Color Models

Lecture 04

BIL464 Multimedia Systems

Mustafa Sert
Asst. Prof.
msert@baskent.edu.tr

Department of Computer Engineering, Başkent University
Ankara 06810 TURKEY

Content

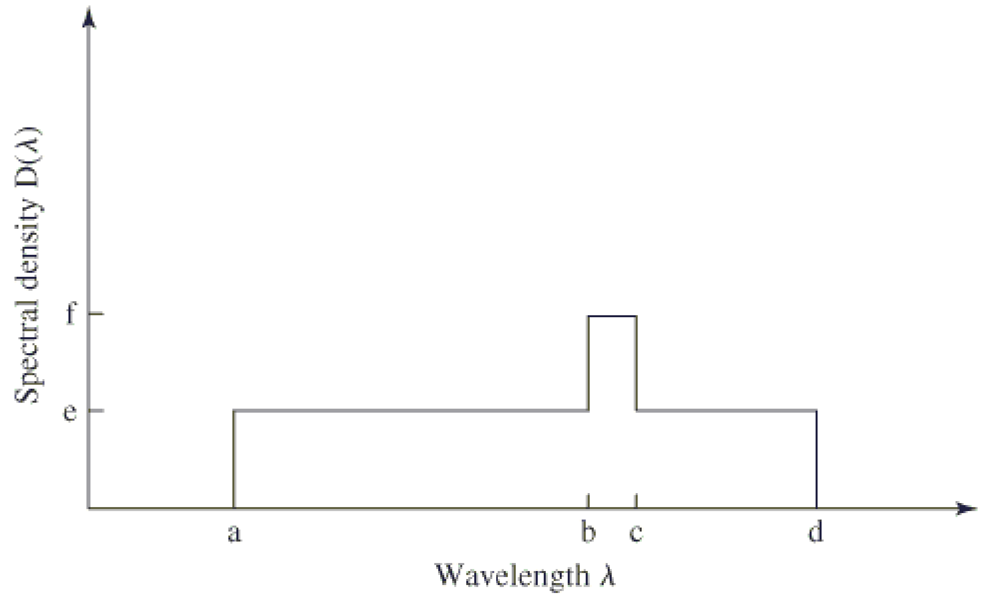
- **Color Perception and Representation**
- **Color Models in Image**
  - RGB Color Model
  - CMY Color Model
  - Transformation from RGB to CMY
  - CMYK Color Model
  - HSV and HLS Color Models
- **Luminance and Chrominance Color Models**
  - YUV Color Model
  - YIQ Color Model
  - YCbCr Color Model

Color Perception and Representation

- Physically, color is composed of **electromagnetic waves**
- For humans, the **wavelengths** of **visible colors** fall between approximately **370 and 780 nanometers** (A nanometer, abbreviated *nm* is 10^{-9} meters.)
- These waves fall upon the color receptors of the eyes, and in a way not completely understood, the human brain translates the interaction between the waves and the eyes as color perception
- It is possible to represent a color by means of a simpler spectral density graph (as in HSV and HLS color models)

