Chapter 4
Color in Image and Video

4.1 color Science

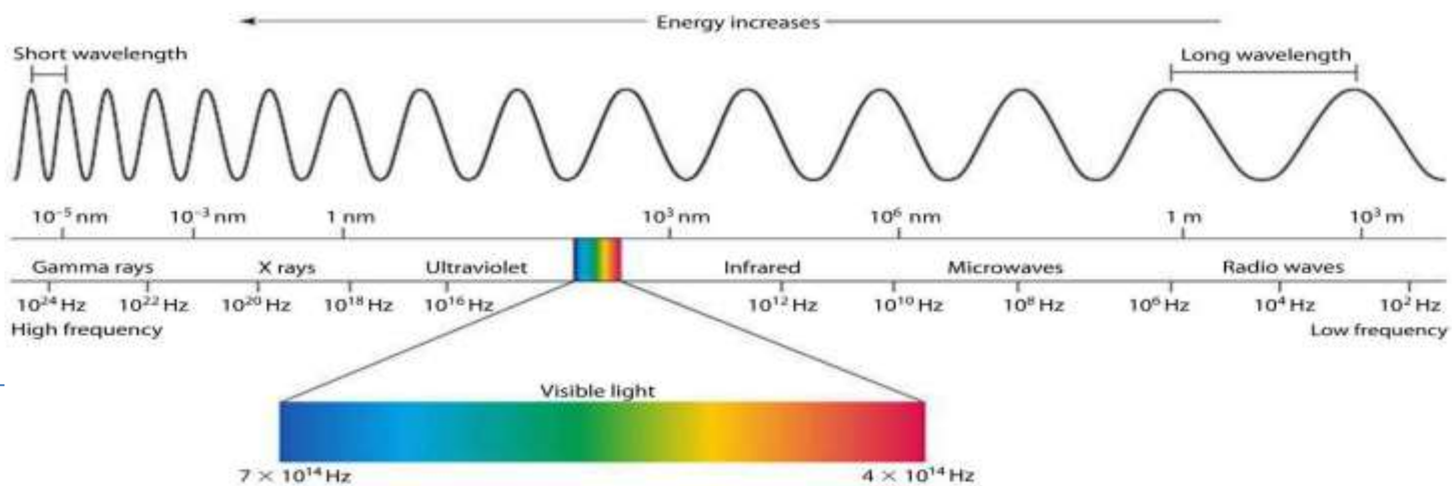
4.2 color Models in Images

4.3 color Models in Video

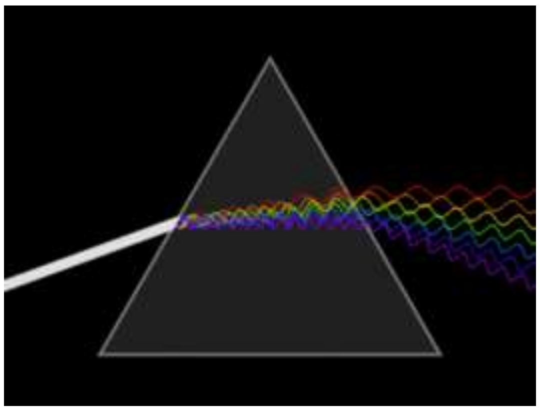
4.1 Color Science

Light and Spectra

- Light is an electromagnetic wave. Its color is characterized by the wavelength content of the light.
  - Most light sources produce contributions over many wavelengths.
  - However, humans cannot detect all light, just contributions that fall in the "visible wavelengths".
  - Short wavelengths produce a blue sensation, long wavelengths produce a red one.



- Figure 4.1 shows the phenomenon that white light contains all the colors of a rainbow.



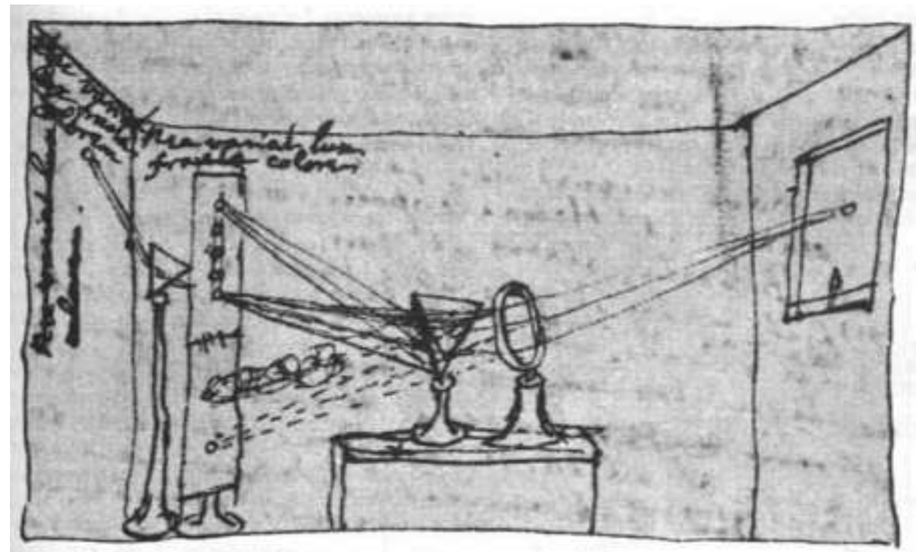


Fig. 4.1: Sir Isaac Newton's experiments.

- Visible light is an electromagnetic wave in the range 400 nm to 700 nm (where nm stands for nanometer, 10^{-9} meters).

- Fig. 4.2 shows the relative power in each wavelength interval for typical outdoor light on a sunny day. This type of curve is called a Spectral Power Distribution (**SPD**) or a **spectrum**.

- The symbol for wavelength is λ. This curve is called E(λ).

