C/C++Linux服务器开发

高级架构师课程

三年课程沉淀 | 五次精益升级 | 十年行业积累 | 百个实战项目 | 十万内容受众

讲师:darren/326873713

扫一扫 升职加薪

班主任:柚子/2690491738

讲师介绍--专业来自专注和实力

Darren老师

曾供职于国内知名半导体公司（珠海扬智/深圳联发科），曾在某互联网公司担任音视频通话项目经理。主要从事音视频驱动、多媒体中间件、流媒体服务器的开发，开发过即时通讯+音视频通话的大型项目，在音视频、C/C++/GO Linux服务器领域有丰富的实战经验。

0 本节内容

1. 图像基础概念
2. YUV/RGB深入讲解

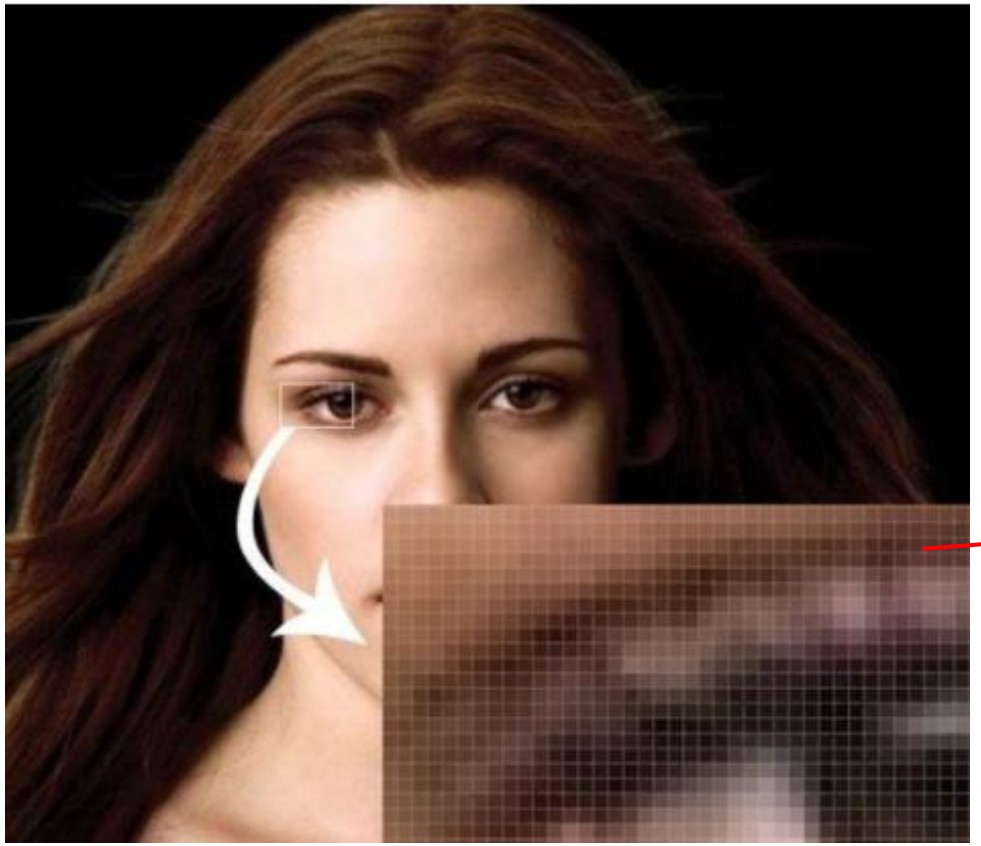
1 图像基础概念

- **像素**：像素是一个图片的基本单位，pix是英语单词picture的简写，加上英语单词“元素element”，就得到了“pixel”，简称px，所以“像素”有“图像元素”之意。
- **分辨率**：是指图像的大小或尺寸。比如1920x1080。
- **位深**：是指在记录数字图像的颜色时，计算机实际上是用每个像素需要的位深来表示的。比如红色分量用8bit。
- **帧率**：在1秒钟时间里传输的图片的帧数，也可以理解为图形处理器每秒钟能够刷新几次。比如25fps表示一秒有25张图片。
- **码率**：视频文件在单位时间内使用的数据流量。比如1Mbps。
- **Stride**：指在内存中每行像素所占的空间。为了实现内存对齐每行像素在内存中所占的空间并不一定是图像的宽度。

1.1 像素

像素是一个图片的基本单位，pix是英语单词picture的简写，加上英语单词“元素element”，就得到了“pixel”，简称px，所以“像素”有“图像元素”之意。

例如2500×2000的照片就是指横向有2500个像素点，竖向有2000个像素点，总共是500万个像素，也俗称500万像素照片。



方块就是像素点

1.2 分辨率

图像（或视频）的分辨率是指图像的大小或尺寸。我们通常用像素表示图像的尺寸。

例如2500×2000的照片就是指横向(宽)有2500个像素点，竖向(高)有2000个像素点。

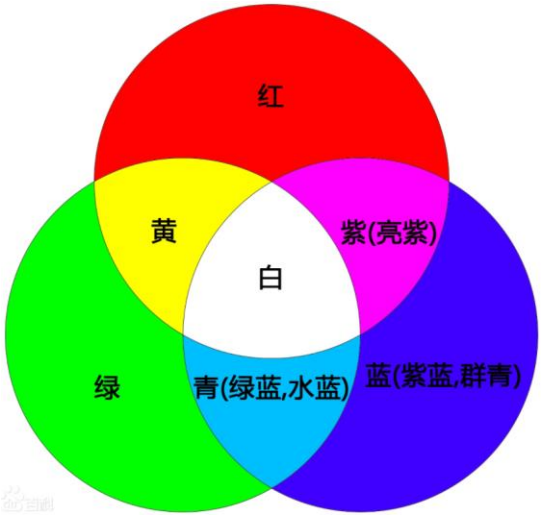
常见的分辨率：
360P(640x360)、720P(1280x720)、1080P(1920x1080)、4K(3840x2160)、8K(7680x4320)