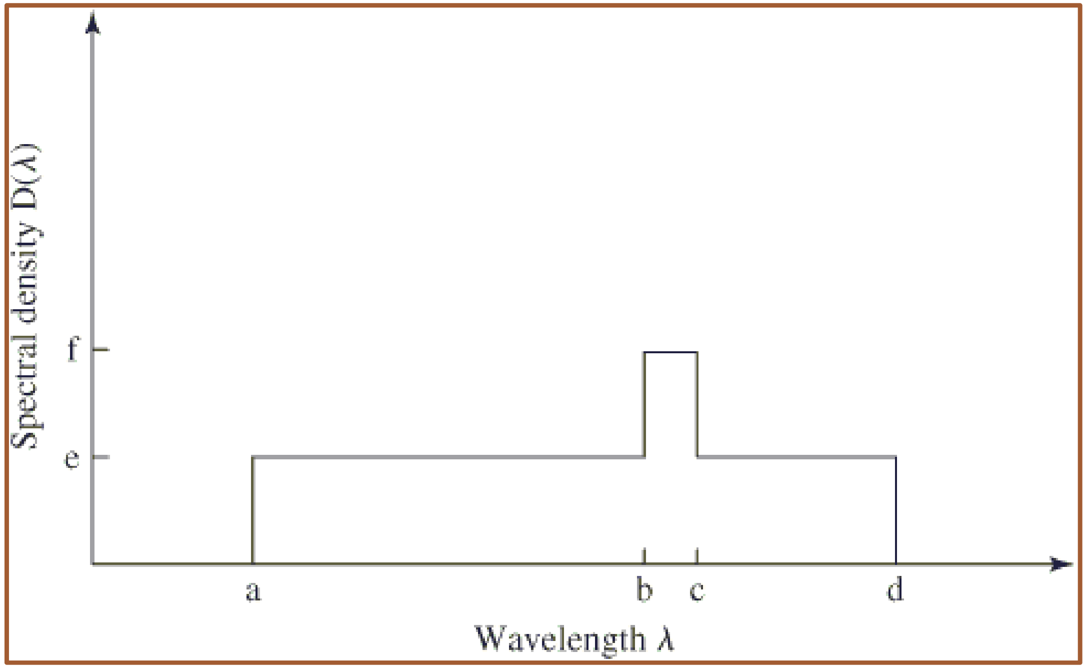
Color Perception and Representation

- Each color in the spectrum can be characterized by a unique graph that has a simple shape, as illustrated in the Figure below
- The graph for each color gives the color's **dominant wavelength**, equivalent to the **hue**; its **saturation** (*i.e.*, color purity); and its **luminance**



Color Perception and Representation

- The **dominant wavelength** is the wavelength at the spike in the graph
- The area beneath the curve indicates the **luminance** L. (This "curve" is a rectangular area with a rectangular spike.)
- **Saturation** S is the ratio of the area of the spike to the total area



$$L = (d - a)e + (f - e)(c - b)$$

$$S = \frac{(f - e)(c - b)}{L}$$

Color Perception and Representation

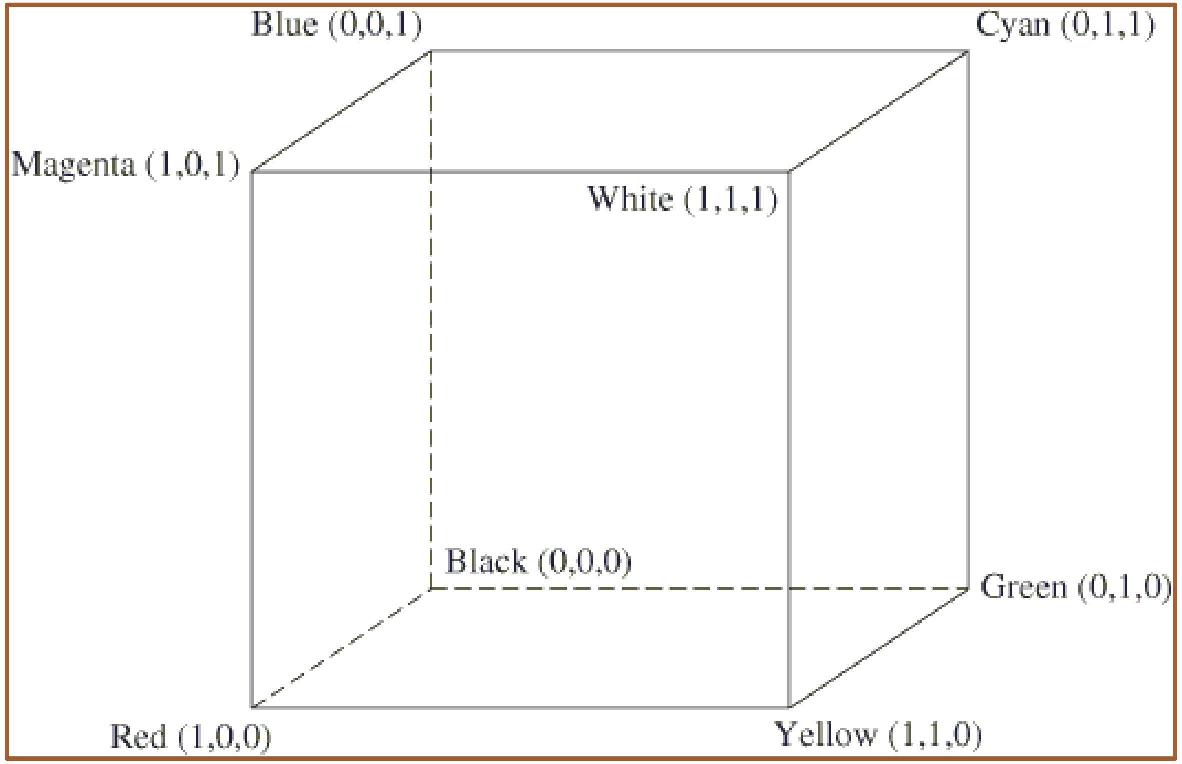
- However, the dimensions of hue, saturation, and brightness do not correspond very well to the way computer monitors are engineered.
- An alternative way to look at a color is as a combination of three primaries. Cathode ray tube (CRT) monitors, for example, display colored light through a combination of red, green and blue (RGB) photosphere that light up at varying intensities when excited by an electron beam.
- Similarly, liquid crystal display (LCD) panels display color with neighboring pixels of red, green, and blue that are either lit up or masked by the liquid crystals.
- So what is the best way to model color for a computer?
  - There is no simple answer, since different models have advantages in different situations

