# Chapter 6
# Color Image Processing

- Color is a powerful descriptor
- Human can discern thousands of color shades.
- "color" is more pleasing than "black and white“.
- Full Color: color from full-color sensor, *i.e.*, CCD camera
- Pseudo color: assign a color to a particular monochromatic intensity.

# 6.1 Color Fundamentals

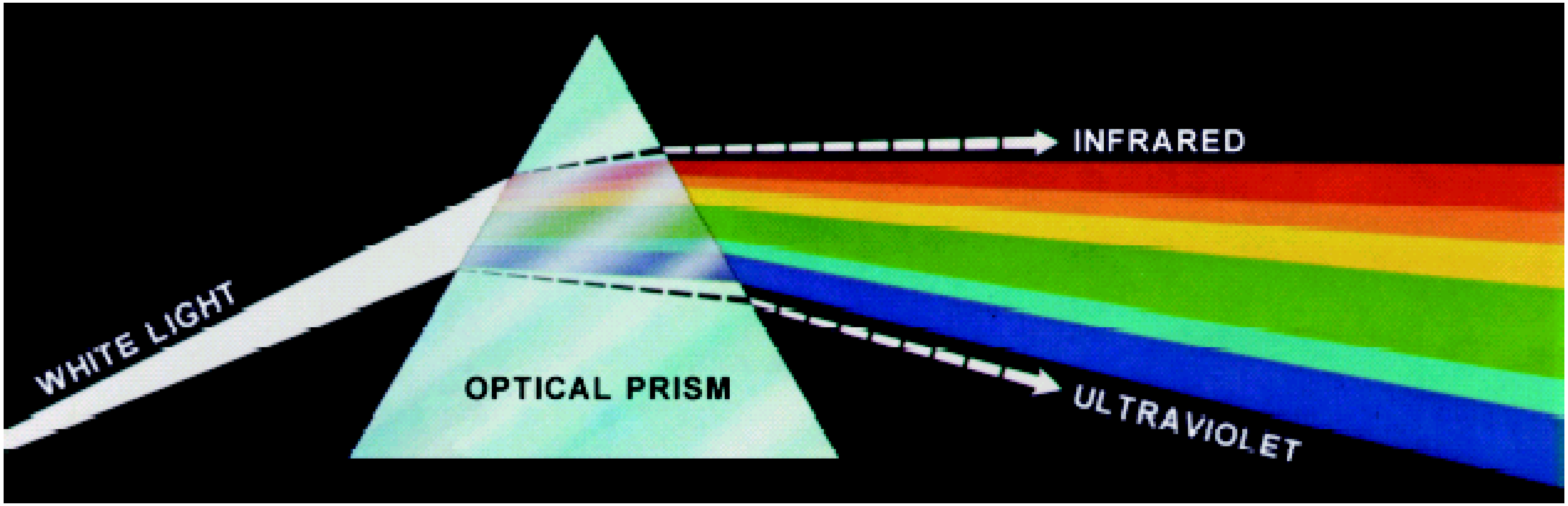
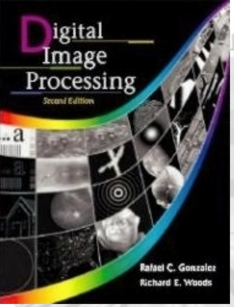



**FIGURE 6.1** Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)




# 6.1 Color Fundamentals

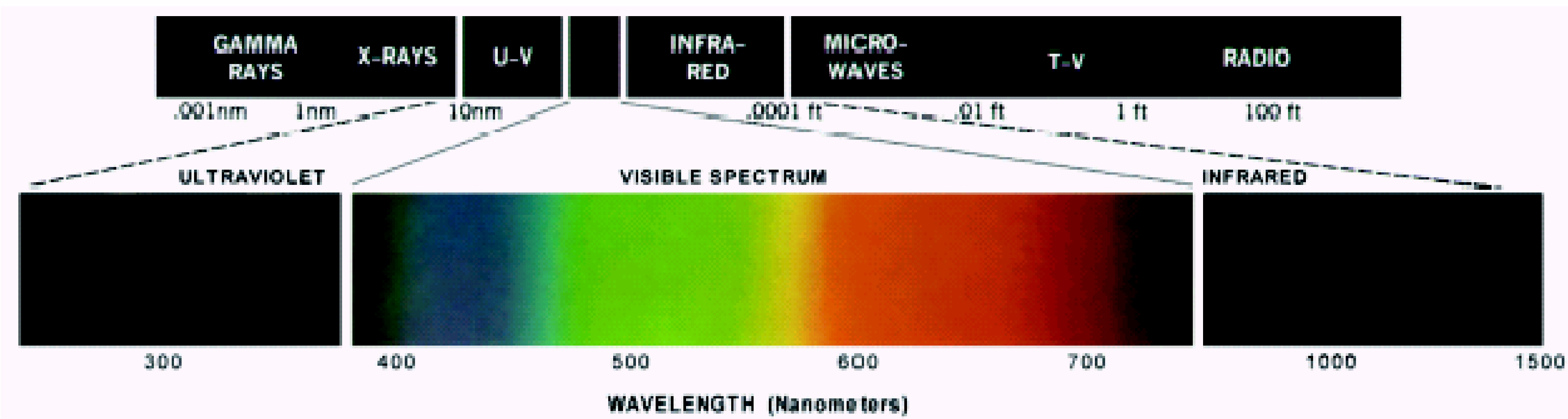
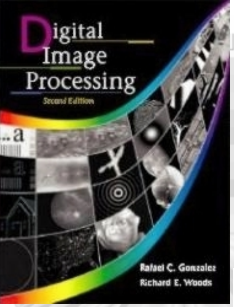



**FIGURE 6.2** Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)




# 6.1 Color Fundamentals

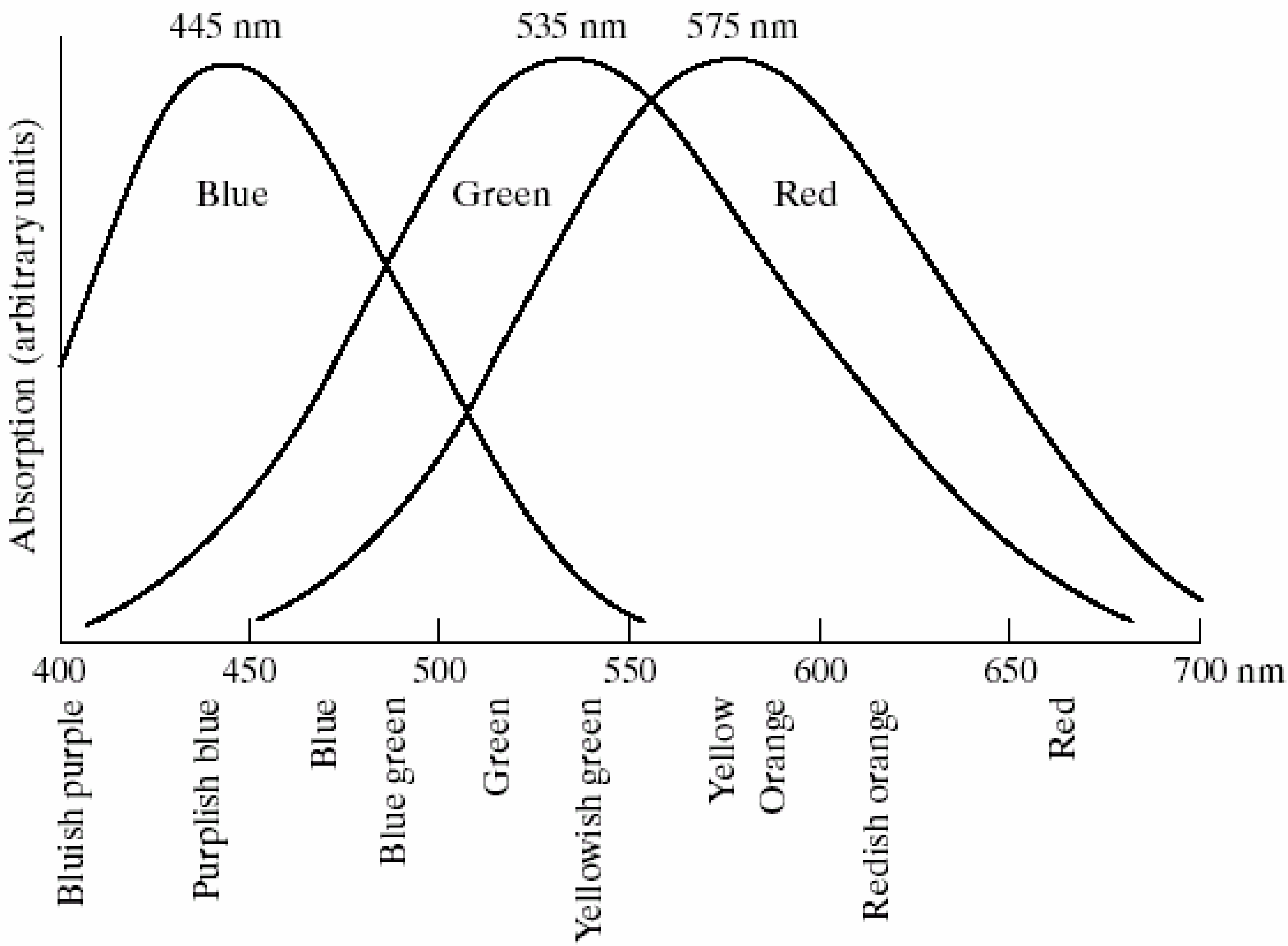
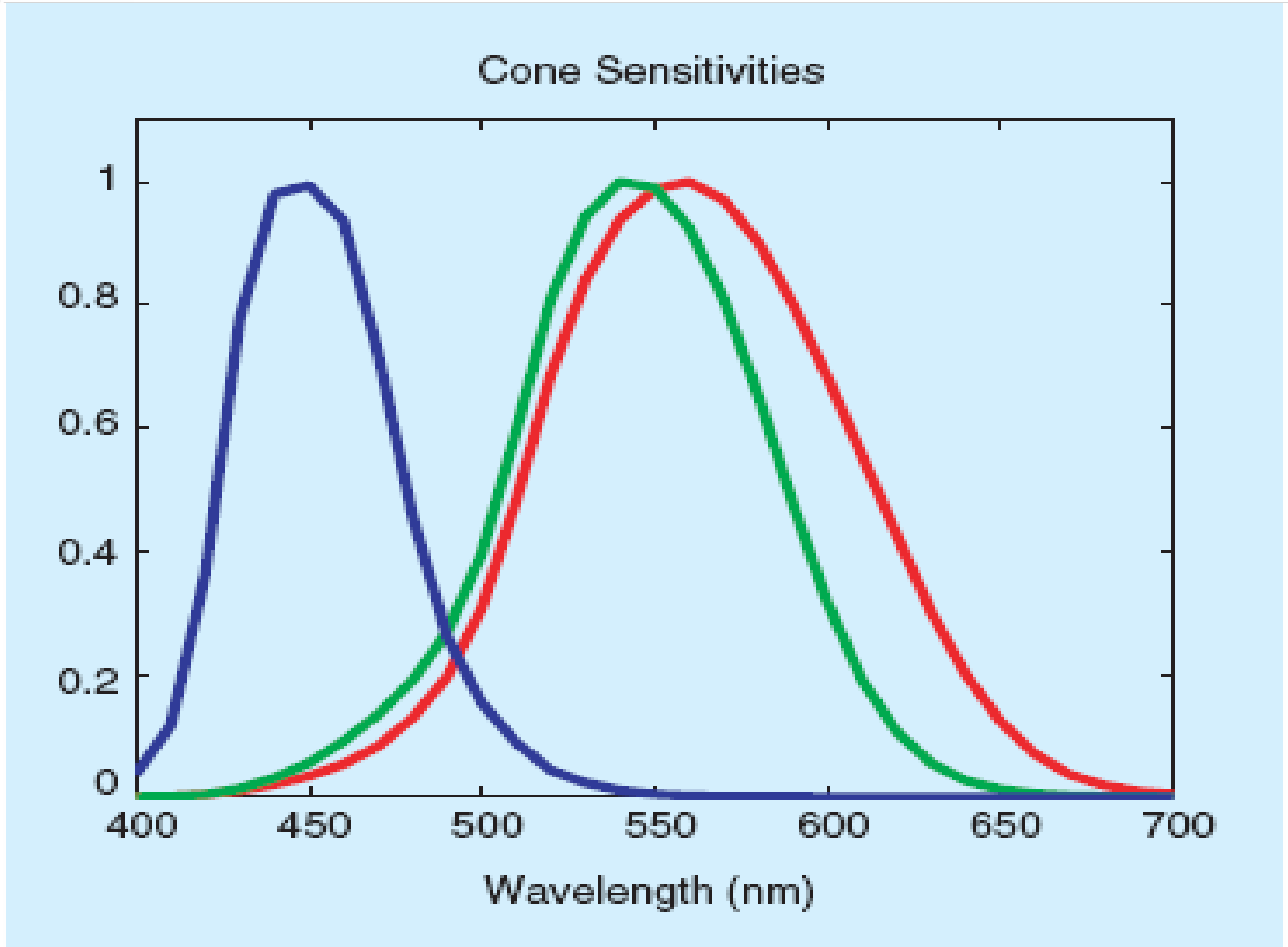



**FIGURE 6.3** Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

# 6.1 Color Fundamentals



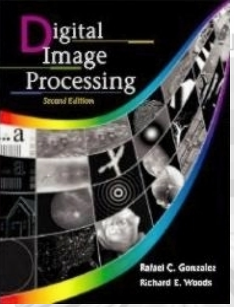

Cone sensitivity

# 6.1 Color Fundamentals

- The colors that humans perceive of an object are determined by the nature of the light reflected from the object.
- Incident light (electromagnetic wave) → human eye
- The light is visible to human eyes if its wavelength is between 380-780 (nm). Human eyes have the following sensitivity :

  1. Brightness : light intensity (energy)

  2. Color : different spectral composition




# 6.1 Color Fundamentals

- how to specify color ?

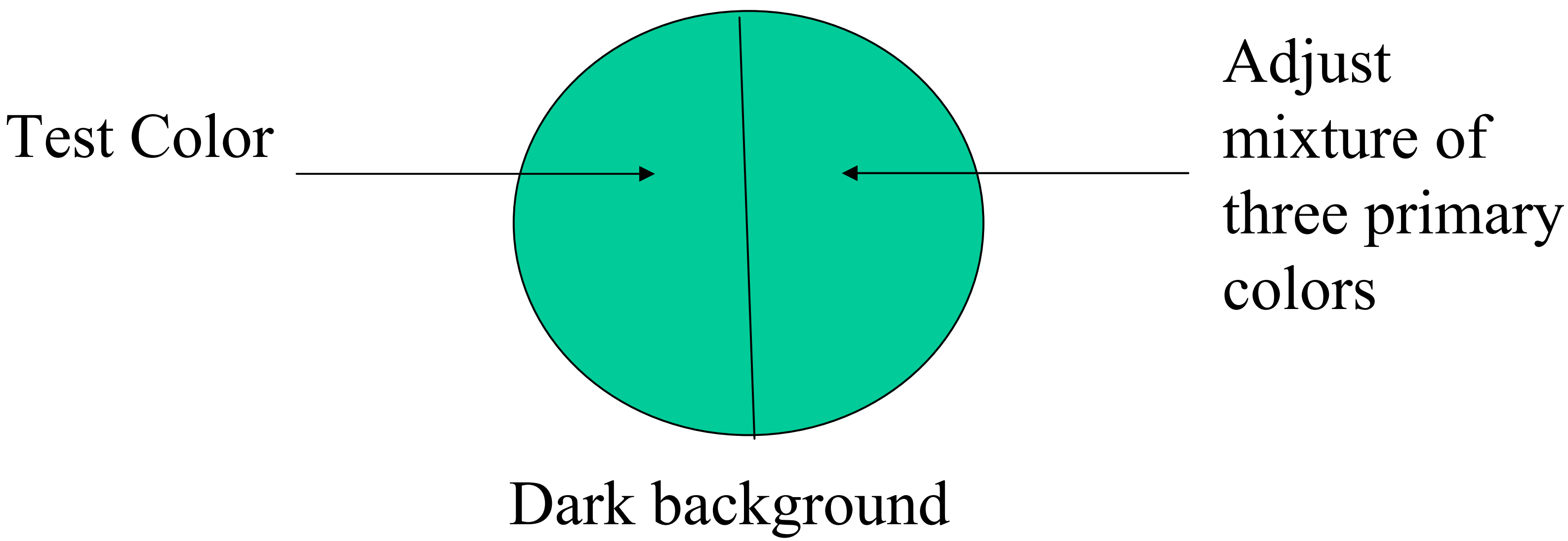
  (1) color matching

  (2) color difference

  (3) color appearance

# 6.1 Color Fundamentals

- **Color Mixture**
- light of any color can be synthesized by an approximation mixture of three primary colors
- Maxwell (1855) provided "colorimetry"



# 6.1 Color Fundamentals

- ***Tristimulus values*** of a test color are the amounts of three primary colors required to give a match by additive mixture.
- Two rules of colorimetry : $\begin{cases} \text{linearity} \\ \text{additivity} \end{cases}$

# 6.1 Color Fundamentals

- linearity:

$$\text{If } S_1(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_2(\lambda) \text{ then } aS_1(\lambda) \xleftrightarrow[\text{match}]{\text{color}} aS_2(\lambda)$$

- additivity :

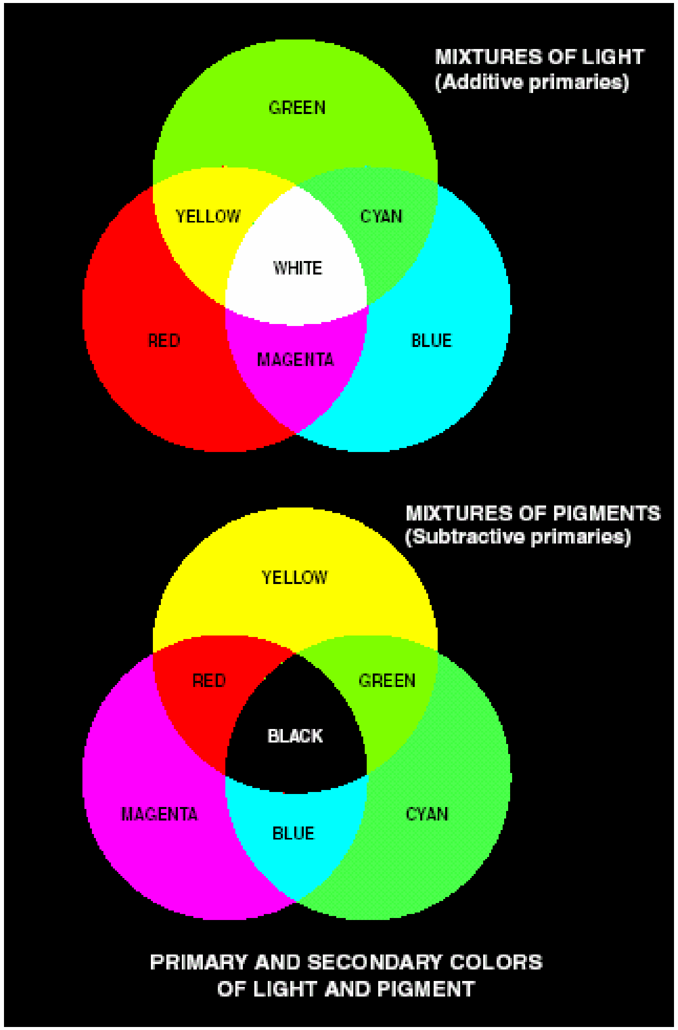
$$\text{If } S_1(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_2(\lambda) \text{ and } S_3(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_4(\lambda)$$

$$\text{then } S_1(\lambda)+S_3(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_2(\lambda)+S_4(\lambda)$$

- Color with negative tri-stimulus values:

$$\text{test color } S \xleftrightarrow[\text{match}]{\text{color}} aR(\lambda) - bG(\lambda)+cB(\lambda)$$

# 6.1 Color Fundamentals

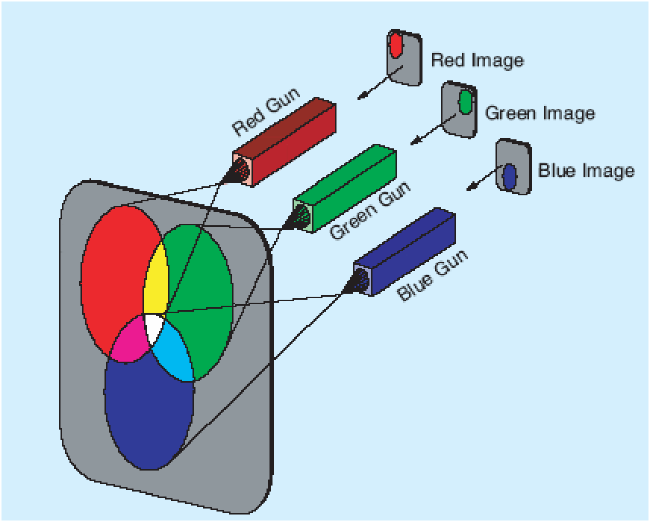


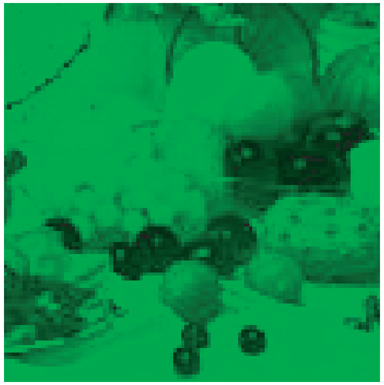

a
b

**FIGURE 6.4** Primary and secondary colors of light and pigments. (Courtesy of the General Electric Co., Lamp Business Division.)

