

Color notes

A color pixel can be represented in terms of three numbers. There are many representations. Some are discussed here.

1 Device independent: the XYZ system

In the XYZ system, each color pixel is described in terms of the three coordinates X, Y, Z . These coordinates are positive, but not restricted otherwise. The value of Y is the luminance.

2 Device independent: the xy system

To get pure color information, use the xy *chromaticity coordinates*, which are obtained by normalizing XYZ in the following way:

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}$$

Notice that $0 < x, y < 1$. These x, y coordinates are *pure* color. A color pixel can be represented in terms of its chromaticity and luminance, i.e., the values of x, y, Y .

$$X = \frac{x}{y}Y, \quad Y = Y, \quad Z = \frac{1 - x - y}{y}Y$$

3 Device dependent: linear RGB

It is possible to completely specify color in terms of 3 arbitrary (linearly independent in XYZ) colors. We refer to them as R,G,B, since they are usually chosen to be approximately red, approximately green and approximately blue. The linear-RGB values are typically in the $[0, 1]$ range.

Assume that the R,G,B colors are defined in terms of their XYZ values as:

$$R = \begin{pmatrix} X_R \\ Y_R \\ Z_R \end{pmatrix}, \quad G = \begin{pmatrix} X_G \\ Y_G \\ Z_G \end{pmatrix}, \quad B = \begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix}$$

then the XYZ values of a pixel specified by the three values (r, g, b) is:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = r \begin{pmatrix} X_R \\ Y_R \\ Z_R \end{pmatrix} + g \begin{pmatrix} X_G \\ Y_G \\ Z_G \end{pmatrix} + b \begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix}$$

Thus, the mapping of (r, g, b) coordinates into (X, Y, Z) coordinates is a simple matrix multiplication:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \begin{pmatrix} r \\ g \\ b \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} r \\ g \\ b \end{pmatrix} = M^{-1} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

where:

$$M = \begin{pmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{pmatrix}$$

In many situations the principal colors R,G,B are not specified in the XYZ coordinate system, but in the xy system. This by itself is not enough to construct the conversion matrix M . The additional information that is used is the xy values of the *white* color:

$$R = \begin{pmatrix} x_R \\ y_R \end{pmatrix}, \quad G = \begin{pmatrix} x_G \\ y_G \end{pmatrix}, \quad B = \begin{pmatrix} x_B \\ y_B \end{pmatrix}, \quad W = \begin{pmatrix} x_W \\ y_W \end{pmatrix}$$

To solve for the matrix M we use the transformation between the XYZ and the xy coordinates with the following additional information: W is to be represented in the (r, g, b) system by $(1, 1, 1)$. As we see it is also needed to arbitrarily fix Y_W , and the accepted choice is $Y_W = 1$. Substituting in M we see that:

$$\frac{Y_W}{y_W} \begin{pmatrix} x_W \\ y_W \\ 1 - x_W - y_W \end{pmatrix} = \frac{Y_R}{y_R} \begin{pmatrix} x_R \\ y_R \\ 1 - x_R - y_R \end{pmatrix} + \frac{Y_G}{y_G} \begin{pmatrix} x_G \\ y_G \\ 1 - x_G - y_G \end{pmatrix} + \frac{Y_B}{y_B} \begin{pmatrix} x_B \\ y_B \\ 1 - x_B - y_B \end{pmatrix}$$

This is a homogeneous system of 3 equations in 4 unknowns. Taking $Y_W = 1$ we can solve for Y_R, Y_G, Y_B and construct M .

A standard value of white is the D_{65} , which is supposed to represent *daylight*. It is also called illuminant white.

$$D_{65} : x = 0.3127, \quad y = 0.3291$$

Other choices for white can be:

$$D_{50} : \text{ambient white} \quad x = 0.3457, \quad y = 0.3585$$

In summary, to construct / display an image given pixel values in linear RGB format we need the values of the three principal colors in XYZ. Alternatively, we need their values in xy coordinates, and in addition the value of white.

4 Device dependent: nonlinear RGB

Nonlinear-RGB is obtained from linear-RGB by a nonlinear transform, that is usually called *gamma correction*. It has the following general form of mapping the value D to the value I :

$$I = A(k_1 D + k_2)^\gamma$$

A, k_1, k_2, γ are global constants. When applied to each coordinate of (R, G, B) , the result is the triplet (R', G', B') . In television sets one can typically change k_1 using the contrast knob, and k_2 using the brightness knob.