RGB Color Model

- One method to create a wide range of colors is by varying combinations of three primary colors.
- Three colors are primary with respect to one another if no one of them can be created as a combination of the other two.
- Red, green, and blue are good choices as primary colors because the cones of the eyes—the colors receptors—are especially sensitive to these hues $C = rR + gG + bB$
  - where r, g, and b indicate the relative amounts of red, green, and blue energy, respectively.
  - R, G, and B are constant values based on the wavelengths chosen for the red, green and blue components.
  - The values r, g, and b are referred to as the values of the RGB ***color components*** (also called ***color channels*** in application programs)

RGB Color Model

- The color space for the RGB color model is easy to depict graphically.
  - Let R, G, and G correspond to three axes in three-dimensional space. We will normalize the relative amounts of red, green, and blue in a color so that each value varies between 0 and 1



RGB Color Model

- The corresponding RGB color mode in image processing programs is more likely to have values ranging between 0 and 255, since each of the three components is captured in eight bits.
- What is important is the relative amounts of each component, and the size of these amounts with respect to the maximum possible values. For example, the light orange described as (1, 0.65, 0.15) above would become (255, 166, 38) in an RGB mode with maximum values of 255
- It's interesting to note that grayscale values fall along the RGB cube's diagonal from (0,0,0) to (1,1,1). All grayscale values have equal amounts of R, G, and B.

RGB Color Model

- When an image is converted from RGB color to grayscale in an image processing program, the below equation can be used for the conversion of each pixel value.
- This equation reflects the fact that the human eye is most sensitive to green and least sensitive to blue.
- Let an RGB color pixel be given by (R, G, B), where R, G, and B are the red, green, and blue color components, respectively. Then the corresponding grayscale value is given by (L, L, L), where

$$L = 0.30R + 0.59G + 0.11B$$

- Since all three color components are equal in a gray pixel, only one of the three values needs to be stored. Thus a 24-bit RGB pixel can be stored as an 8-bit grayscale pixel

CMY Color Model

- Like the RGB color model, the CMY *color model* divides a color into three primaries, but using a subtractive rather than an additive color (as in RGB) creation process.
- The CMY model can be depicted in a unit cube similar to the RGB model.
  - The difference is that the origin of the cube is white rather than black, and the value for each component indicates how much red, green and blue are subtracted out, effectively combining the color components cyan, magenta, and yellow, their respective complements.

CMY Color Model

- For a pixel represented in RGB color, the red, green, and blue color components are, respectively, *R*, *G*, and *B*. Then the equivalent *C*, *M*, and *Y* color components are given by