1.2 分辨率-不同分辨率之间的区别

常说的 1080 和 720 其实是指垂直像素数，分辨率除去垂直像素，还需要考虑到水平像素数。按照 16:9 (宽 : 高) 的比例计算，720p 的水平像素数为 720 ÷ 9 × 16 = 1280，总计像素为921600像素即大约为 92 万像素。1080p 具有 1920 个水平像素，总计2073600像素即约 200 万像素，是 720p 的两倍多。

像素越多视频就越清晰，所以 1080p 比 720p 的视频更加清晰。**图像的分辨率越高，图像就越清晰。**

1.3 位深



我们看到的彩色图片，都有三个通道，分别为红(R)、绿(G)、蓝(B)通道。（如果需要透明度则还有alpha分量）

通常每个通道用8bit表示，8bit能表示256种颜色，所以可以组成256*256*256=16,777,216 = 1677万种颜色。
这里的8bit就是我们讲的位深。

每个通道的位深越大，能够表示的颜色值就越大，比如现在高端电视说的10bit色彩，即是每个通道用10bit表示，每个通道有1024种颜色。1024*1024*1024约为10,7374万色=10亿色， 是8bit的64倍。

常见的颜色还是8bit居多。

1.4 帧率

帧率即 FPS（每秒有多少帧画面），经常玩游戏的同学应该会对这个词很熟悉。我们玩游戏时，FPS 帧率越高就代表游戏画面越流畅，越低则越卡顿。视频也是如此。

由于视觉图像在视网膜的暂时停留，一般图像帧率能达到24帧，我们就认为图像是连续动态的。

电影帧率一般是 24fps（帧每秒）；
电视剧一般是25fps;
监控行业常用 25fps;
音视频通话常用15fps;

帧率越高，画面越流畅，需要的设备性能也越高。

播放 5帧、24帧视频对比。

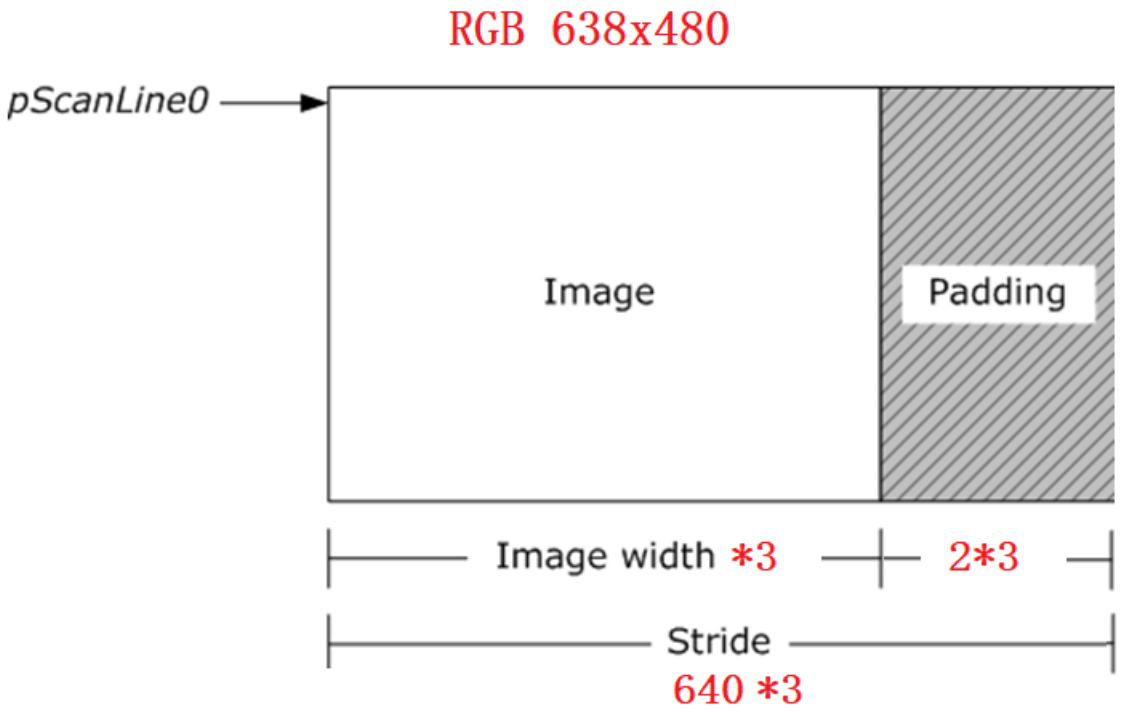
1.5 码率

- 视频文件在单位时间内使用的数据流量。比如1Mbps。
- 大多数情况下码率越高 分辨率越高，也就越清晰。但模糊的视频文件大小（码率）也可以很大，分辨率小的视频文件可能也比分辨率大的视频文件清晰。
- 对于同一个原始图像源的时候，同样的编码算法，则码率越高，图像的失真就会越小，视频画面就会越清晰

对比不同码率的转码。

1.6 Stride跨距

- Stride： 指在内存中每行像素所占的空间。为了实现**内存对齐**每行像素在内存中所占的空间并不一定是图像的宽度。
- Stride 就是这些扩展内容的名称，Stride 也被称作 Pitch，如果图像的每一行像素末尾拥有扩展内容，Stride 的值一定大于图像的宽度值，就像下图所示：
  - 比如分辨率638x480的RGB24图像，我们在内存处理的时候如果要以16字节对齐，则638*3/16=119.625不能整除，因此不能16字节对齐，我们需要在每行尾部填充6个字节。就是(638+2->640)，640*3/16=120。此时该图片的stride为1920字节。

