

## The RGB code

### Part 2: Cracking the RGB code (from XYZ to RGB, and other codes...)

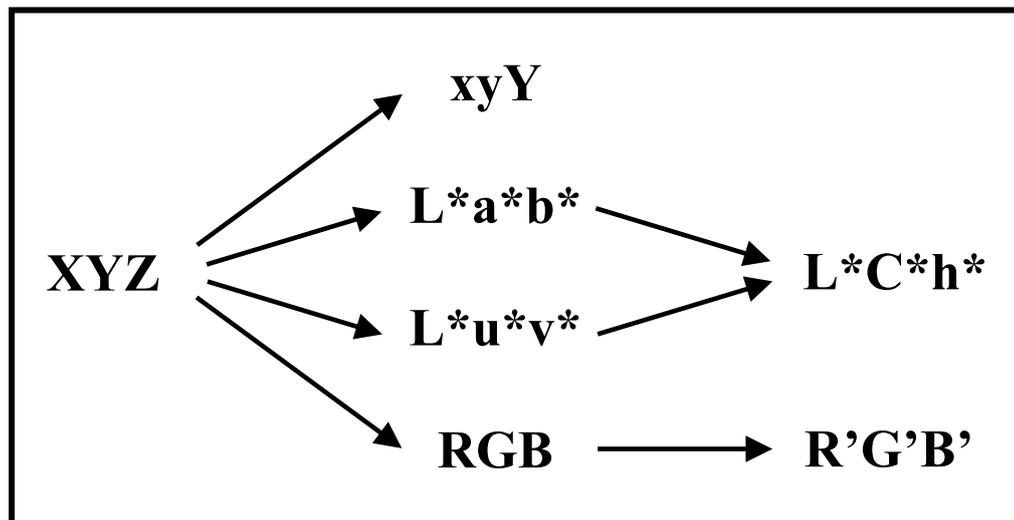
In the first part of his quest to crack the RGB code, our hero saw how to get XYZ numbers by combining a *Standard Observer* (a mathematical representation of how human perceive color) with *Illuminant* data (the light used to see a color).

---

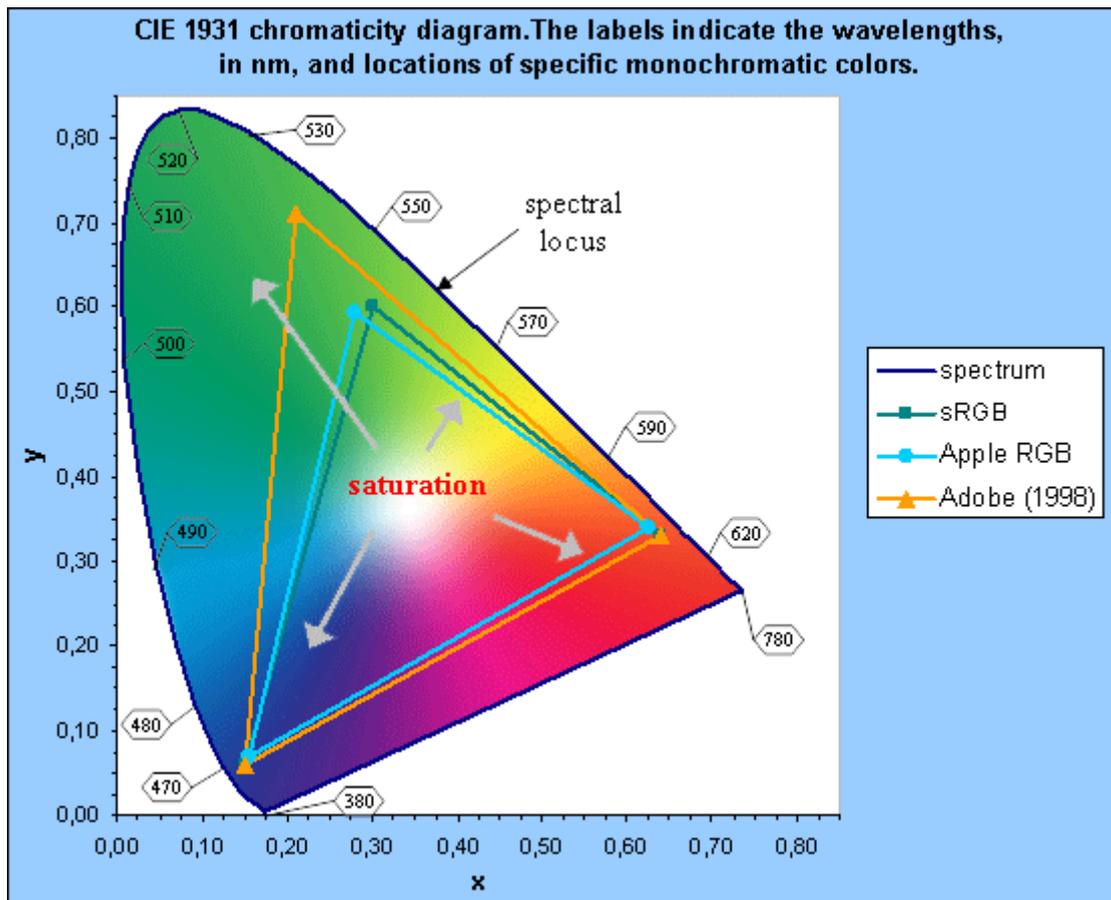
*The image still held its secret and a few more decoding steps were required before he could see it as it was brought into the light many hundred days ago...*