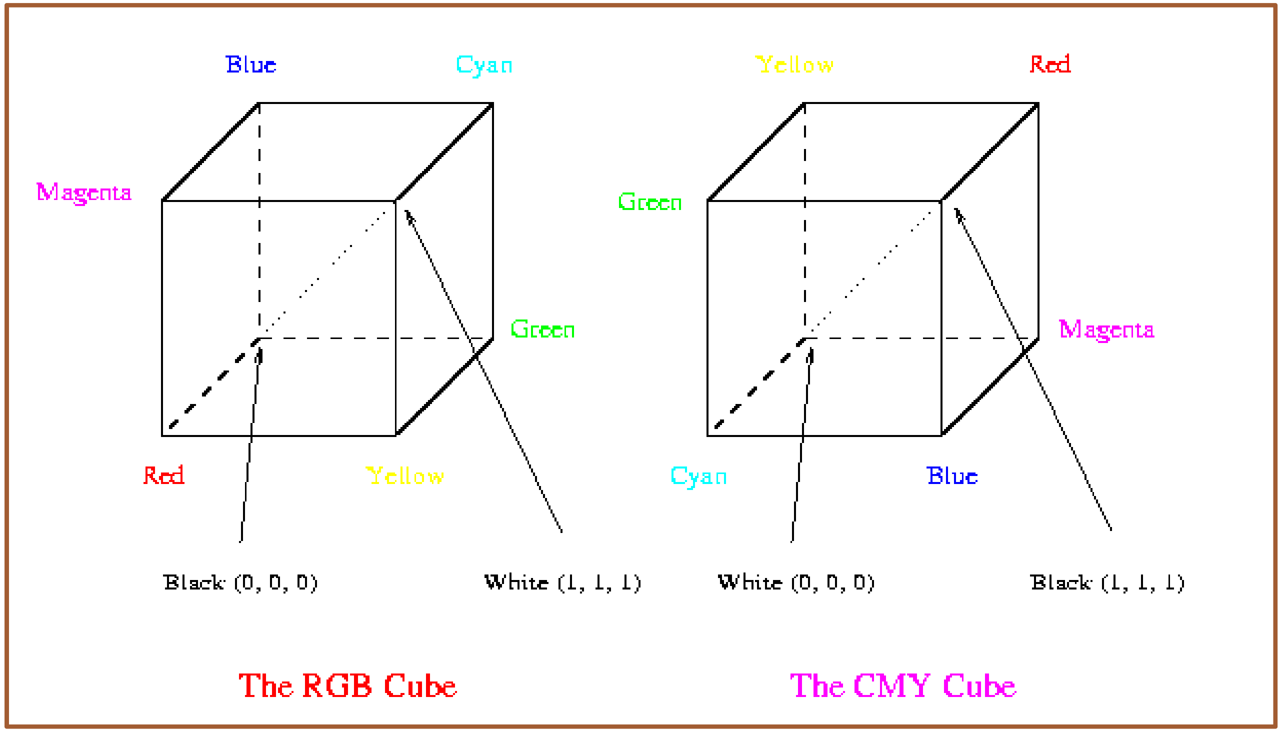
$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

- Similarly, RGB values can be computed from CMY values with

$$R = 1 - C$$
$$G = 1 - M$$
$$B = 1 - Y$$



CMYK Color Model

- The CMY model, used in professional four-color printed processes, indicates how much cyan, magenta, and yellow ink is combined to create color.
- Theoretically, the maximum amount of cyan, magenta, and yellow ink should combine to produce black, but in fact they produce a dark muddy brown.
- In practice, the four-color printing process used in professional presses adds a fourth component, a pure black ink, for greater clarity and contrast.
- The amount of *K*, or black, can be taken as the smallest of the *C*, *M*, and *Y* components in the original CMY model. Thus the CMYK model is defined as follows:

CMYK Color Model

- For a pixel represented in the CMY color model, the cyan, magenta, and yellow color components are, respectively, C, M, and Y.
- Let K be the minimum of C, M, and Y. Then the equivalent color components in the CMYK model, C_{new}, M_{new}, Y_{new}, and K are given by