5 Device independent: sRGB

A particular choice of the device dependent RGB was proposed as a standard for a default color space by HP and Microsoft. The following primary colors are used (specified in the xy coordinate system):

$$R = \begin{pmatrix} 0.64 \\ 0.33 \end{pmatrix}, \quad G = \begin{pmatrix} 0.3 \\ 0.6 \end{pmatrix}, \quad B = \begin{pmatrix} 0.15 \\ 0.06 \end{pmatrix}, \quad W = \begin{pmatrix} 0.3127 \\ 0.3290 \end{pmatrix}$$

With the choice $Y_W = 1$ this gives the following conversion matrix from XYZ to linear sRGB:

$$\begin{pmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{pmatrix} = \begin{pmatrix} 3.240479 & -1.53715 & -0.498535 \\ -0.969256 & 1.875991 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

The resulting RGB values are expected to be in the $[0, 1]$ range. Out of range values are clipped.

The conversion of linear sRGB (in the $[0, 1]$ range) to nonlinear sRGB (in the $[0, 1]$ range) is according to the following gamma correction:

$$I = \begin{cases} 12.92 D & \text{if } D < 0.00304 \\ 1.055 D^{1/2.4} - 0.055 & \text{otherwise} \end{cases}$$

The final step is stretching the range to $[0, 255]$ by multiplying the nonlinear-sRGB values by 255.

The above steps show how to take a color image specified in XYZ and encode each pixel in 3 bytes. The inverse process takes an RGB picture, where each pixel is represented by 3 bytes, and converts it to XYZ.

Suppose the input pixel is (R_8, G_8, B_8) . These are numbers in the $[0, 255]$ range. Convert them to the three numbers (R', G', B') in the $[0, 1]$ range according to:

$$R' = R_8/255 \quad G' = G_8/255 \quad B' = B_8/255$$

Now convert (R', G', B') to (R, G, B) by:

$$R = \text{invgamma}(R') \quad G = \text{invgamma}(G') \quad B = \text{invgamma}(B')$$

where

$$\text{invgamma}(v) = \begin{cases} v/12.92 & \text{if } v < 0.00304 \\ (\frac{v+0.055}{1.055})^{2.4} & \text{otherwise} \end{cases}$$

Now convert (R, G, B) to (X, Y, Z) :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.35758 & 0.180423 \\ 0.212671 & 0.71516 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

6 Device independent: Luv

The representation Luv is an attempt to make the XYZ representation more perceptually uniform. (That is, small changes in values correspond to small changes in the perceived image.)

6.1 XYZ to Luv

To convert from XYZ to Luv we need to know the “white” values (X_W, Y_W, Z_W) . In particular, if D_{65} is used for white with $Y_W = 1$ we have: $(X_W, Y_W, Z_W) = (0.95, 1.0, 1.09)$.

To convert all the XYZ pixels of an image to Luv do the following: Start by computing the constants u_w, v_w defined as

$$u_w = \frac{4 X_W}{X_W + 15 Y_W + 3 Z_W} \quad v_w = \frac{9 Y_W}{X_W + 15 Y_W + 3 Z_W}$$

For each pixel X, Y, Z compute:

$$t = \frac{Y}{Y_W}$$

$$L = \begin{cases} 116 t^{1/3} - 16 & \text{if } t > 0.008856 \\ 903.3 t & \text{otherwise} \end{cases}$$

(Observe that the range of L is: $0 \leq L \leq 100$.)

$$d = X + 15 Y + 3 Z, \quad u' = \frac{4 X}{d}, \quad v' = \frac{9 Y}{d}$$

$$u = 13 L (u' - u_w), \quad v = 13 L (v' - v_w)$$

6.2 Luv to XYZ

With u_w, v_w as defined above compute:

$$u' = \frac{u + 13 u_w L}{13 L}, \quad v' = \frac{v + 13 v_w L}{13 L}$$

Given L , compute Y :

$$Y = \begin{cases} \left(\frac{L+16}{116}\right)^3 Y_W & \text{if } L > 7.9996 \\ \frac{L}{903.3} Y_W & \text{otherwise} \end{cases}$$

Given Y , compute X, Z : If $v' = 0$, $X = 0$, $Z = 0$. If not,

$$X = Y \cdot 2.25 \frac{u'}{v'}, \quad Z = \frac{Y (3 - 0.75 u' - 5 v')}{v'}$$