

Computer Vision, and Digital Image Processing

Prepared by Dr. Mohamed Mostafa

AI 612

Lecture 9





مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY

Color Image Processing

Lecture 9

AI 612

Prepared by
Dr. Mohamed Mostafa

www.midocean.ae

Chapter
06

Image Morphology

Resources for this
lecture

Chapter 6:

Gonzalez, Rafael C. *Digital image processing*.
Pearson education, India, 2009.

Color Image Processing

- **Motivations:**

- Color is **a powerful descriptor** that often simplifies object identification and extraction from a scene.
- A humans can **discern thousands of color shades**, compared to only about two dozen shades of gray.

Color Image Processing

- **Areas:**
- **Pseudo-color processing:**
 - The process of assigning color(s) to a particular grayscale intensity or range of intensities.
- **Full-color processing:**
 - images typically are acquired using a full-color sensor, such as a digital camera, or color scanner.

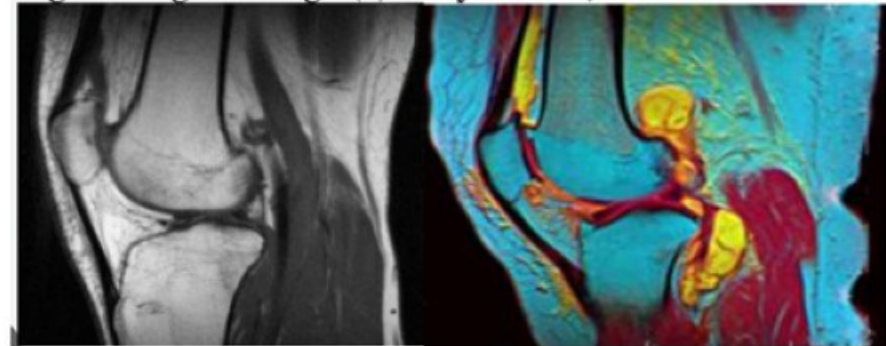


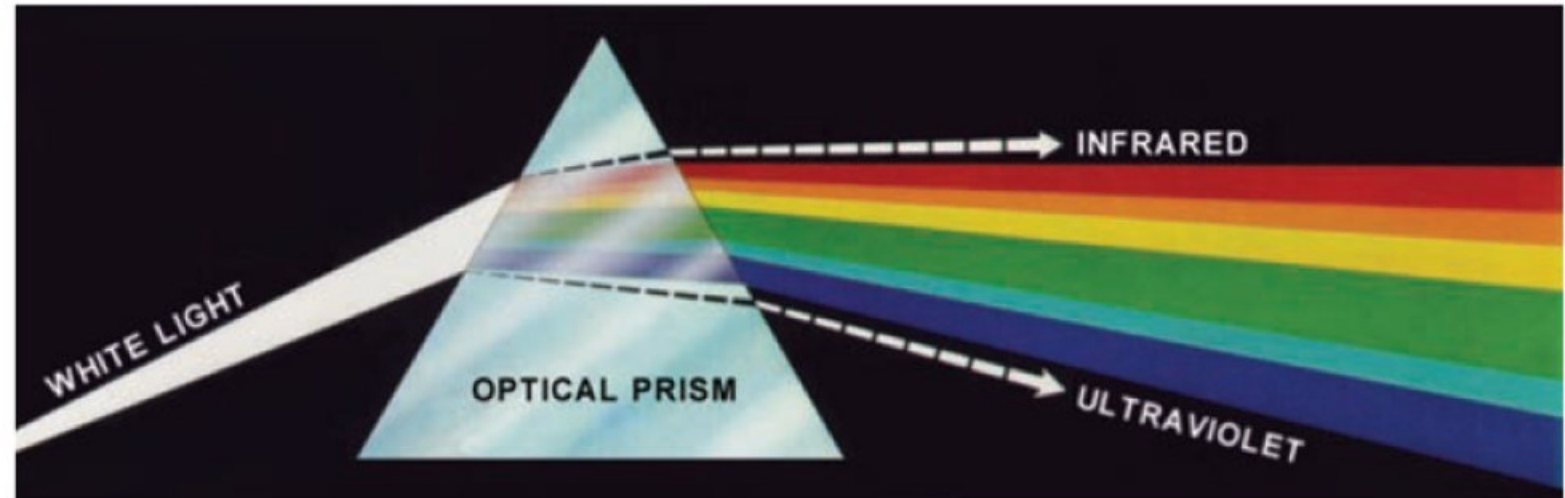
Image source: [Reseachgate](#)



Color Fundamentals

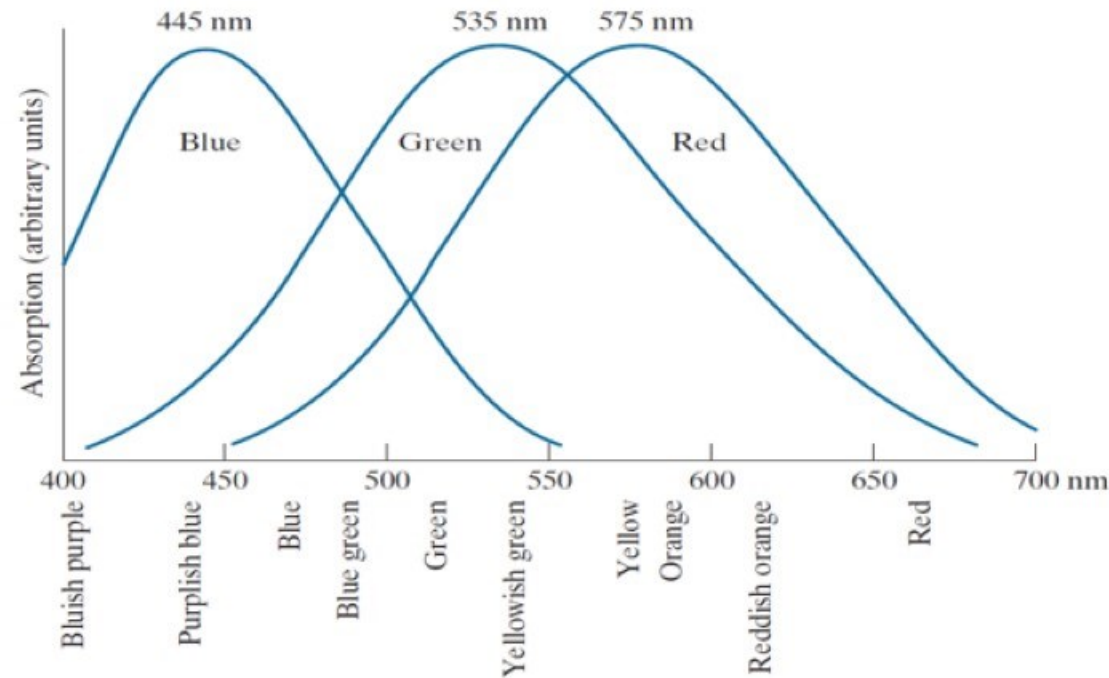
In **1666**, **Sir Isaac Newton**, 24 year old, discovered white light spectrum.

- The emerging light is not white, but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other (violet, blue, green, yellow, orange, and red.).



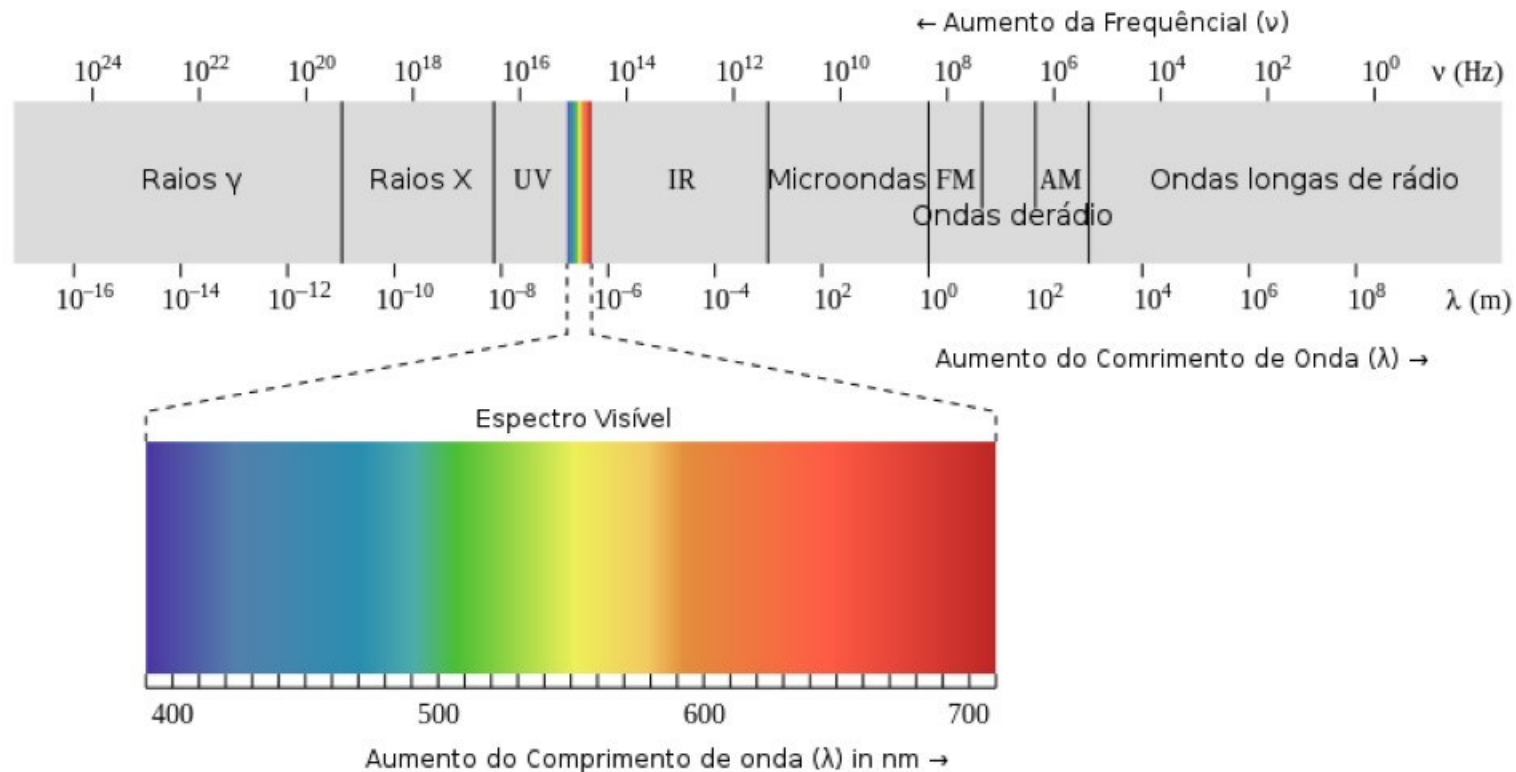
Color Fundamentals

- Basically, the colors that humans and some other animals perceive in an object are determined by the nature of the light reflected from the object.

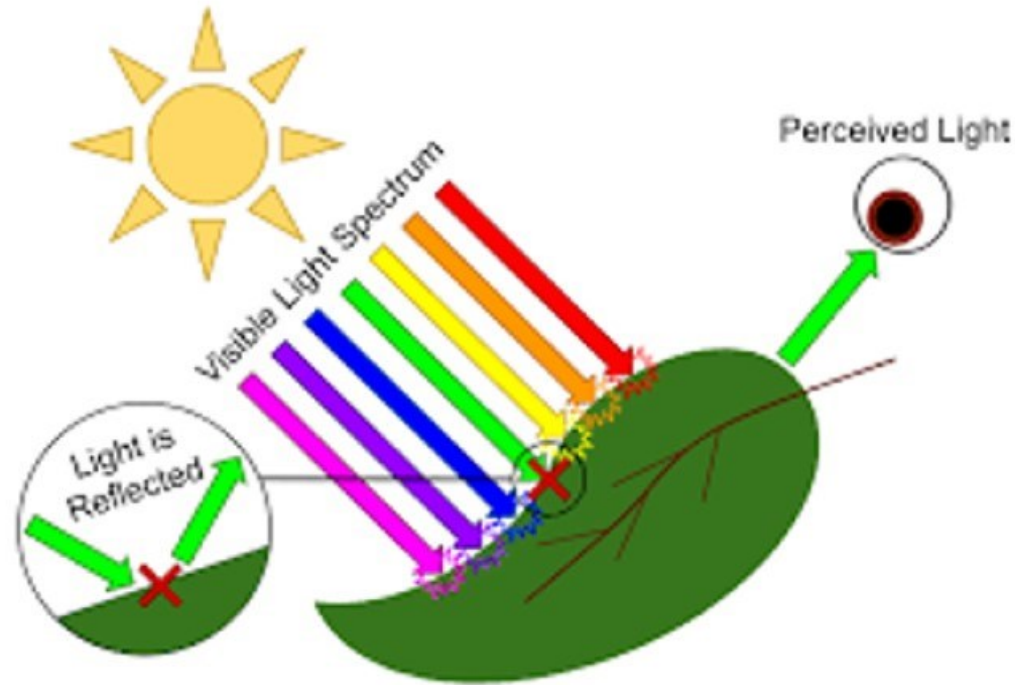


Color Fundamentals

- Basically, **visible light wavelength: from around 400 to 700 nm**



Color Fundamentals



For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range, while absorbing most of the energy at other wavelengths.

