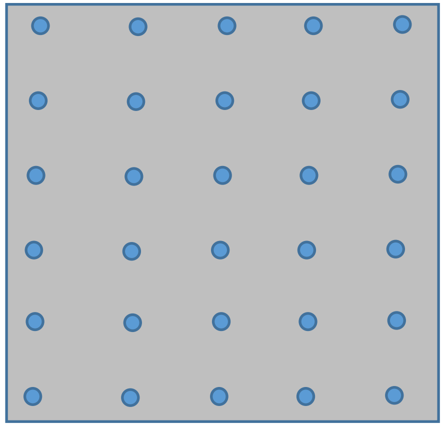
Color Image Processing

IE 404

Instructor: Manish Khare

Gray scale image

- Pixel value is proportional to intensity of light coming from the scene
- Intensity range 0 to 255. Normalized range 0 to 1.
- 0=>0; 255=>1; 130=>0.51; 210=>0.82



gray scale image

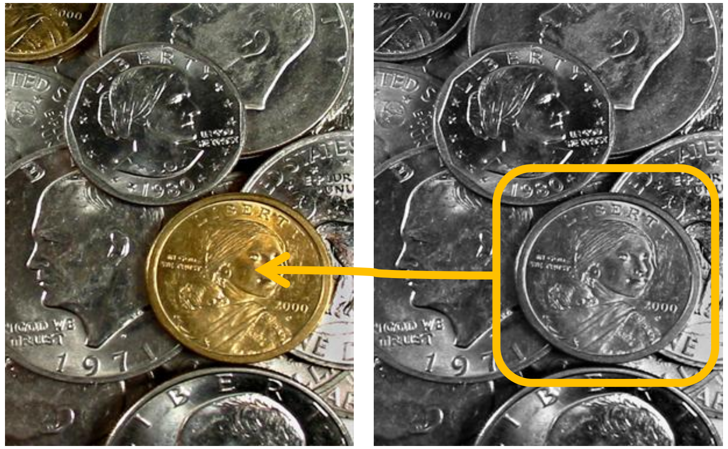
gray scale image

Image courtesy: HFT dataset

Motivation

- In gray scale images, different object types may not be easily distinguishable
  - Color images provide more information about object appearance

color image gray scale image



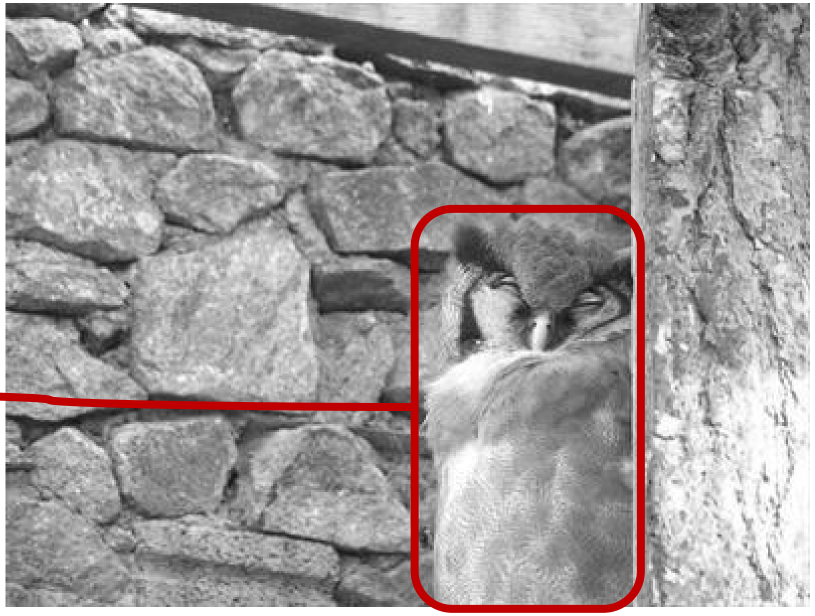
color image gray scale image

Image courtesy: HFT dataset

Motivation

- Object detection - better result using color images

color image



gray scale image

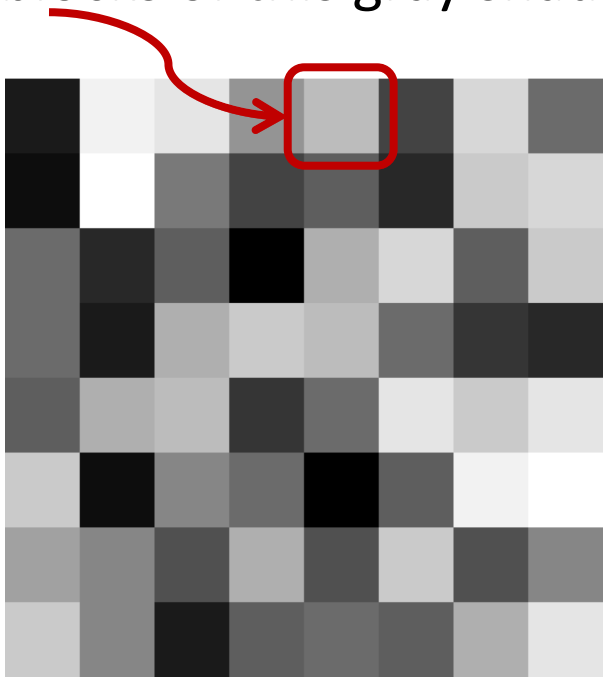
- In gray scale images, objects may appear to be similar to background
- Color information helps to better distinguish objects from background

Image courtesy: HFT dataset

Motivation

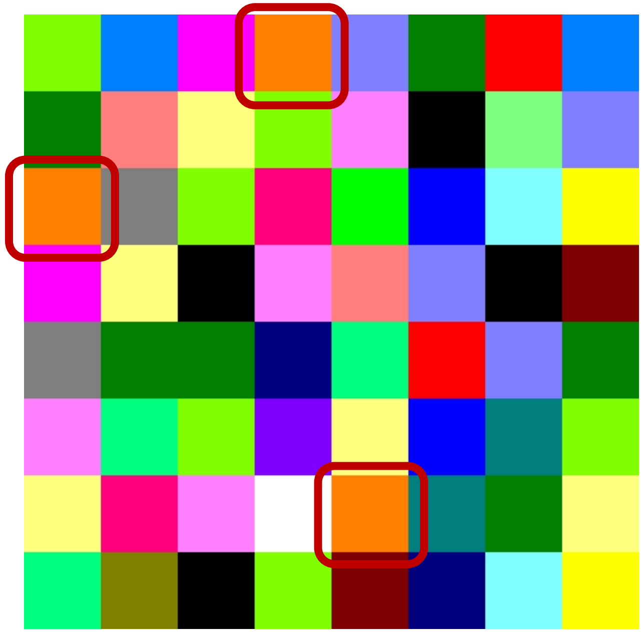
- Humans can distinguish more colors than gray shades

difficult to find the number of blocks of this gray shade



20 unique gray shades

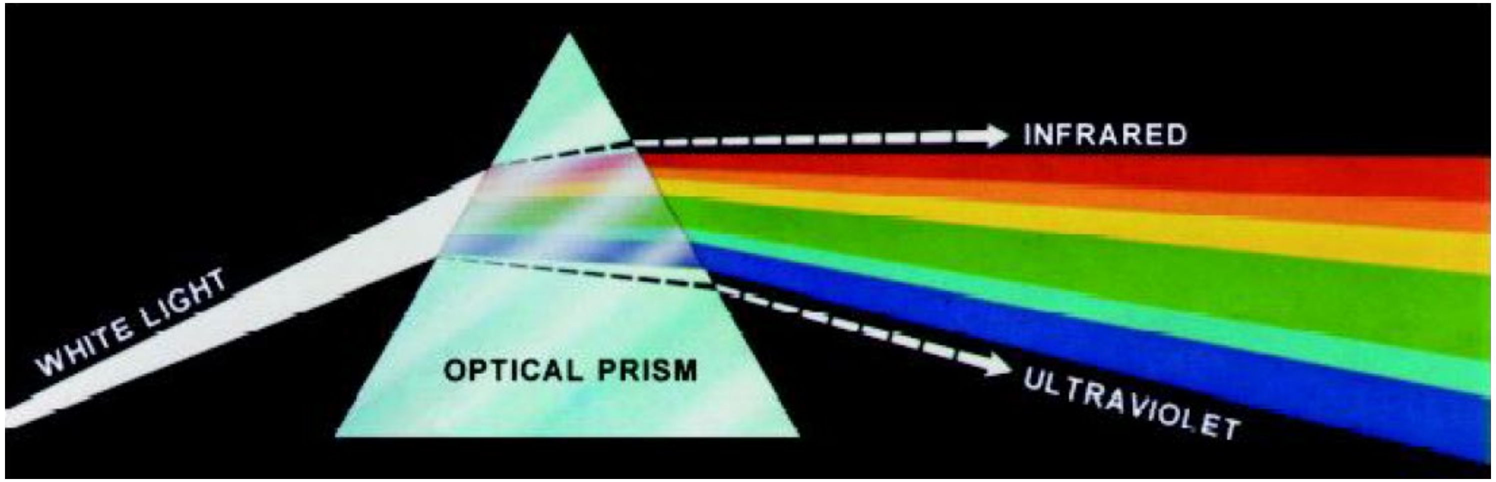
easy to find there are 3 blocks of this color



26 unique colors

Visible light

- Color spectrum (Newton's experiment 1666): white light passes through an optical prism -> split into multiple light components (continuous spectrum of colors) from Violet to Red (VIBGYOR)



Why is the color of an object is 'Blue'?

=> Light reflected of it has predominantly 'Blue' wavelength

Image courtesy: Gonzalez & Woods

Visible light

- We will discuss about *Visible light*
- Wavelegth 400nm to 700nm in EM spectrum
  - narrow band of frequencies

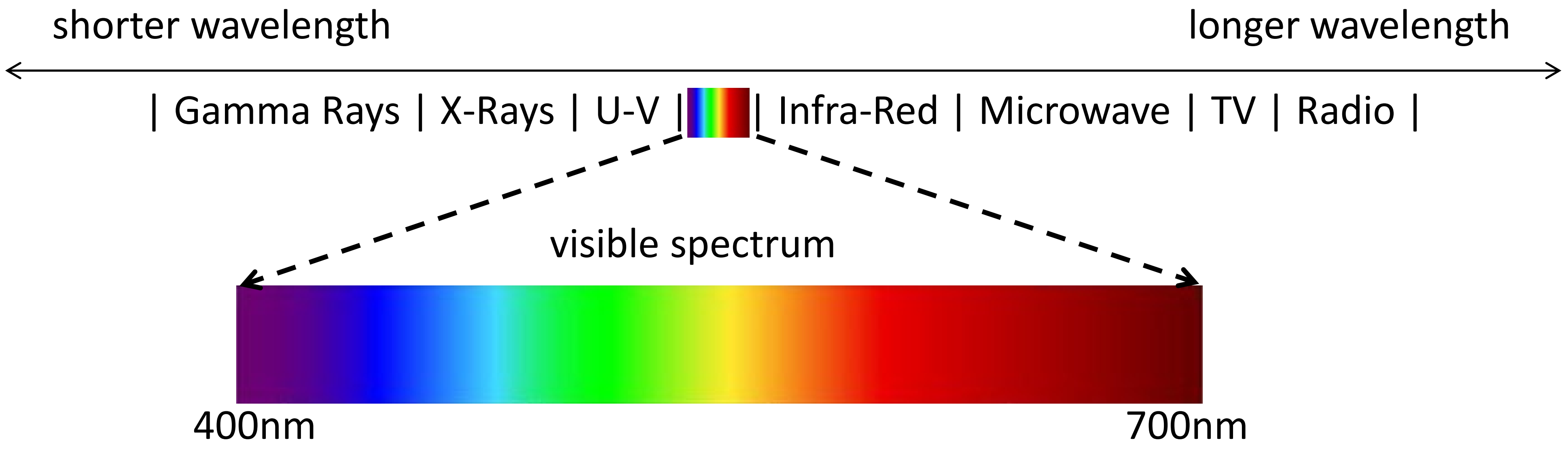
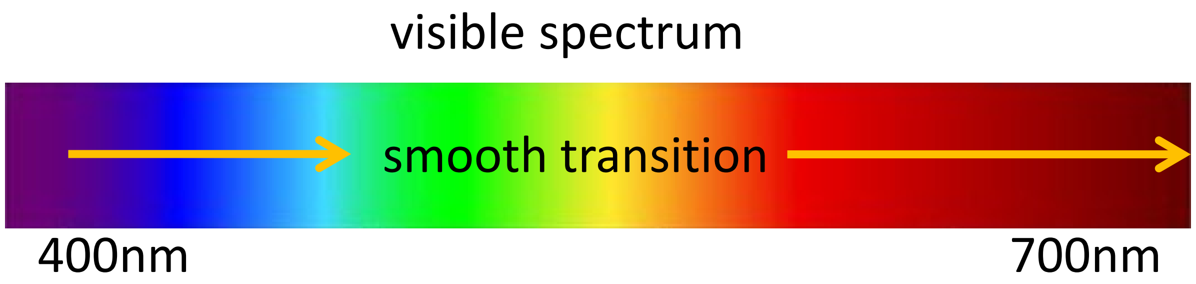


Image courtesy: Gonzalez & Woods

Human perception of colors

- Cone cells in human retina are more sensitive to wavelengths of Red, Green and Blue lights
  - 65%, 33% and 2% cells, respectively
- CIE standard 1931 (*Commission Internationale de l'Eclairage*)
  - Blue light wavelength: 435.8 nm
  - Green light wavelength: 546.1 nm
  - Red light wavelength: 700 nm

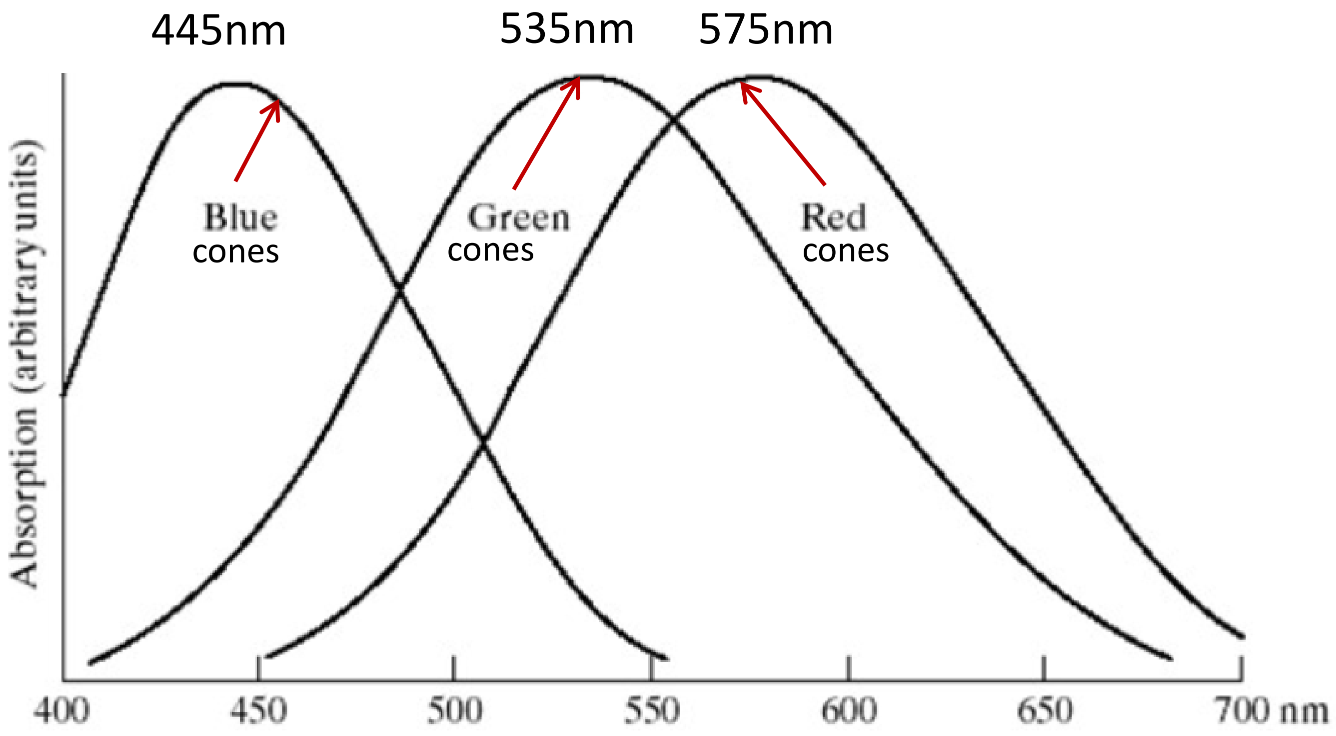


- In the visible light spectrum, there is a smooth transition from one color to another

Human perception of colors

- In practice, for every colored light, there is no single wavelegth. There is a range!
- A every cone cell has different sensitivity to different wavelength
- Humans perceive a color depending upon the combination of wavelengths reaching the eyes

year 1965



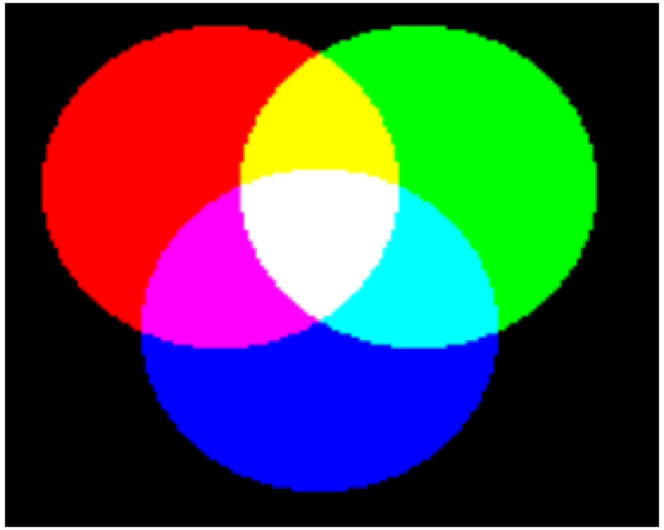
Plot courtesy: Gonzalez & Woods

Primary colors of light

- Lights of most colors in the visible spectrum can be generated by mixing Red, Green and Blue lights in appropriate proportion
- **Primary colors of light**: Red, Green, Blue (R,G,B)
- Result of mixing lights of two primary colors (wavelengths) in equal proportion is additive
  - Green light * Blue light => Cyan light (G*B=>C)
  - Red light * Blue light => Magenta light (R*B=>M)
  - Red light * Green light => Yellow light (R*G=>Y)
- **Secondary colors of light**: Cyan, Magenta, Yellow (C,M,Y)

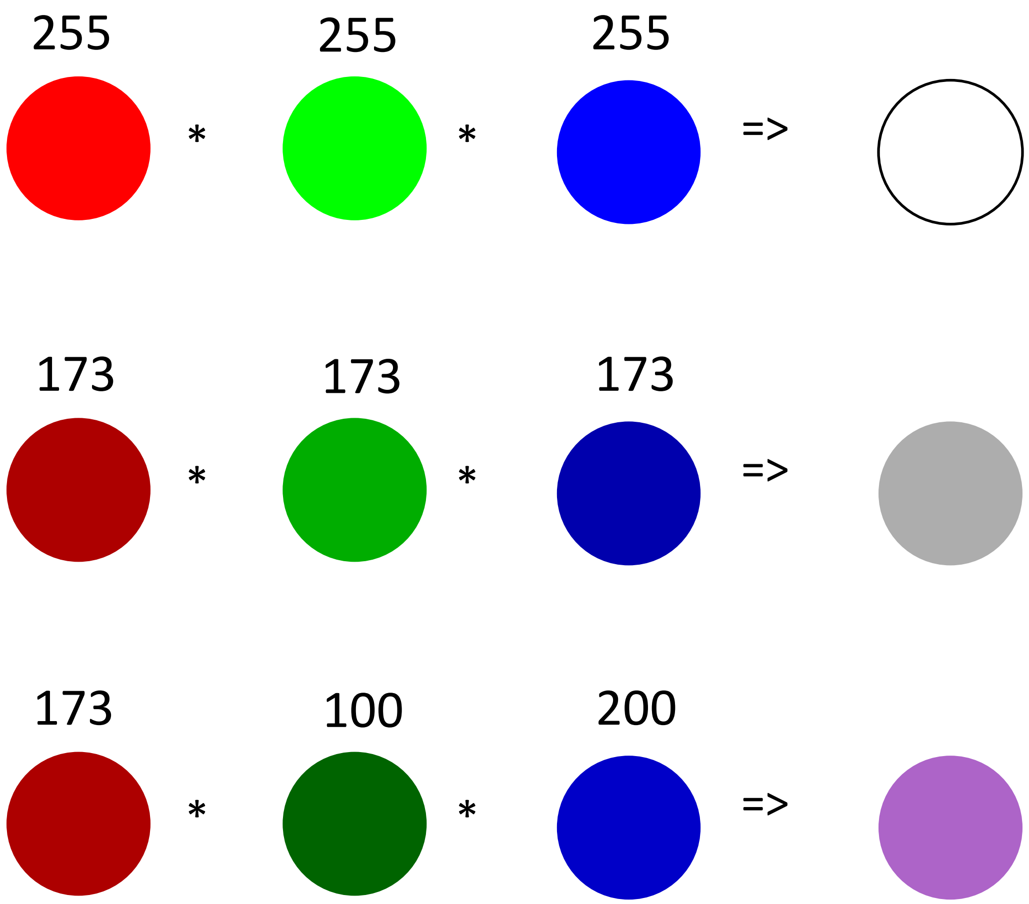
- Red light * Green light * Blue light => White light

mixing colored lights



Primary colors of light

- Red light * Green light * Blue light => White light



Primary colors of pigment

- Color appearance of an object depends on its reflectance property and color of light source