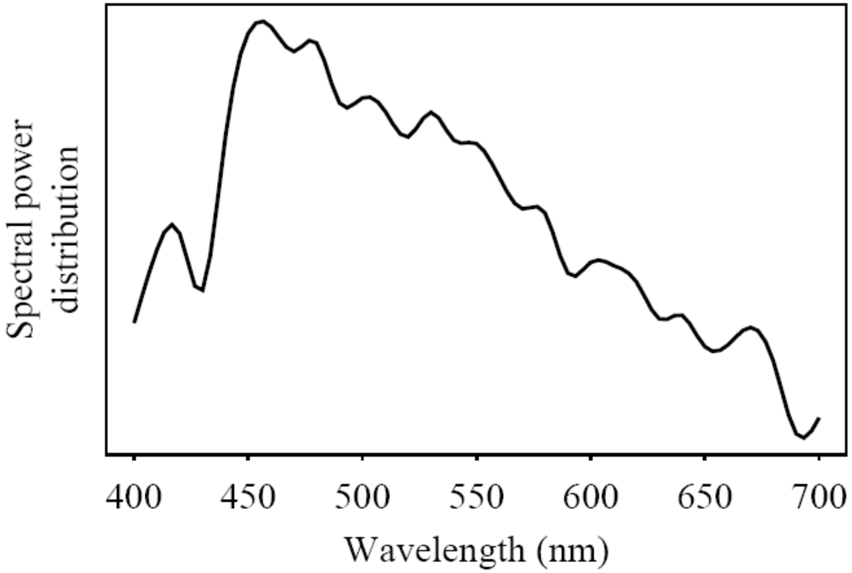
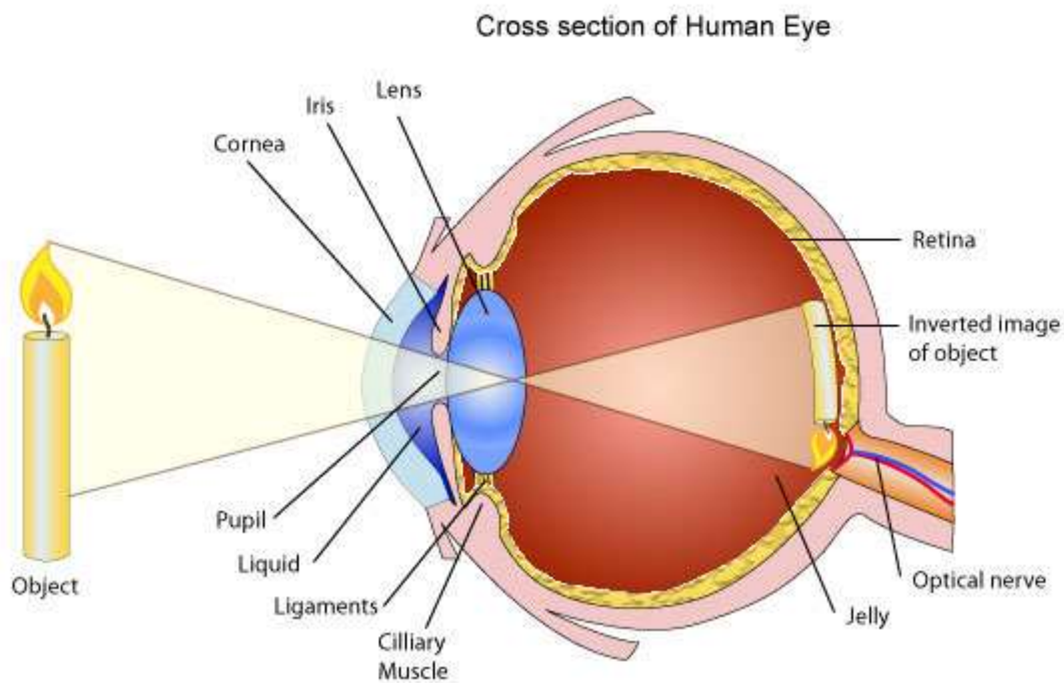


Fig. 4.2: Spectral power distribution of daylight.

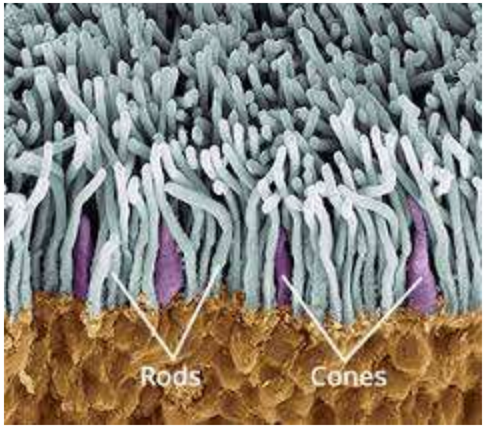
Human Vision

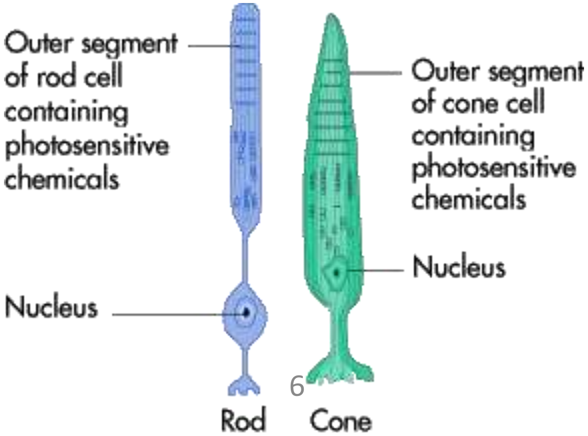
- The eye works like a camera, with the lens focusing an image onto the retina (upside-down and left-right reversed).

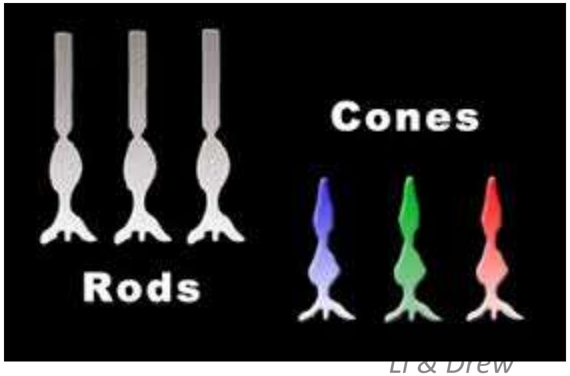


Human Vision

- The retina consists of an *array* of *rods* and three kinds of cones.
- The rods come into play when light levels are low and produce a image in shades of gray ("all cats are gray at night!").
- For higher light levels, the cones each produce a signal. Because of their differing pigments, the three kinds of cones are most sensitive to red (R), green (G), and blue (B) light.
- The eye has about 6 million cones and 120 million **rods.**







Spectral Sensitivity of the Eye

- The sensitivity of our *receptors* is also a function of wavelength (Fig. 4.3 below).

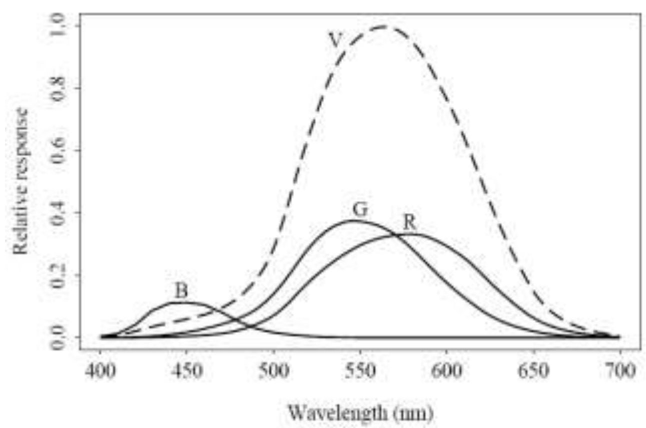


Fig. 4.3: R,G, and B cones, and Luminous Efficiency curve V(λ).

- Blue is a late addition in evolution.
  - Statistically, Blue is the favorite color of humans, regardless of nationality — perhaps for this reason: Blue is a latecomer and thus is a bit surprising!

Image Formation

- Surfaces reflect different amounts of light at different wavelengths, and dark surfaces reflect less energy than light surfaces.

- Fig. 4.4 shows the surface spectral reflectance from (1) orange sneakers and (2) faded blue jeans.