White Color Characteristics

- 1. For an achromatic (monochrome, gray scale) light source,
 - there is only 1 attribute to describe the quality: **intensity (Brightness)**
- 2. For a chromatic light source (**400 to 700 nm**), there are 3 attributes to describe the quality:
 - **Radiance** = total amount of energy flow from a light source (Watts)
 - **Luminance** = amount of energy received by an observer (lumens)
 - **Brightness** = intensity

Color Models

- The purpose of a *color model* (also called a *color space* or *color system*) is to facilitate the specification of colors in some standard way.
- Colors' models:
 - **Hardware oriented models:**
 - RGB (red, green, blue) → for monitors and cameras
 - CMY (cyan, magenta, yellow) → for printing
 - **Image processing oriented models:**
 - HSI (hue, saturation, intensity) → it decouples the color and gray-scale information in an image

RGB Model

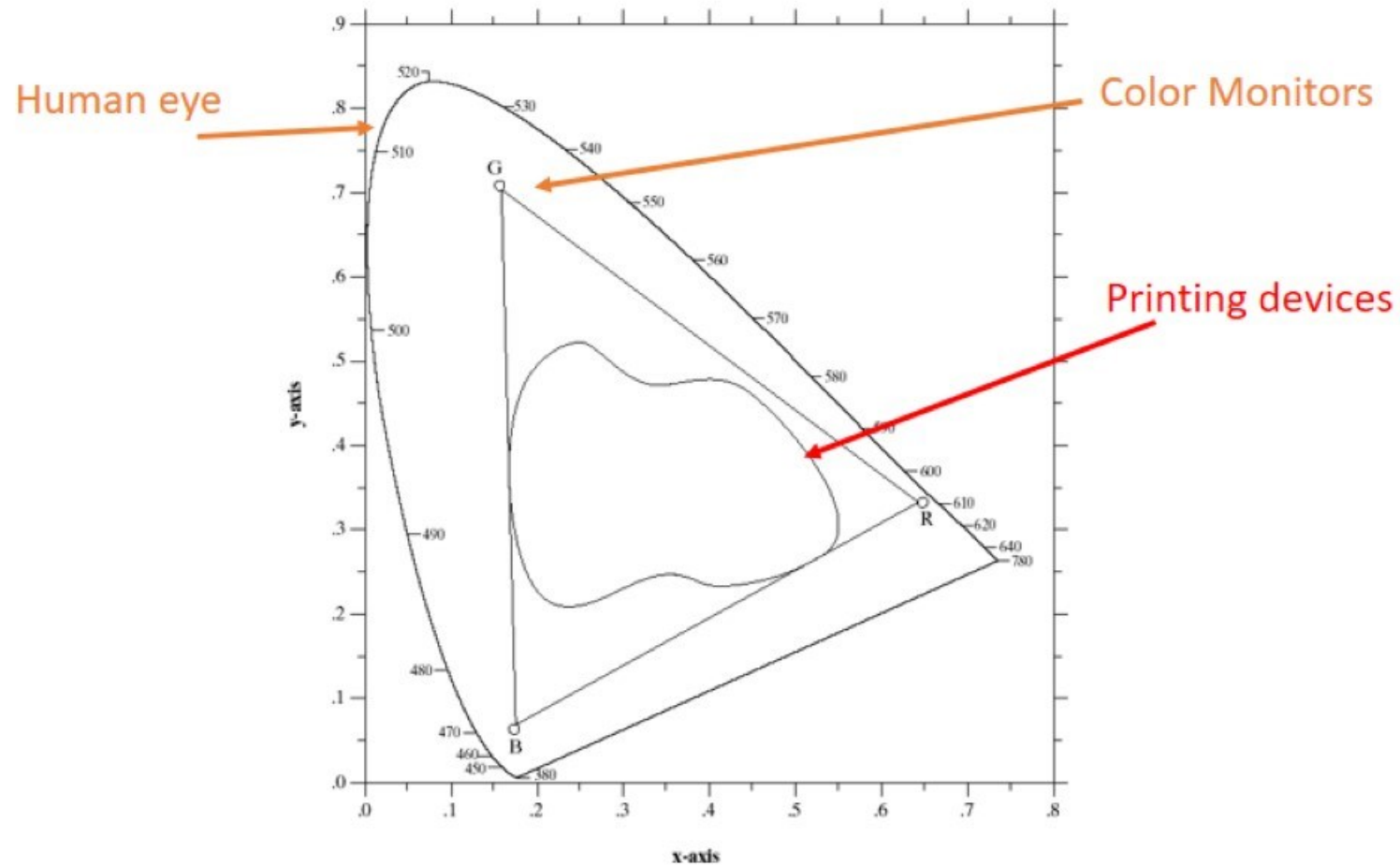
- In the RGB model, each color appears in its primary spectral components of red, green, and blue.
- The number of bits used to represent each pixel in RGB space is called the *pixel depth*.
- An RGB image in which each of the red, green, and blue images is an 8-bit image. Under these conditions, each RGB *color pixel* [that is, a triplet of values (R, G, B)] has a depth of 24 bits (3 image planes times the number of bits per plane).
- The term *full-color* image is used often to denote a 24-bit RGB color image.
- The total number of possible colors in a 24-bit RGB image is $(2^8)^3 = 16,777,216$.

Note :

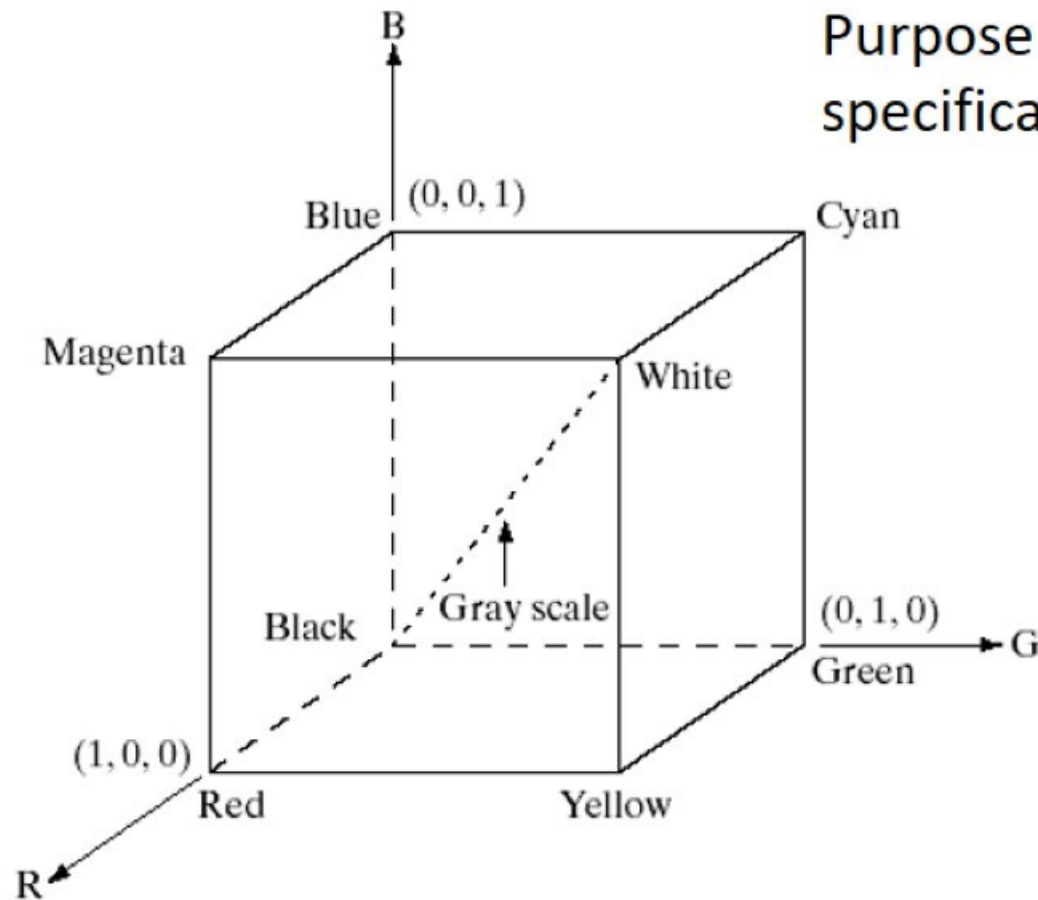
Pixel depth (color depth): refers to the number of bits used to represent the color of each pixel in an image.

- In a grayscale image, a pixel depth of 8 bits represents 256 different shades of gray ($2^8 = 256$).

Color Gamut of Color Monitors and Printing Devices



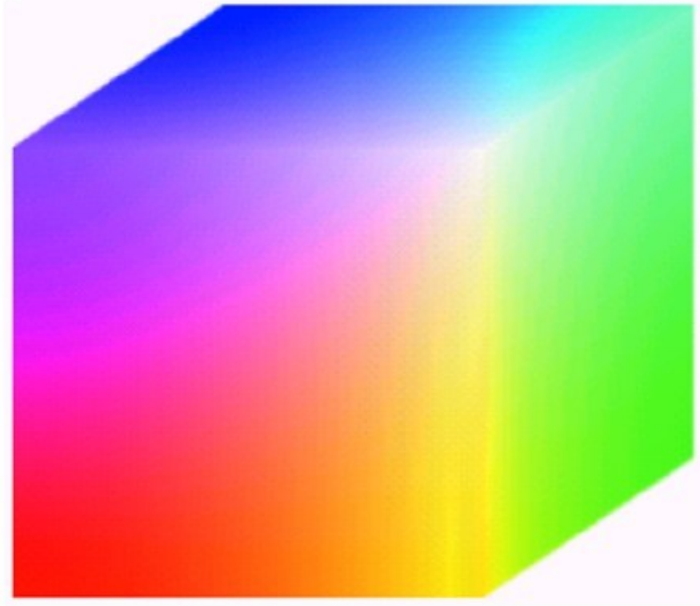
RGB Color Model



Purpose of color models: to facilitate the specification of colors in some standard

RGB color models:
- based on Cartesian coordinate system

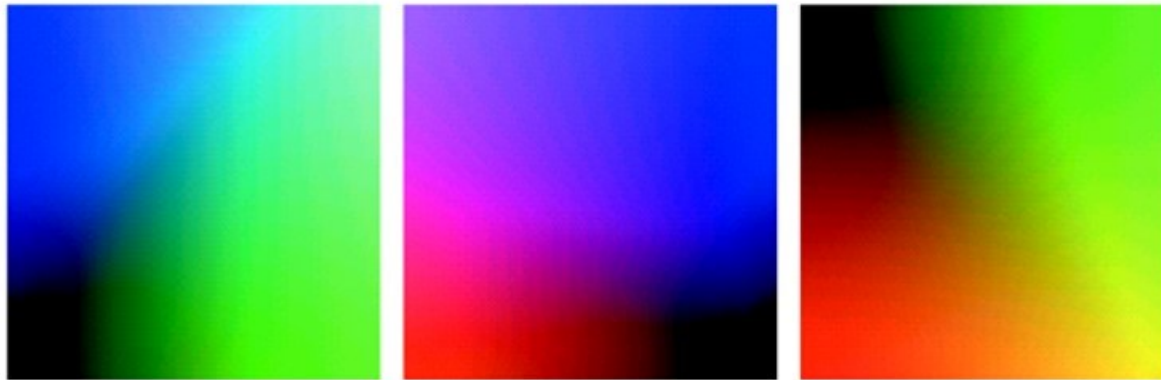
RGB Color Cube



R = 8 bits
G = 8 bits
B = 8 bits



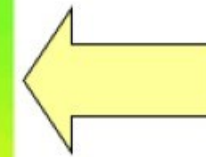
Color depth 24 bits
= 16777216 colors



(R = 0)

