

## 视频知识培训

汪立民

07/16/2017

# 视频基础

## RGB颜色模型

- 三原色组成
  - 红色(Red)
  - 绿色(Green)
  - 蓝色(Blue)
- 一种颜色模型, 将三原色光以不同当比例相加
- 人类有三种视锥细胞分别对红、绿和蓝光最敏感

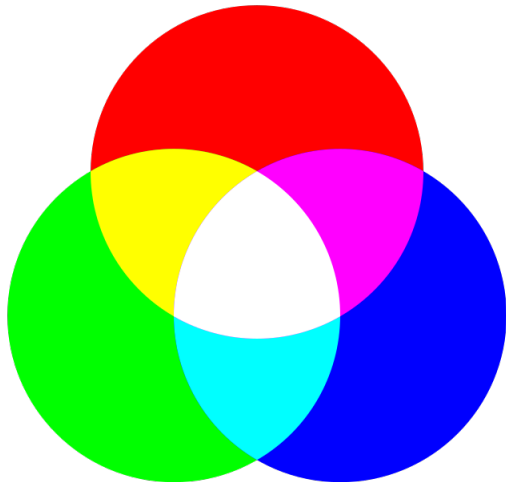


Figure 1: RGB颜色相加

## YUV颜色模型

- 欧洲电视系统所采用的一种颜色编码方法
- YUV设计上可以与黑白电视兼容
- YUV颜色组成
  - Y表示明亮度(Luminance或Luma), 也就是灰度值
  - U和V表示的则是色度 ( Chrominance或Chroma )
- 人眼对Y(亮度)敏感,对UV(色度)不敏感,可以将3个分量以不同带宽传输。
  - 只有Y信号分量而没有U、V分量, 就会显示黑白灰度图像



Figure 2: YUV

## YUV常见格式

- 采样格式
  - YUV 4:4:4
  - YUV 4:2:2
  - YUV 4:1:1
  - YUV 4:2:0
- 存储方式
  - 紧缩格式,
  - 平面格式, 将Y,U,V分别存储在不同的矩阵

