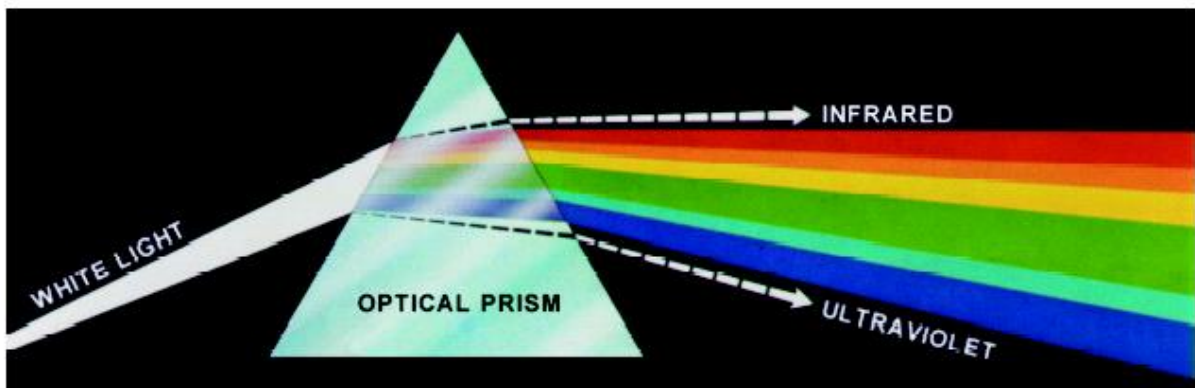


## Unit 8: Color Image Processing

### Colour Fundamentals

In 1666 Sir Isaac Newton discovered that when a beam of sunlight passes through a glass prism, the emerging beam is split into a spectrum of colours

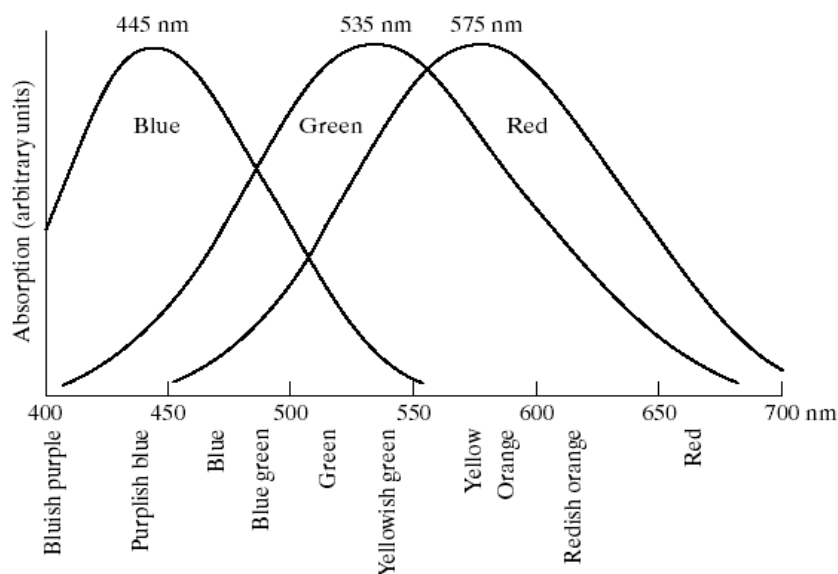


The colours that humans and most animals perceive in an object are determined by the nature of the light reflected from the object

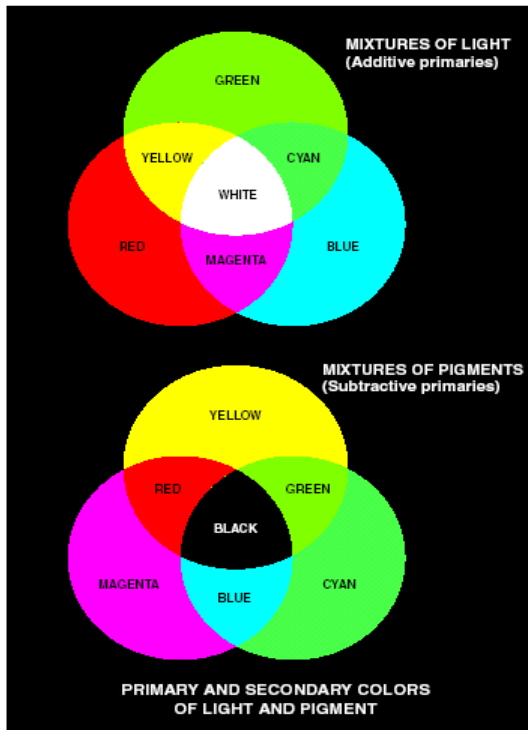
For example, green objects reflect light with wave lengths primarily in the range of 500 – 570 nm while absorbing most of the energy at other wavelengths. Approximately 66% of these cones are sensitive to red light, 33% to green light and 6% to blue light

Absorption curves for the different cones have been determined experimentally

Strangely these do not match the CIE standards for red (700nm), green (546.1nm) and blue (435.8nm) light as the standards were developed before the experiments!



## 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 used in the case of pigments in printing devices

White subtracted by CMY to get Black

## Properties of Color

Three basic qualities are used to describe the quality of a chromatic light source:

- **Radiance:** the total amount of energy that flows from the light source (measured in watts)
- **Luminance:** the amount of energy an observer *perceives* from the light source (measured in lumens)
  - Note we can have high radiance, but low luminance
- **Brightness:** a subjective (practically unmeasurable) notion that embodies the intensity of light

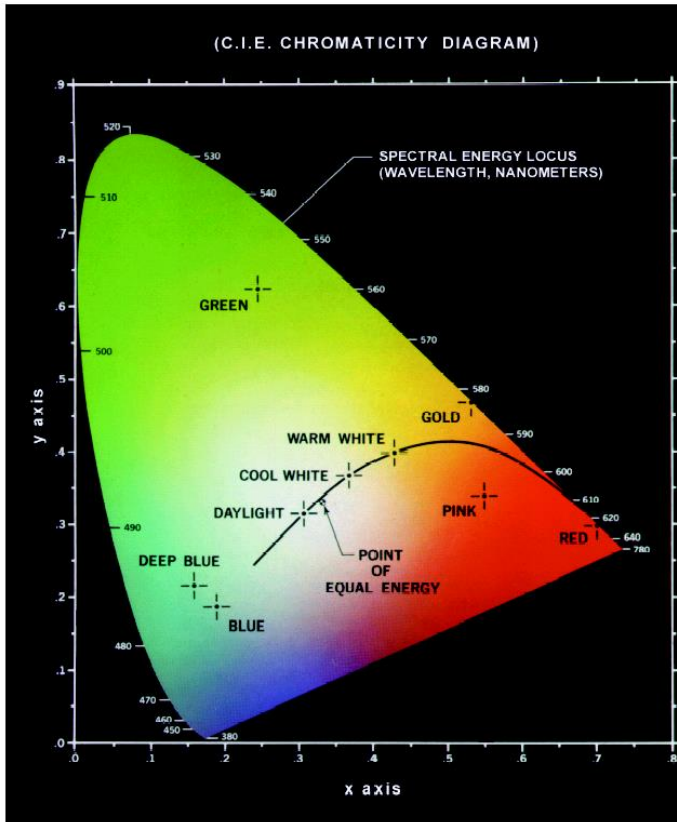
## CIE Chromacity Diagram

Specifying colours systematically can be achieved using the CIE **chromacity diagram**