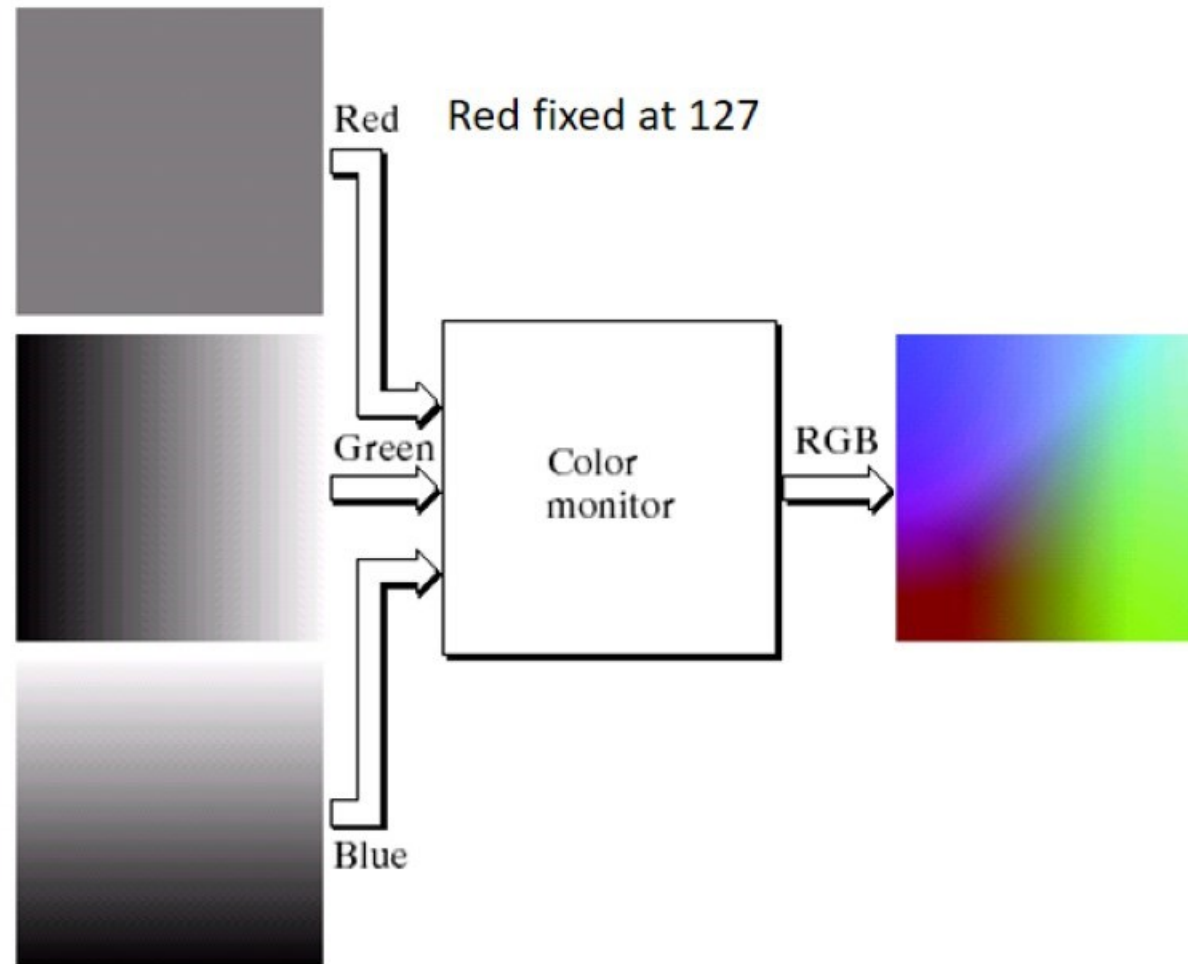
(G = 0)

(B = 0)



Hidden faces
of the cube

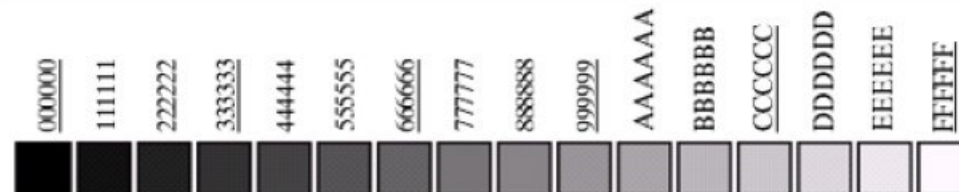
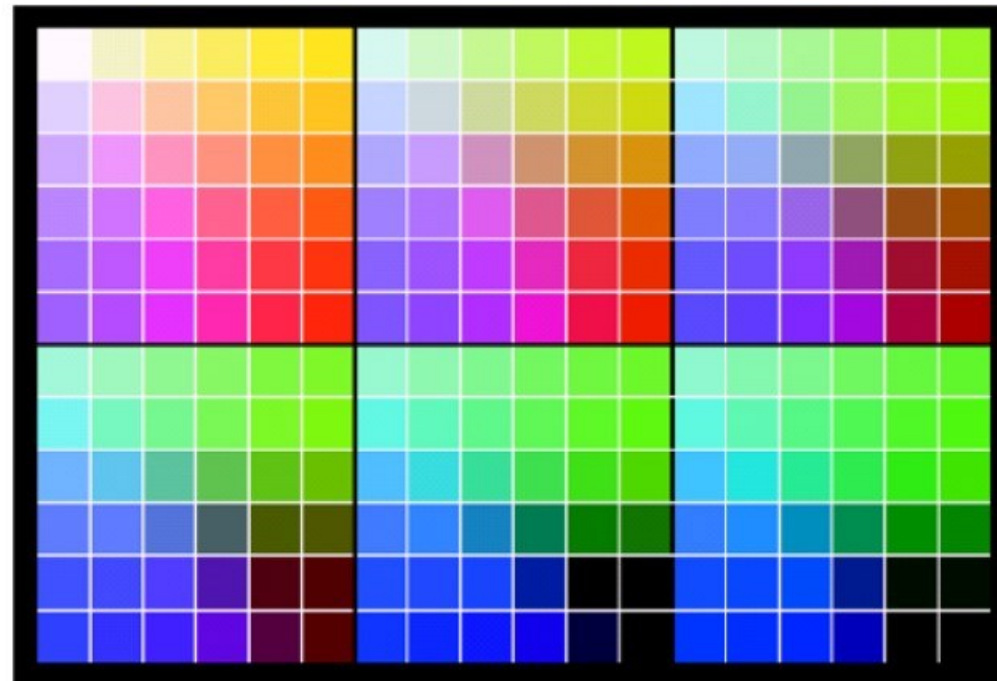
RGB Color Model (cont.)



Safe RGB Colors

Safe RGB colors: a subset of RGB colors.

There are 216 colors common in most operating systems.



a
b

FIGURE 6.10

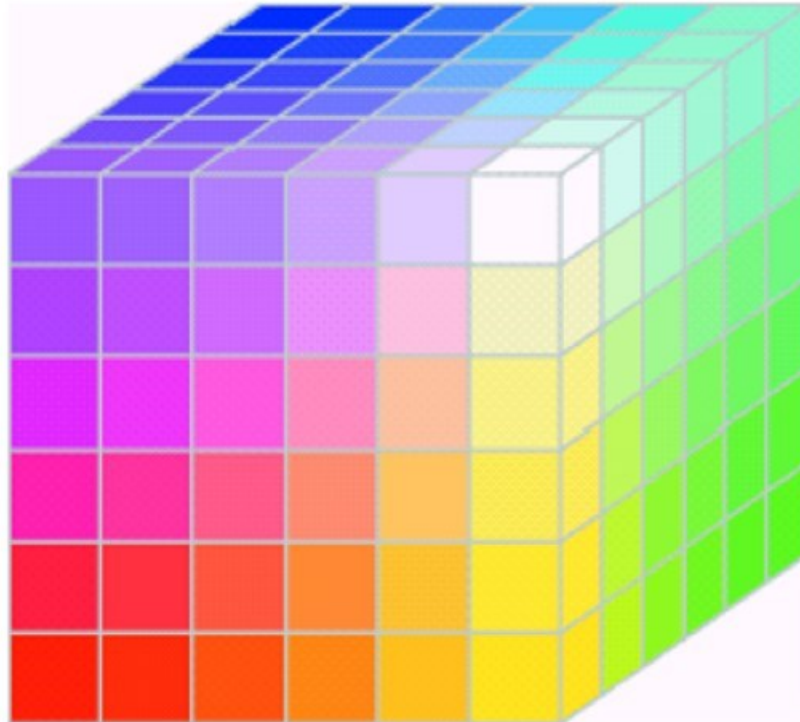
(a) The 216 safe RGB colors.
(b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

RGB Safe-color Cube

Number System		Color Equivalents					
Hex	00	33	66	99	CC	FF	
Decimal	0	51	102	153	204	255	

TABLE 6.1

Valid values of each RGB component in a safe color.

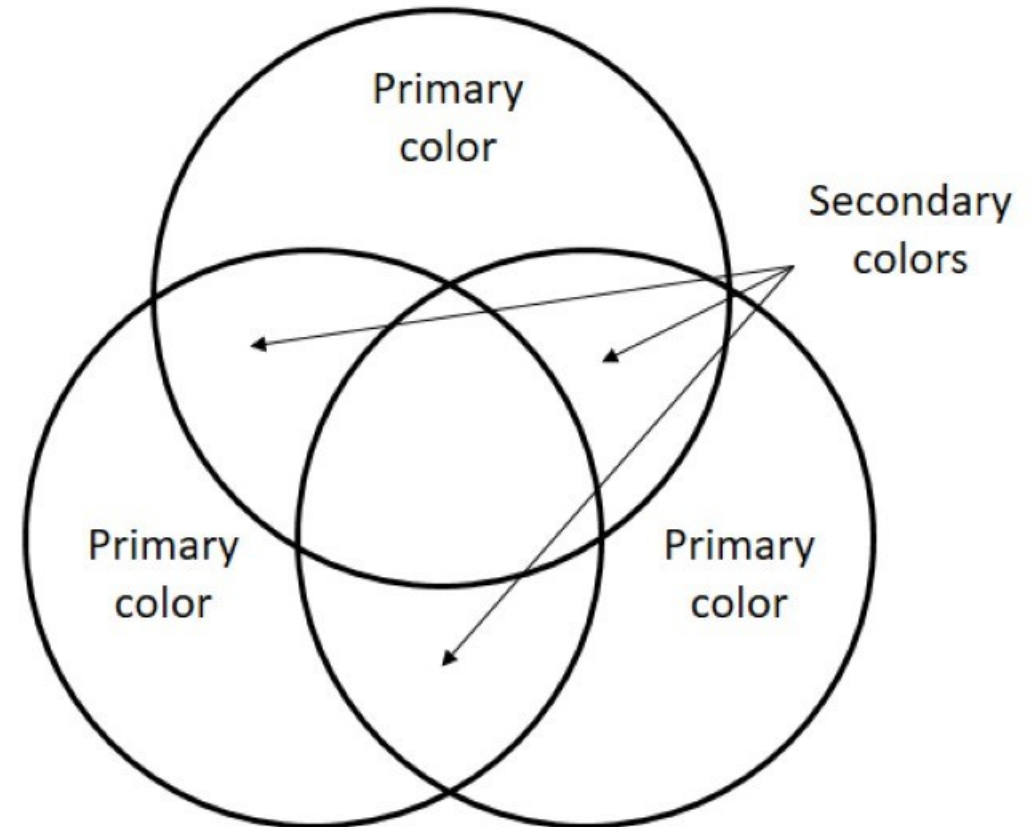


The RGB Cube is divided into 6 intervals on each axis to achieve the total $6^3 = 216$ common colors.

However, for 8 bit color representation, there are the total 256 colors. Therefore, the remaining 40 colors are left to OS.

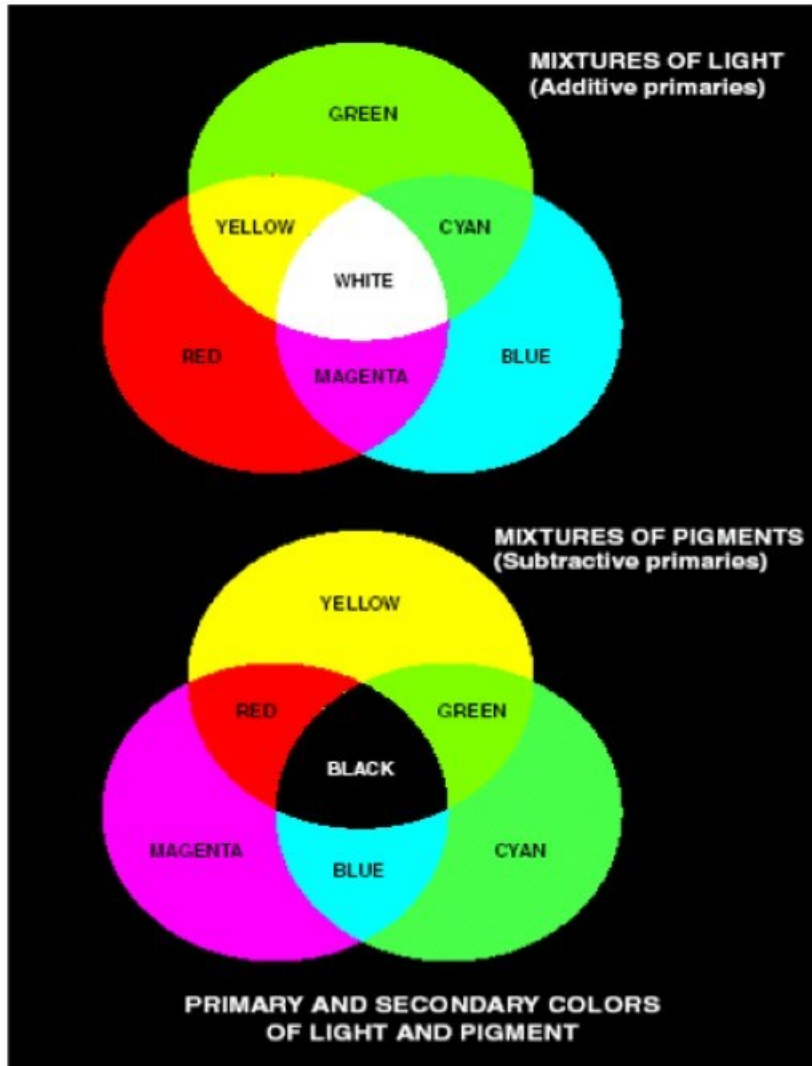
Primary and Secondary Colors

- The purpose of a *color model* (also called a *color space* or *color system*) is to facilitate the specification of colors in some standard way.



