Color Image Processing

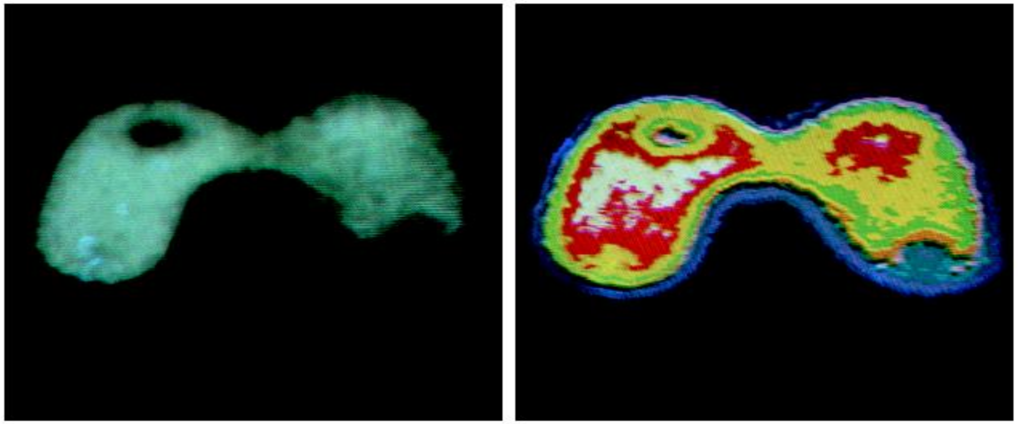
Reference:

Digital Image Processing

Rafael C. Gonzalez, Richard E. Woods

Preview

- Why use color in image processing?
  - Color is a powerful descriptor.
    - Simplifies object identification and extraction.
    - eg. Face detection using skin colors.
  - Humans can discern thousands of color shades and intensities.
    - Can discern only about two dozen shades of grays.



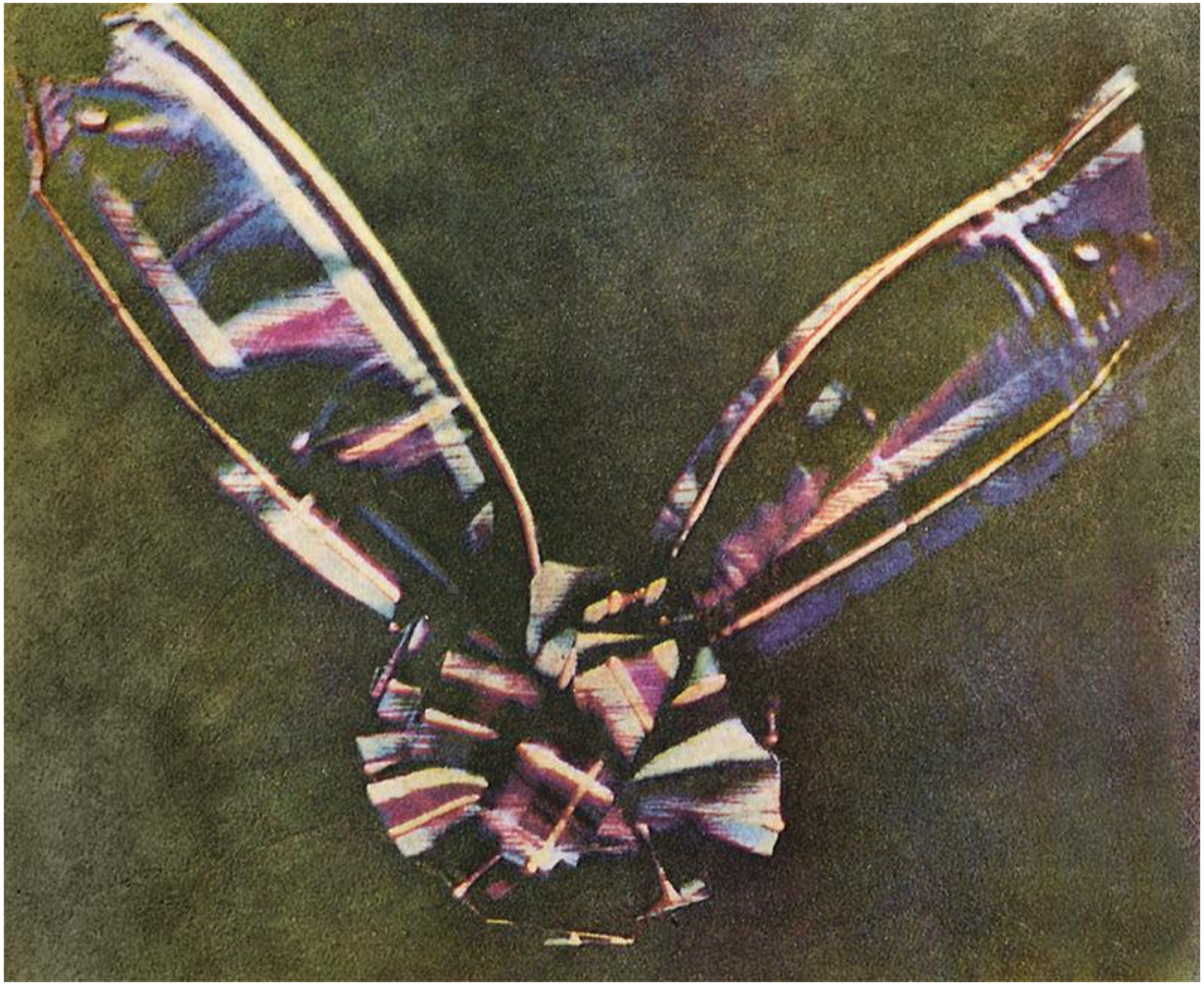
Preview (cont.)

Preview (cont.)

Preview (cont.)

- Two major areas of color image processing:
  - Full color processing
    - Images are acquired from full-color sensor, such as color TV camera or color scanner.
    - Now used in range of applications like publishing, Internet etc.
  - Pseudo-color processing
    - Colors are_assigned to a range of monochrome intensities.
    - Until recently, color sensors and processing hardware were not available.

The first permanent color photograph, taken by J.C. Maxwell in 1861 using three filters, specifically red, green, and violet-blue (ref. Wikipedia)



A photograph of Mohammed Alim Khan (1880–1944), Emir of Bukhara, taken in 1911 by Sergey Prokudin-Gorsky using three exposures with blue, green, and red filters (ref. Wikipedia)

Color fundamentals

The experiment of Sir Isaac Newton in 1666

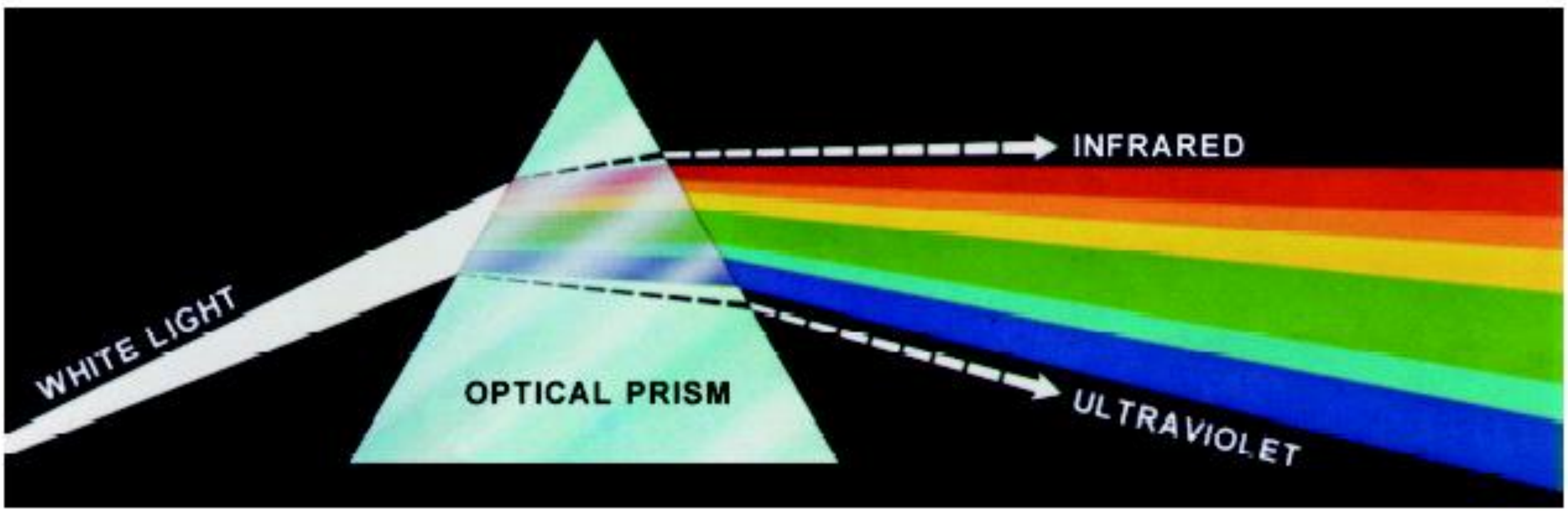


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

Visible light

- Chromatic light span the electromagnetic (EM) spectrum from 400~700 nm.
- Visible light is composed of a narrow band of frequencies.

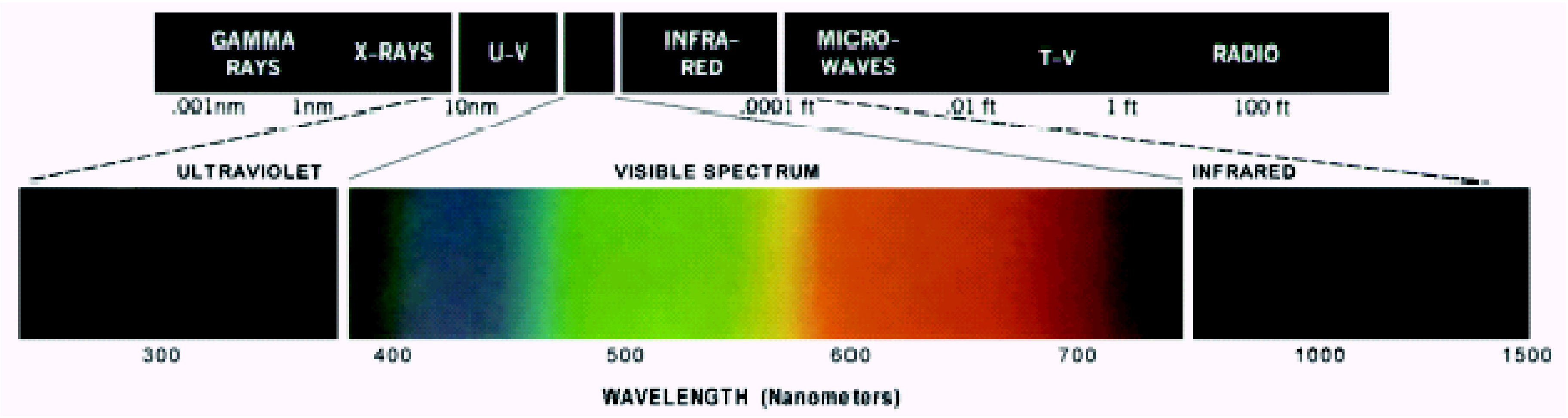
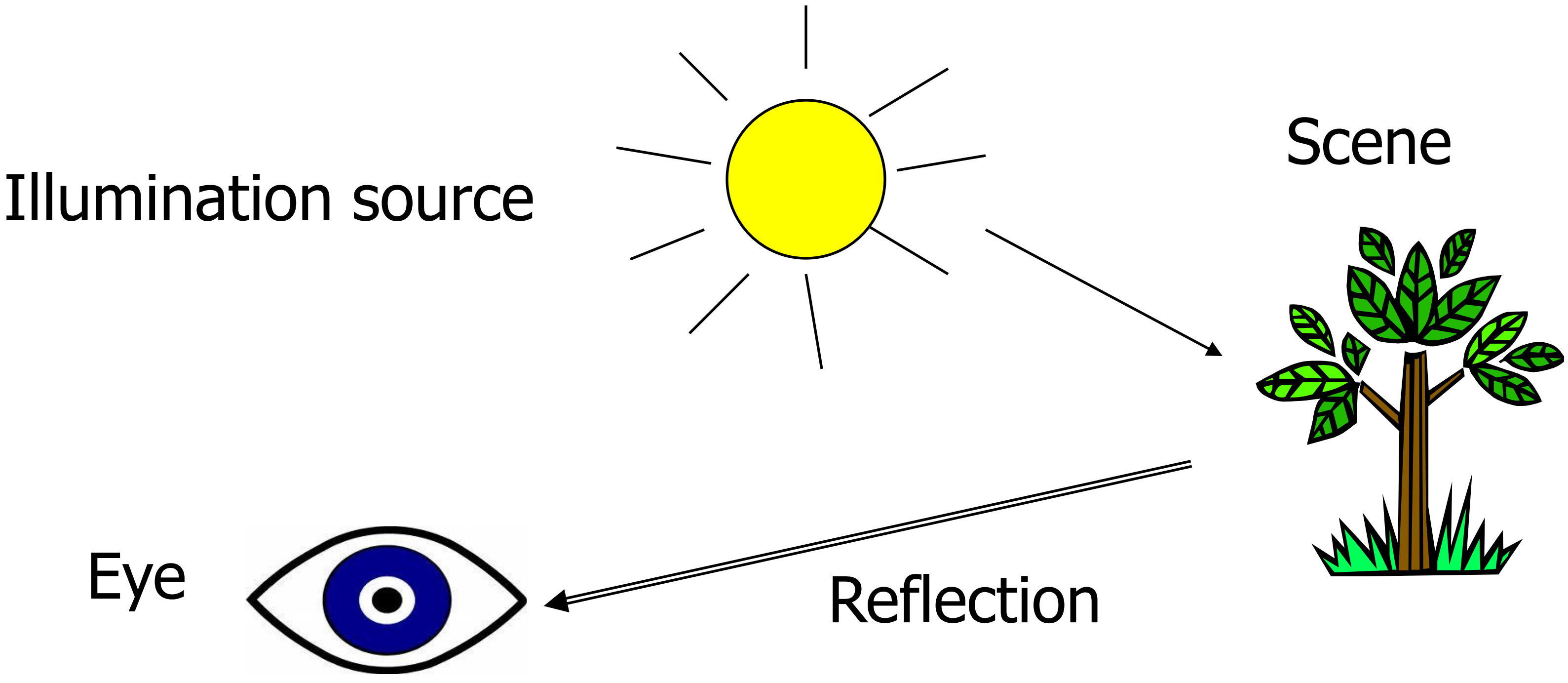


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Color fundamentals (cont.)

- Color that humans perceive in an object = the light reflected from the object.



Quality of chromatic light source

- Radiance: total amount of energy that flow from the light source, measured in watts (W).
- Luminance: amount of energy an observer *perceives* from a light source, measured in lumens (lm).
  - Far infrared light: high radiance, but zero luminance.
- Brightness: subjective descriptor that is hard to measure, similar to the achromatic notion of intensity, practically impossible to measure.

How human eyes sense light?

- 6~7 m cones are the sensors in the eye.
  - 3 principal sensing categories:
    - Red light (65%)
    - Green light (33%)
    - Blue light (2%): most sensitive.
  - CIE (Commission Internationale de l'Eclairage – International Commission on Illumination) designated in 1931:
    - Red: 700 nm
    - Green: 546.1 nm
    - Blue: 435.8 nm

Absorption of light by cones

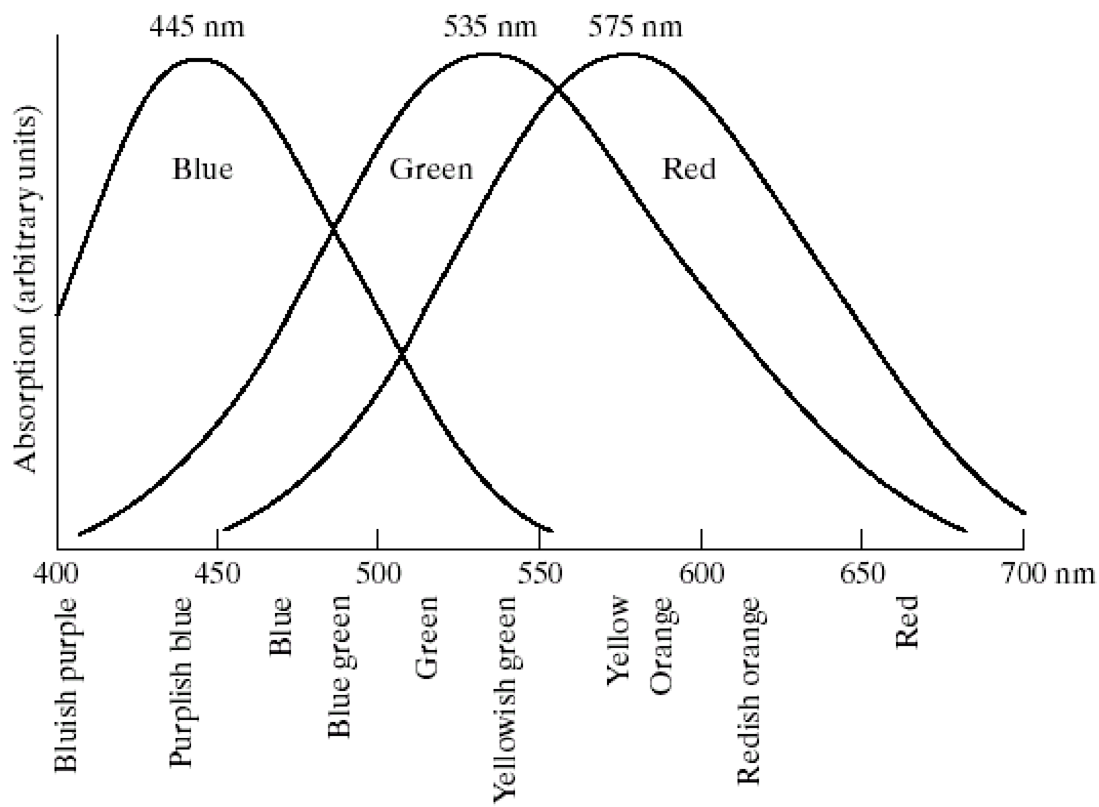


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

Primary and secondary colors

- No single color may be called red, green, blue.
- Primary colors can be added to produce secondary colors of light:
  - G+B=Cyan
  - R+B=Magenta
  - R+G=Yellow
- Mixing the three primaries or secondary with opposite primary color gives white light.

Primary colors of light v.s. primary colors of pigments

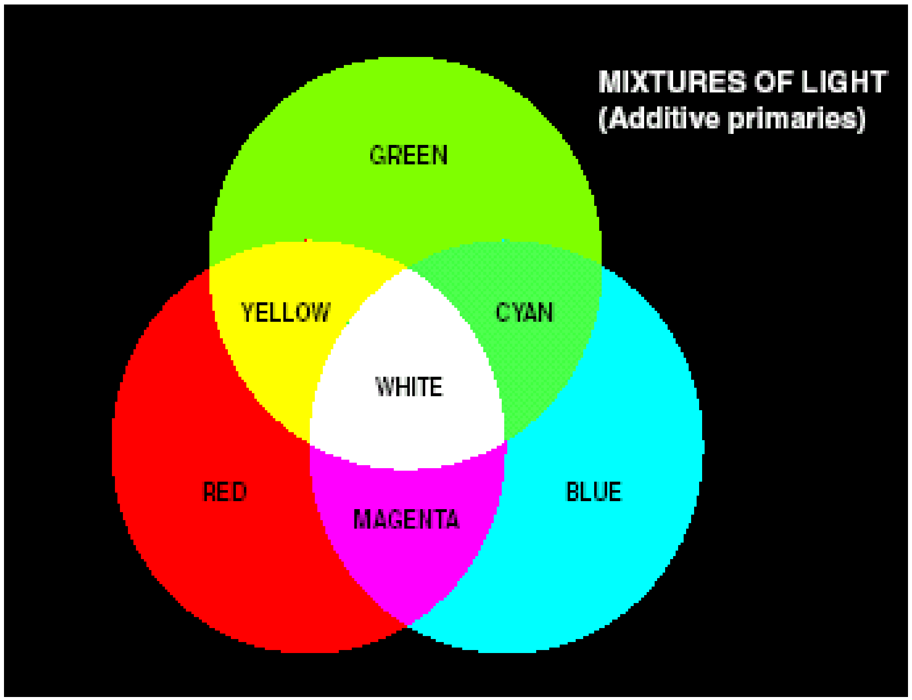
- Primary color of pigments:
  - Color that subtracts or absorbs a primary color of light and reflects or transmits the other two.

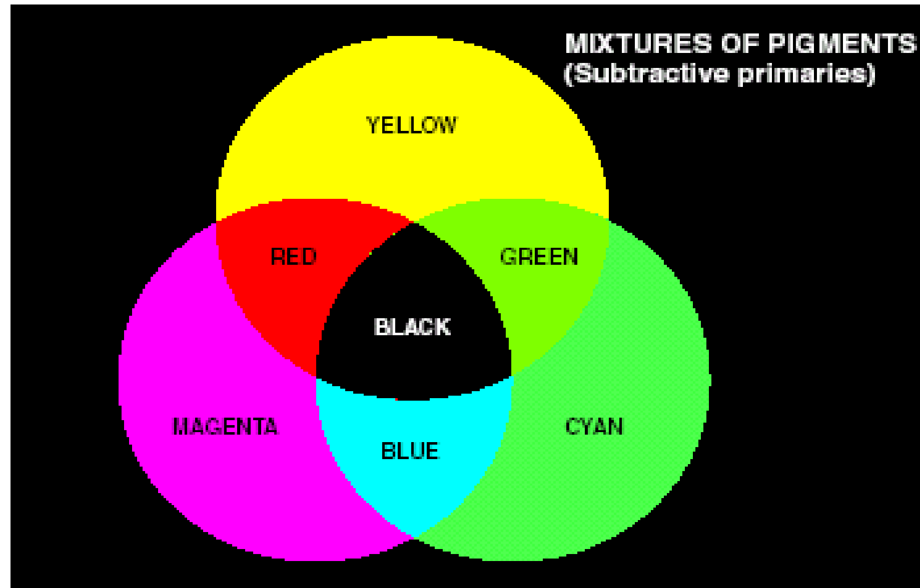
Color of light:	R	G	B
	↓	↓	↓
Color of pigments:	absorb R	absorb G	absorb B
	Cyan	Magenta	Yellow

  - Primary colors of pigments are magenta, cyan and yellow (secondary colors are red, green, blue).

Primary colors of pigments

- Primary colors of pigments are magenta, cyan and yellow (secondary colors are red, green, blue).
- Proper combination of pigment primaries or secondary with its opposite primary, produces black.