On this diagram the x-axis represents the proportion of red and the y-axis represents the proportion of red used

The proportion of blue used in a colour is calculated as:

$$z = 1 - (x + y)$$

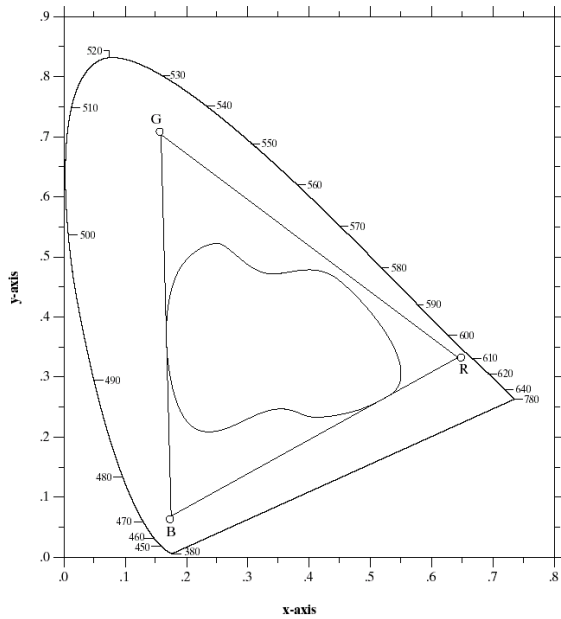


Any colour located on the boundary of the chromacity chart is fully saturated

The point of equal energy has equal amounts of each colour and is the CIE standard for pure white

Any straight line joining two points in the diagram defines all of the different colours that can be obtained by combining these two colours additively This can be easily extended to three points. This means the entire colour range cannot be displayed based on any three colours . The triangle shows the typical colour gamut produced by RGB monitors.

The strange shape is the gamut achieved by high quality colour printers



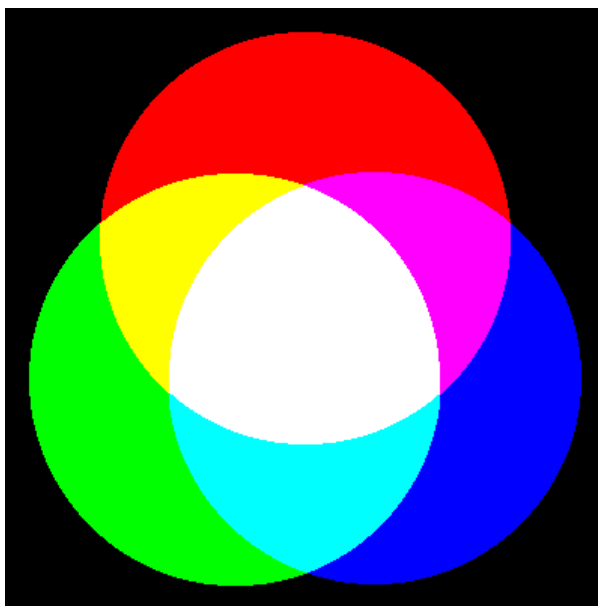
## Colour Models

From the previous discussion it should be obvious that there are different ways to model colour

We will consider two very popular models used in colour image processing:

- RGB (**R**ed **G**reen **B**lue)
- CMY (Cyan Magenta Yellow)
- HSI(**H**ue **S**aturation **I**ntensity)

### RGB Model



The RGB colour space is related to human vision through the tristimulus theory of colour vision.

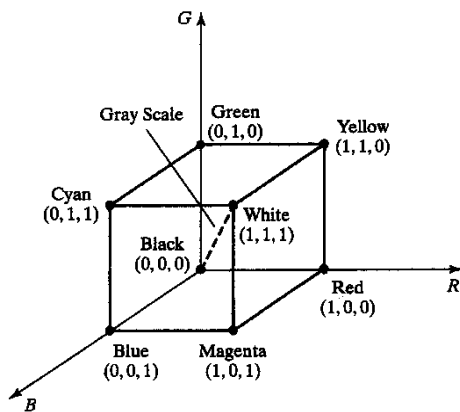
The RGB is an additive colour model. The primary colours red, green and blue are combined to reproduce other colours.

In the RGB colour space, a colour is represented by a triplet  $(r, g, b)$

- $r$  gives the intensity of the red component
- $g$  gives the intensity of the green component
- $b$  gives the intensity of the blue component

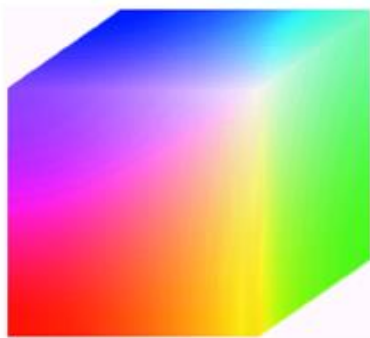
Here we assume that  $r, g, b$  are real numbers in the interval  $[0, 1]$ .

You will often see the values of  $r, g, b$  as integers in the interval  $[0, 255]$ .



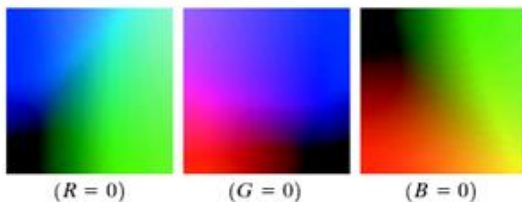
RGB color models based on Cartesian coordinate system

### RGB Color Cube

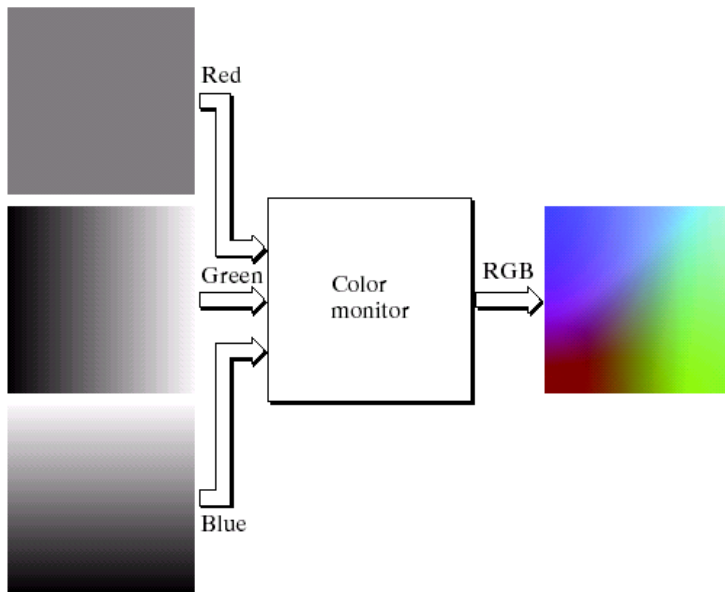


$R = 8 \text{ bits}$   
 $G = 8 \text{ bits}$   
 $B = 8 \text{ bits}$

Color depth 24 bits  
 = 16777216 colors



Hidden faces of the cube



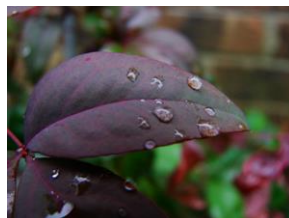
Most modern computer monitors can transmit “true color,” or 24-bit color. This means each “channel” (R, G, or B) contains 8 bits per channel that can transmit color.