$$K = \min(C, M, Y)$$

$$C_{new} = C - K$$

$$M_{new} = M - K$$

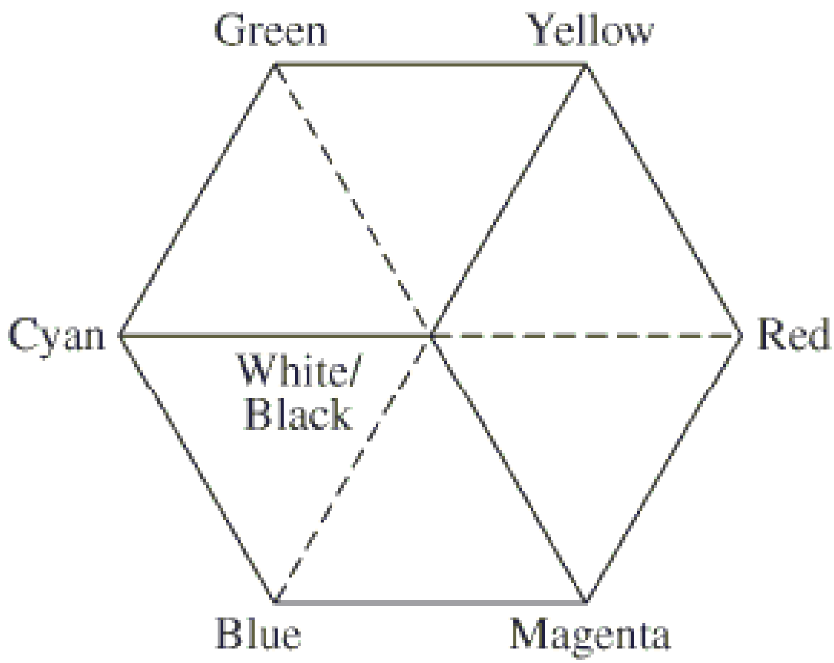
$$Y_{new} = Y - K$$

HSV Color Model

- Instead of representing color by three primary color components, it is possible to speak of a color in terms of its hue (i.e., the essential color), its lightness (or value or luminance), and its saturation (i.e., the purity of the color).
- Both the HSV color model (also called HSB) and the HLS model represent color in this manner.
- Geometrically, the HSV color space is a distortion of the RGB space into a kind of three-dimensional diamond called a hexacone.

HSV Color Model

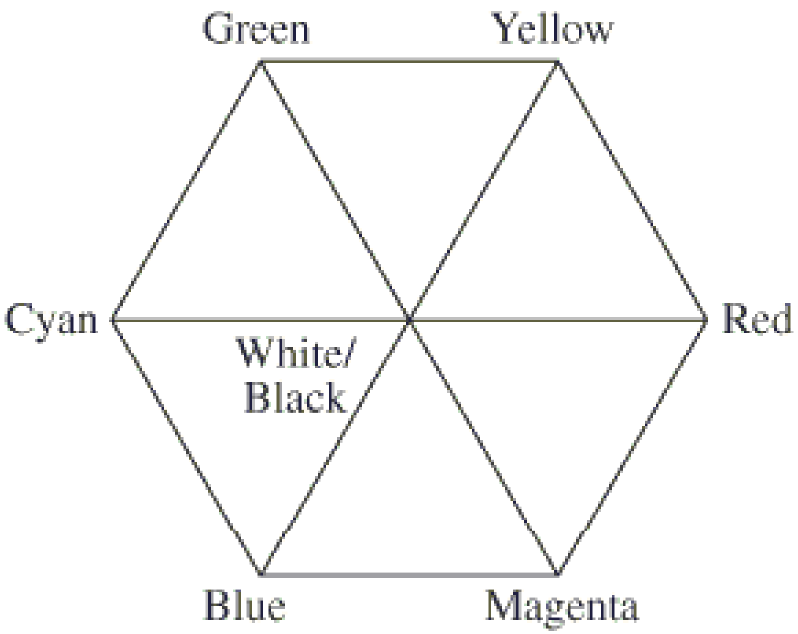
- Picture turning the RGB cube around and tilting it so that you are looking straight into the origin (white/black) with the remaining corners visible—two on the top, two on the bottom, one on the left, and one on the right, as shown in the figure.