---

Starting with the XYZ data, there is an assortment of mathematical transformations that are available to obtain various alternate color notations:  $xyY$ ,  $L^*a^*b^*$ ,  $L^*u^*v^*$ ,  $L^*C^*h^*$ , and even RGB, as shown in the next Figure. At first sight, just one encoding step seems to be required between a “color” and its RGB notation; but beware, the final solution is still a step further!

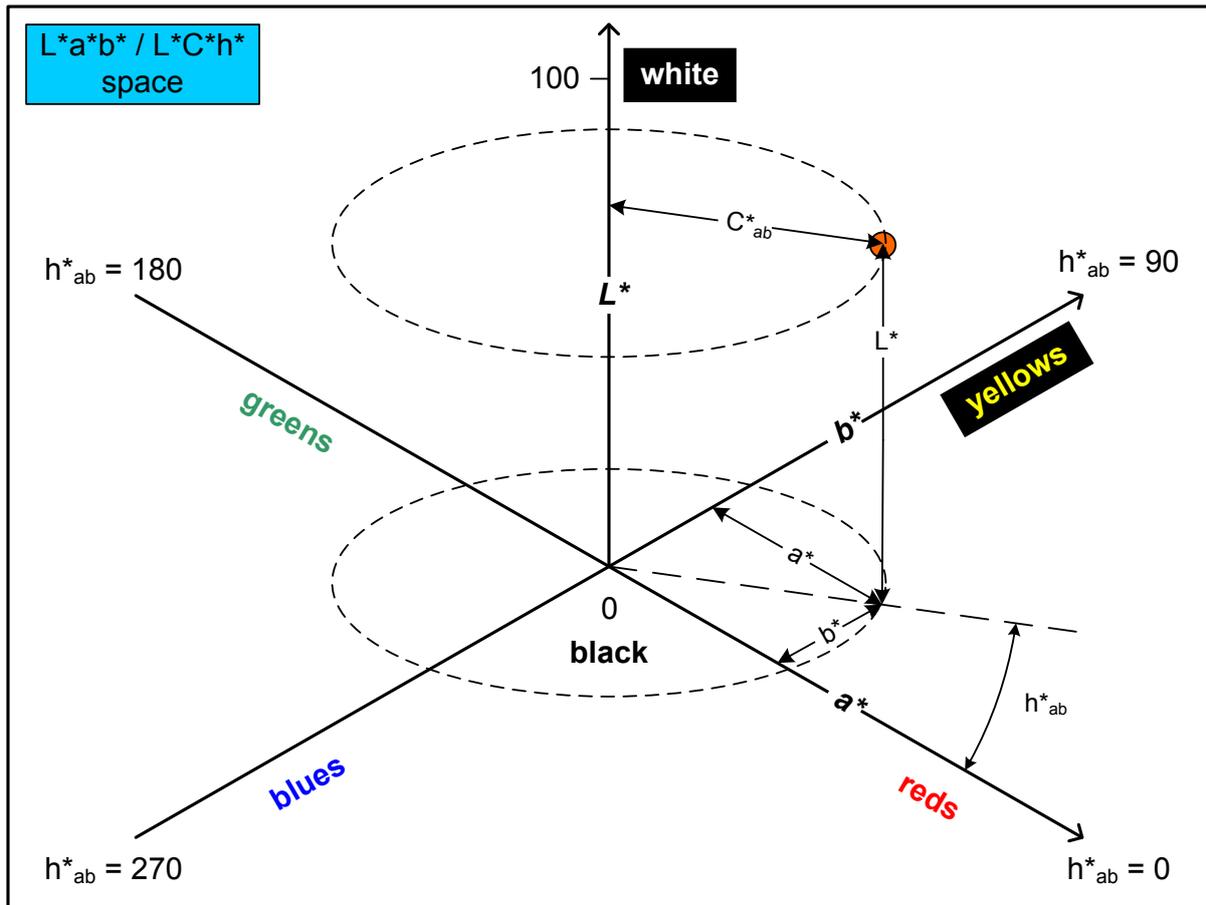


These notations are all equivalent but they present the information in complementary fashions. XYZ values are often seen in numerical form but seldom, if ever, presented graphically; the derived “xy” coordinates are, on the other hand, often presented in diagrams. The xy of the xyY notation are normalized values, between zero and one, of their XY counterparts; they are represented in the well known “horseshoe” diagram shown below (officially called a chromaticity diagram). The region within the horseshoe itself represents all the visible colors. The colors located on the horseshoe’s periphery, called the spectral locus, are pure, monochromatic, saturated colors. The more one goes toward its center, the less saturated a color becomes (i.e. it becomes whitish, greyish or blackish). All Illuminants of interest for practical uses are located in the center region (about where the word **saturation** appears in the diagram below). The “z” of xyz is seldom shown since, as a result of normalization,  $x+y+z=1$ . However, since, by normalizing, the actual luminance (Y) is lost, it is added as the third coordinate instead of “z” when communicating color data (i.e. we use xyY instead of xyz).



The xy diagram is very useful in presenting the relative positions of colors and can also be used to see what colors will result from mixing two others (for additive colors only, such as a computer screen, not for subtractive mixes like paint colors). However, this color space, as with XYZ, is not perceptually uniform, and a given separation between two colors on the diagram corresponds to different perceived differences, depending on their relative positions.

$L^*a^*b^*$  and  $L^*u^*v^*$  are attempts at making the xyY space, in fact the XYZ space to be more specific, more perceptually uniform.  $L^*$  is the *lightness*, the intensity of the reflected light. Where “Y” represented the actual reflected energy,  $L^*$  represents the **perceived** brightness, as the eye is not a linear sensor (twice the energy corresponds to less than twice the perceived brightness).  $a^*$  is the amount of **red** (+ axis) or **green** (- axis) while  $b^*$  is the amount of **yellow** (+ axis) or **blue** (- axis);  $u^*$  and  $v^*$  have similar meanings. Red/green and blue/yellow are called “opponent” colors; you cannot perceive blue when there is yellow and vice-versa. Similarly, you cannot perceive red when there is green. Representing the colors in such a way is based on our understanding of how our eyes and brain process colors.



