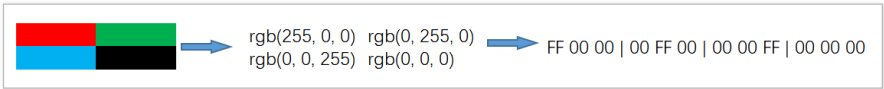


2.1 YUV 格式详解_物联网 / 嵌入式工程师 - 慕课网

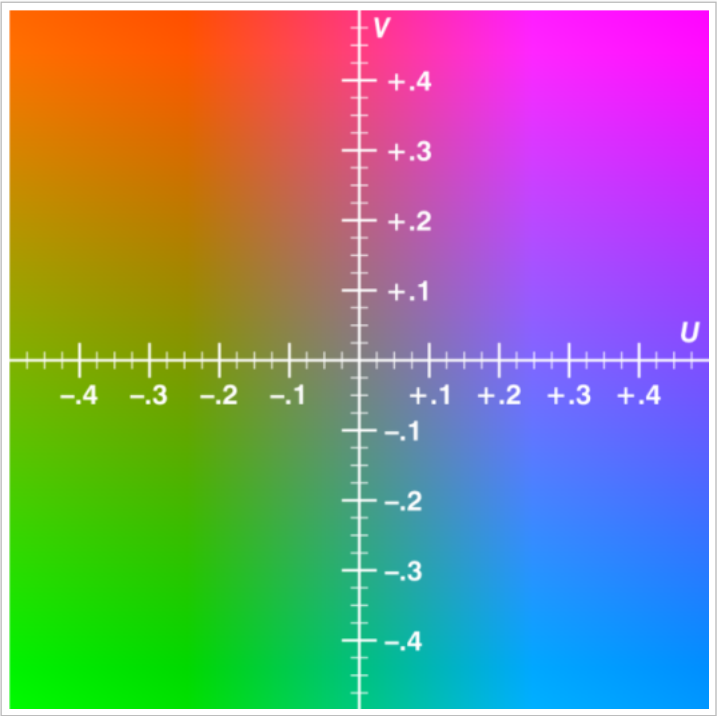
“ 慕课网慕课教程 2.1 YUV 格式详解涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

假设有一张图片的像素是 2x2，也就是说有 4 个像素点。以常见的 RGB 举例，将这 4 个像素点依次写入到文件中去如下：



那么为什么不让相机输出这样的 RGB 数据到文件，这样解析起来应该很方便才是啊？ 1) RGB 描述像素的数据量偏大，不利于传输和存储； 2) 几乎所有的相机都支持 YUV 的方式描述像素，而不一定支持 RGB 的方式； 3) RGB 数据不能适应光线昏暗且色差不明显的情形，也就是说 RGB 数据不支持亮度调节；

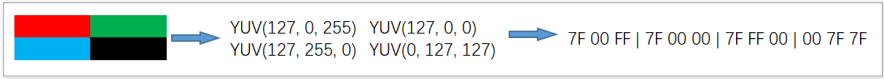
YUV 是指将亮度参量 Y 和色度参量 U/V 分开表示的像素格式，主要用于优化彩色视频信号的传输。也就是说每一个像素都对应一个亮度 Y 和两个色度 UV，当你拿到这 3 个值时，通过查表可以获得其确定的颜色。例如 YUV(127, -0.5, 0.5) 表示的最纯红色，而 YUV(255, -0.5, 0.5) 则表示最亮最纯的红色，它们的关系可以参照下图：



注意：

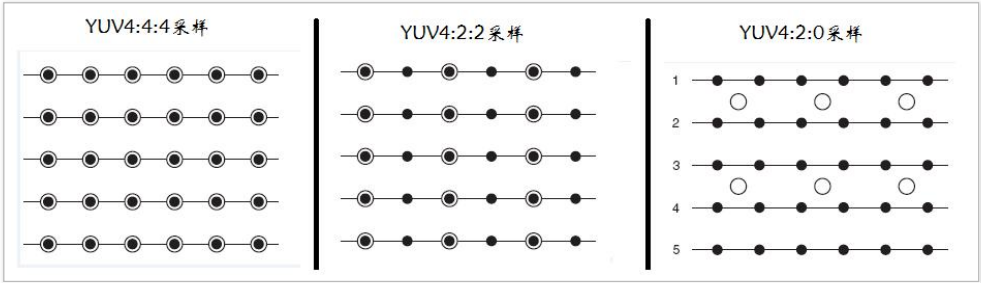
- 1) 这里坐标轴没有出现 Y 是因为 Y 从 0~255 变化时，同样的颜色会有不同的亮度； 2) 实际存储 UV 分量时，还是按照 0~255 来存储的，严格来说需要将 0~255 映射到 -0.5~0.5； 3) 由于 RGB 和 YUV 都是三个值，并且每个分量都是 0~255，这等价于三维空间，专业术语就把 RGB 和 YUV 等描述颜色的三维空间统称为颜色空间 (ColorSpace) ；

还是用上面 RGB 表示图片的例子，将 2x2 的图片用 YUV 表示出来可以是下面的结果：



看上去用 YUV 格式去描述图片和 RGB 描述没有太大的差别，那为什么相机都兼容 YUV？

相机在产生 YUV 像素数据时，并不总是每一个像素产生一组 YUV，很多时候是 2 个像素共用一组 UV 分量，这就必须说到 YUV4:4:4/YUV4:2:2/YUV4:2:0 这三种采样方式：



- 1) YUV 4:4:4 采样，一个 Y 对应一组 UV 分量，上图采样结果 7F 00 FF|7F 00 00|7F FF 00|00 7F 7F； 2) YUV 4:2:2 采样，两个 Y 共用一组 UV 分量，上图采样结果 7F 00 FF|7F|7F FF 00|00； 3) YUV 4:2:0 采样，四个 Y 共用一组 UV 分量，上图采样结果 7F 00 FF|7F|7F|00；

显然 YUV4:2:0 采样结果为 6Byte，相比于 RGB 的 12Byte，节省了几乎 50% 的存储和传输带宽，但是也牺牲了图像的细节；之所以可以这么干，是因为当像素足够多时，相邻 4 个像素的颜色差别是否明显并不会影响整体画质。

在 YUV 表示完所有的像素后，大量的像素 YUV 数据之间往往存在 UV 共用的情况，那么这时 UV 分量存放位置将影响图像解析效率，故通常的存储方式有两种： 1) planar 方式：将所有的 Y 分量连续存储，然后是 U 分量，然后是 V 分量； 2) packed 方式：将单个像素 YUV 分量连续存储，然后再存下一个像素的 YUV 分量；

YUV422planar 方式存储下来得 4*4 的图片如下：

Y1	Y2	Y3	Y4
Y5	Y6	Y7	Y8
Y9	Y10	Y11	Y12
Y13	Y14	Y15	Y16
U1	U2	U3	U4
U5	U6	U7	U8
V1	V2	V3	V4
V5	V6	V7	V8

YUV422packed 方式存储下来得 4*4 的图片如下：

Y1	U1	Y2	V1	Y3	U2	Y4	V2
Y5	U3	Y6	V3	Y7	U4	Y8	V4
Y9	U5	Y10	V5	Y11	U6	Y12	V6
Y13	U7	Y14	V7	Y15	U8	Y16	V8

问题： 如果一个 YUV 文件的像素格式是 Y1U1V1 Y2U2V2 ... 的形式，那么它是什么采样方式？什么存储方式