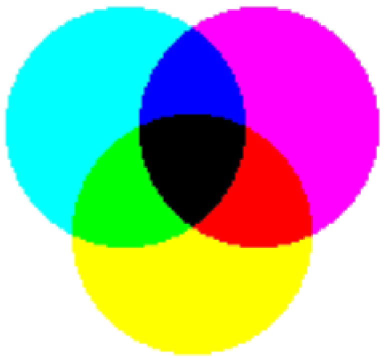
- Example: If reflectance property of an object is it reflects Red and Blue lights and absorbs rest
  - If White light (Red+Green+Blue) is incident, it will reflect Red and Blue, hence appear Magenta
  - If Cyan light (Green+Blue) is incident, it will reflect Blue

- Color of light source illuminating an object is different from how the object color appear (most of time) because some wavelengths are absorbed and some are reflected based on object property

Primary colors of pigment

- A pigment (e.g. printer ink) that absorbs Red, but reflects Green and Blue, it appears Cyan
- **Primary colors of pigment**: Cyan, Magenta, Yellow (C,M,Y)
- Result of mixing pigments of two primary colors is subtractive
  - Magenta pigment * Yellow pigment=> Red color (M*Y=>R)
  - Magenta pigment is absorbs Green light, Yellow pigment absorbs Blue light. So, a mix of them reflects only Red, thus appearing Red
  - Cyan pigment * Yellow pigment=> Green color (C*Y=>G)
  - Cyan pigment * Magenta pigment=> Blue color (C*M=>B)
- **Secondary colors of pigment**: Red, Green, Blue (R,G,B)
- Cyan pigment * Magenta pigment * Yellow pigment=> Black color

mixing colored pigments

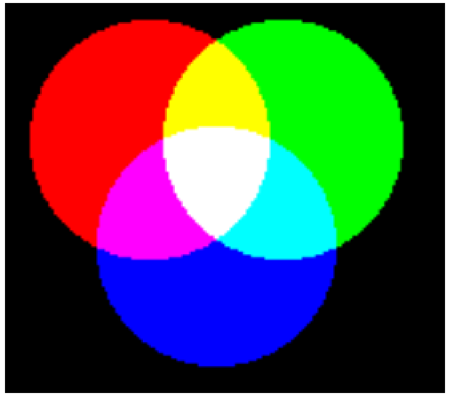


Light and pigment colors

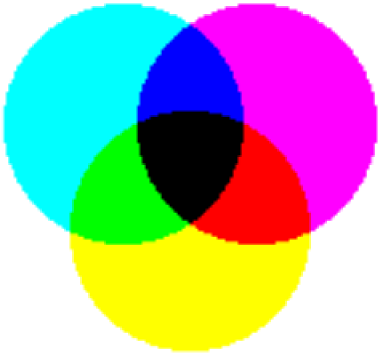
- R,G,B colored lights are used in display devices e.g., TV, monitor
  - Mixing R,G,B lights create lighter colors, this is suitable on dark screens

- C,M,Y colored pigments are used in printer to generate other colors
  - Mixing R,G,B pigments creates darker color shades on white paper. So, R,G,B pigments are not used in printers

mixing colored lights



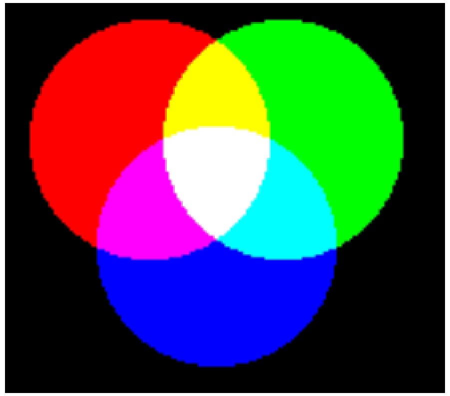
mixing colored pigments



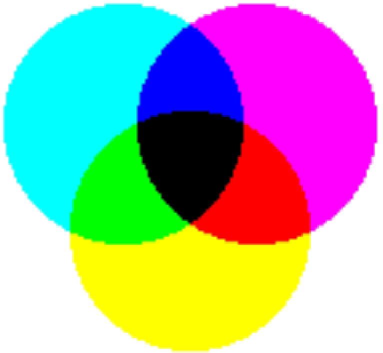
Light and pigment colors

- Important distinction:
  - Display device screens can be treated as light sources. Light coming out of it is perceived as the color of the scene displayed on the screen
  - But a pigment does not emit light. Depending on its property, it absorbs some wavelengths of the incident light and reflects some. Combination of these reflected wavelengths are perceived by human eyes as its color

mixing colored lights



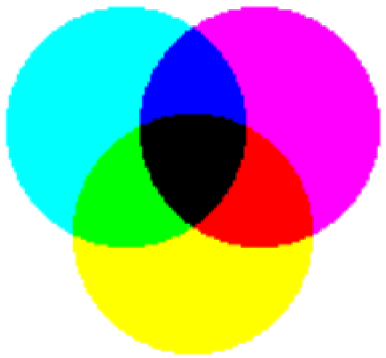
mixing colored pigments



Light and pigment colors

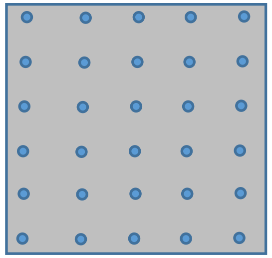
- Note: In printers, additionally Black pigments are used because
  - Mixing C,M,Y pigments together creates muddy black, not dark black
  - To enhance contrast
  - So, this is called CMYK (K for Black or “Key”)

mixing colored pigments

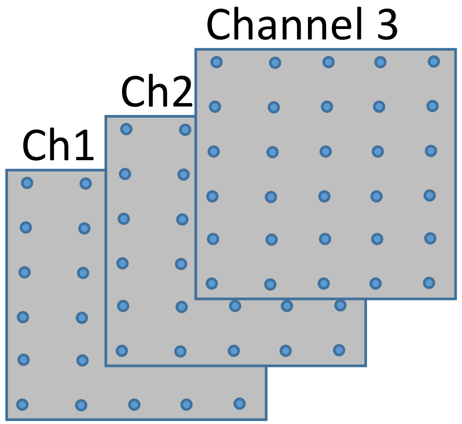


Color image - implementation perspective

- Array format of color image

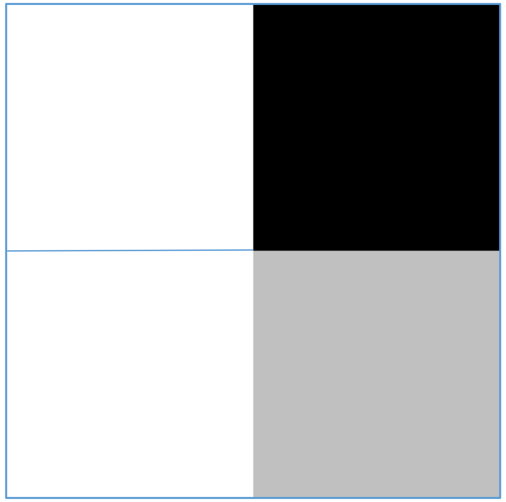


- Gray scale image with MxN pixels is represented using a 2D array

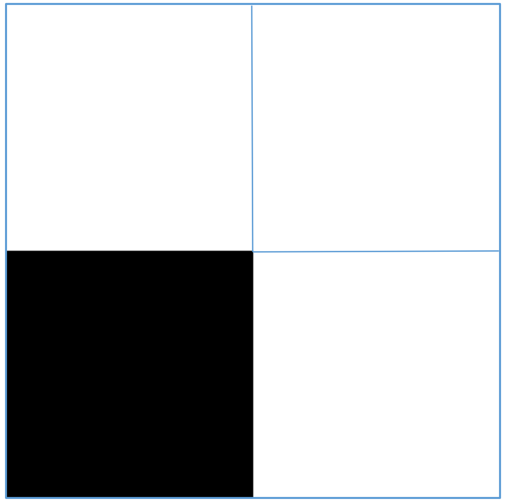


- Color image with MxN pixels is represented using a 3D array (A) i.e. three 2D arrays stacked: Red, Green, Blue channels

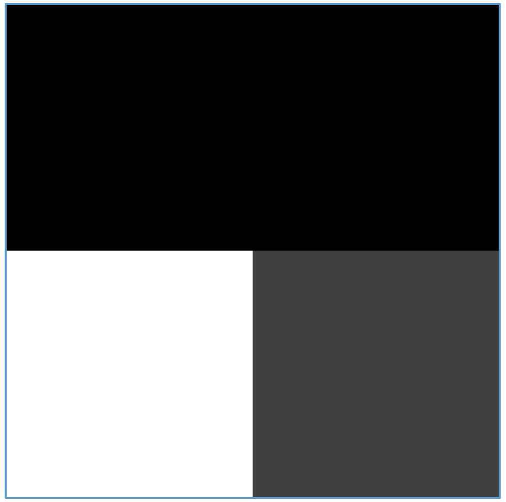
Color image - visualize R,G,B channels



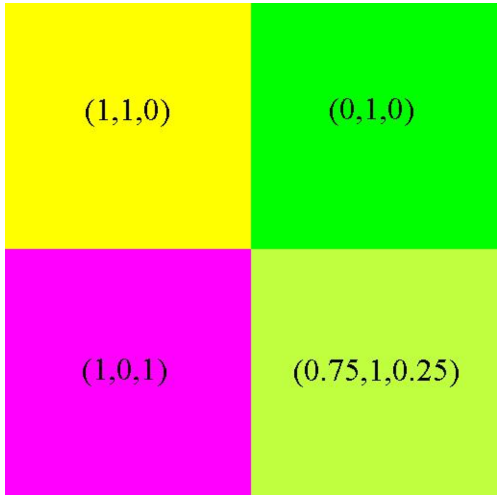
R Channel



G Channel



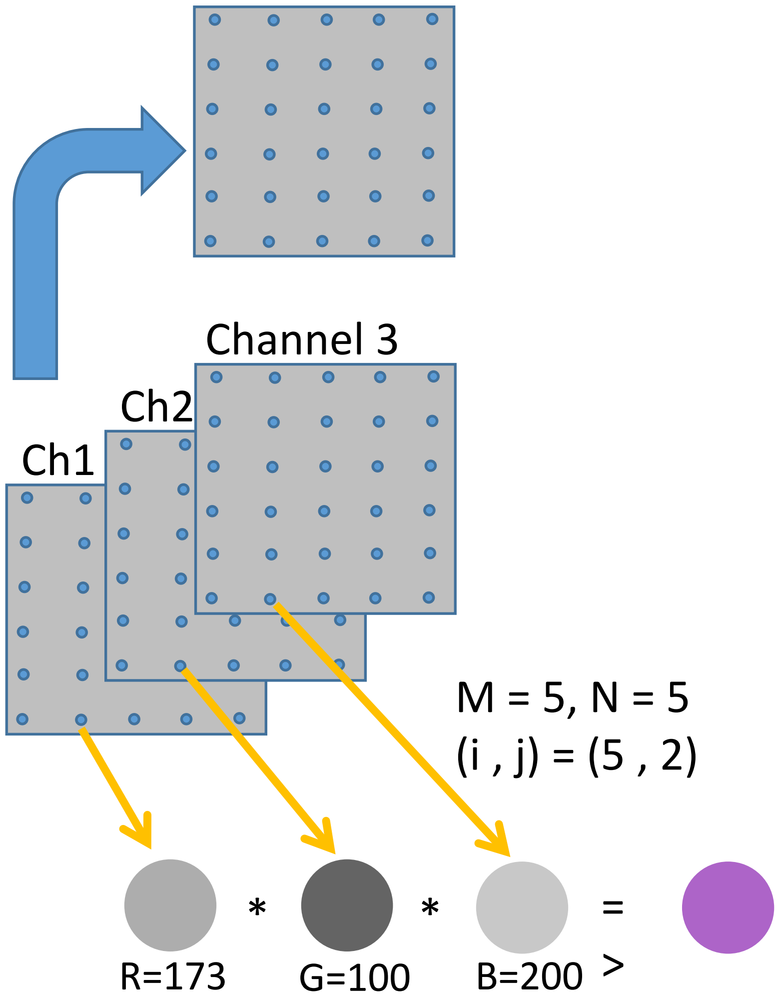
B Channel



Color image - implementation perspective

- R = A(i,j,1)   G = A(i,j,2)   B = A(i,j,3)
- R,G,B values are intensity of primary color wavelengths present
- Display device mixes primary colors of corresponding intensities and produces a color that we visualize

- If this color image is converted to a gray scale image B which will be a 2D array, intensity at every pixel will be average of R,G,B. It will lose all color information
- B(i,j) = [A(i,j,1) + A(i,j,2) + A(i,j,3)]/3
- So, 'mixing' operation is different from 'summing' R,G,B values



Color image - visualize R,G,B channels

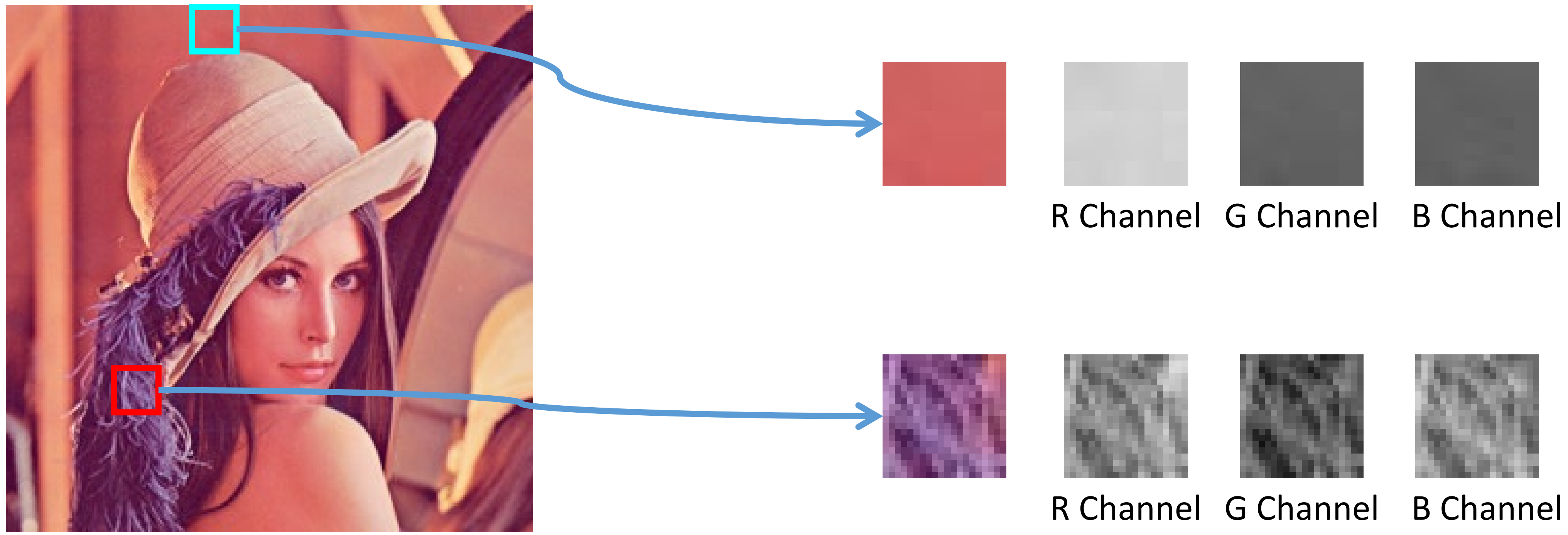


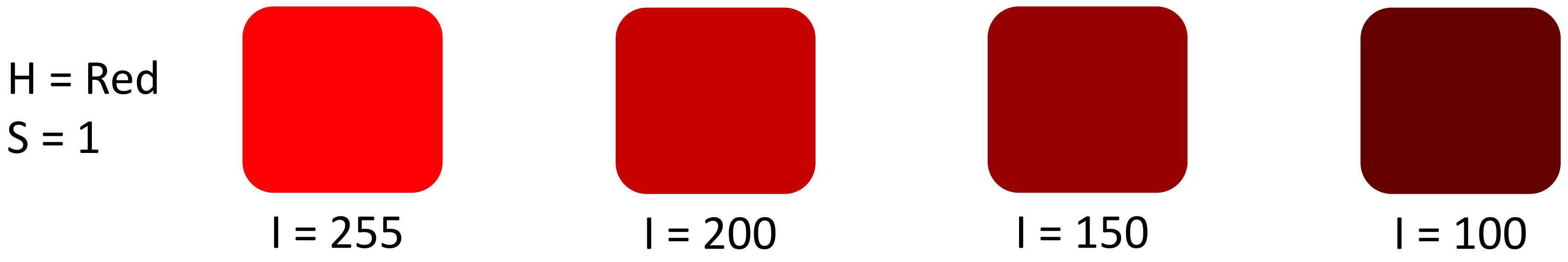
Image courtesy: internet

Hue, saturation, intensity

- Humans perceiving colors is a psycophysiological problem

- While looking at a color, we do not think about how much of it has R, G or B wavelength
    - We perceive a color based on its **Hue** (H), **Saturation** (S) and **Intensity** (I)

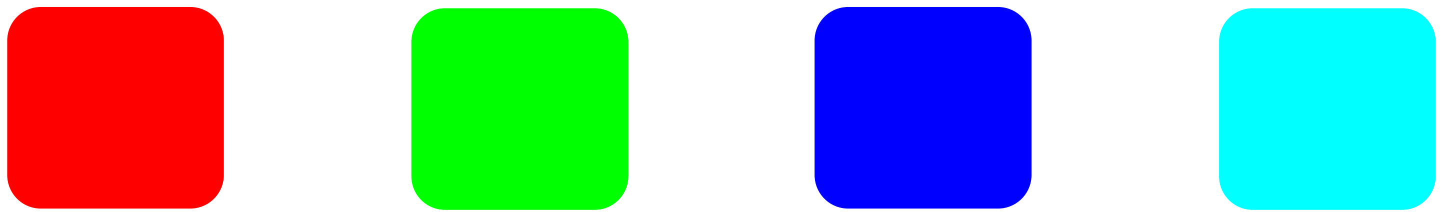
- Intensity: brightness of the light; lower intensity indicates darker shade of color



Hue, saturation, intensity

- Hue: dominant color (wavelengths) present in the light

I = 255
S = 1



H = Red   H = Green   H = Blue   H = Cyan

- Better way to represent Hue in numeric values will be shown later

Hue, saturation, intensity

- Dilution: the amount of White light which has been mixed with that color (Hue) to make it diluted
- Saturation: an inverse measure of dilution; full saturation (S=1) indicates no dilution by white light
  - Colors in the visible spectrum are fully saturated

- Example: mixing White light (Intensity = 80, i.e. R=80,G=80,B=80) with fully saturated Red light (R=120,G=0,B=0) yields dark Pink light (R=200,G=80,B=80) with Saturation = 0.33

Hue, saturation, intensity

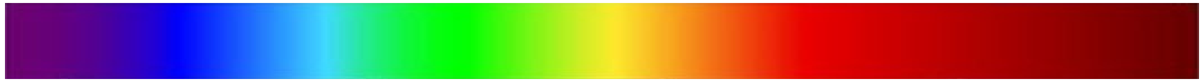
- Saturation computation formula will be shown later

- Hue and Saturation give color sensation; this is called *Chromaticity*

- Lights in visible spectrum are chromatic

- *Achromatic* light does not give color sensation: Hue=0, Saturation=0. Its only property is Intensity. In gray scale images, only pixel Intensity from Black through Gray to White is considered; no color

Chromaticity diagram



- Amount of Red, Green, Blue lights mixed to generate light of another color is called *Tristimulus* (X, Y, Z)

- Normalized values $x = X/(X+Y+Z)$
  $y = Y/(X+Y+Z)$ $z = Z/(X+Y+Z)$

- Chromatic coefficients
  $x + y + z = 1$

- In the *Chromaticity Diagram,* any color (point) can be specified using chromatic coeffs
  $(x, y, z = 1-(x+y))$