Eight bits means the channel can make eight combinations of on or off of the color, per pixel, 256 colors total. You have three channels. How many colors can be generated?

$256 \times 256 \times 256 = 16,777,216$  possible colors.

Eight-bit color also exists, 256 colors total. These are called “web-safe” colors, because they are sure to render accurately on anyone’s monitor. Nowadays we don’t have to worry about that as much.

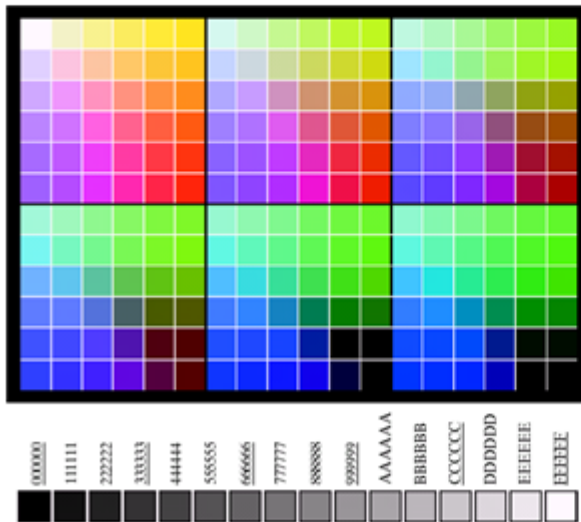
(Below: 8-bit vs. 24-bit color.)



## Safe RGB Colors

Safe RGB colors: a subset of RGB colors.

There are 216 colors common in most operating systems.



**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).

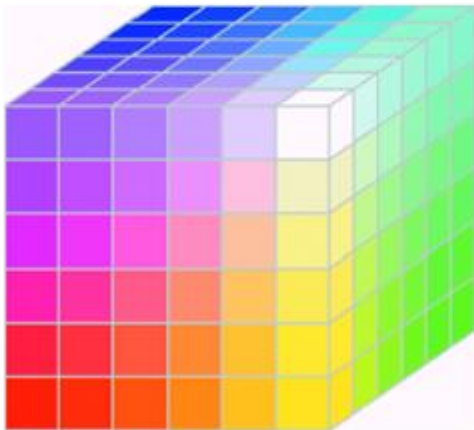
30

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.

## 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.

31

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.

## CMY Color model

Additive color won't work for printing because we can't begin with black.

We must begin with a piece of paper, and that's usually white.

White, as we know, is all colors. So we can't add to all colors. We must subtract.

Furthermore, an offset printing press can't generate the enormous number of colors available on a computer screen.

We need to run a piece of paper through the press for each ink. Printed color, therefore, is based on the subtractive system.

While the additive primaries (used to generate all colors ) are RGB, beginning with black...

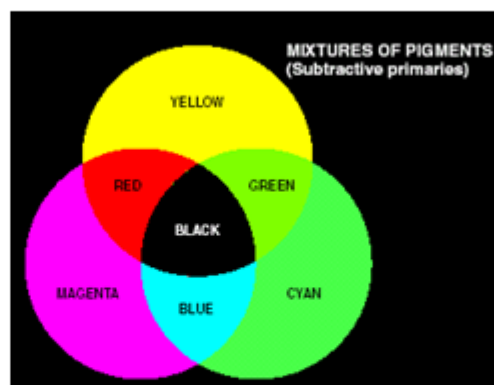
...the subtractive primaries are Cyan, Magenta, Yellow and Black (CMYK), and begin with white. Cyan=blue-green. Magenta=red-blue. Yellow=red-green. Note the relationship between the additive and subtractive primaries.

You can actually project the additive colors to produce the subtractive.



## CMY Color model

C = Cyan  
M = Magenta  
Y = Yellow  
K = Black



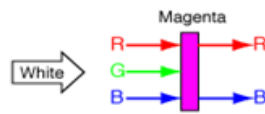
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



## CMY Color model

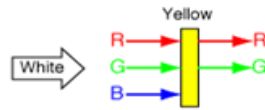
- Why?

- Pigments absorb light



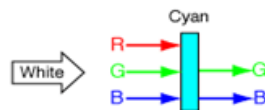
- Thinking:

- the Color Filters

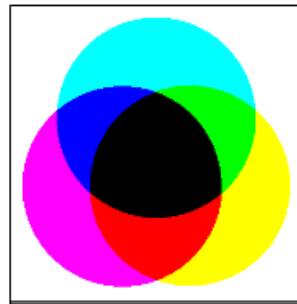


- Question:

- Yellow + Cyan=?

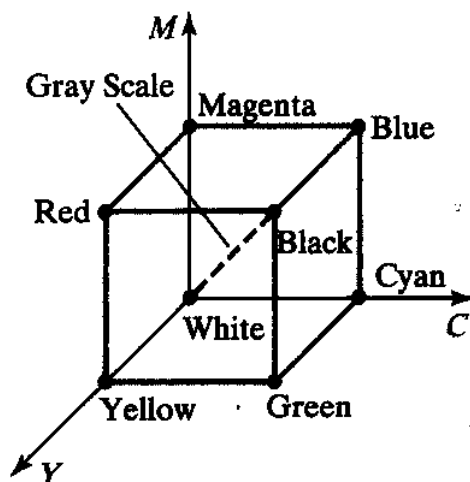


## CMY Color model



Passive displays, such as color inkjet printers, **absorb** light instead of emitting it. Combinations of cyan, magenta and yellow inks are used. This is a **subtractive** color model.

CMY and CMYK color space



The conversion from RGB to CMY is given by the formula

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

**Example 1:** The red colour is written in RGB as (1,0,0). In CMY it is written as

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

that is, magenta and yellow.

**Example 2:** The magenta is written in CMY as (0,1,0). In RGB it is written as

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

giving,

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

that is, red and blue.

## CMYK Colour Model

For printing and graphics art industry, CMY is not enough; a fourth primary, K which stands for black, is added.

## CMYK Color model



The image on the left was printed with only CMY inks  
Black inks add contrast and depth to image on the image on  
the right

44

Subtractive primaries are based on ink colors of CMYK. Black is abbreviated “K” by tradition, perhaps because it is the “key” color. In color printing, you need black to make the other colors vibrant and snappy.

This is why the subtractive process is also called the four-color process, producing color separations, or “seps.” Colors used are called the process colors.

## HSI color model

RGB is useful for hardware implementations and is serendipitously related to the way in which the human visual system work. However, RGB is not a particularly intuitive way in which to describe colours Rather when people describe colours they tend to use hue, saturation and brightness RGB is great for colour generation, but HSI is great for colour description.

