



江疏影独家代言，登录免费送透视戒指，888%超高爆率！



年末大回馈，打怪爆神戒，一直打一直掉！

# 音视频结构：带你深入理解YUV



码农叔叔

关注

1

2019.07.04 18:02:56

字数 3,700

阅读

赏

10赞

赞赏

## 前言

在音视频开发中，RGB和YUV这些颜色空间估计大家都不会陌生。相对来说RGB比较容易理解，但是YUV的一些概念可能会比较抽象，所以这篇文章就主要来介绍YUV这个颜色空间的一些知识。

## 什么是YUV


要理清YUV的本质，那么必须从其Y、U、V这三个维度着手，从网上抄来的对于此的介绍大体如下：

- Y维度：表示明亮度（Luminance或Luma），也就是灰度值。
- U和V维度：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。

但是，对于没有多少美工基础的同学，着实是不好理解，所以接下来开始看图理解教学。

先来欣赏一张小可爱的图片：





码农叔叔

总资产24

关注

QUIC探索（二）：编译第一个QUIC工程

阅读 6,782

ijkplayer 播放器：在Android端进行native调试

阅读 4,111

### 热门故事

为了爱情他把一条蛇放在了...

六道干探 | 沙漠里肆虐的虫族

是我逼死了那个快递小哥？！

我死以后的一年，定要找出凶手

藏在矿山里的碎尸案！起因竟是VX

摇一摇？

### 推荐阅读

FFmpeg入门 - Android移植

阅读 169

numpy多维数组的索引

阅读 307

(9) python PIL

阅读 81

CUDA opencv 彩色图转灰度图

阅读 114

关于科研的5点

阅读 102



原图

然后通过技术手段(万能的代码)，将UV置为默认值(默认值是多少后面解释)，然后得出如下这张图：

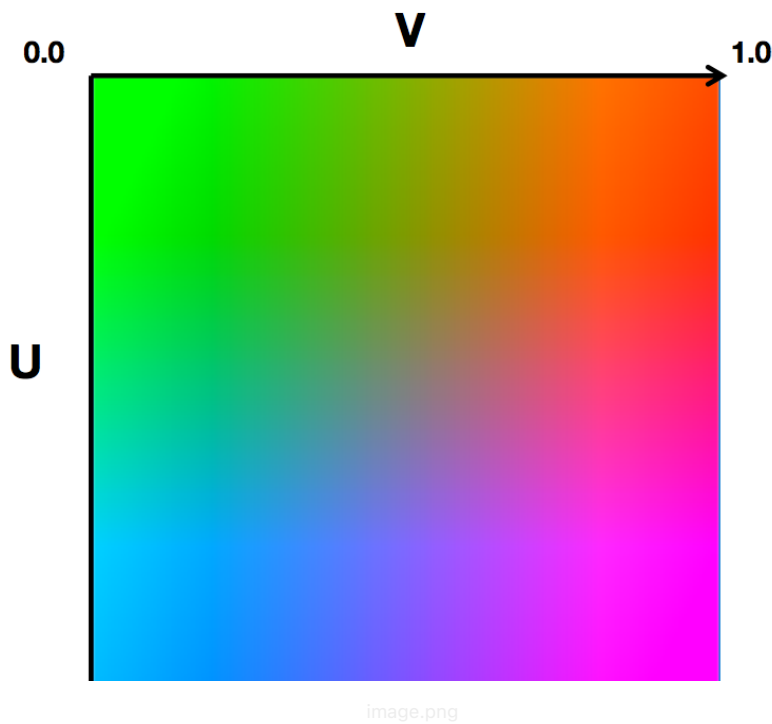


去除UV后的图

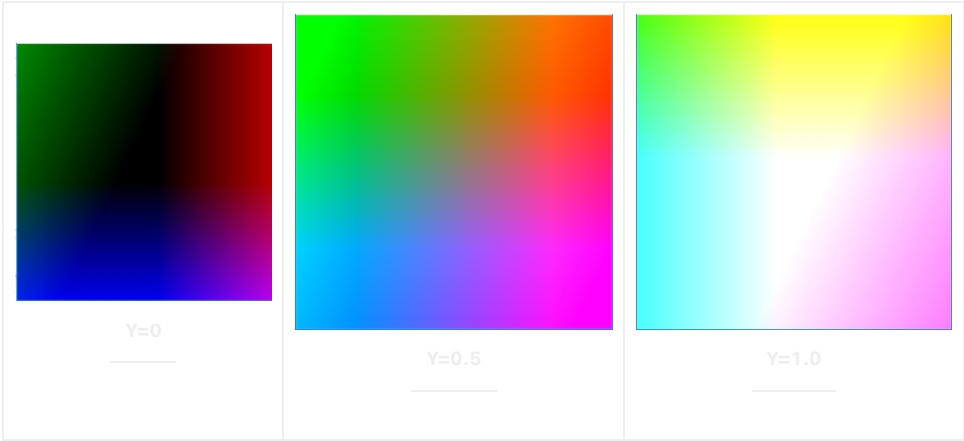
从这里可得出很明显的结论，如果只保留Y通道的话，我们可以获得一张灰度图妹子图，也就是说Y维度其实保留得是图片的轮廓。