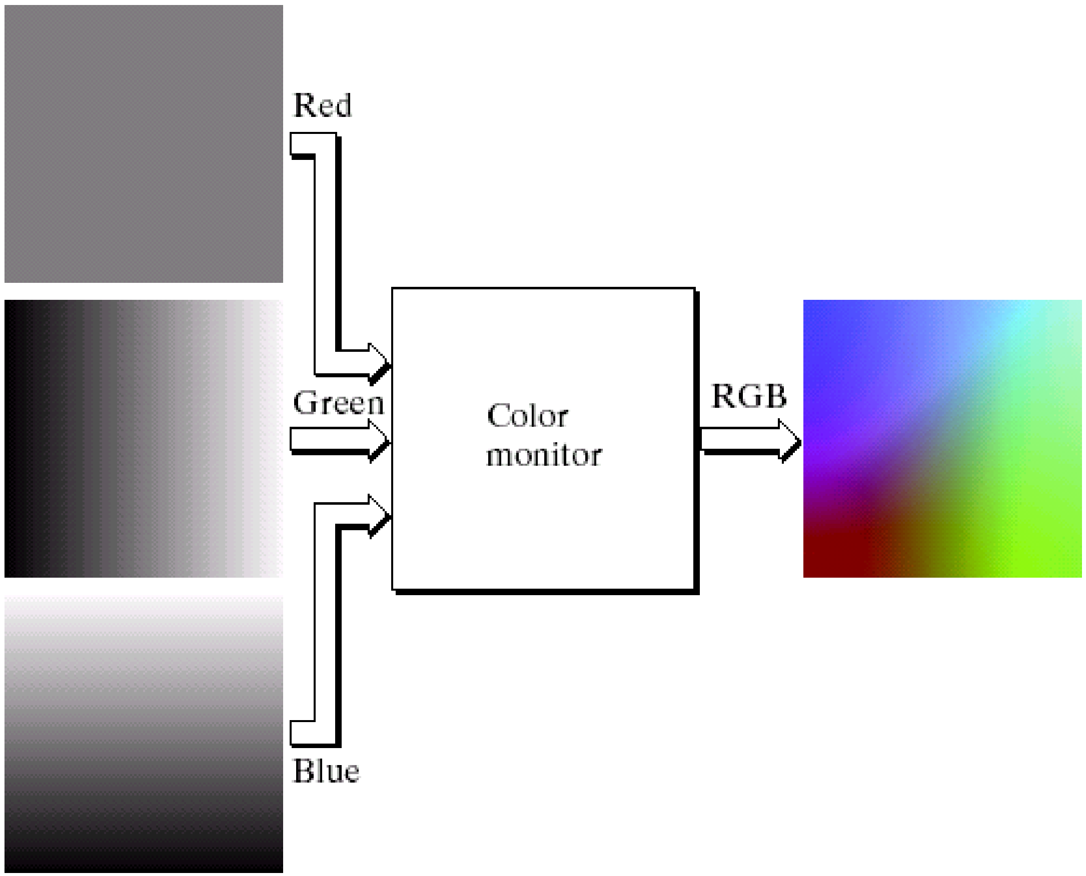




PRIMARY AND SECONDARY COLORS
OF LIGHT AND PIGMENT

Application of additive nature of light colors

- Color TV



Color characteristics

- Characteristics generally used to distinguish one color from another are:
  - Hue: Represents dominant color as perceived by an observer.
    - An object is red, yellow or orange.
  - Saturation: Relative purity or amount of white light mixed with a hue.
    - Pure spectrum colors are fully saturated.
    - Pink (red and white), lavender (violet and white) are less saturated.
    - Degree of saturation is inversely proportional to amount of white light added.
  - Brightness: Chromatic notation of intensity.

Color characteristics

- Hue and saturation are together called *Chromaticity.*
- A color may be characterized by its chromaticity and brightness.
- Amount of red, green and blue needed to form a particular color are called tristimulus values.
  - Denoted by X, Y, Z.

CIE XYZ model

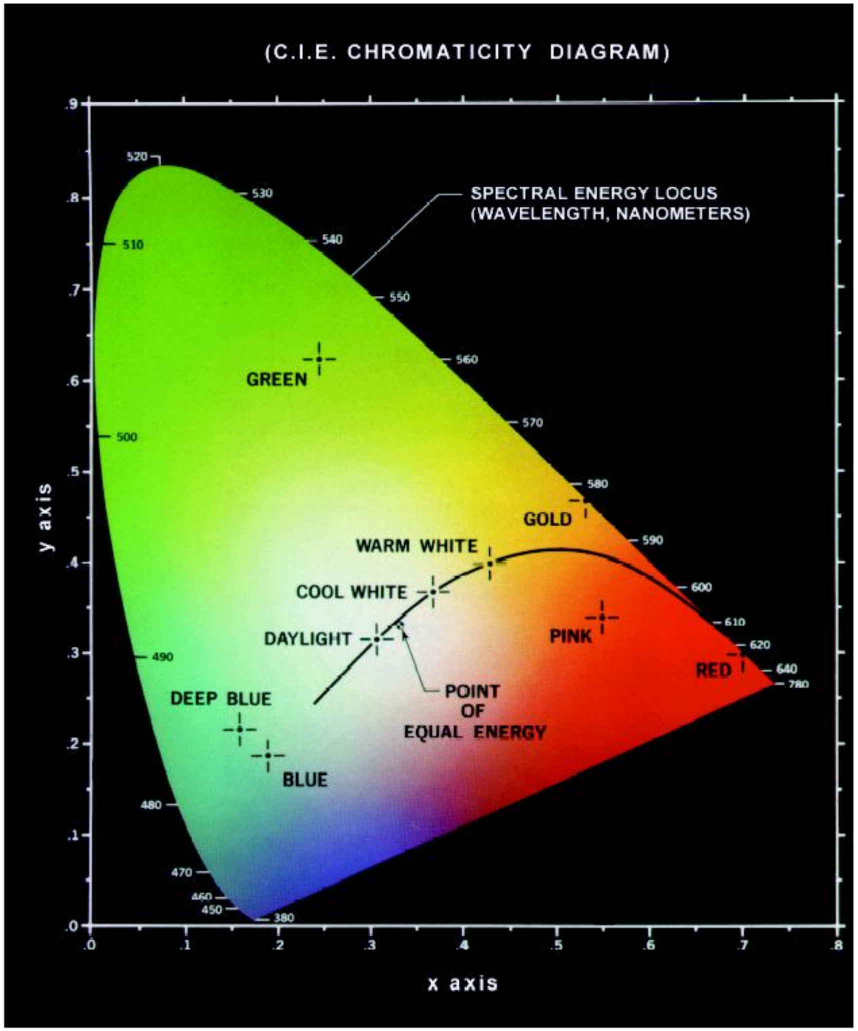
- A color is specified by its trichromatic coefficients:

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = \frac{Z}{X+Y+Z}$$

  - $x + y + z = 1$

Chromaticity diagram

FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)



- GREEN:

  ~62% green

  ~25% red

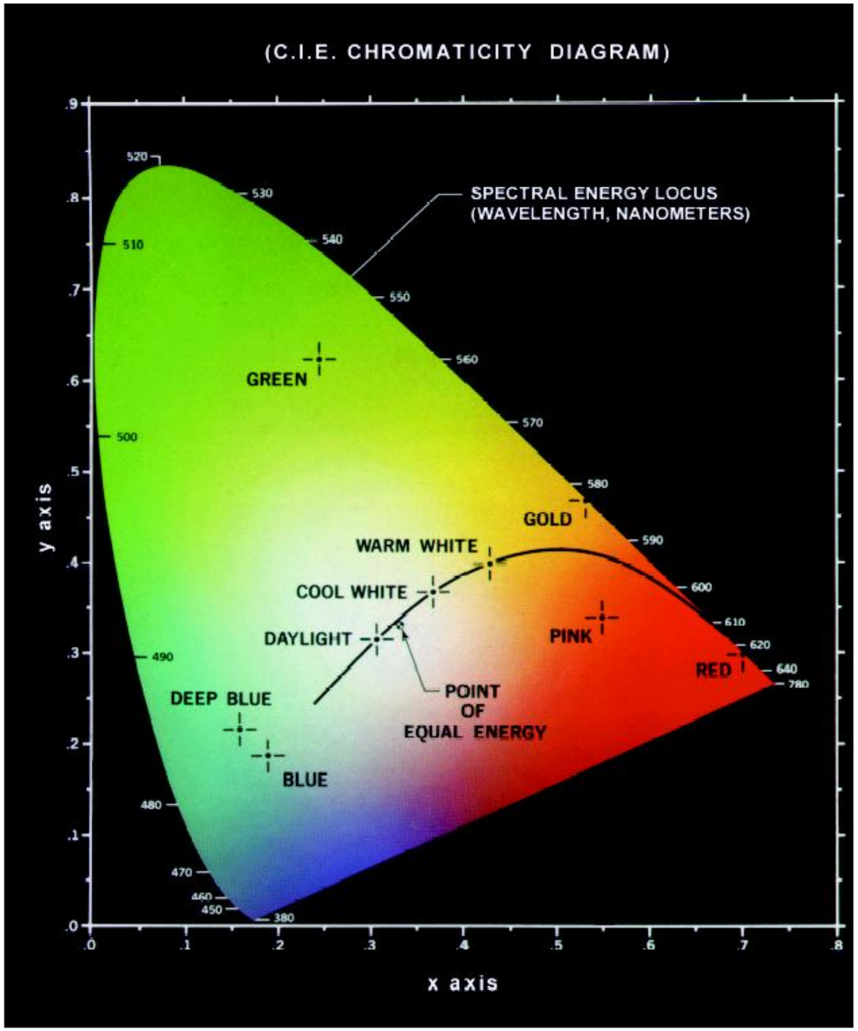
$z=1-(x+y)$

Thus, blue~13%

Chromaticity diagram (cont.)

- Position of spectrum colors is on boundary.
- Points within diagram are mixture of spectrum colors.
- Point of equal energy correspond to equal fractions of primary colors (CIE standard for white light).

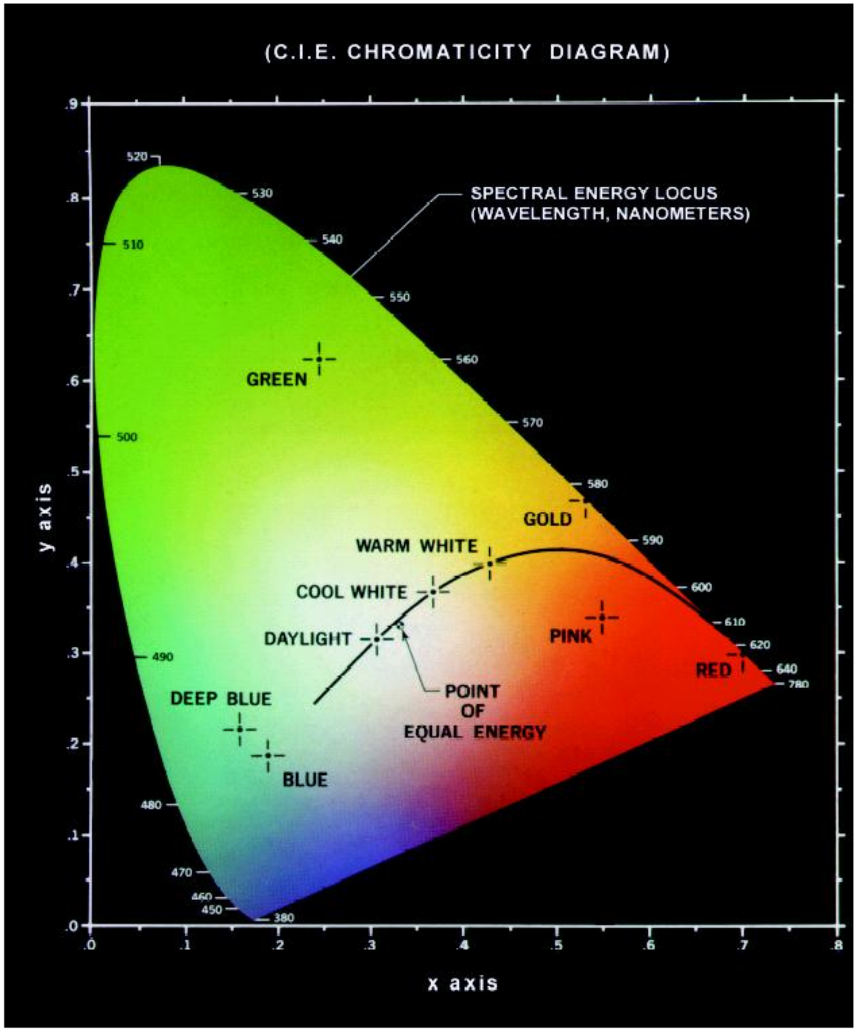
FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)



Chromaticity diagram (cont.)

- Point located on boundary is fully saturated.
- Saturation at point of equal energy is zero.
- A line segment joining any two points shows different color variations obtained by adding these two colors.

FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)



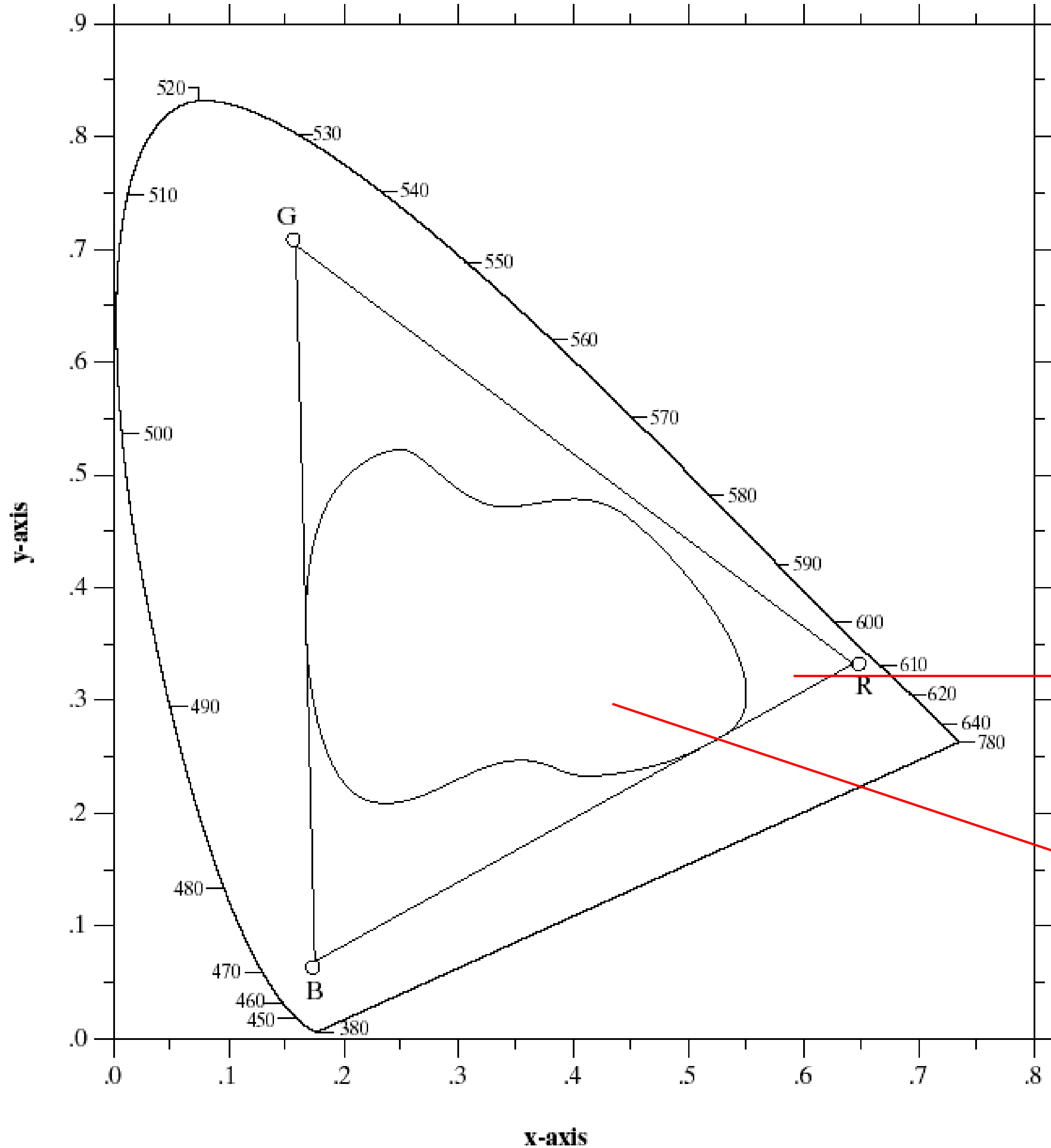


FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

By additivity of colors: Any color inside the triangle can be produced by combinations of the three initial colors.

Color gamut of RGB monitors

Color gamut of printing devics

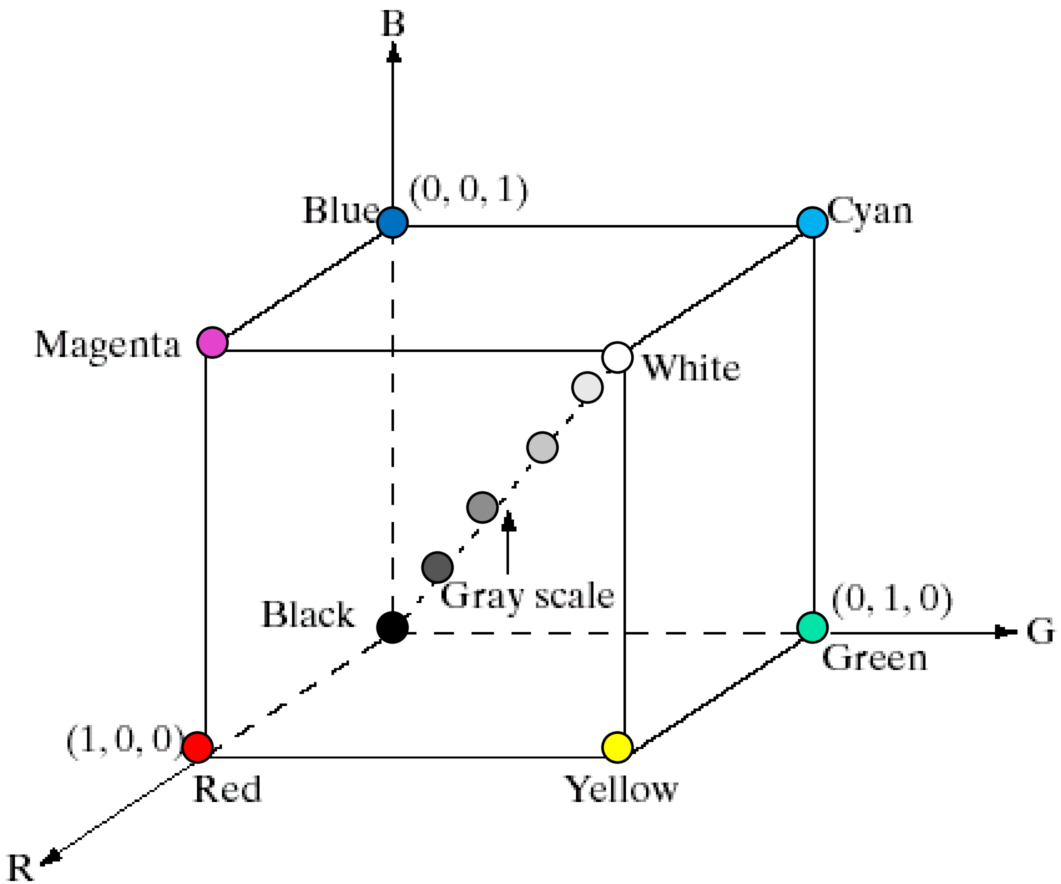
Color models

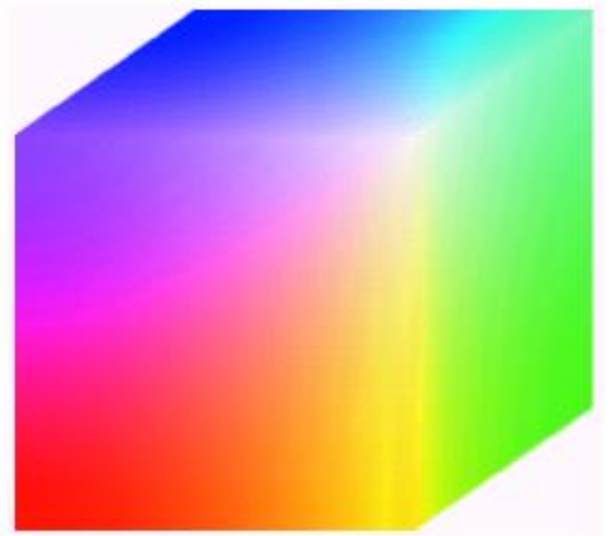
- Color model, color space, color system:
  - Specify colors in a standard way.
  - A coordinate system where each color is represented by a single point.
- RGB model: color monitors/video cameras.
- CMY model: color printing.
- CMYK model: color printing.
- HSI model: match human interpretation, decouples color and gray scale information in an image.

RGB color model

- Based on a Cartesian co-ordinate system.
  - Each color is composed of its primary spectral colors – red, green, blue.
  - Color subspace is represented by a cube.
  - RGB values are at three corners; cyan, magenta and yellow are at other three corners.
  - Black is at origin; white is farthest from the origin.
  - Gray scale (points of equal RGB values) extends from black to white along line joining them.

RGB color model





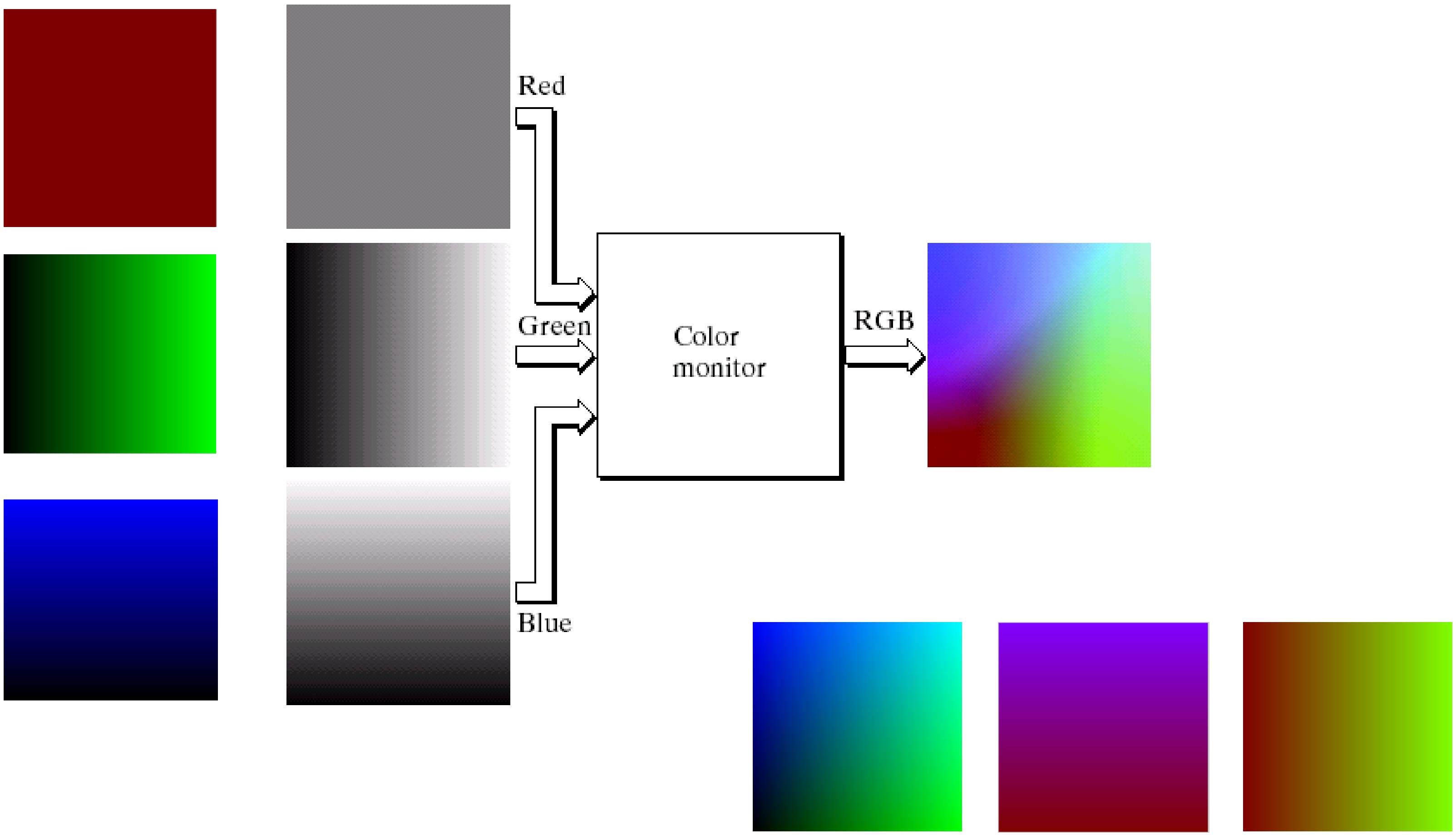
Some basic colors

Colour	Hex Code #RRGGBB	Decimal Code (R,G,B)
Black	#000000	(0,0,0)
White	#FFFFFF	(255,255,255)
Red	#FFFFFF	(255,0,0)
Lime (Saturated green)	#00FF00	(0,255,0)
Green	#008000	(0,128,0)
Blue	#0000FF	(0,0,255)
Yellow	#FFFF00	(255,255,0)
Cyan	#00FFFF	(0,255,255)
Magenta	#FF00FF	(255,0,255)
Silver	#C0C0C0	(192,192,192)
Gray	#808080	(128,128,128)
Navy	#000080	(0,0,128)
Purple	#800080	(128,0,128)

Some shades of red

Colour	Hex Code #RRGGBB	Decimal Code (R,G,B)
Maroon	#800000	(128,0,0)
Dark red	#8B0000	(139,0,0)
Brown	#A52A2A	(165,42,42)
Firebrick	#B22222	(178,34,34)
Crimson	#DC143C	(220,20,60)
Red	#FF0000	(255,0,0)
Tomato	#FF6347	(255,99,71)
Beige	#F5F5DC	(245,245,220)
Plum	#DDA0DD	(221,160,221)
Orange	#FFA500	(255,165,0)
Gold	#FFD700	(255,215,0)
Khakhi	#F0E68C	(240,230,140)
Pink	#FFC0CB	(255,192,203)

RGB color model (cont.)