The HSI model uses three measures to describe colours:

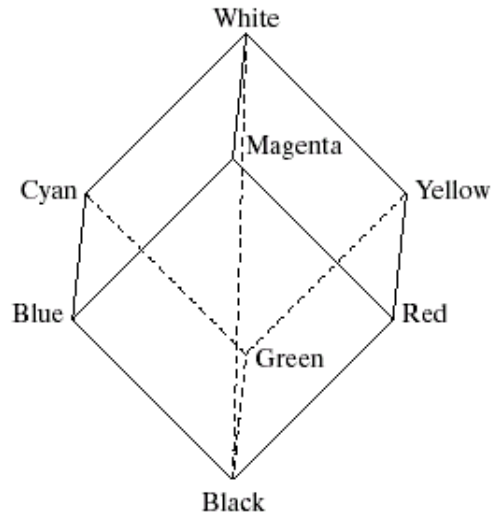
- **Hue:** A colour attribute that describes a pure colour (pure yellow, orange or red)
- **Saturation:** Gives a measure of how much a pure colour is diluted with white light
- **Intensity:** Brightness is nearly impossible to measure because it is so subjective. Instead we use intensity. Intensity is the same achromatic notion that we have seen in grey level images

Intensity can be extracted from RGB images – which is not surprising if we stop to think about it.

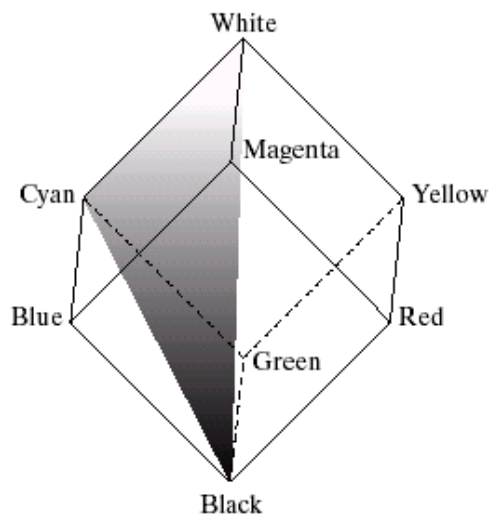
Remember the diagonal on the RGB colour cube that we saw previously ran from black to white

Now consider if we stand this cube on the black vertex and position the white vertex directly above it.

Now the intensity component of any colour can be determined by passing a plane perpendicular to the intensity axis and containing the colour point. The intersection of the plane with the intensity axis gives us the intensity component of the colour.

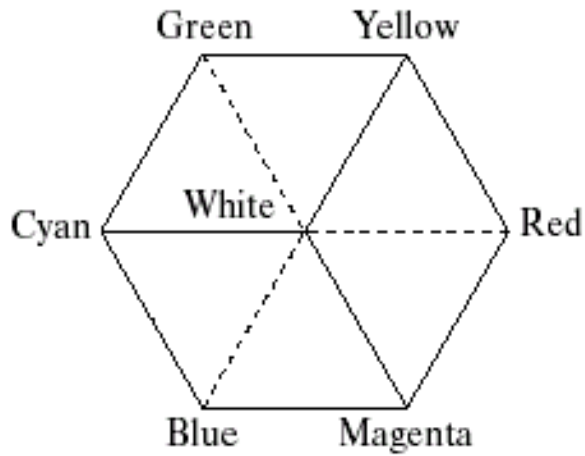


In a similar way we can extract the hue from the RGB colour cube. Consider a plane defined by the three points cyan, black and white. All points contained in this plane must have the same hue (cyan) as black and white cannot contribute hue information to a colour.



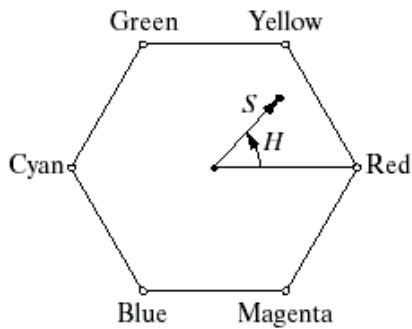
Consider if we look straight down at the RGB cube as it was arranged previously

We would see a hexagonal shape with each primary colour separated by  $120^\circ$  and secondary colours at  $60^\circ$  from the primaries. So the HSI model is composed of a vertical intensity axis and the locus of colour points that lie on planes perpendicular to that axis.

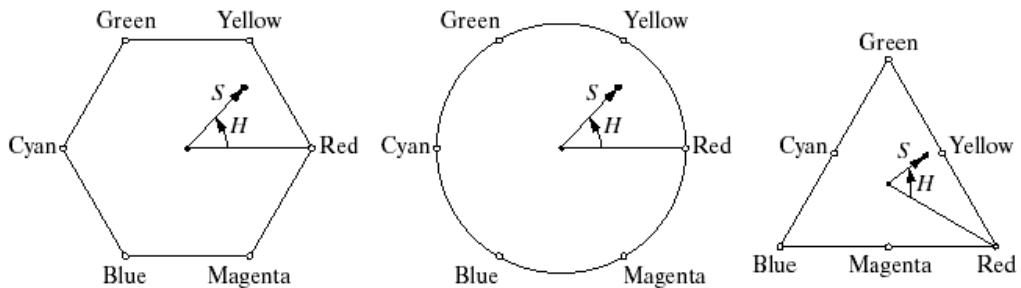


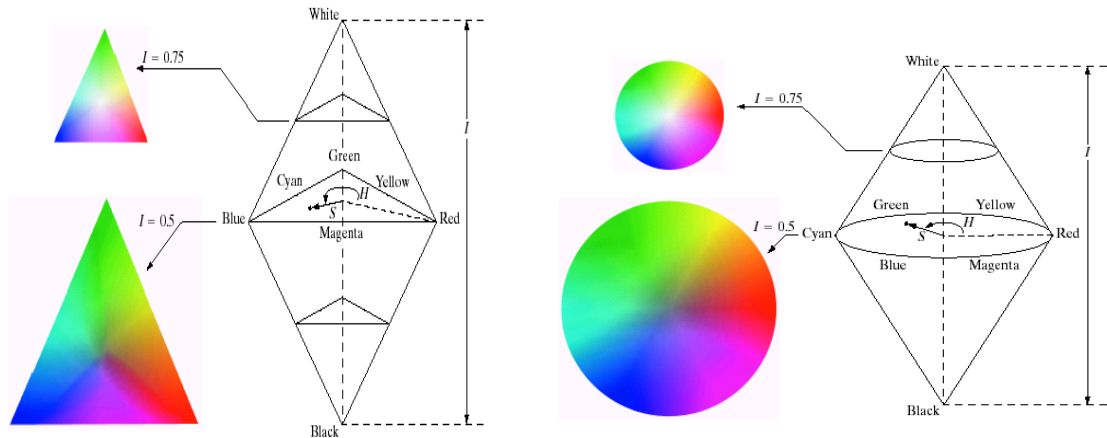
In the below image we see a hexagonal shape and an arbitrary colour point

- The hue is determined by an angle from a reference point, usually red
- The saturation is the distance from the origin to the point
- The intensity is determined by how far up the vertical intensity axis this hexagonal plane sits (not apparent from this diagram)



Because the only important things are the angle and the length of the saturation vector this plane is also often represented as a circle or a triangle