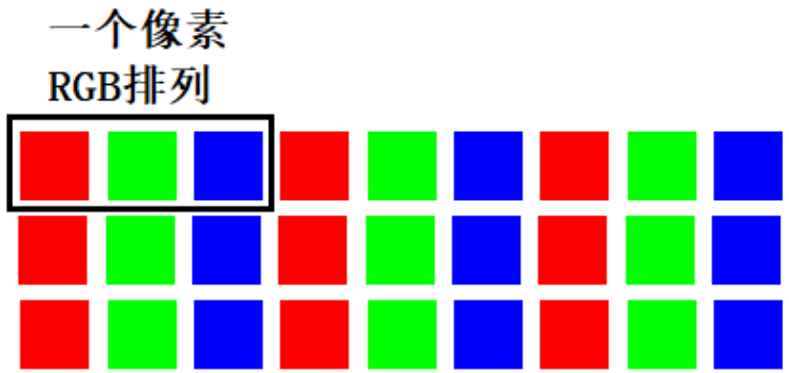


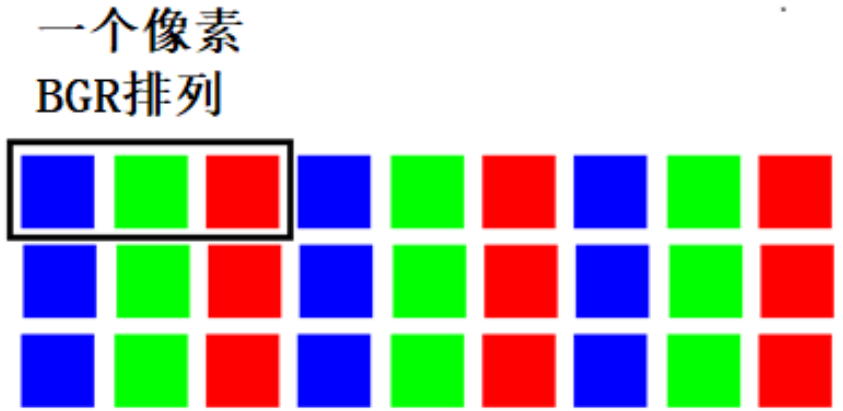
2 RGB、YUV深入讲解

- **RGB**：红R、绿G、蓝B三基色。
- **YUV**：“Y”表示明亮度（Luminance或Luma），也就是灰阶值，“U”和“V”表示的则是色度（Chrominance或Chroma）。

2.1 RGB

- 我们前面已经讲过RGB色彩表示，这里我们重点讲RGB的排列。通常的图像像素是按RGB顺序进行排列，但有些图像处理要转成其他顺序，比如OpenCV经常转成BGR的排列方式。

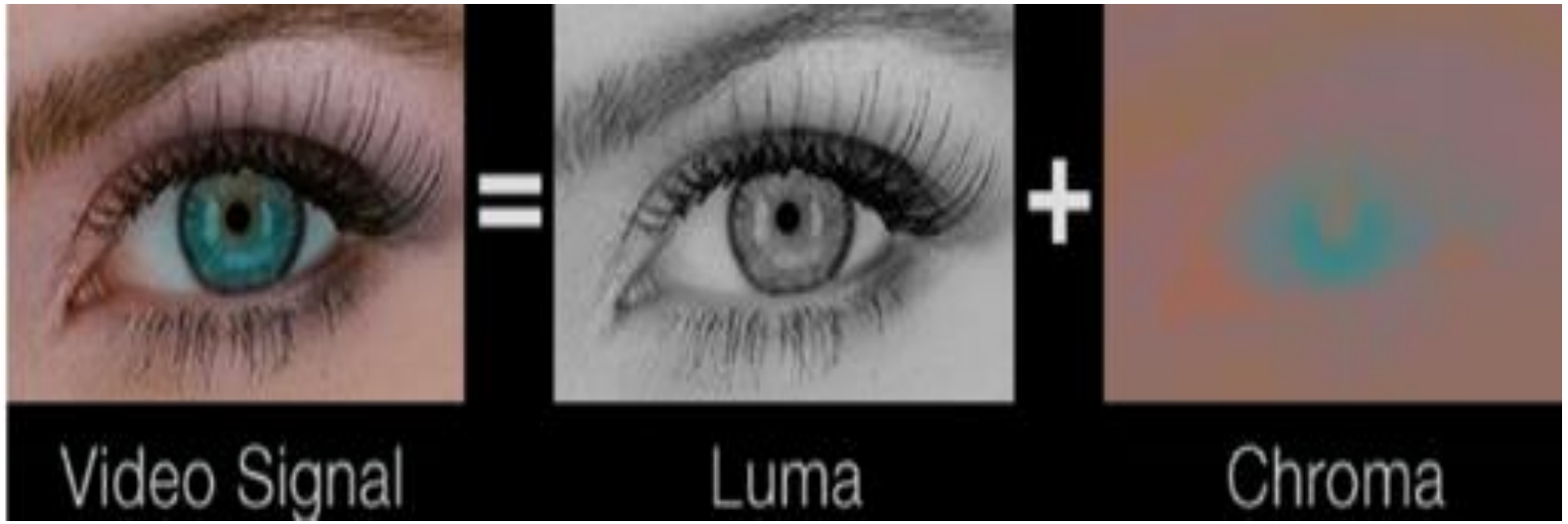





```
AV_PIX_FMT_RGB24,     ///< packed RGB 8:8:8, 24bpp, RGBRGB...
AV_PIX_FMT_BGR24,     ///< packed RGB 8:8:8, 24bpp, BGRBGR...
AV_PIX_FMT_ARGB,      ///< packed ARGB 8:8:8:8, 32bpp, ARGBARGB...
AV_PIX_FMT_RGBA,      ///< packed RGBA 8:8:8:8, 32bpp, RGBARGBA...
AV_PIX_FMT_ABGR,      ///< packed ABGR 8:8:8:8, 32bpp, ABGRABGR...
AV_PIX_FMT_BGRA,      ///< packed BGRA 8:8:8:8, 32bpp, BGRABGRA...
```

