不采用数据划分的片段 | 2,3,4 |
| 2 | 非IDR图像中A类数据划分片段 | 2 |
| 3 | 非IDR图像中B类数据划分片段 | 3 |
| 4 | 非IDR图像中C类数据划分片段 | 4 |
| 5 | IDR图像的片 | 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 | 不保留（RTP打包时会用到） |  |

RTP 打包时的扩展类型：

|  |  |  |
| --- | --- | --- |
| 24 | STAP-A | Single-time aggregation packet |
| 25 | STAP-B | Single-time aggregation packet |
| 26 | MTAP16 | Multi-time aggregation packet |
| 27 | MTAP24 | Multi-time aggregation packet |
| 28 | FU-A | Fragmentation unit |
| 29 | FU-B | Fragmentation unit |
| 30-31 | undefined |  |

5、RBSP

RBSP数据是下表中的一种：

|  |  |  |
| --- | --- | --- |
| RBSP类型 | 所写 | 描述 |
| 参数集 | PS | 序列的全局信息，如图像尺寸，视频格式等 |
| 增强信息 | SEI | 视频序列解码的增强信息 |
| 图像界定符 | PD | 视频图像的边界 |
| 编码片 | SLICE | 编码片的头信息和数据 |
| 数据分割 |  | DP片层的数据，用于错误恢复解码 |
| 序列结束符 |  | 表明一个序列的结束，下一个图像为IDR图像 |
| 流结束符 |  | 表明该流中已没有图像 |
| 填充数据 |  | 亚元数据，用于填充字节 |

从前面的分析我们知道，VCL层出来的是编码完的视频帧数据，这些帧可能是I、B、P帧，而且这些帧可能属于不同的序列，再者同一个序列还有相对应的一套序列参数集和图片参数集等等，所以要完成视频的解码，不仅需要传输VCL层编码出来的视频帧数据，还需要传输序列参数集、图像参数集等数据。

参数集:包括序列参数集 SPS和图像参数集 PPS

SPS包含的是针对一连续编码视频序列的参数，如标识符 seq\_parameter\_set\_id、帧数及 POC 的约束、参考帧数目、解码图像尺寸和帧场编码模式选择标识等等。

PPS对应的是一个序列中某一幅图像或者某几幅图像，其参数如标识符 pic\_parameter\_set\_id、可选的 seq\_parameter\_set\_id、熵编码模式选择标识、片组数目、初始量化参数和去方块滤波系数调整标识等等。

数据分割：组成片的编码数据存放在 3 个独立的 DP（数据分割，A、B、C）中，各自包含一个编码片的子集。分割Ａ包含片头和片中每个宏块头数据；分割Ｂ包含帧内和 SI 片宏块的编码残差数据；分割 C包含帧间宏块的编码残差数据。每个分割可放在独立的 NAL 单元并独立传输

6、NAL的开始和结束

编码器将每个NAL各自独立、完整地放入一个分组，因为分组都有头部，解码器可以方便地检测出NAL的分界，并依次取出NAL进行解码。

每个NAL前有一个起始码 0x000001（或者0x00000001），解码器检测每个起始码，作为一个NAL的起始标识，当检测到下一个起始码时，当前NAL结束。

同时H.264规定，当检测到0x000000时，也可以表征当前NAL的结束。那么NAL中数据出现0x000001或0x000000时怎么办？H.264引入了防止竞争机制，如果编码器检测到NAL数据存在0x000001或0x000000时，编码器会在最后个字节前插入一个新的字节0x03，这样：

0x000000－>0x00000300

0x000001－>0x00000301

0x000002－>0x00000302

0x000003－>0x00000303

解码器检测到0x000003时，把03抛弃，恢复原始数据（脱壳操作）。解码器在解码时，首先逐个字节读取NAL的数据，统计NAL的长度，然后再开始解码。

7、NALU的顺序要求

H.264/AVC标准对送到解码器的NAL单元顺序是有严格要求的，如果NAL单元的顺序是混乱的，必须将其重新依照规范组织后送入解码器，否则解码器不能够正确解码。

1）序列参数集NAL单元

必须在传送所有以此参数集为参考的其他NAL单元之前传送，不过允许这些NAL单元中间出现重复的序列参数集NAL单元。

所谓重复的详细解释为：序列参数集NAL单元都有其专门的标识，如果两个序列参数集NAL单元的标识相同，就可以认为后一个只不过是前一个的拷贝，而非新的序列参数集。

2）图像参数集NAL单元

必须在所有以此参数集为参考的其他NAL单元之前传送，不过允许这些NAL单元中间出现重复的图像参数集NAL单元，这一点与上述的序列参数集NAL单元是相同的。

3）不同基本编码图像中的片段（slice）单元和数据划分片段（data partition）单元在顺序上不可以相互交叉，即不允许属于某一基本编码图像的一系列片段（slice）单元和数据划分片段（data partition）单元中忽然出现另一个基本编码图像的片段（slice）单元片段和数据划分片段（data partition）单元。

4）参考图像的影响：如果一幅图像以另一幅图像为参考，则属于前者的所有片段（slice）单元和数据划分片段（data partition）单元必须在属于后者的片段和数据划分片段之后，无论是基本编码图像还是冗余编码图像都必须遵守这个规则。

5）基本编码图像的所有片段（slice）单元和数据划分片段（data partition）单元必须在属于相应冗余编码图像的片段（slice）单元和数据划分片段（data partition）单元之前。

6）如果数据流中出现了连续的无参考基本编码图像，则图像序号小的在前面

7）如果arbitrary\_slice\_order\_allowed\_flag置为1，一个基本编码图像中的片段（slice）单元和数据划分片段（data partition）单元的顺序是任意的，如果arbitrary\_slice\_order\_allowed\_flag置为零，则要按照片段中第一个宏块的位置来确定片段的顺序，若使用数据划分，则A类数据划分片段在B类数据划分片段之前，B类数据划分片段在C类数据划分片段之前，而且对应不同片段的数据划分片段不能相互交叉，也不能与没有数据划分的片段相互交叉。

8）如果存在SEI（补充增强信息）单元的话，它必须在它所对应的基本编码图像的片段（slice）单元和数据划分片段（data partition）单元之前，并同时必须紧接在上一个基本编码图像的所有片段（slice）单元和数据划分片段（data partition）单元后边。假如SEI属于多个基本编码图像，其顺序仅以第一个基本编码图像为参照。

9）如果存在图像分割符的话，它必须在所有SEI 单元、基本编码图像的所有片段slice）单元和数据划分片段（data partition）单元之前，并且紧接着上一个基本编码图像那些NAL单元。

10）如果存在序列结束符，且序列结束符后还有图像，则该图像必须是IDR（即时解码器刷新）图像。序列结束符的位置应当在属于这个IDR图像的分割符、SEI 单元等数据之前，且紧接着前面那些图像的NAL单元。如果序列结束符后没有图像了，那么它的就在比特流中所有图像数据之后。

11）流结束符在比特流中的最后。