**1.4.1 The RGB Color Model**

The red-green-blue model is formed by a color cube $ \{(R,G,B):0\leq R,G,B\leq 1\}$.

|  |
| --- |
| \begin{figure}\begin{picture}(5,5)(-1,-1) \put(0,0){\makebox(0,0){$\bullet$}\lin... ...kebox(0,0){Black}} \put(1.5,2.2){\makebox(0,0){White}} \end{picture}\end{figure} |
| **Figure:** The RGB-cube |

Conversion from $ (R,G,B)$ to $ (X,Y,Z)$ is given via the chromaticities $ (X_r,Y_r,Z_r)$, $ (X_g,Y_g,Z_g)$ and $ (X_b,Y_b,Z_b)$ of the CRTs phosphors by matrix multiplication via:

$\displaystyle \begin{pmatrix}
X \\ Y \\ Z
\end{pmatrix}=
\begin{pmatrix}
Xr & X...
...Zr & Z_g & Z_b \\
\end{pmatrix}\cdot
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix}$

Let $ C_r:=X_r+Y_r+Z_r$. Then $ X_r=x_r\cdot C_r$, $ Y_r=y_r\cdot C_r$ and $ Z_r=z_r\cdot C_r=(1-x_r-y_r)\cdot C_r$.

This can be calculated from $ X=\frac{x}{y}Y$, $ Y=Y$, $ Z=\frac{1-x-y}{y}Y$.

**1.4.2 The CMY Color Model**

This stands for cyan-magenta-yellow and is used for hardcopy devices. In contrast to color on the monitor, the color in printing acts subtractive and not additive. A printed color that looks red absorbs the other two components $ G$ and $ B$ and reflects $ R$. Thus its (internal) color is G+B=CYAN. Similarly R+B=MAGENTA and R+G=YELLOW. Thus the C-M-Y coordinates are just the complements of the R-G-B coordinates:

$\displaystyle \begin{pmatrix}
C \\ M \\ Y
\end{pmatrix}=
\begin{pmatrix}
1 \\ 1 \\ 1
\end{pmatrix}-
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix}$

If we want to print a red looking color (i.e. with R-G-B coordinates (1,0,0)) we have to use C-M-Y values of (0,1,1). Note that $ M$ absorbs $ G$, similarly $ Y$ absorbs $ B$ and hence $ M+Y$ absorbs all but $ R$.

Black ( $ (R,G,B)=(0,0,0)$) corresponds to $ (C,M,Y)=(1,1,1)$ which should in principle absorb $ R$, $ G$ and $ B$. But in practice this will appear as some dark gray. So in order to be able to produce better contrast printers often use black as $ 4^{\text{th}}$ color. This is the CMYK-model. Its coordinates are obtained from that of the CMY-model by $ K:=\max(C,M,Y)$, $ C:=C-K$, $ M:=M-K$ and $ Y:=Y-K$.

**1.4.3 The YIQ Color Model**

This is used for color TV. Here $ Y$ is the luminance (the only component necessary for B&W-TV). The conversion from RGB to YIQ is given by

$\displaystyle \begin{pmatrix}
Y \\ I \\ Q
\end{pmatrix}=
\begin{pmatrix}
0.30 &...
...& -0.52 & 0.31 \\
\end{pmatrix}\cdot
\begin{pmatrix}
R \\ G \\ B
\end{pmatrix}$

for standard NTSC RGB phosphor with chromaticity values

|  |  |  |  |
| --- | --- | --- | --- |
|  | R | G | B |
| x | 0.67 | 0.21 | 0.14 |
| y | 0.33 | 0.71 | 0.08 |

The advantage of this model is that more bandwidth can be assigned to the Y-component (luminance) to which the human eye is more sensible than to color information. So for NTSC TV there are 4MHz assigned to $ Y$, $ 1.5$MHz to $ I$ and $ 0.6$MHz to $ Q$.

**The HSV color model**

All color models treated so far are hardware oriented. The Hue-Saturation-Value model is oriented towards the user/artist. The allowed coordinates fill a six sided pyramid the 3 top faces of the color cube as base. Note that at the same height colors of different perceived brightness are positioned. Value is given by the height, saturation is coded in the distance from the axes and hue by the position on the boundary.

|  |
| --- |
| \begin{figure}\begin{picture}(5,6)(1,0) \put(2,4){\makebox(0,0){$\bullet$}\makeb... ...(0,-0.3){Black}} \put(1.7,2.2){\makebox(0,0.3){White}} \end{picture}\end{figure} |
| **Figure:** The HSV-model versus the RGB-model |

Note that conversion from RGB to HSV is given by affine coordinate changes on each of the 3 four-sided sub-pyramids corresponding each to 1/3 of the color cube.

**The HLS Color Model**

Here the RGB-cube is deformed in such a way that a six sided double pyramid results with the same base as in the HSV-model, but with two tips at black and at white.

|  |
| --- |
| \begin{figure}\begin{picture}(6,6) \put(2,2){\makebox(0,0){$\bullet$}\makebox(-0... ...put(3,4.5){\line(4,-3){2}} \put(3,4.5){\line(2,-1){1}} \end{picture}\end{figure} |
| **Figure:** The HLS-model |

**Interactive Specification of Color**

Color selectors for GUIs can be represented in various ways, e.g.:

|  |
| --- |
| Image /home/andreas/tex/Books/computer-graphics/img//kcolorchooser.png Image /home/andreas/tex/Books/computer-graphics/img//colorselector-gimp.png |
| **Figure:** kcolor gimp |

|  |
| --- |
| Image /home/andreas/tex/Books/computer-graphics/img//colorselector-GTK.png Image /home/andreas/tex/Books/computer-graphics/img//colorselector-Triangle.png |
| **Figure:** GTK gimp-triangle |

## Interpolating in Color Space

If we interpolate between two colors $ C_1$ and $ C_2$ then the result depends on the color model. Only where the conversion formulas are linear it does not matter which of the two models we use.

## Using Color in Computer Graphics

There could be given many advices concerning the use of color in graphics design. E.g. using color for text web-pages should not be done to extensively. Since here we are mainly concerned with photo realistic images, this is not so much a topic here.