那么UV具体是什么样子呢，这里假设三个维度的取值都是  $[0.0, 1.0]$ ，当我们Y取中间值  $0.5$ ，那么可以得出如下一张映射图，看到这里大家估计就知道UV是怎么表示颜色了吧：





看完了UV，那么Y呢？  
下面是Y分别取 0.0， 0.5， 1.0 的时候，在UV的各个组合的映射图，请观察如下三张图：



很明显可以看出，随着Y维度数值的增大，图片她开始发亮了！

PS：额外说一点，做音视频小伙伴经常会遇到画面绿屏现象的现在，麻烦翻上去观察每个映射图的左上角，是不是一大坨绿色？醒醒，赶紧去检查你的byte数组是不是忘记赋值，全部是0。

看起来YUV很简单嘛，这里麻烦请停下你想吹牛逼说已经理解YUV的冲动想法，因为接下来才是开始划重点。

## YUV与RGB的转换关系

在业务开发过程中，免不了需要进行YUV与RGB之间进行颜色转换，那么请翻开《XX入门指南》的10086页，找到如下两个矩阵计算公式：

## RGB转YUV计算矩阵

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1678 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RGB转YUV计算矩阵

## YUV转RGB计算矩阵

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

YUV转RGB计算矩阵

什么？你忘了矩阵的计算方法？

算了算了，跳过看下面的直接计算公式好了。

由于我们经常用 **byte (8bit)** 来保持YUV，所以这里以 **[0, 255]** 作为取值范围计算。

①、YUV转RGB计算公式：

$$R = Y + 1.402 (V - 128)$$

$$G = Y - 0.34414 (U - 128) - 0.71414 (V - 128)$$

$$B = Y + 1.772 (U - 128)$$

②、RGB转YUV计算公式：

$$Y = 0.299 R + 0.587 G + 0.114 B$$

$$U = -0.1687 R - 0.3313 G + 0.5 B + 128$$

$$V = 0.5 R - 0.4187 G - 0.0813 B + 128$$

下面开始记录知识点：

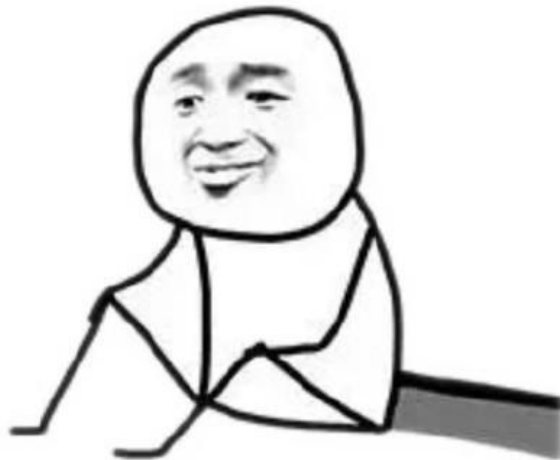
- 从①可知，当U、V取值 **128** 时，可以得出R、G、B都等同于Y值，而在RGB颜色空间我们可以得知当R、G、B数值相等时，显现的画面也正是灰度图，由此可得U、V的缺省默认值应该取 **128**，也就是中间值。
- 由①的R计算公式可知，V与R颜色呈正相关关系，姑且可以认为V越大红色的信息则越强；由B计算公式可得出U越大则蓝色信息越强；而G则受到UV的负相关影响。
- 由②可知，RGB转YUV就仅仅纯粹一个计算公式而已，逃~