2. 2 YUV

- 与我们熟知RGB类似，YUV也是一种颜色编码方法，它是指将亮度参量（Y：Luminance或Luma）和色度参量（UV：Chrominance或Chroma）分开进行表示的像素编码格式。
- 这样分开的好处就是不但可以避免相互干扰，没有UV信息一样可以显示完整的图像，因而解决了彩色电视与黑白电视的兼容问题；还可以**降低色度的采样率**而不会对图像质量影响太大，降低了视屏信号传输时对频宽（带宽）的要求。
- Y Y共用一组UV分量

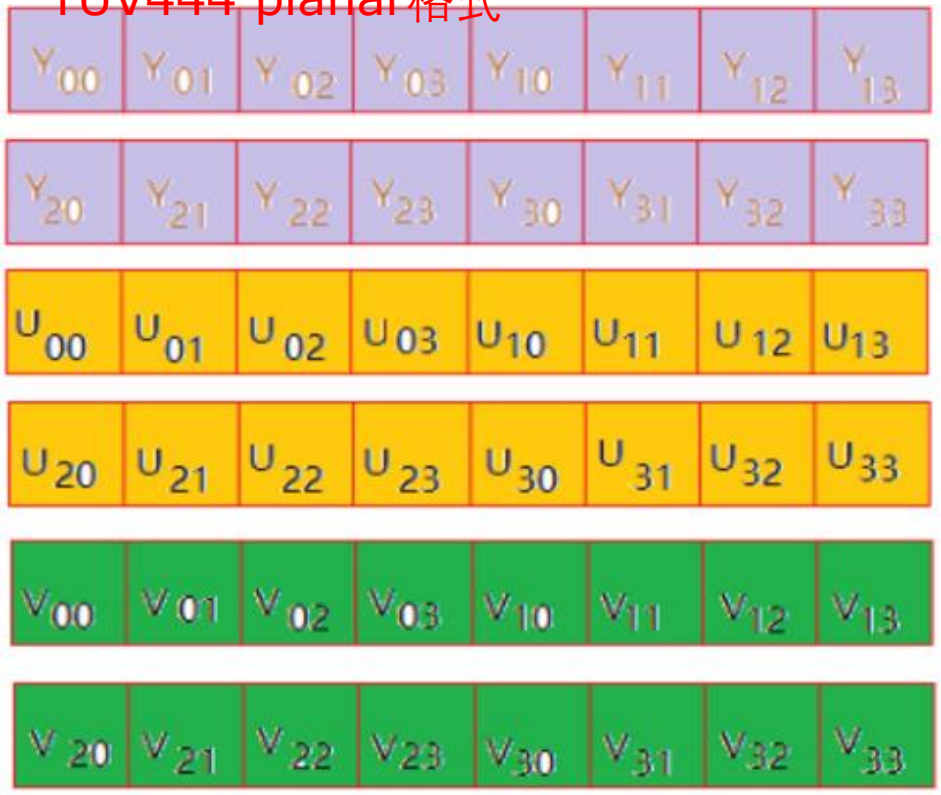


2. 2 YUV

- YUV是一个比较笼统地说法，针对它的具体排列方式，可以分为很多种具体的格式：
  - 打包（packed）格式：将**每个像素点的Y、U、V分量交叉排列**并以像素点为单元连续的存放在同一数组中，通常几个相邻的像素组成一个宏像素（macro-pixel）
  - 平面（planar）格式：**使用三个数组分开连续的存放Y、U、V三个分量**，即Y、U、V分别存放在各自的数组中。

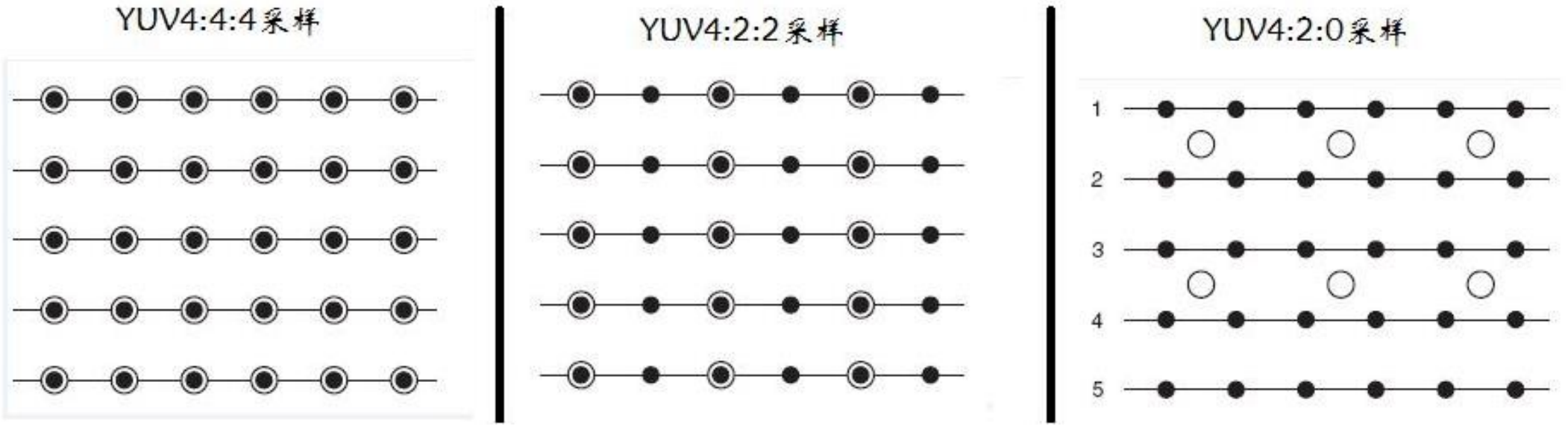
YUV444 packed格式

YUV444 planar格式



2.2.1 YUV采样表示法

- YUV采用A:B:C表示法来描述Y,U,V采样频率比例，下图中黑点表示采样像素点Y分量，**空心圆表示采样像素点的UV分量**。主要分为 YUV 4:4:4、YUV 4:2:2、YUV 4:2:0 这几种常用的类型