# 6.1 Color Fundamental

- Additive Color System
- Primary: RGB

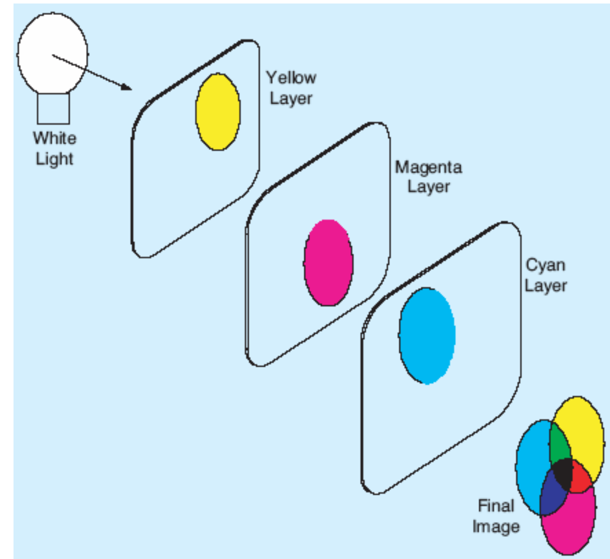


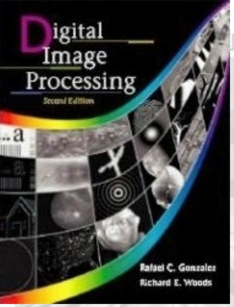
Red + Green + Blue = Color

# 6.1 Color Fundamental

- Subtractive Color System:
- Primary: CMY




White − Yellow − Cyan − Magenta = Color



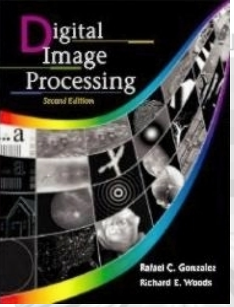



# 6.1 Color Fundamentals

- ***Color matching function*** : the tristimulus values of the *spectral color* with unit intensity light of single wavelength.

- The primary colors are the spectral color of wavelength:

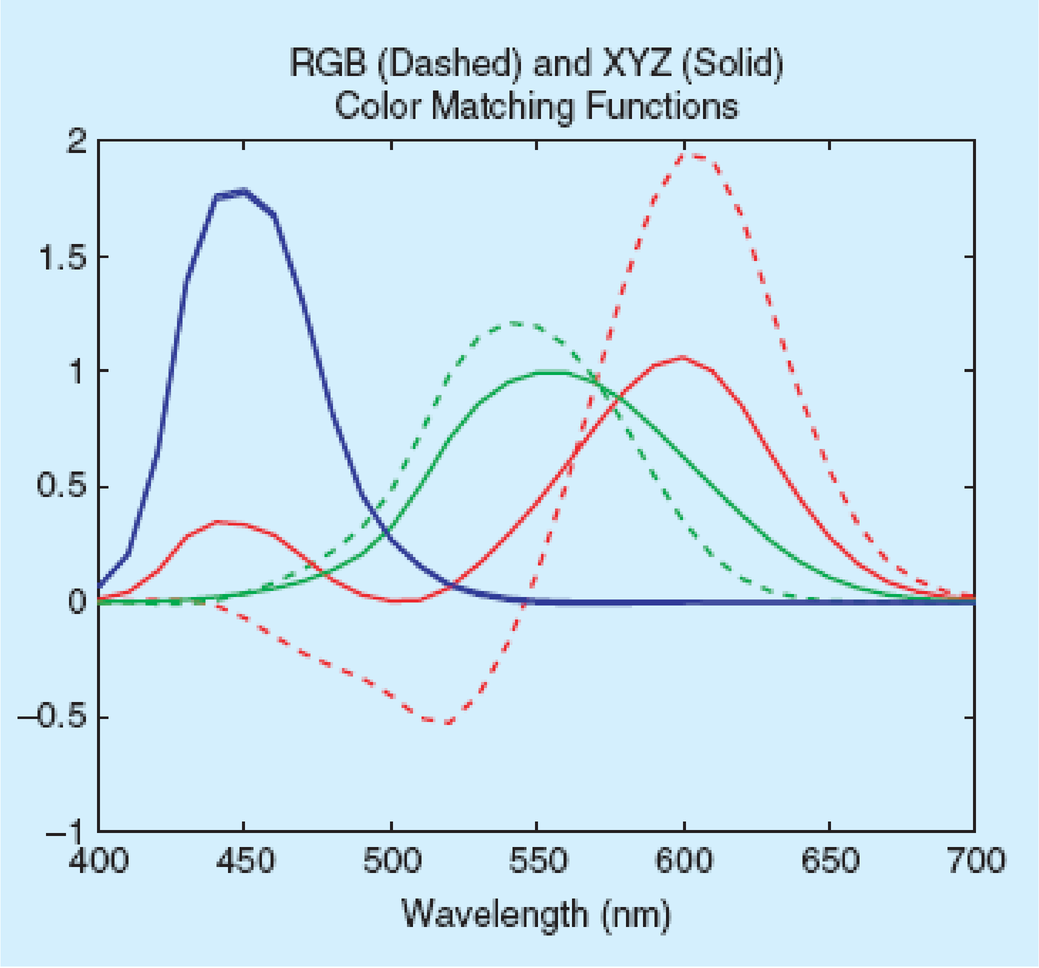
$$\begin{cases} 700.0(R_0) \\ 546.1(G_0) \\ 435.8(B_0) \end{cases}$$






# 6.1 Color Fundamentals

- CIE RGB and XYZ ***color matching functions***: RGB is shown in dashed lines, and XYZ are shown in solid lines.



# 6.1 Color Fundamentals

- Any color S( λ ) can be derived as the color sensitivity summation as

$$S(\lambda)d\lambda = R_s(\lambda)d\lambda + G_s(\lambda)d\lambda + B_s(\lambda)d\lambda$$

$$R_s = \int_\lambda R_s(\lambda)d\lambda$$

$$G_s = \int_\lambda G_s(\lambda)d\lambda \quad R_s, G_s, B_s : \text{tristimulus values of components}$$

$$B_s = \int_\lambda B_s(\lambda)d\lambda$$

Using ***color matching function*** $r(\lambda),\ g(\lambda),\ b(\lambda)$

$$R_s = \int S(\lambda)r(\lambda)d\lambda,\ \ G_s = \int S(\lambda)g(\lambda)d\lambda,\ \ B_s = \int S(\lambda)b(\lambda)d\lambda$$

# 6.1 Color Fundamentals

- Color matches between $S_1 \leftrightarrow S_2$

$$R_1 = \int S_1(\lambda) r(\lambda) d\lambda = \int S_2(\lambda) r(\lambda) d\lambda = R_2$$

$$G_1 = \int S_1(\lambda) g(\lambda) d\lambda = \int S_2(\lambda) g(\lambda) d\lambda = G_2$$

$$B_1 = \int S_1(\lambda) b(\lambda) d\lambda = \int S_2(\lambda) b(\lambda) d\lambda = B_2$$

- metamer : $S_1(\lambda) \neq S_2(\lambda)$, $\quad S_1(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_2(\lambda)$
- isomer: $S_1(\lambda) = S_2(\lambda)$ : the same spectral distribution
- ***Color matching function*** are averaged for people with normal color vision.
- Color matching normally depends on the conditions of observation and previous exposure of eyes.

# 6.1 Color Fundamentals - Color Coordinate System Transformation

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \xrightarrow{normalization} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \qquad r+g+b=1$$

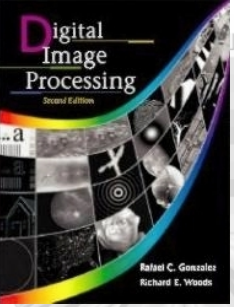
$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B} \qquad b = \frac{B}{R+G+B}$$

where $r+g+b=1$ $\rightarrow$reduced to 2-D color information -→ chromaticity

The 3rd information is the luminance

# 6.1 Color Fundamentals - Color Coordinate System Transformation

- Y(luminance) $\rightarrow$ The 3rd-dimension information
- Luminance (Brightness) sensor
- Different wavelengths contribute different brightness to the sensor
- The relative brightness response for the eye is termed "relative luminous efficiency" $y(\lambda)$
- $y(\lambda)$ is obtained by photometric matches (matching of brightness)




# 6.1 Color Fundamentals - Color Coordinate System Transformation

- The luminance of any spectral distribution S( $\lambda$ ) is

$$Y = k_m \int S(\lambda) y(\lambda) d\lambda$$

where $k_m = 680\ lumens/watt \qquad 1\ lumen = candelas/m^2$

- Brightness match

$$\int S_1(\lambda) y(\lambda) d\lambda = \int S_2(\lambda) y(\lambda) d\lambda$$

$$S_1(\lambda) \neq S_2 \qquad \text{or} \qquad \int S_1(\lambda) d\lambda \neq \int S_2(\lambda) d\lambda$$

# 6.1 Color Fundamentals-Standard CIE Color System

- The tristimulus values for two color-matched colors are different for different observers.
- Standard Observer : by averaging the color matching data of a large number of color normal observers.
- 1931, CIE defined standard observer which consists of color matching functions for primary stimuli of wavelengths: $700(R_0),\ \ 546.1(G_0),\ \ 435.8(B_0)$
- Unit normalized $\Rightarrow$ equal amount of three primaries are required to match the light from equal energy illumination energy.

# 6.1 Color Fundamentals-Standard CIE Color System

- CIE also define three new primaries : X, Y, Z

$$\begin{cases} X = 2.365R_0 - 0.515G_0 + 0.005B_0 \\ Y = -0.897R_0 + 1.426G_0 - 0.014B_0 \\ Z = -0.468R_0 + 0.089G_0 + 1.009B_0 \end{cases} \quad ..(a)$$

- By matrix inversion, we obtain

$$R_0 = 0.490X + 0.177Y$$

$$G_0 = 0.310X + 0.813Y + 0.01Z$$

$$B_0 = 0.200X + 0.010Y + 0.990Z \qquad ..(b)$$

- Y tristimulus value corresponds to the luminance normalized.

# 6.1 Color Fundamentals-Standard CIE Color System

- The tristimulus values and color-matching function are always positive primaries; X, Y, Z are non-real (cannot be realized by actual color stimuli)
- *Normalized tristimulus values*: X, Y, Z→chromaticity

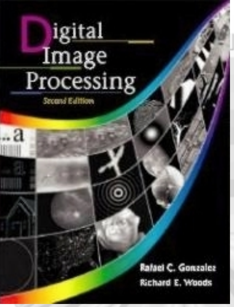
$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

$$\text{color}\begin{cases}\text{chromaticity} & x, y \\ \text{luminance} & Y\end{cases}$$

- $x$ : red light → orange, reddish-purple
- $y$ : green light → bluish-green, yellowish-green.
- small $x$, $y$ : blue light → violet or purple




# 6.1 Color Fundamentals-Standard CIE Color System

- Chromaticity diagram : $(r_0, g_0)$ and $(x, y)$
- Pure spectral colors are plotted on the elongated horseshoe-shaped curve called the spectral locus.
- *line of purples* : straight line consists of two extremes of the spectral locus
- chromaticity diagram $\neq$ color matching function

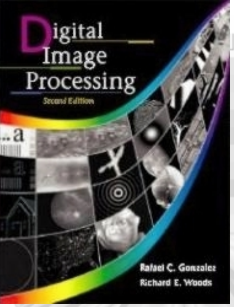
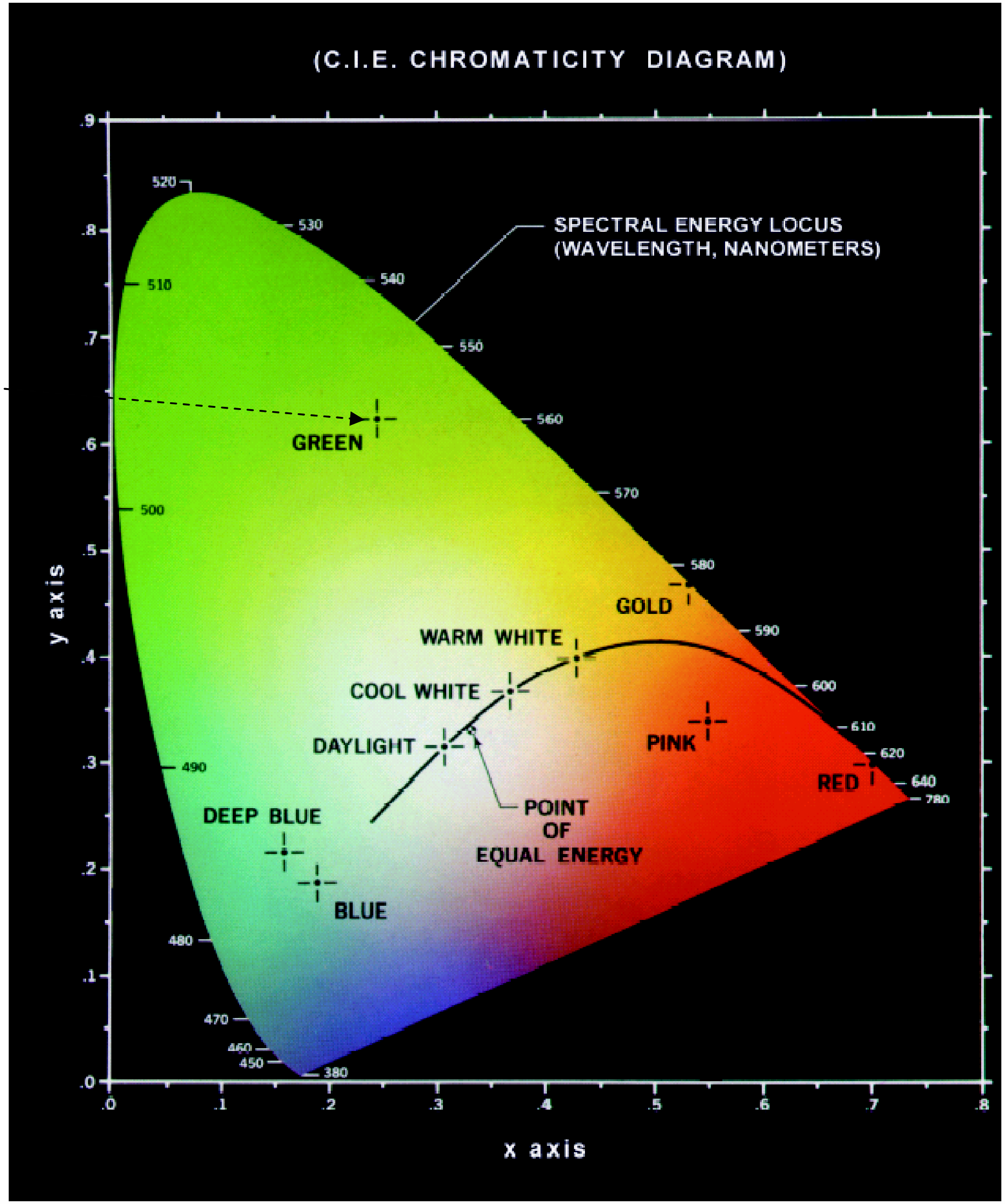



**FIGURE 6.5** Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)

$y$ = 62% green
$x$ = 25% red
$z$ = 13% blue

# 6.1 Color Fundamentals



# 6.1 Color Fundamentals

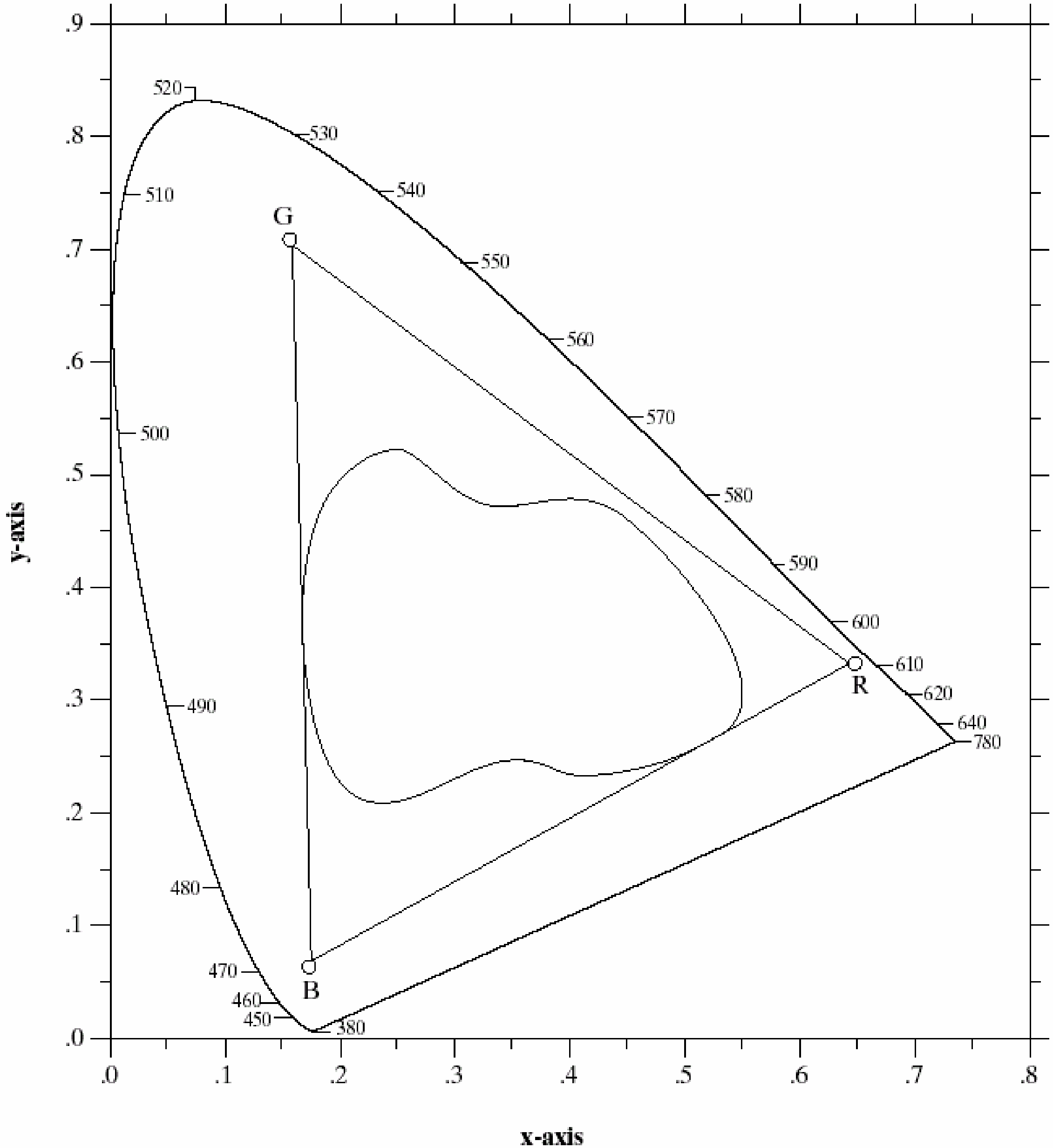
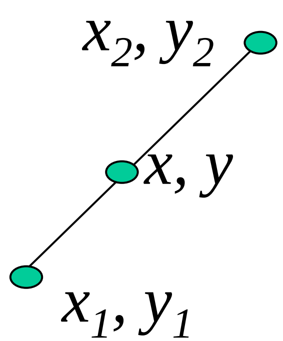



**FIGURE 6.6** Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

# 6.1 Color Fundamentals --Color Mixtures

**Grassman's Law :**

- The tristimulus values of a color mixture are obtained by the vector addition of the tristimulus values of the components of the mixture
- If colors: $S_1=(X_1, Y_1, Z_1)$ and $S_2=(X_2, Y_2, Z_2)$ are mixed as $S=(X, Y, Z)$ then $X=X_1+X_2$ $Y=Y_1+Y_2$ $Z=Z_1+Z_2$
- If colors: $S_1=(x_1, y_1, Y_1)$ and $S_2=(x_2, y_2, Y_2)$ are mixed as $S=(x, y, Y)$ then



