YCbCr是DVD、摄像机、数字电视等消费类视频产品中，常用的色彩编码方案。YCbCr 有时会称为 YCC.。Y'CbCr 在模拟分量视频（analog component video）中也常被称为 YPbPr。YCbCr不是一种[绝对色彩空间](http://baike.baidu.com/view/1928822.htm)，是YUV压缩和偏移的版本。

**YCbCr与YUV的区别**

正如几何上用坐标空间来描述坐标集合, 色彩空间用数学方式来描述颜色集合。常见的3 个基本色彩模型是[RGB](http://baike.baidu.com/view/17423.htm) , [CMYK](http://baike.baidu.com/view/48715.htm)和[YUV](http://baike.baidu.com/view/189685.htm)。

YCbCr 则是在世界数字组织视频标准研制过程中作为ITU - R BT1601 建议的一部分, 其实是YUV经过缩放和偏移的翻版。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。在YUV 家族中, YCbCr 是在[计算机系统](http://baike.baidu.com/view/1130583.htm)中应用最多的成员, 其应用领域很广泛,[JPEG](http://baike.baidu.com/view/7679.htm)、[MPEG](http://baike.baidu.com/view/7689.htm)均采用此格式。一般人们所讲的YUV大多是指YCbCr。YCbCr 有许多取样格式, 如4∶4∶4 , 4∶2∶2 , 4∶1∶1 和4∶2∶0。

**YUV**

[视频编解码器](http://baike.baidu.com/view/89060.htm)功能

视频编码器要求YUV4:2:0格式的视频输入，因此可能根据应用需要进行视频输入的预处理，即对YUV4:2:2隔行扫描(例如从摄像机)到YUV 4:2:0非隔行扫描转换，仅抽取但不过滤UV分。对视频解码器而言，还需要进行后处理，以将解码的YUV 4:2:0数据转换为RGB进行显示，包括：YUV 4:2:0到RGB转换；16位或12位RGB显示格式；0到90度旋转，实现横向或纵向显示。此外，视频编解码器通常还要求具有以下功能和特性：

支持[MPEG-4](http://baike.baidu.com/view/7754.htm)简单类 0、1 与 2 级；

兼容H.263与 MPEG-4 编解码标准；

MPEG-4视频解码器支持的可选项有：AC/DC预测、可逆可变长度编码(RVLC)、再同步标志(RM)、[数据分割](http://baike.baidu.com/view/4466818.htm)(DP)、错误隐藏专利技术、支持每个宏块4个运动矢量(4MV)、自由运动补偿、解码VOS层；

MPEG-4视频编码器选项有：RVLC、RM、DP、支持每个宏块4个运动矢量(4MV)、报头扩展码、支持编码期间码率改变、支持编码期间编码帧率改变、插入或不插入可视对象序列起始码；

支持编码期间序列中插入I帧；

支持编码器自适应帧内刷新(AIR)；

支持多编解码器，可用相同代码运行多个编解码器实例。

**YCbCr**

YCbCr其中Y是指亮度分量，Cb指蓝色色度分量，而Cr指红色色度分量。人的肉眼对视频的Y分量更敏感，因此在通过对色度分量进行子采样来减少色度分量后，肉眼将察觉不到的图像质量的变化。主要的子采样格式有 YCbCr 4:2:0、YCbCr 4:2:2 和 YCbCr 4:4:4。

4:2:0表示每4个像素有4个亮度分量，2个色度分量 (YYYYCbCr)，仅采样奇数扫描线，是便携式视频设备(MPEG-4)以及电视会议(H.263)最常用格式；4：2：2表示每4个像素有4个亮度分量，4个色度分量(YYYYCbCrCbCr)，是DVD、数字电视、HDTV 以及其它消费类视频设备的最常用格式；4：4：4表示全像素[点阵](http://baike.baidu.com/view/545607.htm)(YYYYCbCrCbCrCbCrCbCr)，用于高质量视频应用、演播室以及专业视频产品。

[**编辑本段**](http://baike.baidu.com/view/564370.htm)**YUV主要的采样格式**

主要的采样格式有YCbCr 4:2:0、YCbCr 4:2:2、YCbCr 4:1:1和 YCbCr 4:4:4。其中YCbCr 4:1:1 比较常用，其含义为：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 bits（如下图第一个图）. 而现在仅需要 8+(8/4)+(8/4)=12bites, 平均每个点占12bites(如下图第二个图)。这样就把图像的数据压缩了一半。

上边仅给出了理论上的示例，在实际数据存储中是有可能是不同的，下面给出几种具体的存储形式：

**（1） YUV 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

**（2） YUV 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8比特量化的图像来说，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。

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]

**（3） YUV 4:1:1**

4:1:1的色度抽样，是在水平方向上对色度进行4:1抽样。对于低端用户和消费类产品这仍然是可以接受的。对非压缩的8比特量化的视频来说，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

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）YUV4:2:0**

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8比特量化的视频来说，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。

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] [Y7U7 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] [Y7U2 V7] [Y8 U2 V7]

**YCbCr与RGB的相互转换**

Y=0.299R+0.587G+0.114B

Cb=0.564(B-Y)

Cr=0.713(R-Y)

R=Y+1.402Cr

G=Y-0.344Cb-0.714Cr

B=Y+1.772Cb

**YCbCr的模拟分量部分YPbPr**

YCbCr 有时会称为YCC. Y'CbCr 在模拟分量视频（analog component video）中也常被称为YPbPr。

Y'CbCr讯号(prior to scaling and offsets to place the signals into digital form)被称为[YPbPr](http://baike.baidu.com/view/1745158.htm) ，而且产生时必须合乎gamma-adjusted RGB source有二个不同的常数Kb和Kr如下： YPbPr (analog version of Y'CbCr) from R'G'B' YPbPr (analog version of Y'CbCr) from R'G'B' ==================================================== ================================================== ==

Y' = Kr \* R' + (1 - Kr - Kb) \* G' + Kb \* B' Y'

Pb = 0.5 \* (B' - Y') / (1 - Kb) Pb = 0.5 \* (B' - Y') / (1 - Kb)

Pr = 0.5 \* (R' - Y') / (1 - Kr) Pr = 0.5 \* (R' - Y') / (1 - Kr)

.................................................... .................................................. ..

R', G', B' in [0; 1] R', G', B' in [0; 1]

Y' in [0; 1] Y' in [0; 1]

Pb in [-0.5; 0.5] Pb in [-0.5; 0.5]

Pr in [-0.5; 0.5] Pr in [-0.5; 0.5]