Primary and Secondary Colors

Primary and Secondary Colors



Additive primary colors: RGB
use in the case of light sources
such as color monitors

RGB add together to get white

Subtractive primary colors: CMY
use in the case of pigments in
printing devices

White subtracted by CMY to get
Black

CMY and CMYK Color Models

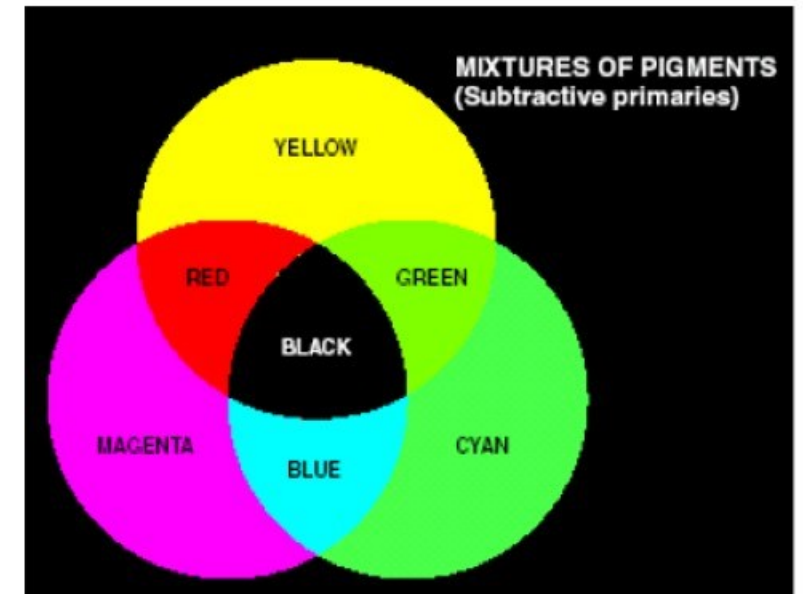
C = Cyan (G+B)

M = Magenta (R+B)

Y = Yellow (R+G)

K = Black (R+G+B) → Muddy brown instead of black

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = 255 - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



RGB, CMY models are not good for human interpreting

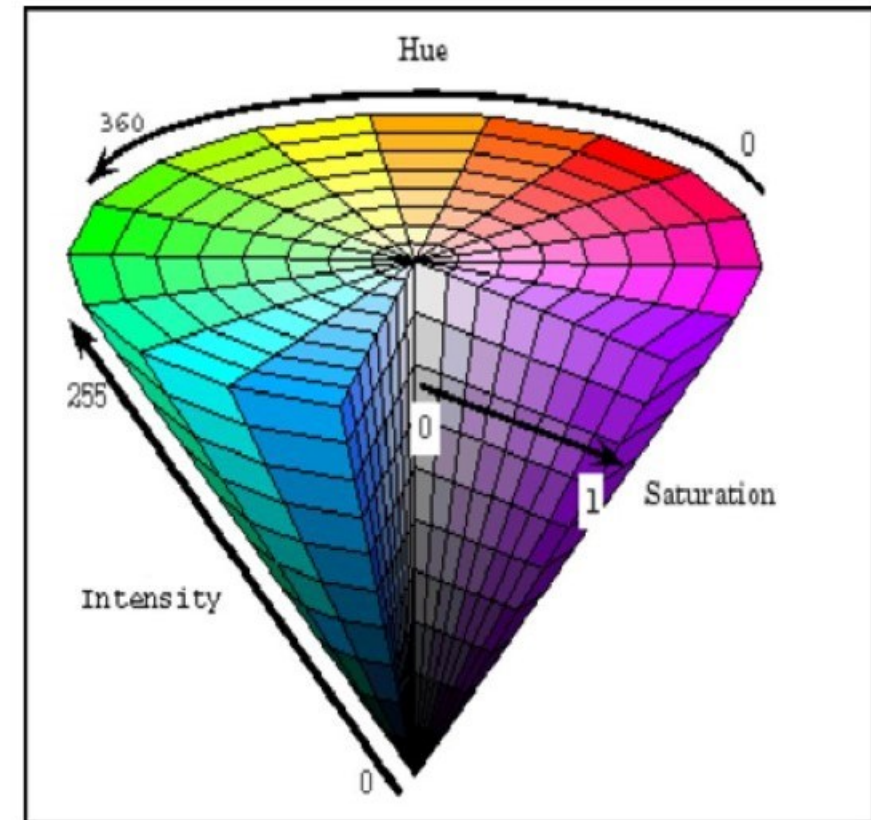
HSI Color model:

Hue: Dominant color

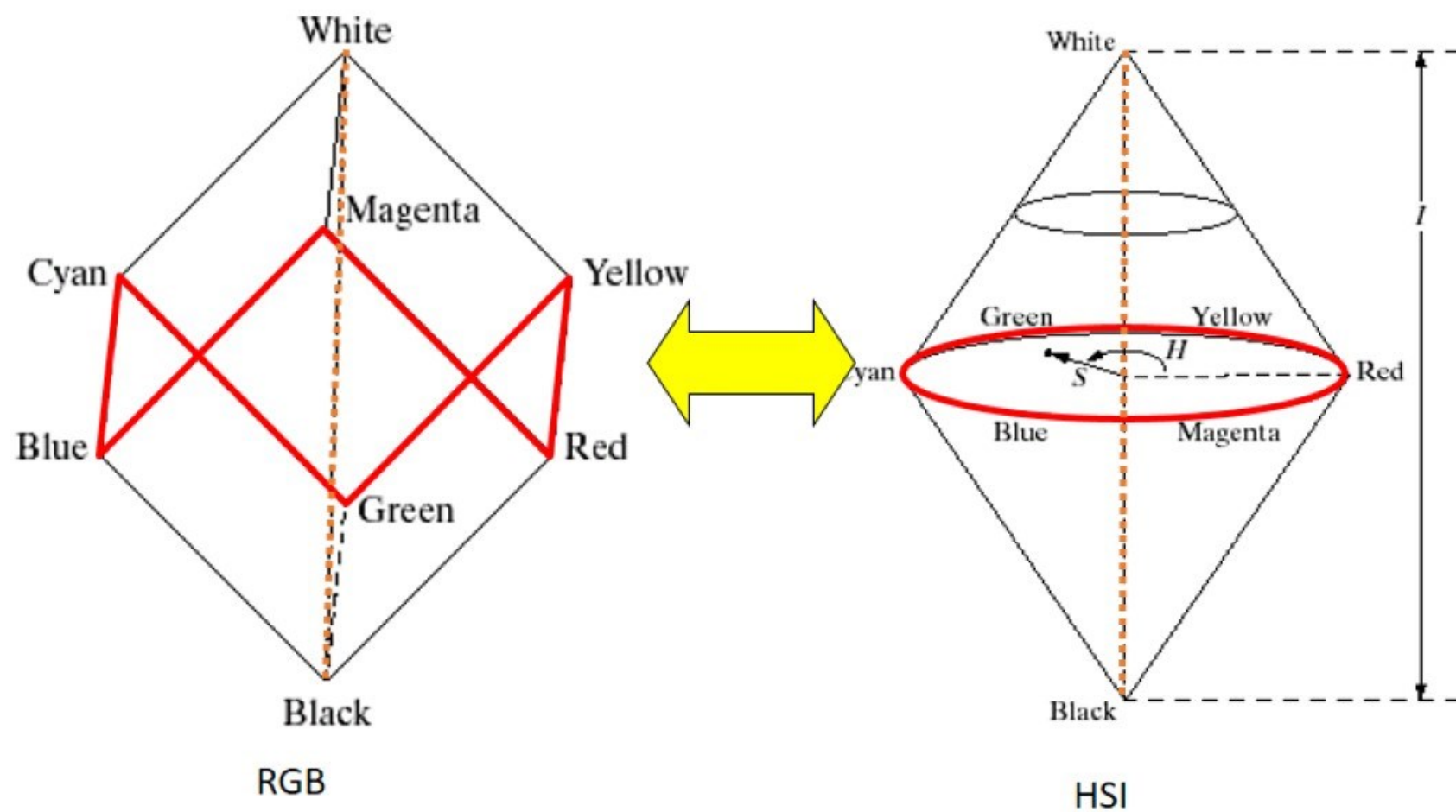
Saturation: Relative purity (inversely proportional to amount of white light added)

Intensity: Brightness

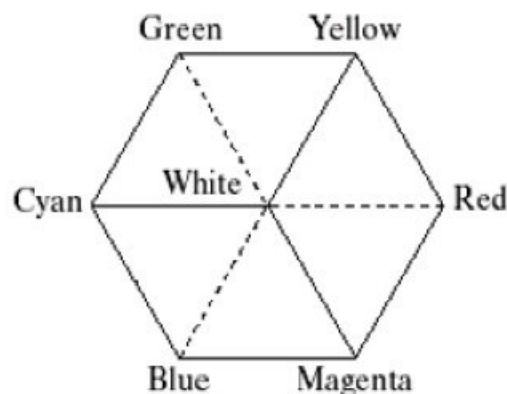
Color carrying information



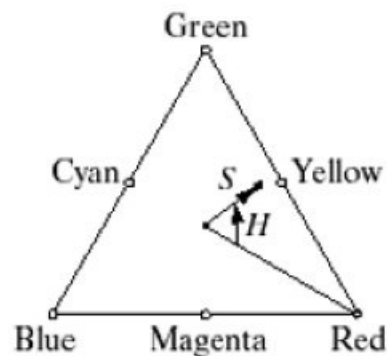
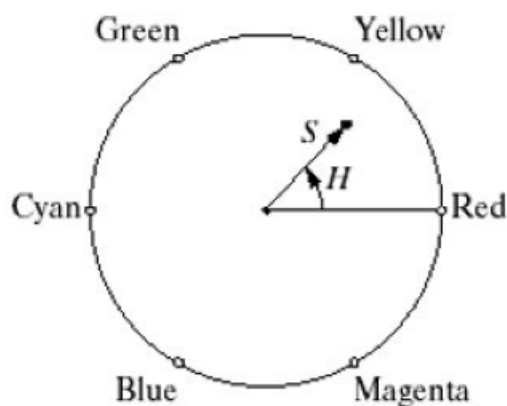
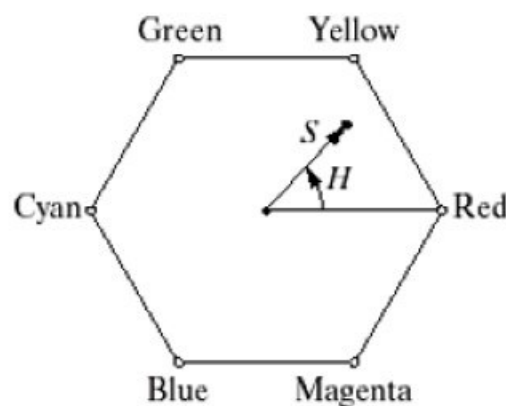
Relationship Between RGB and HSI Color Models

