图像与视频编码基础知识

上一篇文章是对视频封装和协议的大致介绍，并没有对某些细节进行详细阐述，上一篇文章提到视频的本质就是一些静止的图片，当这些图片在单位时间内的数量比较大时，人眼将会看到其是连续动作。由此看来，学习音视频编解码前咱们还得先了解一下图像的基础知识。

> 这一节的内容既是编解码基础，又是开发中经常会接触的知识。

一、图像基础知识

在现实生活中，人们对一张图片的分辨都是主观评价，就是用人的主观感知直接测量。说人话就是——”好不好看我说了算“。但是在电子设备的世界里，并没有这种主观感知。而是通过像素、分辨率、PPI、色彩和一系列算法等侧面体现。

1.1 何为像素

图像，是由很多“带有颜色的点”组成的，你放大后会发现很多小方块（常规是正方形，还有其他形状），每个点就是一个“**像素点**”。

像素点的英文叫 **Pixel** （缩写为PX），这个单词是由 *Picture*（图像）和 *Element*（元素） 这两个单词字母所组成，像素是图像的显示的 **基本单位** 。通常我们说一幅图片的分辨率大小是1920 * 1080，意思就是长度为 1920 个像素点，宽带为 1080 个像素点。乘积是 1920 * 1080 = 2,073,600，也就是说，这个图片是 200万 像素。

1.2 何为 PPI

思考这样一个问题：我们平常使用的手机和平板，是不是屏幕越大就越清晰呢？当然不是，其实清晰度和PPI密切相关。

PPI，就是“Pixels Per Inch”，即 **每英寸像素** 。也就是手机（或显示器）屏幕上每英寸面积到底能存放多少个像素点。理论上屏幕 PPI 越高，屏幕就越精细，画质相对就更出色。

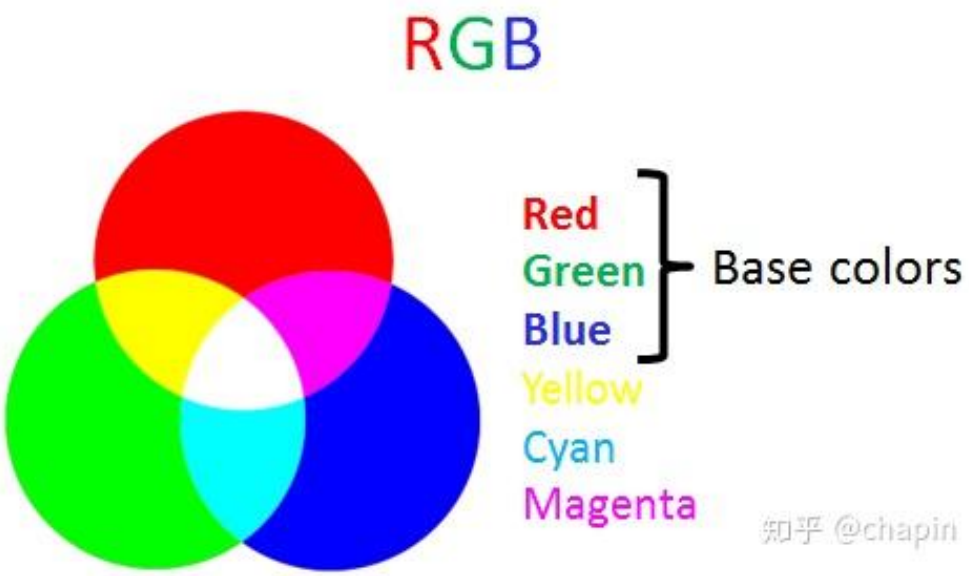
平时咱们在选择手机等电子设备时，可以将PPI考虑进去，应该尽量选择 PPI 高一点的，但是 PPI 过高也会造成费电，成本相对高昂等缺点。个人建议保证 400+ 的 PPI 是最佳的。

1.3 电子设备颜色表示法

像素点必须要有颜色，才能组成缤纷绚丽的图片。那么，这个颜色，又该如何表示呢?

在电子设备系统里，我们不可能用文字来表述颜色。不然，就算我们不疯，电子设备也会疯掉的。在数字时代，当然是用数字来表述颜色。这就牵出了“**彩色分量数字化**”的概念。

以前我们美术课学过，任何颜色，都可以通过红色（Red）、绿色（Green）、蓝色（Blue）按照一定比例调制出来。这三种颜色，被称为“**三原色**”。



在电子设备的世界里，R、G、B 也被称为“**基色分量**”。它们的取值分别从 0 到 255，一共256个等级（256时2的8次方）。所以任何颜色都可以 R、G、B 三个值的组合来表示。