## YUV的采样格式

接下来就得了解一下YUV的采样格式，例如大名鼎鼎的YUV420。

曾经的萌新小朋友一度以为YUV420就是Y:U:V=4:2:0，额，就没发现V被你吃了么？

# 心情复杂

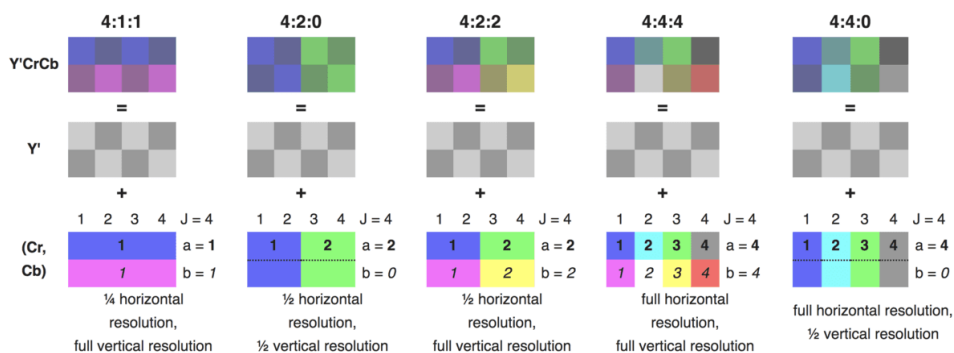


心情复杂

有疑惑就得找解答，没钱请老师就得多翻翻问谷歌度娘，终于我发现了如下几条真理：

- 人类视觉系统对亮度 **luma** 的敏感度高于对色度 **chroma** 的敏感度，因此可以对色度数据进行压缩，官方口径的说法是可以对其进行下采样。
- 由于U和V共同作用产生一个颜色(色度)，所以UV都是成对出现的，一般都是将成对的UV值说成一个色度值。
- 采样比通常表示为 **J:a:b**，以表示一个宽为J像素、高为2像素的采样区域里面的Y U V的采样分部：
  - 1、J表示采样区域的宽度，如果没意外一般都是4
  - 2、a表示采样区域第一行的色度采样数
  - 3、b表示采样区域第二行的色度采样数
  - 4、如果b是0，那么采样区域第二行复用第一行的色度采样

如果你懒得琢磨上面的真理的话，那么一张图可能更适合你：



总结如下：

- **4:1:1**: 在宽度J为4像素、高2像素的采样区域里面，因为a的值是1，所以第一行的4个像素都复用着同一个色度，然而b的值也是1，很明显第二行的4个像素应该也复用着另外一个色度。
- **4:2:0**: 在宽度J为4像素、高2像素的采样区域里面，因为a的值是2，所以第一行的每2个像素都复用着同一个色度，也就是第一行说色度数有 $4/2=2$ 个，然后b的值是0，说明第二行的4个像素需要去复用第一行的色度采样值。
- **4:4:4**: 在宽度J为4像素、高2像素的采样区域里面，因为a的值是4，所以第一行的每1个像素都有其对应的一个色度，也就是第一行说色度数有 $4/1=4$ 个，然后b的值也是4，说明第一行的每1个像素也都有其对应的色度值，也就是说第二行的色度数也有 $4/1=4$ 个。

PS：在计算机领域，YCbCr 与 YUV 基本可以认为是等同的。

接着进入没奖问答环节：

- 1、你能再分析出YUV422和YUV440的采样分布吗？
- 2、如果以byte(8bit)作为Y、U、V的精度来存储，在得知width/height的情况下，请计算出相应内存占用。

## YUV的存储格式

当我们辛辛苦苦分析那么多YUV采样格式，理解了采样规律；但是面对一个存放byte数组的YUV数据，震惊是不是漏了一件很重要的事情，它这个鬼究竟是怎么在内存里面存放的？！

额，反正很多人都这么告诉我，YUV的存储格式大概可以分为 **planar(平面)**、**semi planar(半平面)**、**packed(打包)** 三个大类别，那么具体每个类别有什么特性呢？

- **平面格式**：因为其拥有三个plane也被叫做planar格式，第一个平面存储所有的Y分量，接下来的第二个平面存储所有的U分量，最后的一个平面存储所有的V分量（注！也可能是先存储V分量再存储U分量），一般可以简称为 **YUV 4XX P** 格式，例如应用非常广泛的I420是属于YUV420P的一种存储格式。
- **打包格式**：packed格式也叫做打包格式，顾名思义这种格式只有一个plane，存储规则是取n个采样点的Y、U、V的分量一起打包存储，然后以相同的方式存储接下来的n个采样点；对于n的取值和Y、U、V分量的排列顺序，也存在多种子组合格式，比较常用的格式有YUY2等。
- **半平面格式**：这种格式介于平面格式和打包格式之间，也叫做semi planar格式，对于 **YUV 4XX SP** 格式都是属于semi planar格式，拥有两个plane，第一个平面都是先存储所有的Y分量，接下来的U和V分量按照一定的规则交叉存储在接下来的平面上面，U、V的存储顺序的不同对应不同的子格式，也就是我们经常会看到NV21、NV12等。