RGB color cube viewed from the top

HSV Color Model

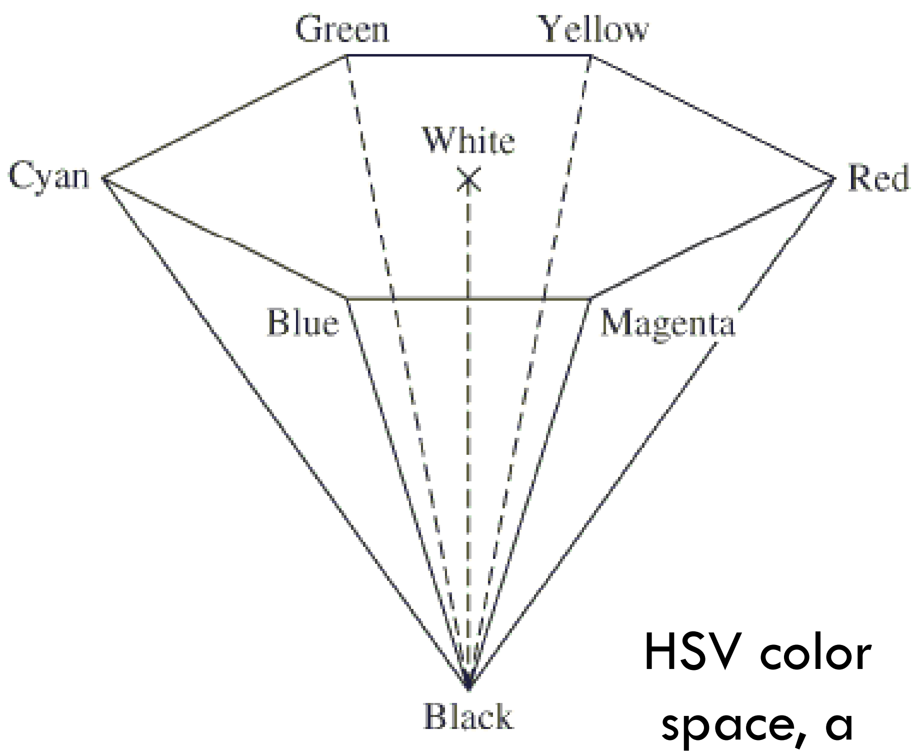
- Imagine this as a flat, two-dimensional hexagon where the center white/black point is connected by a line to each vertex, as shown in the Figure below.
- The six primary colors and their complements are at the outside vertices of this shape



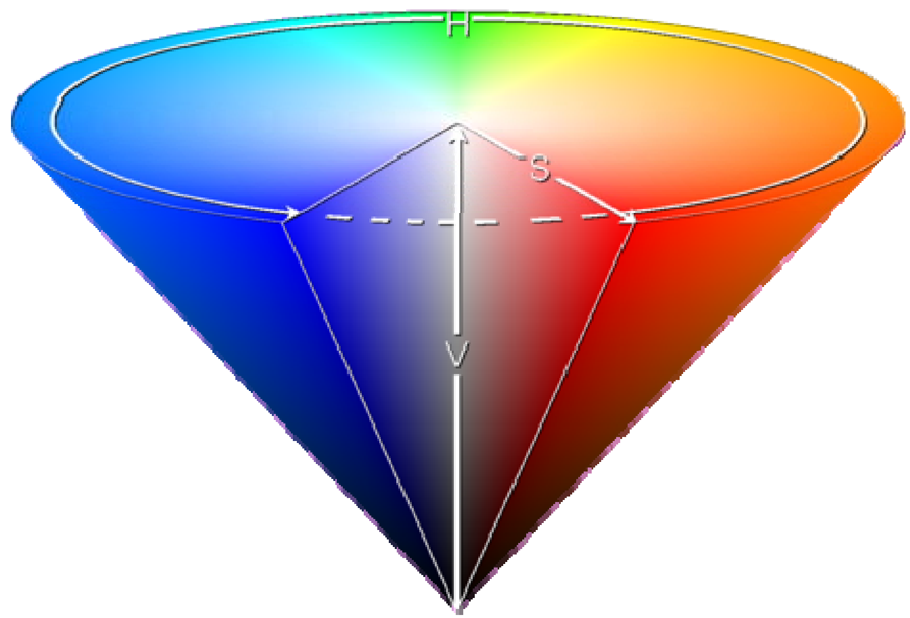
RGB color cube collapsed to 2D

HSV Color Model

- Now imagine expanding this into three dimensions again by pulling down on the center point. You have created the HSV color space, as shown in the Figure below.