(R=0) (G=0) (B=0)

Pixel depth

- Pixel depth: the number of bits used to represent each pixel in RGB space.
- Full-color image: 24-bit RGB color image
  - (R, G, B) triplet = (8 bits, 8 bits, 8 bits).
- Total no. of colors in a 24-bit image =

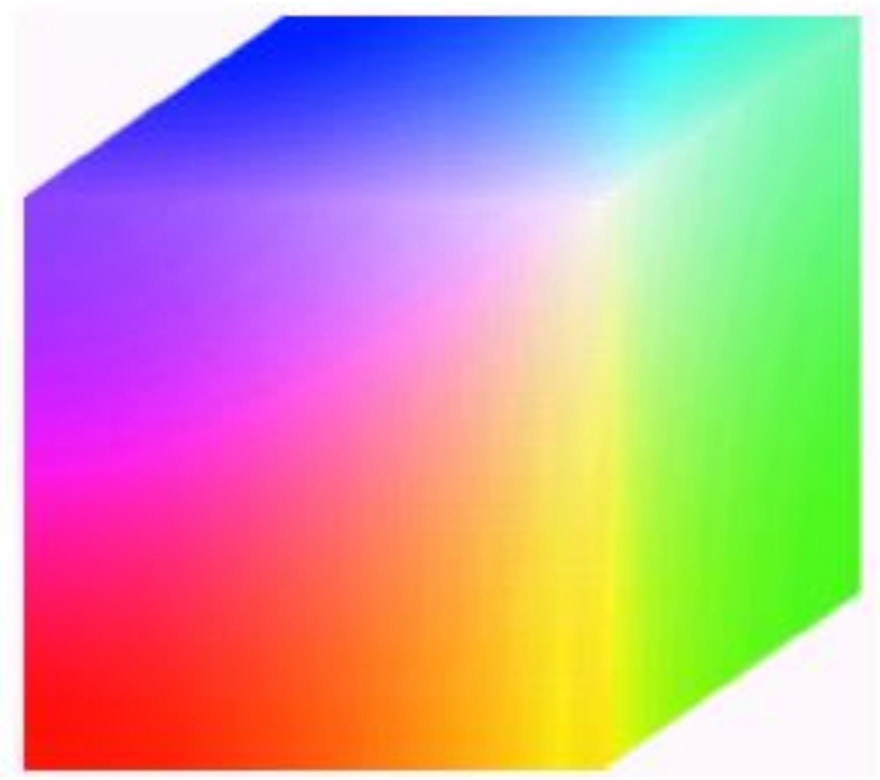
  $(2^8)^3 = 16,777,216$

Safe RGB colors

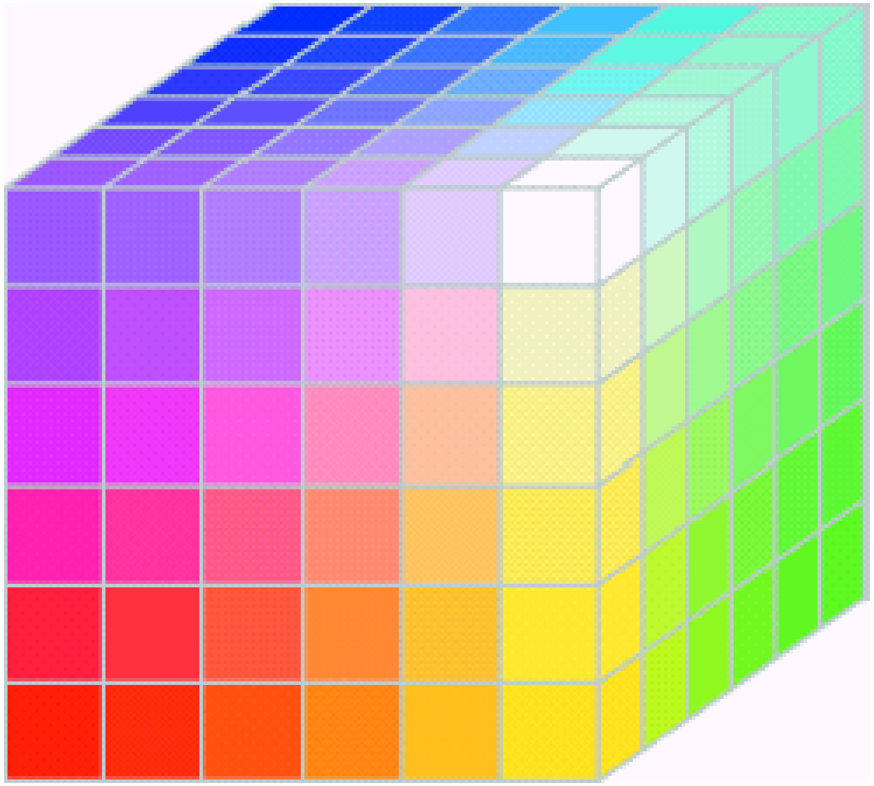
- Subset of colors is enough for some applications.
  - Independent of viewer h/w capabilities.
- Safe RGB colors (safe Web colors, safe browser colors).
  - Assuming 256 colors likely to be reproduced.
  - Forty of these 256 known to be processed differently by different OS.
  - 216 colors are safe colors.

$$(6)^3 = 216$$

Safe RGB color (cont.)



Full color cube



Safe color cube

Safe RGB colors

- Each of the 216 safe colors is formed from three RGB values:
  - Each value can only be 0, 51, 102, 153, 204, 255. (In Hex: 00, 33, 66, 99, CC, FF)
  - Ex. Purest red is FF0000
  - 000000 represents black; FFFFFF represents white.
  - Unlike the full-color cube, which is solid, the RGB safe color cube has valid colors only on the surface planes (each plane has 36 colors).

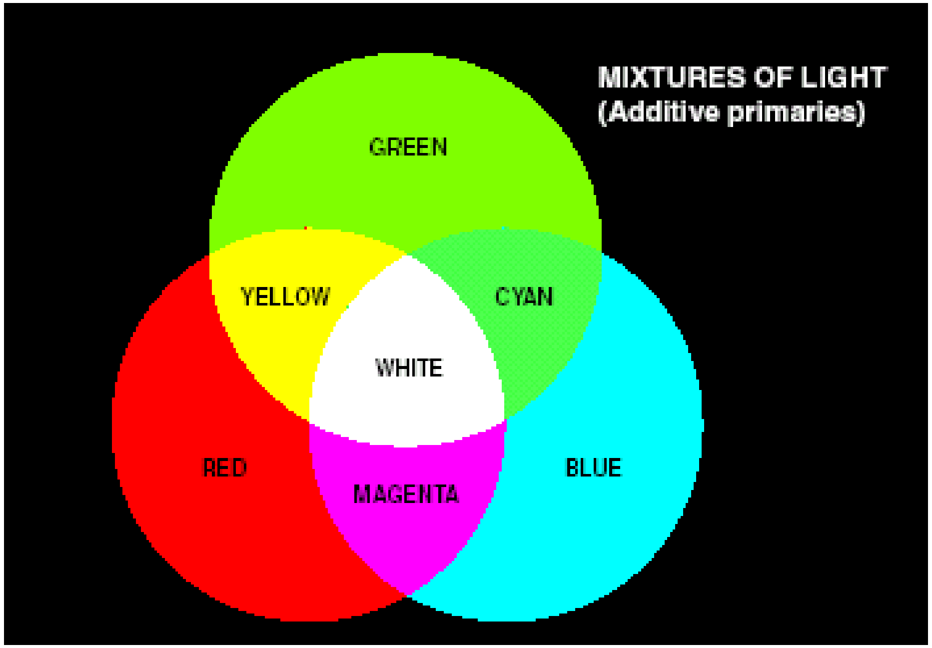
CMY model

- CMY: secondary colors of light, or primary colors of pigments.
- When a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface.
- RGB values can be obtained easily from CMY values by subtracting individual CMY values from 1.

CMY model

- Used by most devices that deposit colored pigments on paper (color printers, copiers).

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



CMY model (+Black = CMYK)

- Ideally, equal amounts of cyan, magenta and yellow should produce black.
  - In practice, these colors for printing produce a muddy black.
  - To produce true black, the predominant color in printing, a fourth color, *black*, is added.
  - This is the CMYK model.

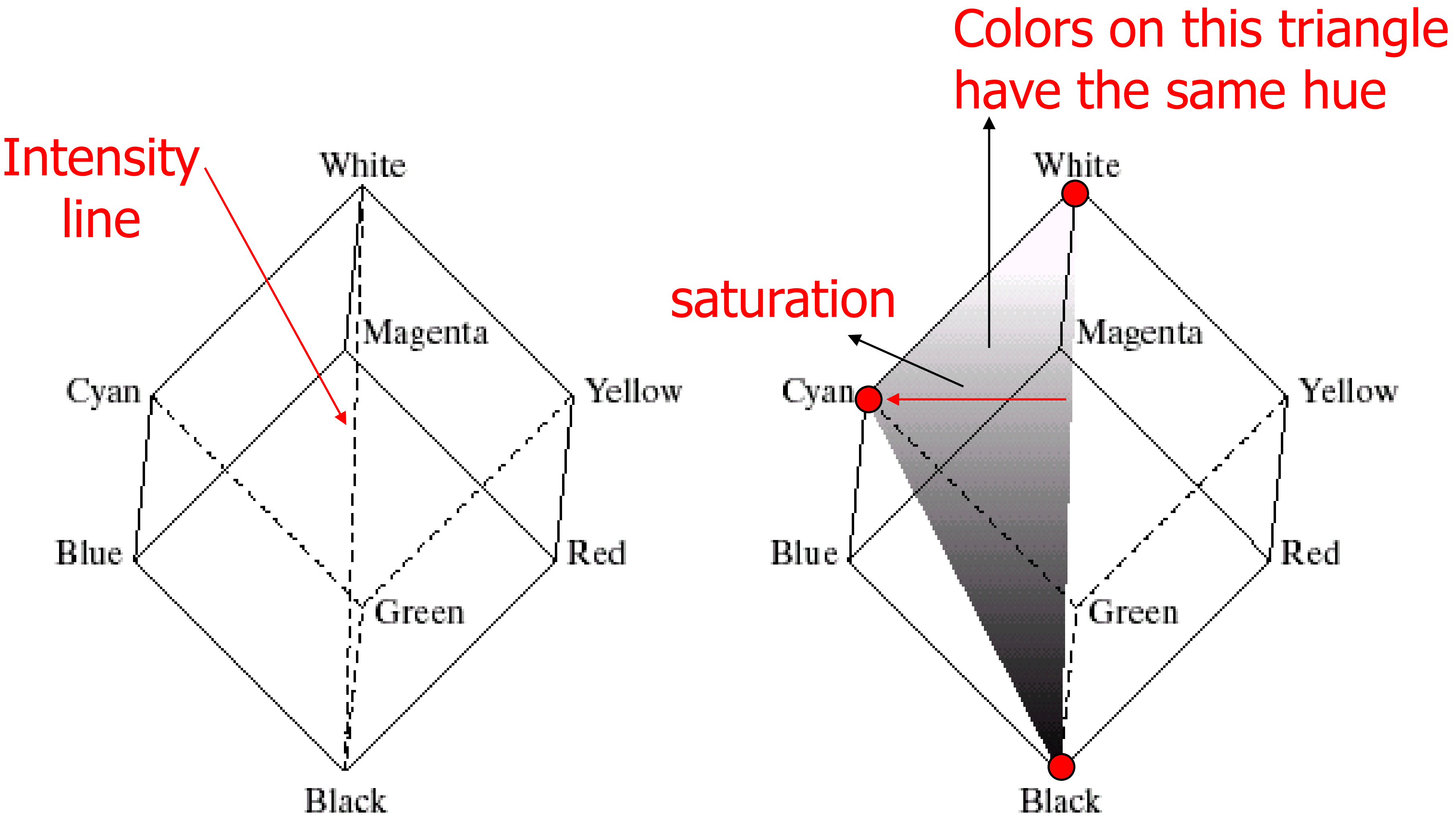
HSI color model

- Will you describe a color using its R, G, B components?
- Human *describe* a color by its hue, saturation, and brightness.
  - Hue: color attribute
  - Saturation: purity of color
  - Brightness: achromatic notion of intensity
- HSI model decouples intensity from color carrying information (hue and saturation).

HSI color model

- RGB image consists of three monochrome intensity images.
  - Intensity can be extracted from an RGB image
- Stand the RGB cube on its black (0,0,0) vertex with the white (1,1,1) vertex directly above it.
  - The intensity line is the line joining these two vertices.

RGB - HSI


Intensity line
White
Magenta
Cyan
Yellow
Blue
Red
Green
Black
Colors on this triangle have the same hue
saturation
White
Magenta
Cyan
Yellow
Blue
Red
Green
Black

HSI color model

- To determine the intensity component of any color point, pass a plane *perpendicular* to the intensity axis and containing the color point.
  - The intersection of plane with intensity axis will give a point with intensity value in the range [0, 1].
- Saturation (purity) of a color increases as function of distance from intensity axis.
  - Saturation of points on intensity axis is zero.

HSI color model

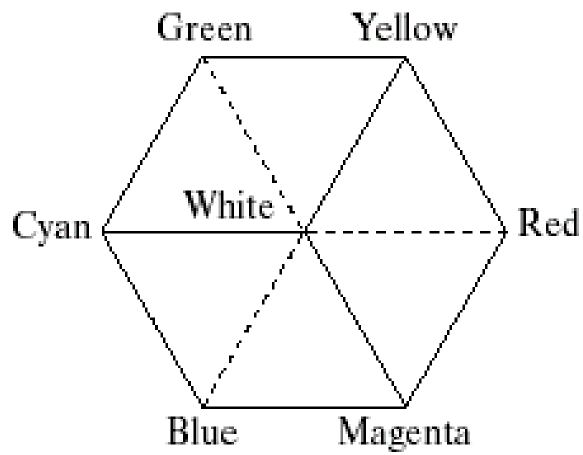
- To determine hue from a given RGB point:
  - Consider a plane defined by three points (black, white, cyan)
  - Black and white points are contained in the plane i.e. intensity axis is also contained in the plane.
  - All points contained in the plane segment defined by intensity axis and boundaries of cube have the same hue (here, cyan)
  - Recall, all colors generated by three colors lie in the triangle defined by those colors.
  - If two points are black and white and third is a color point, all points in triangle would have same hue (black and white do not change hue)

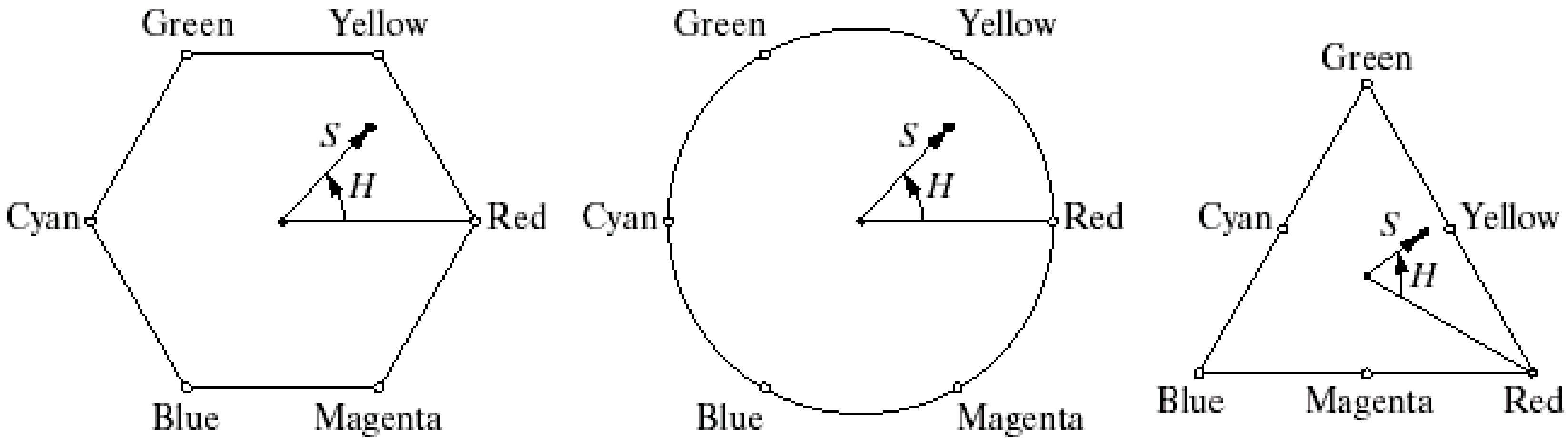
HSI color model

- However, intensity and saturation of points in this triangle would be different.
- By rotating shaded plane about vertical axis, different hues can be obtained.

- HSI space is represented by a vertical intensity axis and the locus of color points that lie on planes perpendicular to this axis.
  - The primary colors are separated by 120 degrees.
  - Secondary colors are 60 degrees from the primaries

HSI model: hue and saturation

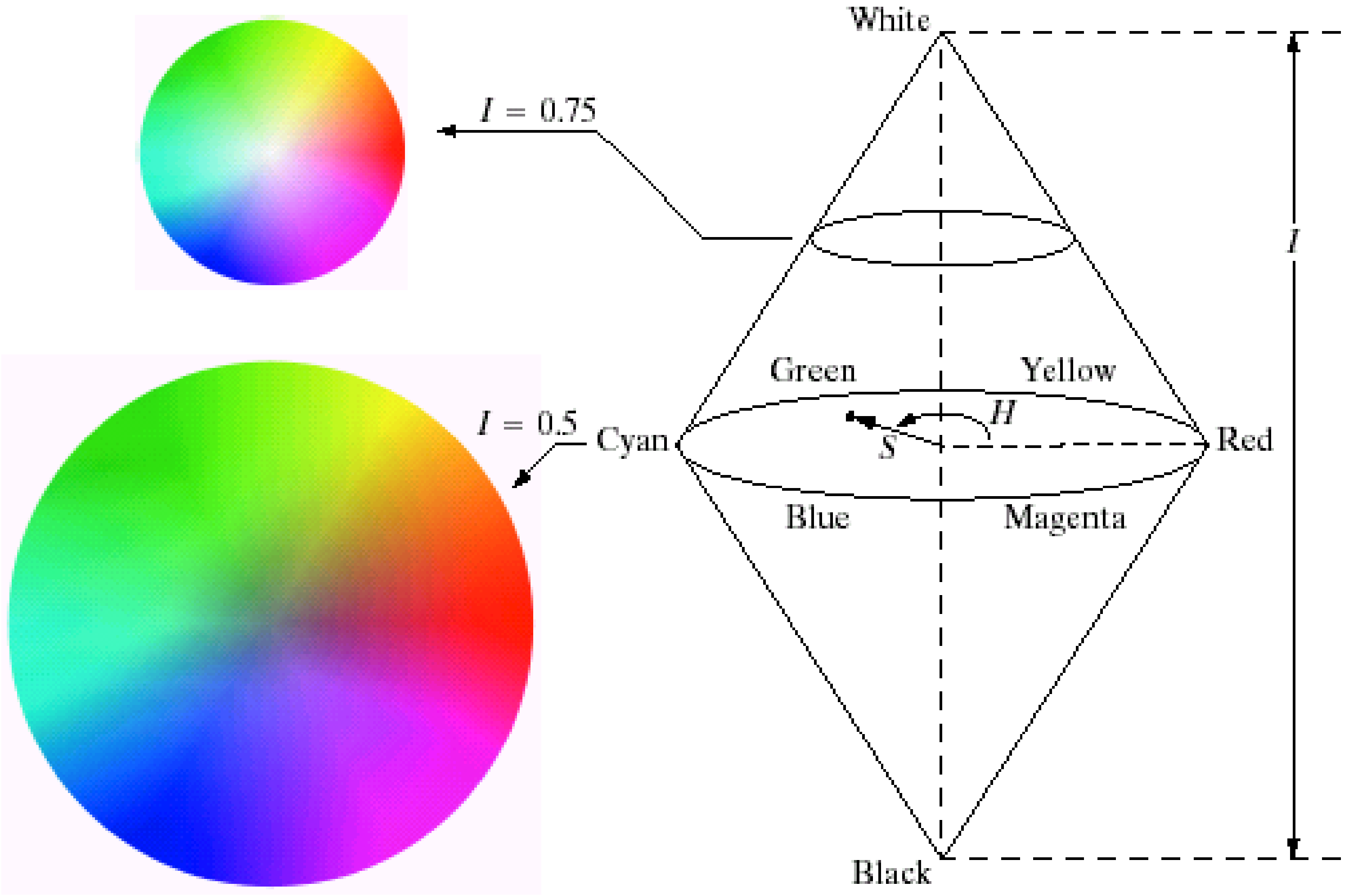




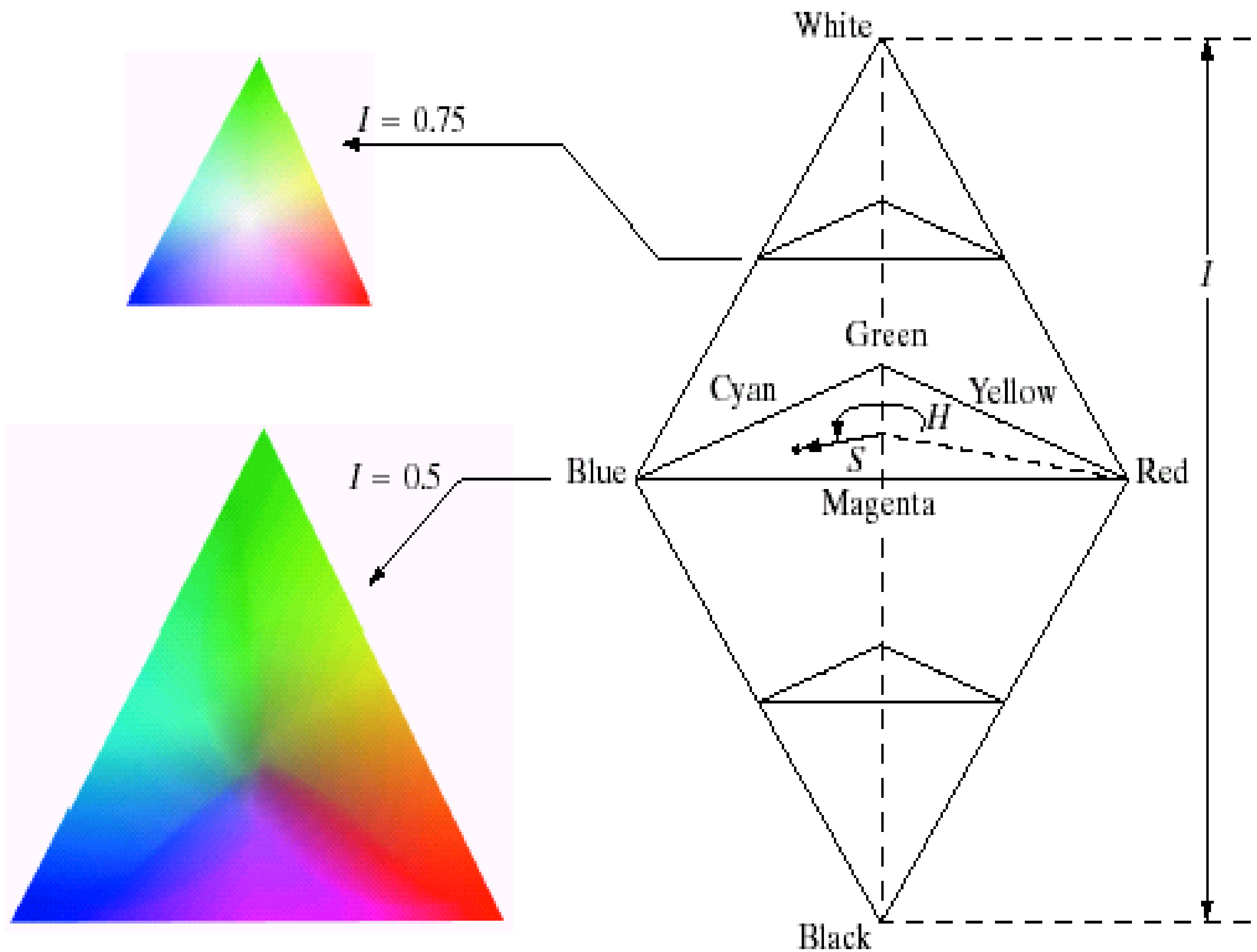
HSI color model

- The hue is determined from some reference point.
  - Usually, an angle of 0 deg from red axis denotes 0 hue.
  - Hue increases counterclockwise from there.
  - Saturation (distance from vertical axis) is length of vector from origin to the point.
  - Origin is defined by intersection of color plane from vertical intensity axis.
  - Intensity is the distance along this vertical axis.
  - HSI planes can be described by hexagon, a triangle or circle