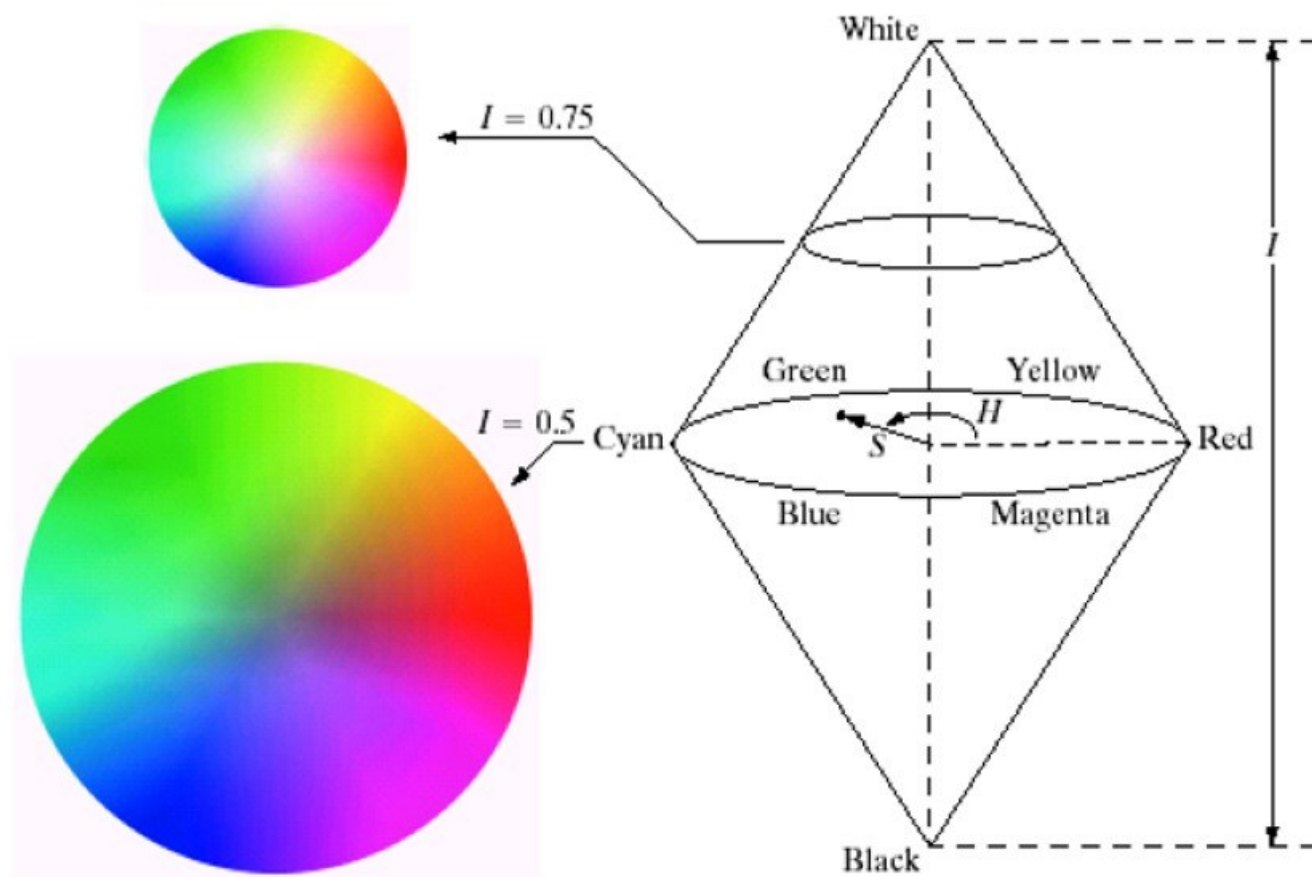


Hue and Saturation on Color Planes



1. A dot in the plane is an arbitrary color
2. Hue is an angle from a red axis.
3. Saturation is a distance to the point.





Intensity is given by a position on the vertical axis.

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

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

$$S = 1 - \frac{3}{R + G + B}$$

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

Converting Colors from HSI to RGB

RG sector: $0 \leq H < 120$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = I(1 - S)$$

$$G = 1 - (R + B)$$

BR sector: $240 \leq H \leq 360$

$$H = H - 240$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = I(1 - S)$$

$$R = 1 - (G + B)$$

GB sector: $120 \leq H < 240$

$$H = H - 120$$

$$R = I(1 - S)$$

$$G = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = 1 - (R + G)$$

There are 2 types of color image processes

1. Pseudocolor image process: Assigning colors to gray values based on a specific criterion. Gray scale images to be processed may be a single image or multiple images such as multispectral images

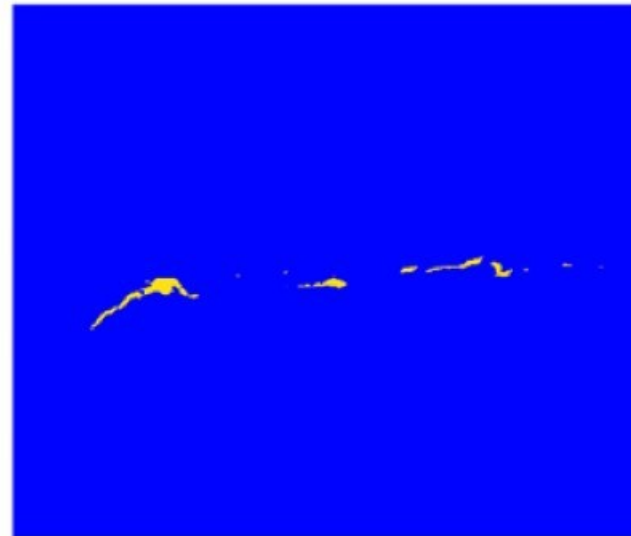
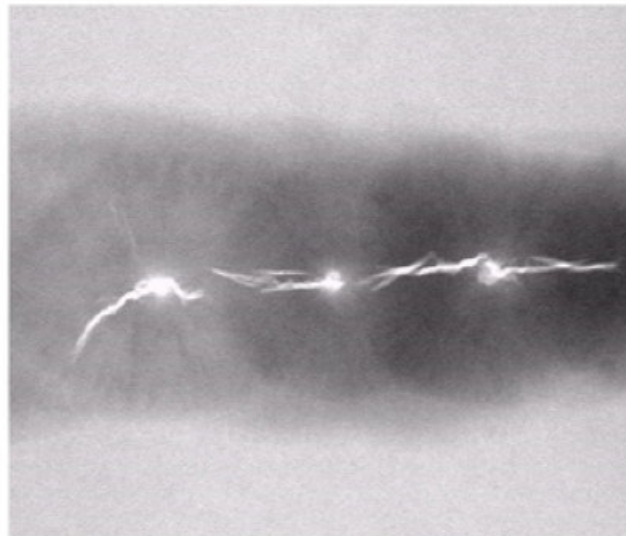
2. Full color image process: The process to manipulate real color images such as color photographs.

Pseudocolor Image Processing

Pseudocolor (sometimes called *false color*) image processing consists of assigning colors to gray values based on a specified criterion

Why we need to assign colors to gray scale image?

Answer: Human can distinguish different colors better than different shades of gray.



Intensity Slicing or Density Slicing

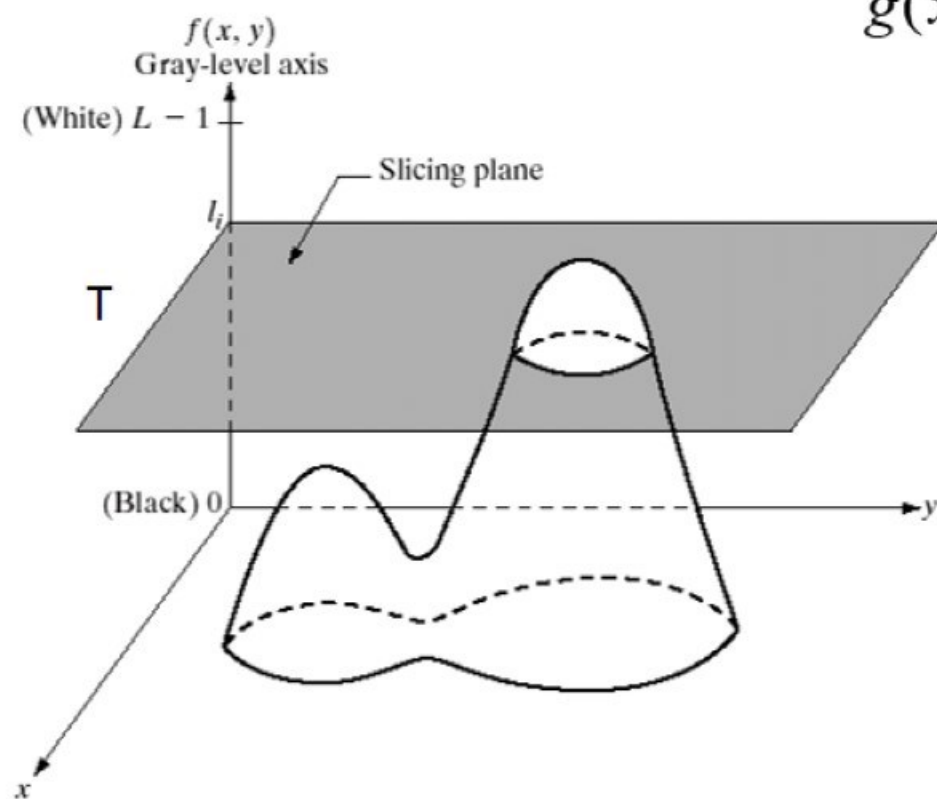
$$g(x, y) = \begin{cases} C_1 & \text{if } f(x, y) \leq T \\ C_2 & \text{if } f(x, y) > T \end{cases}$$

if $f(x, y) \leq T$

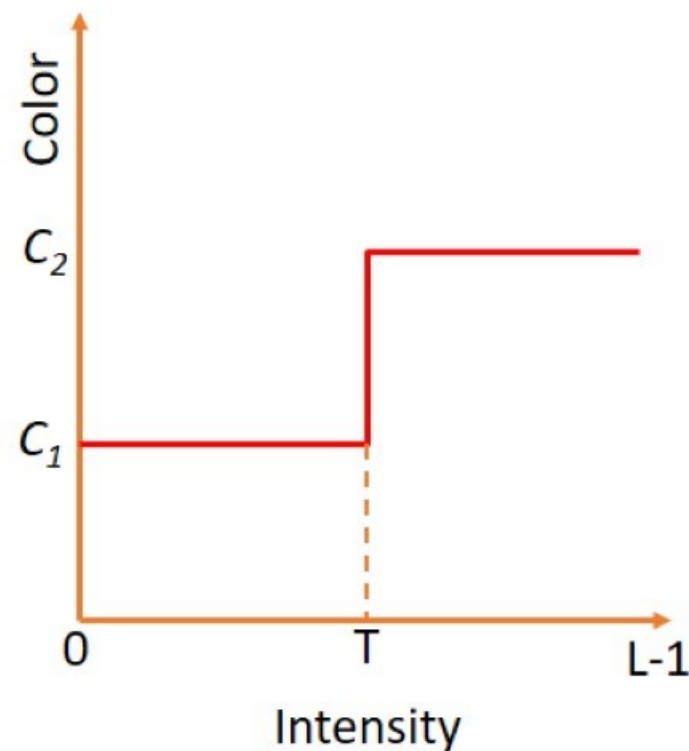
if $f(x, y) > T$

C_1 = Color No. 1

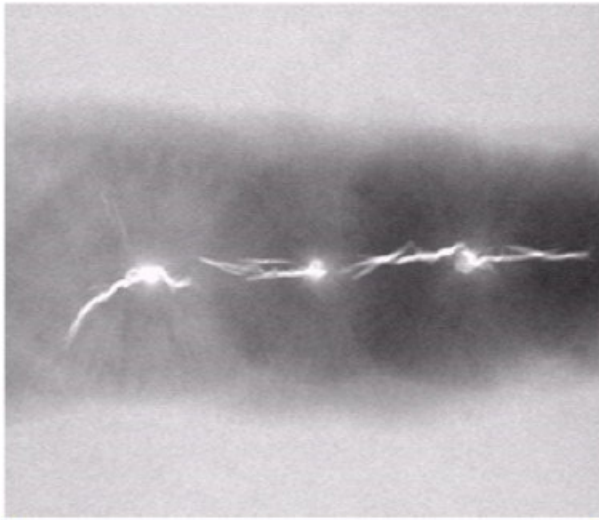
C_2 = Color No. 2



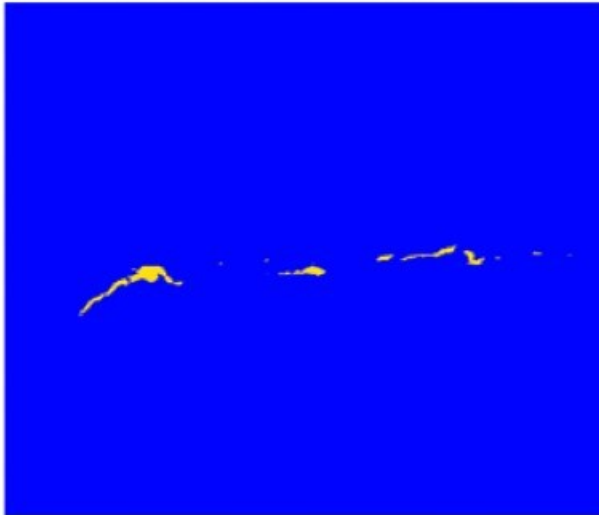
A gray scale image viewed as a 3D surface.



Intensity Slicing Example



An X-ray image of a weld with cracks



After assigning a yellow color to pixels with value 255 and a blue color to all other pixels.

