YUV

# 1.概述

与RGB类似，YUV也是图像的存储方式。

YUV一般还有几种表述方式：YCbCr，YPbPr；

亮度信号被称作Y；色度信号由两个相互独立的信号组成，一般称作UV，也有称作CbCr或者PbPr；

这些不同的称谓，是由不同的编码方式产生的，但他们的概念基本相同。

YUV格式有两大类：packed和planar。

Planar的YUV格式，是按行存储的，先连续存储Y，紧接着存储所有U，最后存储所有V；packed的YUV格式，Y、U、V是连续存储的。

YUV码流的存储方式，主流的有4中：YUV444, 422, 420, 411。概述如下：

* YUV444采样，每一个Y共用一组UV分量；
* YUV422采样，每两个Y共用一组UV分量；
* YUV420采样，每四个Y共用一组UV分量；

按照我现在的理解，一般说的YUV444等代表的是packed格式，YUV444P代表的是planar格式。Yuv422和420同理。

Yuv420除了packed和planar，还有semiPlanar，一般简写为yuv420sp，详见第二章。

# 2.各个不同格式

## 2.1.YUV444

YUV444，一个Y对应一组UV分量，其格式可概述为：

* Packed格式(YUV444)，为YUVYUVYUV........；
* Planar模式(YUV444P)，为YYY....UUU....VVV...；

Y、U、V的数量是相同的，如果分辨率为w\*h，那么yuv文件的size就是：w\*h\*3;

## 2.2.YUV422

YUV422有几种不同的格式划分方式：YUYV，UYVY，YUV422P等；

* YUYV是yuv422--packed的一种，其格式为：YUYVYUYVYUYV.......；
* UYVY是yuv422--packed的一种，其格式为：UYVYUYVYUYVY.......；
* Yuv422p是planar模式，其格式为：YYYY....UU....VV....；

Y、U、V数量不同，两个Y对应一组UV，yuv文件的size就是：w\*h\*2；

## 2.3.YUV420

YUV420分为YUV420P和YUV420SP两种：

* YUV420p，其格式为:YYYY....UU..VV..;
* YUV420sp，其格式为:YYYY....UVUV.....;

Yuv文件的size，是：w\*h\*3/2；

YUV420是使用最为广泛的一种yuv格式，其精细划分也最多，概述如下：

* I420：即标准意义上的YUV420P，存储方式为：YYYY....UU..VV..；
* YU12：与I420一致，是安卓的存储方式；
* YV12：属于YUV420P的一种，存储格式为：YYYY....VV..UU..;
* NV12：属于yuv420sp的一种，存储格式为：yyyy....uvuv....，简单说，就是先存放Y，之后UV交替存储，U在前；
* NV21：属于yuv420sp的一种，存储格式为：yyyy....vuvu....，与NV12不同的仅仅是U和V的顺序不同，V在前；

## 2.4.YUV411

TODO

# 3.转换规则

## 3.1.packed和planar的转换

同种格式的packed和planar模式的转换更为简单一些，以yuv444为例说明：

yuv444packed模式下，yuvyuv.....的格式是其文件格式；yuv444planar模式下，yyyy....uuuu....vvvv....是其文件格式。因此所谓的转换，只是将各个位置的y、u、v分量，重新排列。

## 3.2.YUV444、YUV422、YUV420的转换

H264

# 1.VCL NAL

H264协议分为两层：VCL, NAL；

VCL，Video coding layer，视频编码层；NAL，Network abstraction layer，网络抽象层；

VCL数据表示编码处理的输出，表示被压缩后的视频数据序列。NAL单元用来格式化数据并提供头信息，保证数据适合各种信道和存储介质上的传输。

# 2.SODB RBSP EBSP

SODB：string of data bits，最原始的编码数据，无任何附加内容；

RBSP：在SODB基础上增加了rbsp\_stop\_ont\_bit(bit值为1)，并在其后用0按字节进行了补齐；

EBSP：Encapsulation byte sequence packets，在RBSP基础上，增加防止伪起始码字节0x03；

EBSP存在的原因，是对于H264来说，一般会应用0x00000001/0x000001作为一个有效的NALU的起始码，如果这时候存在原始的编码数据，其中有0x0000的数据部分，就会使用0x03作一次转换，防止与起始码冲突。例如：

0x000000 -> 0x00000300

0x000001 -> 0x00000301

等到解码的时候，会将0x03删除掉，称之为“脱壳”。

# 3.NALU

NAL单元，简称NALU，其构成有两部分：NAL header和RBSP等数据部分；

NAL头部占用1个字节，构成为：

|  |  |  |
| --- | --- | --- |
| Forbiden | 1bit | H264规定这一位必须是0，forbidden\_zero\_bit |
| NalPriority | 2bit | nal\_ref\_idc，0-3，表示这个nalu的优先级 |
| Type | 5bit | nalu的类型 |

对于NALU的type，H264规范定义如下：

|  |  |  |
| --- | --- | --- |
| TypeValue | 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，I帧** | **2，3** |
| **6** | **补充增强信息单元，SEI** | **5** |
| **7** | **序列参数集，SPS** | **0** |
| **8** | **图像参数集，PPS** | **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 |  |

如上的NALU的type规范，是我们解析H264的重要依据；

# 4.起始码

H264规范定义了起始码(startcode)的概念，用来判断是否是一个新的NALU的开始。

如果NALU对应的slice是一帧的开始，则用0x00000001作为起始码；否则用0x000001作为起始码；

# 5.SPS PPS SEI

SPS和PPS是初始化解码器必须要有的数据，没有这两部分数据，视频数据无法解析并播放。

H264文件，有一些是每个IDR前面都存在相应的SPS和PPS，有一些是只有在文件开头才有，文件中就没有了。一般来说:

如果是直播的话，每个IDR前面都要加上sps和pps，因为有的观众会中途进来观看；

如果是本地稳定文件，开头有sps和pps就足够了，也可以每个IDR前面都加入sps和pps，要看具体需求了。

SEI作为辅助增强信息，supplemental enhancement infomation，并非总是存在。

# 6.IDR和I帧

IDR，instantaneous decoding refresh，即时解码刷新；

I帧和IDR帧都是帧内预测的。

IDR的作用是立刻刷新，从IDR开始重新计算一个新的序列开始编码，会导致DPB(decoding picture buffer，参考帧列表)被清空；

I帧不具备这个强制清空DPB的能力。

IDR一定是I帧，但I帧不一定是IDR帧。

IDR帧之后的所有帧，都不能够引用IDR帧之前的帧的内容；但普通I帧则不同，气候的B帧和P帧是可以参考普通I帧之前的I帧的。

# 7.解析流程