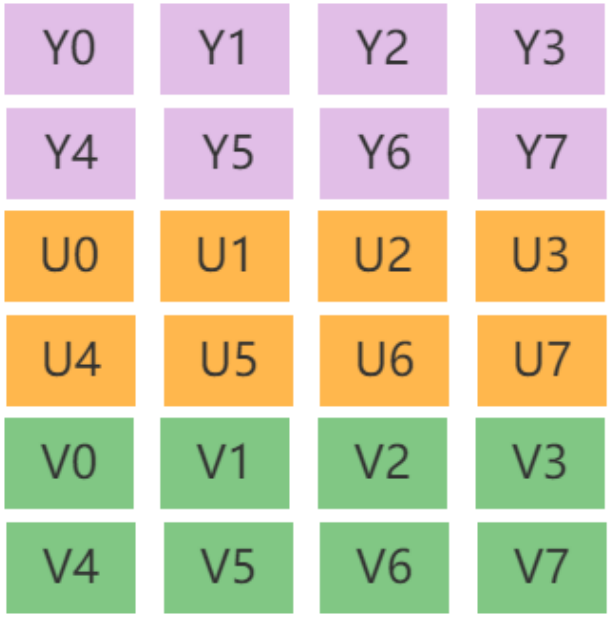
- 4:4:4 表示色度频道没有下采样，即一个Y分量对应着一个U分量和一个V分量。
- 4:2:2 表示 2:1 的水平下采样，没有垂直下采样，即每两个Y分量共用一个U分量和一个V分量。
- 4:2:0 表示 2:1 的水平下采样，2:1 的垂直下采样，即每四个Y分量共用一个U分量和一个V分量。

2.2.2 YUV数据存储

- 下面以每个分量数据存储在一个char（或byte）中为例描述YUV的数据存储方式。
  1. 4:4:4格式
  2. 4:2:2格式
  3. 4:2:0格式

2.2.2 YUV数据存储-4:4:4格式

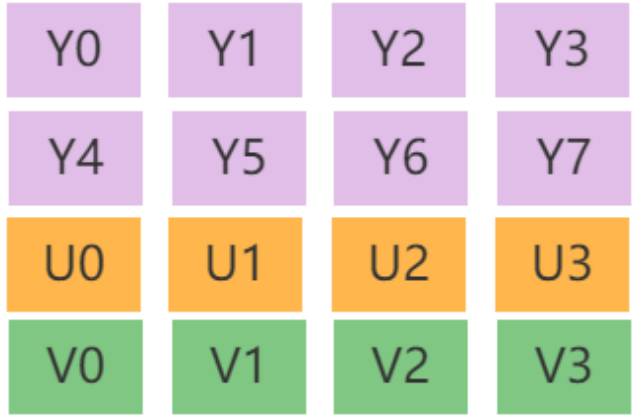
- 比如I444(YUV444P)格式, 对应Ffmpeg像素表示AV_PIX_FMT_YUV444P, ///< planar YUV 4:4:4, 24bpp, (1 Cr & Cb sample per 1x1 Y samples)



I444(YUV444P)格式

2.2.2 YUV数据存储-4:2:2格式

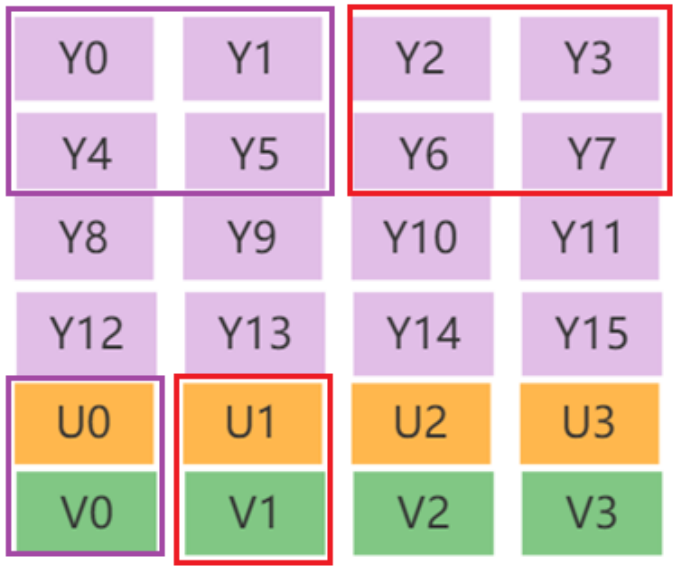
- 比如I422(YUV422P)格式
  - 对应Ffmpeg像素表示AV_PIX_FMT_**YUV422P,** ///< planar YUV 4:2:2, 16bpp, (1 Cr & Cb sample per 2x1 Y samples)
  - 该类型为planar格式