Each notation,  $L^*a^*b^*$  and  $L^*u^*v^*$ , has its pros and cons, with different scaling factors for each variable, except  $L^*$  which is the same for both, but, in essence,  $L^*a^*b^*$  is often preferred for printed color applications and  $L^*u^*v^*$  for applications geared to electronic displays (especially in the UK).

$L^*C^*h^*$  is yet another transformation of either  $L^*a^*b^*$  or  $L^*u^*v^*$  where the data is looked at from the perspective of cylindrical coordinates.  $C^*$ , for *Chroma*, or color saturation, a vector obtained by combining  $a^*$  and  $b^*$ , is the extent of the cylinder radius.  $h^*$ , the *hue angle*, is the angle of the *Chroma* vector.  $L^*$ , the height of the *Chroma* vector on the cylinder, is the same coordinate used in the  $L^*a^*b^*$  notation.

Finally, starting with XYZ, you can also obtain RGB coordinates, those infamous coordinates we are looking for. But what are those R'G'B' symbols seen beside R, G, and B? What is the difference between the two representations?

The conversion to “RGB” involves two steps. The first step, obtaining **linear** RGB values, requires the selection of the *primaries*, i.e. which colors correspond to pure **R**, **G**, and **B**, as well as the selection of an *Illuminant*. Thus, RGB spaces cannot be dissociated from their *primaries* and their *Illuminant*.

In the xy chromaticity diagram above, we see the regions defined by the primaries of the Adobe (1998), Apple RGB and sRGB spaces. The Adobe (1998) space is visibly bigger than the Apple RGB and sRGB spaces and we can safely assume that it can represent more of the

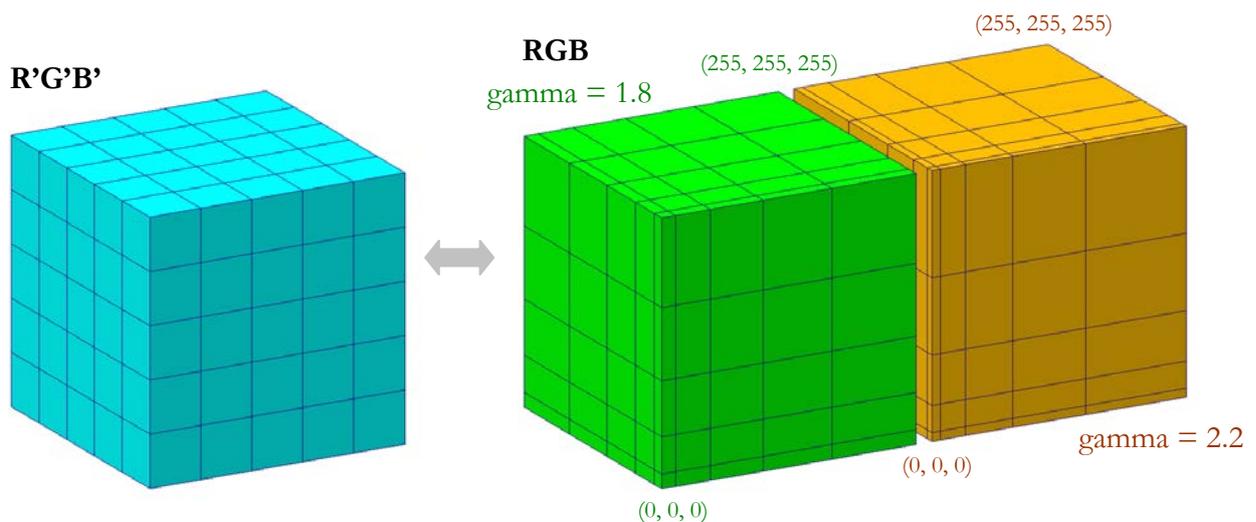
colors humans can see, the visible gamut. However, we have to be careful when comparing the Apple RGB and sRGB spaces; the xy chromaticity diagram cannot be used to compare spaces of slightly different sizes, by simply comparing their areas, and more complex calculations are required. Let's just say that Apple RGB and sRGB are somewhat equivalent and that everybody that tells you one is bigger or better than the other one is right!