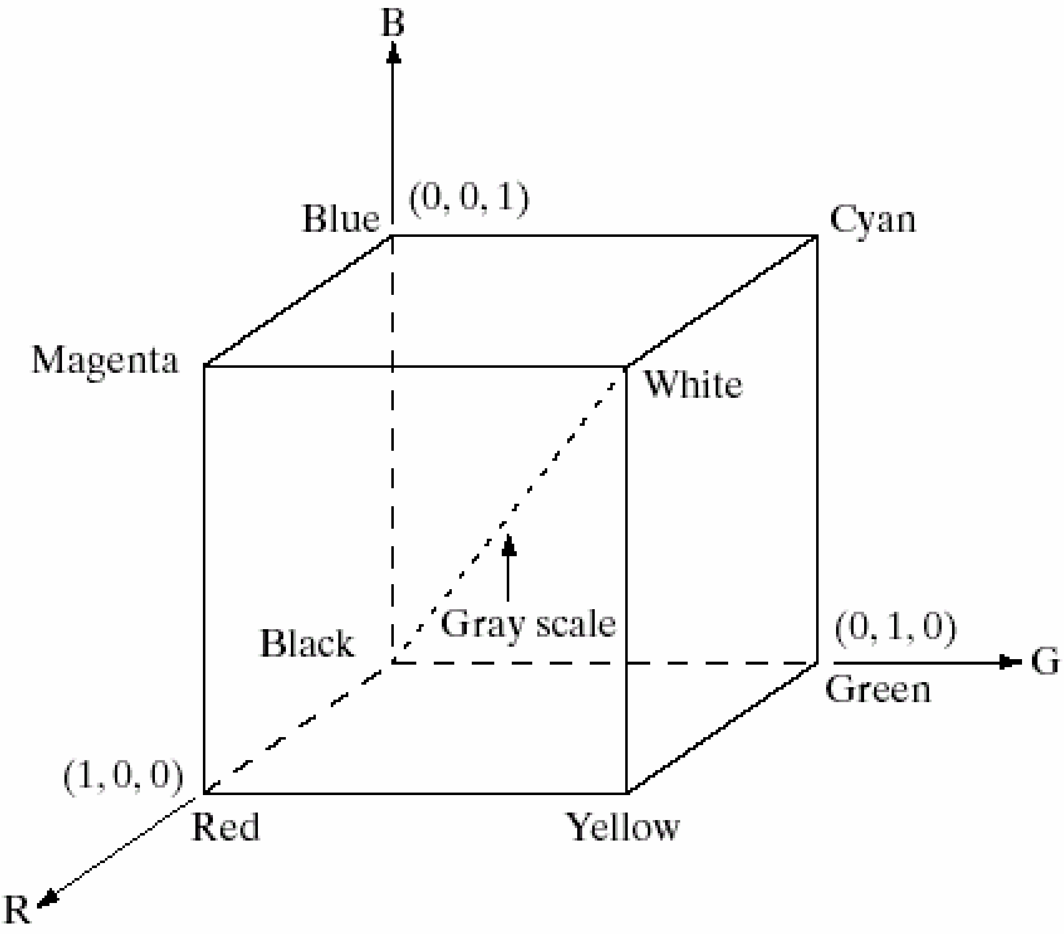

$$x = \frac{x_1(Y_1/y_1) + x_2(Y_2/y_2)}{(Y_1/y_1) + (Y_2/y_2)} \qquad y = \frac{Y_1 + Y_2}{(Y_1/y_1) + (Y_2/y_2)}$$

# 6.2 Color Models

- The color model (color space or color system) is to facilitate the specification of colors in some standards.
- Color model is a specification of a **coordinate system** and a subspace within the system where a color is represented.
- RGB for color monitor.
- CMY (cyan, magenta, yellow) for color printing.
- HIS (hue, intensity and saturation): decouple the color and gray-scale information.

# 6.2 Color Models

**FIGURE 6.7** Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).



# 6.2 Color Models

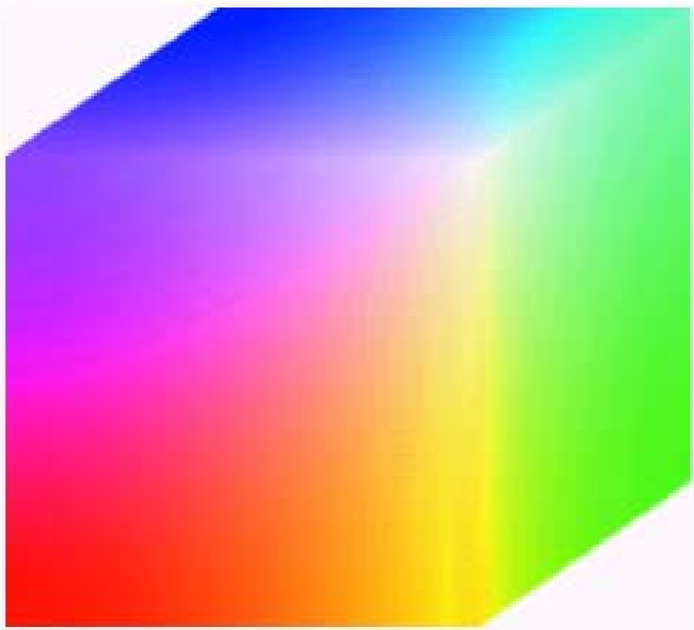
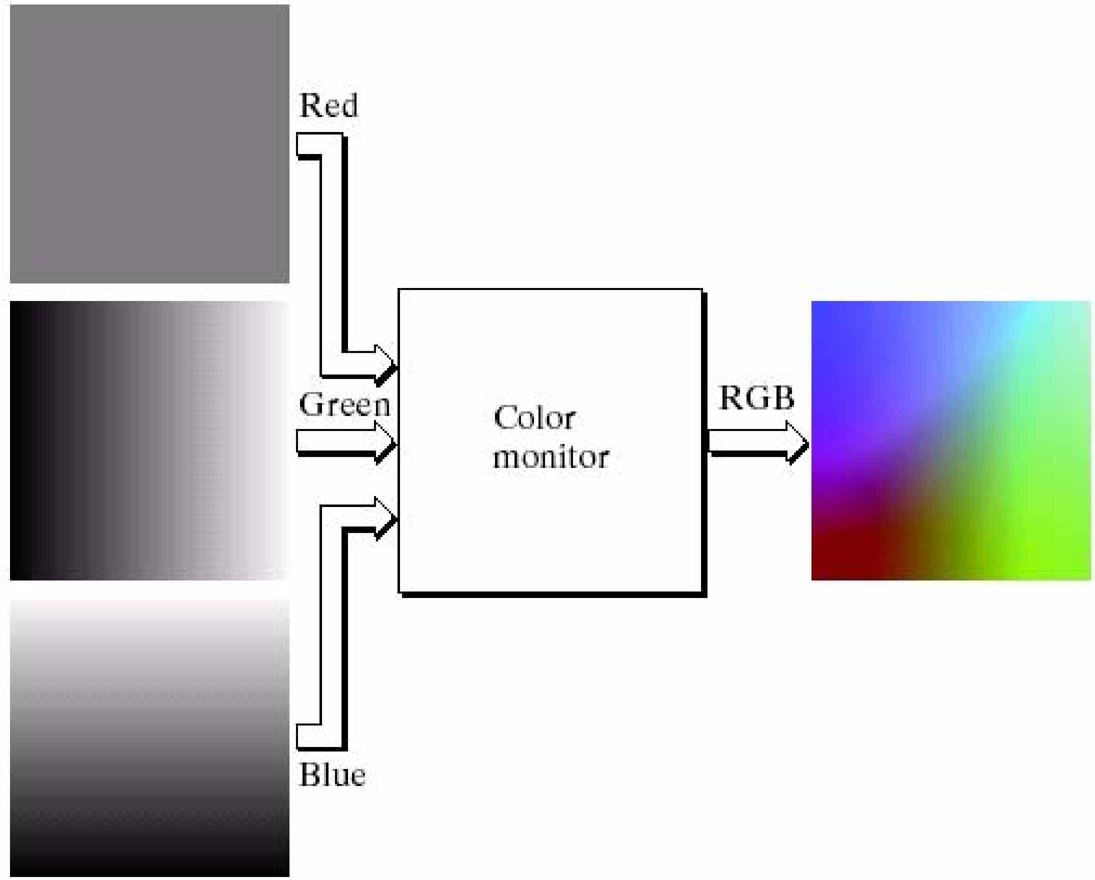


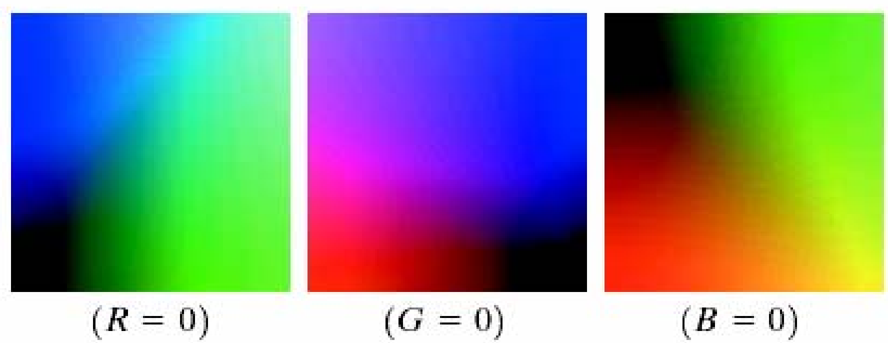
**FIGURE 6.8** RGB 24-bit color cube.

a
b

**FIGURE 6.9**
(a) Generating the RGB image of the cross-sectional color plane (127, $G$, $B$).
(b) The three hidden surface planes in the color cube of Fig. 6.8.

# 6.2 Color Models



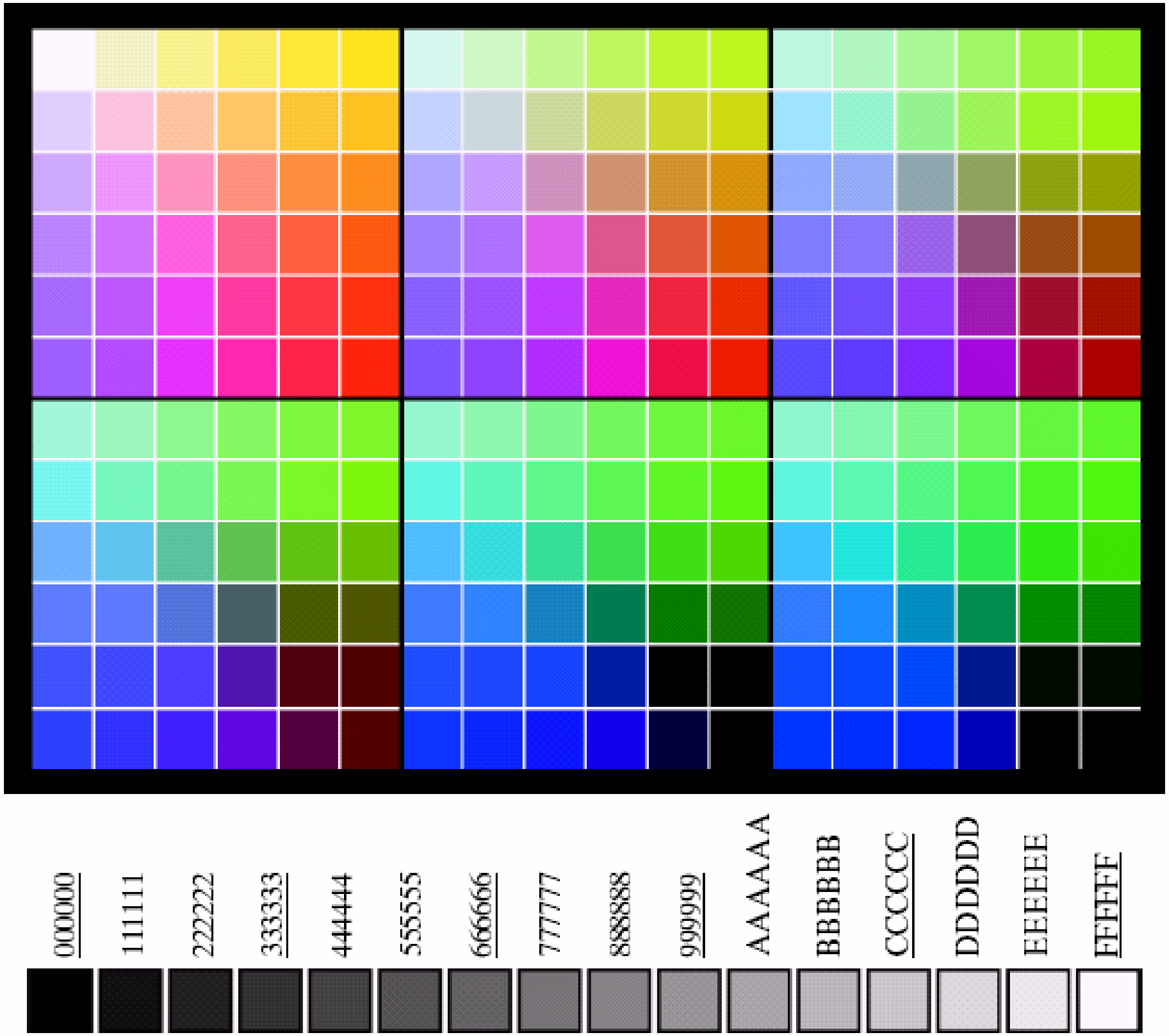
# 6.2 Color Models

- Safe RGB colors (or all-system safe color, safe web color): a subset of colors that are likely to be reproduced faithfully reasonably independently of viewers hardware capability.
- 216 colors = $6\times6\times 6$
- 6 levels in R, G, and B: in decimal: 0, 51, 102, 153, 204, or 255.
- In hex: 00, 33, 66, 99, CC, FF

# 6.2 Color Models

| 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.




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).

# 6.2 Color Models

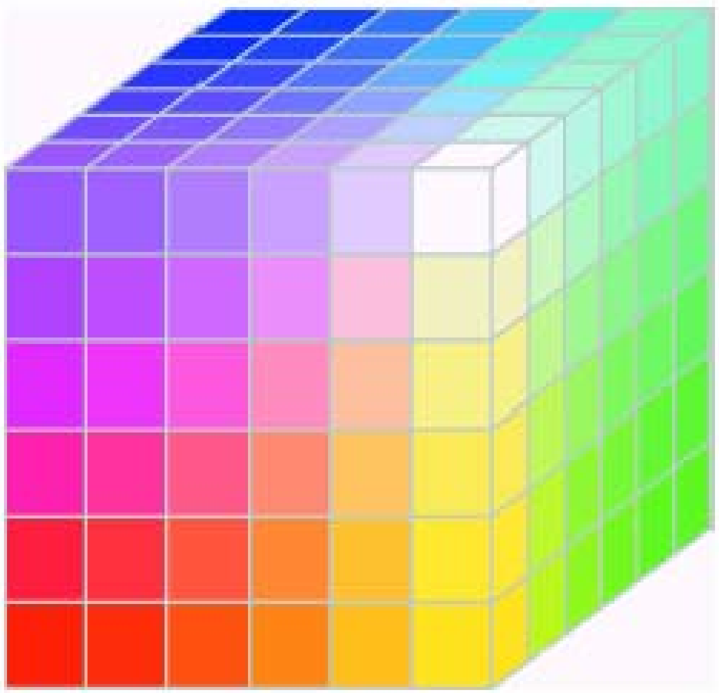


FIGURE 6.11 The RGB safe-color cube.

# 6.2 Color Models

- RGB to CMY conversion

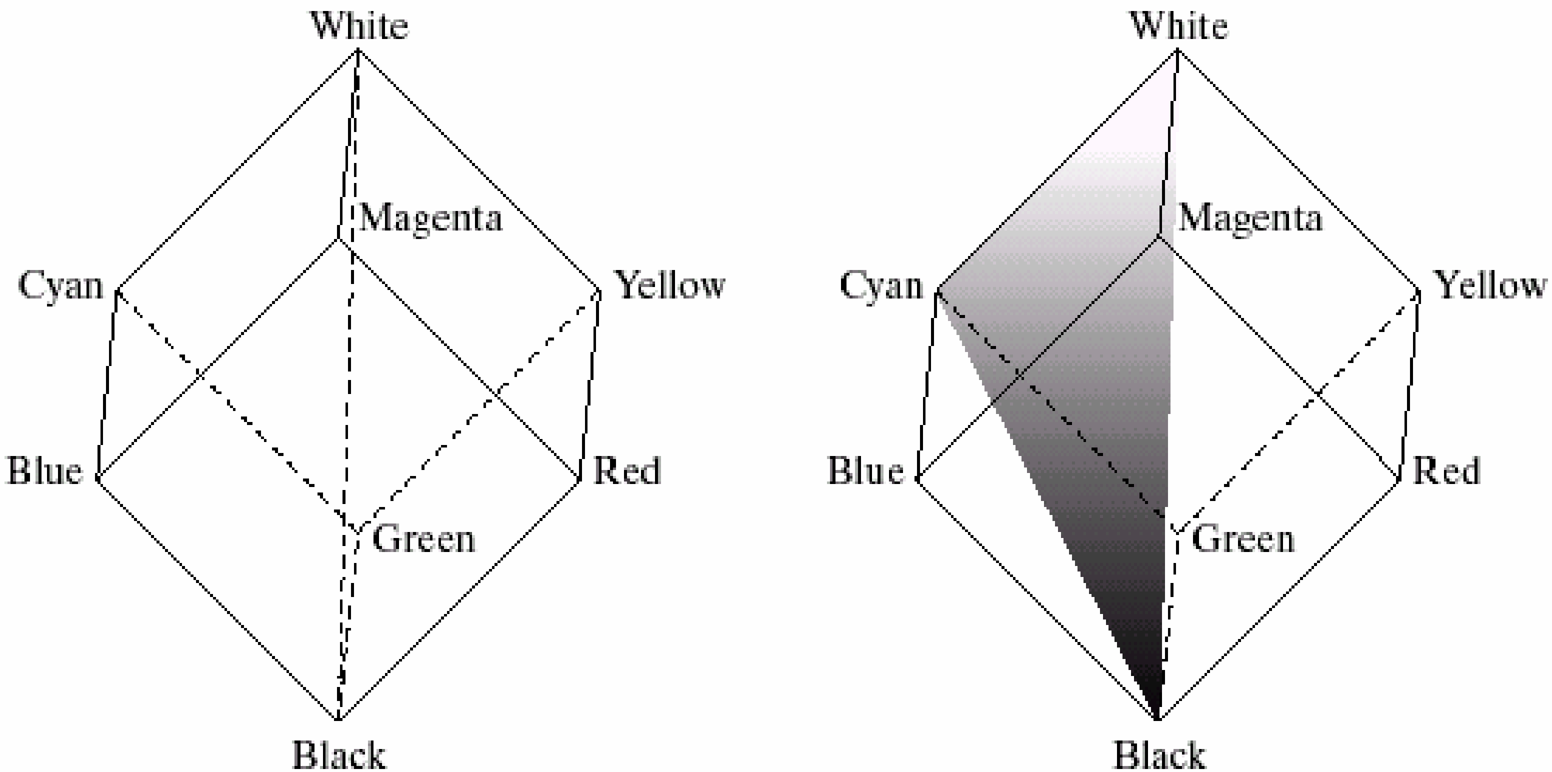
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Instead of adding C,M, and Y to produce black, a fourth color *black* is added

# 6.2 Color Models

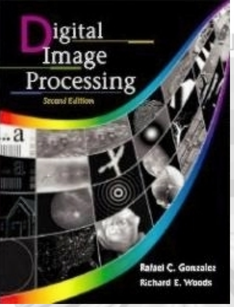
- Human describes color in terms of hue saturation and brightness.
- Hue: describe the pure color, pure yellow, orange, green or red.
- Saturation measures the degree to which a pure color is diluted by white light.
- Brightness is a subjective descriptor difficult to be measured.