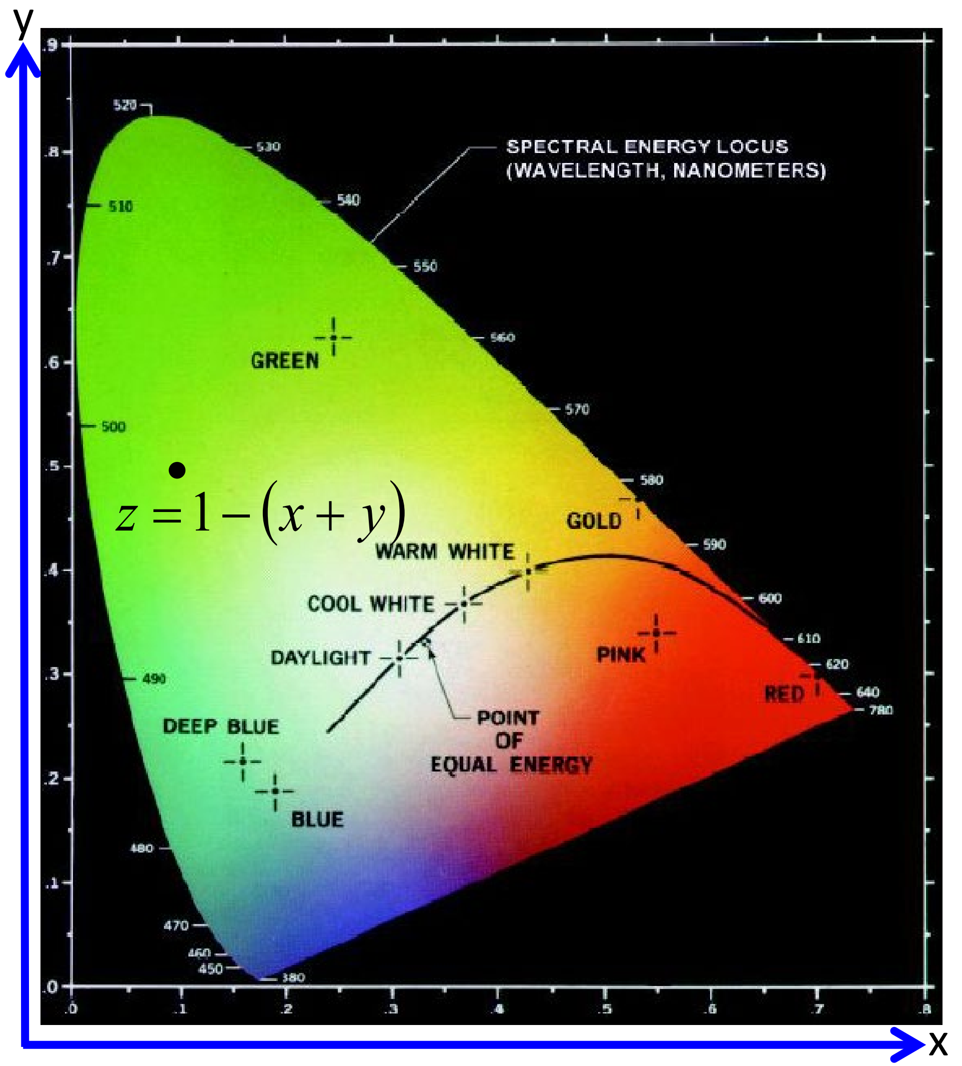


diagram courtesy: Gonzalez & Woods

Chromaticity diagram

- Here color specifies Hue and Saturation only

- Line joining any two points specify all colors that can be generated by mixing those two colors in different proportions

$$C_3 = \alpha C_1 + (1-\alpha)C_2$$

- Triangular region formed by joining any three points specify all colors that can be generated by mixing those three colors in different proportions

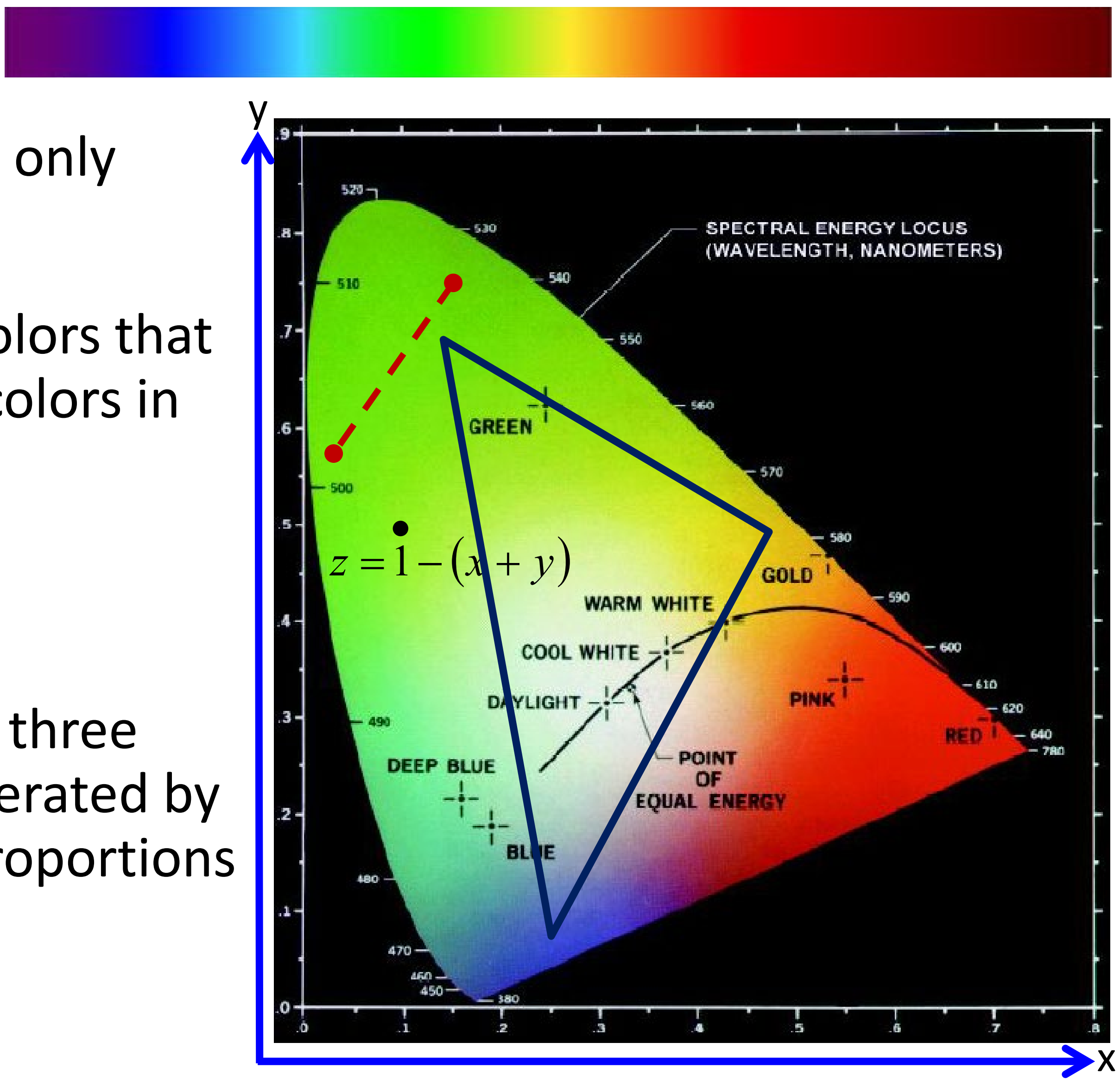
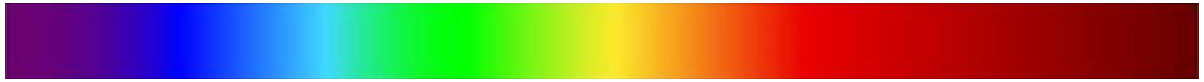


diagram courtesy: Gonzalez & Woods

Chromaticity diagram



- Colors at boundary are fully saturated (S=1)
- White is completely unsaturated (S=0)

$$x = y = z = \frac{1}{3}$$

- If a line joins any point (a color with some Hue) with White, the colors on the line are more unsaturated (with same Hue) as moved towards White because that color becomes more diluted by White
- Colors close to White have small Saturation value

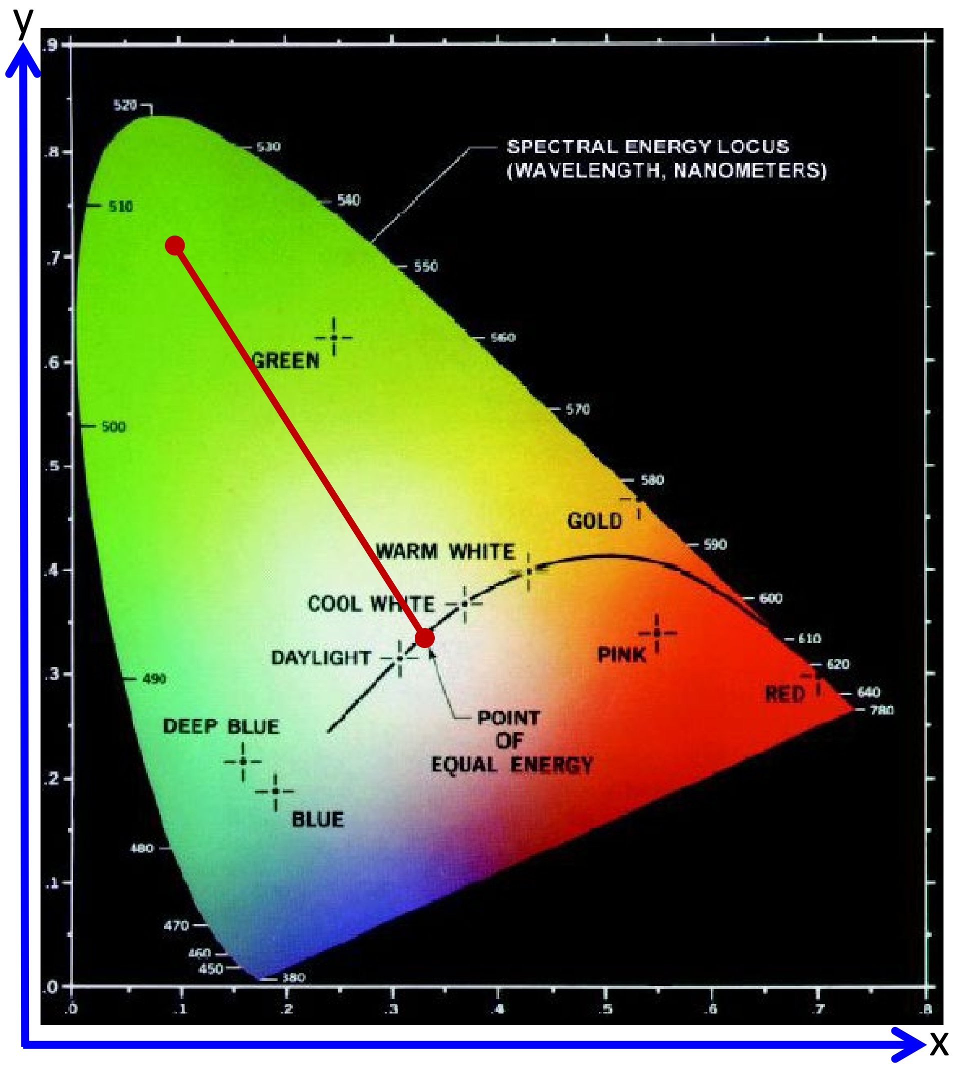


diagram courtesy: Gonzalez & Woods

Chromaticity diagram

- Colors inside have different Saturation values (S<1) because any point is on a line joining White and a boundary point

- Since chromaticity diagaram is not triangular, we cannot generate all possible colors by mixing R,G,B if they had exactly one particular wavelength each

- So, each of R,G,B light is comprised of a range of wavelengths

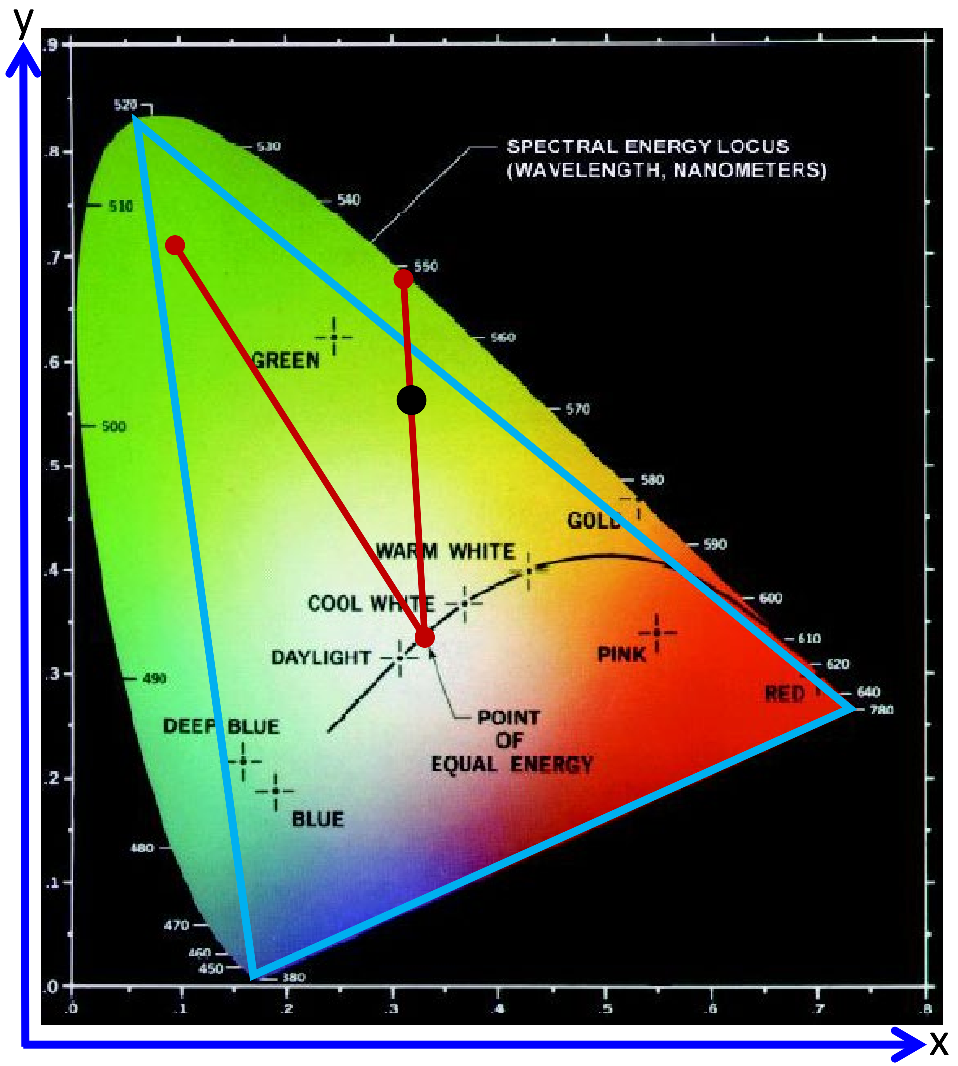


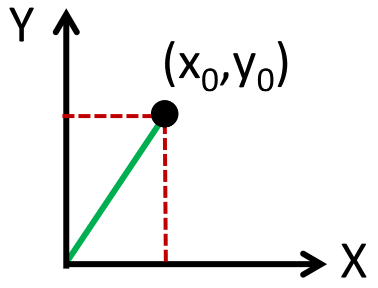
diagram courtesy: Gonzalez & Woods

Color model

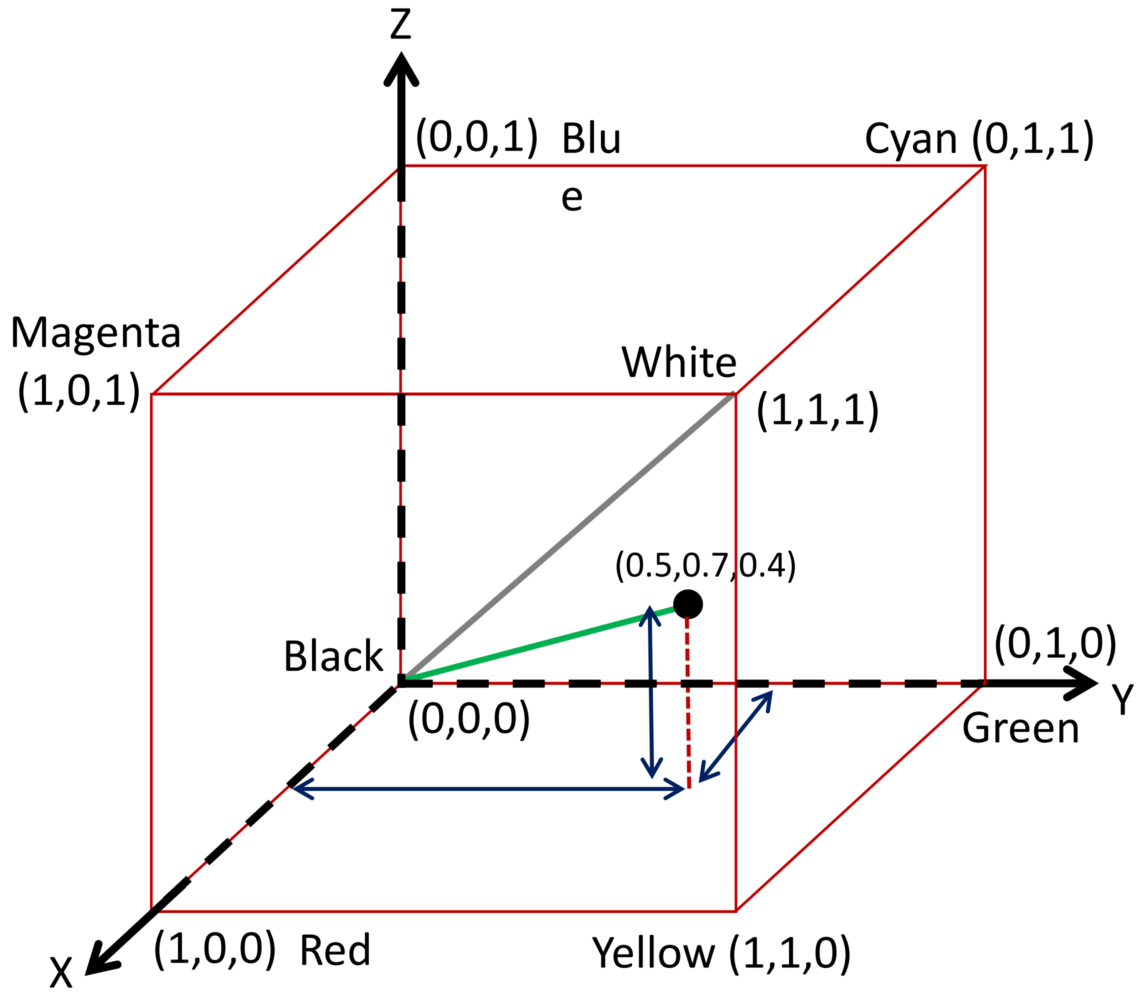
- RGB model: display devices
- CMY or CMYK model: printers
- HSI model: perception oriented
- H,S have chromatic information and I doesn't, i.e. H,S are decoupled from I
- Algorithms developed for gray scale images can be applied I component of HSI

Color model

- A space where any color is specified a point that specifies the proportions of the primaries present in that color