## Converting From RGB To HSI

### Converting Colors from RGB to HSI

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

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

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

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

58

### 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:  $120 \leq H < 240$

$$H = H - 120$$

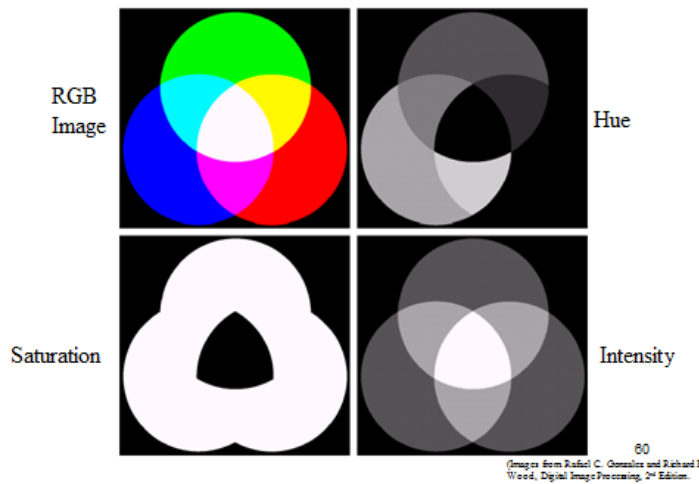
$$R = I(1 - S)$$

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

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

59

### Example: HSI Components of RGB Colors



## *Color Image Processing*

There are 2 types of color image processes

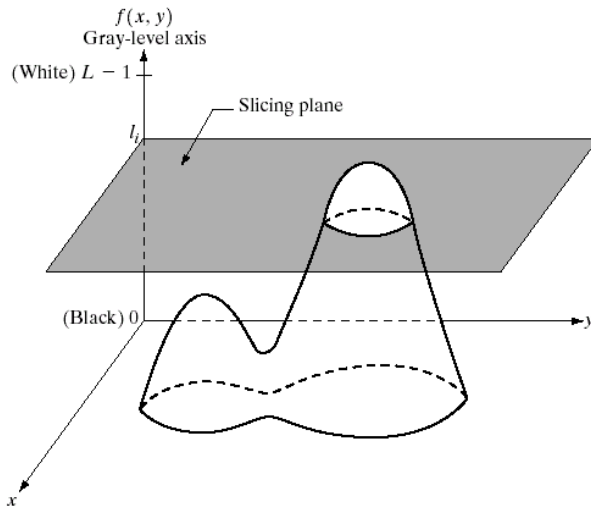
1. Pseudocolor image process: Assigning colors to gray values based on a specific criterion. Gray scale images to be processed may be a single image or multiple images such as multispectral images
2. Full color image process: The process to manipulate real color images such as color photographs.

### *Pseudocolor Image Processing*

Pseudocolour (also called false colour) image processing consists of assigning colours to grey values based on a specific criterion

The principle use of pseudocolour image processing is for human visualisation

- Humans can discern between thousands of colour shades and intensities, compared to only about two dozen or so shades of grey
- Intensity slicing and colour coding is one of the simplest kinds of pseudocolour image processing
- First we consider an image as a 3D function mapping spatial coordinates to intensities (that we can consider heights)
- Now consider placing planes at certain levels parallel to the coordinate plane
- If a value is one side of such a plane it is rendered in one colour, and a different colour if on the other side

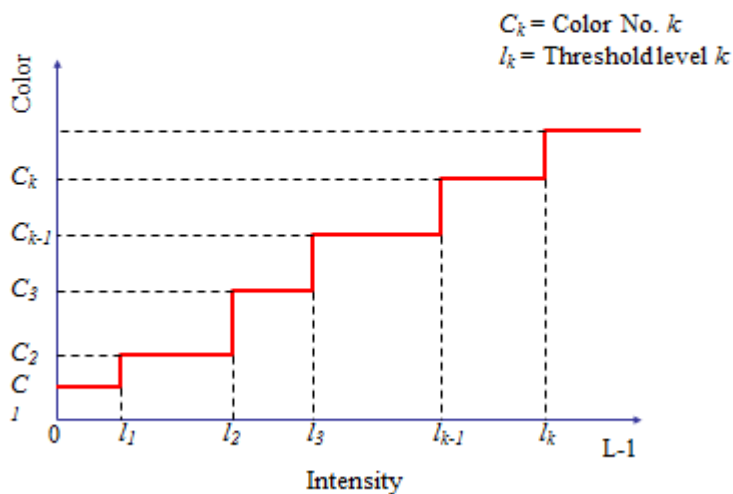


In general intensity slicing can be summarised as:

- Let  $[0, L-1]$  represent the grey scale
- Let  $l_0$  represent black  $[f(x, y) = 0]$  and let  $l_{L-1}$  represent white  $[f(x, y) = L-1]$
- Suppose  $P$  planes perpendicular to the intensity axis are defined at levels  $l_1, l_2, \dots, l_p$
- Assuming that  $0 < P < L-1$  then the  $P$  planes partition the grey scale into  $P + 1$  intervals  $V_1, V_2, \dots, V_{P+1}$
- Grey level colour assignments can then be made according to the relation:
- where  $c_k$  is the colour associated with the  $k^{\text{th}}$  intensity level  $V_k$  defined by the partitioning planes at  $l = k - 1$  and  $l = k$

## Multi Level Intensity Slicing

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

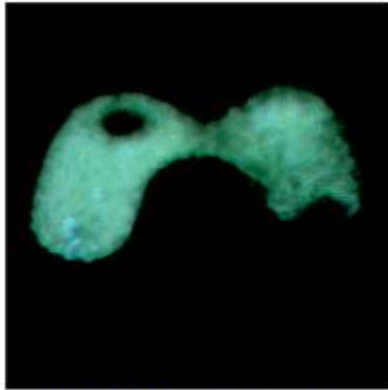




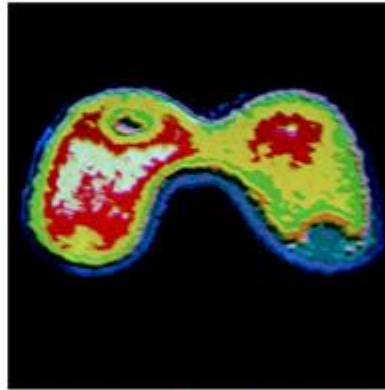
## Multi Level Intensity Slicing Example

$$g(x, y) = C_k \quad \text{for } I_{k-1} < f(x, y) \leq I_k$$

$C_k$  = Color No.  $k$   
 $I_k$  = Threshold level  $k$



An X-ray image of the Picker Thyroid Phantom.

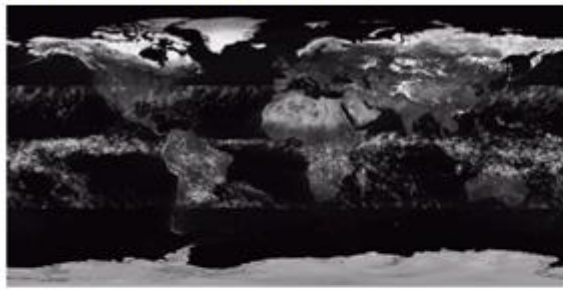


After density slicing into 8 colors

67

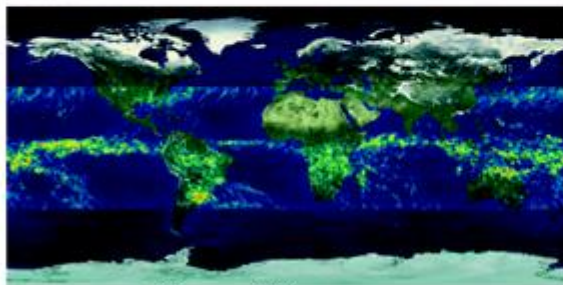
(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.