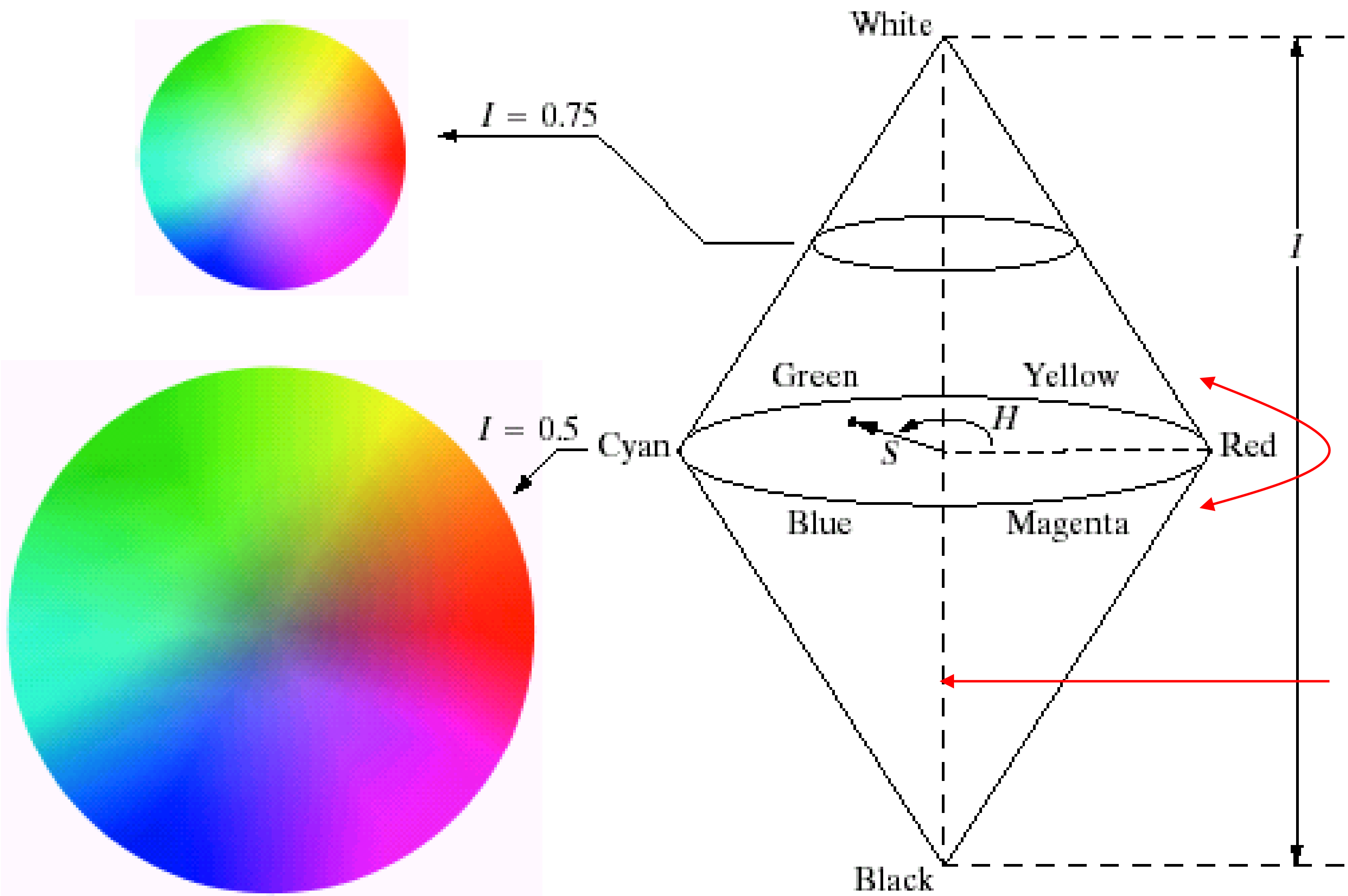
HSI model



HSI color model



Problem of using Hue component



dis-continuous

Un-defined over gray axis

RGB to HSI

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

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

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

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

HSI to RGB

- RG sector ($0 <= H < 120°$)

$$R = I\left[1 + \frac{S\cos H}{\cos(60 - H)}\right] \quad G = 3I - (R + B) \quad B = I(1 - S)$$

- GB sector ($120° <= H < 240°$)

$$R = I(1 - S) \quad G = I\left[1 + \frac{S\cos(H - 120)}{\cos(H - 60)}\right] \quad B = 3I - (R + G)$$

HSI to RGB

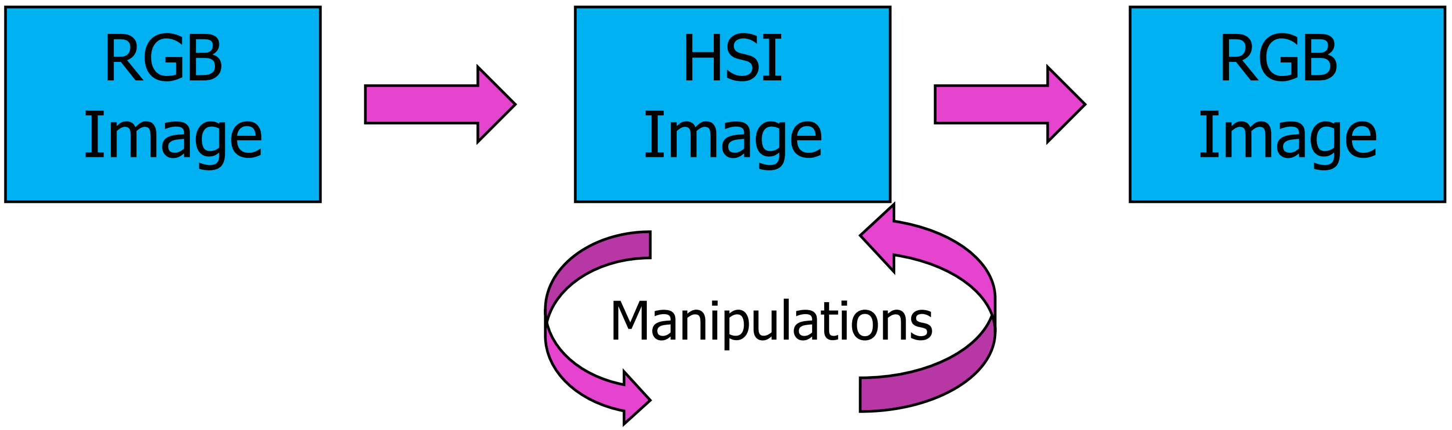
- BR sector ($240^\circ <= H <= 360^\circ$)

$$R = 3I - (G + B) \qquad G = I(1 - S) \qquad B = I\left[1 + \frac{S\cos(H - 240)}{\cos(H - 180)}\right]$$

Manipulating HSI images

In order to manipulate an image under the HSI model we:

- First convert it from RGB to HSI
- Perform our manipulations under HSI
- Finally convert the image back from HSI to RGB

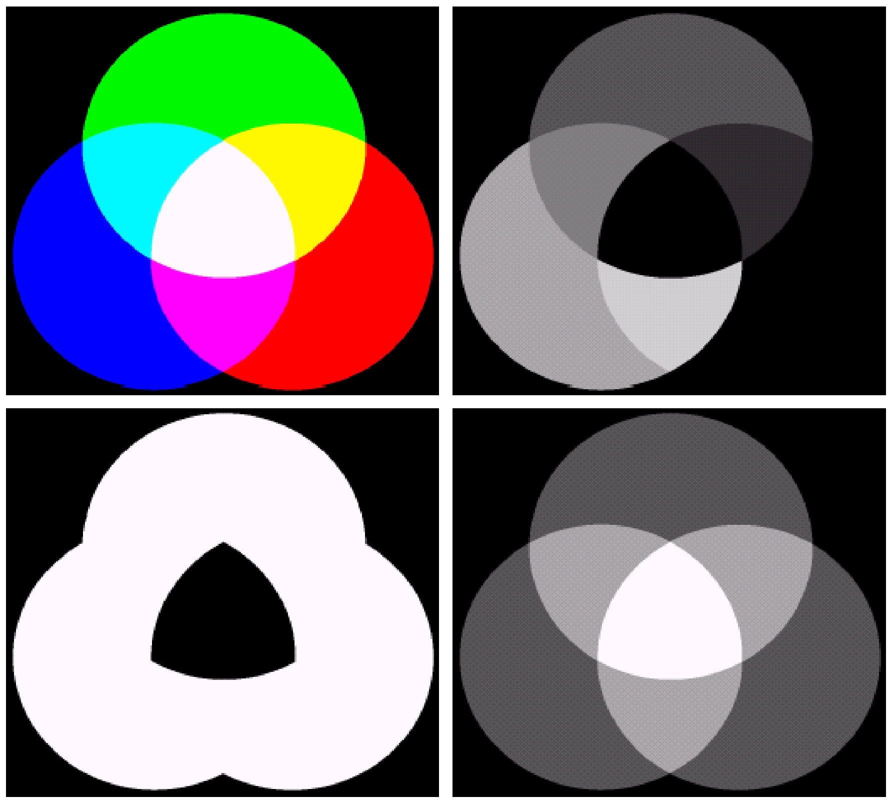


HSI component images

- Consider an image with composite RGB colors.
- H, S and I components can be extracted.

R,G,B

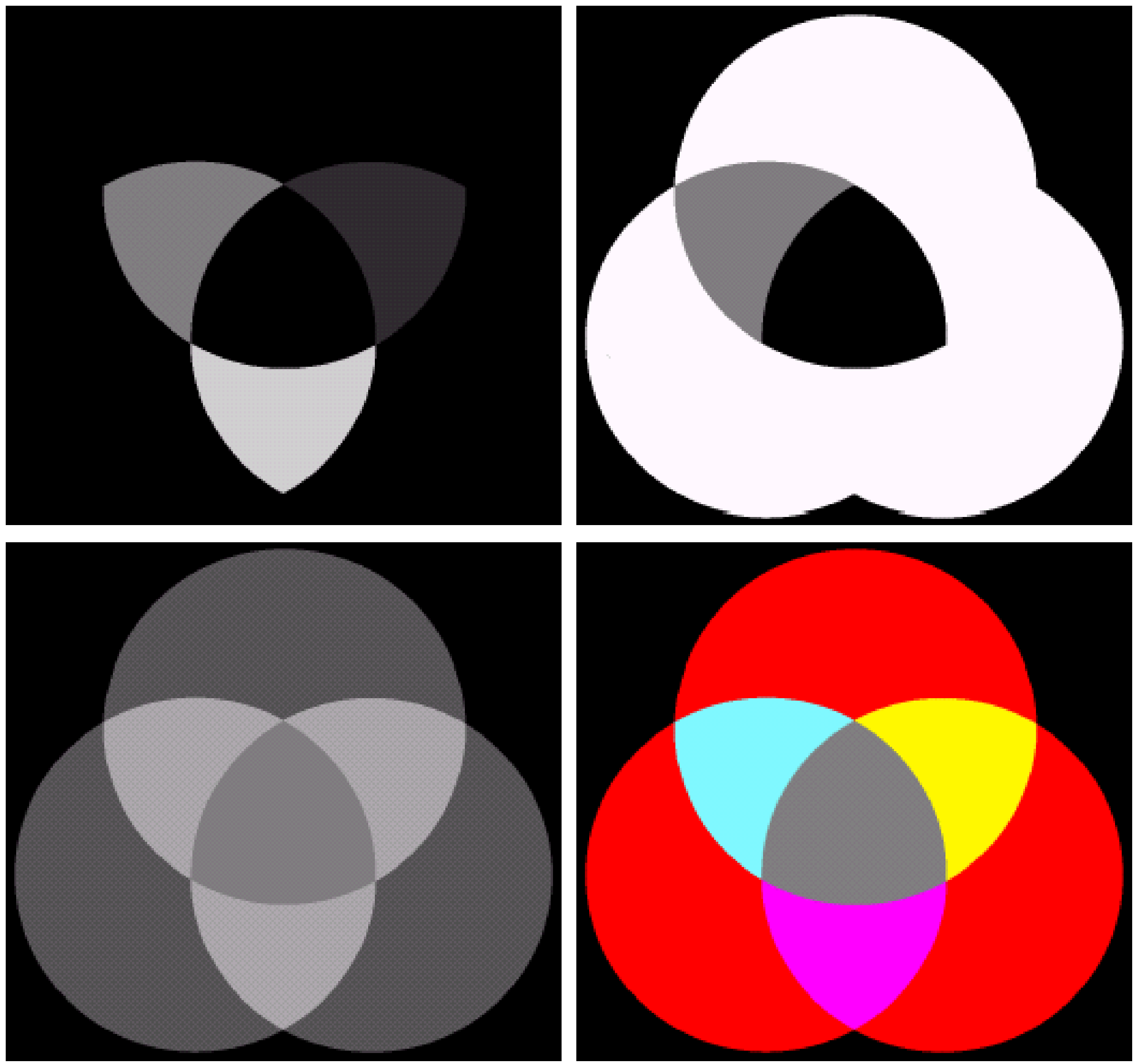
Hue



Saturation

Intensity

HSI component images



Hue

Saturation

Intensity

R,G,B

HSI component images

- In hue component image, red has been mapped to black since red corresponds to 0 deg hue.
- Gray levels for saturation component correspond to saturation (scaled to [0,255])
- Intensity component gives average intensities
- To change individual color of any region in RGB image, change value of corresponding region in hue image
  - Convert the new *H* image, alongwith unchnaged *S* and *I* images back to RGB
- To change purity, make changes in *S* image.
- To change intensity, make changes in *I* image.

Comparison

Example Comparison:

CMYK,

RGB,

and HSI

YC_bC_r Color Space

YC_bC_r Color Space is used in MPEG video compression:

- Y is luminance
- C_b is blue chromaticity
- C_r is red chromaticity

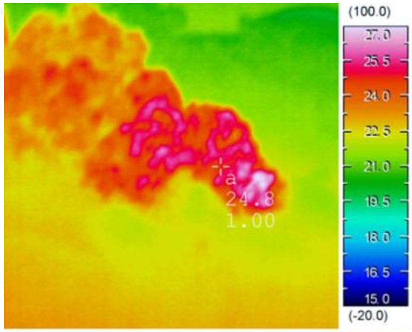
$$Y = 0.257*R + 0.504*G + 0.098*B + 16$$

$$C_r = 0.439*R - 0.368*G - 0.071*B + 128$$

$$C_b = - 0.148*R - 0.291*G + 0.439*B + 128$$

Pseudo-color image processing

- Pseudo-color = false color
- Assign colors to gray values based on a specified criterion.
- For human visualization and interpretation of gray-scale events.
  - Intensity slicing.
  - Gray level to color transformations.

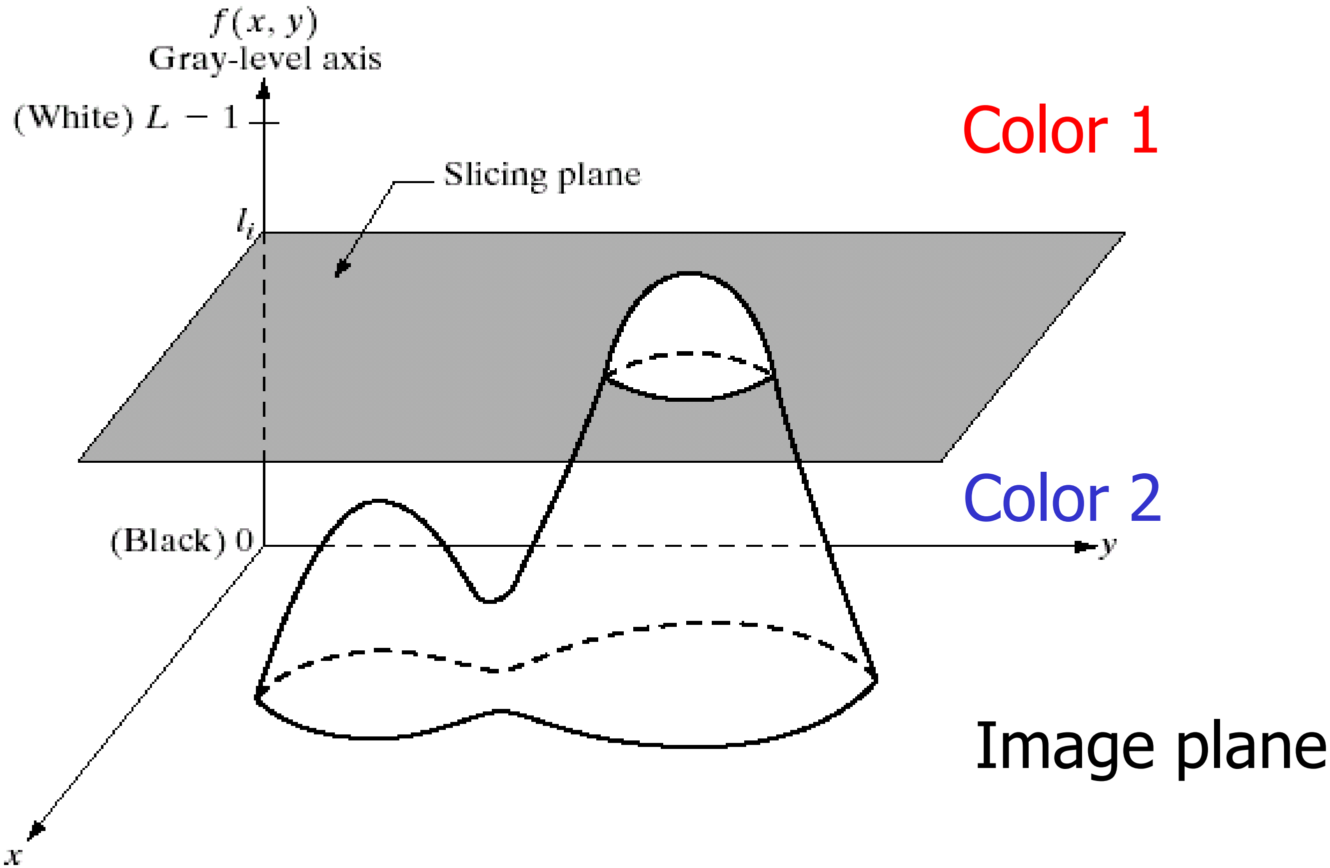


Pseudo-color image processing

- Consider an image as a 3D function mapping:
  - Spatial coordinates to intensities (that we can consider heights)
- Consider placing planes at certain levels parallel to the coordinate plane:
  - If a value is on one side of such a plane, it is rendered in one colour, and a different colour if on the other side

Intensity slicing

- 3-D view of intensity image



Intensity slicing

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 $l_1, l_2, \ldots, l_p$
- Assuming that $0 < P < L-1$ then the P planes partition the grey scale into $P + 1$ intervals $V_1, V_2, \ldots, V_{P+1}$