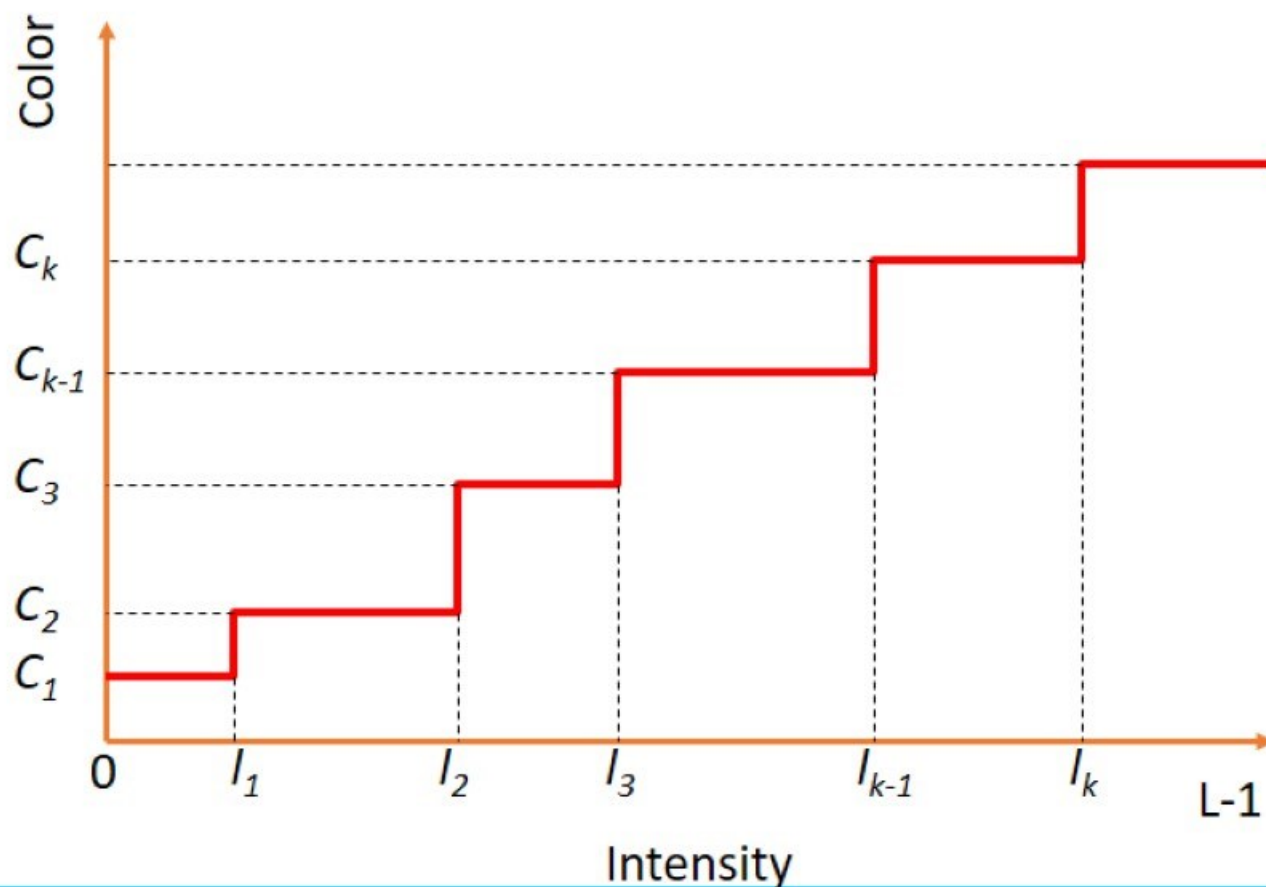
Multi Level Intensity Slicing

$$g(x, y) = C_k$$

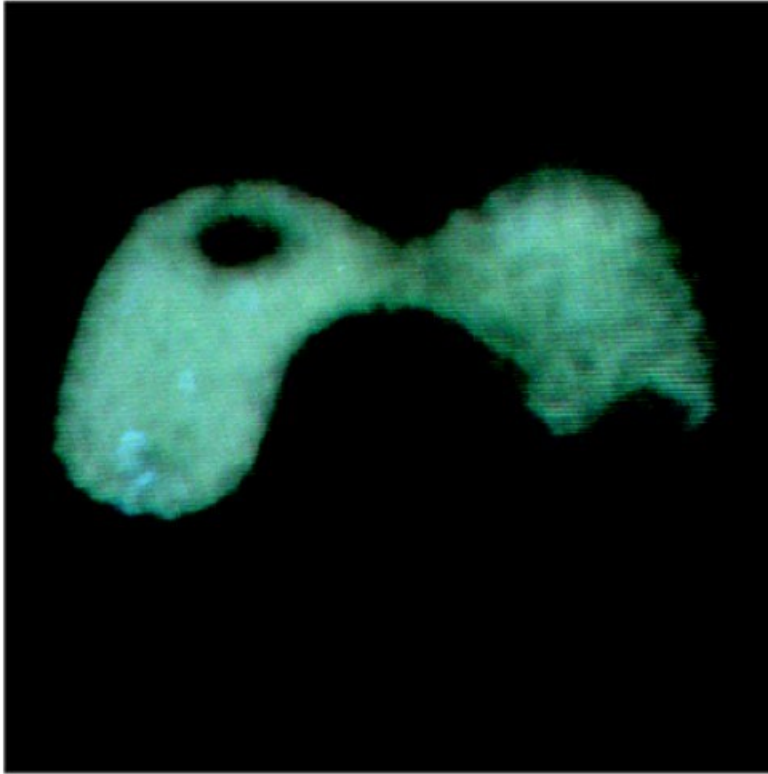
$$\text{for } l_{k-1} < f(x, y) \leq l_k$$

C_k = Color No. k

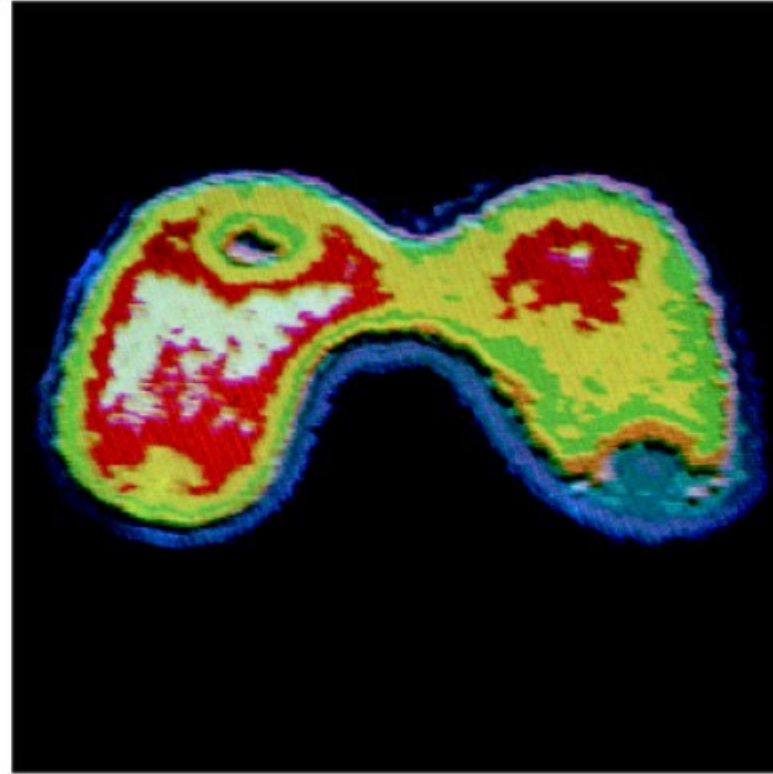
l_k = Threshold level k



Multi Level Intensity Slicing

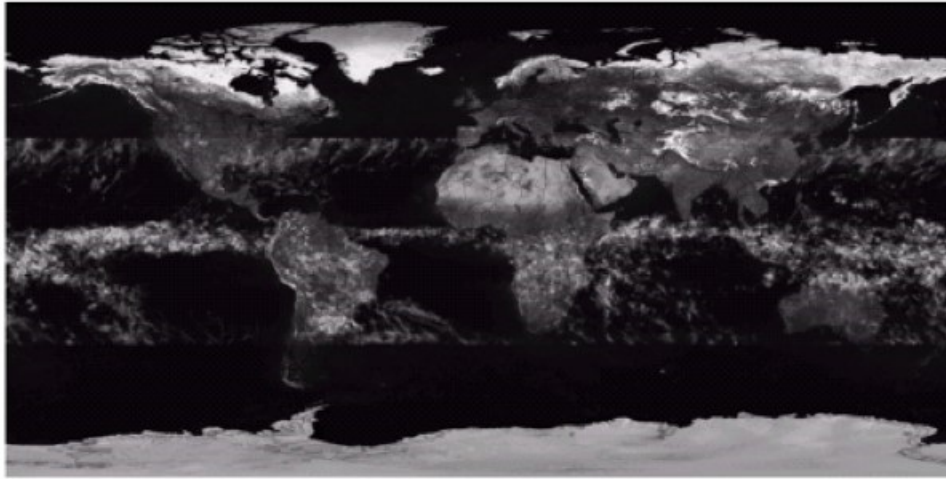


An X-ray image of the Picker Thyroid Phantom.

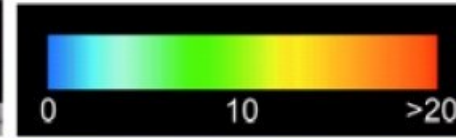


After density slicing into 8 colors

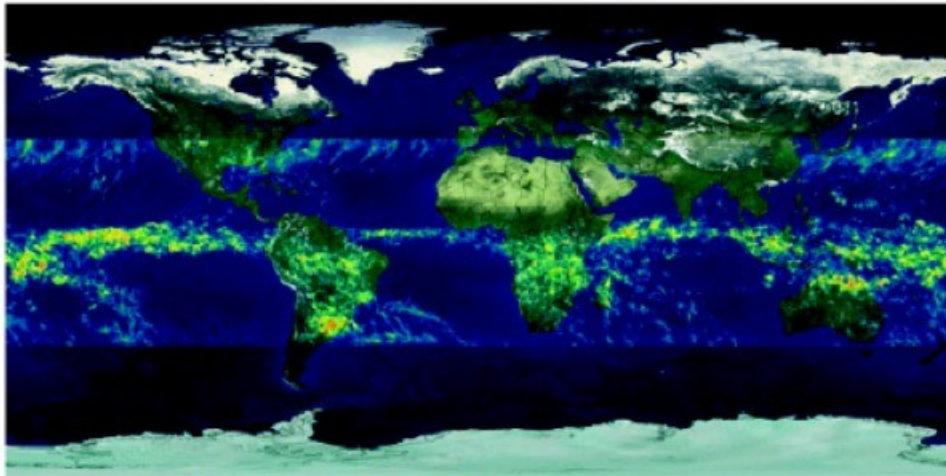
Color Coding Example



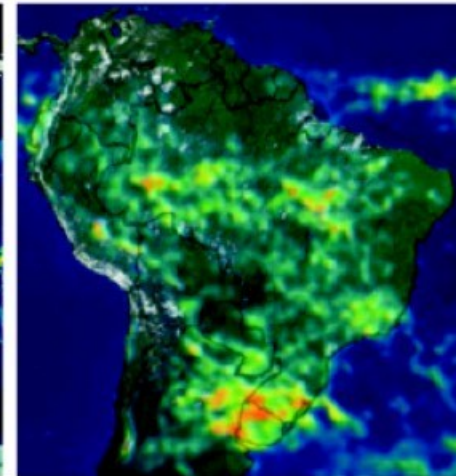
Gray-scale image of average monthly rainfall.



A unique color is assigned to each intensity value.