The *primaries* **and** the *Illuminant* are used to find a conversion matrix between XYZ and RGB (a 3 by 3 matrix, 3 rows x 3 columns). A reverse RGB to XYZ matrix can also be derived.

The second and last step, which scales the RGB values to match how we perceive brightness, in a **non-linear** fashion, in a process similar to what is done between the **Y** of XYZ and the **L\*** of  $L^*a^*b^*$ ; it is called **gamma correction**. Many RGB spaces use a single number to express gamma, such as 1.8 for Apple RGB and 2.2 for Adobe RGB, but you should know that certain spaces, such as sRGB and NTSC, have been defined with a more complex mathematical expression for it. Using a single number gamma when a complex expression is available usually produces an unnoticeable error, for most applications, but it can become significant for critical applications, especially for low brightness (darker) colors.

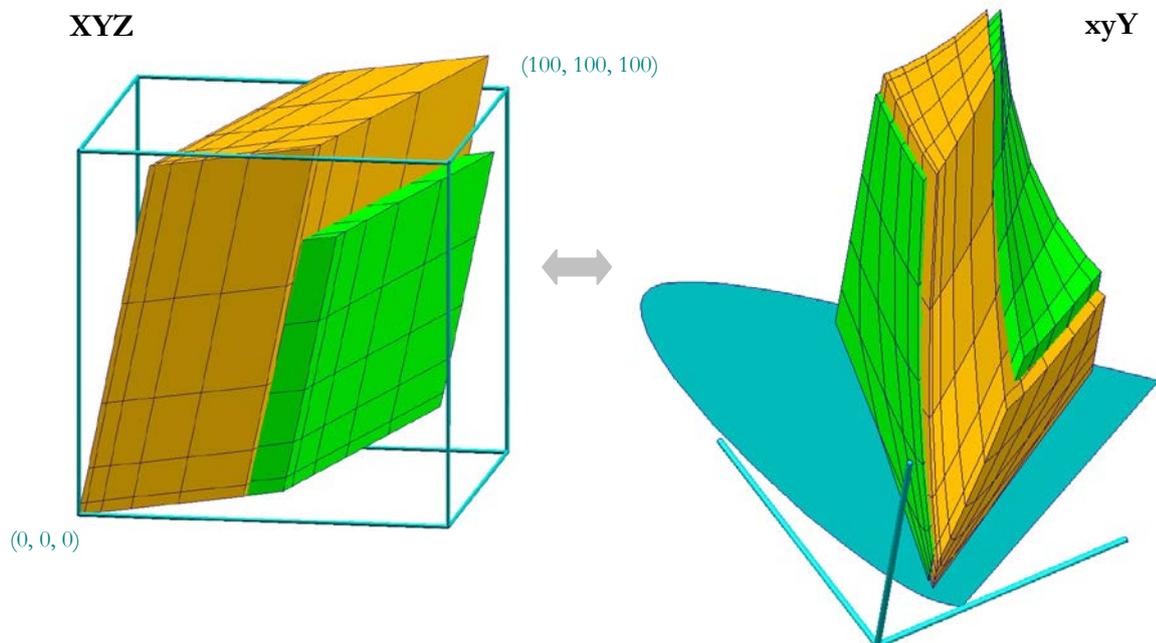
The blue cube on the left of the following image shows an R'G'B' space separated in 5 equal size zones for each coordinate, for a total of 125 identically sized smaller cubes (i.e. cubelets). The green and orange cubes on the right show how the blue cube is transformed in linear RGB. The green cube corresponds to a gamma of 1.8, such as used for ColorMatch or Apple RGB, and the orange one to a gamma of 2.2, such as used for Adobe (1998) (and a good approximation of the complex gamma defined for sRGB).