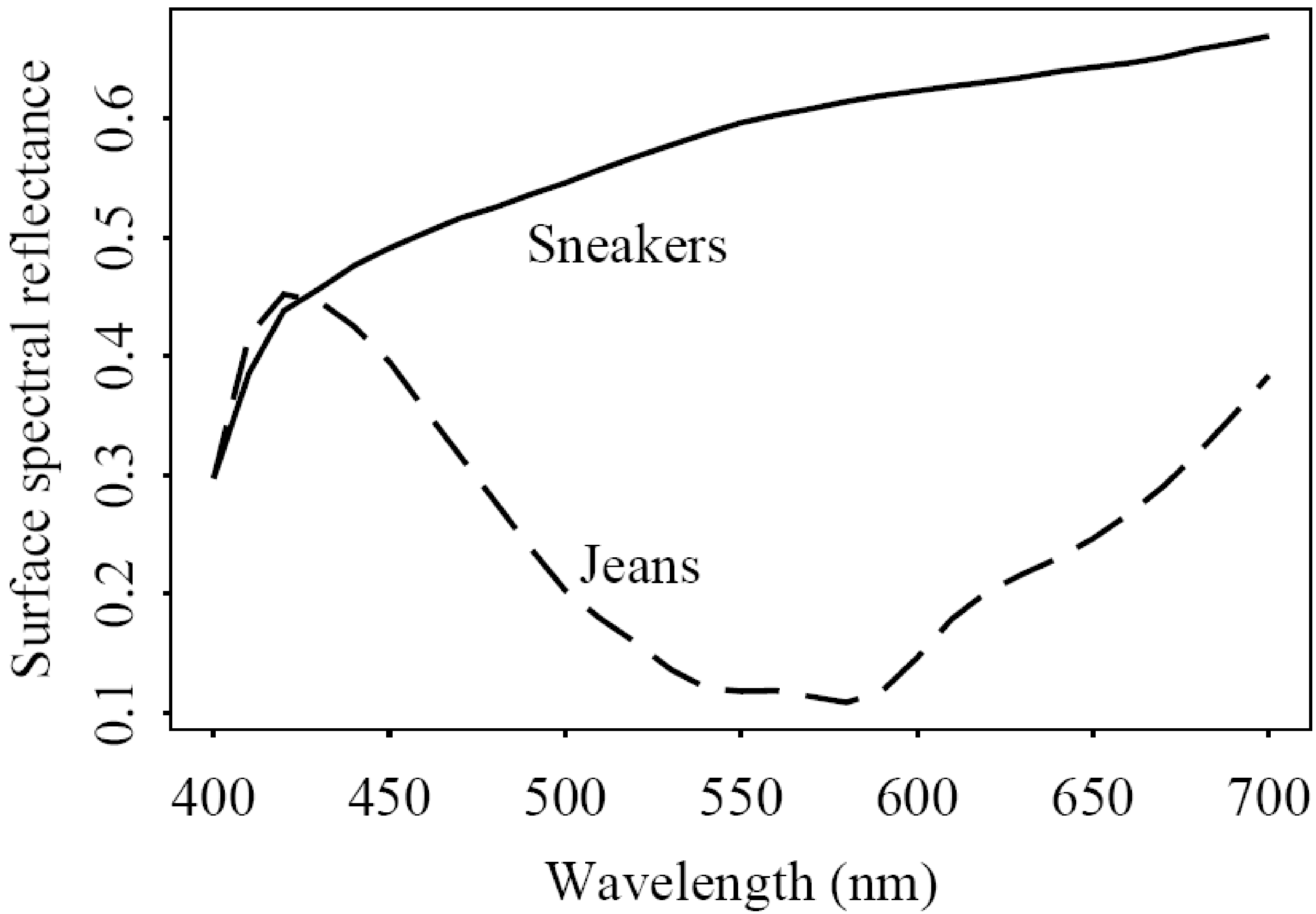
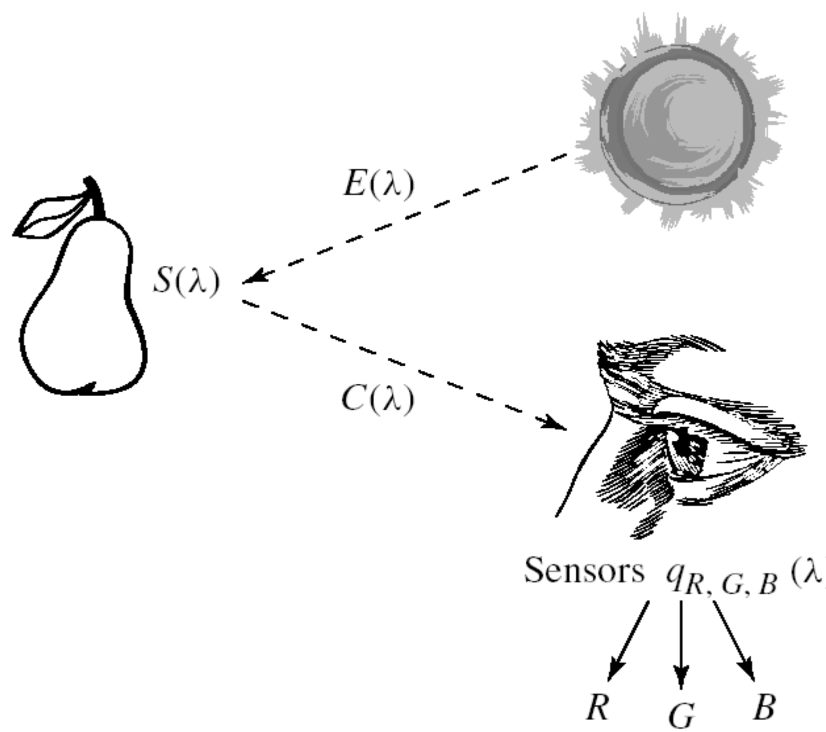


Fig. 4.4: Surface spectral reflectance functions S(λ) for objects.

- Image formation is thus:
  - Light from the illuminant with SPD $E(\lambda)$ falling on a surface, with surface spectral reflectance function $S(\lambda)$, is reflected, and then is filtered by the eye's cone functions q (λ).
  - The function $C(\lambda)$ is called the color signal and consists of the product of $E(\lambda)$, the illuminant, times reflectance:

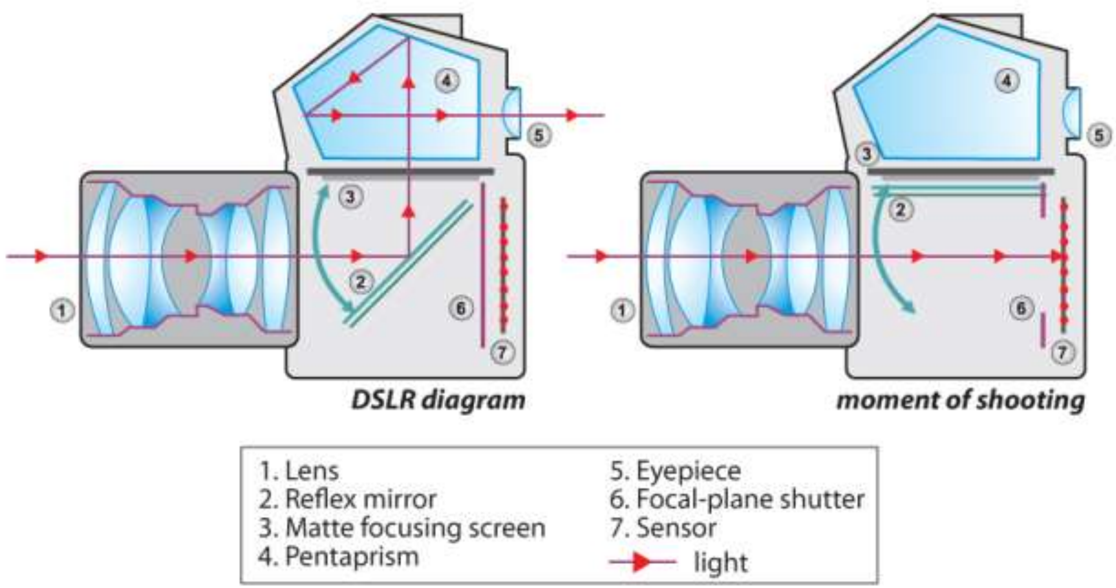
$$C(\lambda) = E(\lambda)\ S(\lambda).$$



Camera Systems

- Camera systems are made in a similar fashion:
  - a camera has three signals produced at each pixel location (corresponding to a retinal position).