# 6.2 Color Models




All pointes contained in the plane segment define by the intensity and boundary of the cube have the same hue

**FIGURE 6.12** Conceptual relationships between the RGB and HSI color models.



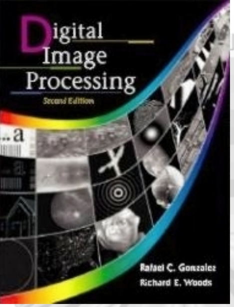

# 6.2 Color Models - Converting colors

- From *RGB* to *HSI*

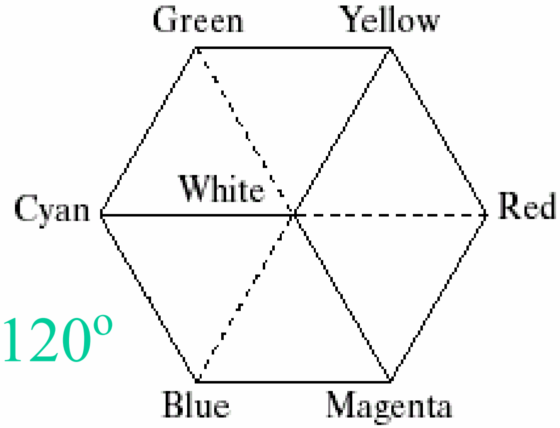
$$H = \begin{cases} \theta & if \quad B \le G \\ 360 - \theta & if \quad B > G \end{cases}$$

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

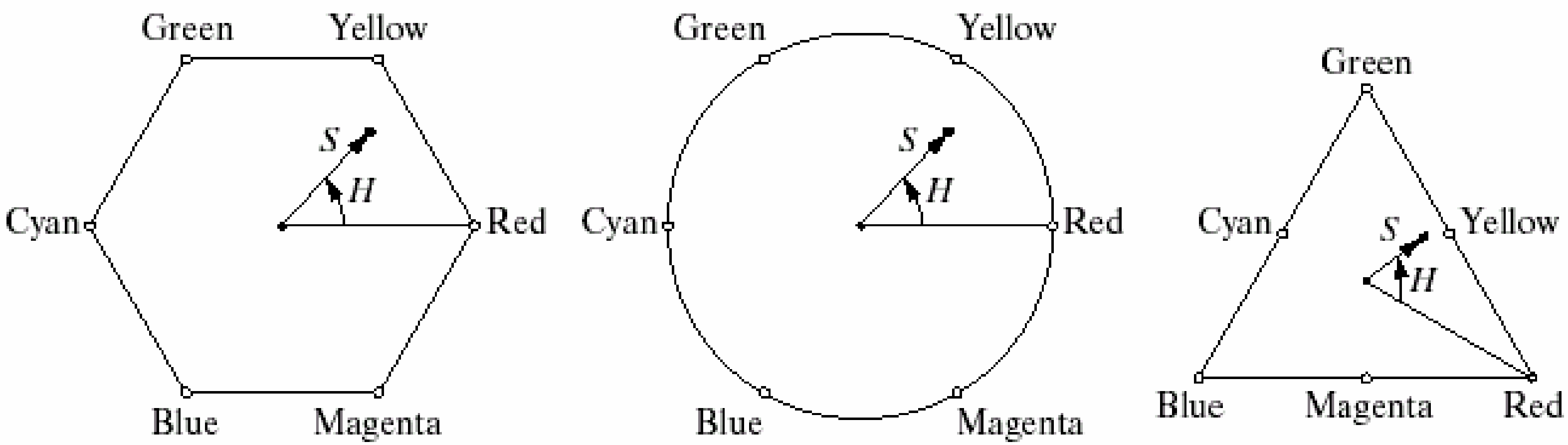
- $S = 1 - [3/(\mathrm{R}+\mathrm{G}+\mathrm{B})][min(\mathrm{R}, \mathrm{G}, \mathrm{B})]$
- $I = (\mathrm{R}+\mathrm{G}+\mathrm{B})/3$




# 6.2 Color Models




Primary colors are separated by 120°




a
b c d

**FIGURE 6.13** Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

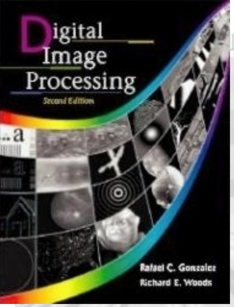
## 6.2 Color Models - Converting colors

- From *HSI* to *RGB*
- *RG sector* $(0 \le H < 120)$, $min(R, G, B) = B$

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

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

$$G = 3I - (R + B)$$

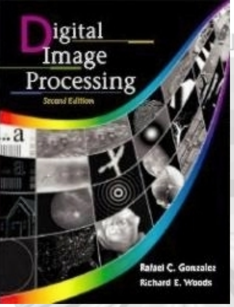




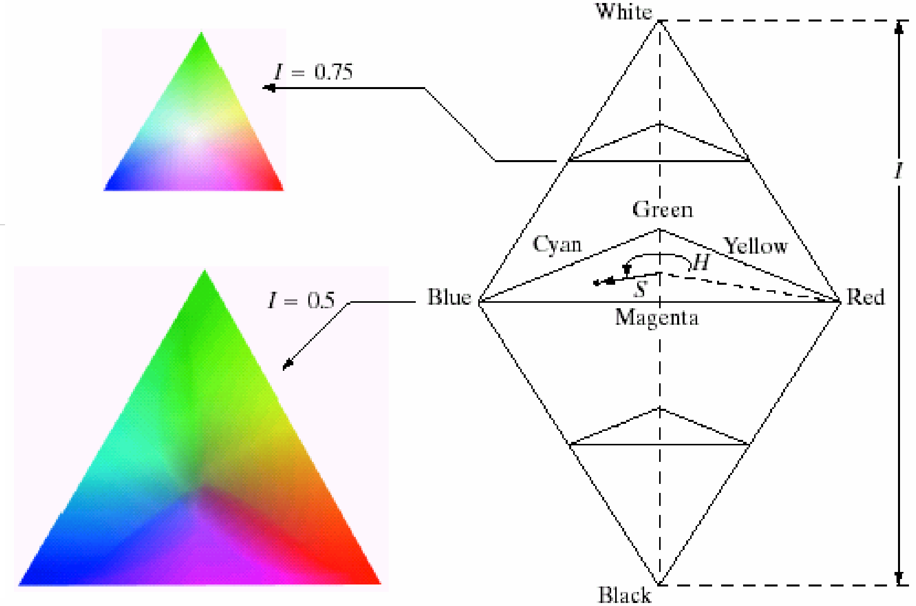

# 6.2 Color Models - Converting colors

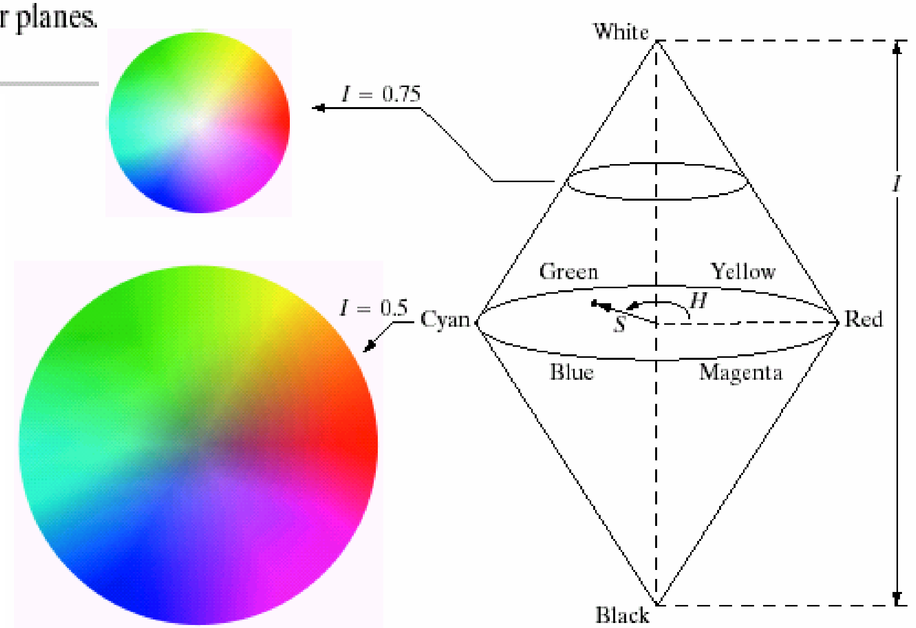
- *GB sector* $(120 \le H < 240)$
- $\min(R, G, B) = R$
- $H = H - 120$
- $R = I(1-S)$
- $$G = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$
- $B = 3I - (R+G)$

- *BR sector* $(240 \le H < 360)$
- $\min(R, G, B) = G$
- $H = H - 240$
- $G = I(1-S)$
- $$B = I\left[1 + \frac{S\cos H}{\cos(60^o - H)}\right]$$
- $R = 3I - (G+B)$




# 6.2 Color Models

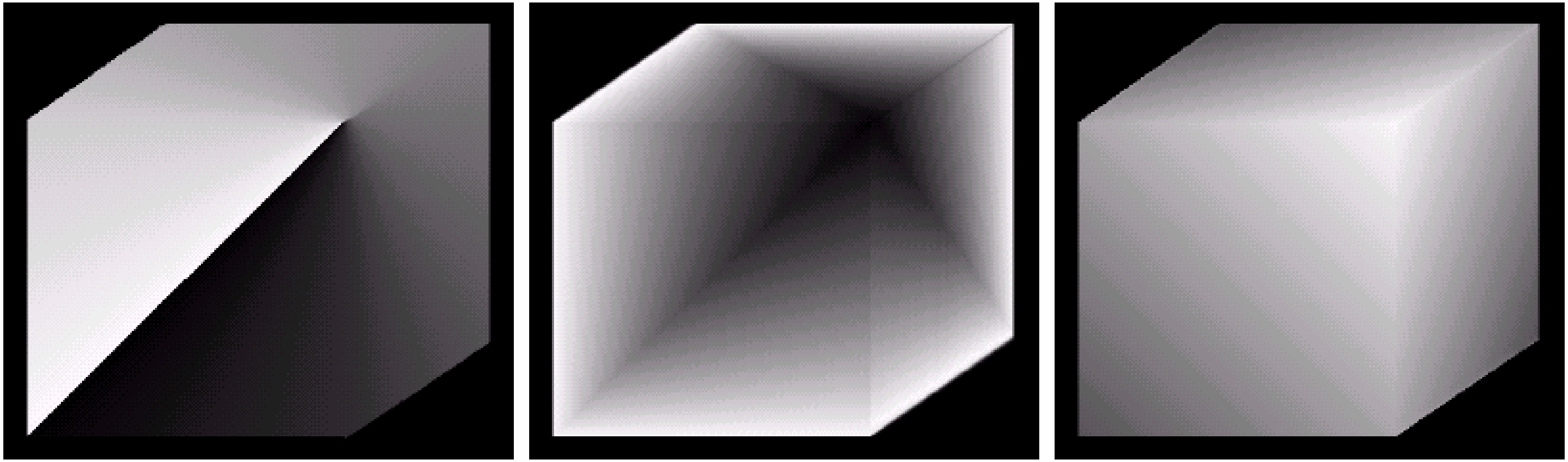



a
b

**FIGURE 6.14** The HSI color model based on (a) triangular and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis.

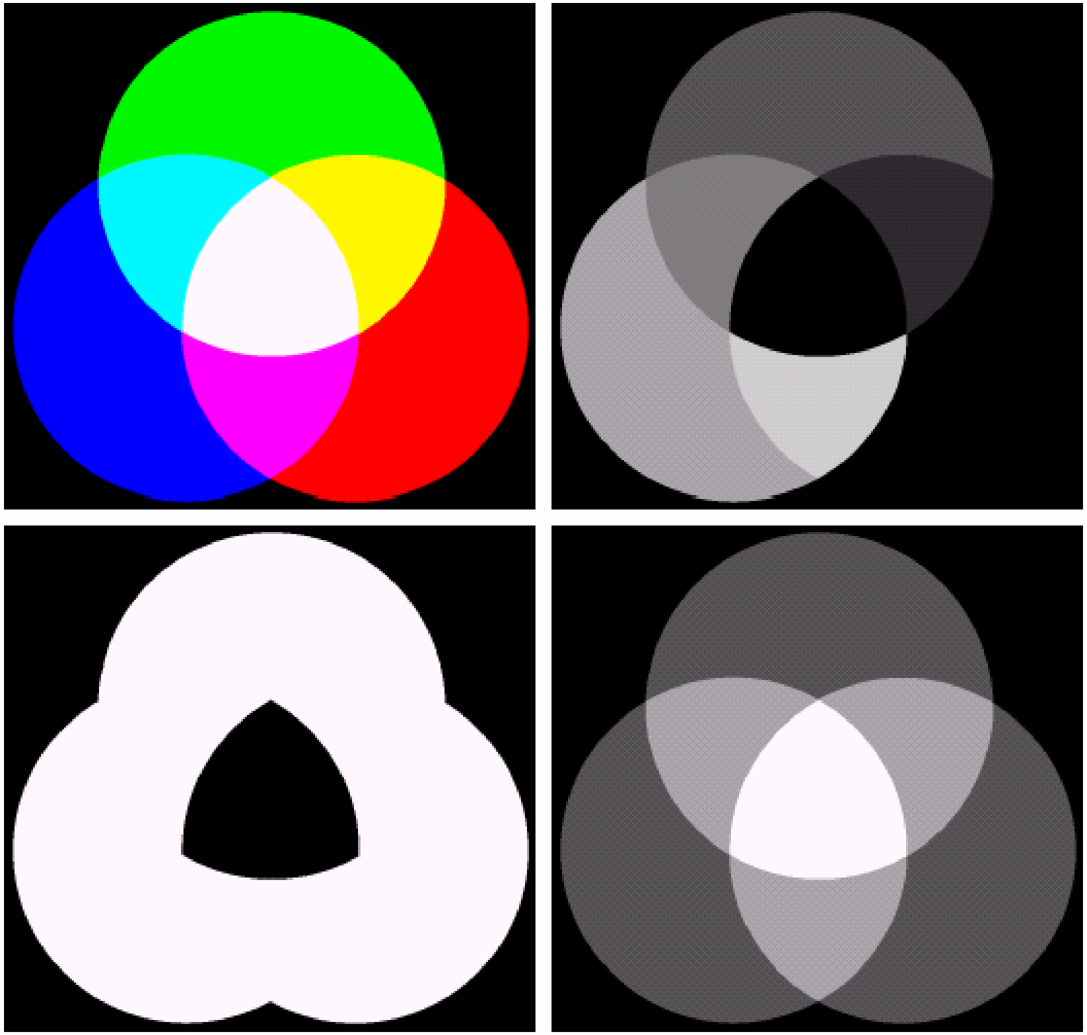
# 6.2 Color Models



a b c

**FIGURE 6.15** HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.

# 6.2 Color Models



a b
c d

**FIGURE 6.16** (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

# 6.2 Color Models

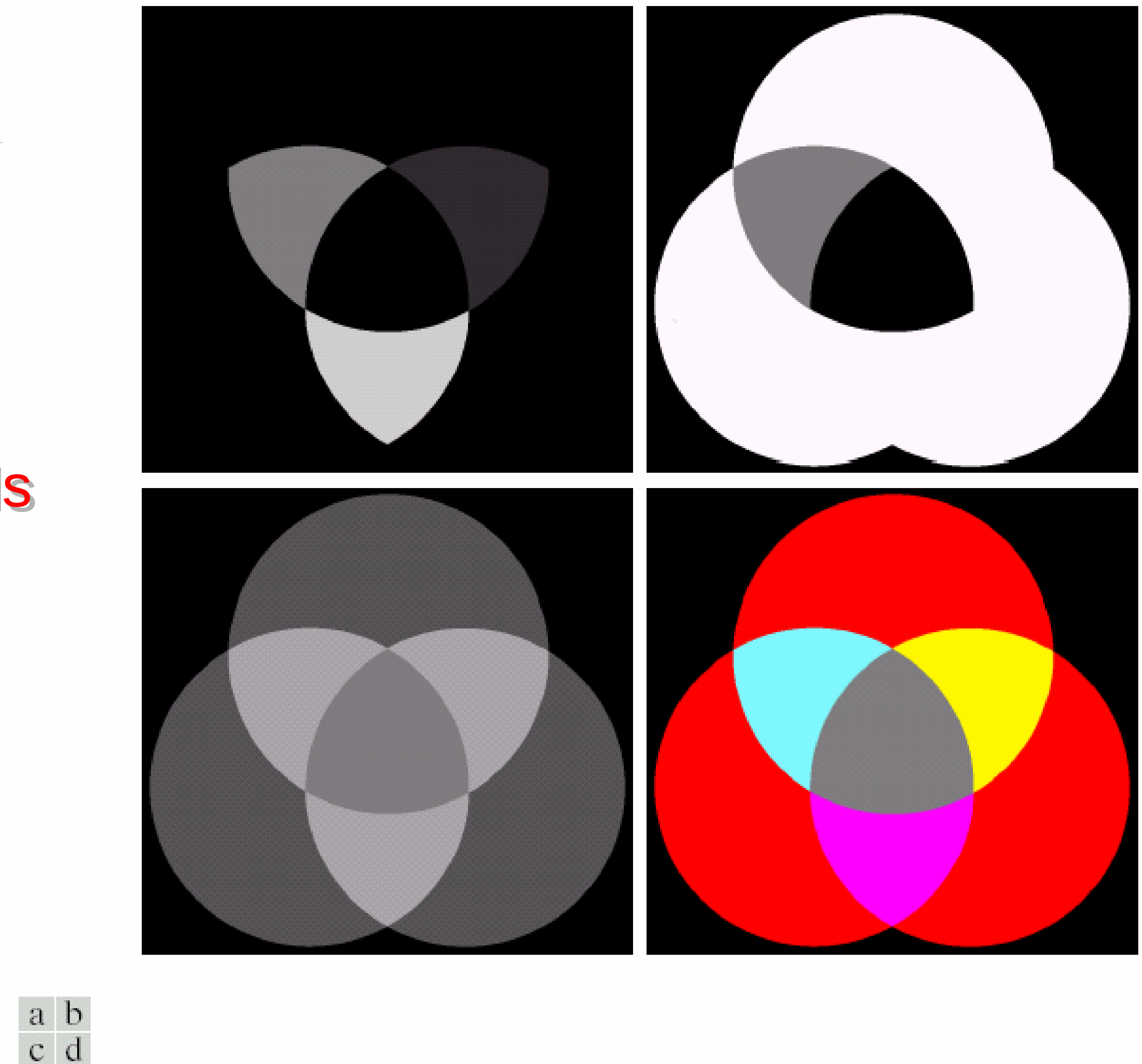
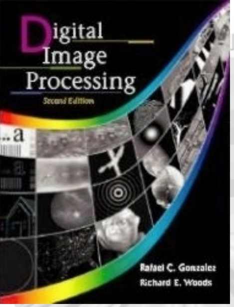



**FIGURE 6.17** (a)–(c) Modified HSI component images. (d) Resulting RGB image. (See Fig. 6.16 for the original HSI images.)

# 6.3 Pseudo Image Processing

- Assigning colors to gray values based on a specified criterion.
- *Intensity slicing:* using a plane at $f(x, y)=l_i$ to slice the image function into two levels.
- In general, we assume that P planes perpendicular to the intensity axis defined at level $l_i$ $i=1,2,..P$. These $P$ planes partition the gray level in to $P+1$ intervals: $V_k$ $k=1,2,..P+1$
- $f(x, y)=c_i$ if $f(x, y) \in V_k$
- where $c_i$ is the color associated with the $k$th intensity interval $V_k$ defined by the partition lanes at $l=k-1$ and $l=k$.
- From Figure 6.19; if more levels are used, the mapping function takes on a staircase form.