## YUV444

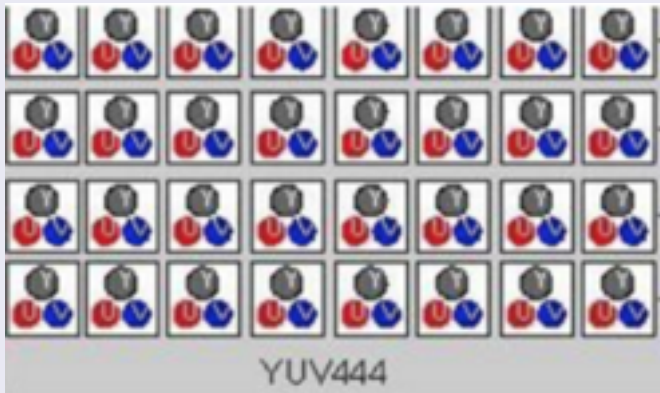


Figure 3: YUV444



## YUV422

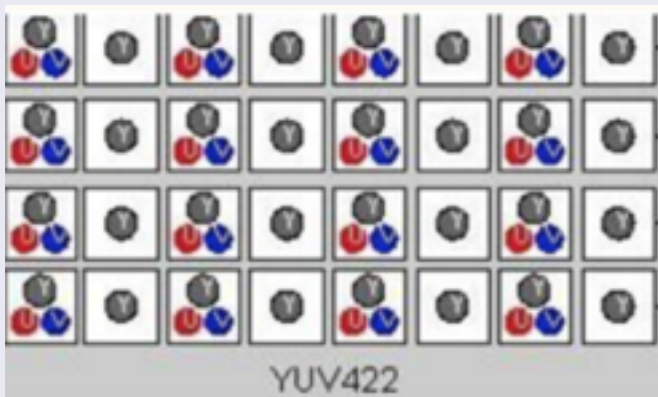


Figure 4: YUV422

## YUV411

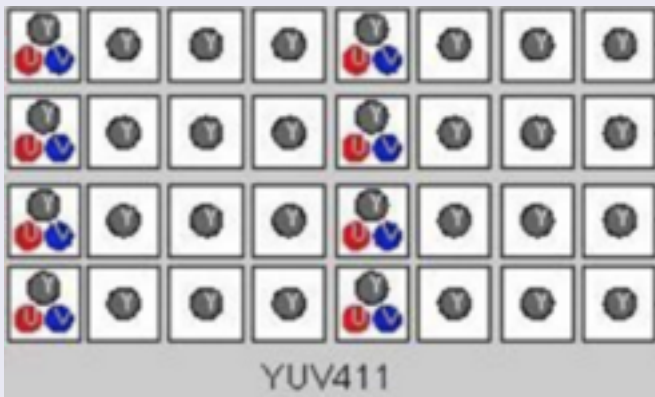


Figure 5: YUV411

## YUV420

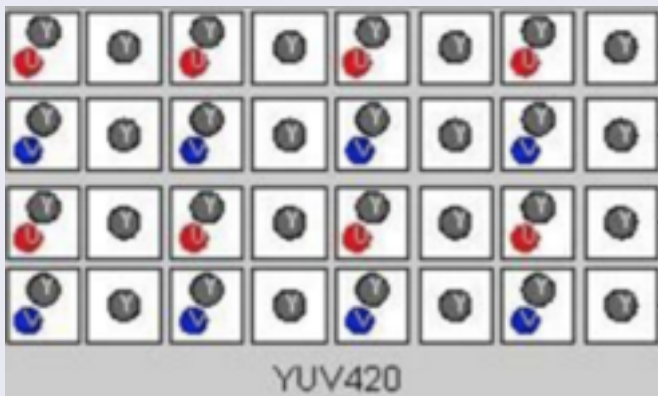


Figure 6: YUV420

## YUV与RGB颜色空间互转

- RGB转YUV

$$Y = (0.257 R) + (0.504 G) + (0.098 B) + 16$$

$$Cr = V = (0.439 R) - (0.368 G) - (0.071 B) + 128$$

$$Cb = U = -(0.148 R) - (0.291 G) + (0.439 * B) + 128$$

- YUV转RGB

$$B = 1.164(Y - 16) + 2.018(U - 128)$$

$$G = 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128)$$

$$R = 1.164(Y - 16) + 1.596(V - 128)$$

## 像素

- 像素，为视频显示的基本单位
- 像素可以是长方形的或者方形的
- 最基本的位图的像素值仅有两个，即黑(0)或白(1)
- 每个像素由不同强度的三原色组成 )
  - 红色像素: 红色 255， 绿色0， 蓝色0
  - 粉色像素: 红色 255、绿色 192、蓝色 203



Figure 7: 像素

## 位深

- 每个颜色的强度，采用一定数量的二进制位数来存储
- 位数大小被称为颜色深度
- 假如每个颜色的强度占用8bit，那么颜色深度就是24

## 分辨率

- 一个平面内像素的数量
- 可以表示成宽 x 高，例如下面这张 4x4 的图像



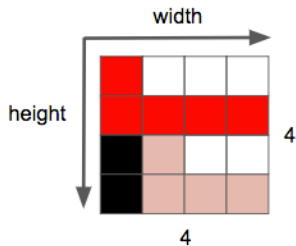


Figure 8: 图像分辨率

## 宽高比

- 图像或像素的宽度和高度之间的一个比例关系。
- 当人们说这个电影或图像是 16:9 时，通常是指显示纵横比（DAR）
- 形状不同的单个像素，我们称为像素纵横比（PAR）。

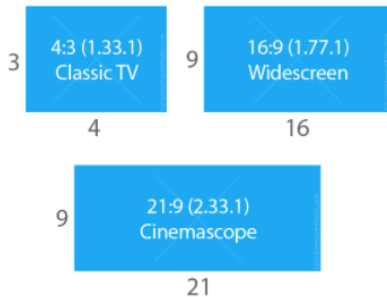


Figure 9: display aspect ratio



Figure 10: pixel aspect ratio

## DVD 4:3 显示纵横比

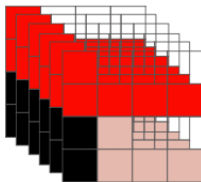
- DVD 的实际分辨率是704x480，但他的纵横比却为4:3
- 它的像素纵横比是10:11 ( 704x10 / 480x11)。

## 帧率

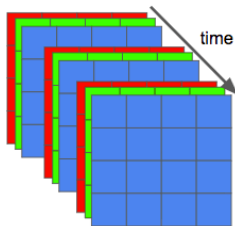
- 视频可以定义为在单位时间内连续的  $n$  帧
- 时间可以视为另一个维度
- $n$  即为帧率，或每秒帧数(FPS)。



a single frame



30 frames per sec  
(FPS)  
*framerate*



4D

Figure 11: 视频

## 比特率

- 播放一段视频每秒所需的比特数就是它的比特率
- 原始比特率 = 宽 \* 高 \* 位深 \* 帧率
  - 例如，帧率30，位深为24，分辨率为480x240的视频
  - 未压缩数据量为82.944 Mbps (30x480x240x24)。
- CBR, 当比特率接近恒定时称为恒定比特率
- VBR, 当比特率是波动的，但最大码率和最小码率范围有限制





Figure 12: VBR

# 帧和场

## 逐行扫描

- 电子束从显示屏的左上角一行接一行地扫到右下角
- 新的显示设备已经可以实时处理和扫描整个帧, 逐行扫描将慢慢成为主流

## 隔行扫描

- 将一帧图像的奇数行画素及偶数行画素分开, 分成为两个场(field)
- 轮流扫描奇数行所构成的上场及偶数行所构成的场
- 上场, 下场优先顺序也是需要关注

## 场出现的原因

- 人眼感受到连贯运动的最小帧率是24fps, 能接受的最小闪烁次数是48次/秒
- 电视带宽有限, 传输50帧的视频数据, 成本太高
- 发明了场的技术, 将一帧分两次传输和显示来解决最低闪烁要求