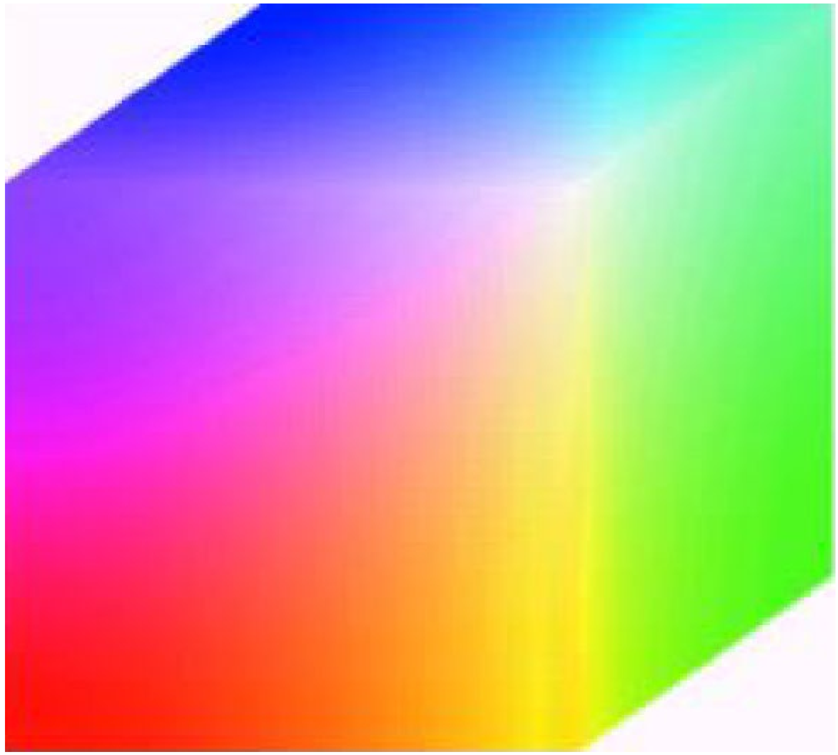


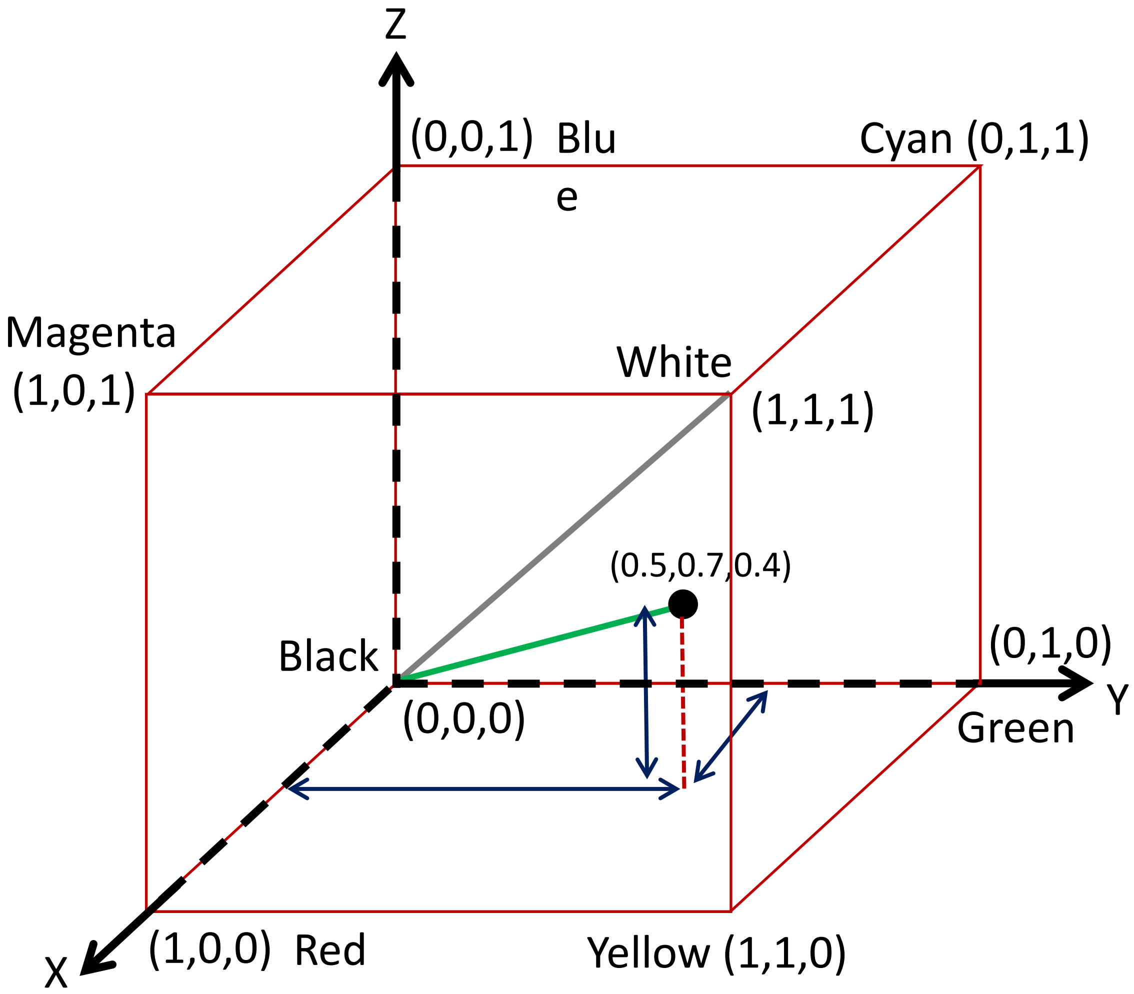
- RGB model is built on cartesian coordinate system
- Line joining Black and White indicates Gray Scale; no color (achromatic)
- Any color is a vector joining that point and origin



unit cube with normalized colors

RGB color model

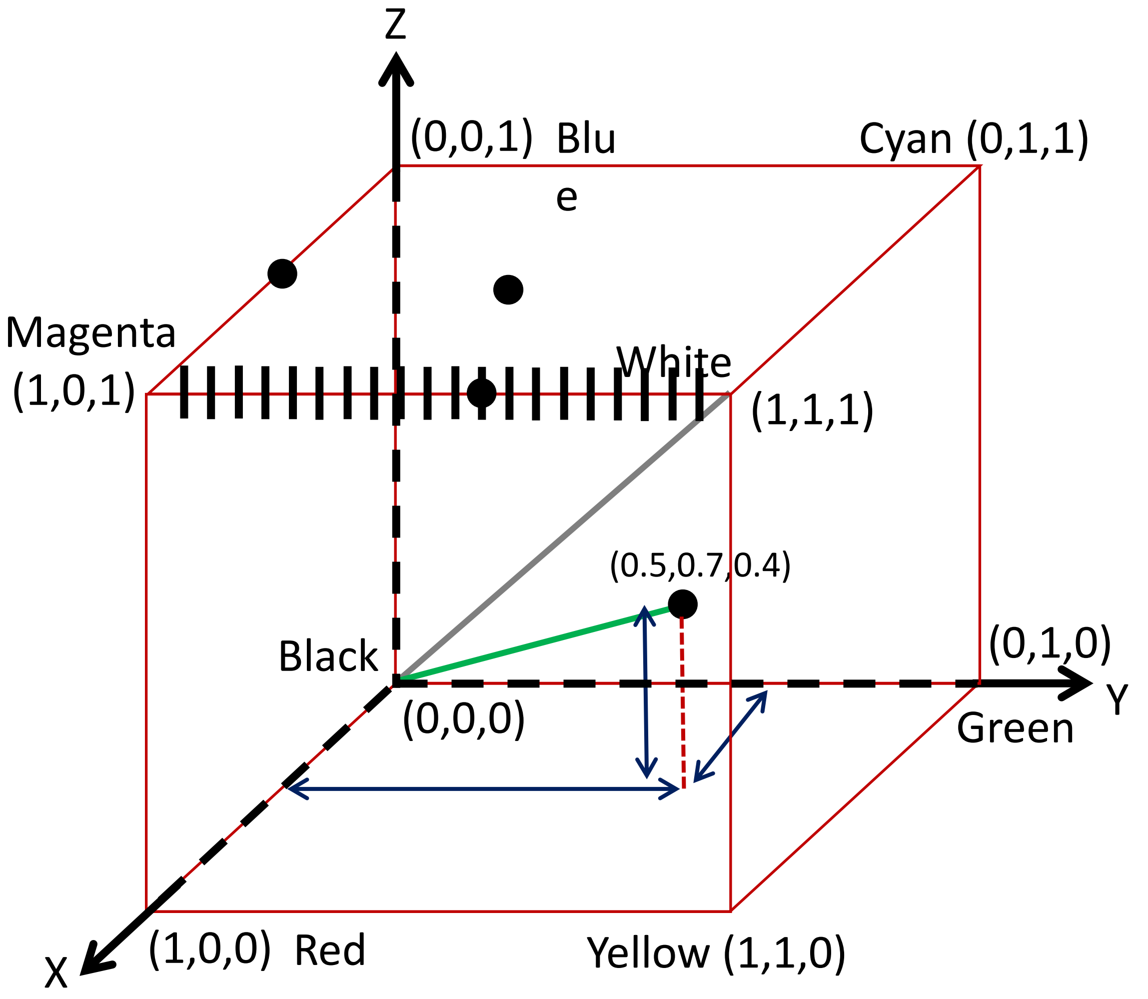




unit cube with normalized colors

RGB color model

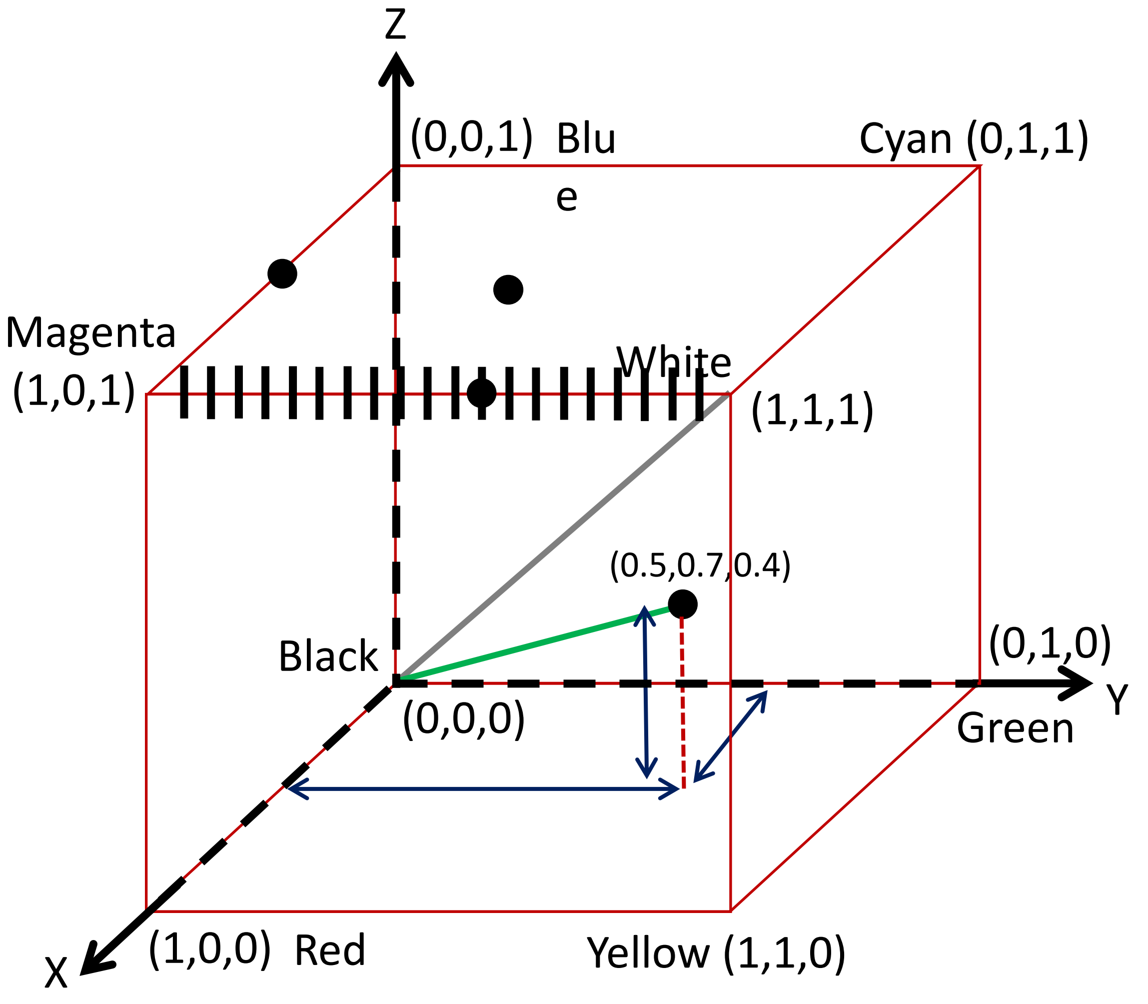
- Normalized range [0, 1]
- Quantization: divide into 256 equal parts
- Unnormalized range [0, 255] i.e. [0, 2^8-1] for "unsigned integer 8-bit" system or *Uint8* format
- White = (255 255 255); Black = (0 0 0)
- integer 253 = binary 11111101 (8 bits)



unit cube with normalized colors

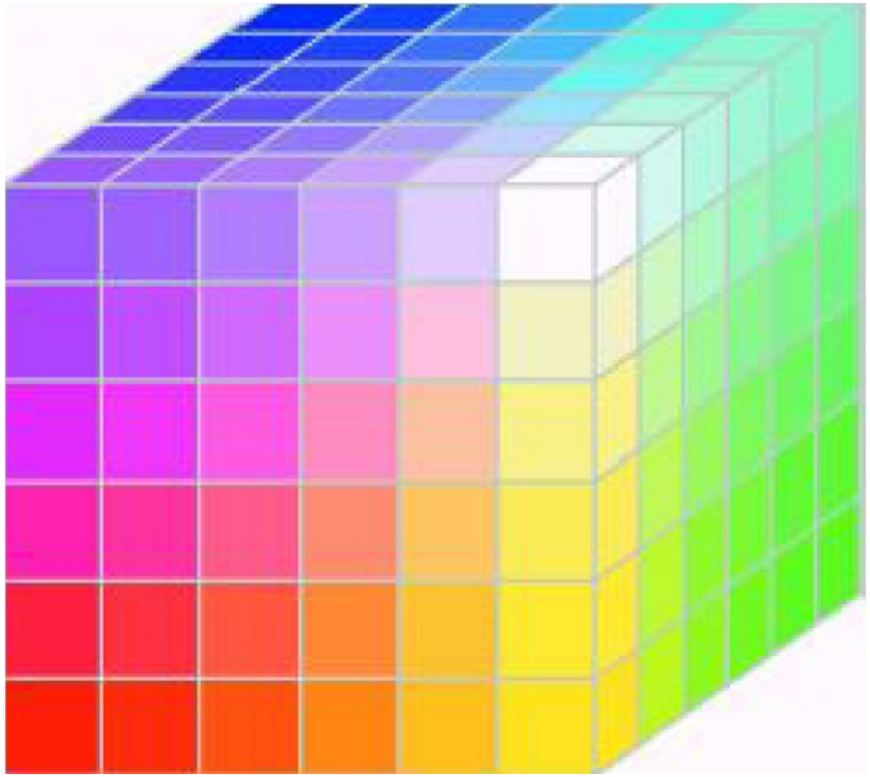
RGB color model

- Total 24 bits to represent R,G,B values of every pixel
- Total 2^{24} (i.e. 256^3) number of colors can be generated from this cube!
- Unnormalized range [0, 65535] i.e. [0, 2^{16}-1] for *Uint16* format and [0, 2^{32}-1] for *Uint32* format
- Some displays cannot show so many colors
    - *Safe RGB model*: only 216 colors used
    - Quantize each axis in 6 equal parts

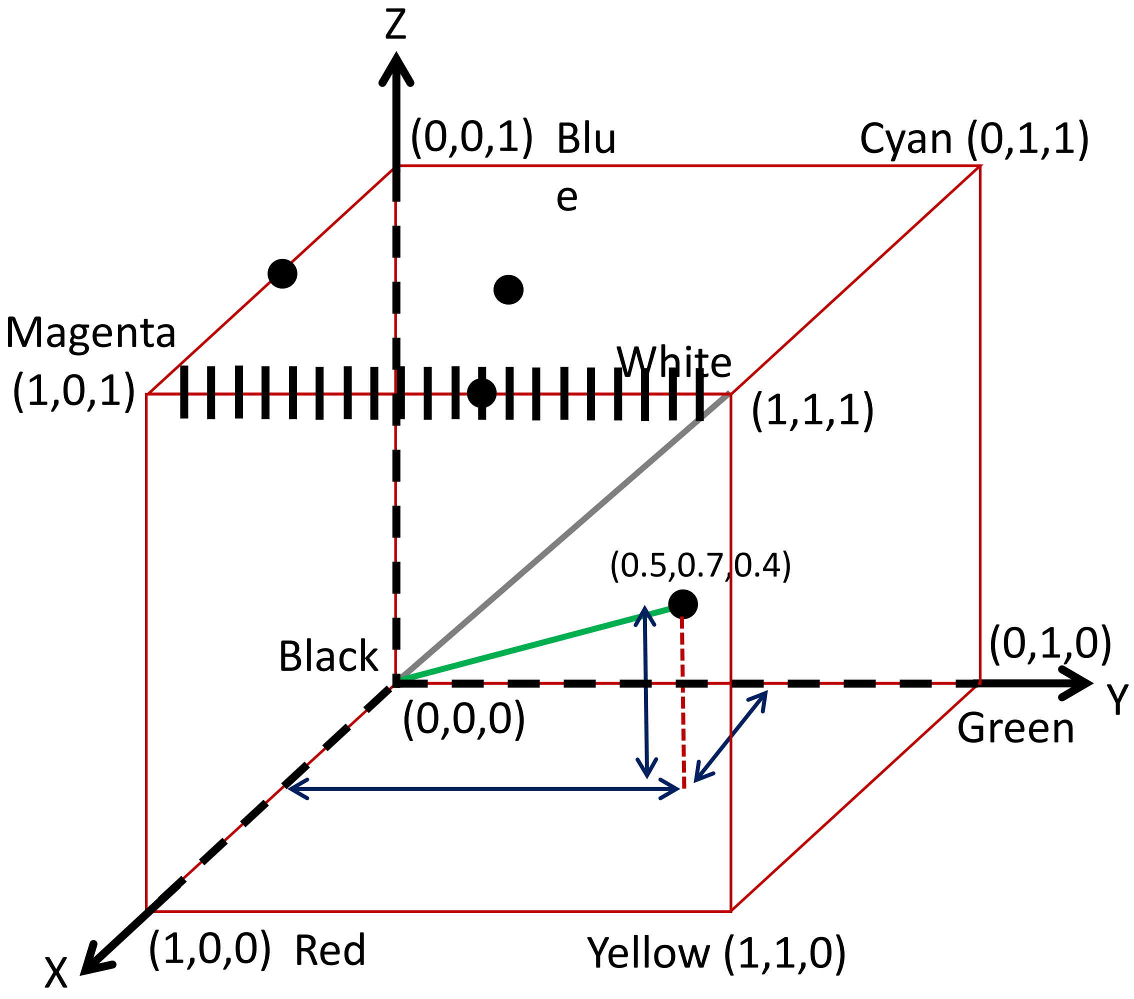


unit cube with normalized colors

Safe RGB color model



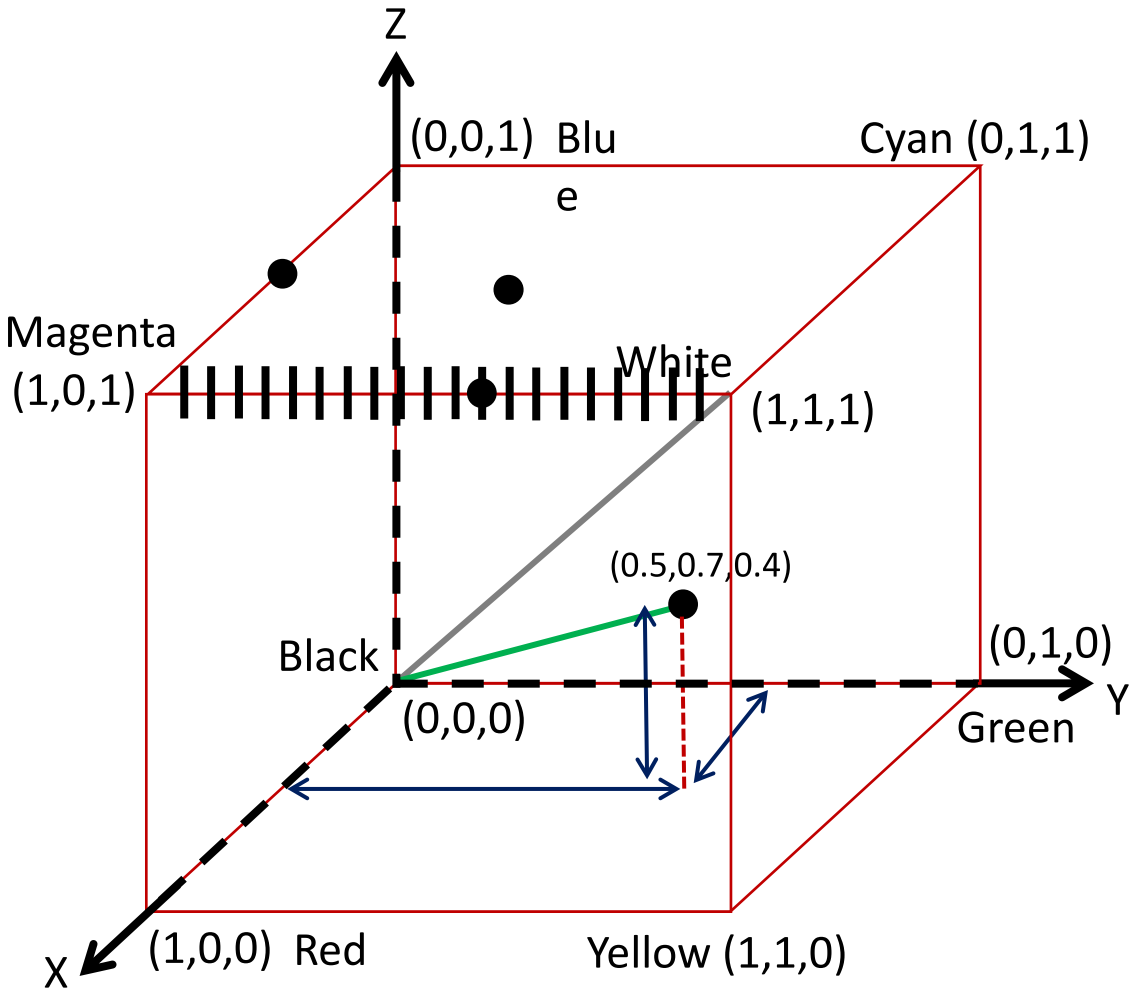
216 different colors



unit cube with normalized colors

Safe RGB color model

- Six intensity values 0,51,102,153,204,255 used for each channel
- Hexadecimal format used to represent
- Corresponding hex numbers 00,33,66,99,CC,FF
- Red : $(255,0,0)_{10} = (FF0000)_{16}$