# 6.3 Pseudo Image Processing

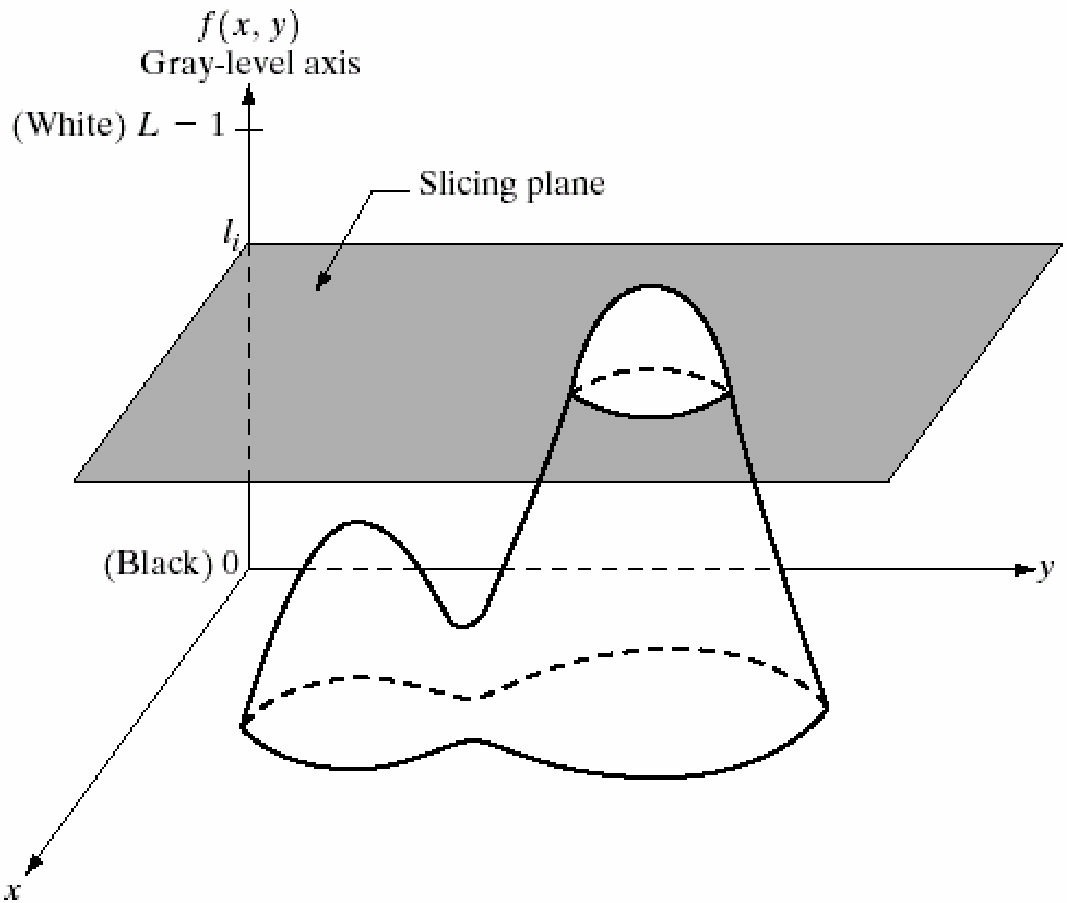



**FIGURE 6.18** Geometric interpretation of the intensity-slicing technique.

# 6.3 Pseudo Image Processing

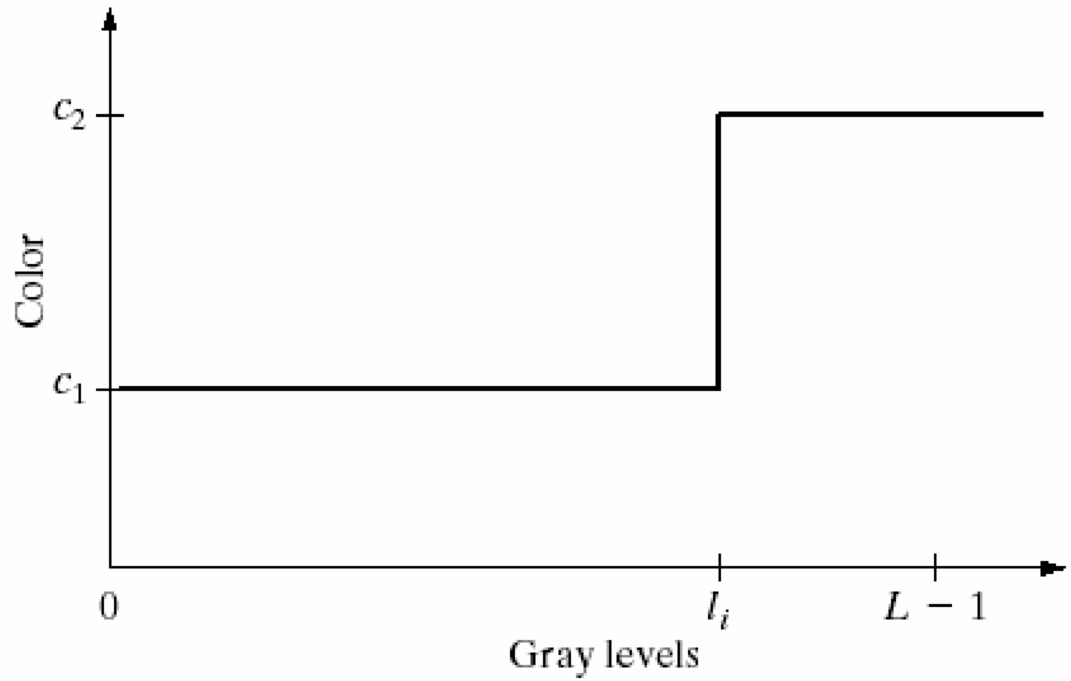
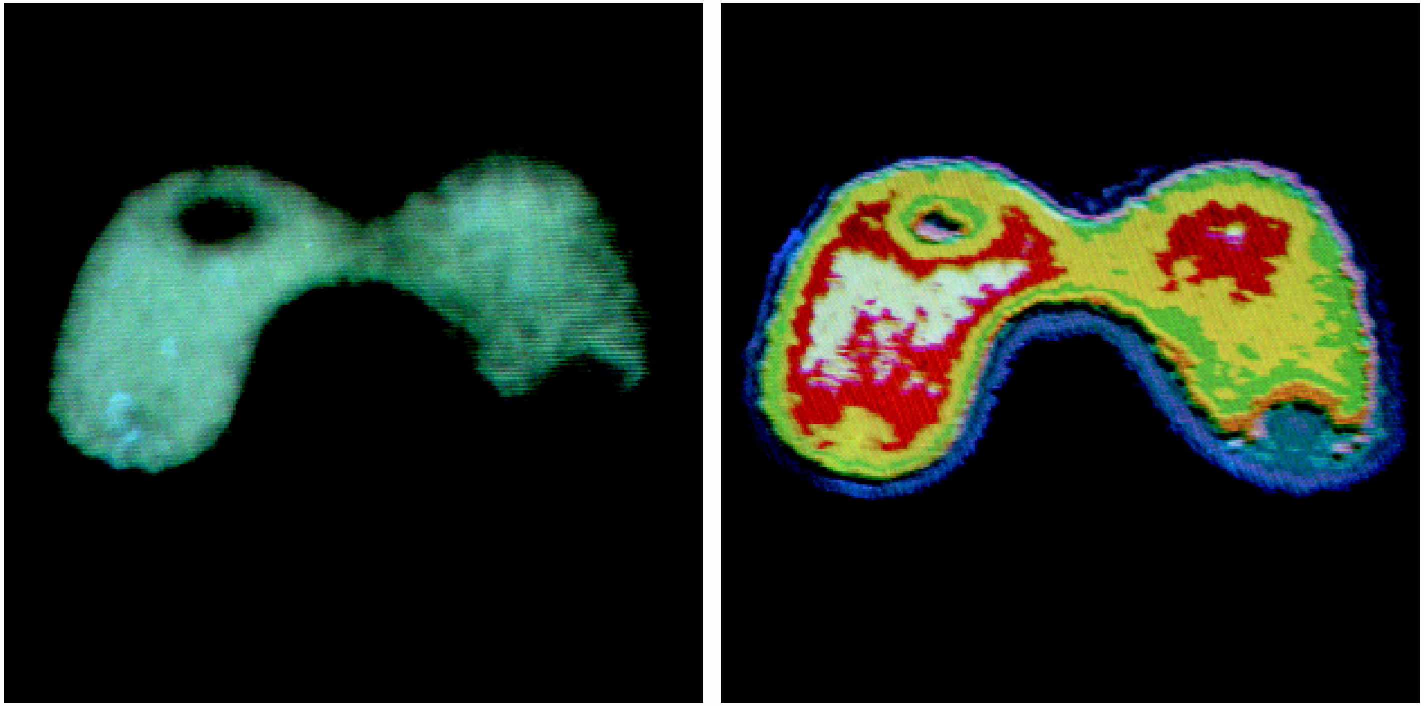



**FIGURE 6.19** An alternative representation of the intensity-slicing technique.

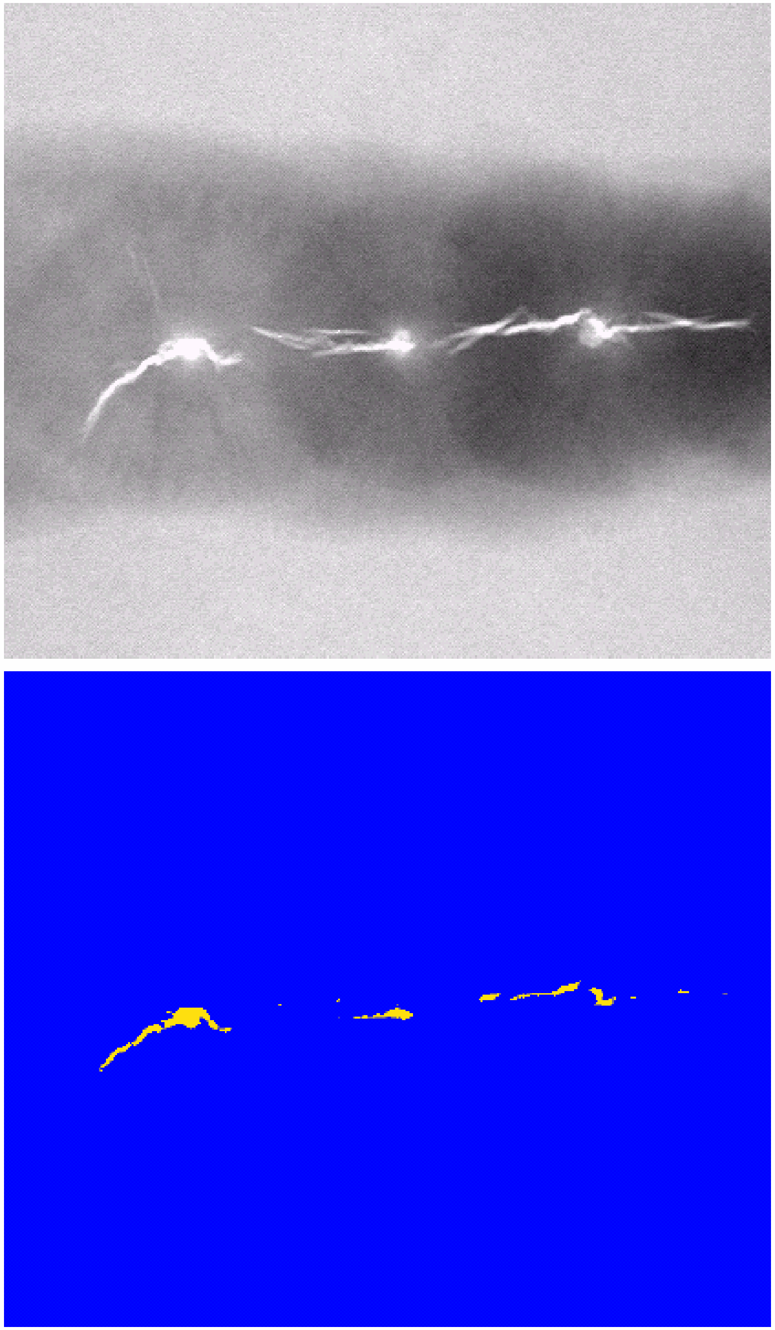
# 6.3 Pseudo Image Processing



a b

**FIGURE 6.20** (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)

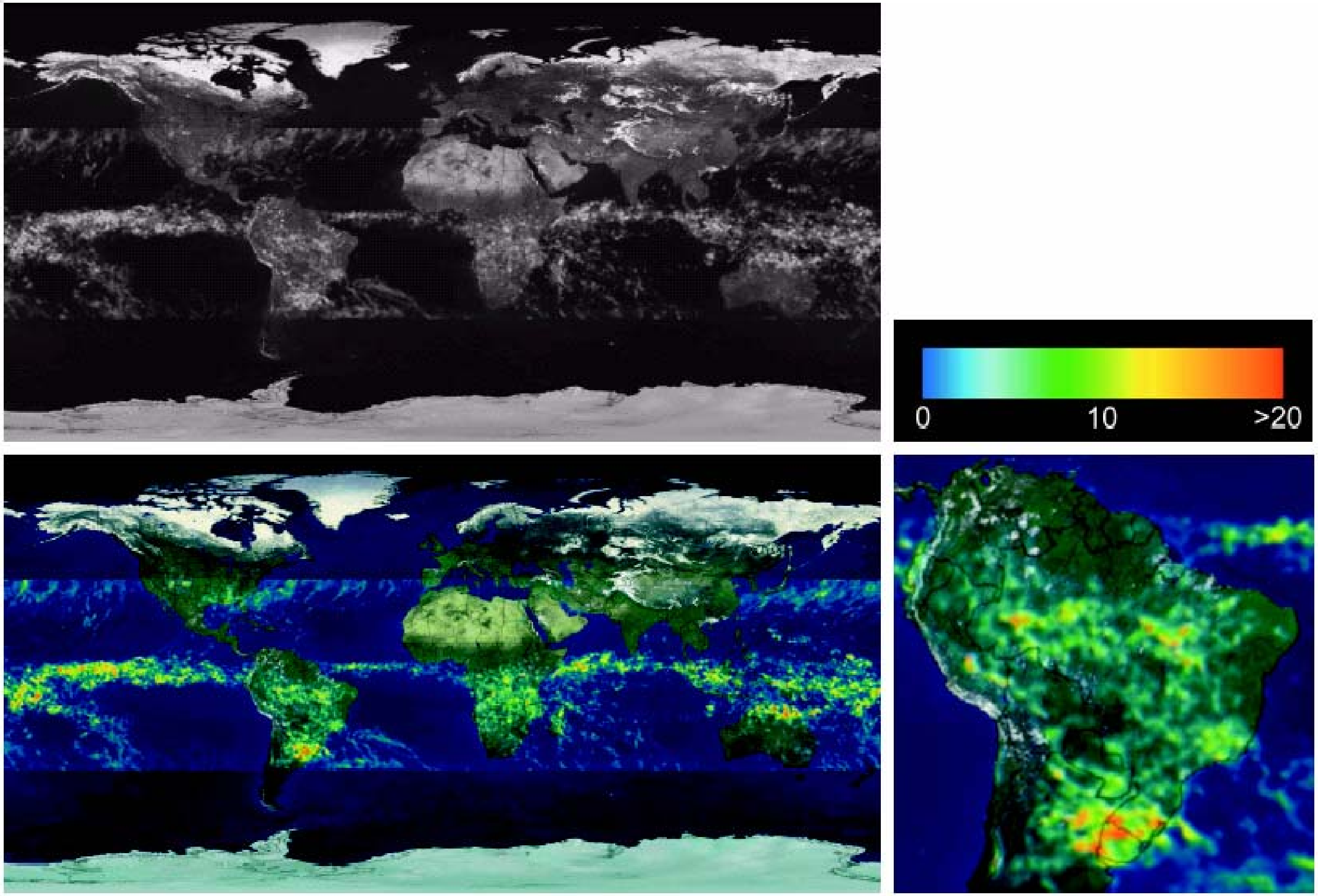
# 6.3 Pseudo Image Processing



a
b

**FIGURE 6.21** (a) Monochrome X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)

# 6.3 Pseudo Image Processing




a b
c d

**FIGURE 6.22** (a) Gray-scale image in which intensity (in the lighter horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South America region. (Courtesy of NASA.)

# 6.3 Pseudo Image Processing- gray-level to color transformation

- Three independent transformation functions on the gray-level of each pixel.
- Piecewise linear function
- Smooth non-linear function

# 6.3 Pseudo Image Processing

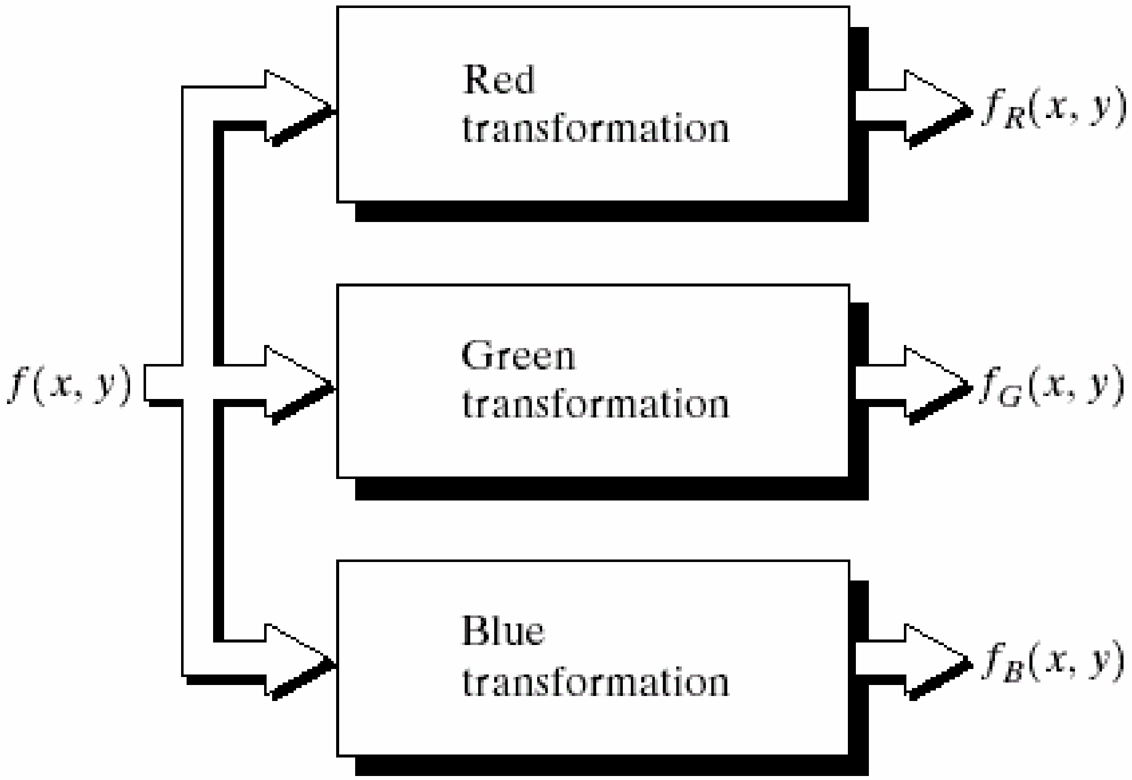
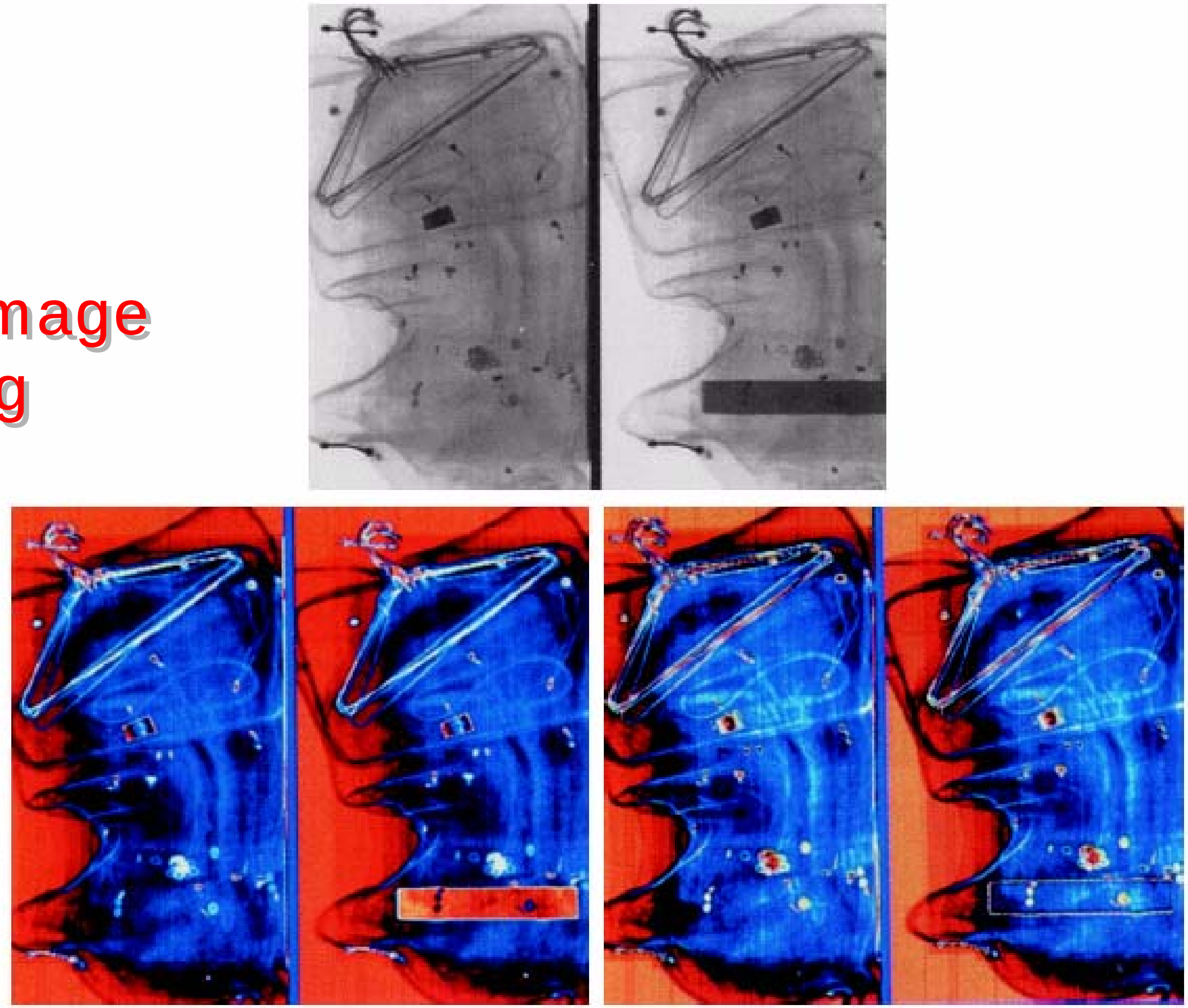



**FIGURE 6.23** Functional block diagram for pseudocolor image processing. $f_R$, $f_G$, and $f_B$ are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

# 6.3 Pseudo Image Processing

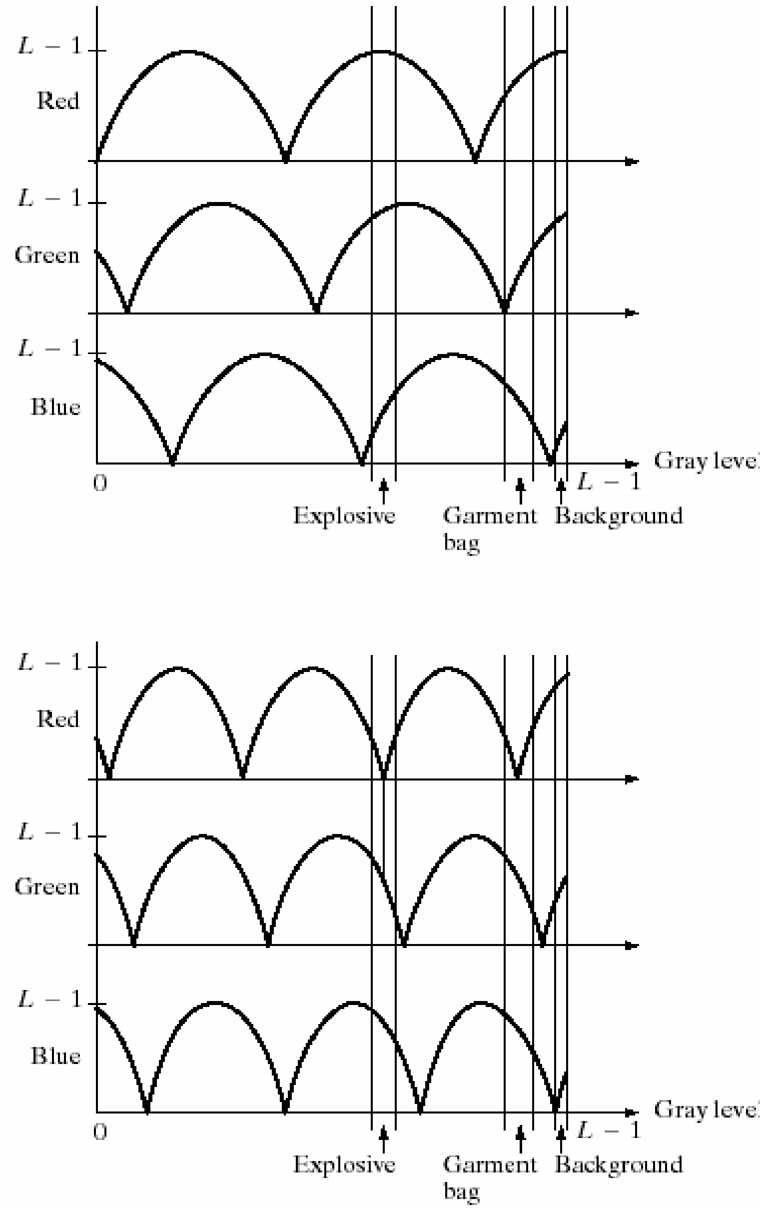


a
b c

Figure 6.25 (a)    Figure 6.25 (b)

**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.)

# 6.3 Pseudo Image Processing




a
b

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

# 6.3 Pseudo Image Processing

- Change the phase and frequency of each sinusoid can emphasize (in color) ranges in the gray scale.
- Peak → constant color region.
- Valley → rapid changed color region.
- A small change in the phase between the three transforms produces little change in pixels whose gray level corresponding to the peaks in the sinusoidal.
- Pixels with gray level values in the steep section of the sinusoids are assigned much strong color.

