

Source from <http://gentlelogic.blogspot.co.uk/2011/11/exploring-h264-part-1-color-models.html>  
Compiled by <http://guoh.org/lifelog/> Thursday, October 24 2013  
'Cause some known reason, this post can not be accessed in the Chinese mainland, so this copy is generated, please obey the original copyright while using this article. If possible, please visit the original website.

WEDNESDAY, NOVEMBER 9, 2011

## Exploring H.264. Part 1: Color models

Currently I'm doing some research work related with H.264 video signal decoding. So I decided to dedicate my first post to color models. Why? Because if we set aside all unnecessary (for now) information – we will see, that decoding the video signal is reduced to the one problem: stripping out of the bitstream information about pixel colors...

What is digital signal? Like everything in the digital world – it is the sequence of bits. It can be stored, copied, transmitted or compressed.

Video is the set of pictures, and every picture consists of pixels. And every single pixel has its own color and/or luminance. This values stored in digital form. Let's see, how we can store color values. This is where **color model** (color space) comes to help. What is color model? It is a mathematical model that helps to represent colors as a list of numbers. Let's start with most popular one – **RGB** model.

### 1. RGB color model.

Look at the picture below. This is well known RGB additive color model with three base colors.

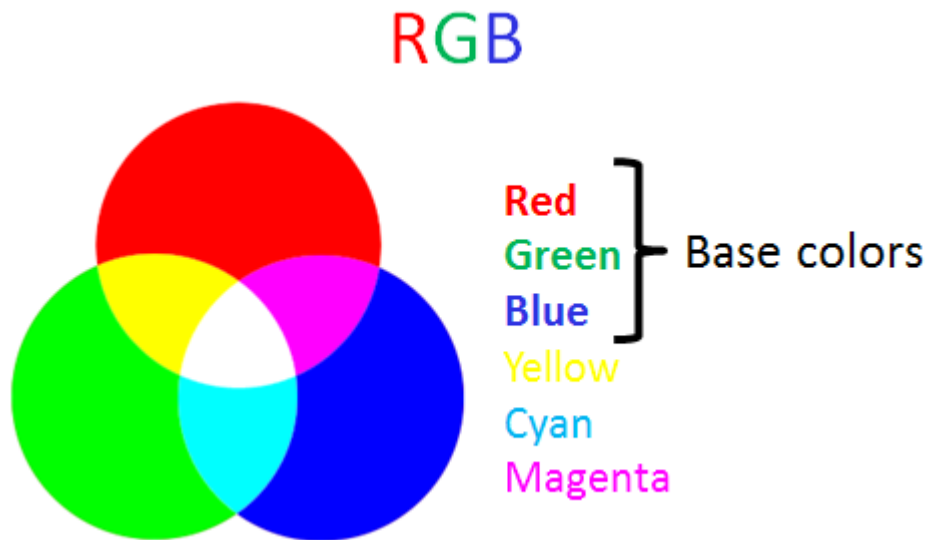


Figure 1. RGB color model

Any color can be represented in three digits. For example (255, 0, 0) stands for Red color. Three colors – three numbers. Pretty clear, but not very practical: we need to know all three numbers in order to reconstruct the color. We definitely need more efficient model and such model exists.

## 2. YCbCr color model.

So, how to represent color in numbers more effectively? Let's assume that we have color **luminance** which actually represents the brightness of the color. Now we can express luminance, using RGB:

$$Y = k_r R + k_g G + k_b B,$$

where Y – luminance (luma),

k – weighting coefficients (described in ITU-R BT.601-7 recommendation)

And now it is possible to calculate chrominance (color difference) which represents the color information:

$$C_r = R - Y,$$

$$C_g = G - Y,$$

$$C_b = B - Y$$

Now we can grab those weighting coefficients from ITU-R BT.601-7 recommendation, skip some math, and we get the following:

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

$$C_b = 0.564(B - Y),$$

$$C_r = 0.713(R - Y)$$

$$R = Y + 1.402C_r,$$

$$B = Y + 1.772C_b$$

Note how we can represent green:

$$G = Y - 0.344C_b - 0.714C_r$$

Now we can represent any color with one luma and two chroma components.