- Analog signals are **converted to digital**, **truncated to integers**, and stored. If the precision used is 8-bit, then the maximum value for any of R,G,B is 255, and the minimum is 0.



- The combined effect is shown in Fig. 4.7(b). Here, a ramp is shown in 16 steps from gray-level 0 to gray-level 255.

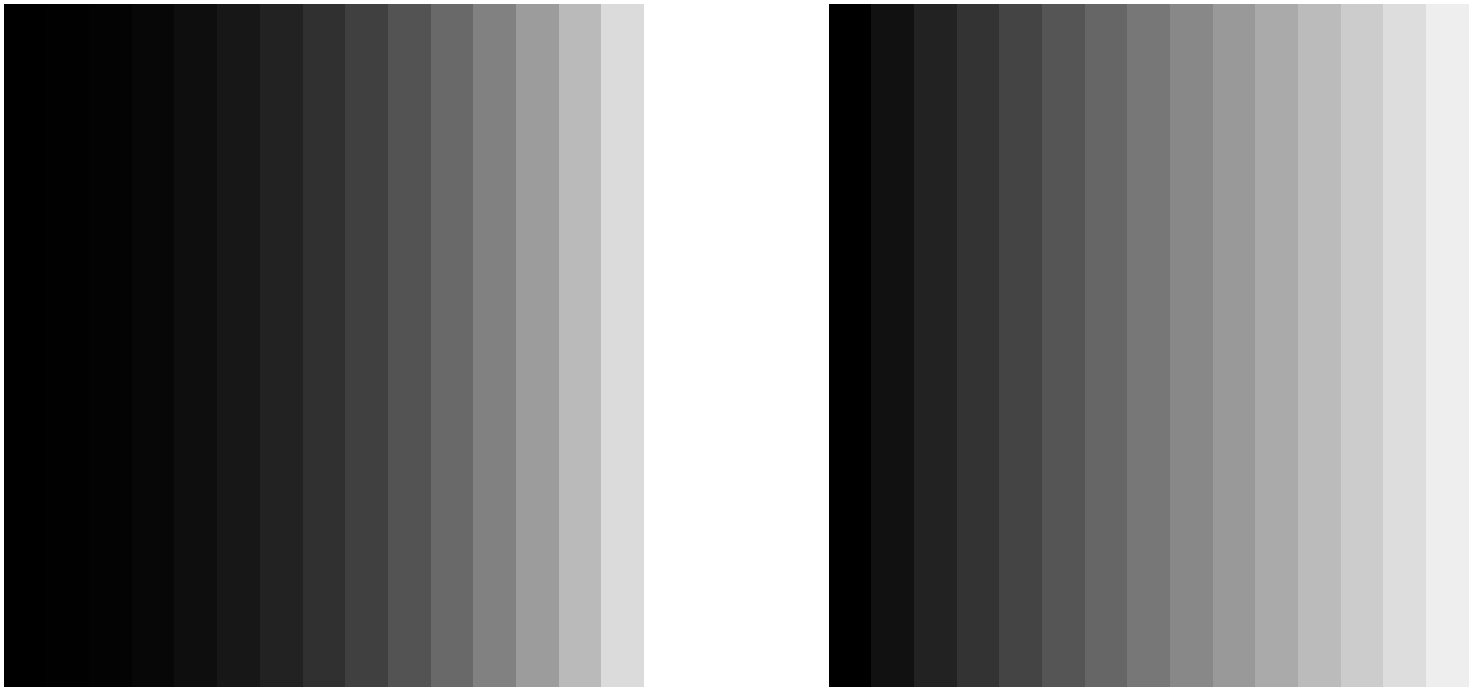


Fig. 4.7: (a): Display of ramp from 0 to 255, with no gamma correction. (b): Image with gamma correction applied

Color-Matching Functions

- We do not know the eye-sensitivity curves of Fig.4.3,
- a technique evolved in psychology for matching a combination of basic R, G, and B lights to a given shade.

- The basic situation is shown in Fig.4.8. A device for carrying out such an experiment is called a **colorimeter**.

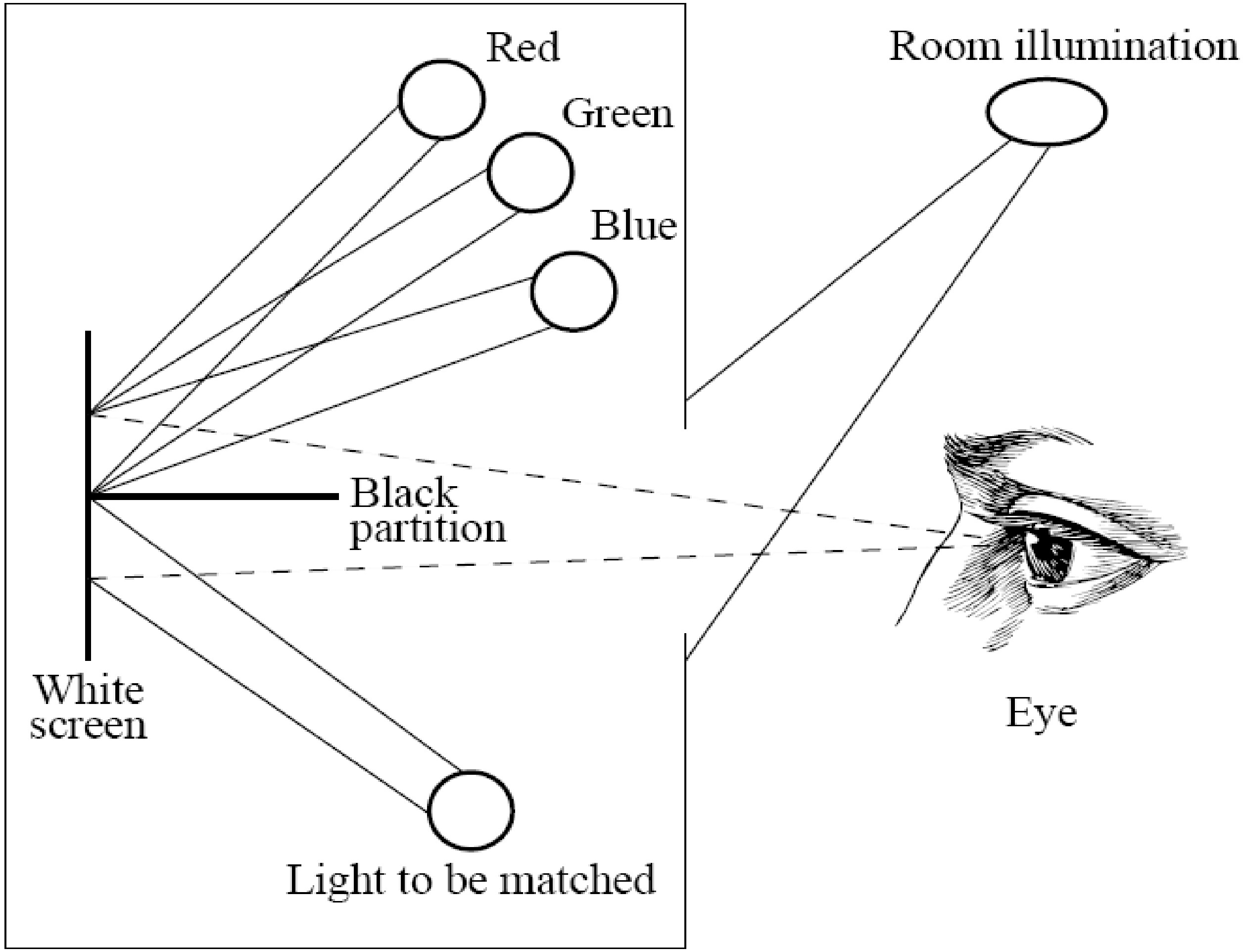


Fig. 4.8: colorimeter experiment.