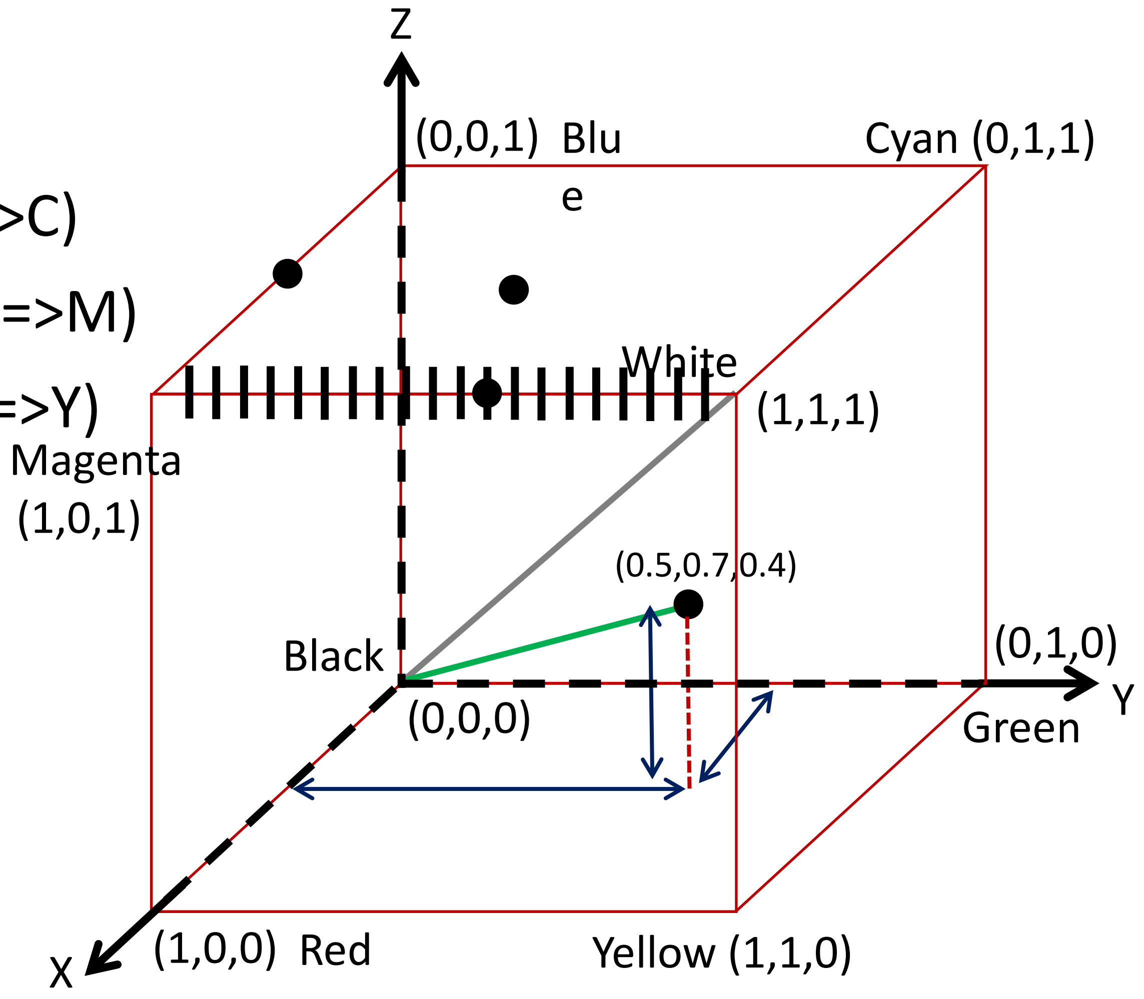
unit cube with normalized colors

RGB to CMY model transformation

- Recap
- Green light * Blue light => Cyan light (G*B=>C)
- Red light * Blue light => Magenta light (R*B=>M)
- Red light * Green light => Yellow light (R*G=>Y)

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

- Other color models
  - Y,Cb,Cr
  - L,a,b



unit cube with normalized colors

RGB to CMYK model transformation

- CMY model

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

- CMYK model

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

$$K = bK_b$$

$$K_b = \min\{1-R, 1-G, 1-B\}$$

Undercolor removal factor $0 \le u \le 1$

Blackness factor $0 \le b \le 1$

Gamma corrected color components

- In television system: Cathode ray tube generated light has amplitude proportional to the input signal raised to the power of 2 to 3

- Gamma correction needed before transmission, possibly at the output of the television camera

$$\tilde{K} = c_1 K^{c_2} + c_3, \text{ for } k \geq b$$

$$\tilde{K} = c_4 K, \text{ for } 0 \leq k < b$$

$$0.33 \leq c_2 \leq 0.45$$

YIQ and YUV models

- These models are used in television system signal transmission (NTSC and PAL)

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \approx \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5959 & -0.2746 & -0.3213 \\ 0.2115 & -0.5227 & 0.3112 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad R{=}1, G{=}1, B{=}1 \text{ imply } Y = 1, I = 0, Q{=}0$$

- Y is Luma and I (in-phase), Q (quadrature) are chroma components
- Y signal is compatible with monochrome receivers
- Advantage: Bandwidth of I, Q signals can be limited without noticeable degradation
- Using quadrature amplitude modulation technique, transmission b/w restricted
- U, V to I, Q conversion

$$I = -U\sin 33^\circ + V\cos 33^\circ$$

$$Q = U\cos 33^\circ + V\sin 33^\circ .$$

YCbCr model

- Used in video and digital photography systems

$$
\begin{aligned}
Y' &= 16 + (65.481 \cdot R' + 128.553 \cdot G' + 24.966 \cdot B') \\
C_B &= 128 + (-37.797 \cdot R' - 74.203 \cdot G' + 112.0 \cdot B') \\
C_R &= 128 + (112.0 \cdot R' - 93.786 \cdot G' - 18.214 \cdot B')
\end{aligned}
$$

- Y' (gamma corrected) is luma, *Cb* is blue-difference, *Cr* is red difference channel

- Y' range [16, 235]. *Cb* and *Cr* range [16, 240]

- Errors in *Cb*, *Cr* due to compression, bandwidth reduction etc. are not perceived significantly by human

CIE L*a*b* model

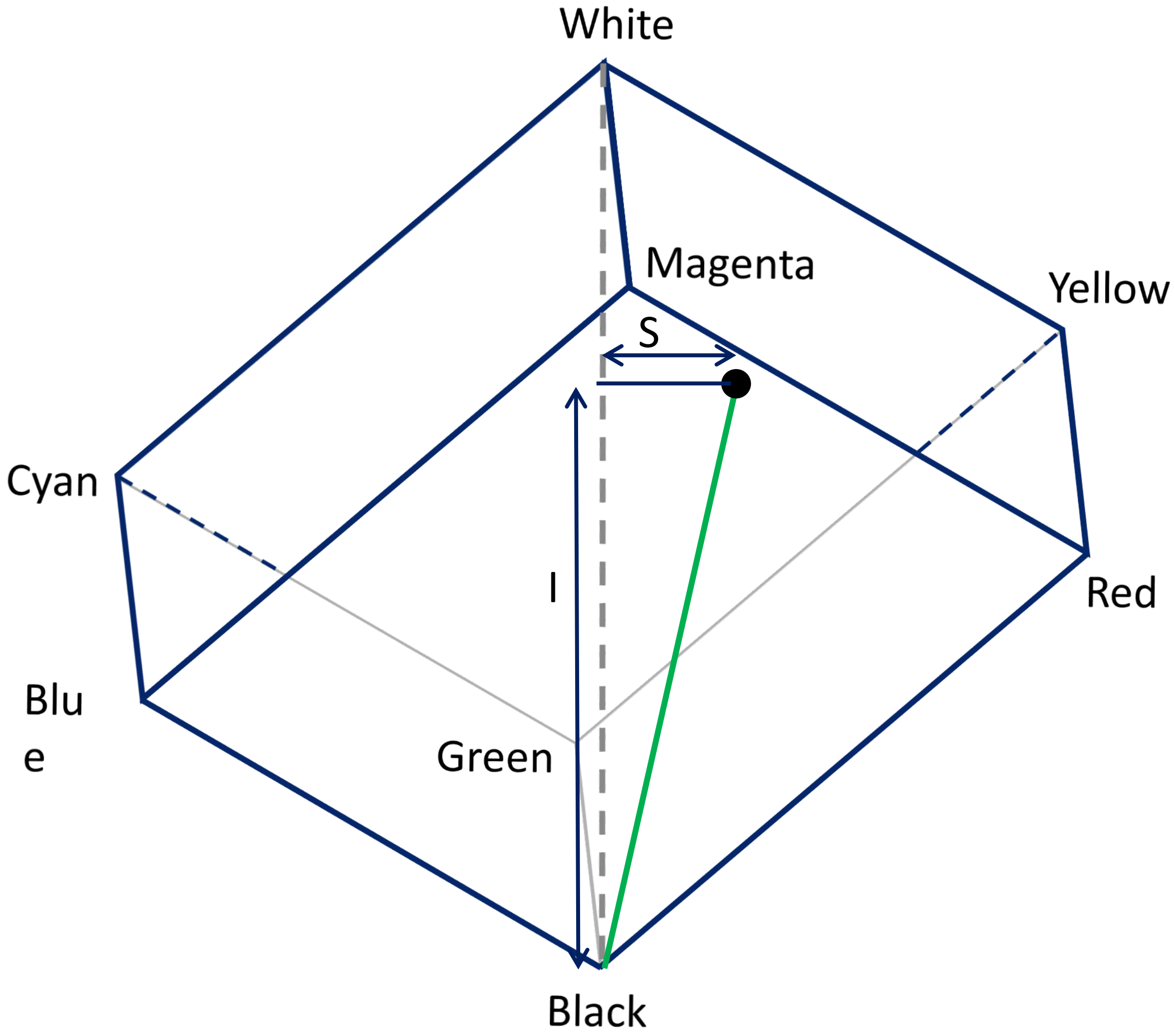
- Related to human visual system. Like HSI, intensity can be decoupled
- If values are changed, visually perceived change is also similar
- Useful for image manipulation and compression

$$L^\star = 116\, f\left(\frac{Y}{Y_\mathrm{n}}\right) - 16$$
$$a^\star = 500\left(f\left(\frac{X}{X_\mathrm{n}}\right) - f\left(\frac{Y}{Y_\mathrm{n}}\right)\right)$$
$$b^\star = 200\left(f\left(\frac{Y}{Y_\mathrm{n}}\right) - f\left(\frac{Z}{Z_\mathrm{n}}\right)\right)$$

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise} \end{cases}$$
$$\delta = \frac{6}{29}$$

- L^* is lightness, *a** is redness-greenness, *b** is yellowness-blueness
- X, Y, Z are tri-stimulus for any color and X_n, Y_n, Z_n are tri-stimulus for White
- L^* range [0, 100]. *a** and *b** range [-128, 127]

RGB to HSI model transformation



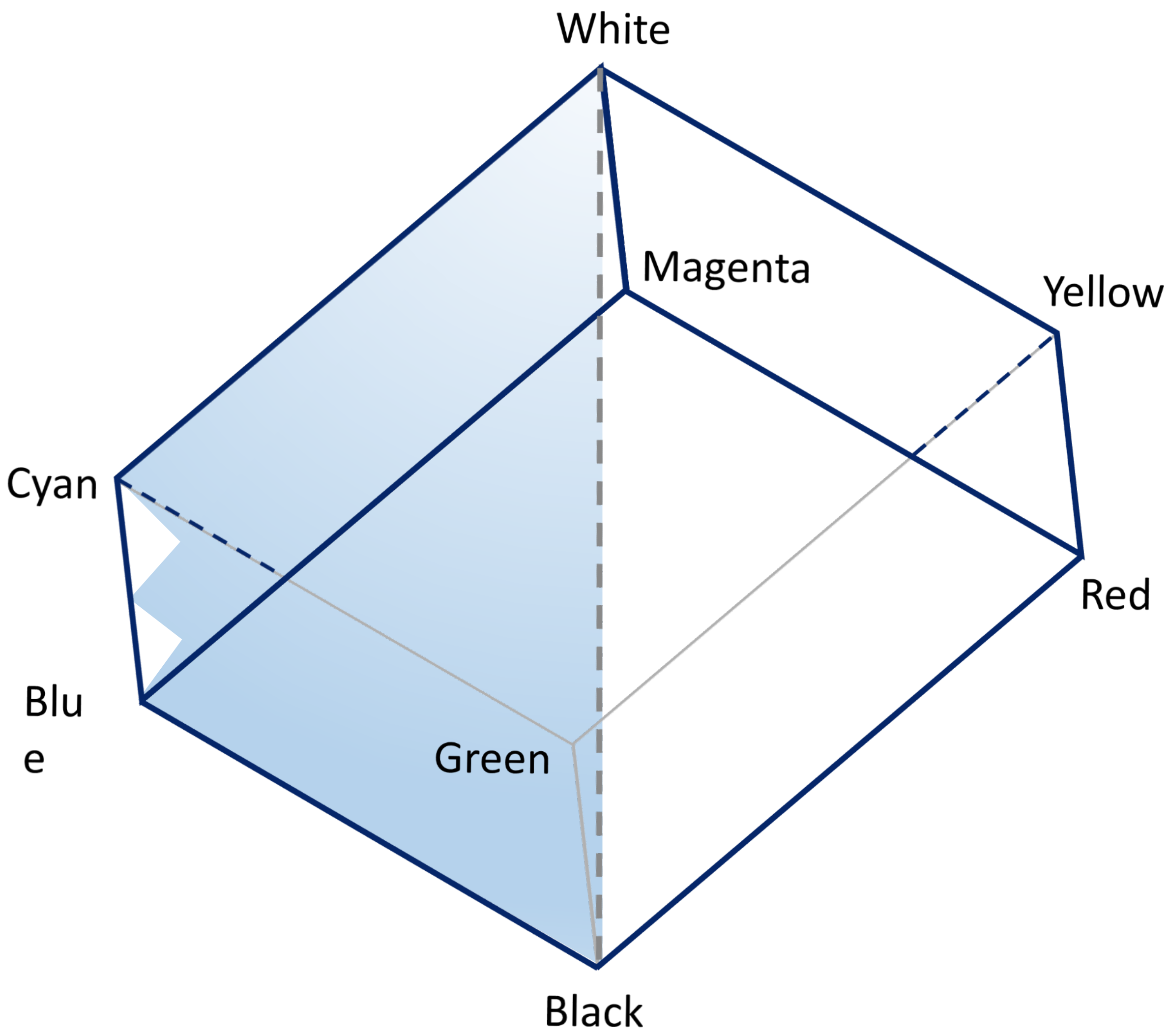
- Intensity $I = (R + G + B)/3$
- (R,G,B) = (200, 50, 170)

  = (150, 0, 120) + (50, 50, 50)

  = Fully saturated color + White
- Intensity of White that causes dilution is min(R,G,B) = 50
- Dilution $\frac{1}{I}[\min(R,G,B)]$
- Saturation $S = 1 - \frac{1}{I}[\min(R,G,B)]$

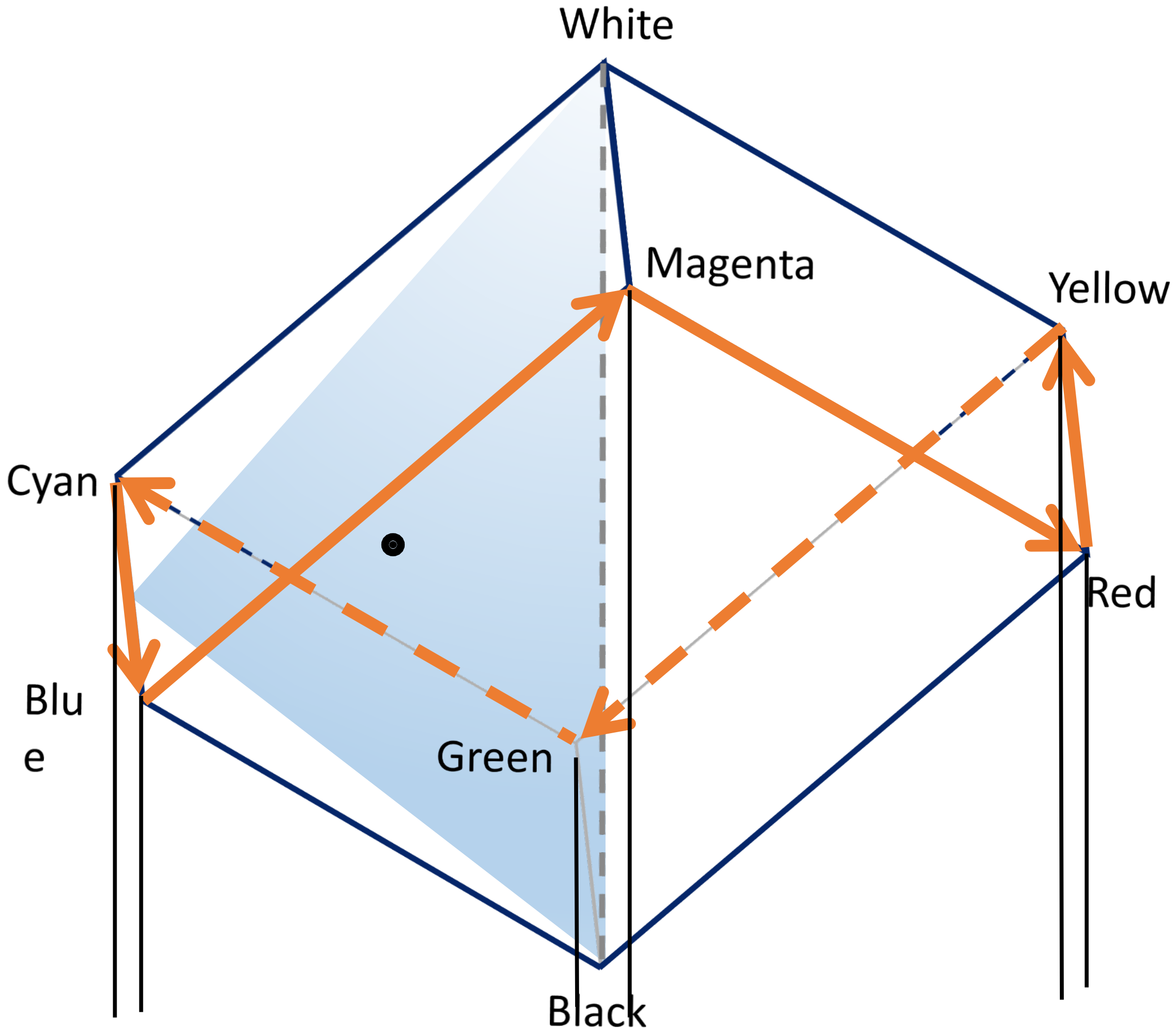
$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)]$$

RGB to HSI model transformation

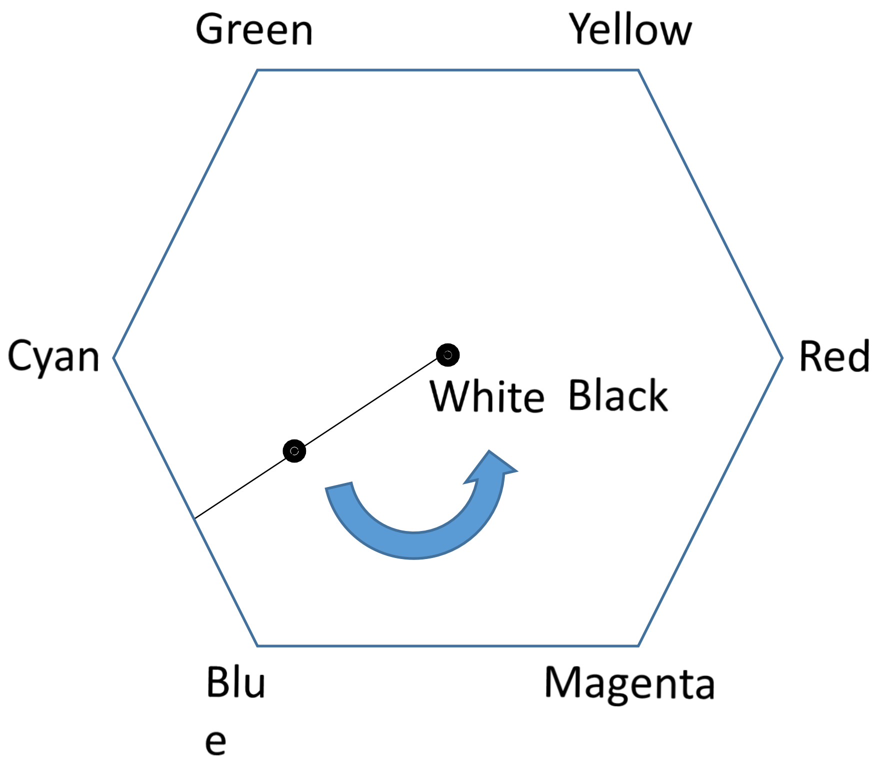


- Red, Yellow, Green, Cyan, Blue, Magenta are fully saturated
- Consider the triangular slice (plane) joining White-Cyan-Black-White
- From chromaticity diagram, colors inside a triangle joining Black, White, Cyan are 'diluted Cyan' (with same Hue)
- Rotate this trianglular plane around Black-White axis along the path C-B-M-R-Y-G-C
- This rotation covers entire volume of cube i.e. all possible (R,G,B) pts