I422(YUV422P)格式

2.2.2 YUV数据存储-4:2:0格式-YUV420P

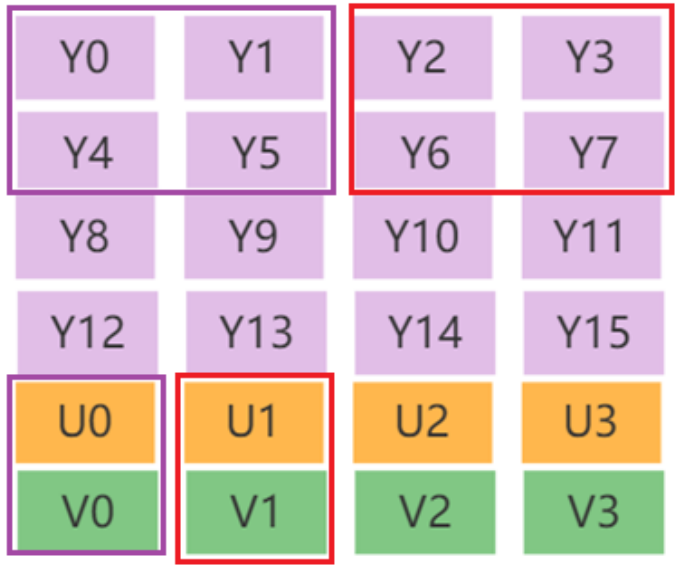
- 比如I420(YUV420P)格式
  - 对应Ffmpeg像素表示AV_PIX_FMT_YUV420P, ///< planar YUV 4:2:0, 12bpp, (1 Cr & Cb sample per 2x2 Y samples)
  - 该类型为planar格式
  - (4+1+1)/4 = 1.5字节



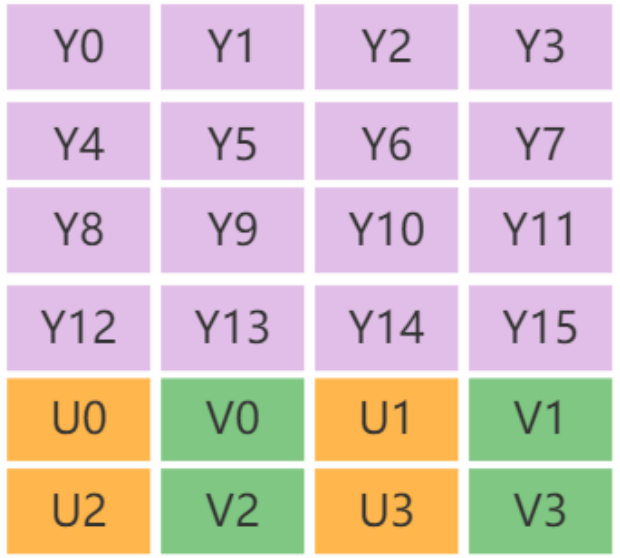
I420(YUV420P)格式

2.2.2 YUV数据存储-4:2:0格式-NV12

- 比如NV12格式
  - 对应Ffmpeg像素表示AV_PIX_FMT_NV12, ///< planar YUV 4:2:0, 12bpp, 1 plane for Y and 1 plane for the UV components, which are interleaved (first byte U and the following byte V)