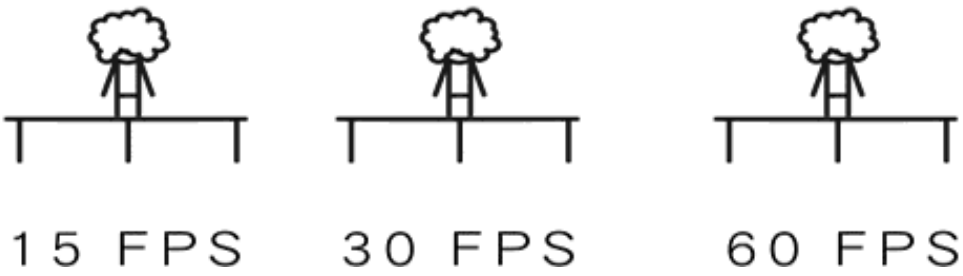
通过这种方式，一共能表达多少种颜色呢？256 * 256 * 256 = 16,777,216 种，因此也简称为 **1600 万色** 。RGB三色，每色有8bit，这种方式表达出来的颜色，也被称为 **24位色** （占用24bit）。这个颜色应超过了人眼可见的全部色彩。再高的话，对于我们人眼来说，已经没有意义了，完全识别不出来。

二、视频编码基础知识

2.1 视频和图像和关系

刚才说了图像，现在，我们开始说视频。所谓视频，大家从小就看动画，都知道视频是怎么来的吧？没错，大量的图片连续起来，就是视频。

衡量视频，又是用的什么指标参数呢？最主要的一个，就是帧率（Frame Rate）。在视频中，一个帧（Frame）就是指一幅静止的画面。帧率，就是指视频每秒钟包括的画面数量（FPS，Frame per second）。帧率越高，视频就越逼真、越流畅。



2.2 未经编码的视频数据量会有多大?

正因为如此，屌丝工程师们就提出了，必须对视频进行编码。

有了视频之后，就涉及到两个问题：

- 一个是存储；
- 二个是传输。

而之所以会有视频编码，关键就在于此：一个视频，如果未经编码，它的体积是非常庞大的。

以一个分辨率1920×1280，帧率30的视频为例：

> 共：1920×1280=2,073,600（Pixels 像素），每个像素点是24bit（前面算过的哦）； 也就是：每幅图片2073600×24=49766400 bit，8 bit（位）= 1 byte（字节）； 所以：49766400bit=6220800byte≈6.22MB。

这是一幅1920×1280图片的原始大小，再乘以帧率30。

也就是说：每秒视频的大小是 186.6 MB，每分钟大约是 11GB，一部 90 分钟的电影，约是 1000 GB。。。

吓尿了吧？就算你现在电脑硬盘是 4TB 的（实际也就 3600GB），也放不下几部大姐姐啊！不仅要存储，还要传输，不然视频从哪来呢？如果按照 100M 的网速（12.5MB/s），下刚才那部电影，需要 22 个小时。。。再次崩溃。。。

正因为如此，屌丝工程师们就提出了，必须对视频进行编码。

2.3 什么是编码

编码： 就是按指定的方法，将信息从一种形式（格式），转换成另一种形式（格式）。

视频编码： 就是将一种视频格式，转换成另一种视频格式。编码的终极目的，说白了，就是为了压缩。各种五花八门的视频编码方式，都是为了让视频变得体积更小，有利于存储和传输。

首先是视频采集。通常我们会使用摄像机、摄像头进行视频采集。

采集了视频数据之后，就要进行模数转换，将模拟信号变成数字信号。其实现在很多都是摄像机（摄像头）直接输出数字信号。信号输出之后，还要进行预处理，将 RGB 信号变成 YUV 信号。

2.4 什么是 YUV 信号

前面我们介绍了 RGB 信号，那什么是 YUV信号 呢?