## Color Coding Example

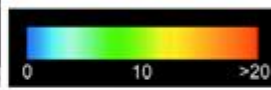


A unique color is assigned to each intensity value.

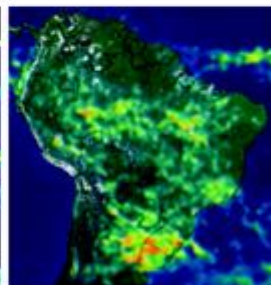
Gray-scale image of average monthly rainfall.



Color coded image



Color map



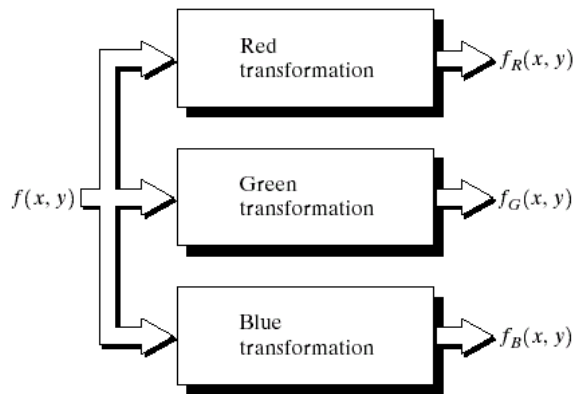
South America region

68

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.

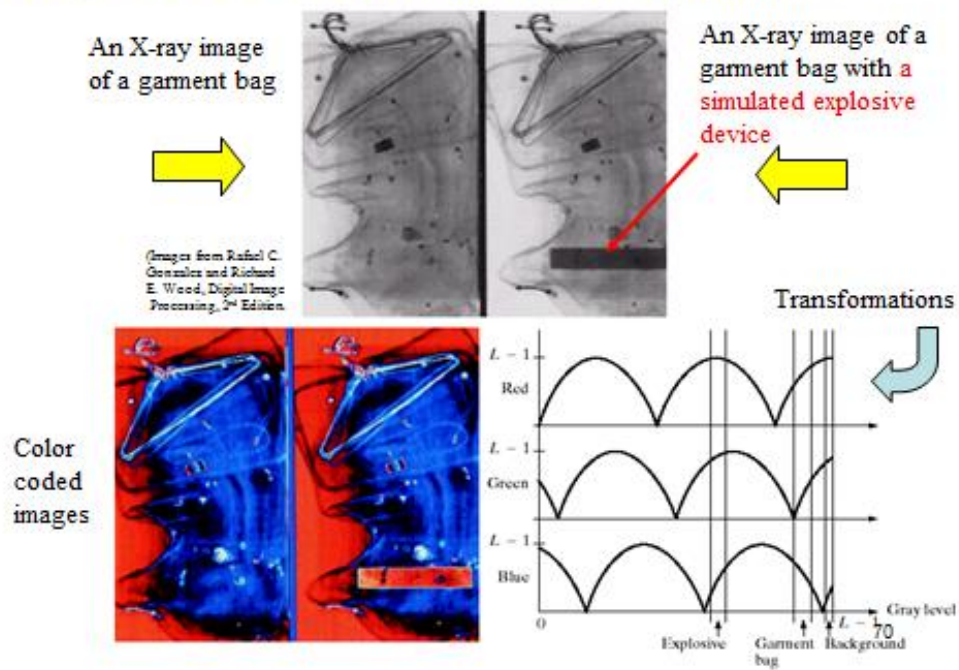
## Gray Level to Color Transformation

Assigning colors to gray levels based on specific mapping functions

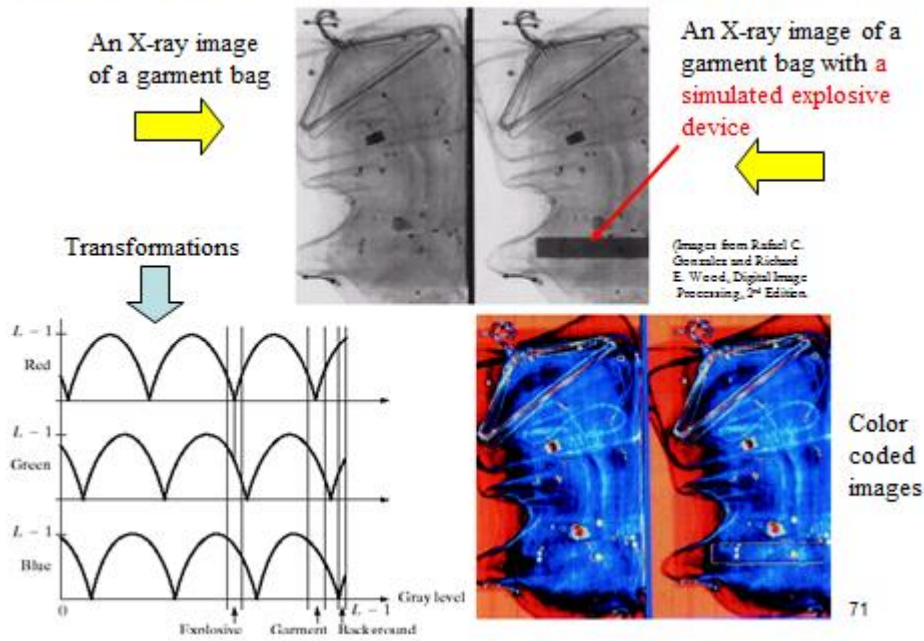


**FIGURE 6.23** Functional block diagram for pseudocolor image processing.  $f_R$ ,  $f_G$ , and  $f_B$  are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