## 常用的YUV格式

对于任何一个格式的分析，一般需要从以下几个维度进行区分：

- 1、分析采样格式，确认其属于YUV420、YUV444、YUV422等等的哪一个采样格式。
- 2、确定好采样格式后，接下来分析其打包格式，例如是否是平面、半平面、或者打包格式。
- 3、在确定打包格式的情况下，对于半平面/平面格式，我们需要关注UV的排序规则，对于打

包格式就需要注意采样点个数和YUV的储存顺序。

通过上面三个步骤基本就可以确定一个具体的YUV格式，下面提供一些常用的格式进行分析。

## I420格式

I420数据模型图如下：



I420（属于YUV420P）

按照数组存储如下所示：



数组存储排序

基本都是以一快4x2像素进行分析，也就是上图中红色部分，Y1/Y2/Y9/Y10共用同一个U1/V1色度对，介绍上文的知识点可以判断出其属于YUV420采样格式。

然后从平面上面看，U和V的分量分别存储在两个平面，其也就是属于平面格式YUV420P。对于任何平面格式，首个平面都是存储Y分量；对于第二个平面存储U以及最后一个平面存储V，那么可以推断出其就是I420格式。

## NV12格式

NV12数据模型图如下：



同样的，取一块4x2的像素区域来分析，Y1/Y2/Y9/Y10共用同一个U1/V1对，所以其属于YUV420采样格式。

这种格式的数据存在两个平面，也就是传说中的半平面(semi planar)格式，山寨点的可以表示为YUV420SP。

接下来我们需要关注第二个平面的内容，因为先U后V，按照U1V1U2V2这样排列存储，推断格式就是NV12格式。

YUY2格式

YUY2数据模型图如下：



继续对一块4x2的像素区域进行分析，我们发现Y1/Y2共用同一个U1/V1色度对，Y5/Y6共用U3/V3这另外一个色度对，所以可以得出属于YUV422格式。

由于其仅存在一个平面，毫无意外就是打包(packed)格式，可以表示为YUV422 packed。

接下来可以通过分析Y/U/V的排序规则来得出其子格式，其是以n=2两个像素为一组，按照Y1/U1/Y2/V1来排序，按照YUV标准其应该是YUY2格式。

I444格式

I444数据模型图如下：





I444 (属于YUV444P)

按照惯例，一块4x2的像素区域是我们所需要关注的，很容易发现任意一个Y都对应单独的一个U/V色度对，这也是YUV444的采样情况。

由于其存在三个平面，也就是YUV444P格式，根据在ffmpeg源码的分析，YUV444P有且只有这一种格式，也就是第二个平面存放所有U分量，第三个平面存放所有V分量，这种格式也叫做I444格式。

由于YUV的格式太多不方便全部列出来分析，但是只要掌握：采样格式、储存平面、UV排序规则，基本就是掌握绝大部分的YUV格式的。

更多格式可以参考下面资料：

<http://www.fourcc.org/yuv.php>

<https://github.com/FFmpeg/FFmpeg>

<https://wiki.videolan.org/YUV>

## 对齐问题和取值范围

接下来说一下两个比较偏的知识点。

### 对齐问题

不知道大家有没有想过，YUV中的J我们一般都是取4，那么问题来了，对于采样的单个范围就是4x2像素区域，例如当图片的宽度为7个像素时（换句话说就是非4对齐），那么有问题如下：

YUV	YUV	YUV	YUV	YUV	YUV	YUV	?
YUV	YUV	YUV	YUV	YUV	YUV	YUV	?