- The particular set of three basic lights used in an experiment are called the set of **color primaries**.

- To match a given color, a subject is asked to separately adjust the brightness of the three primaries using a set of controls until the resulting spot of light most closely matches the desired color.

4.2 Color Models in Images

- Color models and spaces used for stored, displayed, and printed images.

- **RGB Color Model for Displays**

  1. We expect to be able to use 8 bits per color channel for color that is accurate enough.

  2. However, in fact we have to use about 12 bits per channel to avoid an aliasing effect in dark image areas — contour bands that result from gamma correction.

  3. For images produced from computer graphics, we store integers proportional to intensity in the frame buffer. So should have a gamma correction between the frame buffer and the display.

  4. If gamma correction is applied to floats before quantizing to integers, before storage in the frame buffer, then in fact we can use only 8 bits per channel and still avoid contouring artifacts.

Subtractive color: CMY color Model

(Cyan, Magenta, Yellow)

- So far, we have effectively been dealing only with **additive color**. Namely, when two light beams impinge on a target, their colors add; when two phosphors on a CRT screen are turned on, their colors add.

- But for ink deposited on paper, the opposite situation holds: yellow ink *subtracts* blue from white illumination, but reflects red and green; it appears yellow.

1. Instead of red, green, and blue primaries, we need primaries that amount to -red, -green, and -blue. I.e., we need to *subtract* R, or G, or B.

2. These subtractive color primaries are Cyan (C), Magenta (M) and Yellow (Y) inks.

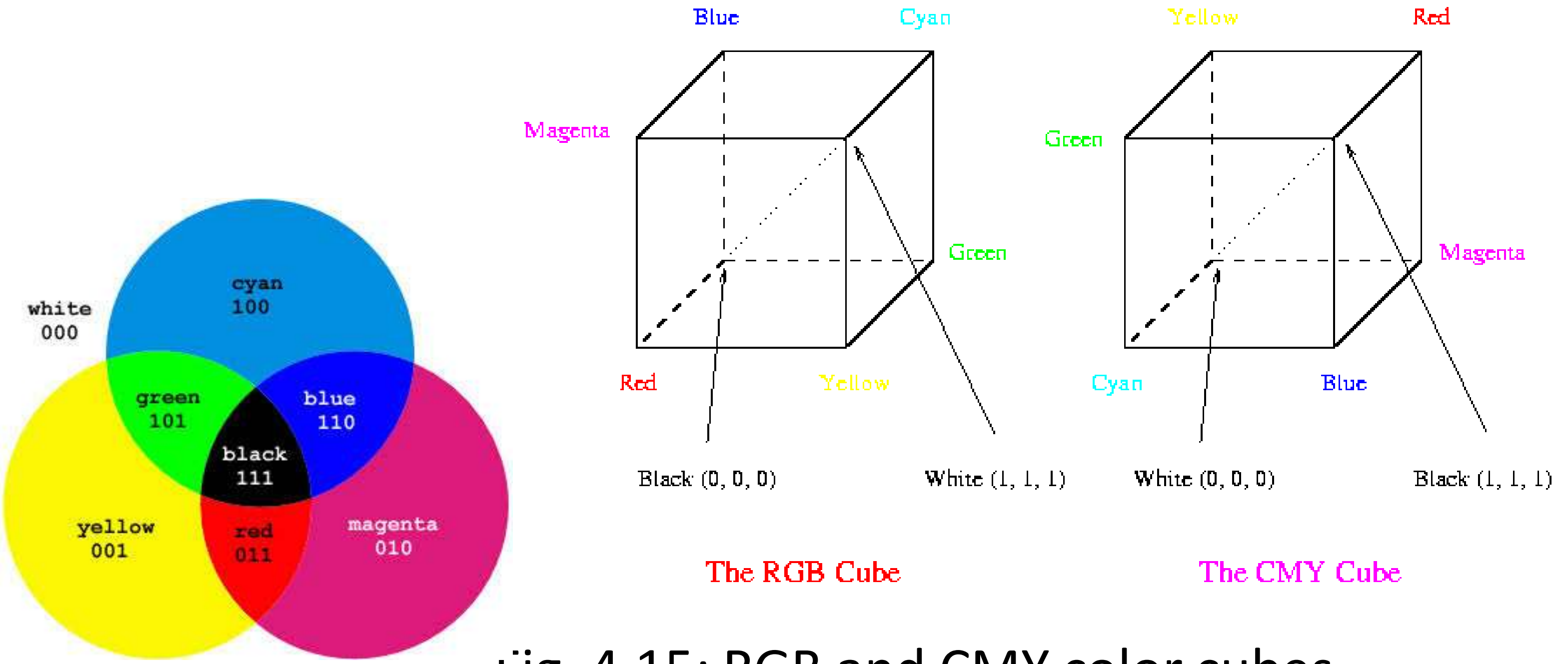


Fig. 4.15: RGB and CMY color cubes.

- Fig. 4.16: color combinations that result from combining primary colors available in the two situations, additive color and subtractive color.

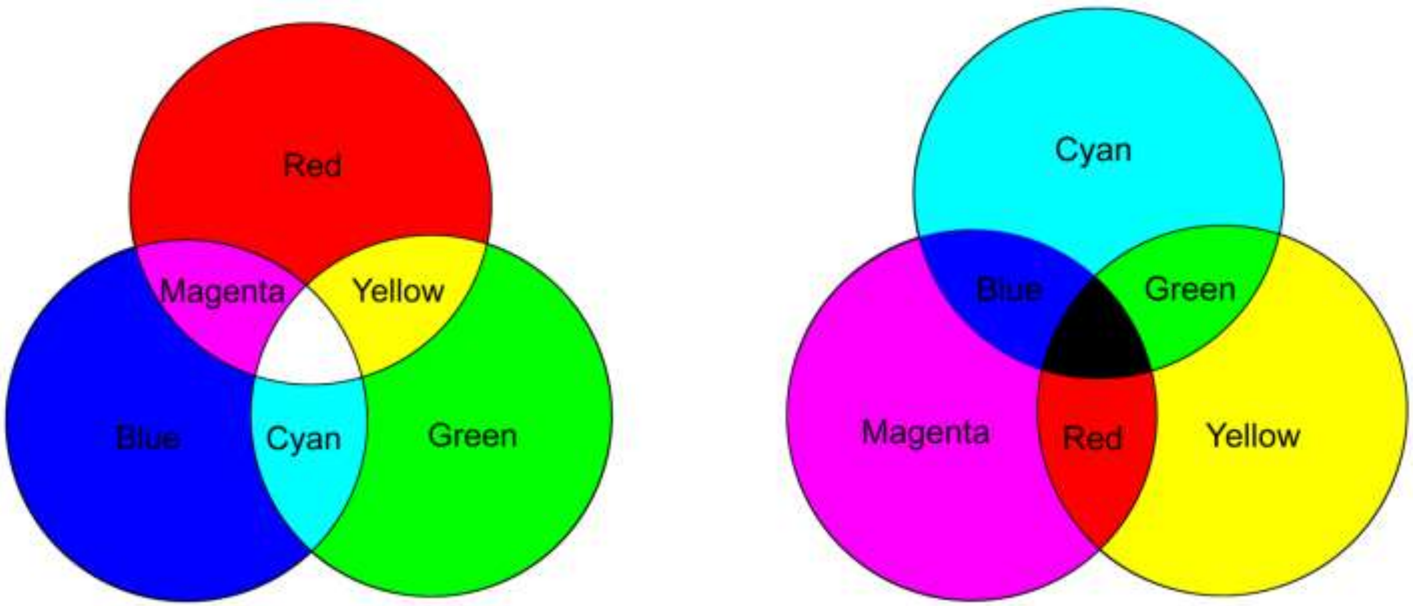


Fig. 4.16: Additive and subtractive color. (a): RGB is used to specify additive color. (b): CMY is used to specify subtractive color