### Gray Level to Color Transformation Example

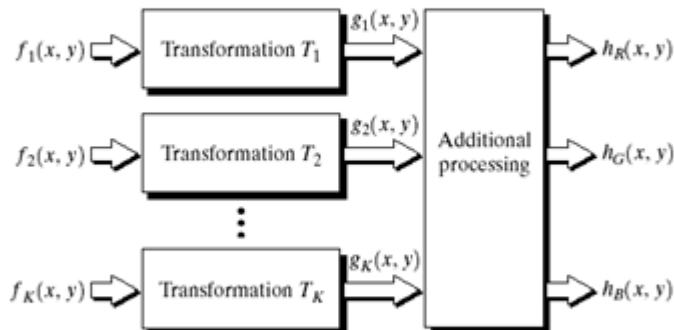


## Gray Level to Color Transformation Example

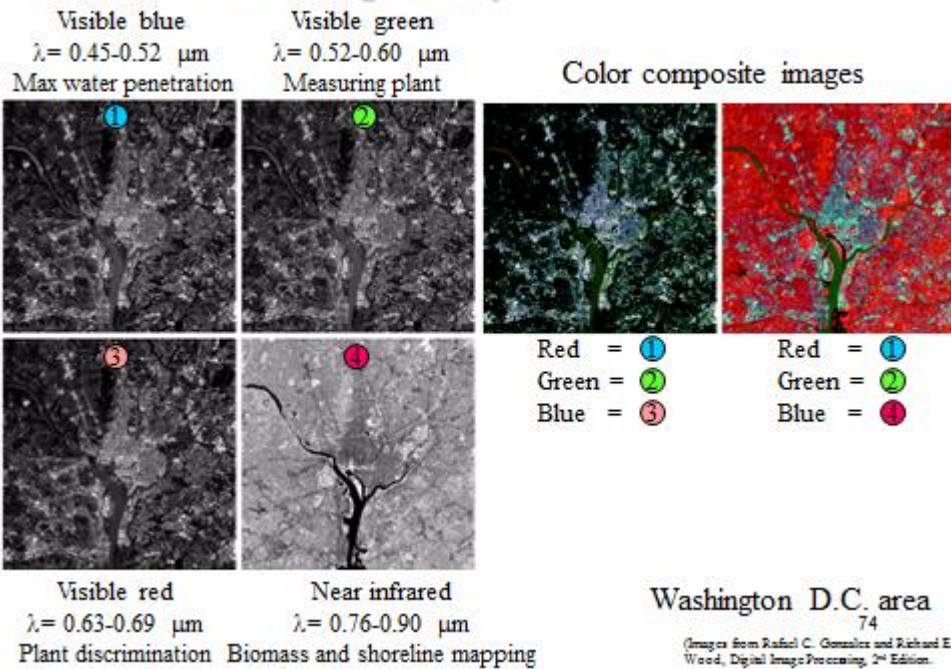


## Pseudocolor Coding

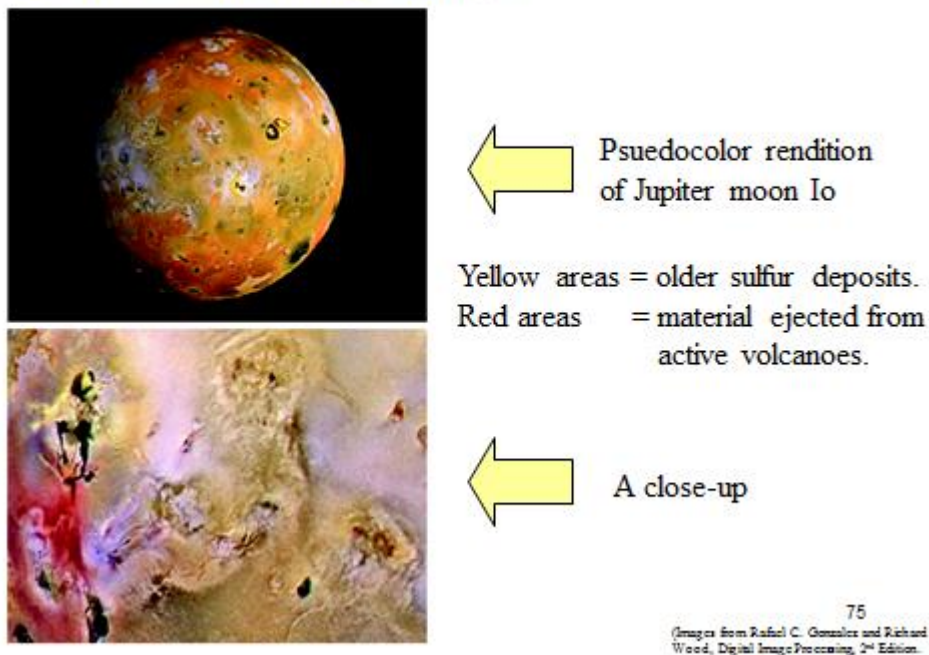
Used in the case where there are many monochrome images such as multispectral satellite images.



## Pseudocolor Coding Example



## Pseudocolor Coding Example



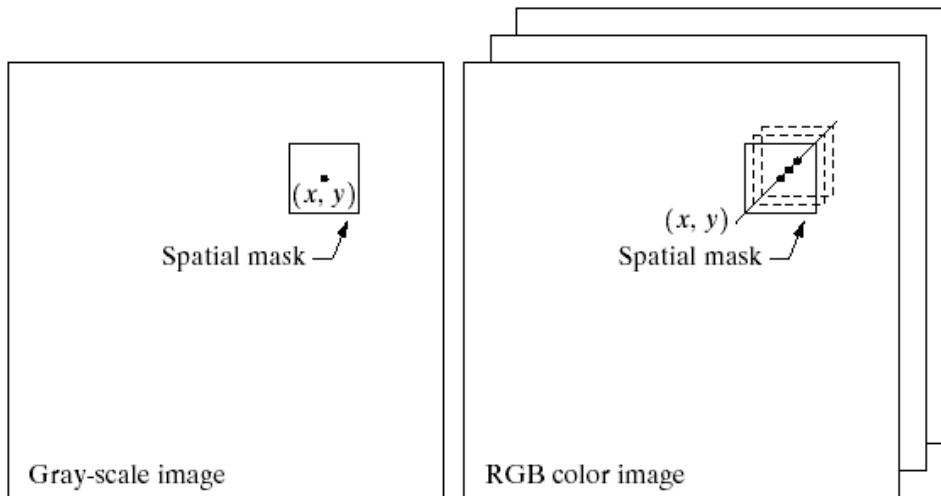
## Basics of Full-Color Image Processing

2 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

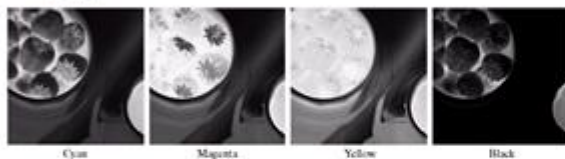
By smoothing each RGB component separately.



**Example: Full-Color Image and Various Color Space Components**



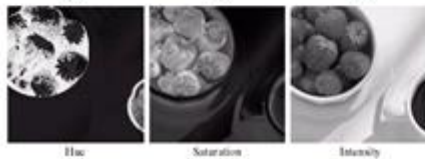
Color image



CMYK components



RGB components



HSI components

77  
(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.)

**Color Transformation**

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:

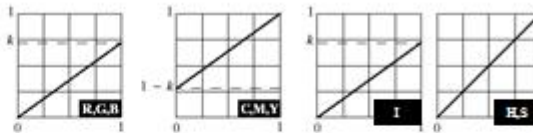
$$s_I(x, y) = kr_I(x, y)$$

Formula for CMY:

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

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

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



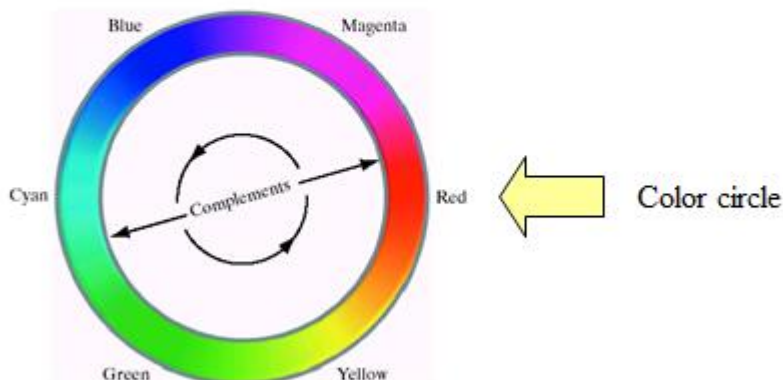
These 3 transformations give the same results.