HSV color space, a hexacone

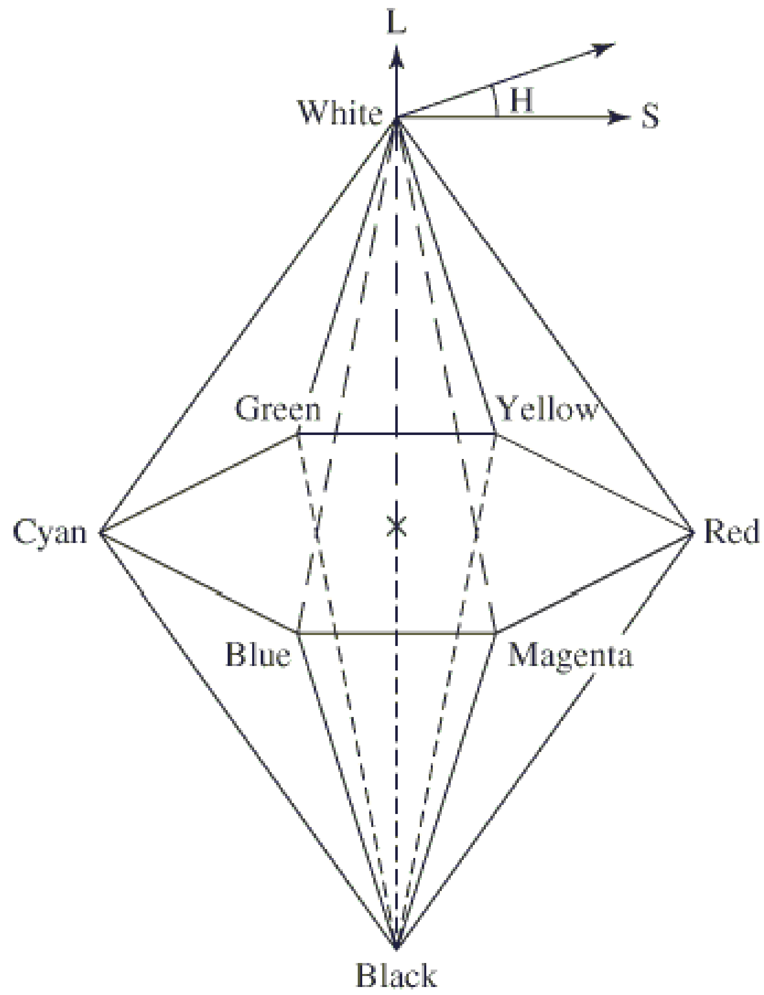


HSV Color Model

- To see how this shape captures the HSV color model, draw an imaginary circle that touches all the vertices of the hexacone's base.
- The hue is represented by a position on this circle given in degrees, from 0 to 360, with red conventionally set at 0. As the hue values increase, you move counterclockwise through yellow, green, cyan, etc.
- Saturation is a function of the color's distance from the central axis (*i*.e., the value axis). The farther a color is from this axis, the more saturated the color.
- The value axis lies from the black point of the hexacone through the center of the circle, with values ranging from 0 for black to 1 for white, where 0 is at the tip and 1 is on the surface of the hexacone. For example, (58°, 0.88, 0.93) is a bright yellow.

HLS Color Model

- The HLS color model is essentially the same.
- To create the HLS color space from the HSV space (and hence from RGB), go through the same steps illustrated in the previous three figures.
- Then take a mirror image of the shape in last figure and connect it to the top, as in the figure given on the right.
- Hue and saturation are given as before, but now lightness varies from 0 at the black tip to 1 at the white tip of the double cones



HLS Color Space

RGB to HSV and HLS Conversions

- RGB color space to either HSV or HLS is a non-linear transformation. In other words, to translate from RGB to HSV, you can't simply multiply each of the R, G, and B components by some coefficient.
- Algorithm for RGB to HSV:

RGB TO HSV

```
algorithm RGB_to_HSV
/* Input: r, g, and b, each real numbers in the range [0 . . . 1].
Output: h, a real number in the range of [0 . . . 360),
except if s = 0, in which case h is undefined.
s and v are real numbers in the range of [0 . . . 1].*/
{
  max = maximum(r,g,b)
  min = minimum(r,g,b)
  v = max
  if max ≠ 0 then s = (max − min)/max
  else s = 0
  if s == 0 then h = undefined
  else {
    diff = max − min
    if r == max then h = (g − b) / diff
    else if g == max then h = 2 + (b − r) / diff
    else if b == max then h = 4 + (r − g) / diff
    h = h * 60
    if h < 0 then h = h + 360
  }
}
```