# 6.3 Pseudo Image Processing

Combine several monochrome images into a single color image.

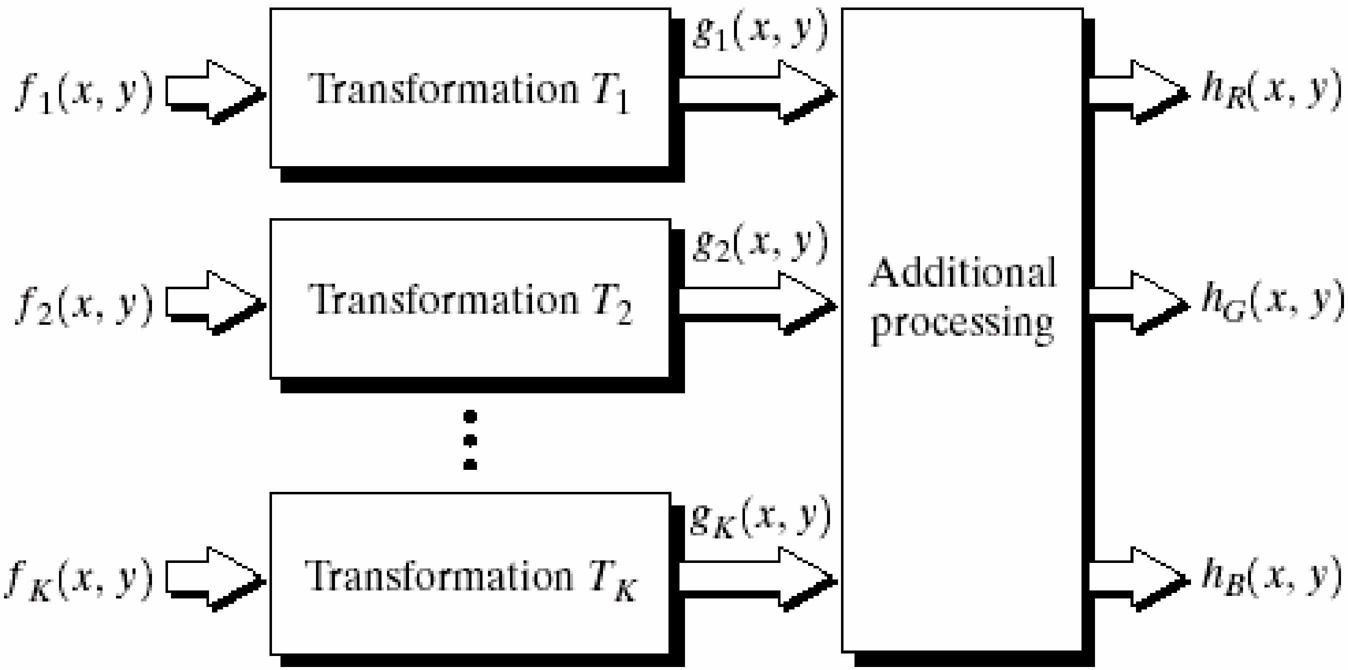
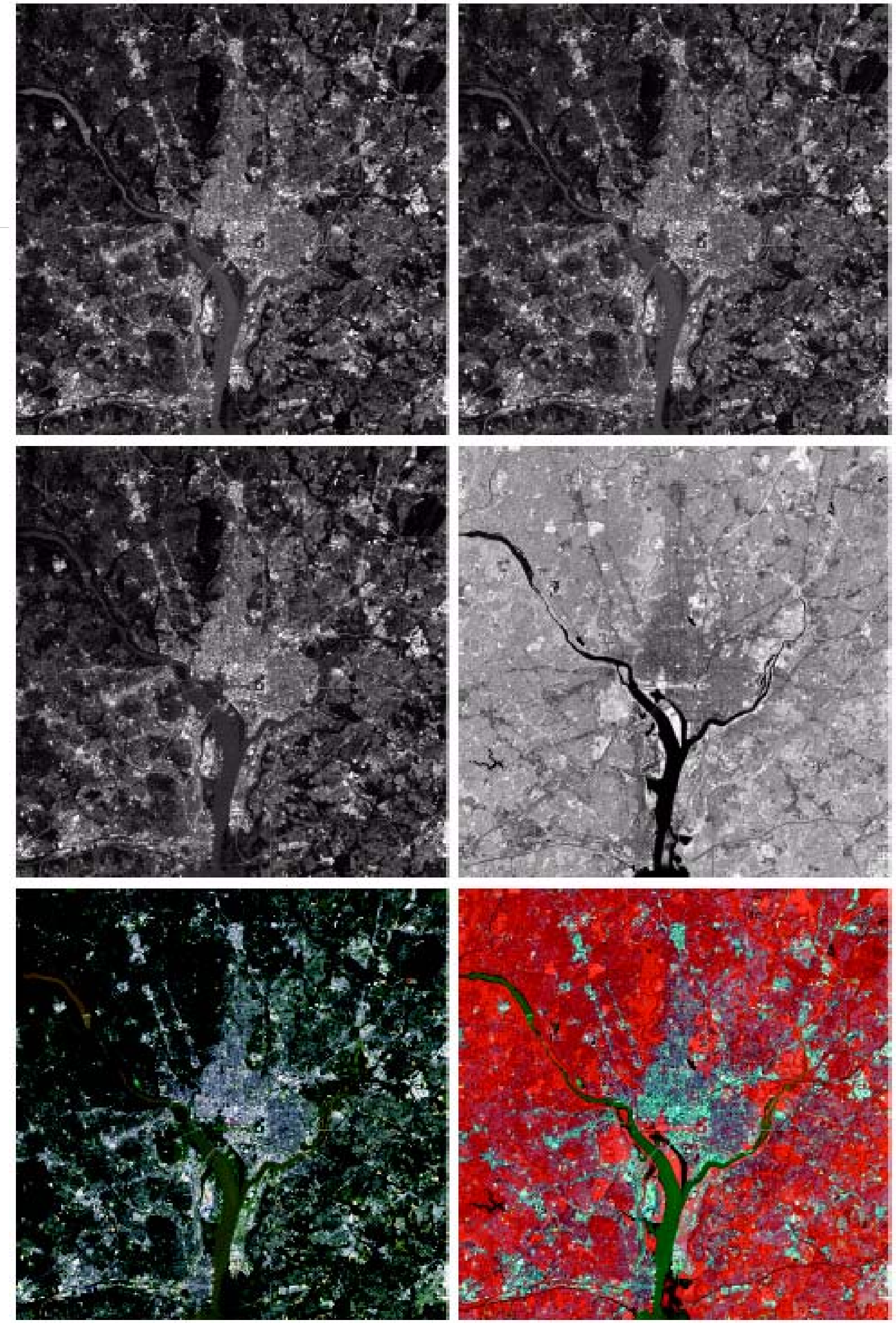



**FIGURE 6.26** A pseudocolor coding approach used when several monochrome images are available.

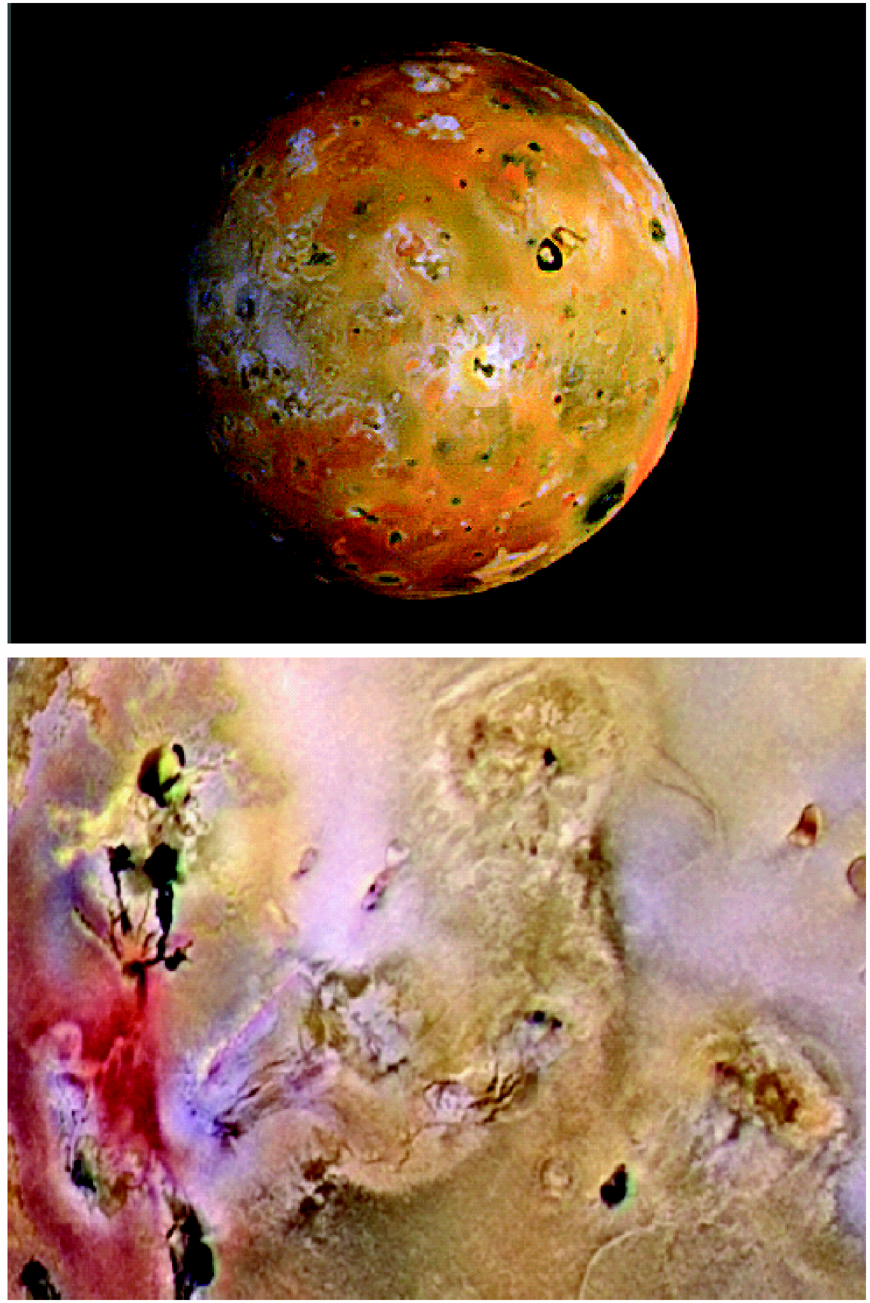
# 6.3 Pseudo Image Processing



| | |
|---|---|
| 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.)

# 6.3 Pseudo Image Processing



a
b

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

# 6.4 Full-Color Image Processing

- Two categories:
  - Process each component individually and then form a composite processed color image from the components.
  - Work with color pixels directly. In RGB system, each color point can be interpreted as a vector.
  - $\mathbf{c}(x, y)=[c_R(x, y), c_G(x, y), c_B(x, y)]$

# 6.4 Full-Color Image Processing

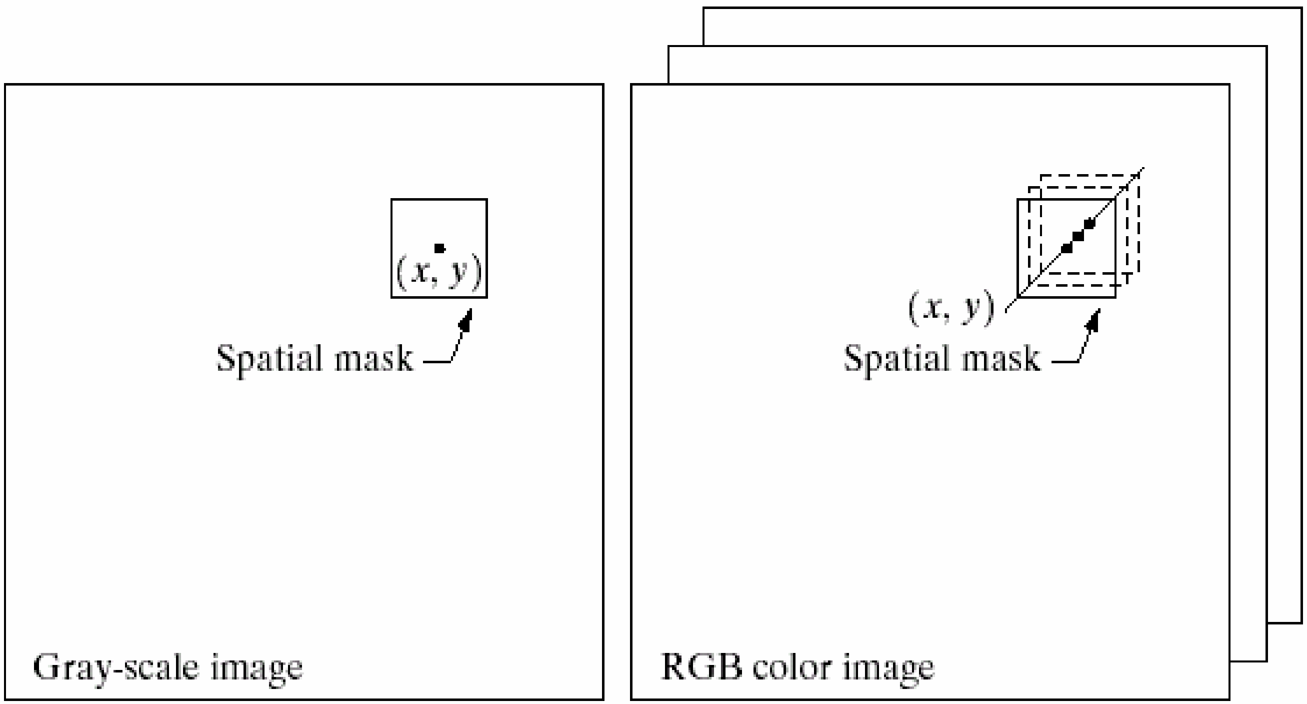



**FIGURE 6.29** Spatial masks for gray-scale and RGB color images.

# 6.5 Color Transformation- formulation

Gray-level transformation

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

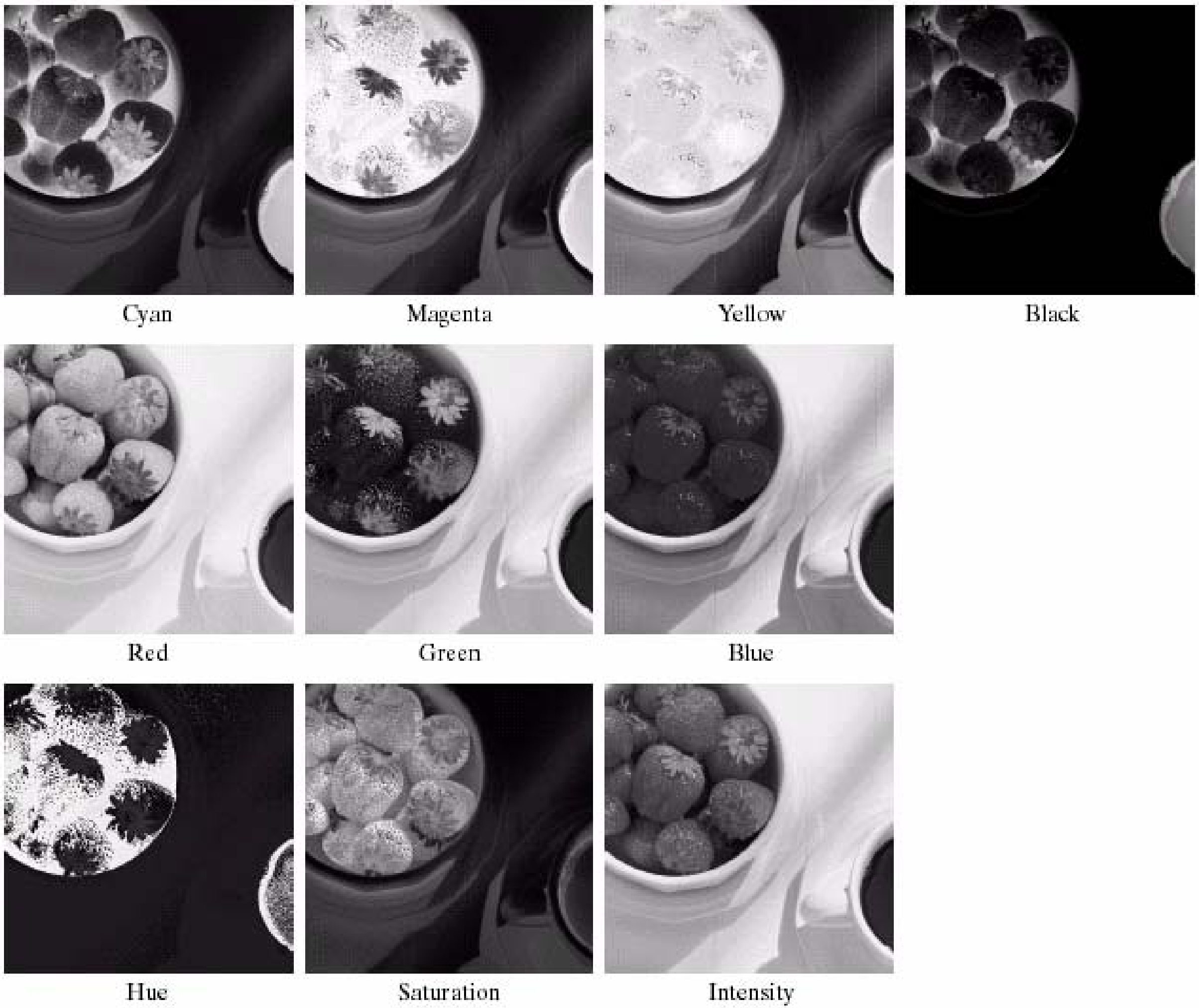
Color transformation

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

Where $r_i$ and $s_i$ are variables denoting the color component of $f(x, y)$ and $g(x, y)$ at any point $(x, y)$, $n$ is the number of color components, and $\{T_i\}$ is a set of transformation or color mapping functions.

# 6.5 Color Transformation


FIGURE 6.30 A full-color image and its various color-space components. (Original image courtesy of Med-Data Interactive.)