对齐问题

我们会发现最后一个黄色区域凑不齐4x2的采样单元区域。

一般来说，当以8bit存储精度来存放分量时，我们需要对其进行16bit或者32bit对齐，补齐的位置以0为初始值，如果不对齐呢？可能你的编码器、渲染器就莫名其妙的不能正常工作了。

当然这个在我们对YUV计算合适存储缓存区大小时也需要关注对齐问题。

### 取值范围

做过iOS采集的同学可能会发现，在iOS平台利用Camera采集YUV有ITU601和ITU709两种标准，而ITU601还细分两种Rang，一种取值范围是[0, 255]，另外一种取值范围是[16, 235]，当然这里所需要关注的关键点其实是GRB颜色空间转换的矩阵其实是不同的，具体矩阵这里就不列出来了。

## 结语

这篇文章从YUV的成像原理、与RGB的转换关系、采样规律、存储方式等方面进行深度解析，如果看完这篇文章还不懂YUV的话，麻烦拿刀来请教我啊。

本文发布于[简书](#)

End!

 10人点赞 >



 格式剖析



更多精彩内容，就在简书APP



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



**码农叔叔** 小码农，曾混迹过华为、酷狗、搜狐等公司。  
总资产24 共写了2.9W字 获得187个赞 共162个粉丝

关注

11月17日高爆打金服刚开1S，装备满地爆，神装秒回收！



高爆打金服·效卓

广告

写下你的评论...

全部评论 4 [只看作者](#)

[按时间倒序](#) [按时间正序](#)



Poseidong

2楼 2019.08.29 09:12

问一下，RGB和YUV的转换公式，为什么要  $\pm 128$  呢？

码农叔叔 作者

2019.08.30 17:06

因为UV这些色度记录的是对颜色的偏移值，所以取中间值也就是意味着没有偏移，对于[0.0, 1.0]的默认值是0.5f，对于[0,255]当然取128了，而计算矩阵都是按照这个转换关系推导出来的。



brnv

01.04 10:31

@码农叔叔 图上少抄一行 [0,128,128] 吧

<https://blog.csdn.net/hongszh/article/details/104354231>©



还有1条评论，[查看更多](#)

## 被以下专题收入，发现更多相似内容



iOS



音视频，多媒体...



FFmpeg iOS-FFm...

## 推荐阅读

[更多精彩内容 >](#)

### Android音视频—YUV格式深入浅出

文章参考：图文详解YUV420数据格式 YUV主要采样格式理解 YUV格式详解 百度百科和维基百科 概述 本文基...



Felix\_lin 阅读 2,373 评论 0 赞 5

### iOS视频开发（四）：通俗理解YUV数据

前言 系列文章：《iOS视频开发（一）：视频采集》《iOS视频开发（二）：视频H264硬编码》《iOS视频开发（三）...



GenoChen 阅读 6,681 评论 6 赞 20

$$\begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix}$$

新区20倍爆率，一秒999刀，上班挂机，下班收米！

### 视音频数据处理入门：RGB、YUV像素数据处理

前一阵子在梳理以前文章的时候，发现自己虽然总结了各种视音频应用程序，却还缺少一个适合无视音频背景人员学习的“最基础...

 视频音频小白 阅读 1,728 评论 1 赞 3



### YUV格式小结

导语 今天跟大家分享的这篇文章，也是之前自己总结的，大致就是想说明一下，视频的裸数据yuv格式的各种分类。刚开始接...

 bigonelby 阅读 7,666 评论 0 赞 12

### 游泳的第一天

原创：千木梦梵 今日 我去游泳了 首先我很感谢我的小伙伴 因为我要上课的原因 等了我将近2个小时 才跟我一起去游泳...

 千木梦梵 阅读 162 评论 0 赞 1

### 安吉拉姑娘

一大早就听到了安吉拉姑娘的好消息，“我跟XX公司谈妥了，三月底就离职啦！”我一直都知道她为自己的事业规划，在考虑跳...

 豆豆掌门 阅读 240 评论 0 赞 0



### 云南之旅2

飞机飞行三个小时抵达昆明，一出舱门，寒风迎面袭来。因目的地是腾冲，下午一点半飞，因此在昆明机场候机三个小时，这期间...

 任亚 阅读 91 评论 0 赞 0



### 换季没衣服穿？拿好这10家好看又时髦的淘宝店！

你的钱就跟对象一样，不知道为什么，说没就没了 眼看夏天就要过去了，翻遍了整个衣柜才发现，又到找不到衣服出门的季节。...

 人间少数派 阅读 632 评论 0 赞 17



### 做事不要想太多

也许是遗传父母的原因，面对事情总是想得多做得少，难道自己就是传说中思想的巨人，行动的矮子？最近上班，明明很简单的事...

 卧马听风 阅读 374 评论 0 赞 0