Color coded image

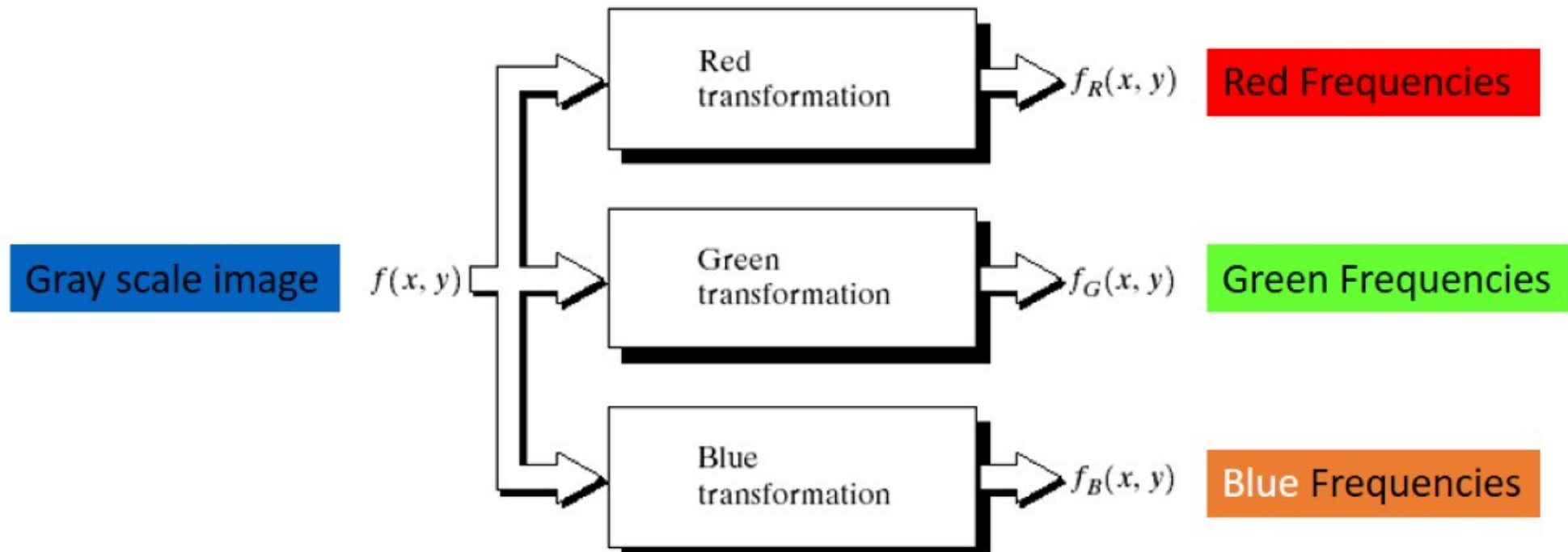


Color
map

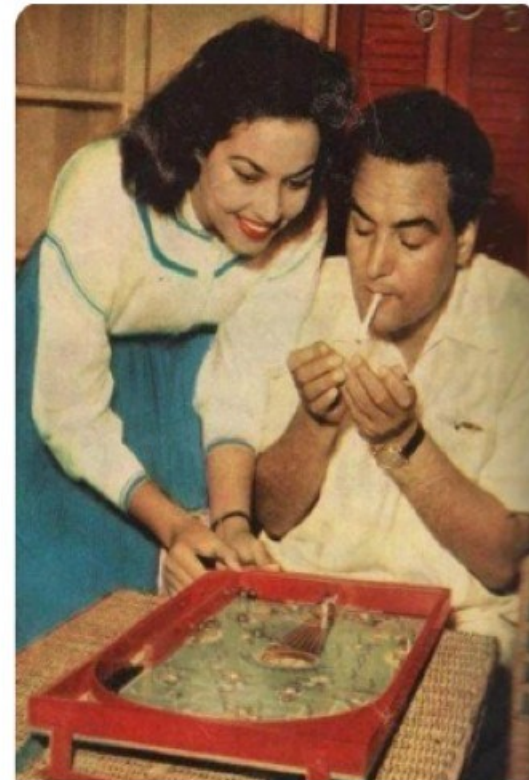
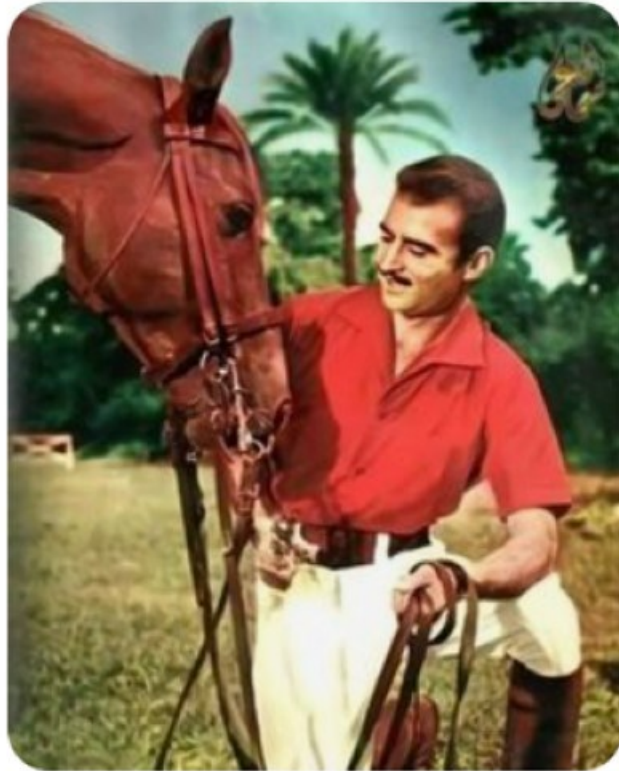
South America region

Gray Level to Color Transformation

Assigning colors to gray levels based on specific mapping functions



Gray Level to Color Transformation



Old Movies Coloring

The two methods:

1. Per-color-component processing: process each component separately.
2. Vector processing: treat each pixel as a vector to be processed.

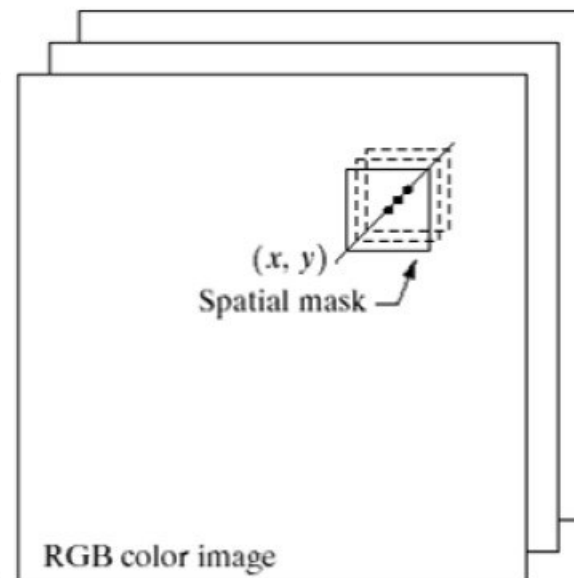
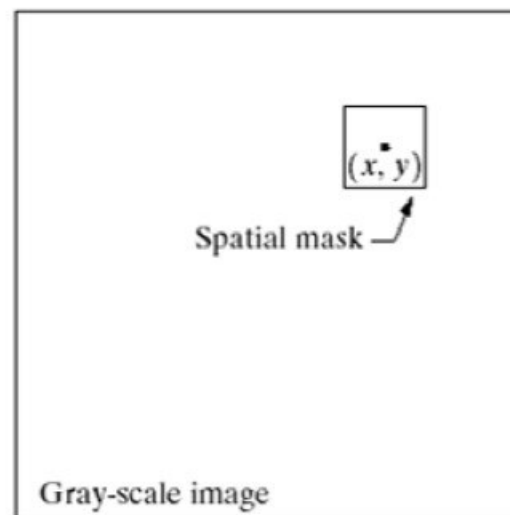
Example of per-color-component processing: smoothing an image
By smoothing each RGB component separately.



In order for per-component-image and vector-based processing to be equivalent, two conditions have to be satisfied:

first, the process has to be applicable to both vectors and scalars;

second, the operation on each component of a vector (i.e., each voxel) must be independent of the other components.

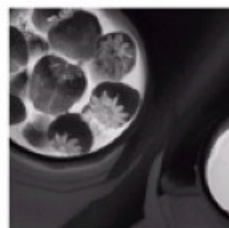


Example: Full-Color Image and Various Color Space Components



Full color

Color image



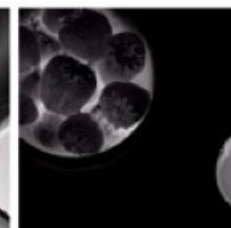
Cyan



Magenta



Yellow

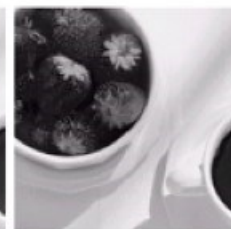


Black

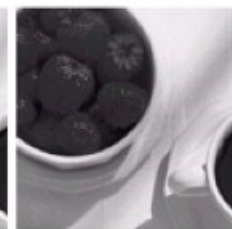
CMYK components



Red



Green

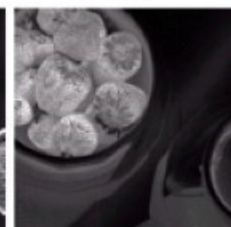


Blue

RGB components



Hue



Saturation



Intensity

HSI components

Use to **transform colors to colors**.

Formulation:
$$g(x, y) = T[f(x, y)]$$

$f(x, y)$ = input color image, $g(x, y)$ = output color image

T = operation on f **over a spatial neighborhood of (x, y)**

When **only data at one pixel is used** in the transformation, we can express the transformation as:

$$s_i = T_i(r_1, r_2, \dots, r_n) \quad i = 1, 2, \dots, n$$

Where r_i = color component of $f(x, y)$

s_i = color component of $g(x, y)$

For RGB images, $n = 3$

Example: Color Transformation

Formula for RGB:

$$s_R(x, y) = kr_R(x, y)$$

$$s_G(x, y) = kr_G(x, y)$$

$$s_B(x, y) = kr_B(x, y)$$

Formula for HSI:

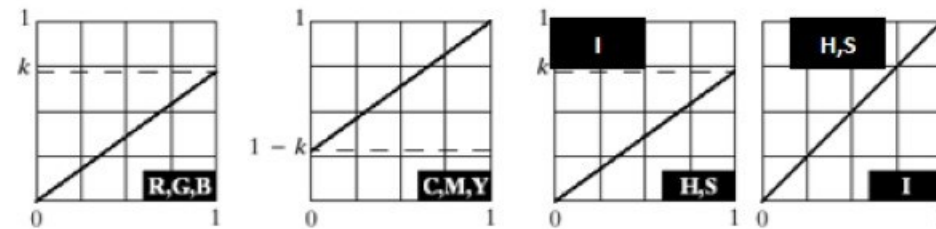
$$s_I(x, y) = kr_I(x, y)$$

Formula for CMY:

$$s_C(x, y) = kr_C(x, y) + (1 - k)$$

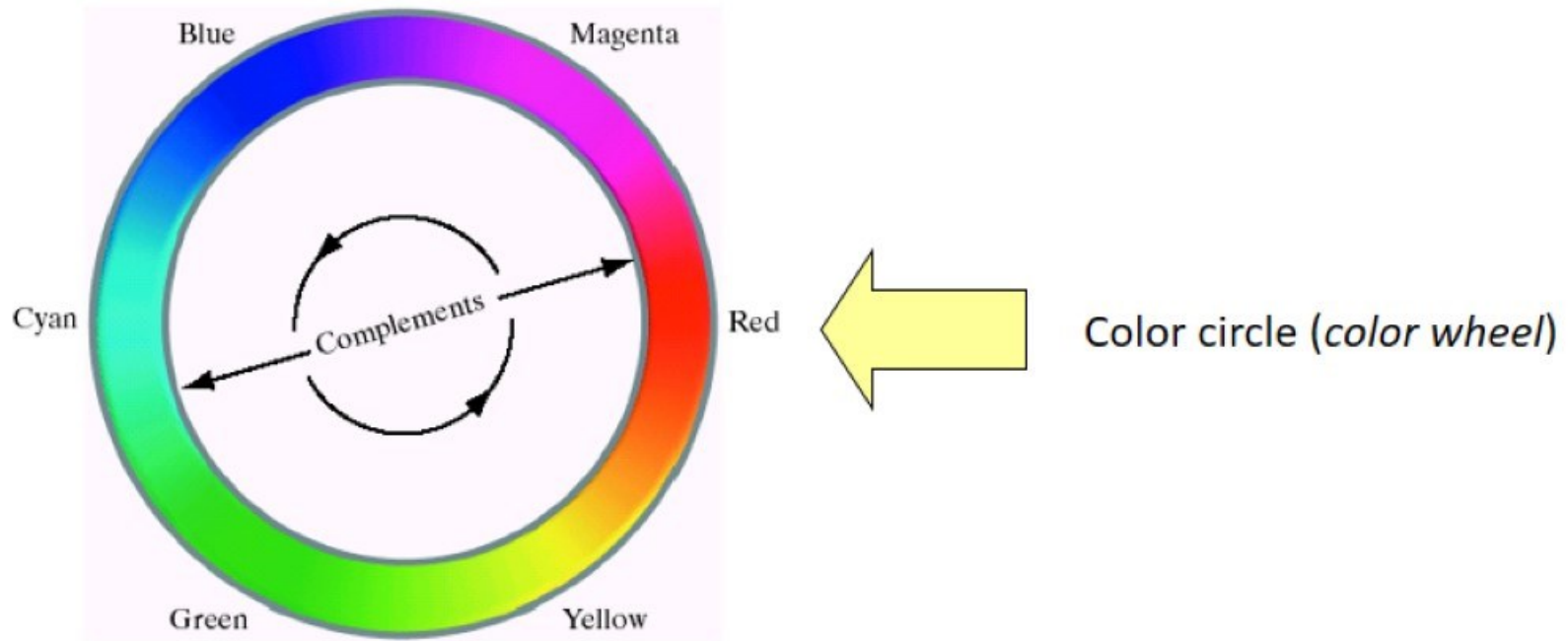
$$s_M(x, y) = kr_M(x, y) + (1 - k)$$

$$s_Y(x, y) = kr_Y(x, y) + (1 - k)$$

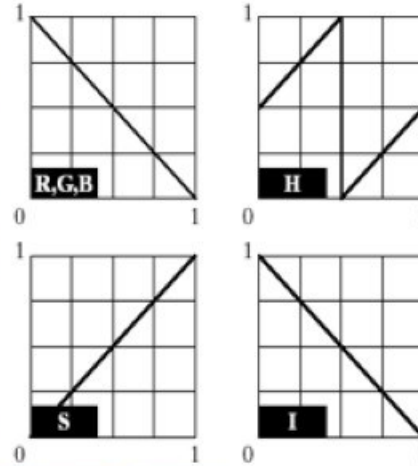


These 3 transformations give the same results.