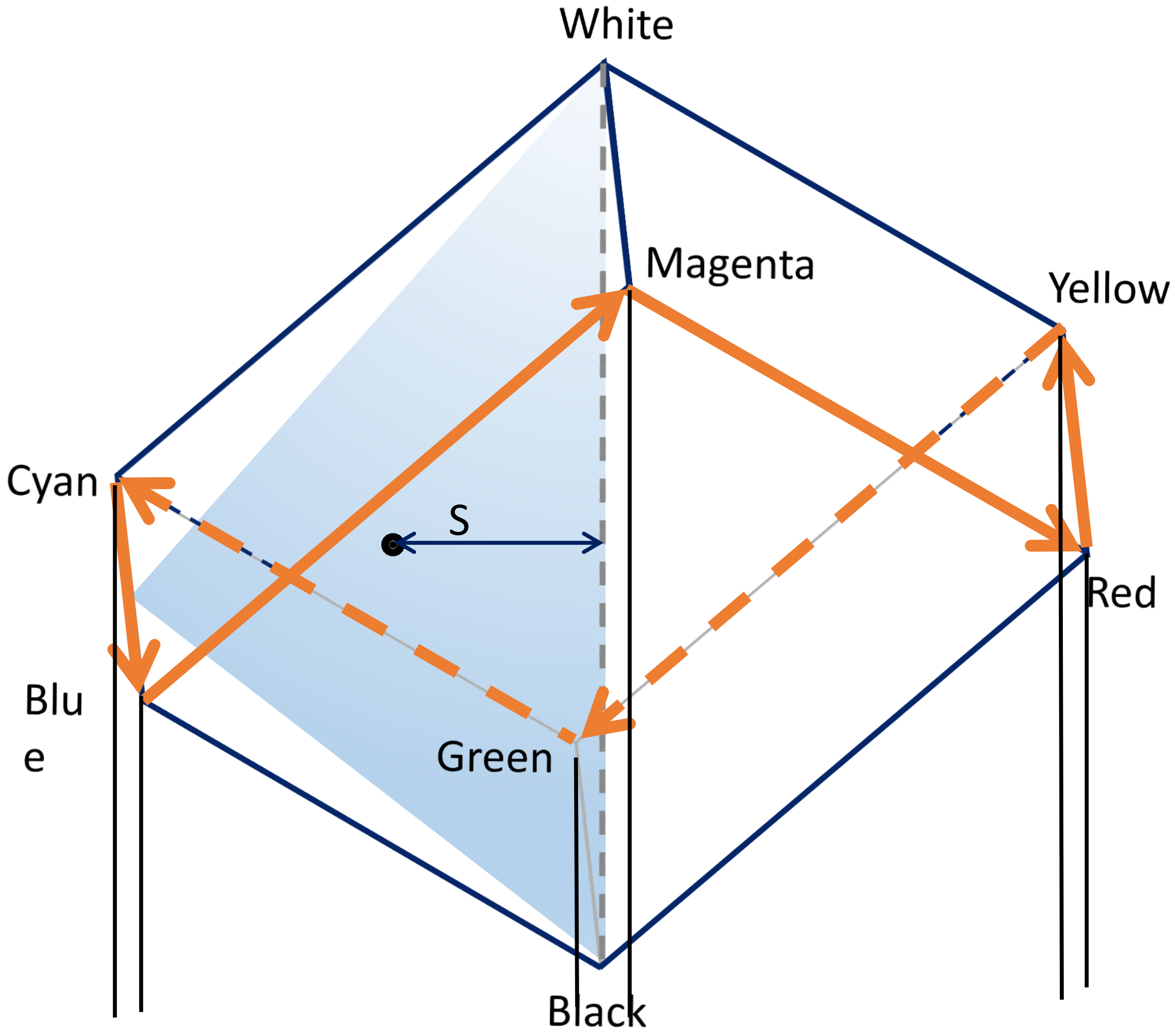
RGB to HSI model transformation



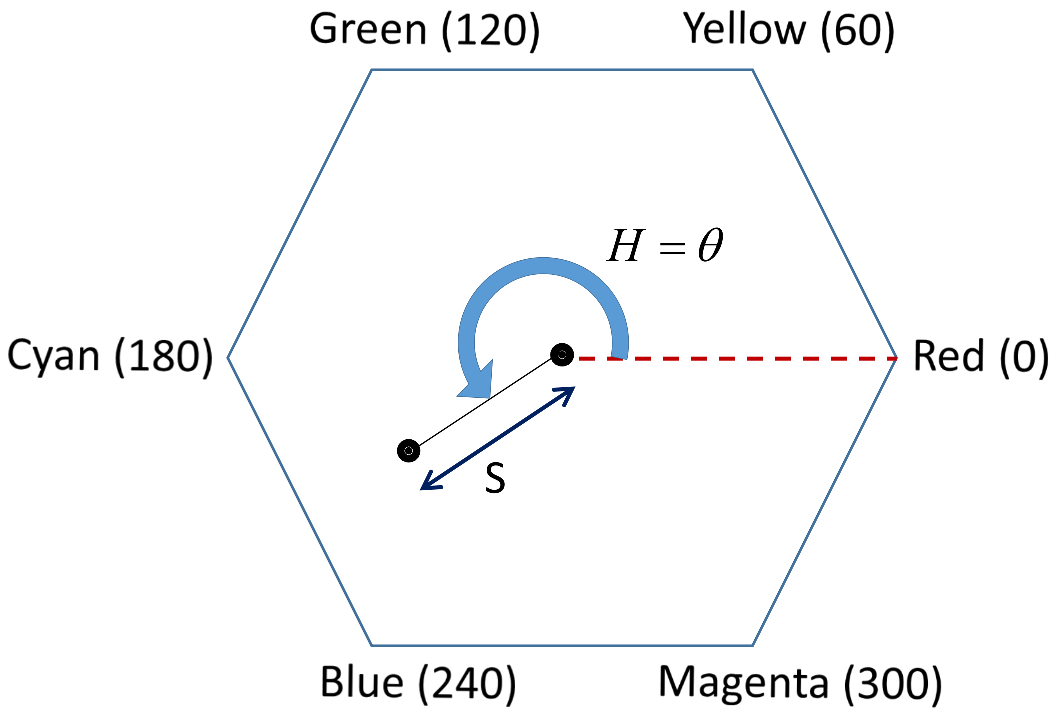
- Volume projects to hexagon
- Triangular plane projects to straight line
- Any pt on the plane maps to a pt on line



RGB to HSI model transformation



- Saturation is the length of the projected line
- S=1 for colors on hexagon boundary
- Hue is the angle this vector makes with reference



RGB to HSI

$$H = \begin{cases} \theta & \text{if} \quad B \le G \\ 360^\circ - \theta & \text{if} \quad B > G \end{cases}$$

$$\theta = \cos^{-1}\left\{ \frac{0.5\left[(R-G)+(R-B)\right]}{\left[(R-G)^2 + (R-B)(G-B)\right]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{R+G+B}\left[\min(R,G,B)\right]$$

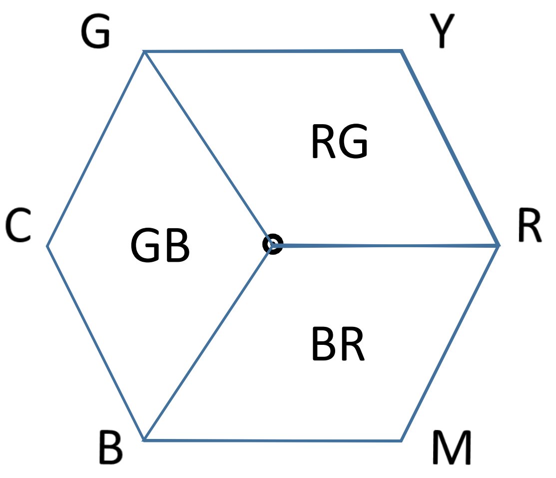
$$I = \frac{1}{3}(R+G+B)$$

- sign of inverse cosine depends on the quadrant, hence Hue computation is split into two ranges

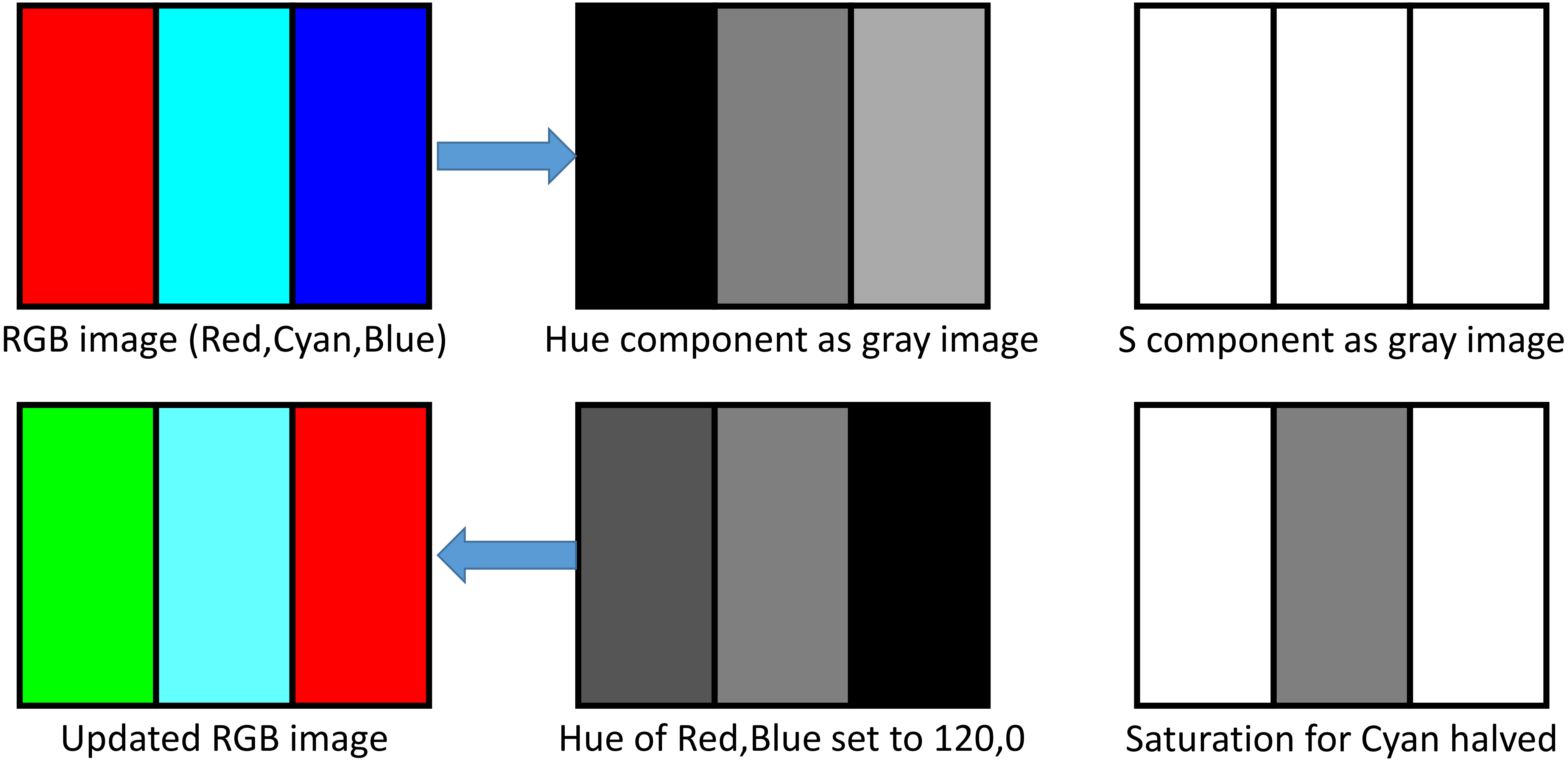
HSI to RGB

- RG sector $(0° \leq H < 120°)$ ➡ $B = I(1-S)$ $R = I\left[1 + \frac{S\cos H}{\cos(60^{\circ} - H)}\right]$ $G = 1-(R+B)$
- GB sector $(120° \leq H < 240°)$ ➡ $R = I(1-S)$ $H = H - 120°$ $G = I\left[1 + \frac{S\cos H}{\cos(60^{\circ} - H)}\right]$ $B = 1-(R+G)$
- BR sector $(240° \leq H < 360°)$ ➡ $G = I(1-S)$ $H = H - 240°$ $B = I\left[1 + \frac{S\cos H}{\cos(60^{\circ} - H)}\right]$ $R = 1-(G+B)$
- In RG sector, B is minimum $S = 1 - \frac{1}{I}[\min(R,G,B)]$

$$I(1-S) = [\min(R,G,B)]$$



HSI manipulation



Pseudo color processing

- Color image processing: two categories
  - Full color processing: acquired image is a color image
  - Pseuso color processing: acquired image is gray scale
    - assign different colors to different gray scale values to generate a false color image
- Useful for human interpretation of gray scale images in which distinguishing difficulty is reduced

Pseudo color processing

- Gray scale image: pixel intensity values range [0,1,2,3,...,255] (Uint8)

- **Intensity slicing**: divide this intensity range into K intervals: u_1, u_2, ..., u_K

- Example: for K=4, u_1 = [0, 64), u_2 = [64, 128), u_2 = [128, 192), u_4 = [192, 255] (uniform intervals)

- Example: for K=4, u_1 = [0, 51), u_2 = [51, 153), u_2 = [153, 217), u_4 = [217, 255] (nonuniform)

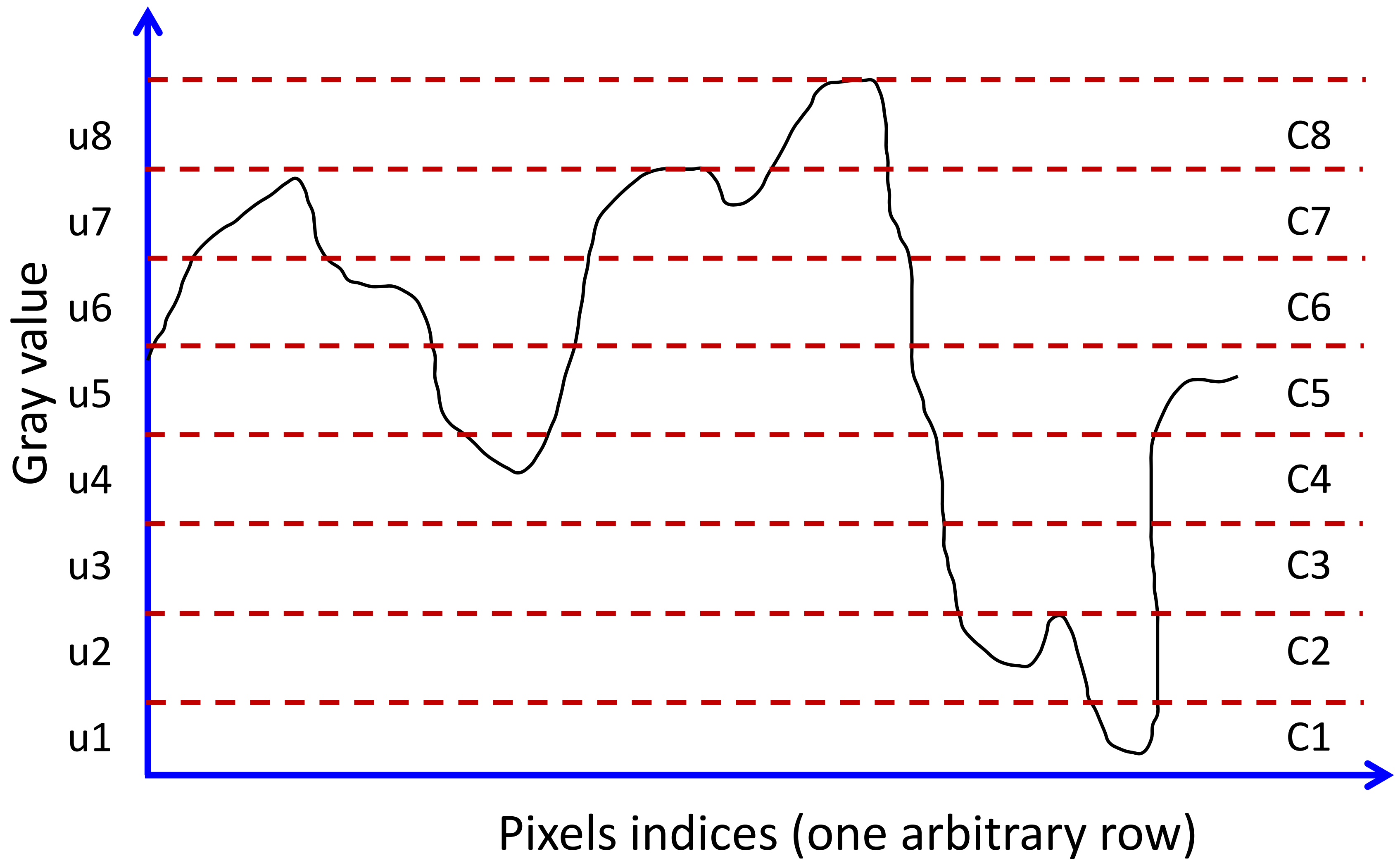
Pseudo color processing

- **Pseudo coloring**: For pixel value v in u_k, assign color C_k where C_1, C_2, C_3, C_4 are predefined colors

- Uint8: L = 256 levels; $t_0 = 0$, $t_L = 255$. Interval $u_k = [t_i, t_j)$, i<j

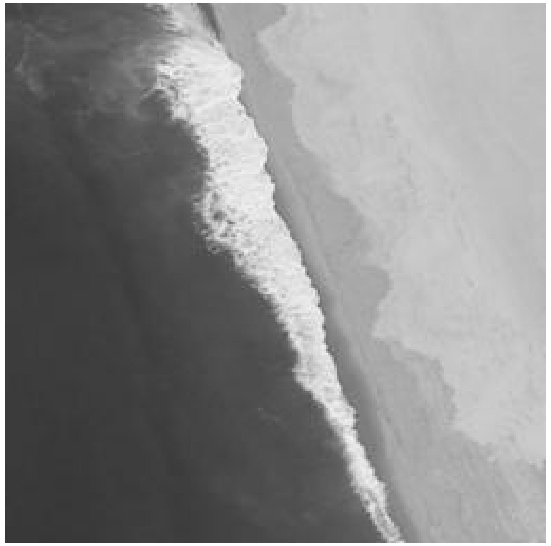
- In gray scale image I, assignment function g to obtain color image I_c
- $g(I(x,y)) := C_k$ if $I(x,y) \in u_k$
- $I_c(x,y,1) := C_k(R)$, $I_c(x,y,2) := C_k(G)$, $I_c(x,y,3) := C_k(B)$

Pseudo color processing

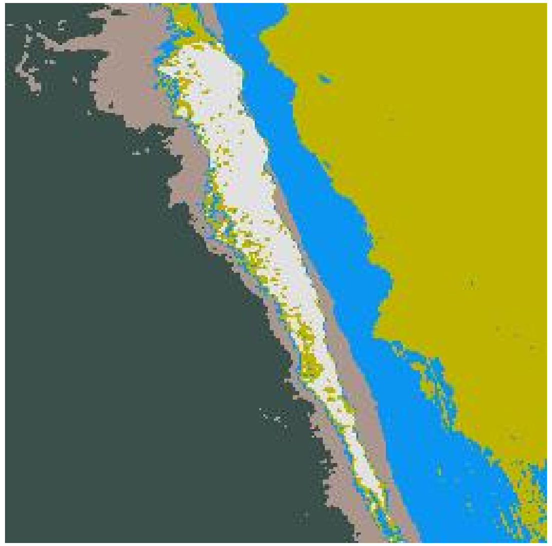


Pseudo coloring

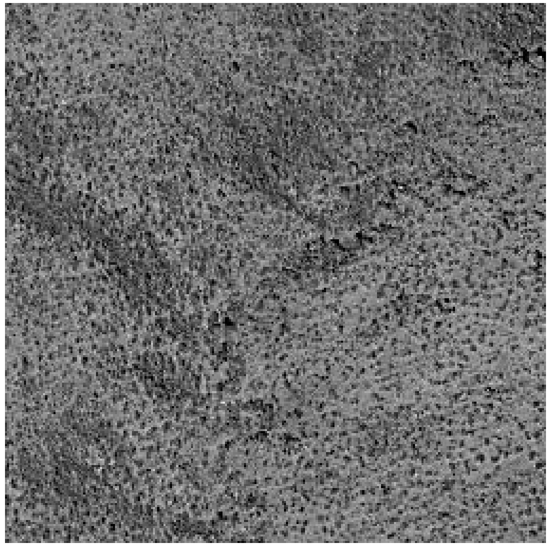
Input



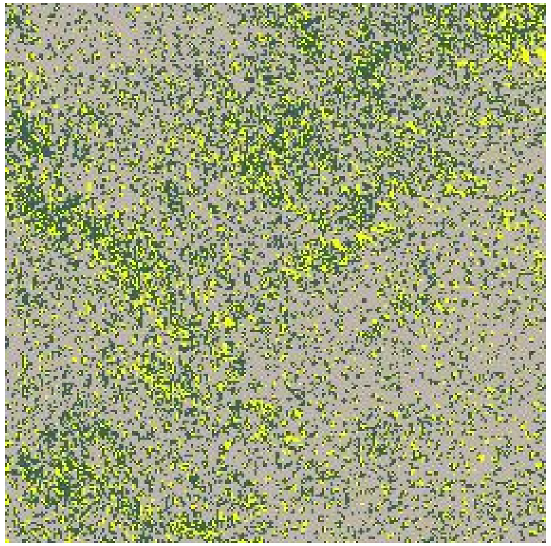
Output



Input



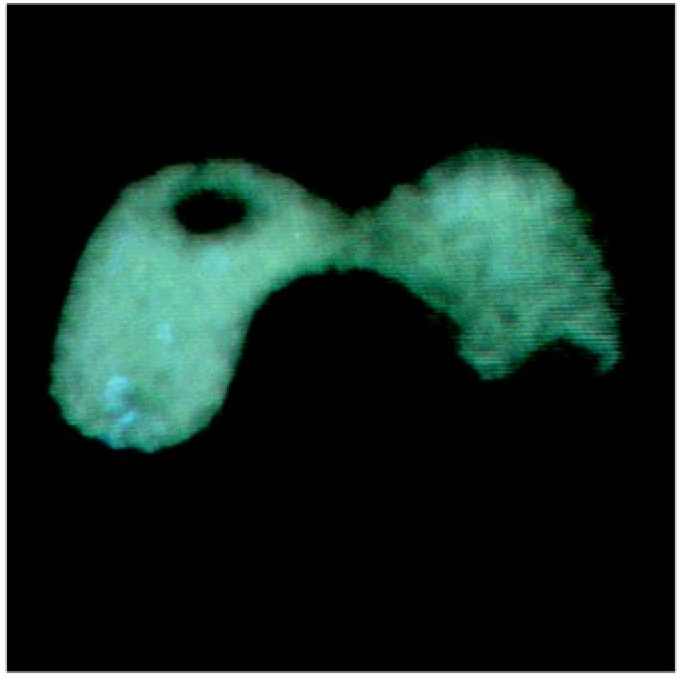
Output



Gray images from UC Merced Land Use dataset

Pseudo coloring

X-Ray image



Pseudo colored image (8 colors)