79

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2nd Edition.

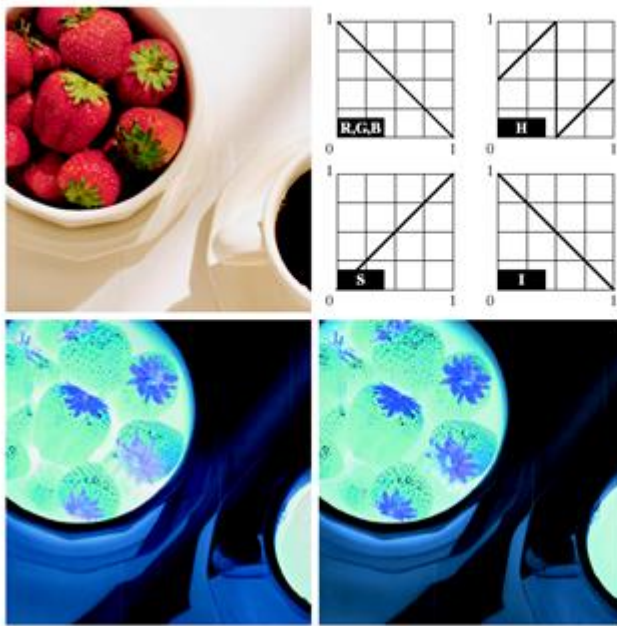
### Color Complements

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.



(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2nd Edition.

## Color Complement Transformation Example



**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.

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.)

## 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.

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

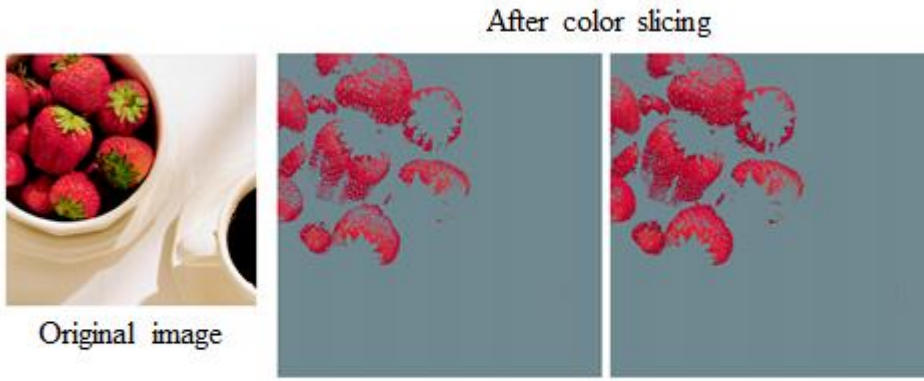
➔ Set to gray  
➔ Keep the original color

or

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

➔ Set to gray  
➔ Keep the original color

## Color Slicing Transformation Example

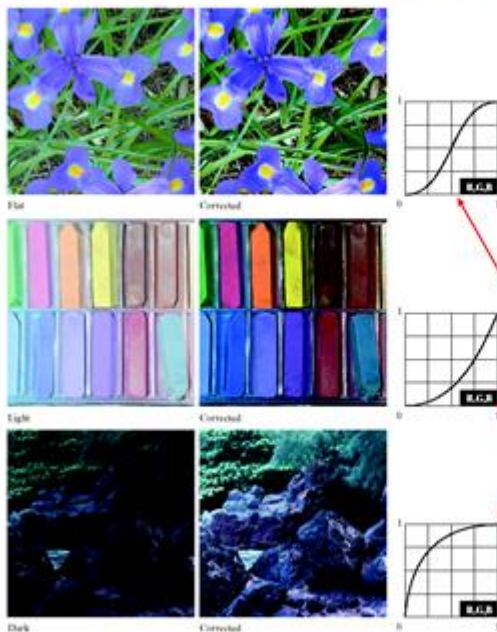


Original image

**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  $(0.5, 0.5, 0.5)$ .

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.)

## Tonal Correction Examples



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

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.)



## Color Balancing Correction Examples

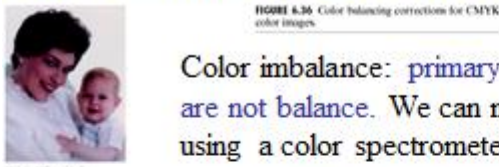
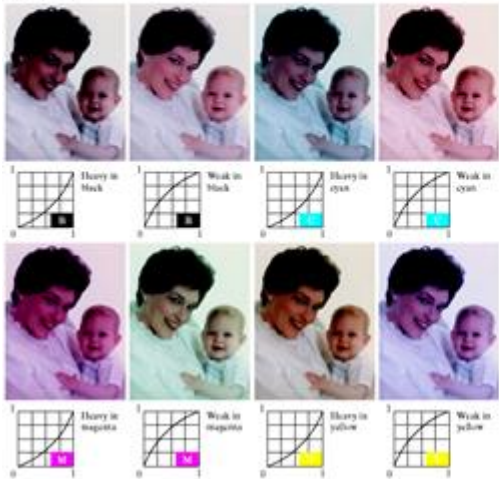


FIGURE 6.36 Color balancing corrections for CMYK color images

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.

(Images from Rafael C. Gonzales and Richard E. Wood, Digital Image Processing, 2<sup>nd</sup> Edition.

### Histogram Equalization of a Full-Color Image

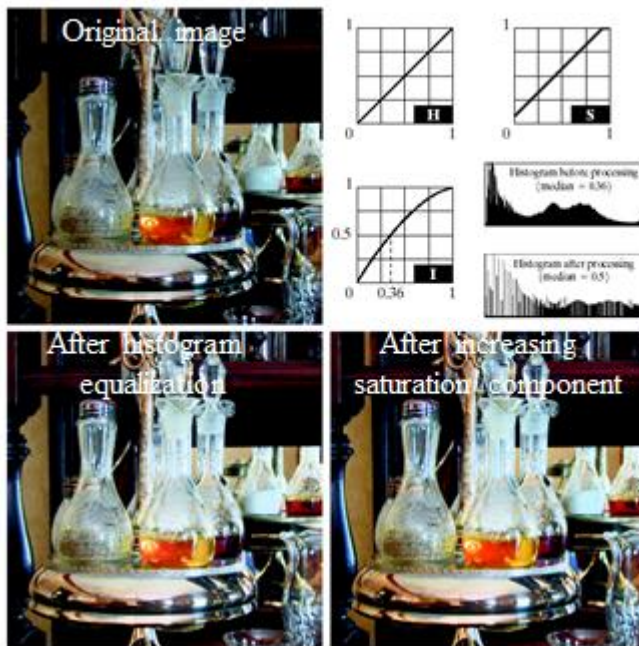
- ❖ Histogram equalization of a color image can be performed by adjusting color intensity uniformly while leaving color unchanged.
- ❖ The HSI model is suitable for histogram equalization where only Intensity (I) component is equalized.

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$

$$= \sum_{j=0}^k \frac{n_j}{N}$$

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

## Histogram Equalization of a Full-Color Image



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

(Images from Rafael C. Gonzales and Richard E. Wood, *Digital Image Processing*, 2<sup>nd</sup> Edition.