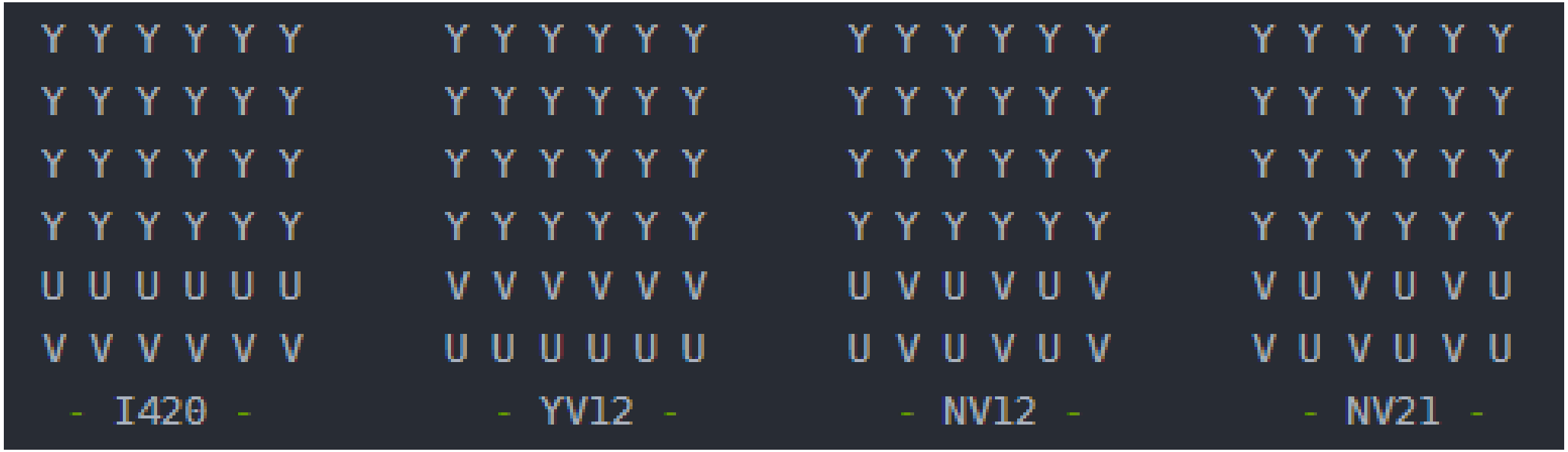


I420(YUV420P)格式



NV12(YUV420SP)格式

2.2.2 YUV数据存储-4:2:0格式-参考



■ YUV420p：

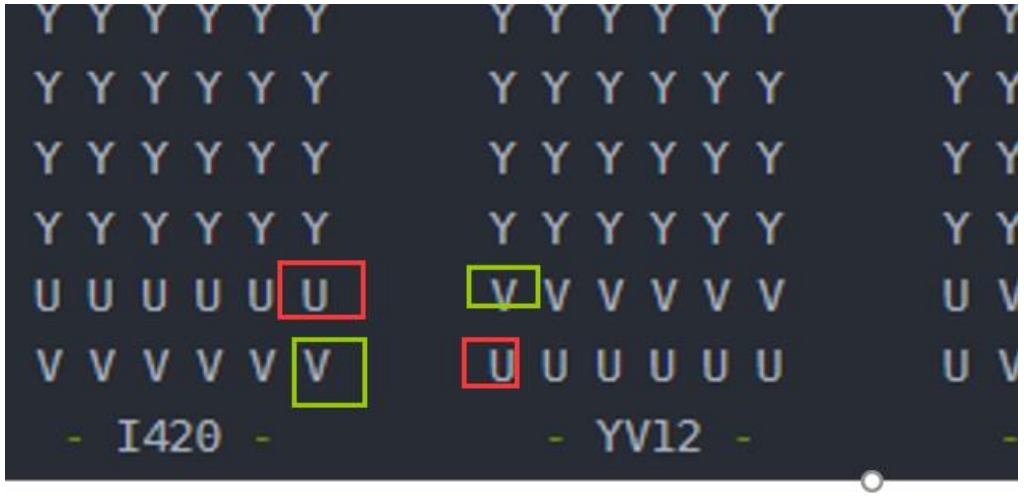
YV12： YYYYYYYY VV UU

I420： YYYYYYYY UU VV

■ YUV420sp：

NV12： YYYYYYYY UV UV

NV21： YYYYYYYY VU VU



2.3 RGB和YUV的转换

- 通常情况下RGB和YUV直接的相互转换都是调用接口实现，比如Ffmpeg的swscale或者libyuv等库。
- 主要转换标准是 BT601 和 BT709。

8bit位深的情况下

- TV range是16-235(Y)、16-240(UV) , 也叫Limited Range
- PC range是0-255，也叫Full Range
- 而RGB没有range之分，全是0-255

- BT601 TV Range转换公式

YUV(256 级别) 可以从8位 RGB 直接计算:

Y = 0.299*R + 0.587*G + 0.114*B;
U = -0.169*R - 0.331*G + 0.5 *B ;
V = 0.5 *R - 0.419*G - 0.081*B;

反过来，RGB 也可以直接从YUV (256级别) 计算:

R = Y + 1.402 (Y-128)
G = Y - 0.34414 (U-128) - 0.71414 (U-128)
B = Y + 1.772 (V-128)

- 从YUV 转到 RGB 如果值小于0要取0，如果大于255要取255

2.3 RGB和YUV的转换-为什么解码出错显示绿屏

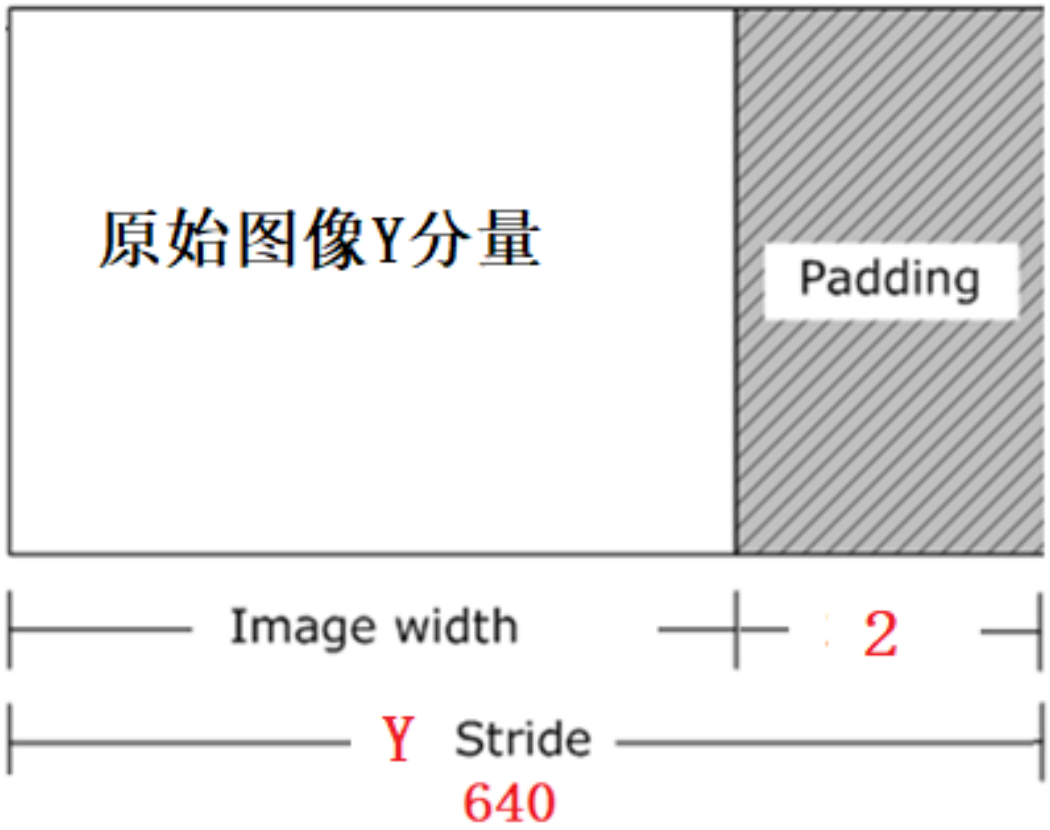
因为解码失败时YUV分量都填为0值，然后根据公式：
R = 1.402 * (-128) = -126.598
G = -0.34414*(-128) - 0.71414*(-128) = 44.04992 + 91.40992 = 135.45984
B = 1.772 * (-128) = -126.228