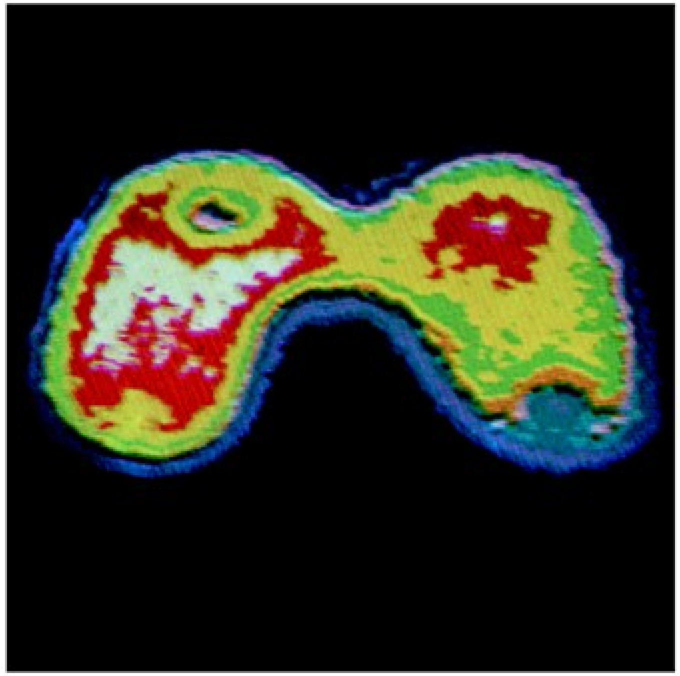
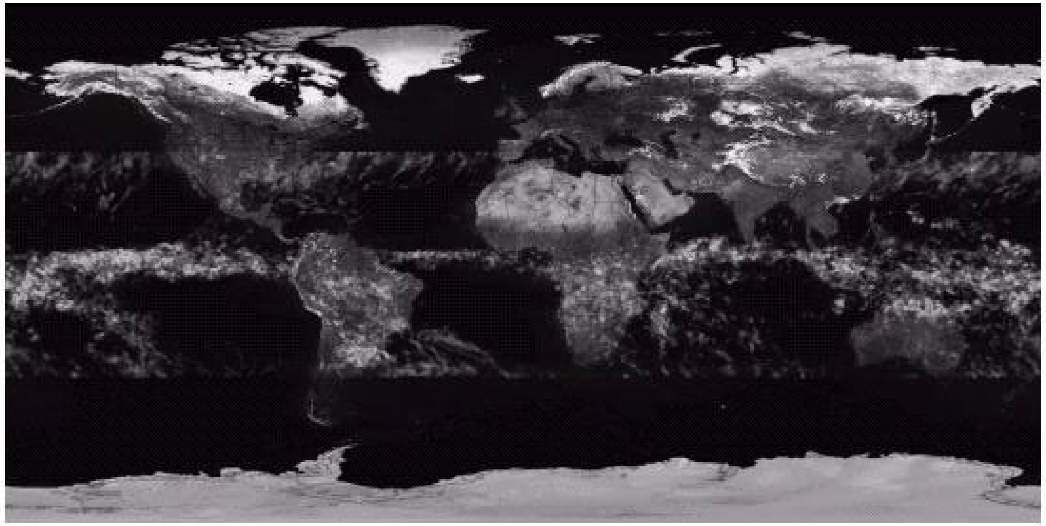
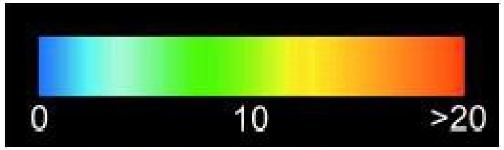


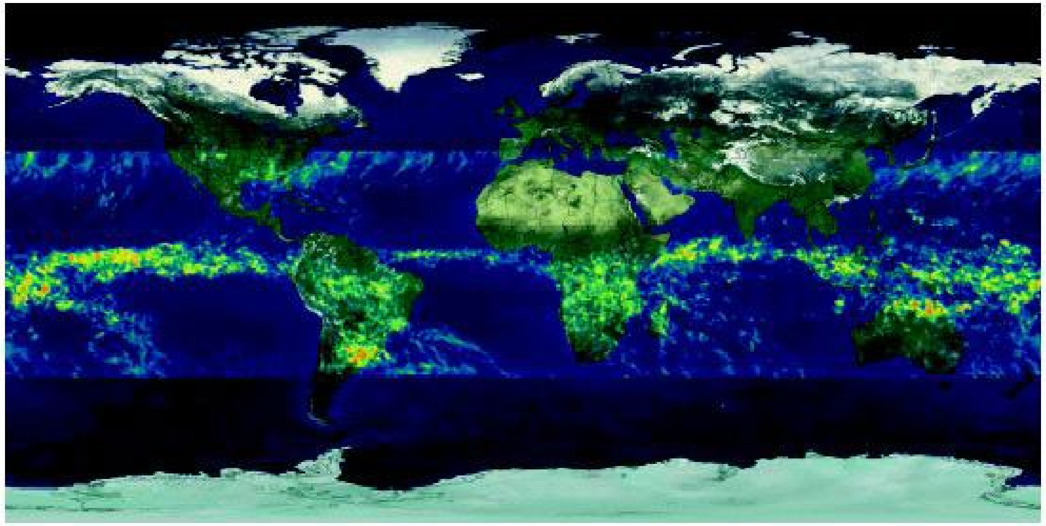
Image courtesy: Gonzalez & Woods

Pseudo coloring



Gray scale image of average rainfall





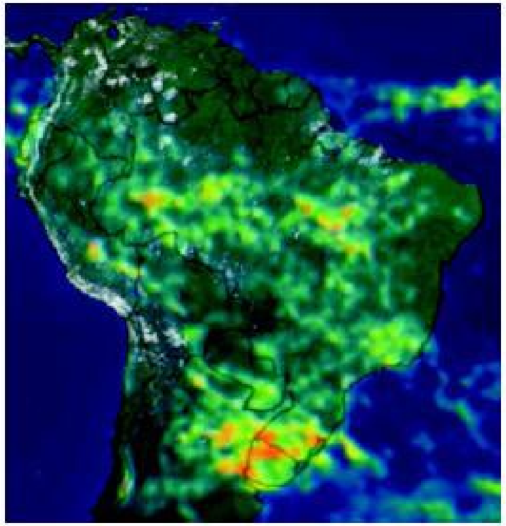
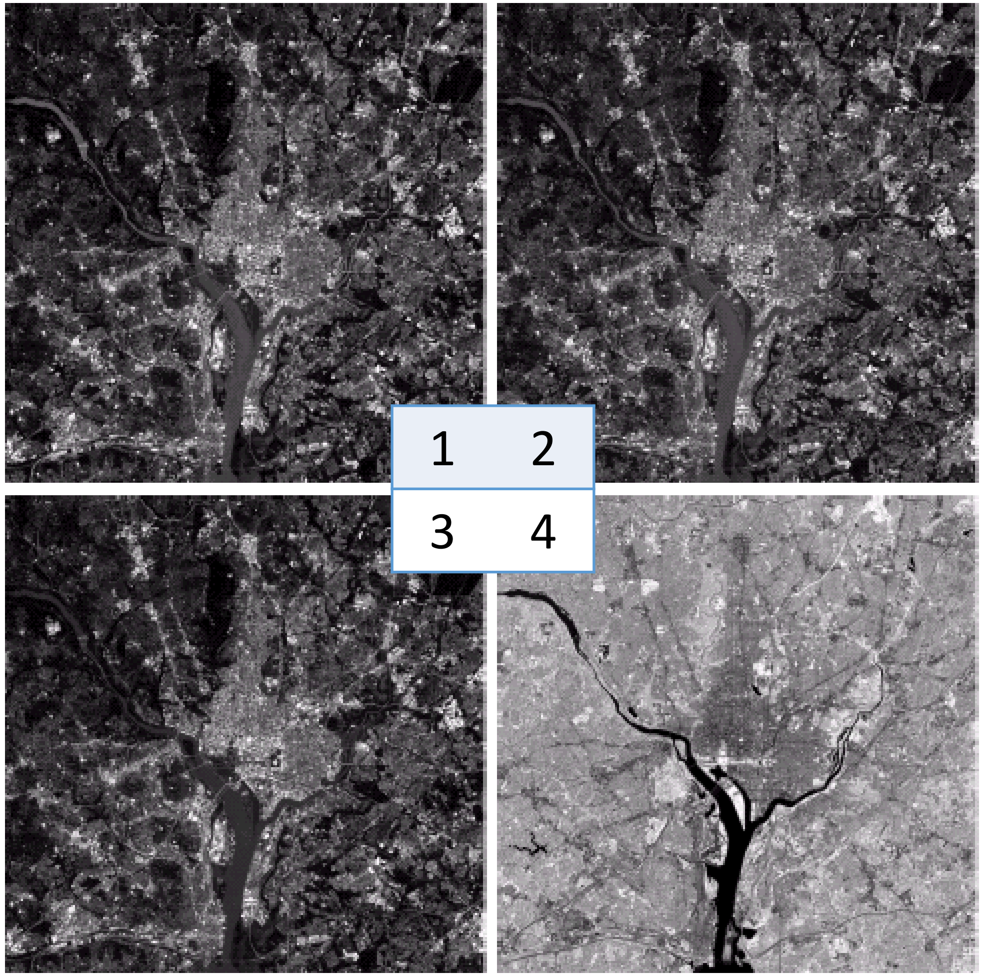


Image courtesy: Gonzalez & Woods

Pseudo coloring

Satellite image bands



Bands 1, 2, 3 as RGB Bands 4, 2, 3 as RGB

Image courtesy: Gonzalez & Woods

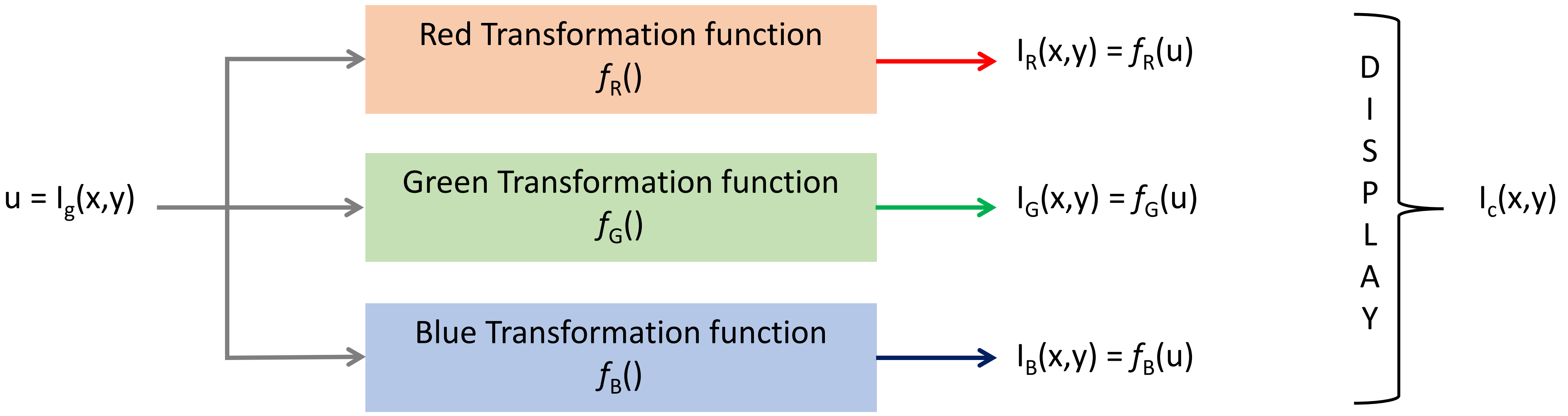
Gray to color image conversion

- Gray image I_g has one channel; Color image I_c (e.g. RGB) has three channels

- Transform I_g and generate three gray images I_R, I_G, I_B which, when combined, will be the three channels of the target RGB image

Gray to color image conversion

- Using transformation function $f()$, input u = A(x,y) is mapped to output v = B(x,y) i.e. $v = f(u)$



- $I_c(x,y,1) = I_R(x,y);$ $I_c(x,y,2) = I_G(x,y);$ $I_c(x,y,3) = I_B(x,y);$

Gray to color image transformation function

- $f_R(u) = u$, $f_G(u) = 0.63u$, $f_B(u) = 0.11u$

- $I_R(x,y) = I_g(x,y)$, $I_G(x,y) = 0.63\, I_g(x,y)$, $I_B(x,y) = 0.11\, I_g(x,y)$

Gray to color image transformation function

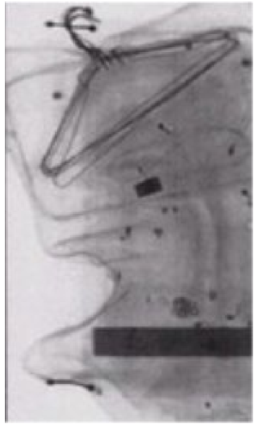
- $f_R(u) = 0.7u$, $f_G(u) = u$, $f_B(u) = 0.2u$

- $I_R(x,y) = 0.7I_g(x,y)$, $I_G(x,y) = I_g(x,y)$, $I_B(x,y) = 0.2I_g(x,y)$

Gray image from MSRA dataset

Gray to color image transformation function

- Non-linear transformation
- Sinusoidal



Input

Output

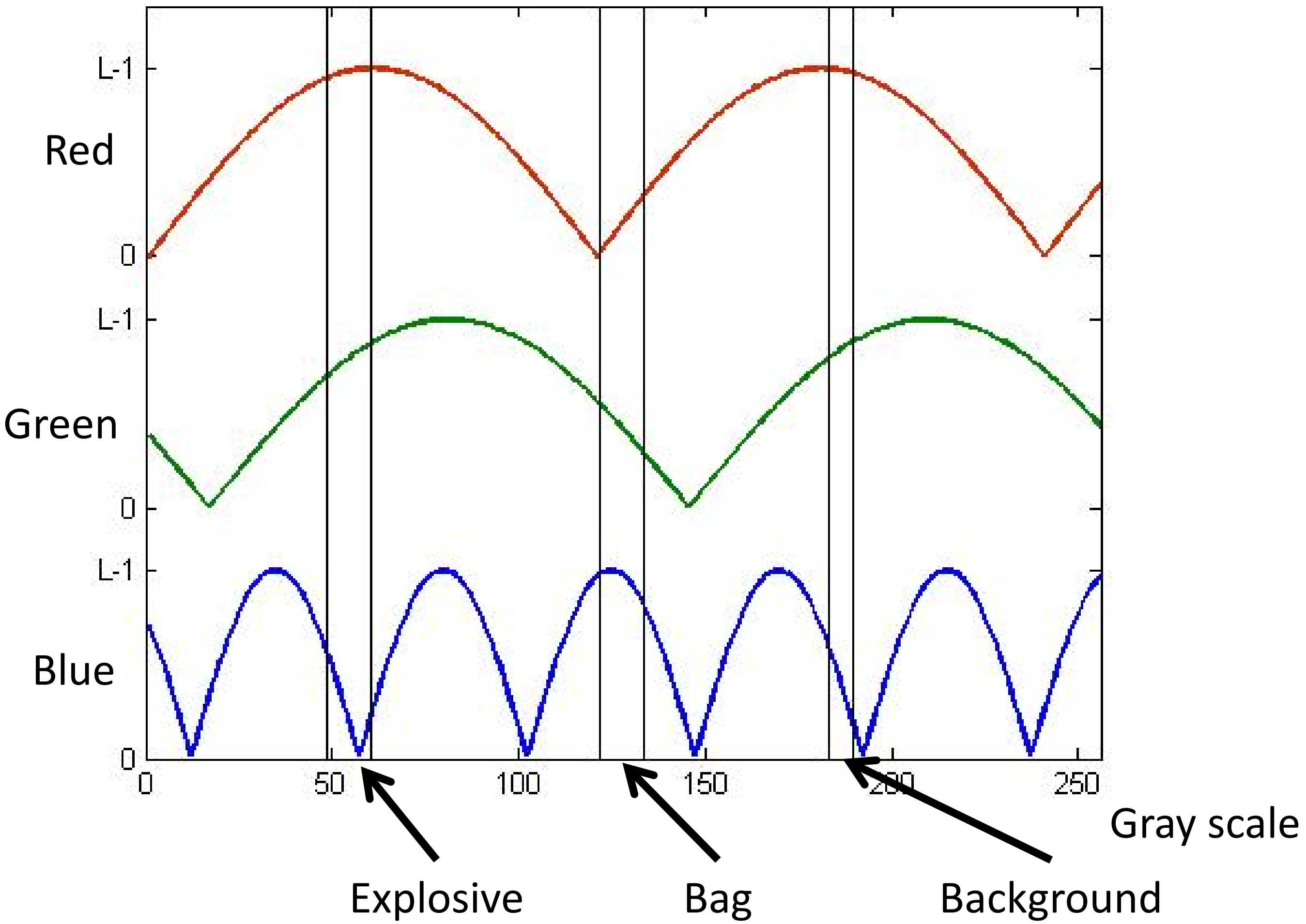
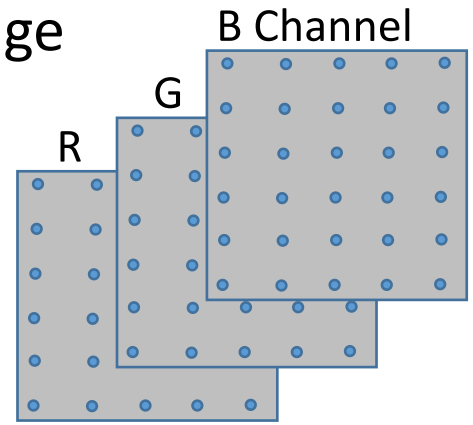


Image courtesy: Gonzalez & Woods

Full color processing

- Per-color-plane processing
  - Process individual planes $I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ independently
  - Example - mask operation for enhancement / filtering
  - Combine the processed channels to get processed color image



- Pixel level processing
  - Use vectors
  - Color at pixel-(x,y) given by $\mathbf{c}(x,y) = [I_R(x,y)\ I_G(x,y)\ I_B(x,y)]^T$

Full color processing

- Color transformation (using vectors)
  - Apply three transformation functions f_1, f_2, f_3 on every pixel $\mathbf{c}(x,y) = [u_1\ u_2\ u_3]$
  - $v_1 = f_1(u_1,u_2,u_3)$, $v_2 = f_2(u_1,u_2,u_3)$, $v_3 = f_3(u_1,u_2,u_3)$
  - Assign new color values $[v_1\ v_2\ v_3]$ to pixel-(x,y) to obtain processed image
  - In general, $v_j = f_j(u_1,u_2,u_3,\ldots,u_N)$ for $j=1,2,3,\ldots,N$ where N is the number of image channels as well as transformation functions
  - Note for hyperspectral images, number of channels is > 3

Intensity scaling

- Transformation function $f()$ scales input u by a factor k such that $v = f(u) = ku$ where $0<k<1$

- RGB: $v_j = ku_j$ for $j = 1,2,3$ where u_1,u_2,u_3 are R,G,B channel values at pixel-(x,y)

Intensity scaling

R

G

B

Scaled by 0.5

Intensity scaling

- Transformation function $f()$ scales input u by a factor k such that $v = f(u) = ku$ where $0<k<1$

- RGB: $v_j = ku_j$ for $j = 1,2,3$ where u_1,u_2,u_3 are R,G,B channel values at pixel-(x,y)

- HSI: $v_1 = u_1$, $v_2 = u_2$, $v_3 = ku_3$ where u_1,u_2,u_3 are H,S,I channel values at pixel-(x,y)

Intensity scaling

R

G

B

Scaled by 0.5

H

S

I

Intensity scaling

- Transformation function $f()$ scales input u by a factor k such that $v = f(u) = ku$ where $0<k<1$

- RGB: $v_j = ku_j$ for $j = 1,2,3$ where u_1,u_2,u_3 are R,G,B channel values at pixel-(x,y)

- HSI: $v_1 = u_1$, $v_2 = u_2$, $v_3 = ku_3$ where u_1,u_2,u_3 are H,S,I channel values at pixel-(x,y)

- CMY: $v_j = ku_j + (1-k)$ for $j = 1,2,3$ where u_1,u_2,u_3 are C,M,Y channel values at pixel-(x,y)

- Some transformations are easier in some color spaces. For intensity scaling, is it HSI?