Color complement replaces each color with its opposite color in the color circle of the Hue component. **This operation is analogous to image negative in a gray scale image.**



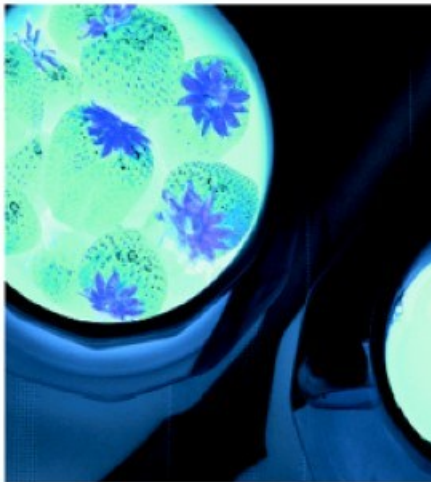
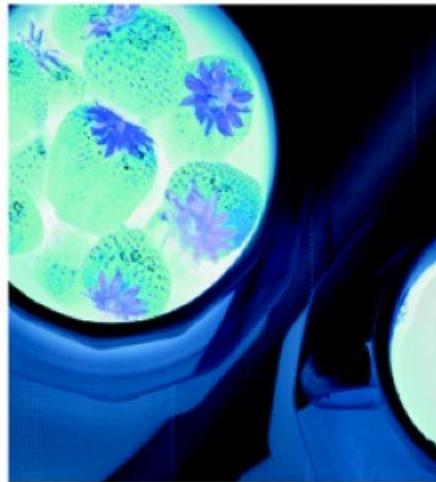
Color Complement Transformation Example



a b
c d

FIGURE 6.33

Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.



Color Slicing Transformation

We can perform “slicing” in color space: if the color of each pixel is far from a desired color more than threshold distance, we set that color to some specific color such as gray, otherwise we keep the original color unchanged.

$$s_i = \begin{cases} 0.5 & \text{if } \left[|r_j - a_j| > \frac{W}{2} \right]_{\text{any } 1 \leq j \leq n} \\ r_i & \text{otherwise} \end{cases}$$

$i = 1, 2, \dots, n$

or

$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^n (r_j - a_j)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases}$$

$i = 1, 2, \dots, n$



Set to gray



Keep the original color



Set to gray



Keep the original color

Color Slicing Transformation Example



Original image



a b



After color slicing

FIGURE 6.34 Color slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at $(0.6863, 0.1608, 0.1922)$, and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color $(0.5, 0.5, 0.5)$.

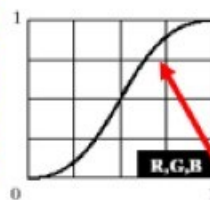
Flat (low contrast image)



Flat



Corrected



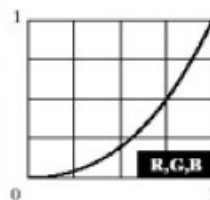
light (high key) Image



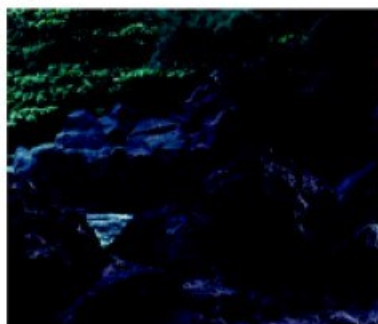
Light



Corrected



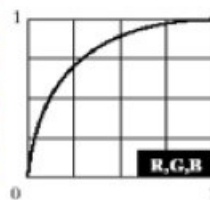
dark (low key) color image



Dark



Corrected



In these examples, only brightness and contrast are adjusted while keeping color unchanged.

This can be done by using the same transformation for all RGB components.

Contrast enhancement

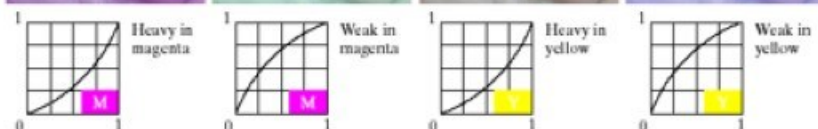
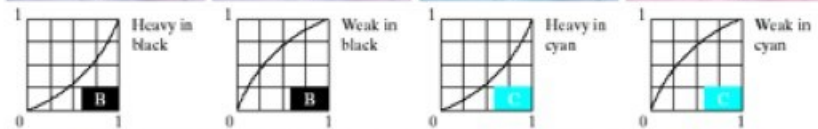
Power law transformations

Color Balancing Correction Examples



Original/Corrected

FIGURE 6.36 Color balancing corrections for CMYK color images.



Color imbalance: **primary color components in white area are not balance**. We can measure these components by using a color spectrometer.

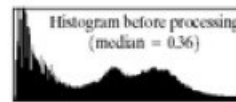
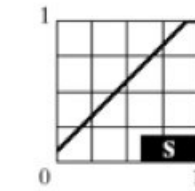
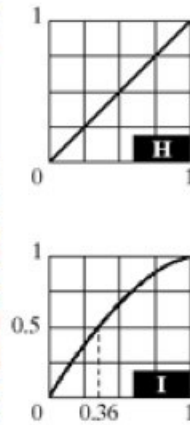
Color balancing can be performed by adjusting color components separately as seen in this slide.

- ❖ Histogram equalization of a color image can be performed by adjusting color intensity uniformly while leaving color unchanged.
- ❖ The HSI model is suitable for histogram equalization where **only Intensity (I) component is equalized.**

$$\begin{aligned} s_k &= T(r_k) = \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{N} \end{aligned}$$

where r and s are intensity components of input and output color image.

Histogram Equalization of a Full-Color Image



a b
c d

FIGURE 6.37
Histogram equalization (followed by saturation adjustment) in the HSI color space.



Normalized

Two Methods:

1. **Per-color-plane method:** for RGB, CMY color models
Smooth each color plane using moving averaging and
the combine back to RGB

$$\bar{\mathbf{c}}(x, y) = \frac{1}{K} \sum_{(x, y) \in S_{xy}} \mathbf{c}(x, y) = \begin{bmatrix} \frac{1}{K} \sum_{(x, y) \in S_{xy}} R(x, y) \\ \frac{1}{K} \sum_{(x, y) \in S_{xy}} G(x, y) \\ \frac{1}{K} \sum_{(x, y) \in S_{xy}} B(x, y) \end{bmatrix}$$

2. **Smooth only Intensity component** of a HSI image while leaving
H and S unmodified.

Note: 2 methods are not equivalent.

Color Image Smoothing Example (cont.)

Color image



Red



Green



Blue



Color Image Smoothing Example (cont.)

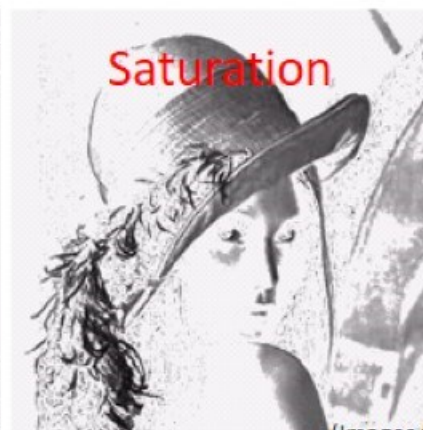


Color image

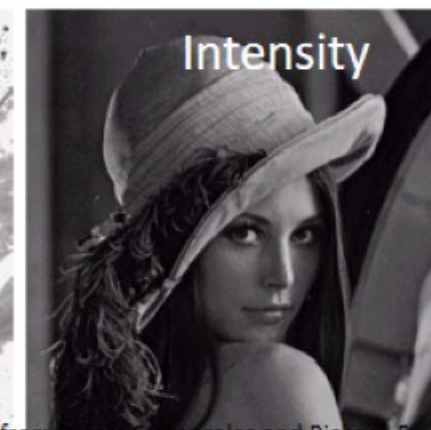
HSI Components



Hue



Saturation

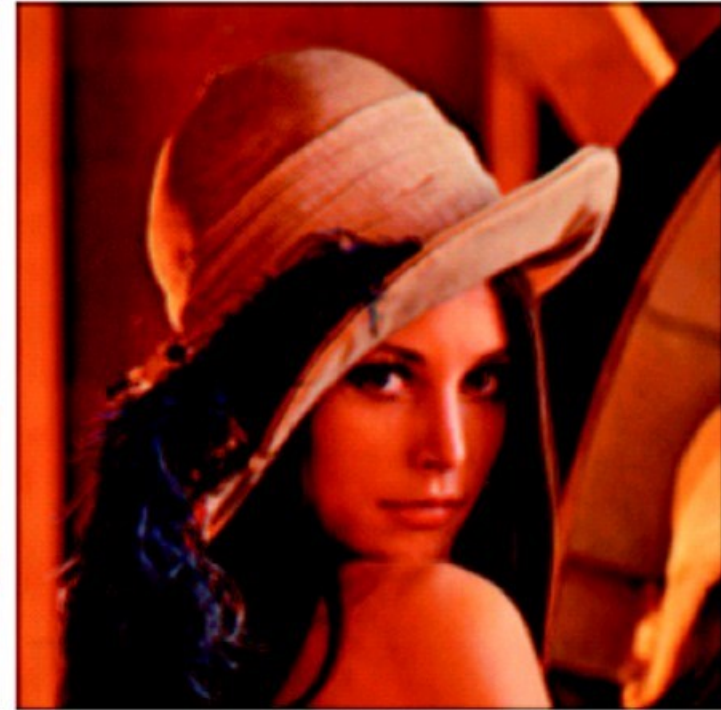


Intensity

Color Image Smoothing Example (cont.)

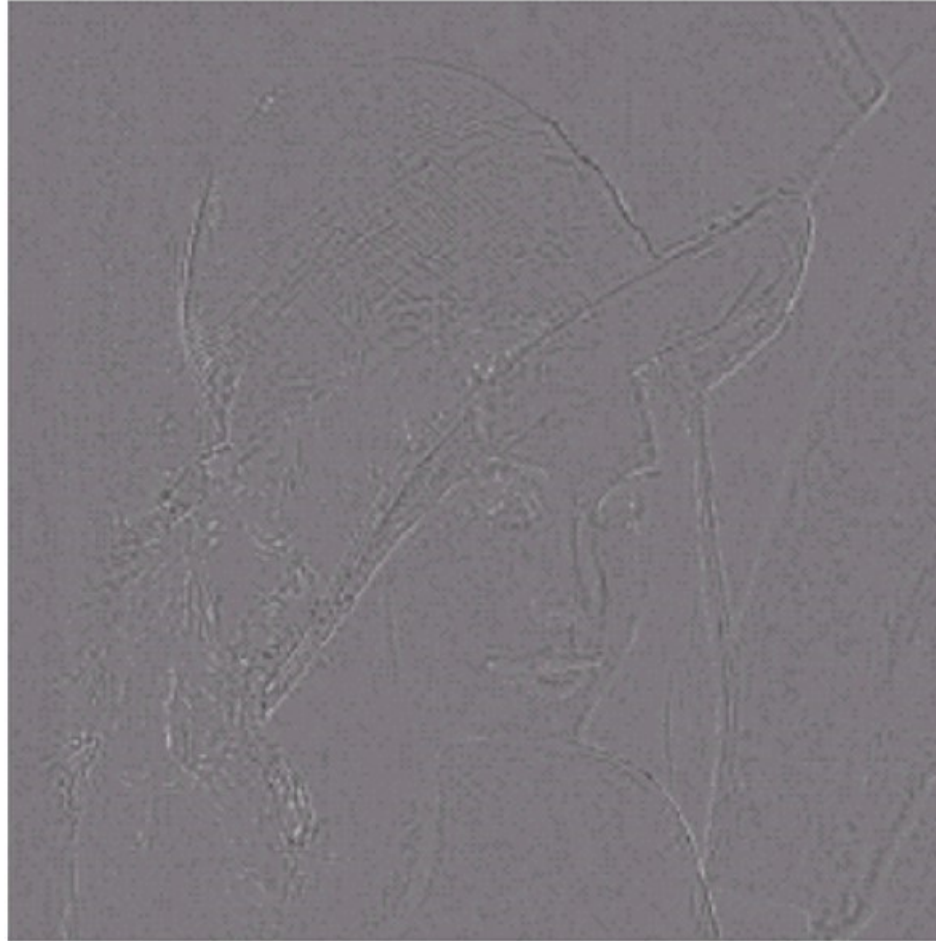


Smooth all RGB components



Smooth only I component of HSI
(faster)

Color Image Smoothing Example (cont.)



Difference between
smoothed results from 2
methods in the previous
slide.

We can do in the same manner as color image smoothing:

1. Per-color-plane method for RGB,CMY images
2. Sharpening only I component of a HSI image



Sharpening all RGB components



Sharpening only I component of HSI



Difference between
sharpened results from 2
methods in the previous
slide.



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY



Thank You