Intensity slicing

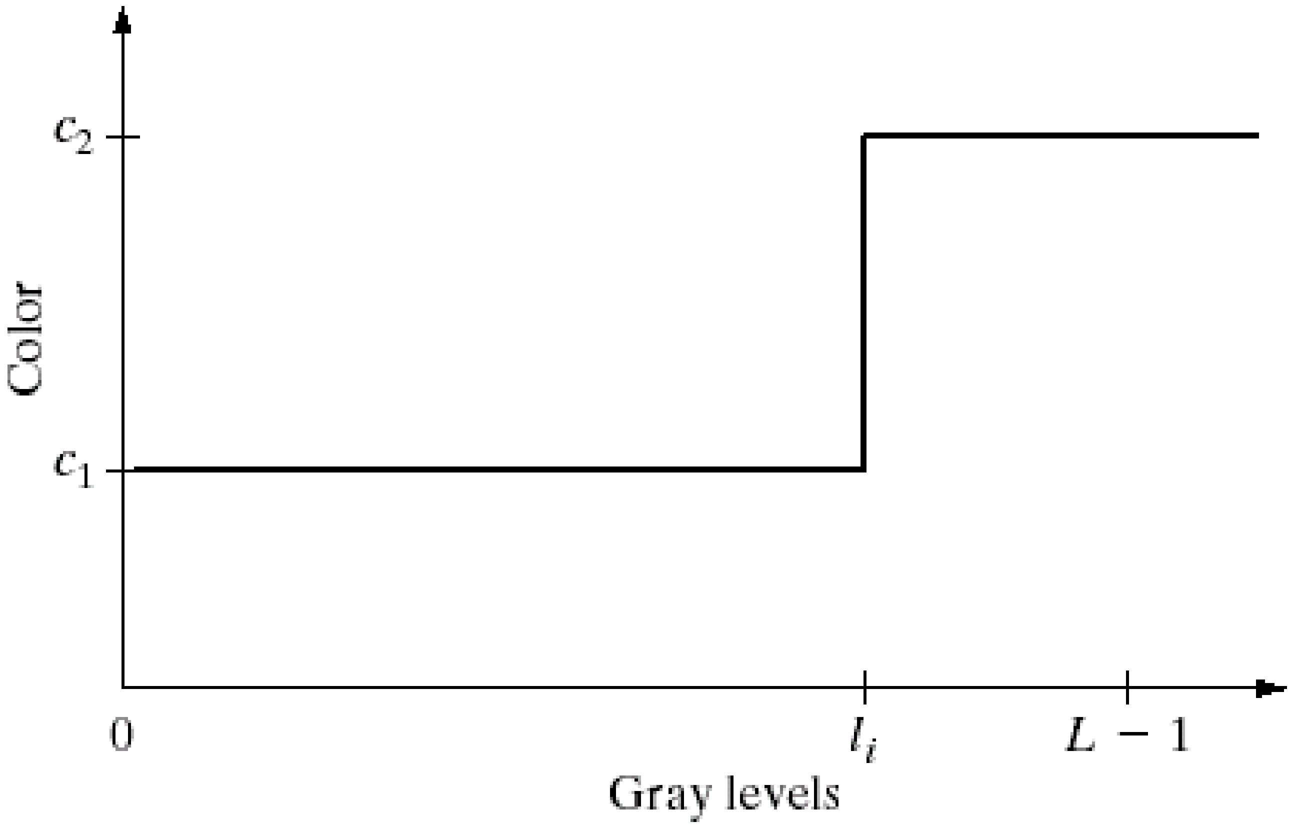
- Grey level colour assignments can then be made according to the relation:

$$f(x,y) = c_k \qquad \text{if } f(x,y) \in V_k$$

- where c_k is the colour associated with the k^{th} intensity level V_k defined by the partitioning planes at $l = k - 1$ and $l = k$

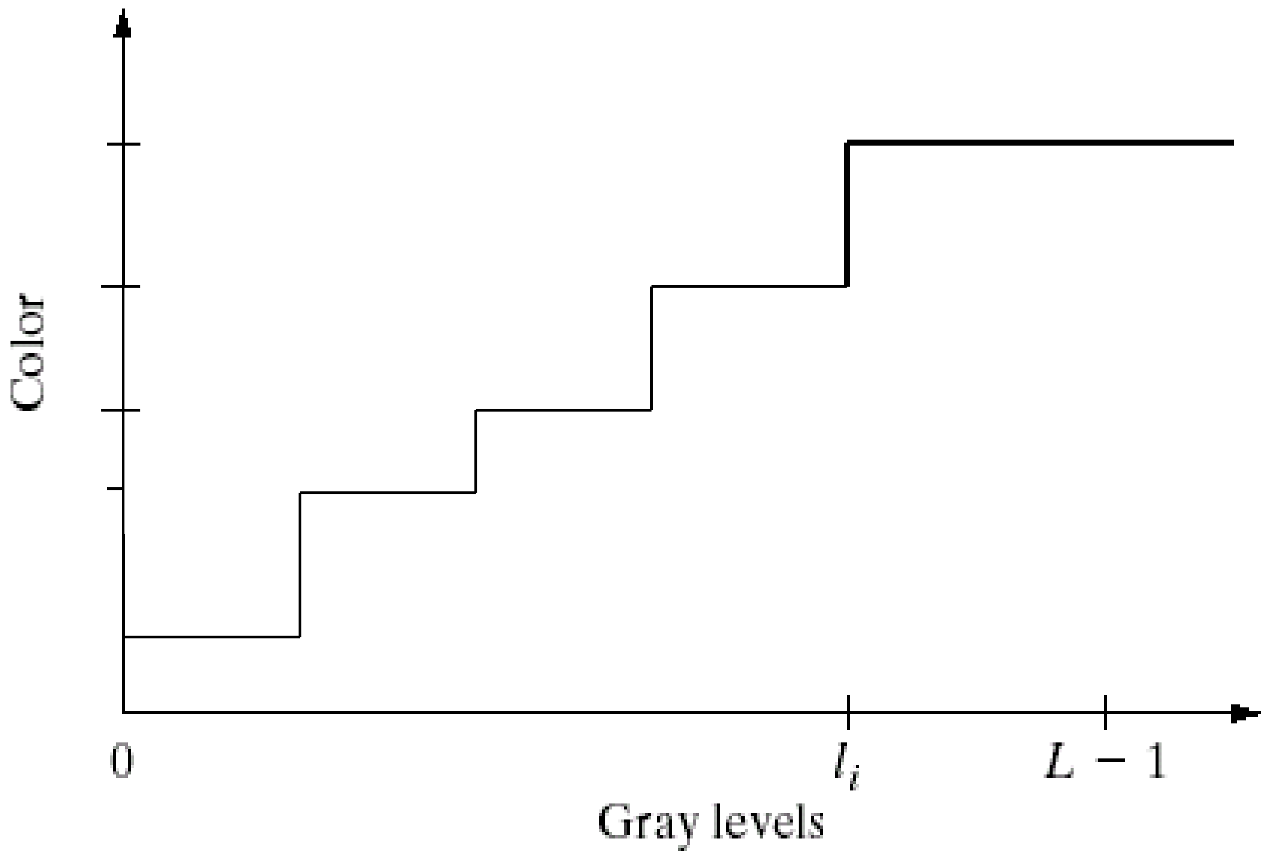
Intensity slicing (cont.)

- Alternative representation of intensity slicing

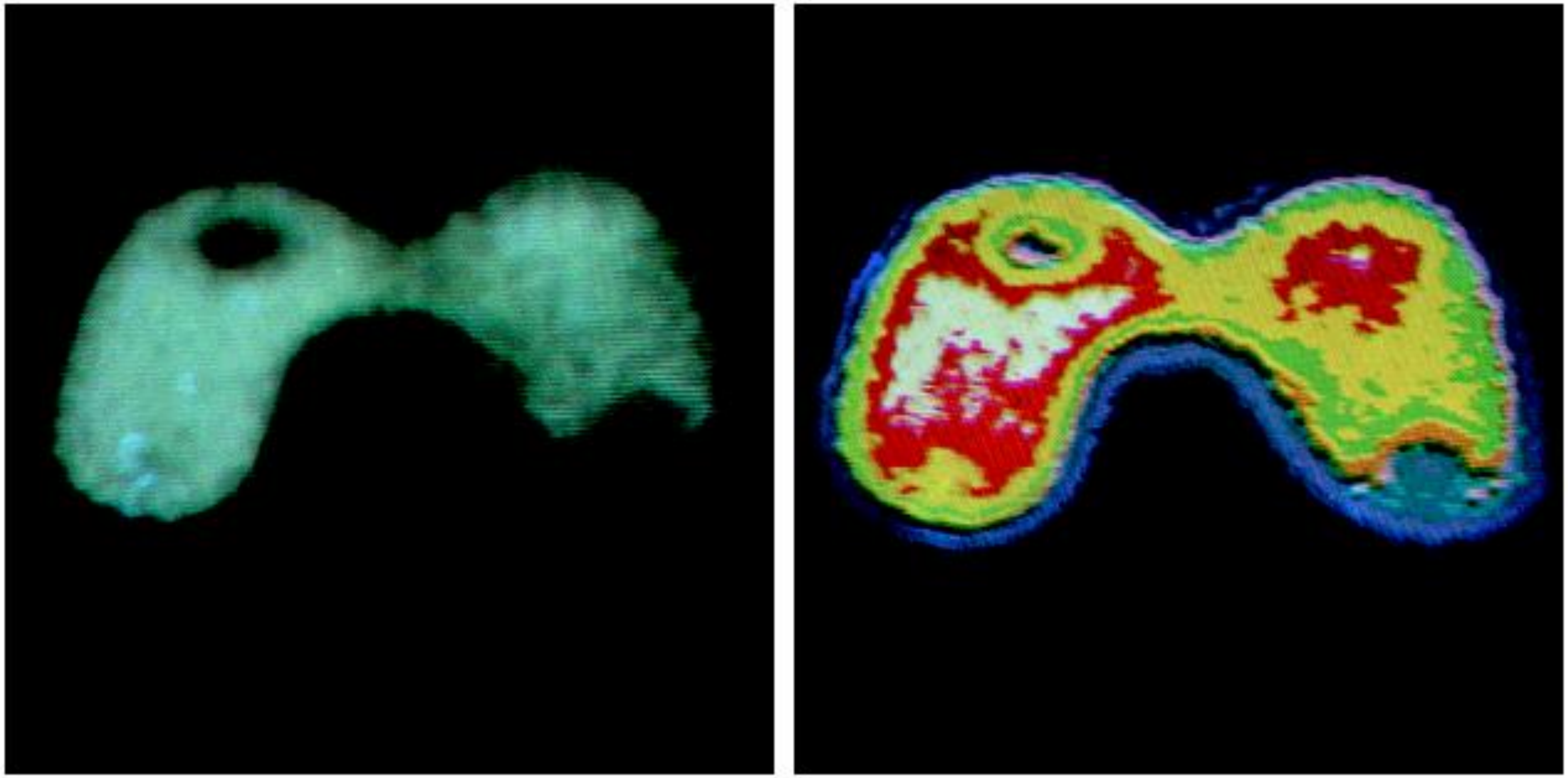


Intensity slicing (cont.)

- More slicing plane, more colors



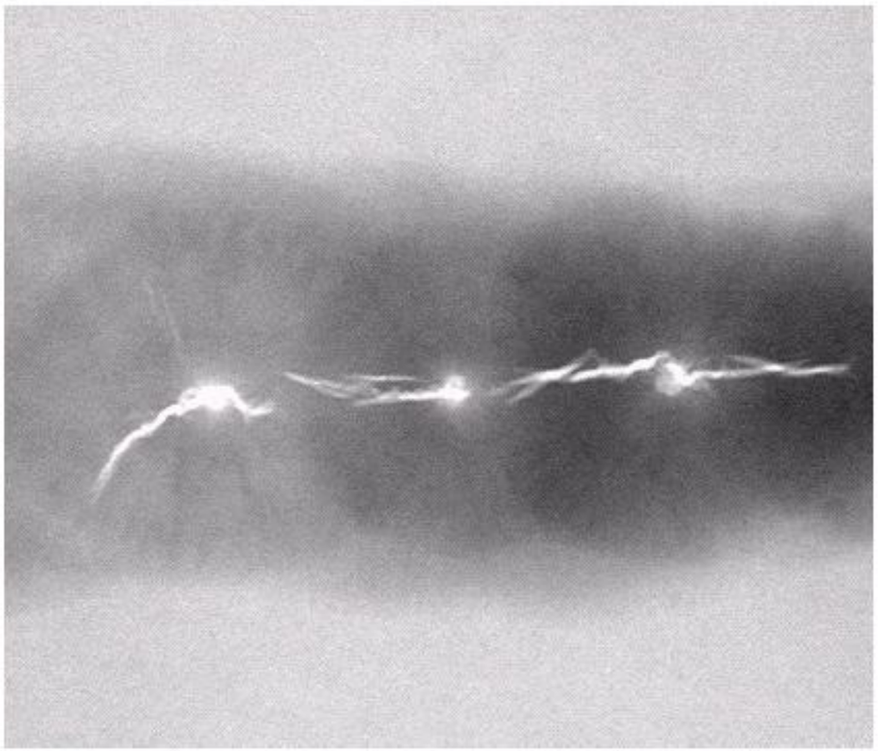
Application 1



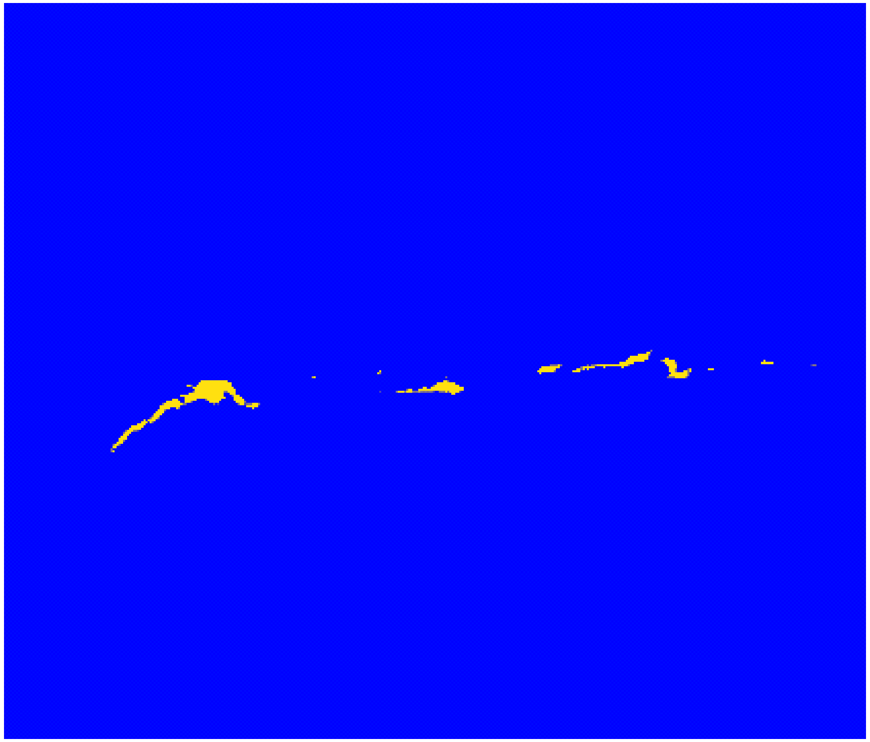
Radiation test pattern → 8 color regions

* See the gradual gray-level changes

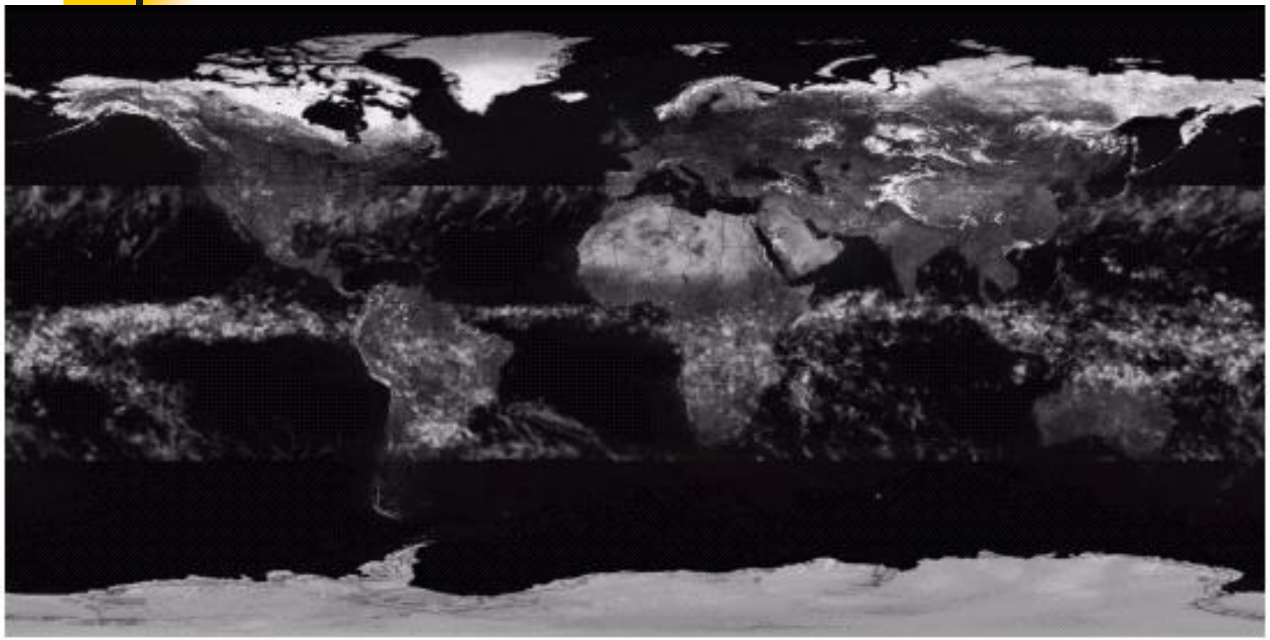
Application 2



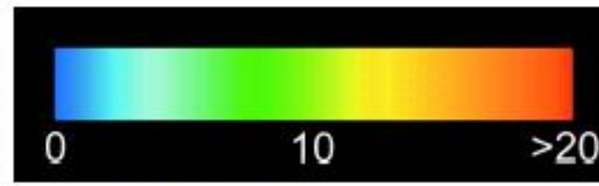
X-ray image of a weld

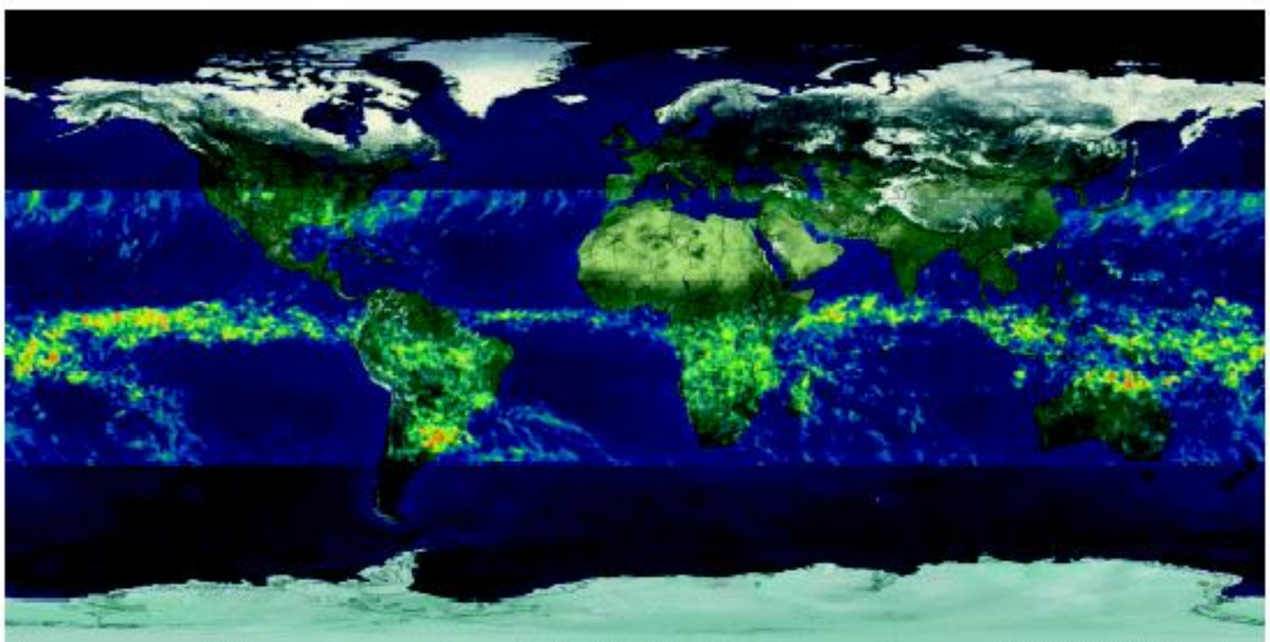


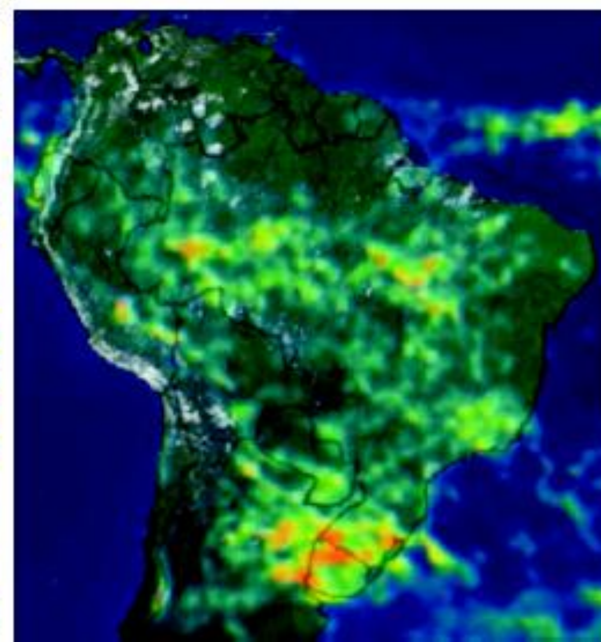
Application 3



Rainfall statistics







Gray level to color transformation

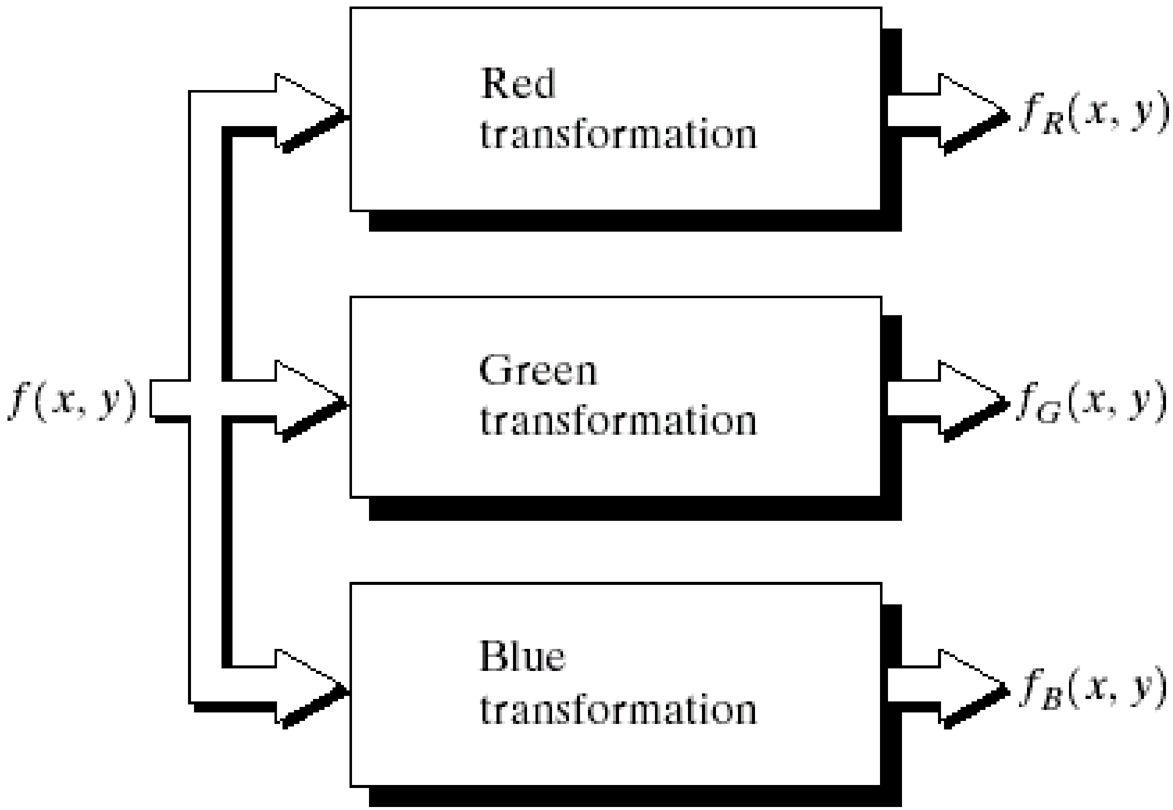
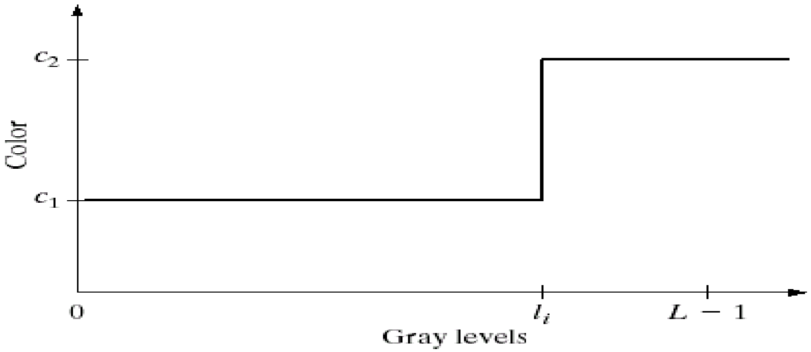