Transformation from RGB to CMY

- Simplest model we can invent to specify what ink density to lay down on paper, to make a certain desired RGB color:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.24}$$

Then the inverse transform is:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \tag{4.25}$$

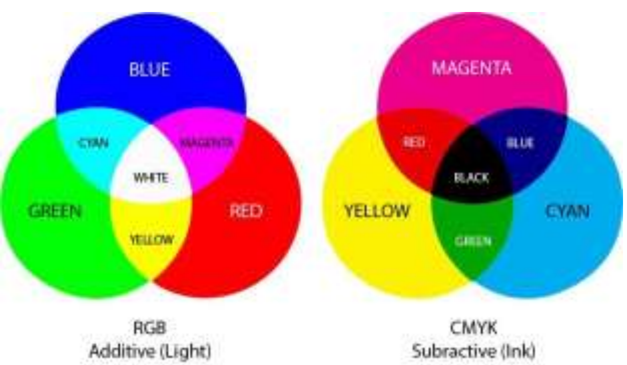
Undercolor Removal: CMYK System

- **Undercolor removal**: Sharper and cheaper printer colors: calculate that part of the CMY mix that would be black, remove it from the color proportions, and add it back as real black.

- The new specification of inks is thus:

$$K \equiv min\{C, M, Y\} \qquad (4.26)$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \Rightarrow \begin{bmatrix} C-K \\ M-K \\ Y-K \end{bmatrix}$$



4.3 Color Models in Video

- **Video Color Transforms**

(a) Largely derive from older analog methods of coding color for TV. Luminance is separated from color information.

(b) For example, a matrix transform method called YIQ is used to transmit TV signals in North America and Japan.

(c) This coding also makes its way into VHS video tape coding in these countries since video tape technologies also use YIQ.

(d) In Europe, video tape uses the PAL or SECAM codings, which are based on TV that uses a matrix transform called YUV.

(e) Finally, digital video mostly uses a matrix transform called YCbCr that is closely related to YUV

YUV Color Model

(a) YUV codes a luminance signal (for gamma-corrected signals) equal to Y ′ in Eq. (4.20). the “luma”.

(b) **Chrominance** refers to the difference between a color and a reference white at the same luminance. → use color differences *U*, *V*:

$$U = B' - Y', \qquad V = R' - Y' \tag{4.27}$$

From Eq. (4.20),

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{4.28}$$

(c) For gray, $R' = G' = B'$, the luminance Y' equals to that gray, since 0.299+0.587+0.114 = 1.0. And for a gray (“black and white”) image, the chrominance (U, V) is zero.

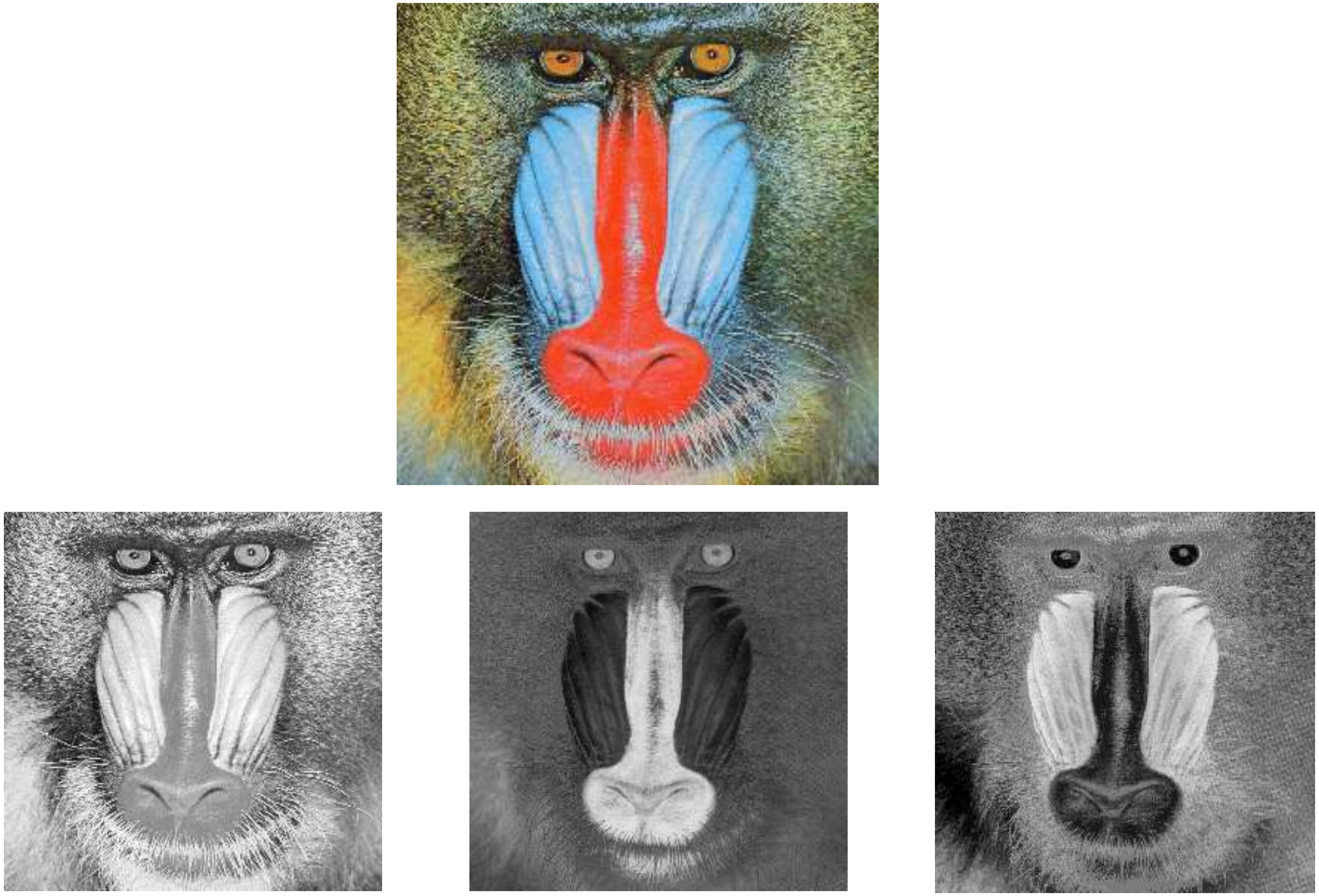


Fig. 4.18: Y ’UV decomposition of color image. Top image (a) is original color image; (b) is Y ’; (c,d) are (U, V)

YIQ Color Model

- YIQ is used in NTSC color TV broadcasting. Again, gray pixels generate zero (I, Q) chrominance signal.