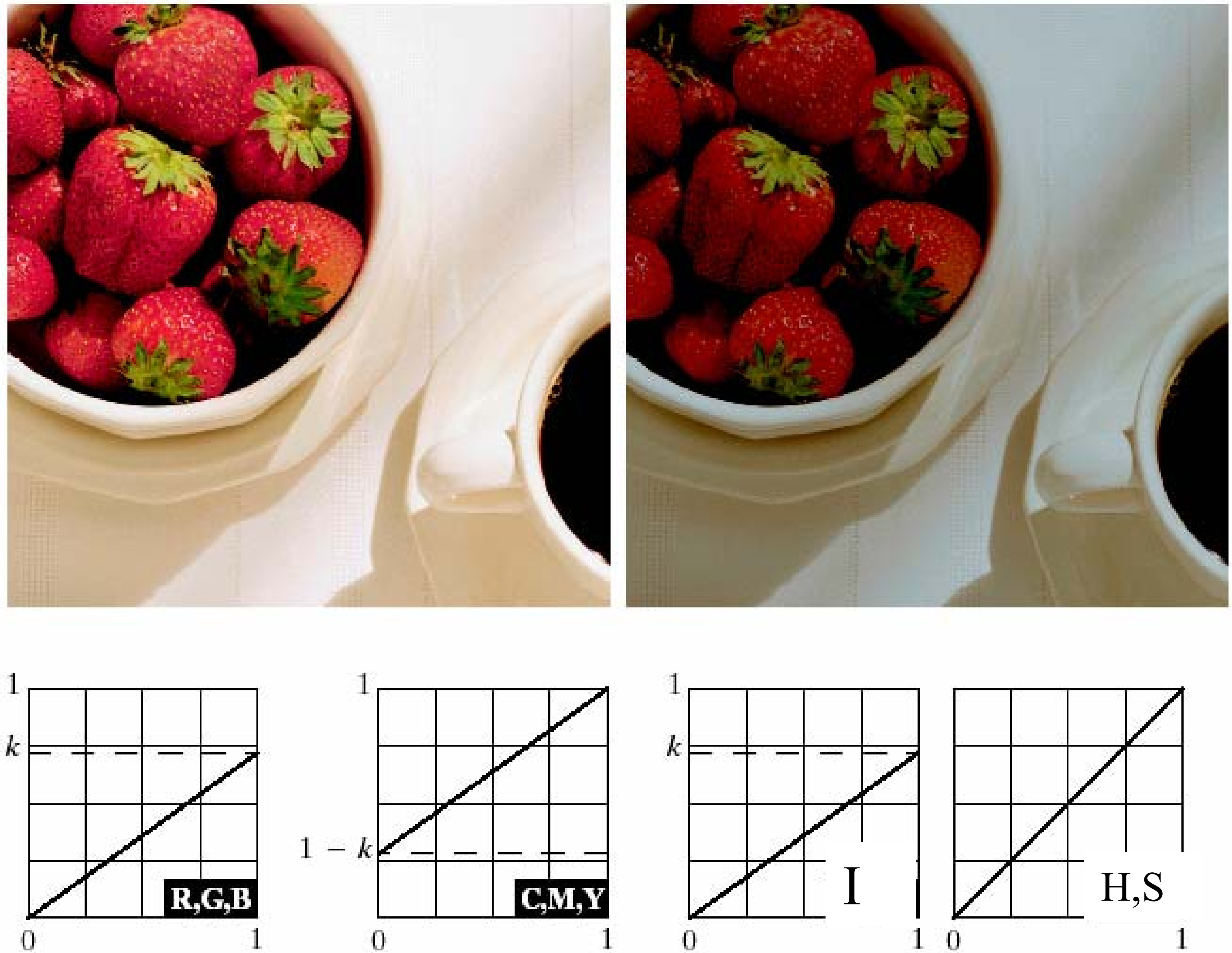
# 6.5 Color Transformation

- To modify the intensity of the image

$$g(x, y)=kf(x, y) \quad 0<k<1$$

- $HSI : s_3=kr_3$
- $RGB: \ s_i=kr_i \quad i=1, 2, 3$
- $CMY: s_i=kr_i+(1-k) \quad i=1, 2, 3$

# 6.5 Color Transformation




a b
c d e

**FIGURE 6.31** Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)

# 6.5 Color Transformation - Color Complements

- The hues directly opposite one another on the *color circle* are called *complements*
- *Color complements* are useful for enhancing detail that is embedded in dark regions of a color image

# 6.5 Color Transformation - Color Complements

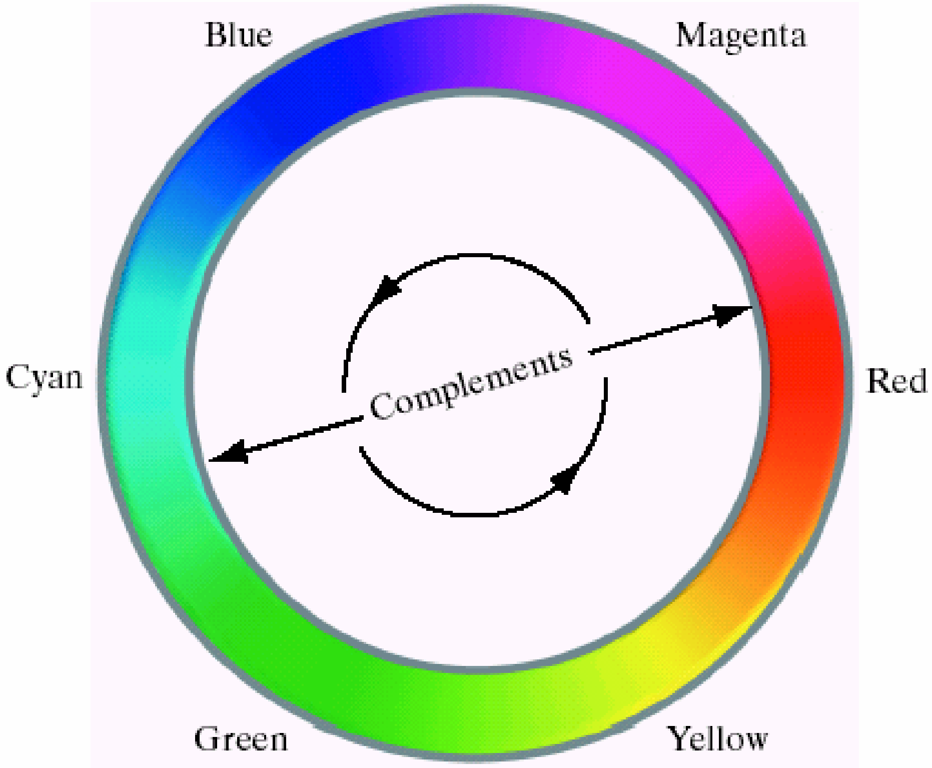



**FIGURE 6.32** Complements on the color circle.

# 6.5 Color Transformation - Color Complements

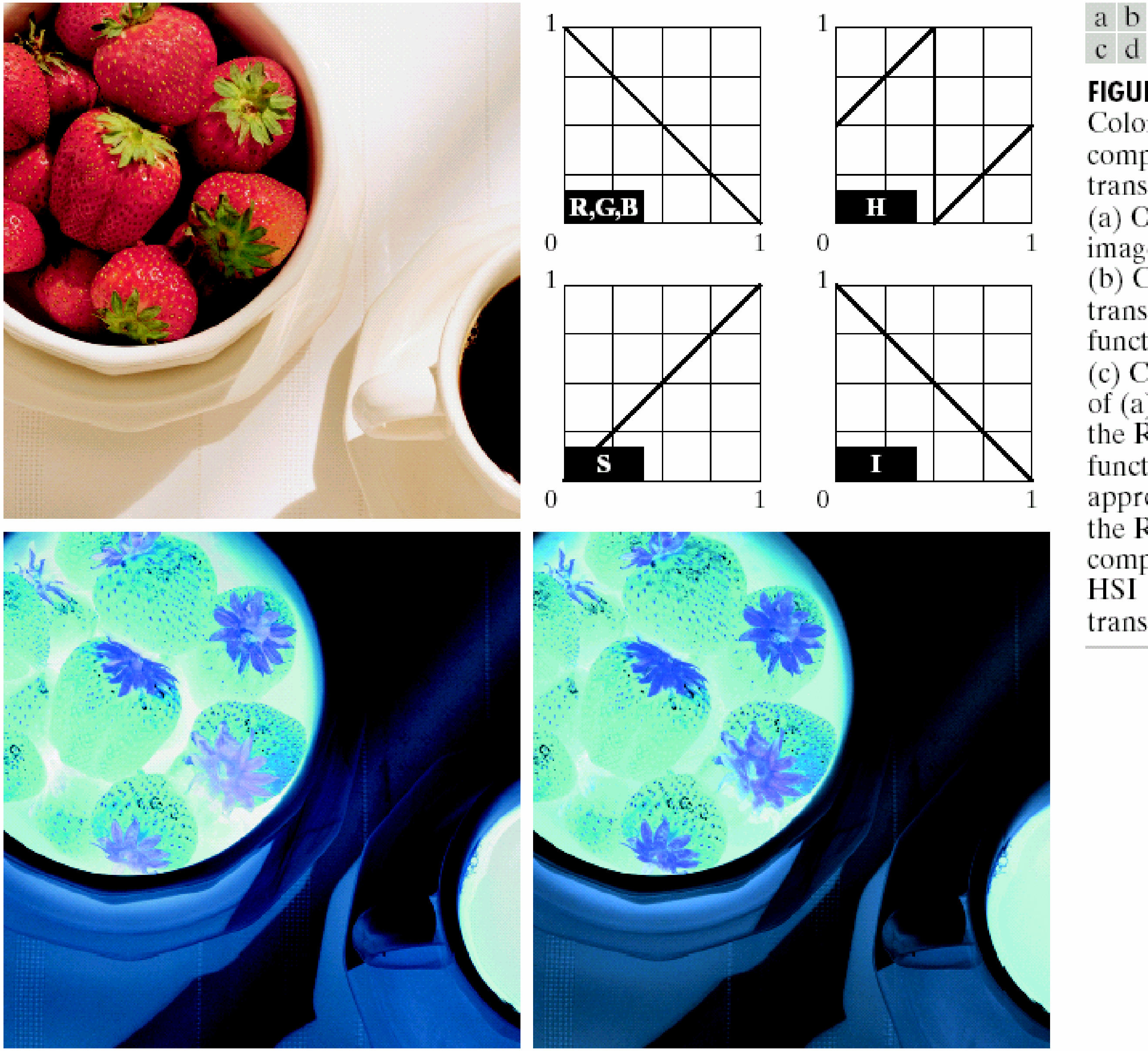



**FIGURE 6.33** Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.

# 6.5 Color Transformation - Color Slicing

- Highlighting a specific range of colors in an image is useful for separating object from their surrounding.
- The simplest way to "slice" a color image is to map the colors outside some range of interest to a nonprominent *neutral color* (*e.g.*, (*R*, *G*, *B*)=(*0.5*, *0.5*, *0.5*)). If the colors of interest are enclosed by a cube (or hypercube for n>3) of width W and centered at a average color with component $(a_1, a_2, \ldots a_n)$ the necessary set of transformation is

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

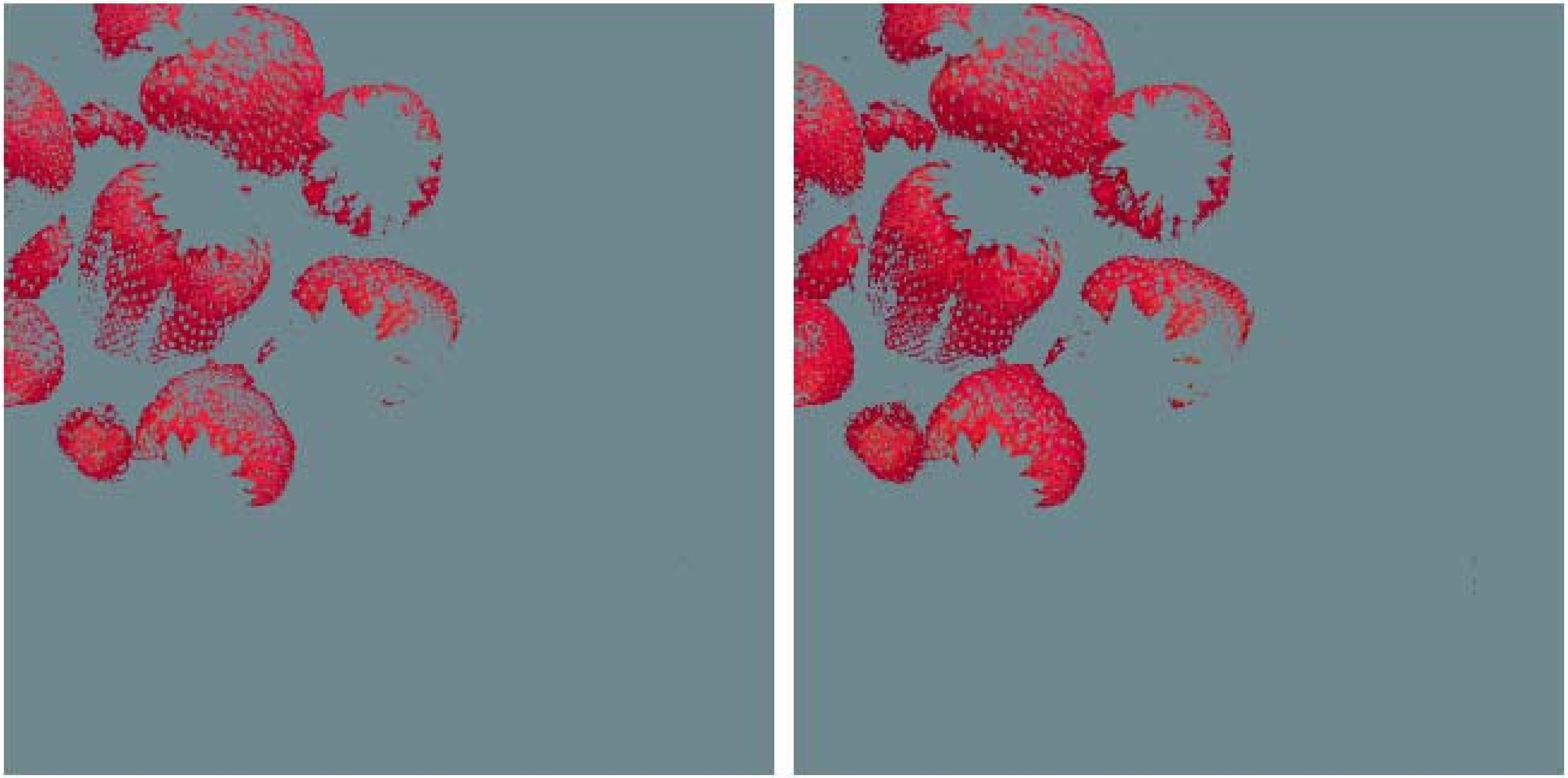
# 6.5 Color Transformation - Color Slicing

- If a sphere is used to specify the colors of interest then

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

- Forcing all other colors to the mid point of the reference color space.
- In RGB color space, the *neural color* is (0.5, 0.5, 0.5)

# 6.5 Color Transformation - Color Slicing



a b

**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).

# 6.5 Color Transformation – Tone and Color Correction

- ***Digital Darkroom***
- Effective transformation are developed to maintain a high degree of ***color consistency*** between the monitor used and the eventual output devices.
- ***Device independent color model***: relate the color gamut (see Fig. 6.6) of the monitor and output devices as well as other devices to one another.

# 6.5 Color Transformation – Tone and Color Correction

- The model choice for many **color management systems** (CMS) is the CIE L*a*b* model called CIELAB.
- The *L*a*b** color component is given as

$$L^* = 116h(Y/Y_W) - 16,$$

$$a^* = 500[h(X/X_W) - h(Y/Y_W)]$$

$$b^* = 200[h(Y/Y_W) - h(Z/Z_W)]$$

where $$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.878q + 16/116 & q \le 0.008856 \end{cases}$$

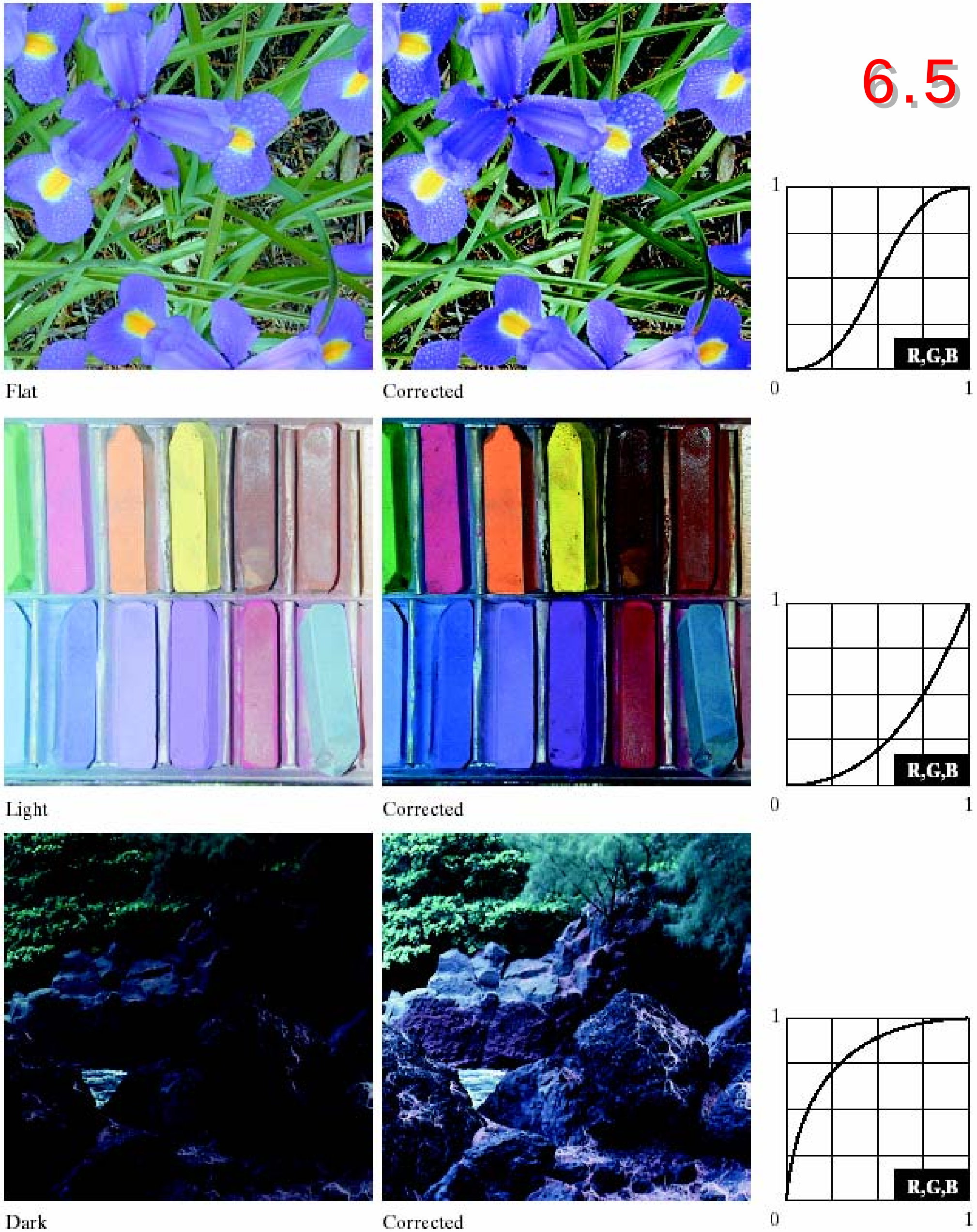
# 6.5 Color Transformation – Tone and Color Correction

- $X_W, Y_W, Z_W$ are ***reference white tristimulus values***.
- The $L^*a^*b^*$ color is ***colormetric*** (*i.e.*, colors perceived as matching are encoded identically), ***perceptual uniform*** (*i.e.*, color differences among various hues are perceived uniformly), and ***device independent***.
- *It is not a directly displayable format.*
- The *gamut* of $L^*a^*b^*$ encompasses the entire visible spectrum and can represent accurately the colors of any display, print, or input device.
- $L^*a^*b^*$ decouples intensity (L*) and color (a* and b*)

# 6.5 Color Transformation – Tone and Color Correction

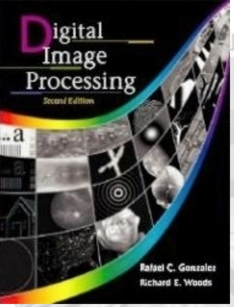
- Before color irregularities are solved, the image's tonal range are corrected.
- The tonal range of an image (*key type*) refers to its general distribution of color intensity.
- *High key image* is concentrated at high/light intensity
- *Low key image* is concentrated at low intensity.
- *Middle key image* lies in between.
- It is desireable to distribute the intensities of a color image equally between the highlights and the shadows

# 6.5 Color Transformation - Color Correction




Tonal transformation for *flat*, *light* and *dark* images

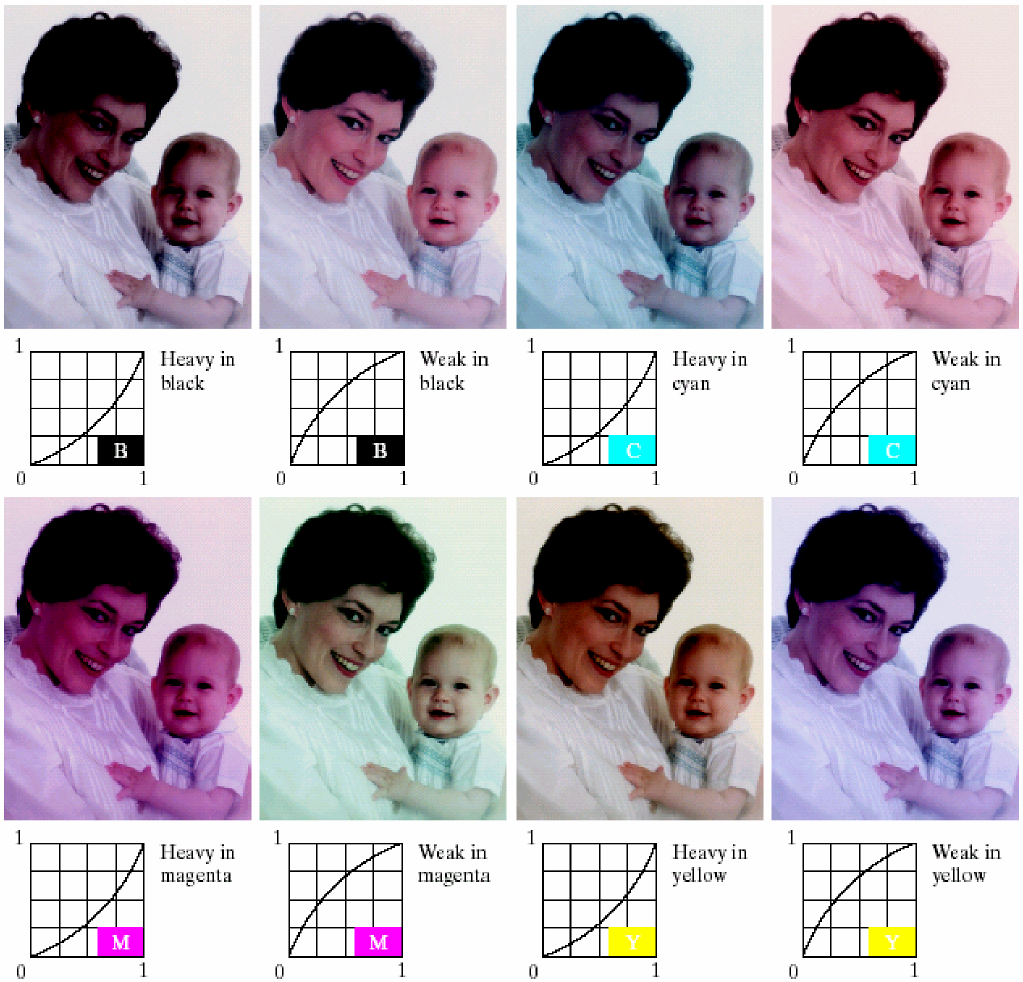
**FIGURE 6.35** Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.