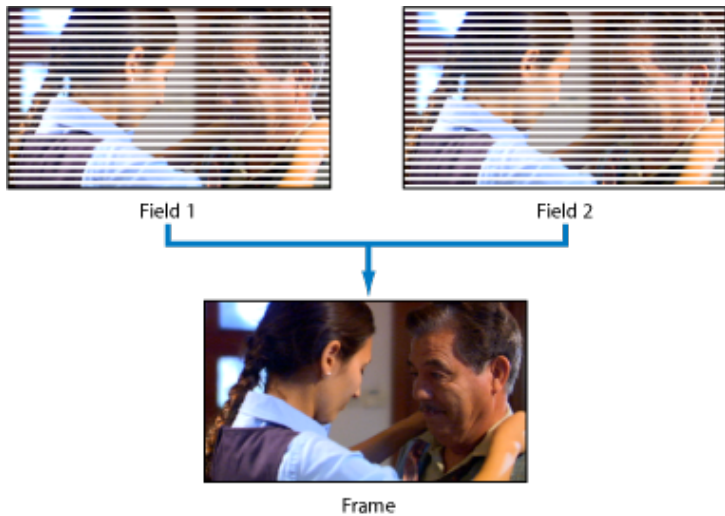


Figure 13: interlaced\_field

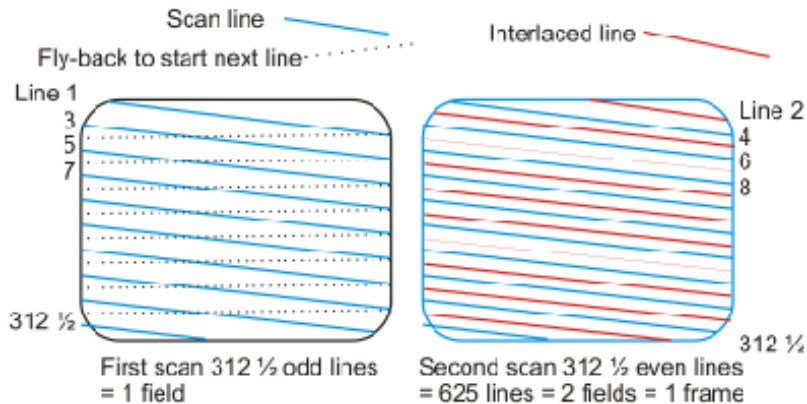


Figure 14: interlaced scan

# 反交错方法

## 反交错背景

- 电脑显示器，手机，PAD等播放设备只能逐行播放显示
- 新式显示设备如高清，4K电视只能支持逐行显示
- 在逐行显示设备上，直接播放交错式视频有严重的闪烁和锯齿现象

## 反交错的方法, 取决于源

- 经过3:2 Pulldown后的电影, 本身就是逐行视频，可以完美反交错
- 拍摄交错式影像的摄影机,  
如果物品有移动动作，直接将上下场混合为帧有锯齿效果

### 3:2 Pulldown

- 将每秒24帧的渐进式影像转换为每秒60场的交错式影像
- 做过3:2 pull-down的电影就变成30fps的了，在NTSC的电视上播放就没有同步问题
- 转换步骤：
  - 先将每个帧拆开成为两个场，场A与场B
  - 先第一帧的场A、第一帧的场B、第一帧的场A
  - 第二帧的场B、第二帧的场A
  - 第三帧的场B、第三帧的场A、第三帧的场B、
  - 第四帧的场A、第四帧的场B的顺序排列

## IVTC(Inverse Telecine)

- 将源自电影的交错式影像去交错从30帧恢复为24帧逐行视频
- 检测到第一场和第三场一样
- 转换步骤：
  - 只要将收到的前两个场合并为一个帧
  - 第三个场丢弃，第四个与第五个场合并成为第二个帧
  - 第六个场丢弃，第七个与第八个场合并成为第三个帧
  - 第九个与第十个场合并成为第四个帧
  - 重复以上步骤

## 反交错隔行影像

- 无论用任何方法都无法完美的回复失去的一半资讯
- 常用方法
  - 单一场去交错, 简单来说就是将一个场放大成为一个帧的大小再播出
  - 场间去交错, 将连续的两个场直接结合成为一个帧, 不做任何修改
  - 动态补偿去交错, 检测边缘, 动态补偿, 算法非常复杂





Figure 15: 反交错比较

## 制式说明

- NTSC电视标准, 用于美、日等国家和地区
  - 525 行/帧, 每秒29.97帧 (简化为30帧)
  - 隔行扫描, 一帧分成2 场(field), 262.5 线/场
  - 高宽比: 电视画面的长宽比(电视为4:3; 电影为3:2; 高清晰度电视为16:9)
- PAL制式, 主要用于中国、欧洲等国家和地区
  - 625 行(扫描线)/帧, 每秒25帧
  - 隔行扫描, 2 场/帧, 312.5 行/场
  - 画面的宽高比为4:3
- SECAM制式, 是法文的缩写, 法国提出并制定标准

结束，谢谢！