(a) I and Q are a rotated version of U and V .

(b) Y' in YIQ is the same as in YUV; U and V are rotated by 33°:

$$I = 0.492111(R' - Y') \cos 33° - 0.877283(B' - Y') \sin 33°$$

$$Q = 0.492111(R' - Y') \sin 33° + 0.877283(B' - Y') \cos 33° \qquad (4.31)$$

(c) This leads to the following matrix transform:

$$\begin{bmatrix} Y' \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595879 & -0.274133 & -0.321746 \\ 0.211205 & -0.523083 & 0.311878 \end{bmatrix} = \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \qquad (4.32)$$

(d) Fig. 4.19 shows the decomposition of the same color image as above, into YIQ components.

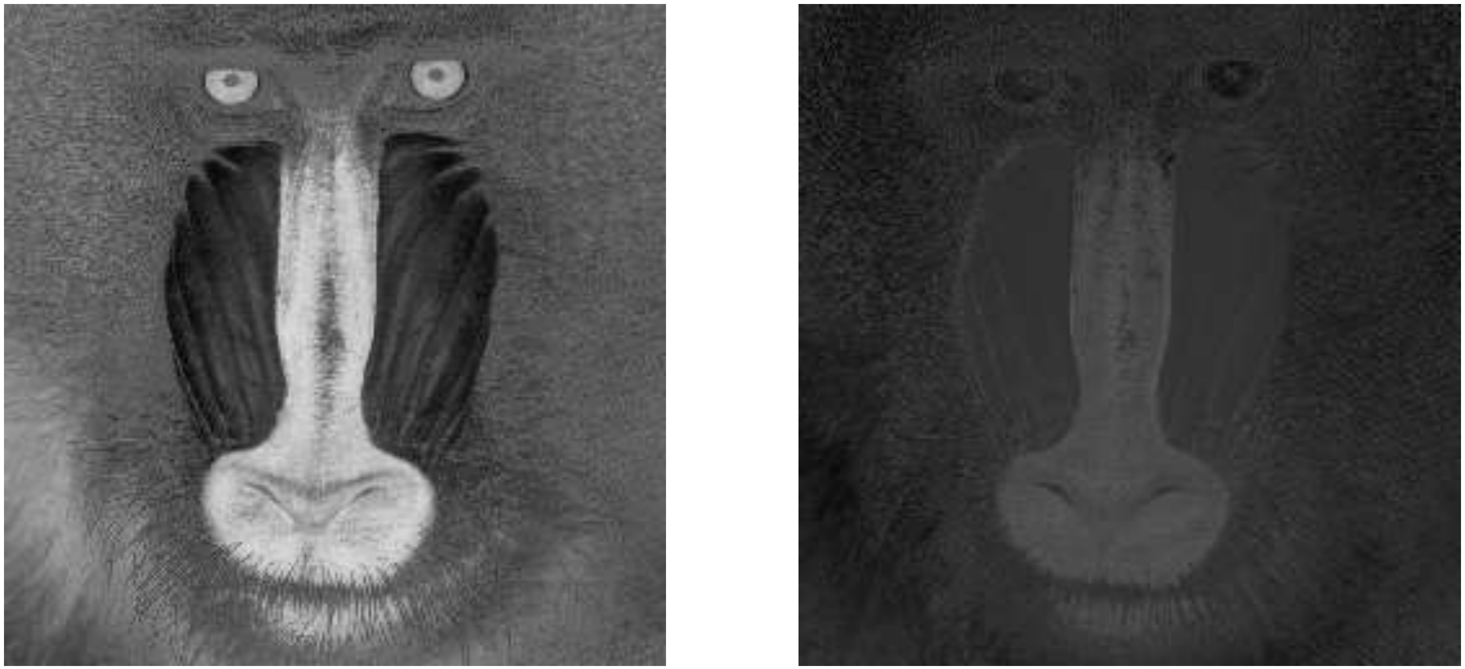


Fig.4.19: I and Q components of color image.

YCbCr Color Model

- The Rec. 601 standard for digital video uses another color space, YC_bC_r, often simply written YCbCr — closely related to the YUV transform.

  (a) YUV is changed by scaling such that C_b is U, but with a coefficient of 0.5 multiplying B'. In some software systems, C_b and C_r are also shifted such that values are between 0 and 1.

  (b) This makes the equations as follows:

$$C_b = ((B' - Y')/1.772) + 0.5$$

$$C_r = ((R' - Y')/1.402) + 0.5 \qquad (4.33)$$

  (c) Written out:

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix} \qquad (4.34)$$

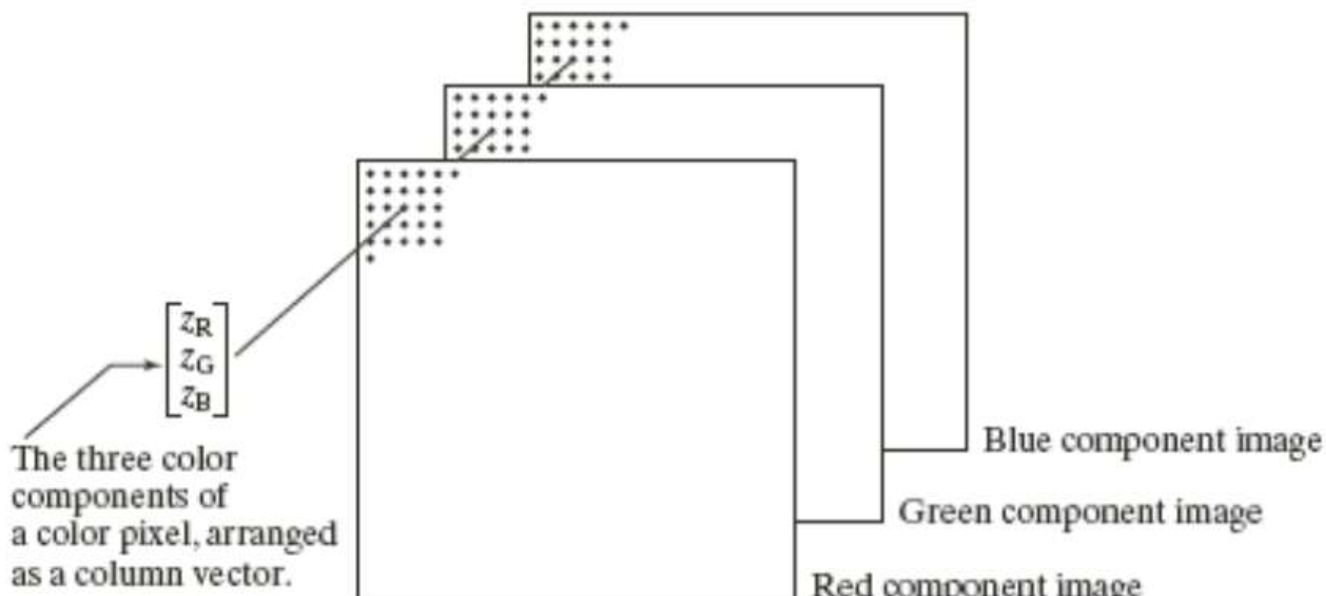
(d) In practice, however, Recommendation 601 specifies 8-bit coding, with a maximum Y' value of only 219, and a minimum of +16. Cb and Cr have a range of ±112 and offset of +128. If R', G', B' are floats in [0.. + 1], then we obtain Y', Cb, Cr in [0..255] via the transform:

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{4.35}$$

(f) The YCbCr transform is used in JPEG image compression and MPEG video compression.