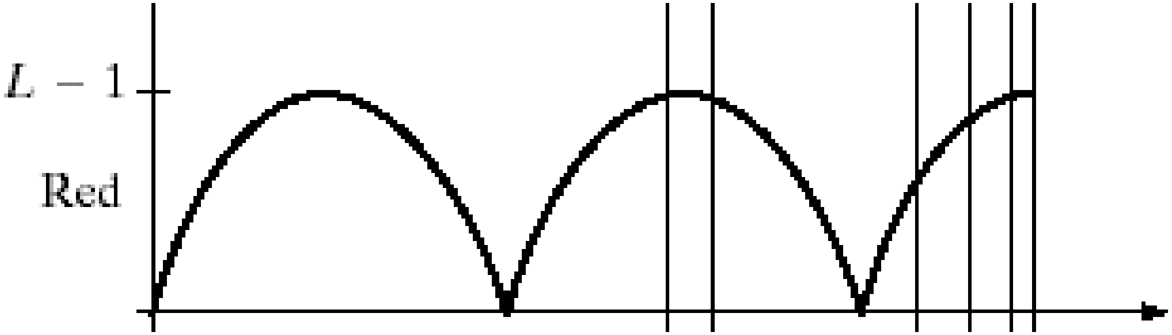
FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Gray level to color transformation

- Intensity slicing: piecewise linear transformation



- General gray level to color transformation



Application 1

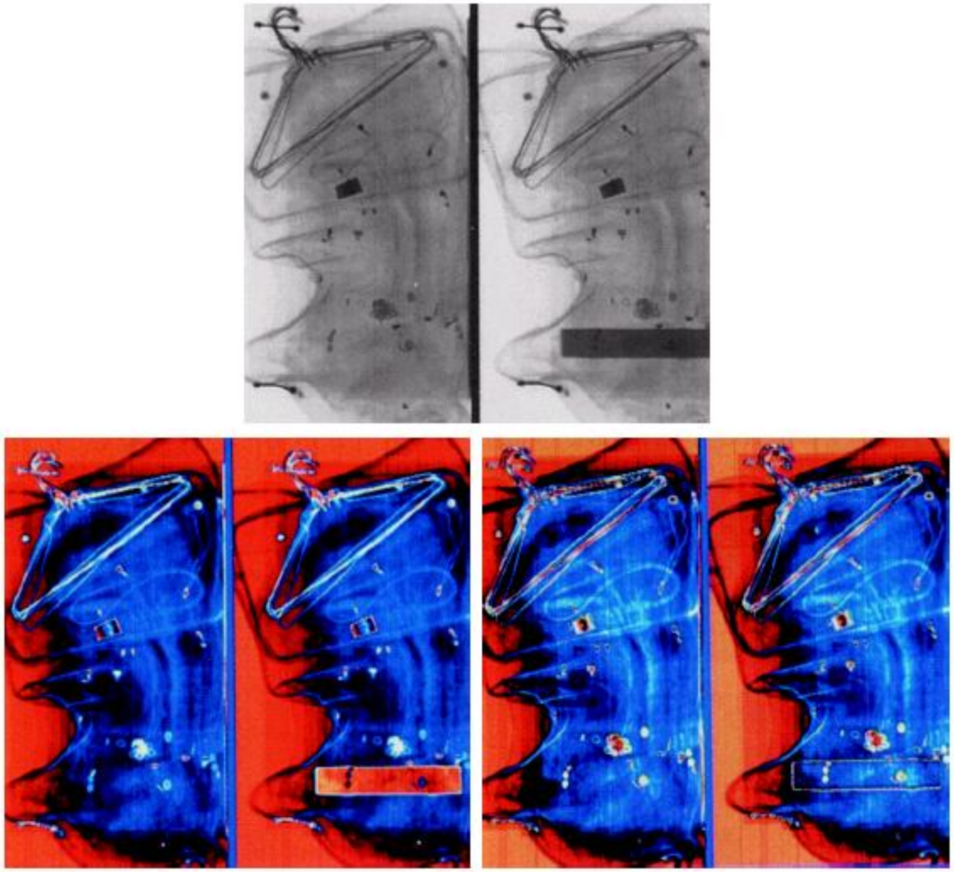
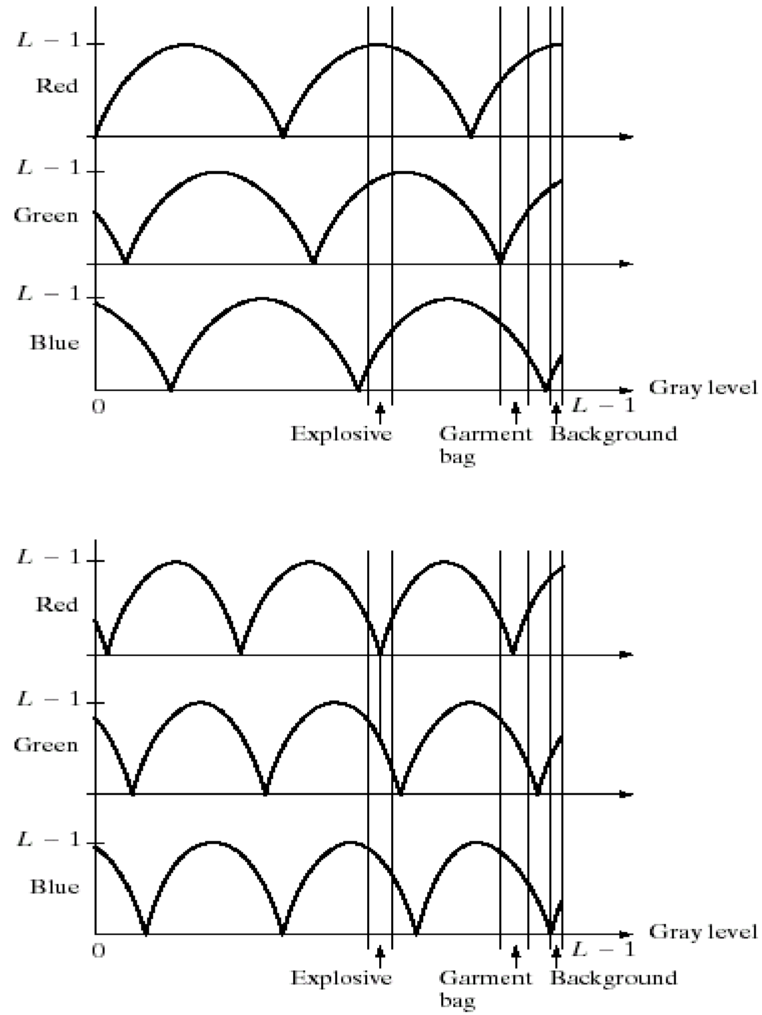


FIGURE 6.24 Pseudocolor enhancement by using the gray-level to color transformations in Fig. 6.25. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)

Application 1 (contd...)



a
b

FIGURE 6.25 Transformation functions used to obtain the images in Fig. 6.24.

Combine several monochrome images

Example: multi-spectral images

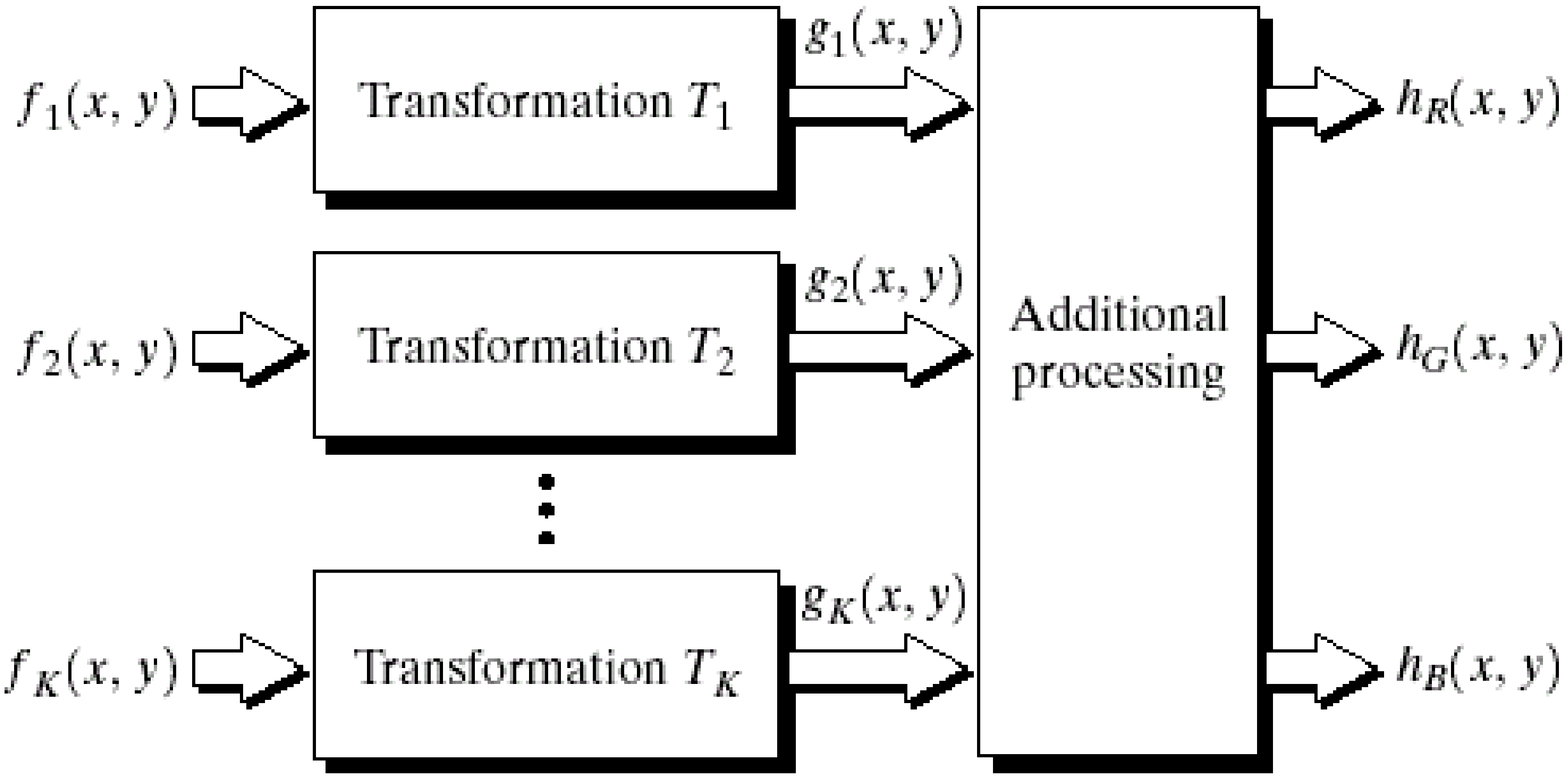
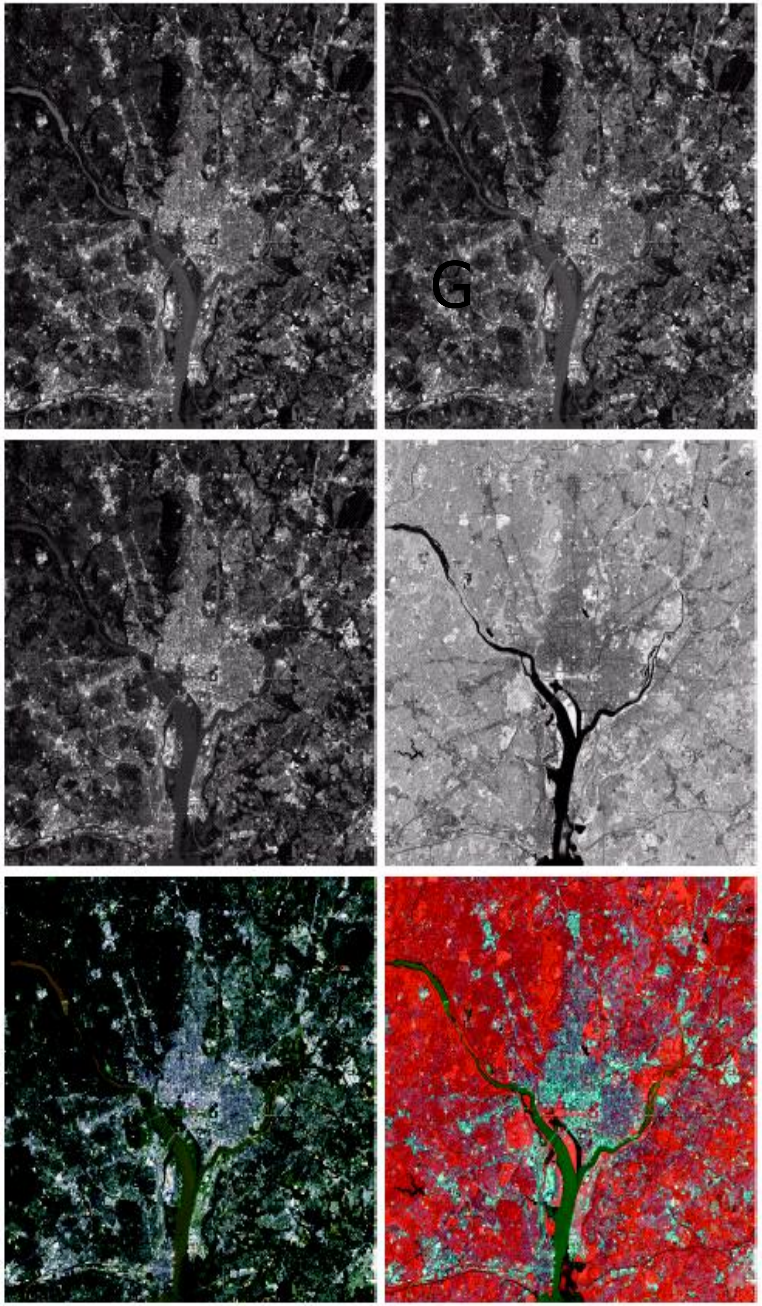


TABLE 1.1 Thematic bands in NASA's LANDSAT satellite.

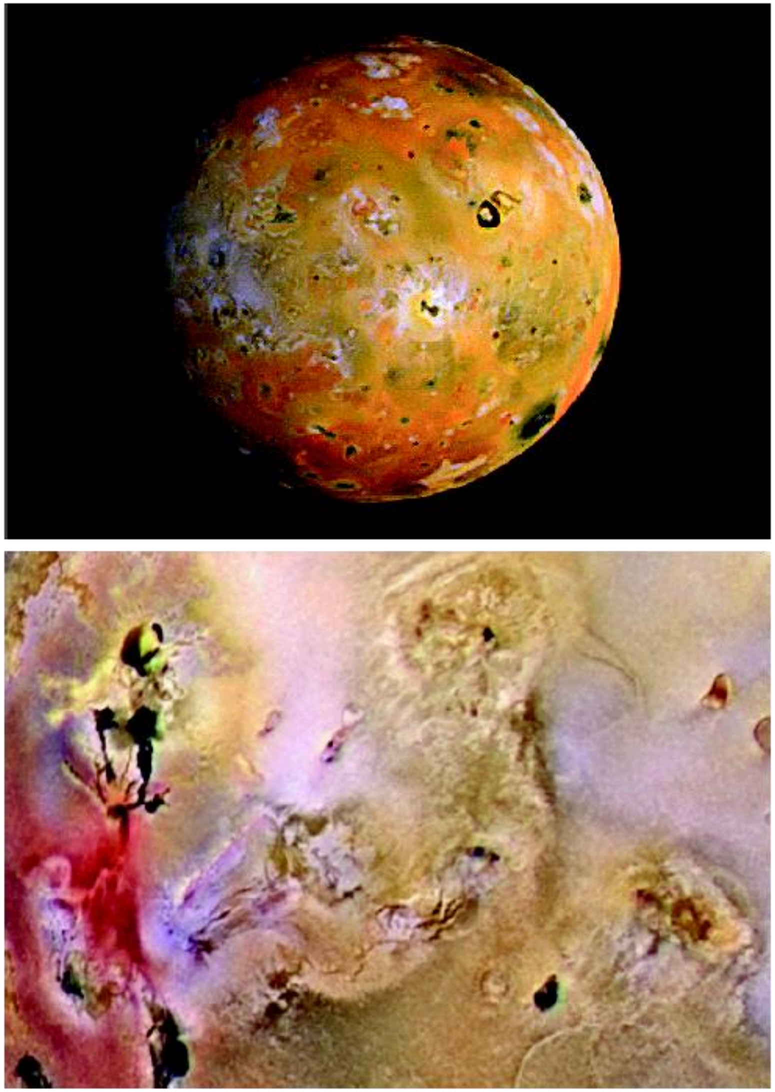
Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping



a	b
c	d
e	f

FIGURE 6.27 (a)–(d) Images in bands 1–4 in Fig. 1.10 (see Table 1.1). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (Original multispectral images courtesy of NASA.)

Gray level to color transformation



a
b

FIGURE 6.28
(a) Pseudocolor rendition of Jupiter Moon Io. (b) A close-up. (Courtesy of NASA.)

Full color image processing

- Let ***c*** be a vector in RGB color space:

$$c = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Color components are a function of *spatial* coordinates (*x*,*y*):

$$c(x, y) = \begin{bmatrix} c_R(x, y) \\ c_G(x, y) \\ c_B(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}$$

Full color image processing

- For an image of size $M*N$, there are M x N such vectors $\mathbf{c}(x,y)$, for

$$x = 0, 1, 2, \ldots, M-1$$

$$y = 0, 1, 2, \ldots, N-1$$

How to deal with color vector?

- Per-color-component processing:
  - Process each color component.
- Vector-based processing:
  - Process the color vector of each pixel.
- When can the above methods be equivalent?
  - Process can be applied to both scalars and vectors.
  - Operation on each component of a vector must be independent of the other component.

Color transformation

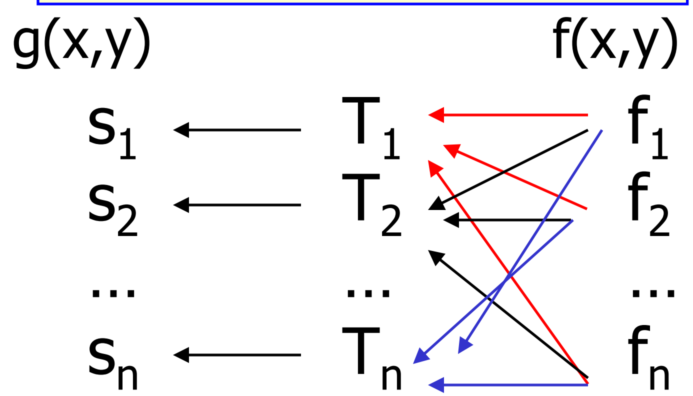
- Similar to gray scale transformation:
  - $g(x,y) = T[f(x,y)]$
- Color transformation

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

Color components of ***g***

Color mapping functions

Color components of ***f***

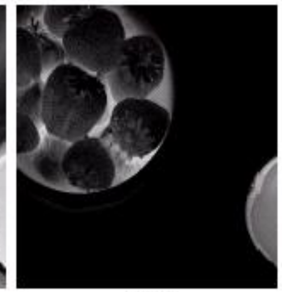



Full color


Cyan


Magenta

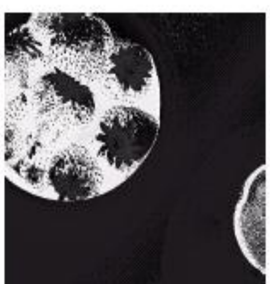

Yellow


Black


Red

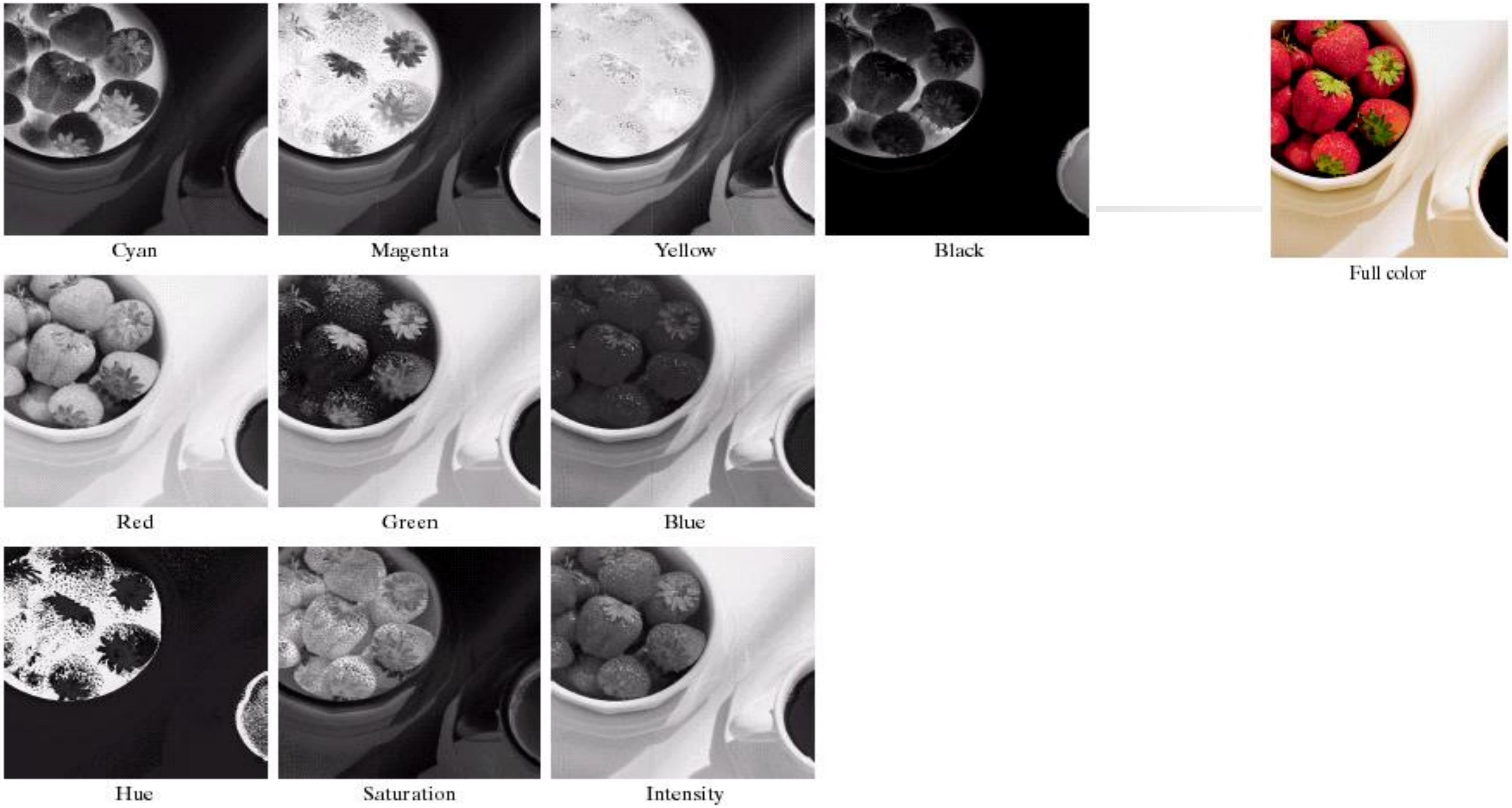

Green


Blue


Hue


Saturation


Intensity



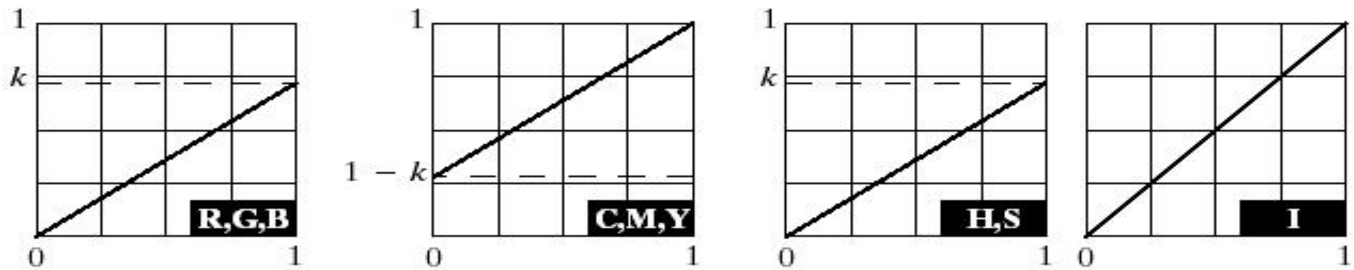
- Some difficulty in interpreting the HUE:
  - Discontinuity where 0° and 360° meet.
  - Hue is undefined for a saturation 0.