We see that as many R'G'B' combinations are assigned to the very small cubelet located close to RGB = (0, 0, 0), in either the green or orange cubes, than to the largest cubelet located at RGB = (255, 255, 255). Seen in another angle, more R'G'B' coordinates (attention: NOT RGB) are assigned to low brightness, darker, colors than to high brightness ones.



The gamma corrected RGB values are labelled as R'G'B' to make a distinction with the linear RGB data. It is now time to tell you that when you select a color using RGB notation in most, if not all, computer programs, you are selecting R'G'B' (gamma corrected) numbers. The fact that everybody uses RGB instead of R'G'B' is yet another sign of the old plot to hide the real meaning of RGB code (of course, the previous statement is oblivious to the fact that it is also a simpler way of identifying the coordinates!).

While we are going backward from R'G'B' to RGB, why not go all the way back to xyY? The following figure shows the corresponding XYZ coordinates of the two RGB cubes shown above. However, because we are dealing with XYZ data, we need to assign an *Illuminant* and *primaries*. The green data is assigned *Illuminant* D50 and the primaries of ColorMatch, while the orange data is assigned *Illuminant* D65 and the primaries of sRGB. The blue wire frame, outlining a cube, is just there to give a sense of scale for the data. Its sides are 100 "units" of X, Y, or Z. The tip of the orange shape, which extends over the wire frame, shows the maximum "Z" value of the D65 *Illuminant* (= 108.9 for the math maniacs among you).



On the xyY diagram we see that many colors are packed near the edges of the RGB spaces triangles, showing the non-uniformity of the xy representation and the danger of using it for space size comparison purposes.

### **The truth on the RGB code!**

As no two RGB spaces have the same primaries, and often different gamma corrections, the result is that identical RGB coordinates can basically describe ANY color, unless the space characteristics are known.

If you recall what started this quest, an image which colors looked odd, lifeless, with non-saturated colors, when plainly shown on a display, you may now infer that we could restore its looks by assigning its RGB numbers to the original RGB space in which they were conceived.

---

*Looking once again at his summer vacation image, our hero had a flash (Even though he did not use it for this particular picture!). The answer to his problem may be in the image itself. It was a nice summer indeed, it was during this vacation that he met the one that would become his companion, and that he took the picture of her he was looking at. The fact that they had common interests in photography was certainly for something in their relationship, and he particularly knew that she was quite versed in digital camera lingo.*

*Asking her about it, she recalled that they had assigned the Adobe (1998) as the working space for his digital camera. This particular image was cropped and edited a bit, before being resaved as a different format, with no embedded color space profile. Because he now knew the profile name, he only needed to manually associate it with the image in his graphics editing program. By assigning the Adobe (1998) profile to his image, its original colors were restored.*

*Our heroes (they found the solution as a team!) were so enthused of breaking the RGB code that they decided to write a book about it. (To be continued...)*

---

In many graphic editing programs, such as Photoshop, Paint Shop Pro, and the Corel suite, you can assign a color space profile (called ICC profile) to an image. The key to the code was simple indeed; understanding what it meant, was not.

The trek to RGB (I meant R'G'B' !) may have seemed long and tortuous, but it has shown us the light (!), so that we too can understand the hidden messages in our images. Well, it may not be that esoteric but at least it should provide you some hints of what is going on behind the scene when dealing with RGB images, and what is happening when assigning and converting color profiles.

The last part of this series will show how we can juggle and use the various color notations and color encodings that we saw so far.

### **About the author**

Danny Pascale: After many years of research and development and project management work in the academic and industrial sectors, he now does technology assessment and helps companies bring new products into the market (a consultant if you prefer!) in the computer and consumer electronics industries. He recently formed a new company, BabelColor, dedicated to the development of colorimetric software tools.

To communicate with the author: [dpascale@BabelColor.com](mailto:dpascale@BabelColor.com)

The BabelColor Web site: [www.BabelColor.com](http://www.BabelColor.com)