RGB Images

An RGB *color image* is an *M* x *N* x 3 array of *color pixels,* where each color pixel is a triplet corresponding to the red, green, and blue components of an RGB image at a specific spatial location.



RGB Images

Let fR , fG , and fB represent three RGB component images.

An RGB image is formed from these images by using the cat (concatenate) operator to stack the images:

rgb_image = cat(3, fR, fG, fB)

fR, fG, fB should be of the same size.

In general

cat (dim, A1 , A2, ...)

Concatenates A1, A2, … along the dimension specified by dim.

For example, if dim = 1, the arrays are arranged vertically, if dim = 2, they are arranged horizontally, and, if dim = 3, they are stacked in the third dimension

RGB Images

If all component images are identical, the result is a gray-scale image.

The following commands extract the three component images:

fR=rgb_image (: , : , 1) ;

fG=rgb_image (: , : , 2) ;

fB=rgb_image (: , : , 3) ;

rgb2gray

gray_image = rgb2gray(rgb_image)

converts an RGB image to a gray-scale image.

The input RGB image can be of class uint8, uint16, or double; the output image is of the same class as the input.

rgb2ntsc

yiq_image = rgb2ntsc(rgb_image)

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

the input RGB image can be of class uint8, uint16, or double.

The output image is an M X *N* X 3 array of class double.

Component image yiq_image (:, :, 1) is the luminance, yiq_image (:, :, 2) is the hue, and yiq_image(:, :,3) is the saturation image.

ntsc2rgb

rgb_image = ntsc2rgb(yiq_image)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

Both the input and output images are of class double.

The YCbCr Color Space

ycbcr_image = rgb2ycbcr(rgb_image)

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

rgb_image = ycbr2rgb(ycbcr_image)

The input image can be of class uint8, uint16, or double. The output image is of the same class as the input.

The HSV Color Space

HSV (hue, saturation, value) is one of several color systems used by people to select colors (e.g., of paints or inks) from a color wheel or palette.

This color system is considerably closer than the RGB system to the way in which humans experience and describe color sensations.

In artists' terminology, hue, saturation, and value refer approximately to tint, shade, and tone.

The HSV color space is formulated by looking at the RGB color cube along its gray axis (the axis joining the black and white vertices), which results in the hexagonally shaped color palette

Exercise

Exercise: RGB to YIQ and YCbCr Conversion in MATLAB

Objective:

- Convert an image from the **RGB color space** to **YIQ and YCbCr**.
- Visualize and analyze how the color channels are represented.

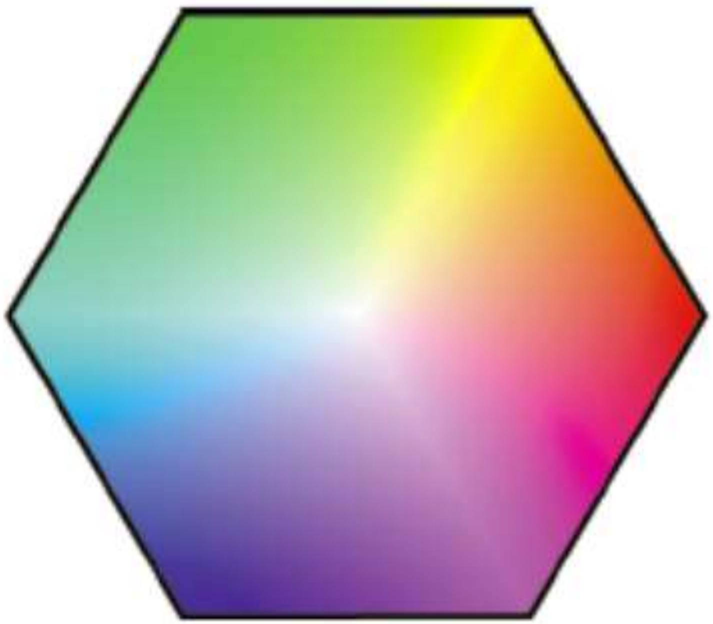
Step 1: Load an Image

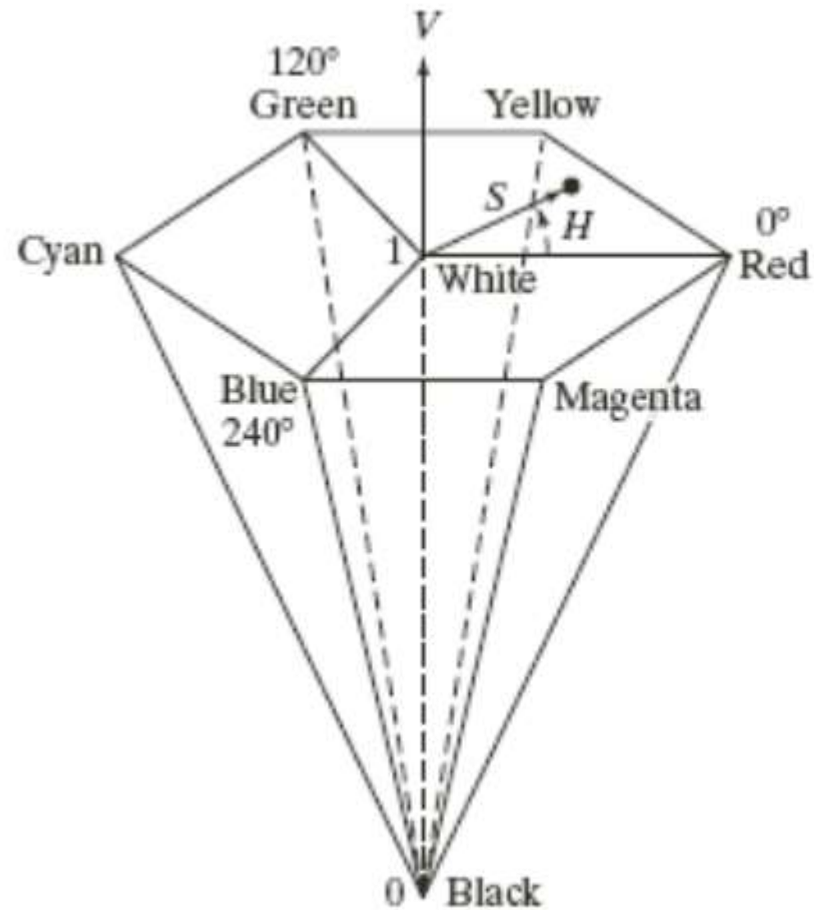
1. Select an **RGB image** to work with.
2. Convert the image into **YIQ and YCbCr** models.
3. Display the individual channels for better understanding.

Student Task

1. Load an image in MATLAB.
2. Write functions to convert **RGB → YIQ** and **RGB → YCbCr**.
3. Display the **original image and the separate channels** for each model.
4. Compare how luminance and chrominance are stored in both models.

The HSV Color Space





The HSV Color Space

Hue is expressed as an angle around a color hexagon, typically using the red axis as the reference (0°) axis.

The value component is measured along the axis of the cone.

- The $V = 0$ end of the axis is black.
- The $V = 1$ end of the axis is white.

Saturation (purity of the color) is measured as the distance from the V axis.

The HSV Color Space

hsv_image = rgb2hsv(rgb_image)

The input RGB image can be of class uint8, uint16, or double; the output image is of class double.

rgb_image = hsv2rgb(hsv_image)

The input image must be of class double. The output is of class double also