Use which color model in color transformation?

- RGB ⇔CMY(K) ⇔ HSI
- Theoretically, any transformation can be performed in any color model.
- Practically, some operations are better suited to specific color model.

Example: modify intensity of a color image

- Example: $g(x,y) = k\,f(x,y),\quad 0 < k < 1$
- HSI color space
  - Intensity: $s_3 = k\,r_3 \qquad (s_1 = r_1,\ s_2 = r_2)$
  - Note: transform to HSI requires complex operations
- RGB color space
  - For each R,G,B component: $s_i = k\,r_i \quad i=1,2,3$
- CMY color space
  - For each C,M,Y component:
  - $s_i = k\,r_i + (1-k) \qquad i=1,2,3$



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

$$s_i = kr_i \qquad i = 1,2,3$$

$$s_i = kr_i + (1-k) \qquad i = 1,2,3$$

$$s_1 = r_1$$

$$s_2 = r_2$$

$$s_3 = kr_3$$

Color complement

- Hues directly opposite one another on the *color circle* are called complements (analogous to gray scale negatives).

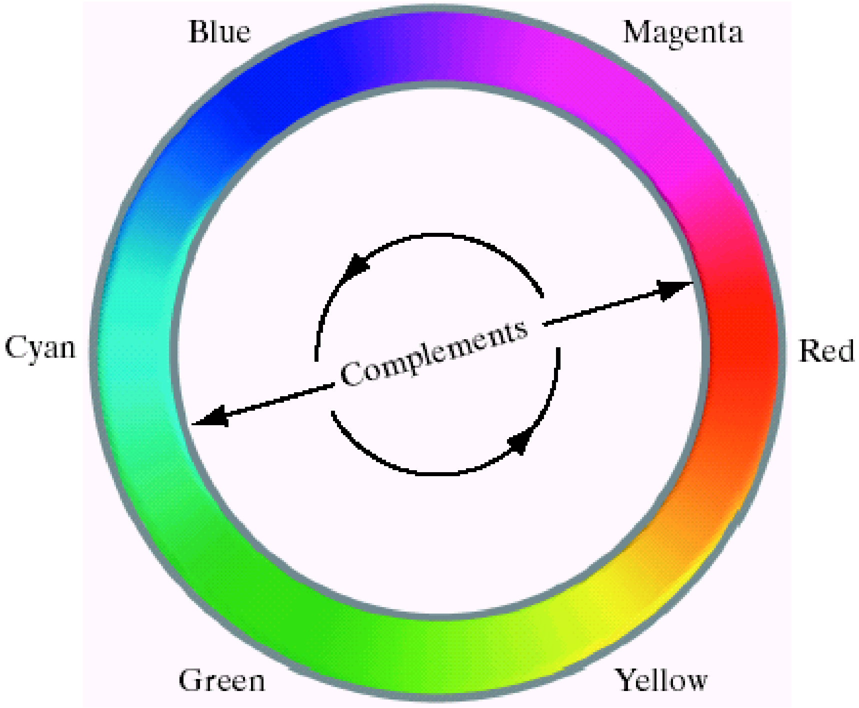
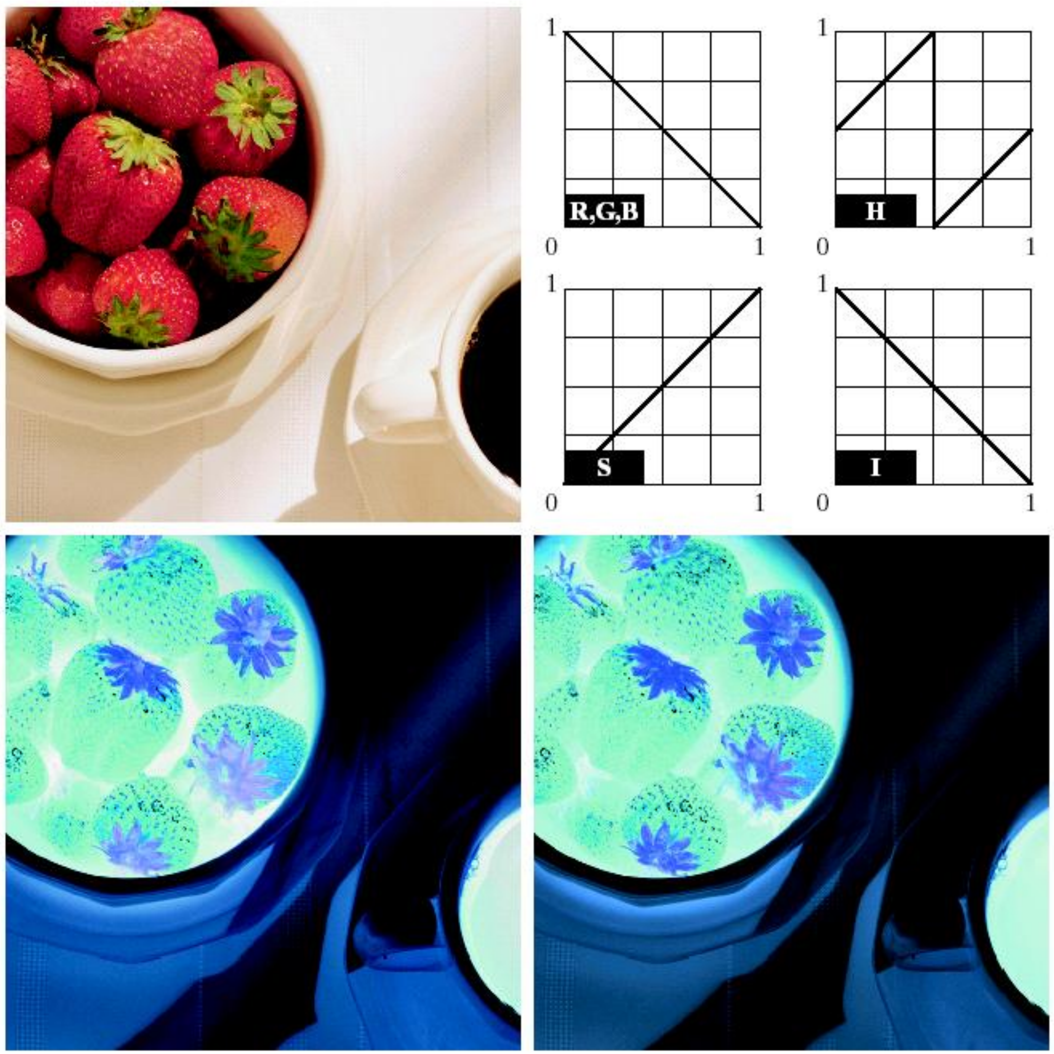


FIGURE 6.32 Complements on the color circle.

Color complement



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

- Highlighting a specific range of colors in an image.
  - Display a color of interest to make it stand out from the background.
  - Use the region defined by colors as a mask for further processing.
- Each pixel's transformed color components should be a function of all *n* original pixel's color components.
  - Map the colors outside some range of interest to a non-prominent neutral color.

Color slicing

- If colors of interest are enclosed by a cube (for n=3) of width W and centered at some prototypical color with components (a_1, a_2,…,a_n), the transformations are:

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

- These transformations highlight the colors around the prototype by forcing all other colors to the midpoint of the reference color space.
  - For RGB color space, a suitable neutral point is middle gray (0.5, 0.5, 0.5)

Implementation of color slicing

- Colors of interest are enclosed by a cube (or hypercube for n>3):

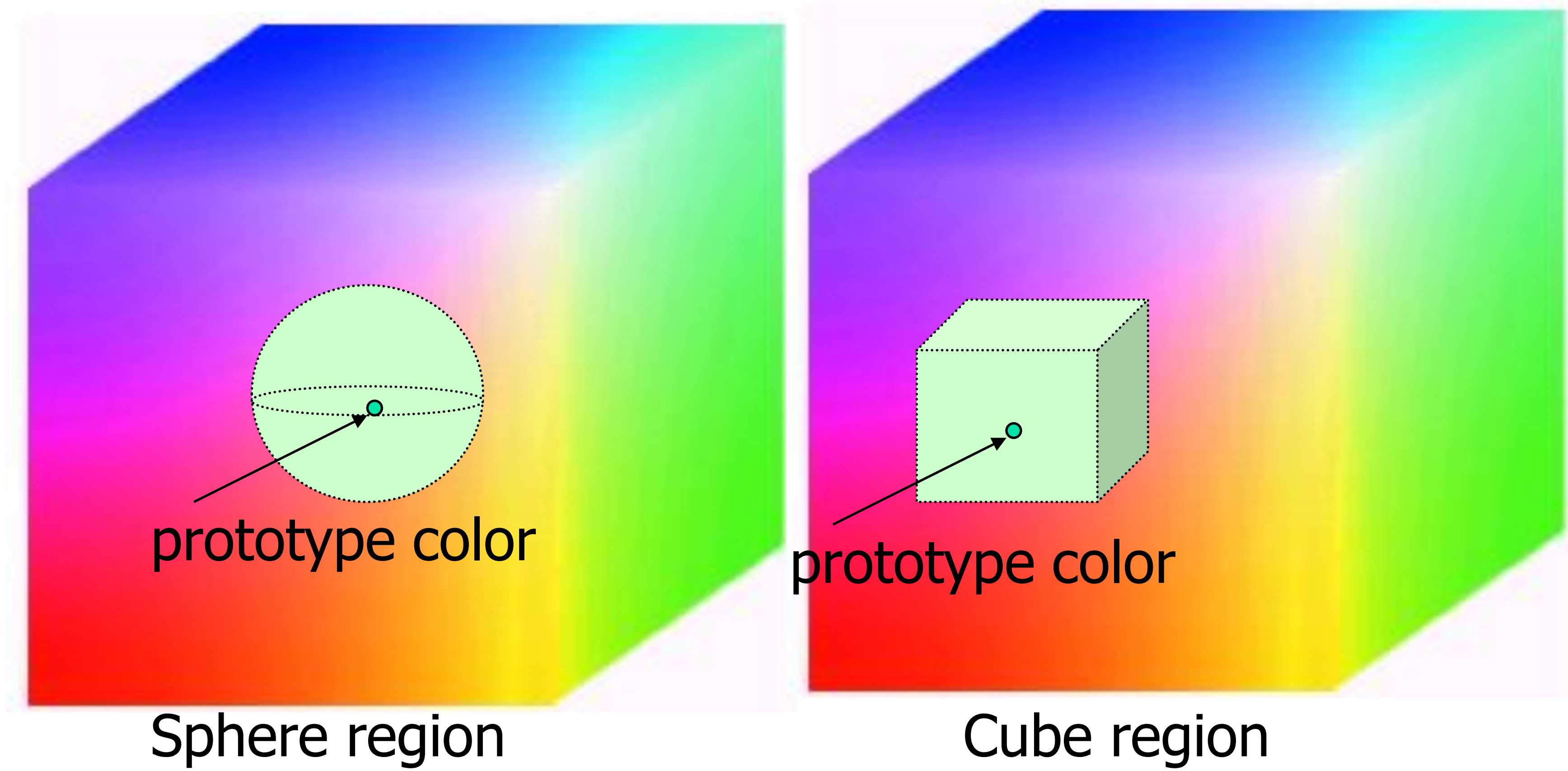
$$s_i = \begin{cases} 0.5 & if \left[\left| r_j - a_j \right| > \frac{W}{2} \right]_{any 1 \le j \le n} \\ r_i & otherwise \end{cases}, \qquad i = 1,2,\ldots,n$$

- Colors of interest are enclosed by a sphere with radius R_0 $(a_1, a_2, \ldots, a_n)$ are the components of its protypical color :

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

Implementation of color slicing

- How to take a region of colors of interest?



Application

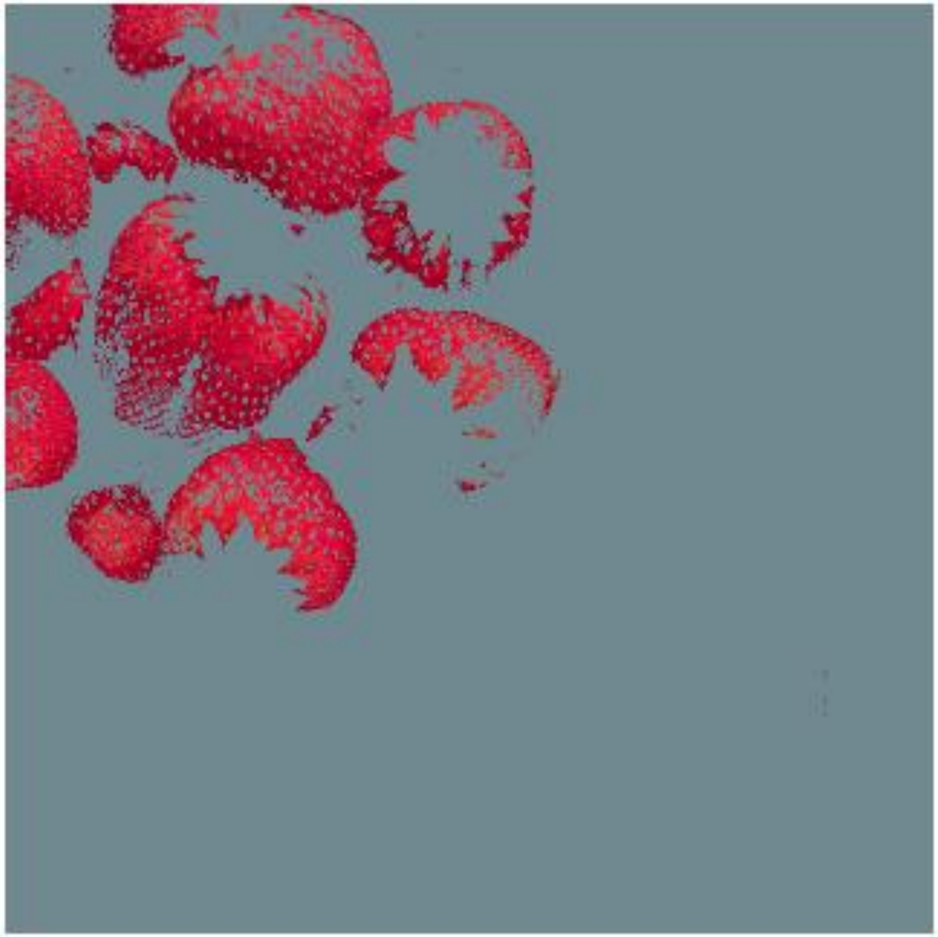
Full color

- The transformations have been used to separate the edible part of the strawberries from the background.
  - In each case, a prototype red with RGB color coordinate (0.686, 0.161, 0.192) was selected from the most prominent strawberry.
  - W and R_0 were chosen so that the highlighted region would not expand to undesirable portions of the image (W= 0.255, R_0=0.177).

Application

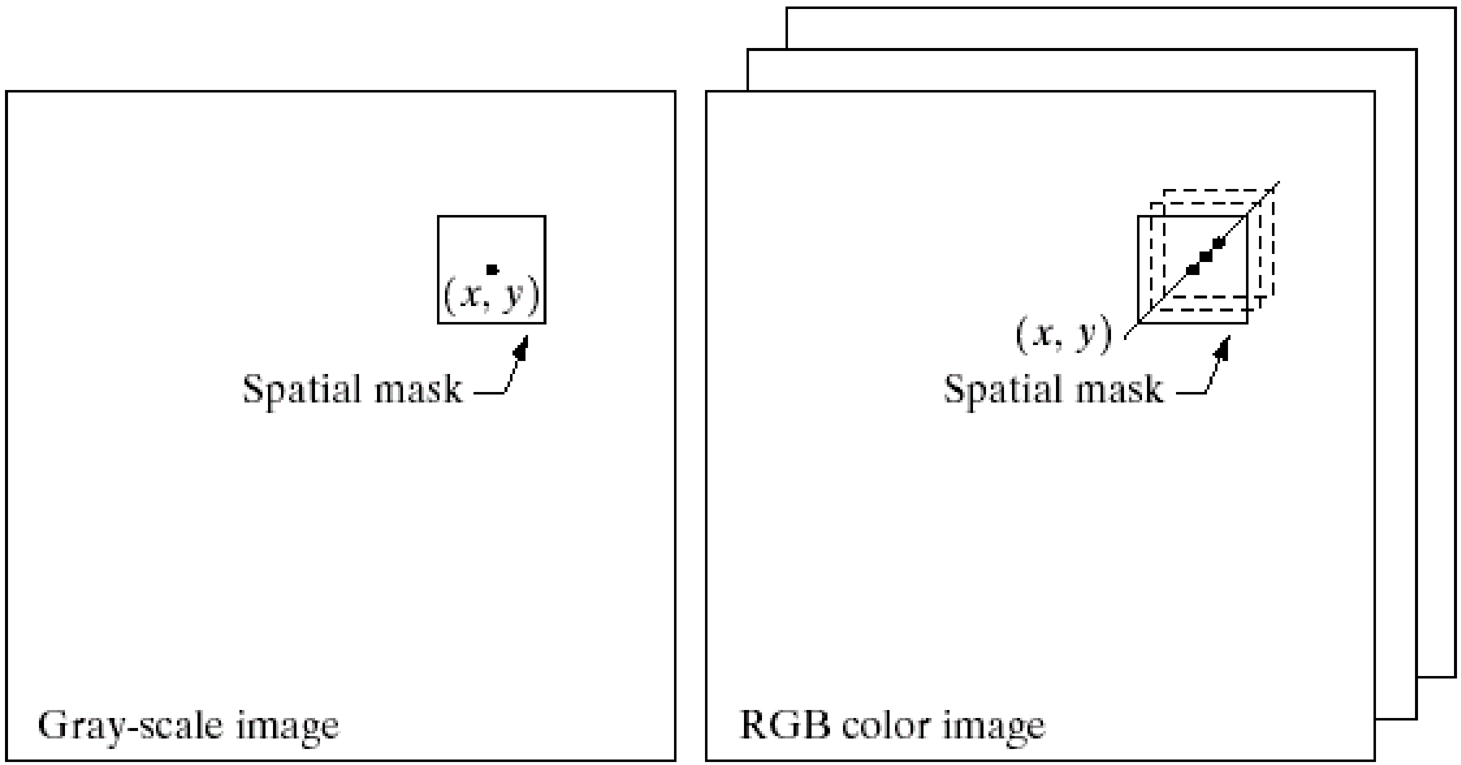

Full color


cube


sphere

Color image smoothing

- Neighborhood processing: modify value of each pixel based on the characteristics of its surrounding pixels.



Averaging mask

- The average of RGB component vectors in the neighbourhood is:

$$\bar{\mathbf{c}}(x, y) = \frac{1}{K} \sum_{(x,y) \in S_{xy}} \mathbf{c}(x, y)$$

Vector processing

Neighborhood centered at (x,y)

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

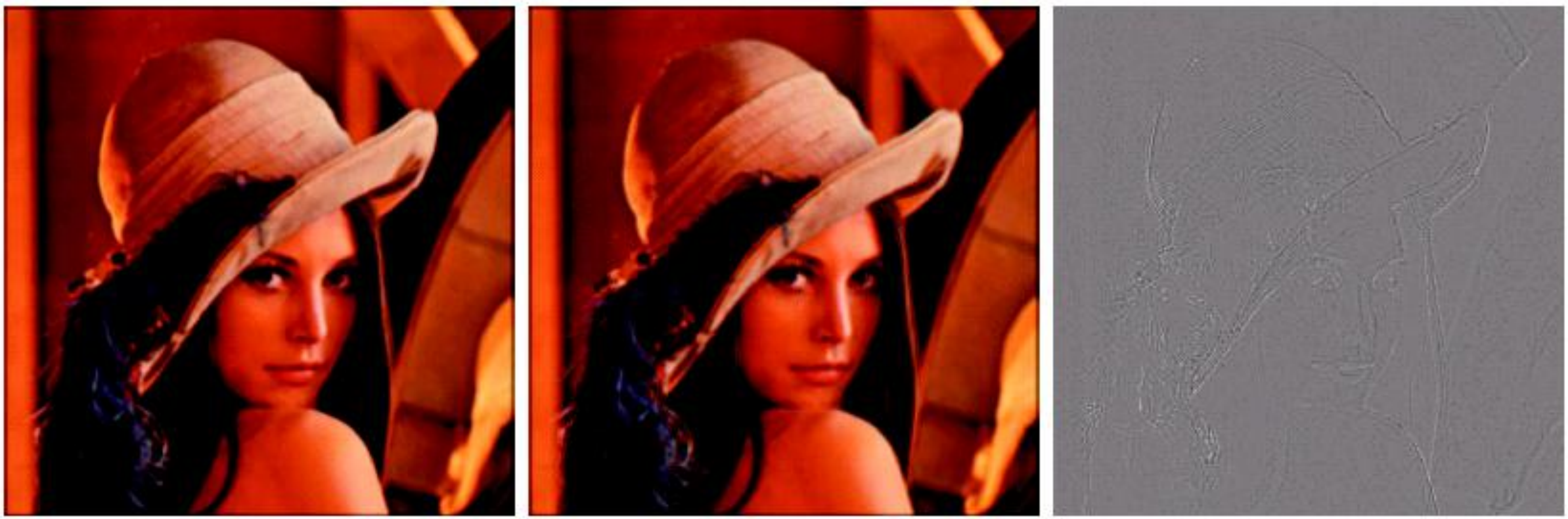
Per-component processing

The components of this vector are scalar images obtained by independently smoothing each plane of the RGB image


original
R
G
B
H
S
I

Example: 5x5 smoothing mask

RGB model | Smooth I in HSI model | difference



a b c

FIGURE 6.40 Image smoothing with a 5×5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

Example: 5x5 smoothing mask

- RGB color planes are smoothed independently and then combined to form a smoothed image.
- In HSI color model, only intensity component is smoothed.
  - The smoothed image is combined with the original hue and saturation image to form RGB image.
- Both images are similar but the difference is present.
- The average of two pixels of different colors is a mixture of the two colors, not either of the two colors.
- By smoothing only intensity image, pixels maintain their original hue and saturation (and thus their original color).