RGB to HSV and HLS Conversions

- Algorithm for RGB to HLS:

RGB TO HLS

```
algorithm RGB_to_HLS
/* Input r, g, and b, each real numbers in the range [0 . . . 1]
representing the red, green, and blue color components, respectively
Output: h, a real number in the range of [0 . . . 360]
except if s = 0, in which case h is undefined.
L and s are real numbers in the range of [0 . . . 1].
h, L, and s represent hue, lightness, and saturation, respectively.*/
{
  max = maximum(r,g,b)
  min = minimum(r,g,b)
  L = average(max, min)
  if max == min then s = 0
  else {
    sum = max + min
    diff = max - min
    if L ≤ 0.5 then s = diff / sum
    else s = diff / (2 - max + min)
    r_temp = (max - r) / diff
    g_temp = (max - g) / diff
    b_temp = (max - b) / diff
    if r == max then h = b_temp - g_temp
    else if g == max then h = 2 + r_temp - b_temp
    else if b == max then h = 4 + g_temp - r_temp
    h = h * 60
    if h < 0 then h = h + 360
  }
}
```

Luminance and Chrominance Color Models

- Another way to specify a color is to capture all the luminance information in one value and put the color (*i.e.*, *chrominance*) information in the other two values.
- The YIQ model is one example that takes this approach
- The YIQ model is a simple translation of the RGB model, separating out the information in a way that is more efficient for television broadcasting.
- In the early days of color television, both black and white and color signals had to be transmitted because not all consumers had color television sets.
- It was convenient to consolidate all of the “black and white” information—which is luminance—in one of the three components and capture all the color information in the other two. That way, the same transmission worked for both kinds of consumers.

Luminance and Chrominance Color Models (YIQ)

- A linear transformation of the values makes this possible
- For a pixel represented in RGB color, let the red, green, and blue color components be, respectively, *R*, *G*, and *B*. Then the equivalent *Y*, *I*, and Q color components in the YIQ color model are given by

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- *Y* is the luminance component, and *I* and Q are chrominance. The inverse of the matrix above is used to convert from YIQ to RGB.
- YIQ is the model used in U.S. commercial television broadcasting.

Luminance and Chrominance Color Models (YIQ)

- Isolating luminance in one of the three terms has a further advantage, aside from its advantage in color/black and white broadcasting.
  - **Human vision is more sensitive to differences in luminance than differences in color.**
  - Therefore, it makes more sense to give a more finely nuanced representation of the luminance component than of the chrominance.
  - In practical terms, this means that we don't need as many bits—and **therefore as much bandwidth—for the transmission of the I and Q components relative to the Y component.**
  - It would not be possible to make this savings in bandwidth using the RGB model because in RGB the luminance is not a separate element but instead is implicit in the combination of the three components

Luminance and Chrominance Color Models (YUV, YCbCr)

- The YUV color model, originally used in the European PAL analog video standard, is also based upon luminance and chrominance.
- The YCbCr model is closely related to the YUV, with its chrominance values scaled and shifted.
- YCbCr is used in JPEG and MPEG compression.
- These compression techniques benefit from the separation of luminance from chrominance since some chrominance information can be sacrificed during compression without visible loss of quality in photographic images.
  - This is called ***chroma subsampling*** (We will explain chroma subsampling in more detail).

Luminance and Chrominance Color Models (YUV)

- RGB → YUV

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Ranges:
R/G/B [0 ... 1]
Y [0 ... 1]
U [-0.436 ... +0.436]
V [-0.615 ... +0.615]

- YUV → RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Ranges:
Y [0 ... 1]
U [-0.436 ... +0.436]
V [-0.615 ... +0.615]
R/G/B [0 ... 1]

Luminance and Chrominance Color Models (YCbCr)

- Typically used for JPEG images (it may be different in SD/HD TVs)
- RGB → YCbCr

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Ranges:
R/G/B [0 ... 255]
Y/Cb/Cr [0 ... 255]

- YCbCr → RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.400 \\ 1.000 & -0.343 & -0.711 \\ 1.000 & 1.765 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ (Cb - 128) \\ (Cr - 128) \end{bmatrix}$$

Ranges:
Y/Cb/Cr [0 ... 255]
R/G/B [0 ... 255]