RGB 值范围为[0，255]， 所以最终的值为：
R = 0
G = 135.45984
B = 0
此时只有G分量有值所以为绿色。

YUV -> RGB

2.4 YUV Stride对齐问题

比如分辨率638x480的YUV420P图像，我们在内存处理的时候如果要以16字节对齐，则638不能被16整除，我们需要在每行尾部填充2个字节。就是640。此时该图片的Y stride为640字节。



测试视频下载地址

1、地址：http://clips.vorwaerts-gmbh.de/big_buck_bunny.mp4 1分钟
2、地址：http://vjs.zencdn.net/v/oceans.mp4
3、地址：https://media.w3.org/2010/05/sintel/trailer.mp4 52秒
4、http://mirror.aarnet.edu.au/pub/TED-talks/911Mothers_2010W-480p.mp4 10分钟

其他各种格式，MP4, flv, mkv, 3gp 视频下载
https://www.sample-videos.com/index.php#sample-mp4-video

参考文档

[看视频常见的 720p、1080p、4k，这些分辨率到底包含了什么 - SegmentFault 思否]

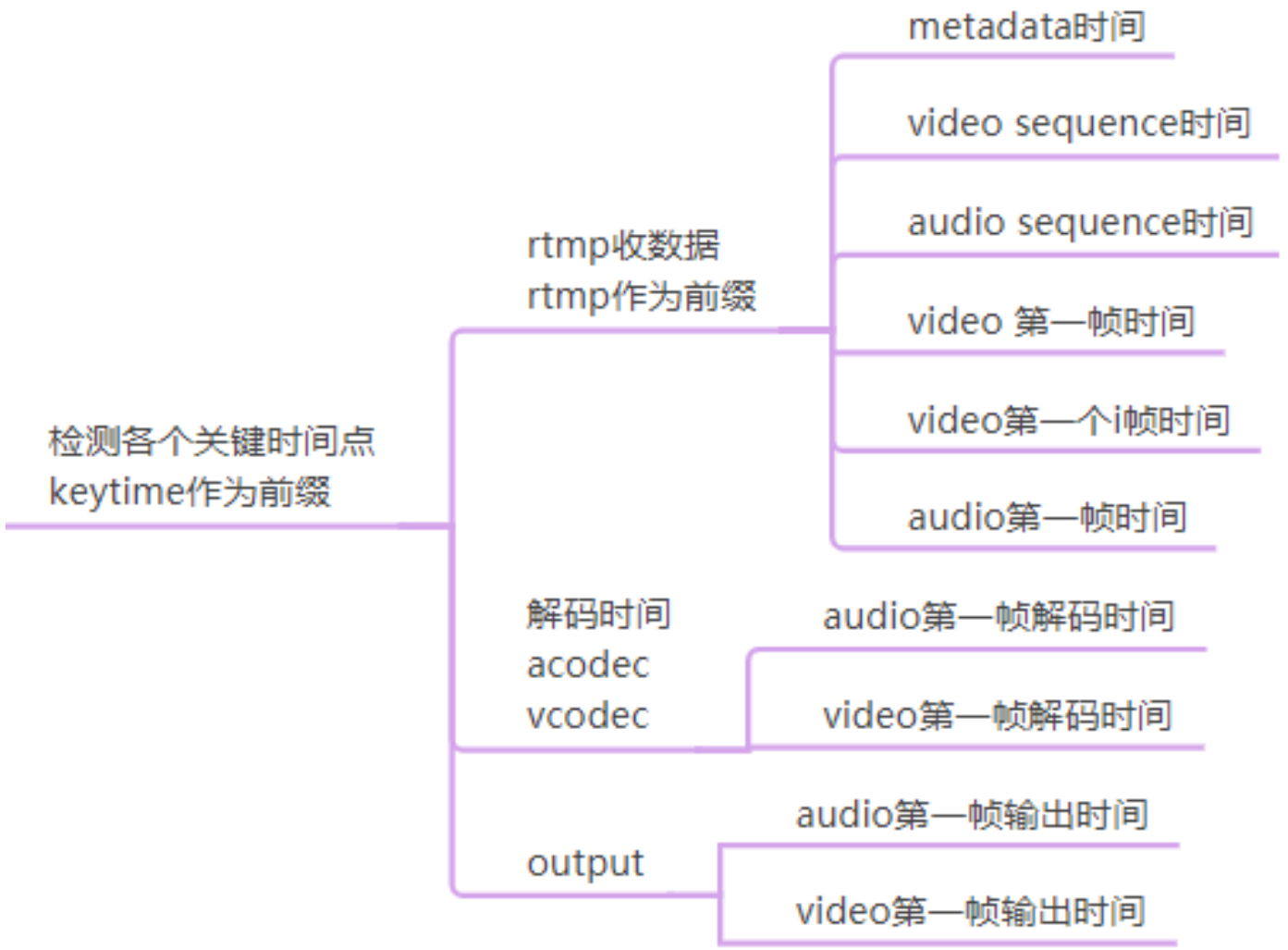
YUV详解：https://blog.csdn.net/u013898698/article/details/54927203

音视频同步

Avsync模块

目前只支持audio master的方式。

各个模块关键时间点的监测



其他

1. 客户端的首帧秒开，本质上就是不做同步先把第一帧显示出来。

2. 推流没有问题时，如果拉流不能正常播放：
    1. 没有声音：dump rtmp拉流后的数据是否可以正常播放
    2. 声音异常：是否有解码错误报告，重采样前的pcm数据是否正常
    3. 没有图像： dump rtmp拉流后的数据是否可以正常播放
    4. 画面异常：是否有解码错误报告，scale前的数据是否正常

服务器首帧秒开：这个功能不能降低延迟