Tone and color correction

In conjunction with digital cameras, flatbed scanners, and inkjet printers, they turn a personal computer into a *digital darkroom*

↓

The colors of monitor should represent accurately any digitally scanned source images, as well as the final printed out

↓

Device Independent Color Model

↓

The model of choice for many color management system (CMS) is *CIE L*a*b** (also called CIELAB) model

Tone and color correction

- L^*a^*b is an excellent decoupler of intensity (lightness L^*) and color (a^* for red minus green and b^* for blue minus green)
- The tonal range of an image, also called its key-type, refers to its general distribution of color intensities.
  - High-key images: Most of the information is concentrated at high intensities.
  - Low-key images: Most of the information is concentrated at low intensities.
  - Middle key images lie in between.

Tone and color correction

Middle-key Image

FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Tone and color correction

High-key Image

FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Tone and color correction

Low-key Image

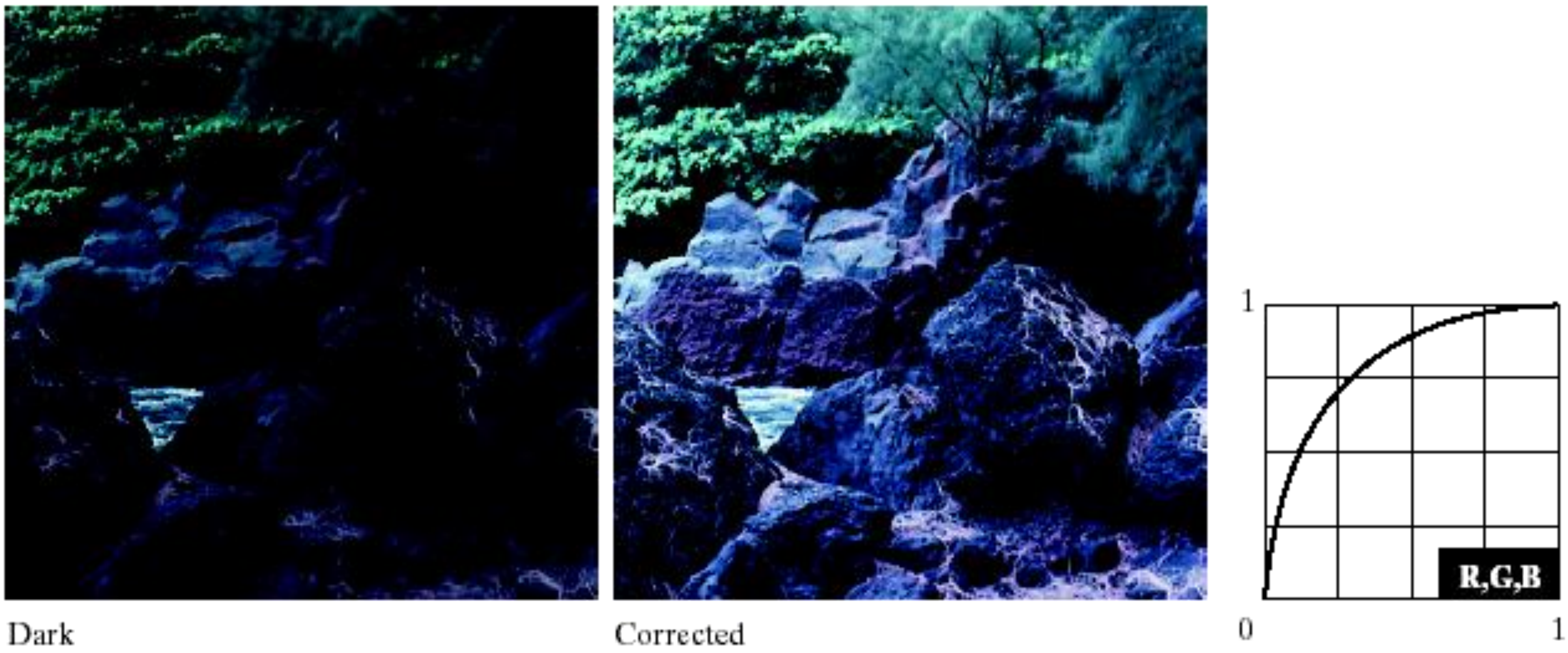


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Color image sharpening

- Laplacian of a vector is a vector whose components are equal to the Laplacian of the individual scalar components of the input vector.
- For RGB color system, Laplacian of vector **c** is:

$$\nabla^2\left[\mathbf{c}(x,y)\right] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}$$

Color image sharpening

Sharpening R,G and B

Sharpening Intensity

Difference

a b c

FIGURE 6.41 Image sharpening with the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the intensity component and converting to RGB. (c) Difference between the two results.

Colors segmentation

- Segmentation is a process that partitions an image into regions
  - Segmentation in HSI Color Space
  - Segmentation in RGB Vector Space
  - Color Edge Detection

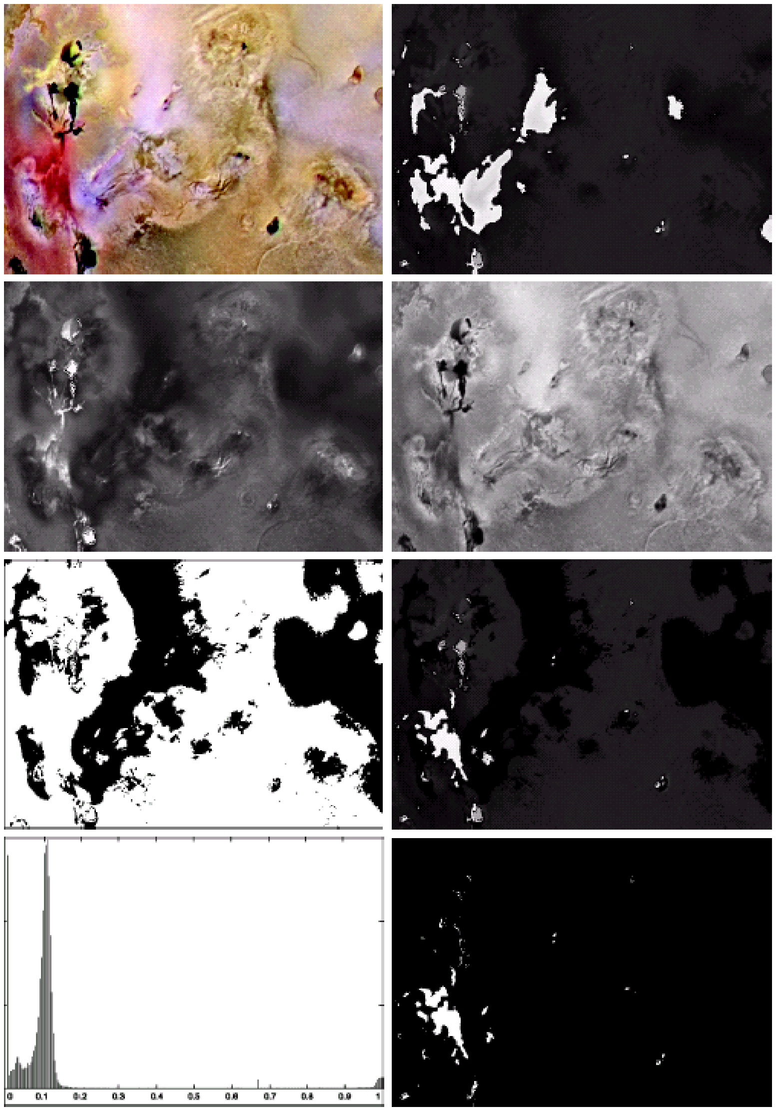
Color segmentation (HSI)

- Color is conveniently represented in hue image.
- Typically, saturation is used as a masking image to isolate further regions of interest in hue image.
- Intensity image carries no color information
- Example: Image of Jupiter:
  - The reddish region in the lower left of the image is to be segmented.
  - It can be seen in hue image that our region-of-interest has a higher hue, i.e. colors are on blue-magenta side of red.
  - Create binary mask: threshold saturation image with a threshold equal to 10% of maximaum value in saturation image.
    - Pixel value greater than threshold is set to white. All others are set to black.

Color segmentation

- Take product of mask with hue image.
- Compute histogram of product image.
- It is seen that high values (values are of interest) are near 1.0.
- Threshold product image with 0.9 to give final binary image.
- Spatial location of white points maps to reddish hue of interest.

Color segmentation



a b
c d
e f
g h

FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).

Color segmentation (RGB space)

- Better segmentation results are generally obtained by using RGB color vectors.
- Given a set of sample color points representing colors of interest, obtain an estmate of average color.
  - Let color average be denoted by vector **a**.
- Objective of segmentation is to classify each RGB pixel as having color in specified range or not.
  - Define a measure of similarity eg. Euclidean distance.
  - Euclidean distance between an arbitrary point in RGB space **z** and **a** should be less than a specified threshold D_0 for them to be similar.

$$
\begin{aligned}
D(\mathrm{z},\mathrm{a}) &= \|\mathrm{z}-\mathrm{a}\| \\
&= \left[(\mathrm{z}-\mathrm{a})^T(\mathrm{z}-\mathrm{a})\right]^{1/2} \\
&= \left[(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2\right]^{1/2}
\end{aligned}
$$

Color segmentation (RGB space)

- The locus of points such that $D(\mathbf{z}, \mathbf{a}) \le D_0$ is a solid sphere of radius D_0.
  - Points within or on surface of sphere satisfy specified color criterion.
  - Coding them with black and white produces binary image.
- A bounding box can also be used.
  - Box is centered on **a** and its dimensions along each of the color axes is chosen proportional to standard deviation of samples along each axes.

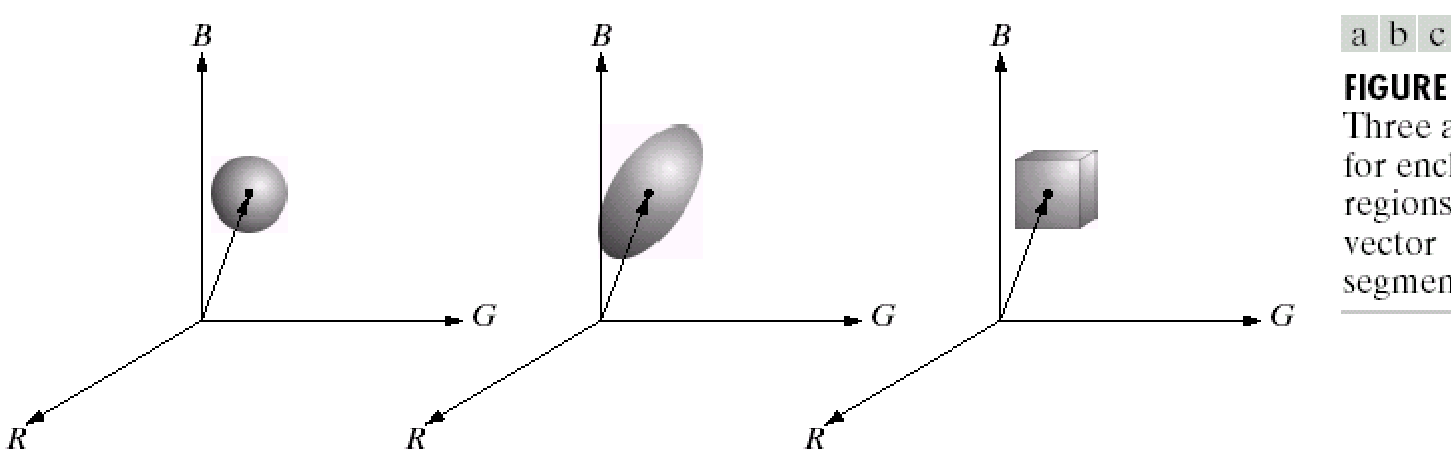
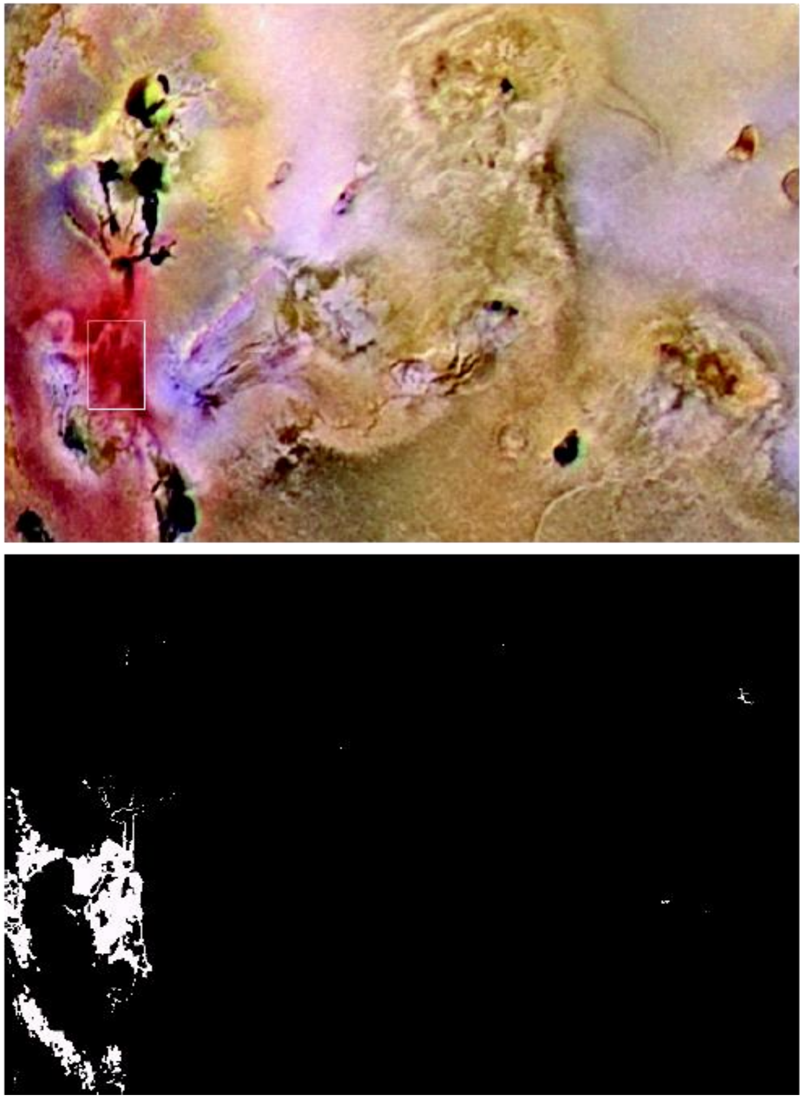


FIGURE 6.43 Three approaches for enclosing data regions for RGB vector segmentation.

Color segmentation (RGB space)



a
b

FIGURE 6.44 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 6.42(h).

Colors in Matlab

- Let fR, fG and fB represent three RGB component images.
  - An RGB image is formed from these images

    I=cat(3, fR, fG, fB)
  - The order in which the operands are placed in the operand matters.
- To convert RGB image to gray:

    I1=rgb2gray(I)
- To convert RGB image to HSV space:

    I2=rgb2hsv(I)
- To convert RGB image to CMY:

    I3=imcomplement(I)

Colors in Matlab

- Colors can be specified in three different ways in MATLAB:
  - short name (eg. 'r'), long name (eg. 'red'), RGB triple (eg. [1 0 0])
- Here are the eight predefined colors in MATLAB with their names and **RGB** equivalents:

RGB value	Short name	Long name
[1 1 0]	y	yellow
[1 0 1]	m	magenta
[0 1 1]	c	cyan
[1 0 0]	r	red
[0 1 0]	g	green
[0 0 1]	b	blue
[1 1 1]	w	white
[0 0 0]	k	black

Colors in Matlab

- RGB color has a range [0 255] for each of the 3 colors (red/green/blue)
- Each color is divided by 255 to keep it within the [0 1] range that MATLAB requires
- How to just change the color of the line?

```
x = -pi:0.01:pi;
plot(x,sin(x),'Color',[200/255 200/255 10/255]);
```

- How to just change the background color of the plot?

```
whitebg([255/255 153/255 255/255])
```

Colormaps

- A colormap is an *m* x 3 array, each row of which contains the [R G B] levels of an individual color using numbers in the range 0 to 1
- For example:
  - row [1 0 0] represents red
  - [0 1 0] represents green
  - [0 0 1] represents blue
  - [1 1 1] represents white
  - [:5 :5 :5] represents medium gray and so on

Colormaps

- The MATLAB statement

  ```
  >>colormap(M)
  ```

  installs the matrix M to be used as the colormap in the current Figure window.
- Matlab has a number of buit in colormaps, each one, by default is a 64 x 3 array.
- The one that most concerns us is the gray color map. It displays monochorme graphics ranging from black ([0 0 0]) to white ([1 1 1]) in 64 grades.
- To see the contents of the gray colormap type

  ```
  >>gray
  ```

Colormaps

- Other colormaps are hot, cool, winter, summer,.... These colormaps accept an integer argument m. For example, the statement

  ```
  >>hot(8)
  ```

  rescales the range of colors of the colormap hot into 8 colors instead of 64. To see these colors type

  ```
  >>colormap(hot(8))
  ```

Colormaps

- Images are displayed using two statements: image() and imagesc().
- An image in Matlab is a data matrix that contains information about the pixels in the image. Usually a colormap is needed to display an image.
- There are three types of image data matrices:
  - indexed images
  - intensity images
  - RGB images.

Colormaps

- Indexed images: If an image is represented by a matrix X in indexed mode, then
  - Element X_{ij} represents the color of the pixel P_{ij} as an index into the colormap.
  - For example, if $X_{ij} = k$; then the color of pixel P_{ij} is the color represented by row k of the color map cmap.
  - This means that the values in the matrix X must be integer values in the range [1 length(cmap)].

Colormaps

- Intensity images: This type is usually used in monochromatic displays, e.g. gray colormap.
- Data in the matrix *X* do not have to be of any specific numerical type and they are rescaled over a given range and the result is used to index into the colormap.
- For example, the statements

```
>>imagesc(X, [0 1]);
>> colormap(gray)
```

  assigns the smallest value 0 to the first color (black) and the last value 1 to the last color (white).
- Values in X are then used to obtain an index into the color map by proportionality.
- If scale is omitted, its default is [min(min(X)), max(max(X))].

Colormaps

- True color or RGB images are created from m x n x 3 arrays containing RGB triplets.
- So $X(i; j, :)$ contains the RGB values that specify the color of pixel P_{ij} , In this case no colormap is needed.
- To display such an image we use

      >> image(X)

Colormaps

- Example:

```
>> load clown          % load a file called clown.mat (the image)
>> who                 % determine the variables associated with clown
     X caption map
>> size(X)             % determine the size of X
     200 320           % this means that clown is indexed
>>X(20,160)
     77                % the color of pixel (20,160) in the clown
                       image is map(77,:))
>>map(77,:)
     .9961 .8672 .7031    % a whitish color
>>image(X); colormap(map)      %display the image of the clown
                               using the colormap "map"
>>colormap(gray)          %change the colormap to black and
                          white.
```

Colors in Matlab

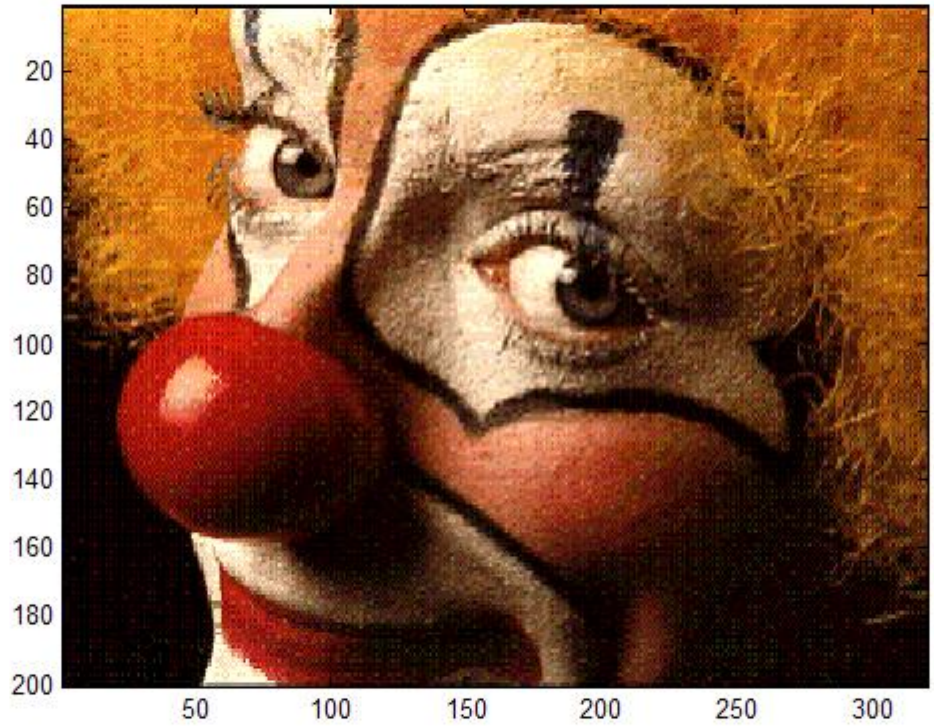
- To determine the color of the (5,5) pixel, we must first determine what X(5,5) is:
  - X(5,5)

    ans = 61
- Then we incorporate that value as a row index into the colormap matrix, map:
  - map(61,:)

    ans = 0.9961 0.5781 0.1250
- The (5,5) pixel has a lot of red, some green, and a little blue

Colors in Matlab

- Displaying the image requires two MATLAB commands, one to create the image and one to set the figure's colormap:
  - image(X)
  - colormap(map)



- Indexed image displayed using image and colormap.

Colors in Matlab

- Example: the built-in color map jet with 16 colors is specified and numerically displayed:

```
>> colormap jet(16)          %specifies the colormap
>> colormap                  %displays the colormap
0 0 0.7500
0 0 1.0000
0 0.2500 1.0000
0 0.5000 1.0000
0 0.7500 1.0000
0 1.0000 1.0000
0.2500 1.0000 0.7500
0.5000 1.0000 0.5000
0.7500 1.0000 0.2500
1.0000 1.0000 0
1.0000 0.7500 0
1.0000 0.5000 0
1.0000 0.2500 0
1.0000 0 0
0.7500 0 0
0.5000 0 0
```

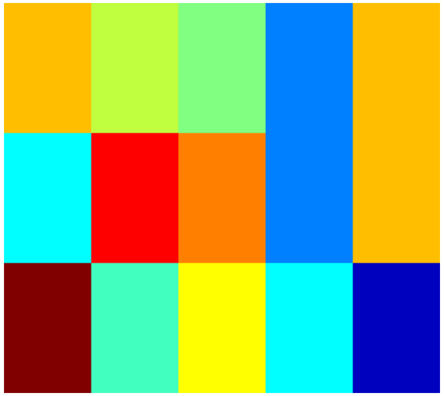
Colors in Matlab

- Now if I have an array *A* each of whose entries is an integer from 1 to 16 and use the MATLAB command

```
>> image(A)
```

each element of *A* will be associated with a color according to its value.

```
>> A=ceil(16*rand(3,5));        %entries are integers between 1 and 16

>> A

A=

        11  9  8  4  11
        6  14  12  4  11
        16  7  10  6  1

>> image(A);

>> colorbar        %displays the colormap
```



Colors in Matlab

- The MATLAB command

  >>imagesc(A)

  will "scale" the values in the matrix *A* so that they range from the lowest-number color to the highest-number color.

- For example:

  >> A=rand(3,5);

  >>A

  A=

0.9925	0.2476	0.6391	0.0665	0.0594
0.3879	0.0504	0.2129	0.5469	0.6470
0.2991	0.5542	0.2264	0.3505	0.9731

  >> imagesc(A);

  >> colorbar

END