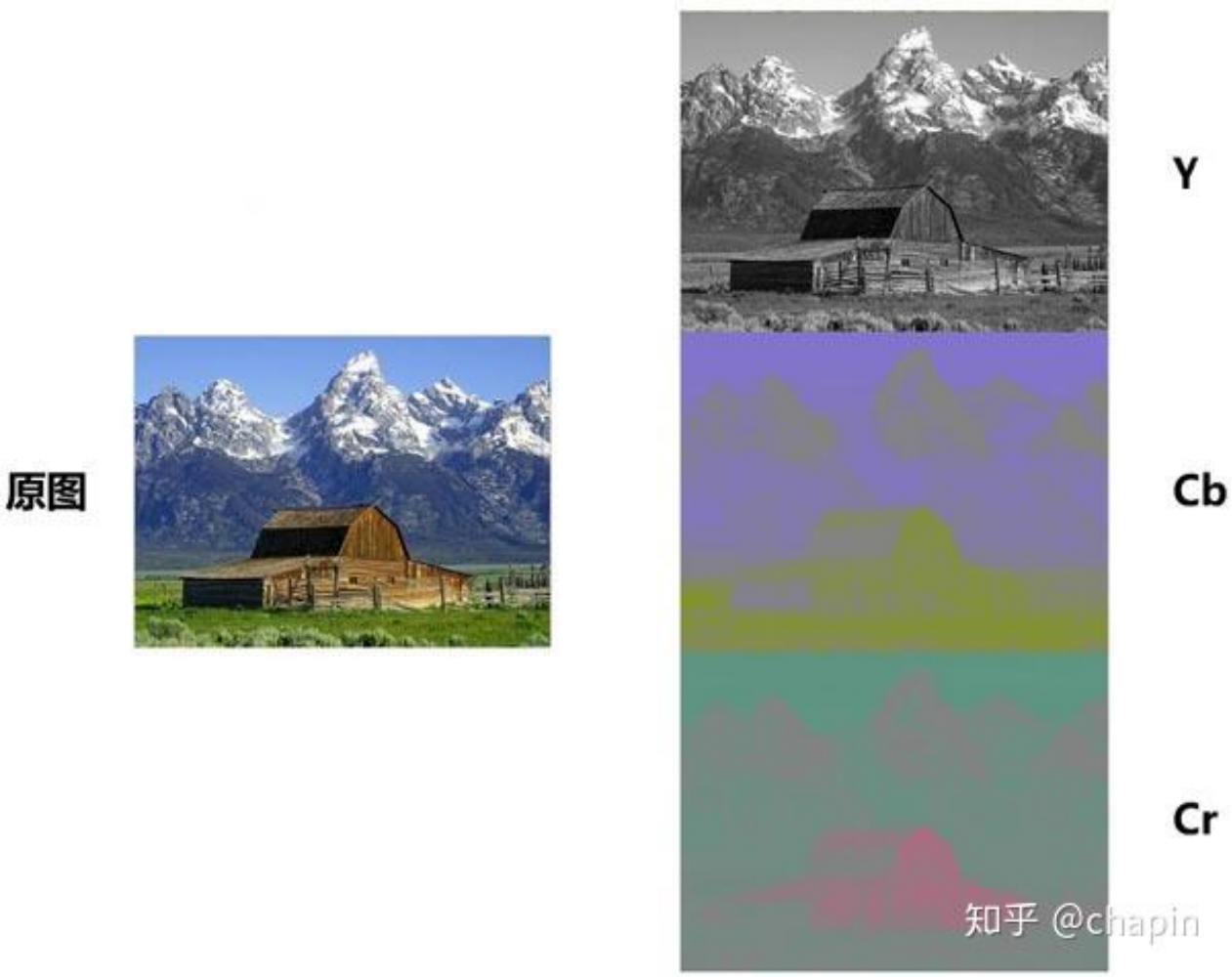
简单来说，YUV就是另外一种颜色数字化表示方式。视频通信系统之所以要采用YUV，而不是RGB，主要是因为**RGB信号不利于压缩** 。在YUV这种方式里面，加入了亮度这一概念。在最近十年中，视频工程师发现，眼睛对于亮和暗的分辨要比对颜色的分辨更精细一些，也就是说，人眼对色度的敏感程度要低于对亮度的敏感程度。

所以，工程师认为，在我们的视频存储中，没有必要存储全部颜色信号。我们可以把更多带宽留给黑—白信号（被称作“亮度”），将稍少的带宽留给彩色信号（被称作“色度”）。于是，就有了YUV。

YUV 三个字母中，其中 "Y" 表示明亮度（Lumina nce 或 Luma），也就是灰阶值；而 "U" 和 "V" 表示的则是色度（Chrominance 或 Chroma），作用是描述影像色彩及饱和度，用于指定像素的颜色。

在编码文档里，YUV 经常有另外的名字 YCbCr。 YCbCr 模型来源于 YUV 模型，算是 YUV 的压缩版本，不同之处在于 YCbCr 用于数字图像领域，YUV 用于模拟信号领域，MPEG、DVD、摄像机中常说的 YUV 其实就是 YCbCr。

其中 Y 与 YUV 中的 Y 含义一致，Cb , Cr 同样都指色彩，只是在表示方法上不同而已，Cb、Cr 就是本来理论上的“分量/色差”的标识。C 代表分量(是 component 的缩写)Cr、Cb，分别对应r(红)、b(蓝)分量信号，Y 除了 g(绿) 分量信号，还叠加了亮度信号。



2.5 YUV的存储方式

YUV有packed（打包）和 planar（平面）两种存储方式。

- packed： packed格式是先连续储存所有的Y分量，然后一次交叉存储U、V分量；

- planar：planar格式也会先连续存储所有的Y分量，但planar会先连续储存U分量的数据，再连续存储V分量的数据；或者先连续存储V分量的数据，再连续存储U分量的数据。