Intensity scaling

- RGB: $v_j = ku_j$ for $j = 1,2,3$
- CMY: $v_j = ku_j + (1-k)$ for $j = 1,2,3$
- HSI: $v_1 = u_1$, $v_2 = u_2$, $v_3 = ku_3$

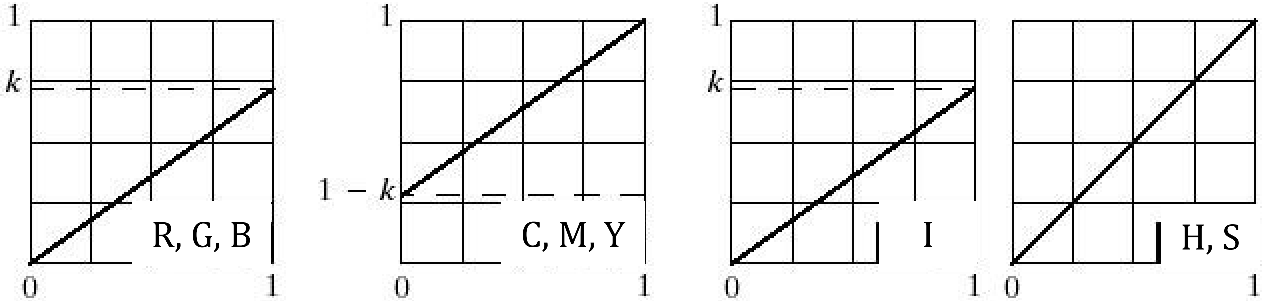


Image courtesy: Gonzalez & Woods

Color complement

- Transformation function *f*() computes additive inverse of input u = I(x,y) such that

- v = *f*(u) = 1-u where u ranges [0, 1] or
- v = *f*(u) = (L-1)-u where u ranges [0, L-1] with L = 256 for Uint8 format.

- RGB: $v_j = L-1-u_j$ for j = 1,2,3 where u_1,u_2,u_3 are R,G,B channel values at pixel-(x,y)

- The processed image is sometimes called photographic negative

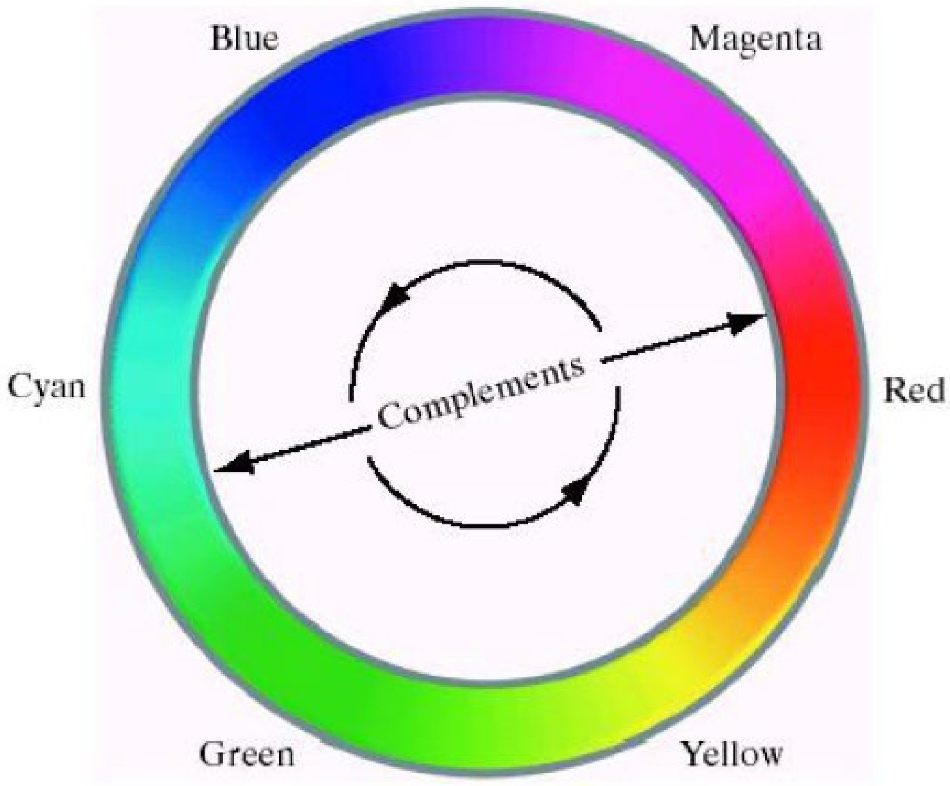


Image courtesy: Gonzalez & Woods

Color complement

R

G

B

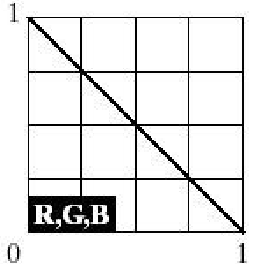
Color complement

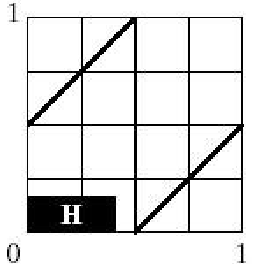
R complement

G complement

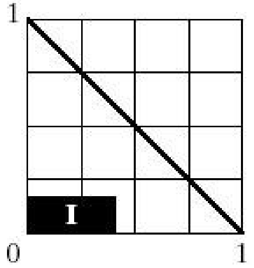
B complement

Color complement

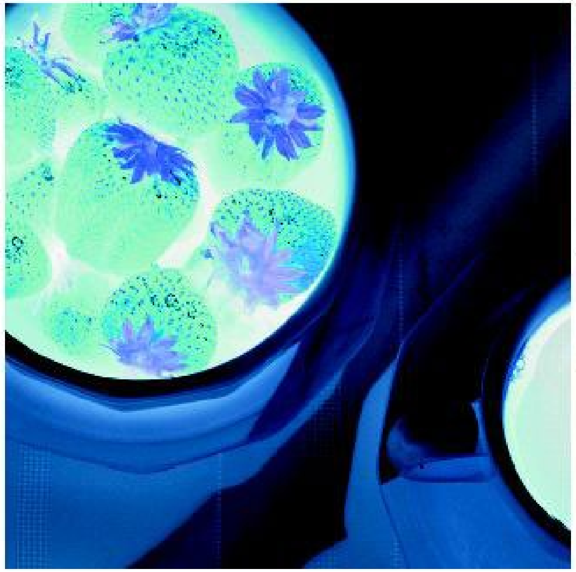


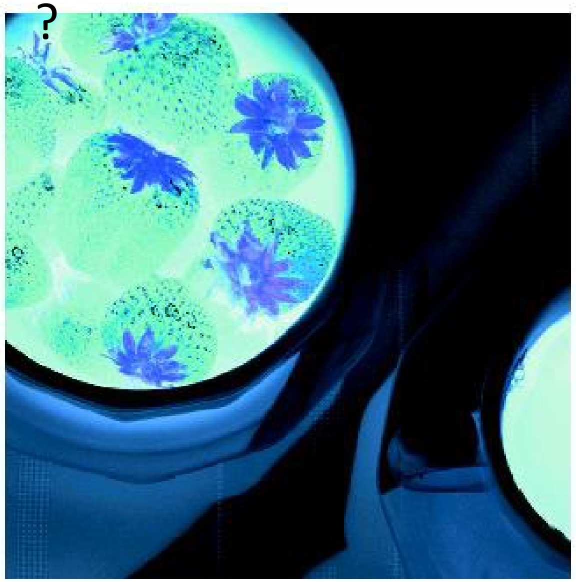


What happens to Saturation ?



Color complement obtained by complementing R,G,B channels





Color complement obtained by modifying H,I channels

Image courtesy: Gonzalez & Woods

Color slicing

- Highlight certain regions in color space

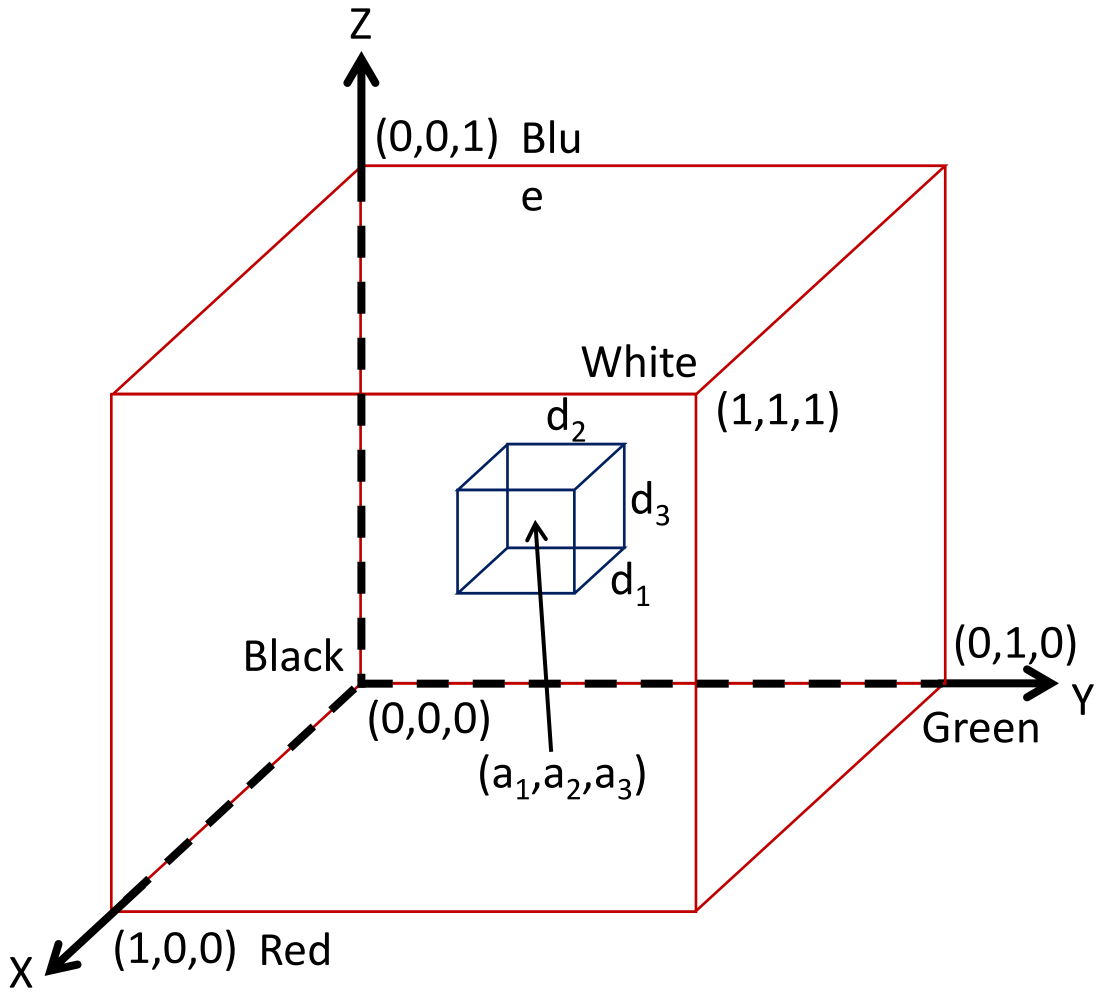
- Application: Identify objects from background

- Color at pixel-(x,y) is $\mathbf{c}(x,y) = [u_1\ u_2\ u_3]$

- $v_j = u_j$ for $j = 1,2,3$

if $|u_1 - a_1|<d_1/2$ and $|u_2 - a_2|<d_2/2$ and $|u_3 - a_3|<d_3/2$

- v_j = constant value for $j = 1,2,3$ else

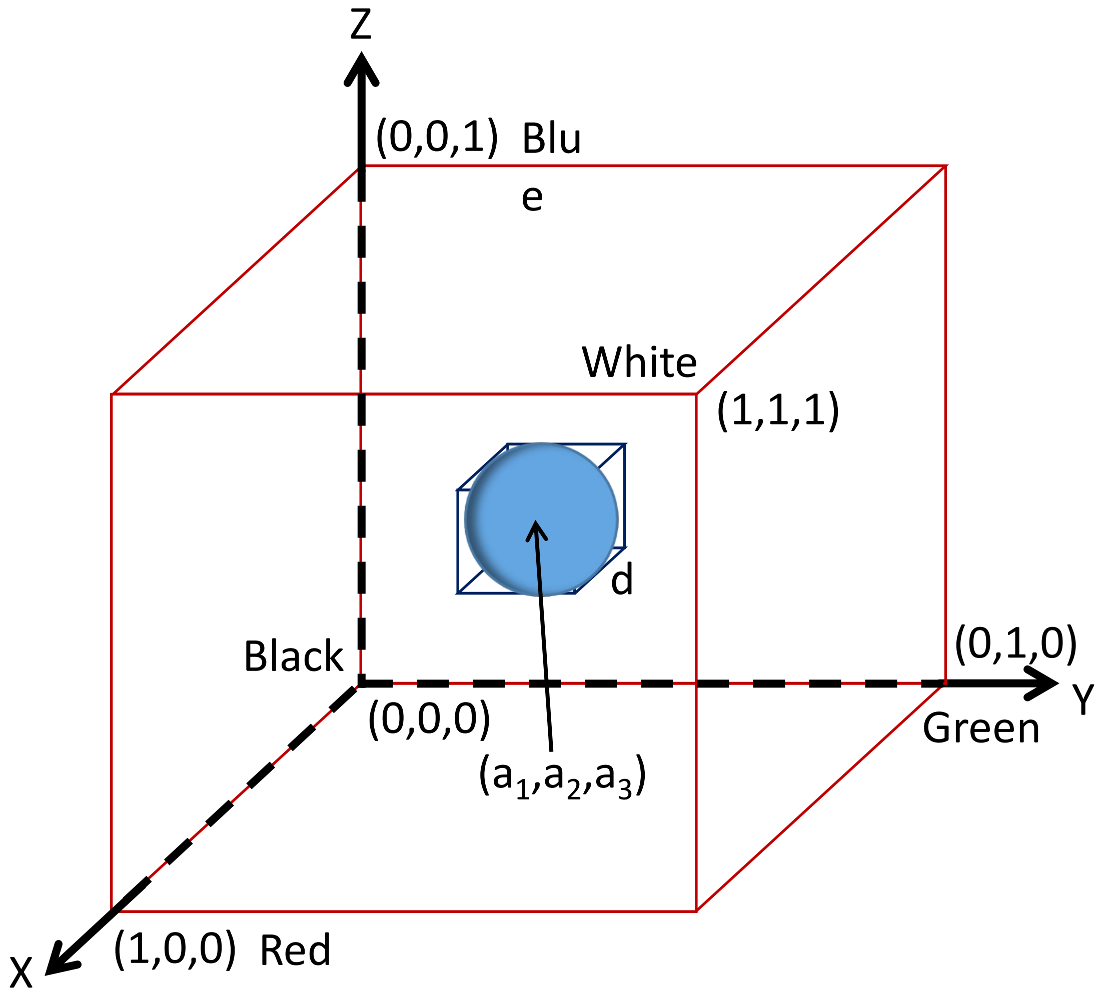


Color slicing

- Highlight certain regions in color space
- Application: Identify objects from background
- Color at pixel-(x,y) is $\mathbf{c}(x,y) = [u_1\ u_2\ u_3]$
- $v_j = u_j$ for $j = 1,2,3$

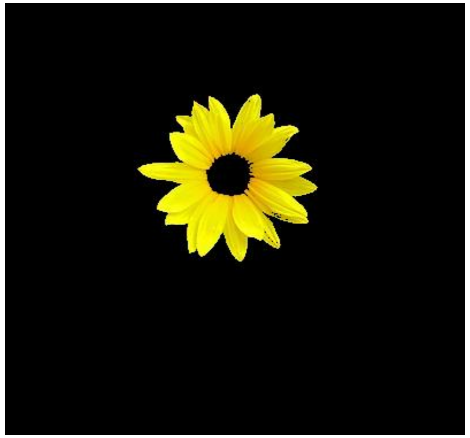
if $|u_1 - a_1|^2 + |u_2 - a_2|^2 + |u_3 - a_3|^2 < d^2$

- v_j = constant value for $j = 1,2,3$ else



Color slicing

- $v_j = 0$ for background in this example



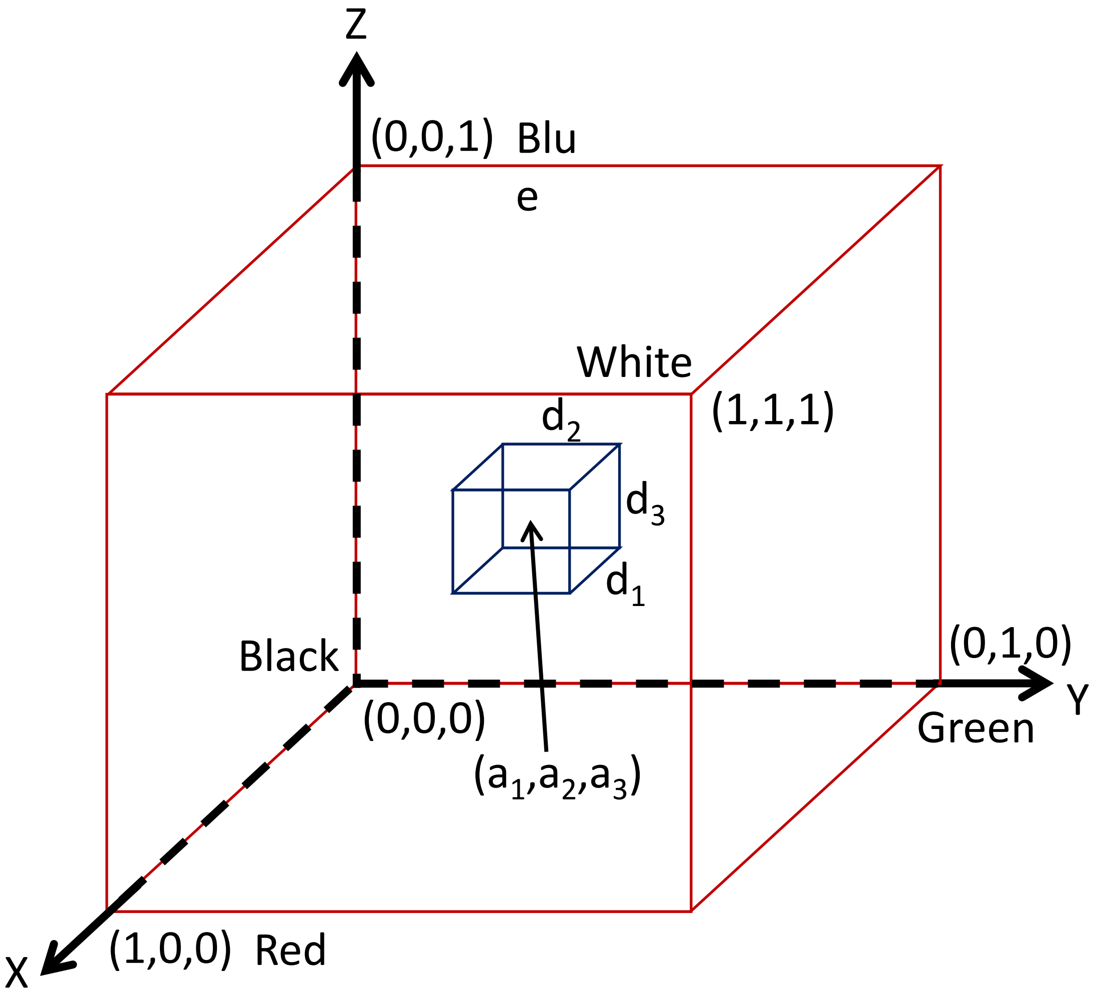
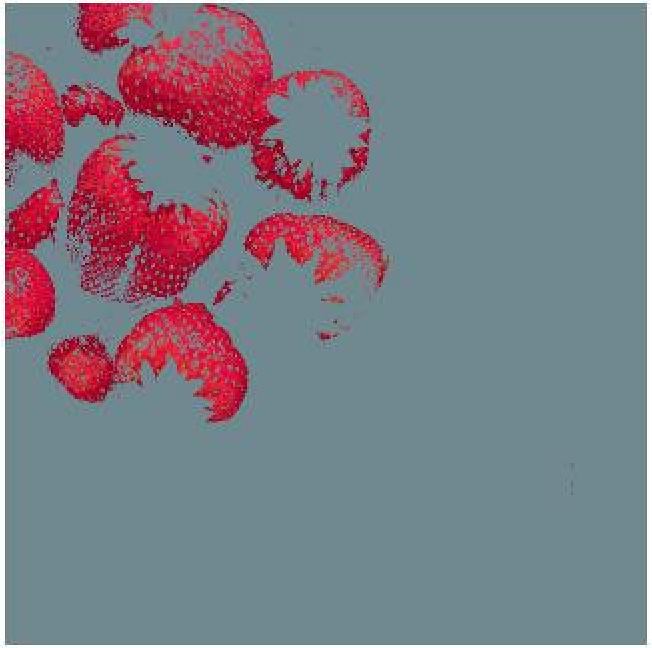


Image from: MSRA dataset

Color slicing

- $v_j = 0.5$ for background in this example

Using cubical region



Using spherical region

Image courtesy: Gonzalez & Woods