So, what do we have? To represent a color in YCbCr model we need 3 numbers. To represent same color in RGB model we need... wait a second – 3 numbers. But why YCbCr should be more efficient?

To answer this question let's split our video into pictures and examine a single picture. The picture consists of pixels. Like this one:

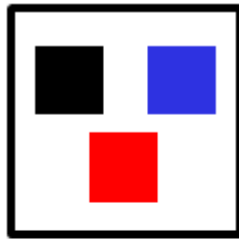


Figure 2. Pixel

And the picture as a set of pixels looks like this:

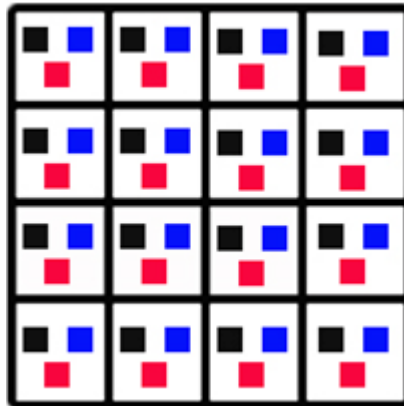


Figure 3. 4:4:4 subsampling

And here is the trick: it is possible to sacrifice some chroma components without visible quality loss. For example, this is how we can represent our image:

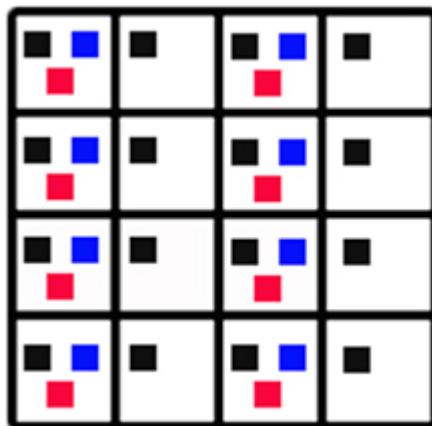


Figure 4. 4:2:2 subsampling

Each pixel in a row has its luma, and chroma components are discarded for every odd pixel in a row. Still we have

our quality with a less amount of data. We can even do something like this:

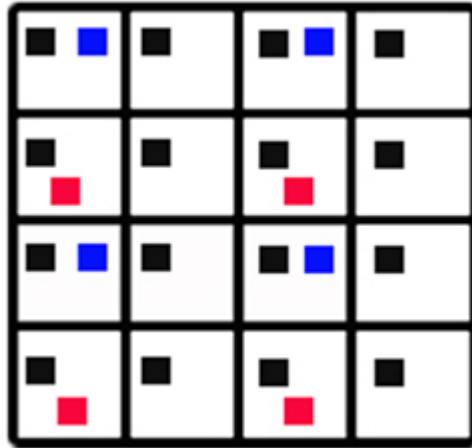


Figure 5. 4:2:0 subsampling

This one is really low cost subsampling and still the quality is fine. The size of picture can be greatly reduced without significant loss of quality.

It is all for now. Feel free to comment or ask any questions.

In my next topic I'll go deeper in video decoding process and describe H264 bitstream format.

## 4 comments:

**Suchitradaemon** June 28, 2012 at 2:07 PM

thnx a lot for this simplification. Wherever I'd read there'd just be the explanation that our eyes are less sensitive to Cr and Cb than Cg. But this explains in a better way with all the pictures.

**Denis M.** June 29, 2012 at 3:50 AM

I wrote this article during the research of H264 bitstream format, just to understand the color models. I'm glad it helped someone except me. Good luck :)

**Unknown** May 4, 2013 at 4:02 AM

how can i known my image is in which format??

**Denis M.** May 27, 2013 at 7:02 AM

It is hard to provide good answer because of lack of the information in your question, but usually if you have raw image bitstream you can analyze its header (like first four bytes) - information about image format will be there.