在一个 YUV 图像中，每个像素点有3个 8 bit 的值，分别对应 Y、U、V 3个分量。人眼对亮度信息 Y 更敏感，因此对色度分量 U、V 可以采用 **采样** ，但是仍然能保持很好的可视图像质量。采样可以减少传输带宽，可以用 X:X:X 格式表示，其中第一数字表示亮度采样的数目，用作参考，往往设定为“4”，第二和第三个数字表示色度采样的数目，与Y的数目有关。

例如，4:1:1表示每4个 Y 抽样点有一个 U 和 V 抽样点。每个点保存一个 8bit 的亮度值(也就是Y值)，每 2x2 个点保存一个 Cr 和Cb 值, 图像在肉眼中的感觉不会起太大的变化。所以, 原来用 RGB(R，G，B 都是 8bit unsigned) 模型，每个点需要 8x3=24 bites。而现在仅需要 8+(8/4)+(8/4)=12bites, 平均每个点占 12bites。这样就把图像的数据压缩了一半。

接下来我们介绍常见的采样格式（以packed格式为例）：

- **4:4:4**

YUV 三个信道的抽样率相同，因此在生成的图像里，每个象素的三个分量信息完整（每个分量通常8比特），经过8比特量化之后，未经压缩的每个像素占用3个字节。

下面的四个像素为:

```
[Y0 U0 V0]
[Y1 U1 V1]
[Y2 U2 V2]
[Y3 U3 V3]
```

存放的码流为:

```
Y0 U0 V0  Y1 U1 V1  Y2 U2 V2  Y3 U3 V3
```

- **4:2:2**

每个色差信道的抽样率是**亮度信道的一半**，所以水平方向的色度抽样率只是 4:4:4 的一半。对非压缩的 8bit量化的图像来说，每个由两个水平方向相邻的像素组成的宏像素需要占用4字节内存（亮度2个字节,两个色度各1个字节）。

下面的四个像素为:

```
[Y0 U0 V0]
[Y1 U1 V1]
[Y2 U2 V2]
[Y3 U3 V3]
```

存放的码流为:

```
Y0 U0  Y1 V1  Y2 U2  Y3 V3
```

映射出像素点为：

```
[Y0 U0 V1]
[Y1 U0 V1]
[Y2 U2 V3]
[Y3 U2 V3]
```

- **4:1:1**

在 **水平方向上对色度进行4:1抽样** 。对于低端设备和消费类产品这仍然是可以接受的。对非压缩的 8bit 量化的视频来说，每个由4个水平方向相邻的像素组成的宏像素需要占用6字节内存（亮度4个字节，两个色度各1个字节）。

下面的四个像素为:

```
[Y0 U0 V0]
[Y1 U1 V1]
[Y2 U2 V2]
[Y3 U3 V3]
```

存放的码流为:

```
Y0 U0  Y1  Y2 V2  Y3
```

映射出像素点为:

```
[Y0 U0 V2]
[Y1 U0 V2]
[Y2 U0 V2]
[Y3 U0 V2]
```

- **4:2:0**

4:2:0 并不意味着只有 Y、Cb，而没有Cr分量。**它指得是对每行扫描线来说，只有一种色度分量以 2:1 的抽样率存储。相邻的扫描行存储不同的色度分量** ， 也就是说，如果一行是 4:2:0 的话，下一行就是 4:0:2，再下一行是4:2:0... 以此类推。对每个色度分量来说，水平方向和竖直方向的抽样率都是 2:1，所以可以说色度的抽样率是4:1。对非压缩的 8bit 量化的视频来说，每个由 2x2 个 2 行 2 列相邻的像素组成的宏像素需要占用 6 字节内存（亮度4个字节,两个色度各1个字节）。

下面八个像素为：

```
[Y0 U0 V0]
[Y1 U1 V1]
[Y2 U2 V2]
[Y3 U3 V3]
[Y5 U5 V5]
[Y6 U6 V6]
[Y7 U7 V7]
[Y8 U8 V8]
```

存放的码流为：

```
Y0 U0  Y1  Y2 U2  Y3  Y5 V5  Y6  Y7 V7  Y8
```

映射出的像素点为：

```
[Y0 U0 V5]
[Y1 U0 V5]
[Y2 U2 V7]
[Y3 U2 V7]
[Y5 U0 V5]
[Y6 U0 V5]
[Y7 U2 V7]
[Y8 U2 V7]
```

三、总结

本节主要讲了图像和视频基础。

- 图像基础包括 Pixel、PPI和颜色等
- 视频基础包括 FPS、YUV、YCbCr、格式采样等

在下一节中，我们将继续介绍 **视频编码原理** ，介绍格式采样的主要目的主要是为视频编码原理作铺垫。