# 6.5 Color Transformation – Tone and Color Correction

FIGURE 6.36 Color balancing corrections for CMYK color images.

Original/Corrected




***Color Balancing***: The proportion of any color can be increased by decreasing the amount of opposite (complementary) color in the image. Refer to the color wheel (Figure 6.32) to see how one color component will affect the other.

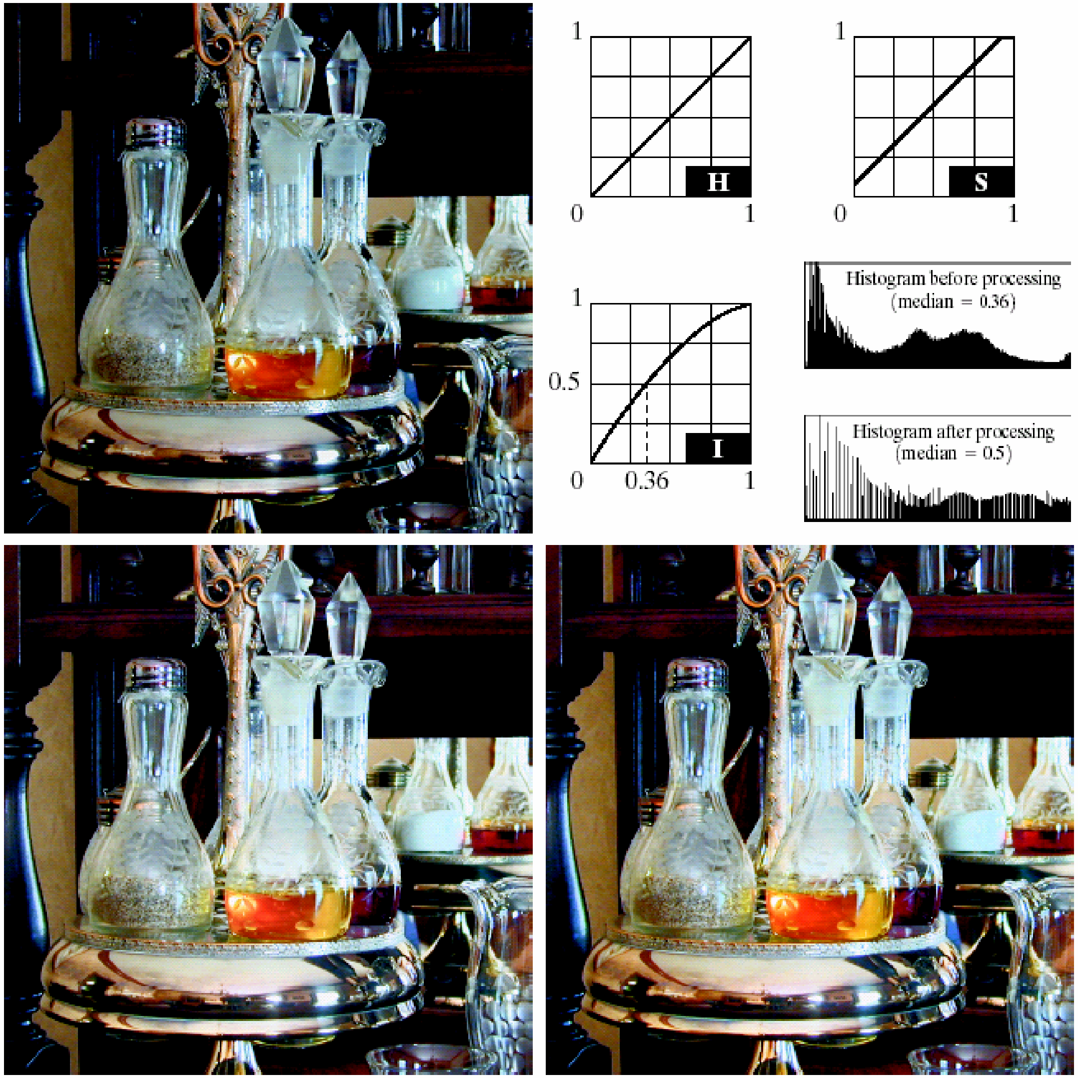
# 6.5 Color Transformation – Histogram Processing

- Equalized the histogram of each component will results in error color.
- Spread the color intensity (I) uniformly, leaving the color themselves (hues) unchanged.
- Equalizating the intensity histogram affects the relative appearance of colors in an image.
- Increasing the image's saturation component after the intensity histogram equalization.

# 6.5 Color Transformation – Histogram Processing

Mean=0.36

Mean=0.5




a b
c d

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

# 6.6 Smoothing and Sharpening

- Let $S_{xy}$ denote the set of coordinates defining a neighborhood centred at *(x, y)* in an RGB color space.

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

# 6.6 Smoothing and Sharpening

a b
c d

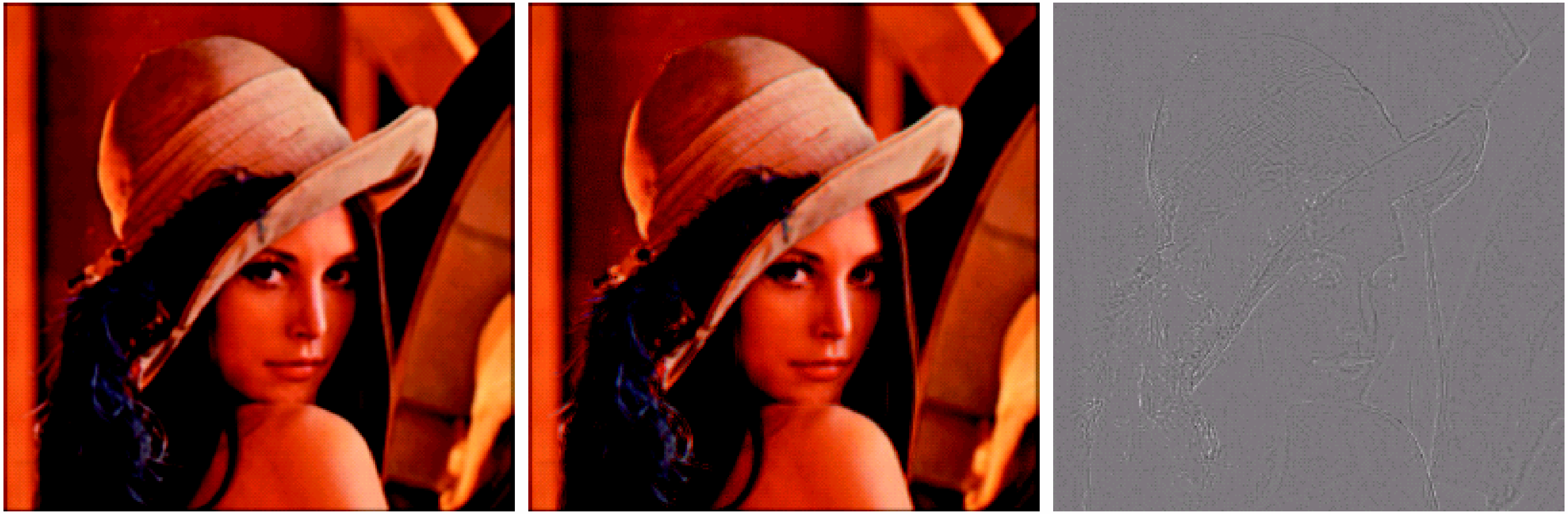
**FIGURE 6.38** (a) RGB image. (b) Red component image. (c) Green component. (d) Blue component.

# 6.6 Smoothing and Sharpening

a b c

**FIGURE 6.39** HSI components of the RGB color image in Fig. 6.38(a). (a) Hue. (b) Saturation. (c) Intensity.

# 6.6 Smoothing and Sharpening



a b c

Smooth only the intensity

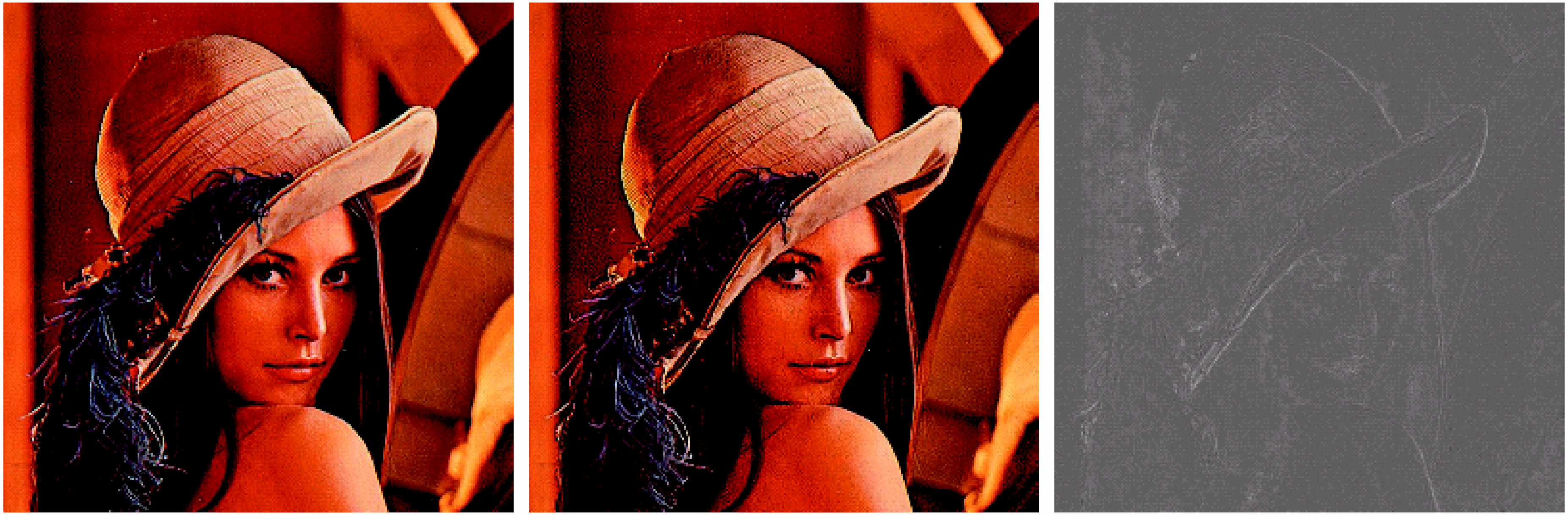
**FIGURE 6.40** Image smoothing with a 5 × 5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

# 6.6 Smoothing and Sharpening

- Image sharpening using Laplacian

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

# 6.6 Smoothing and Sharpening



Hue and Saturation unchanged

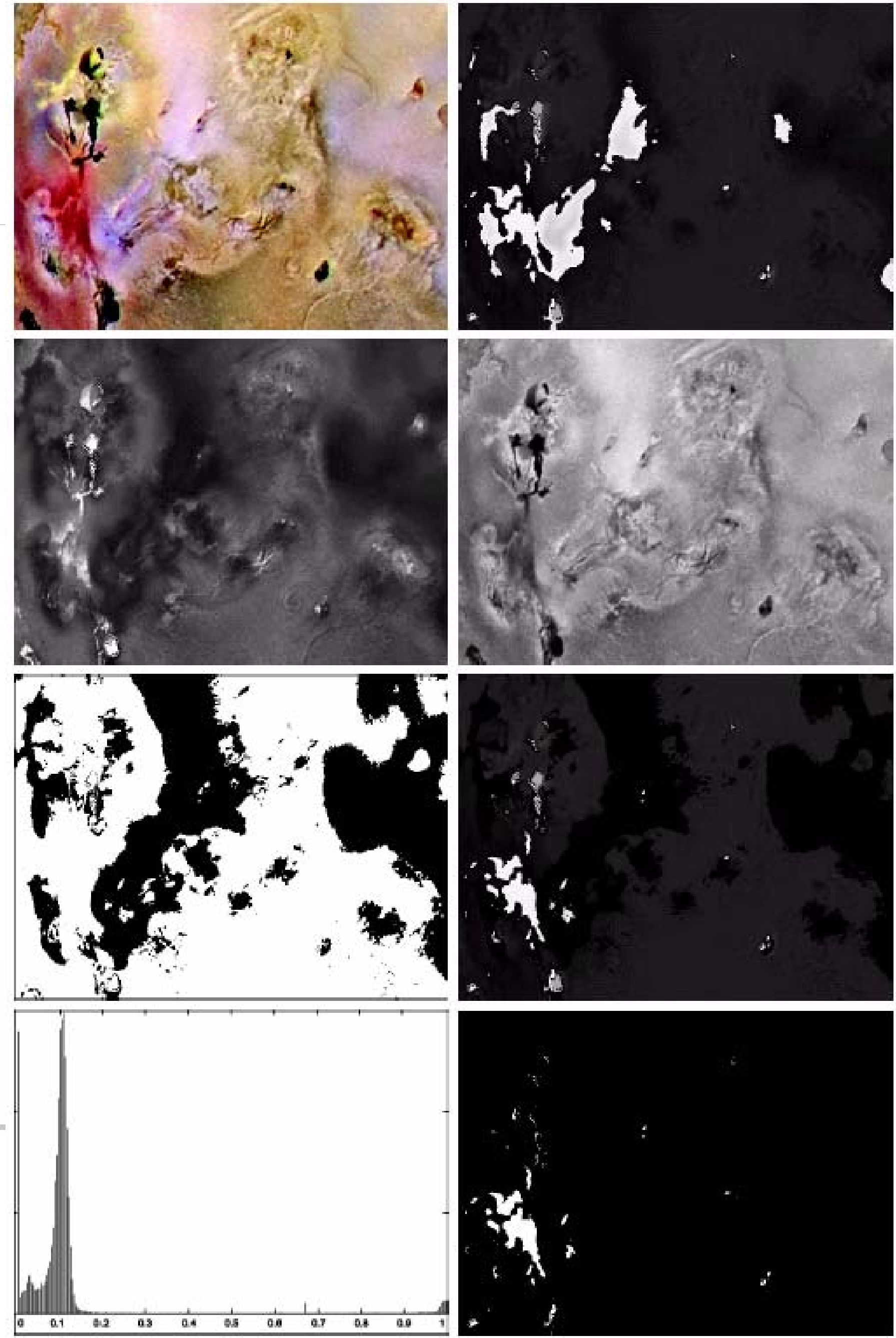
a b c

**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.

# 6.7 Color Segmentation

- Partition an image into regions.
- Segmentation in *HIS* color space.
- Saturation is used as a masking image to isolate further regions of interest in the hue image.
- The intensity image is used less frequently.

# 6.7 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).

# 6.7 Color Segmentation

- Segmentation in RGB color space
- The measurement of color similarity is the Euclidean distance between two colors **z**, and **a**, (*i.e*. Fig. 6.43(a)),

$$D(\mathbf{z}, \mathbf{a}) = \|z - \mathbf{a}\| = [(z - \mathbf{a})^T (\mathbf{z} - \mathbf{a})]^{1/2}$$

- $= [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{1/2}$
- A generalization of distance measure is

$$D(\mathbf{z}, \mathbf{a}) = \|z - \mathbf{a}\| = [(z - \mathbf{a})^T \mathbf{C}^{-1} (\mathbf{z} - \mathbf{a})]^{1/2}$$

- Where **C** is the covariance matrix of the samples representative of the color we want to segment.
- In Figure 6.43(b) describes the solid elliptical body with the principal axes oriented in the direction of maximum data spread.

# 6.7 Color Segmentation

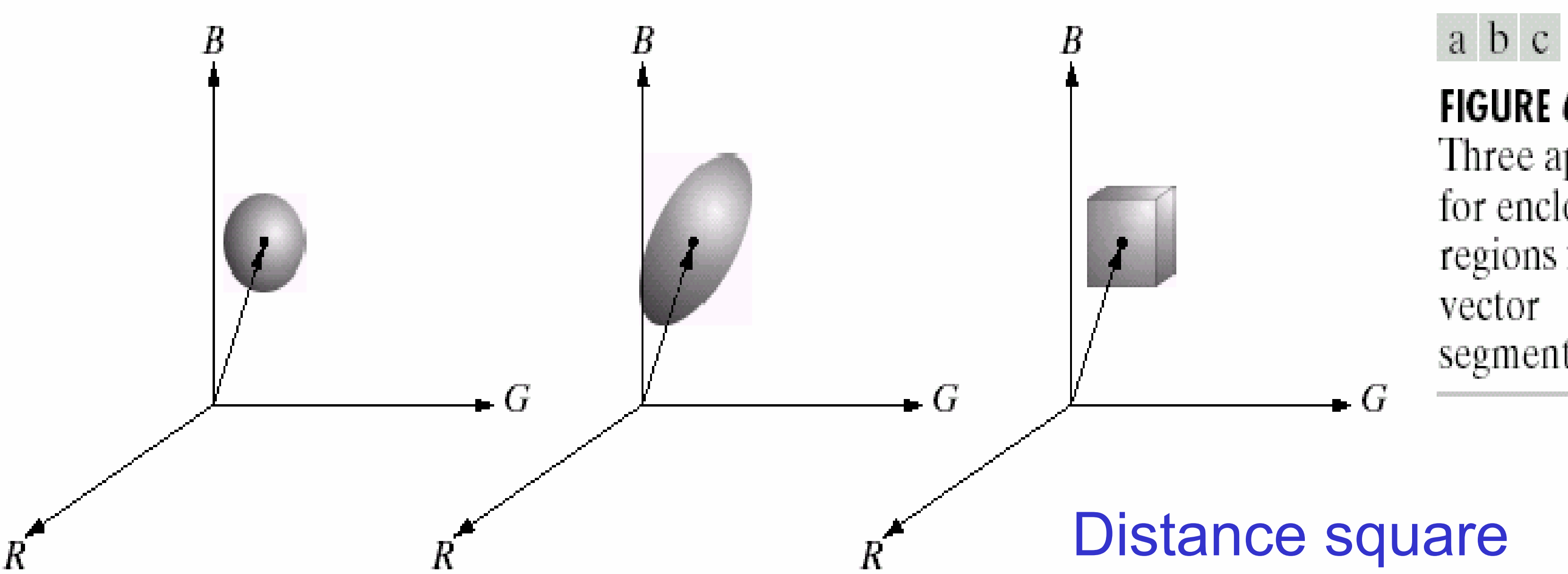



**FIGURE 6.43** Three approaches for enclosing data regions for RGB vector segmentation.

Distance square without square root operation.

# 6.7 Color Segmentation

Find the distance of color ( $H_j$, $S_j$, $I_j$ ) and the dominant color ( $\bar{H}, \bar{S}, \bar{I}$ )

$$d_{\text{intensity}}(j) = | I_j - \bar{I} |$$

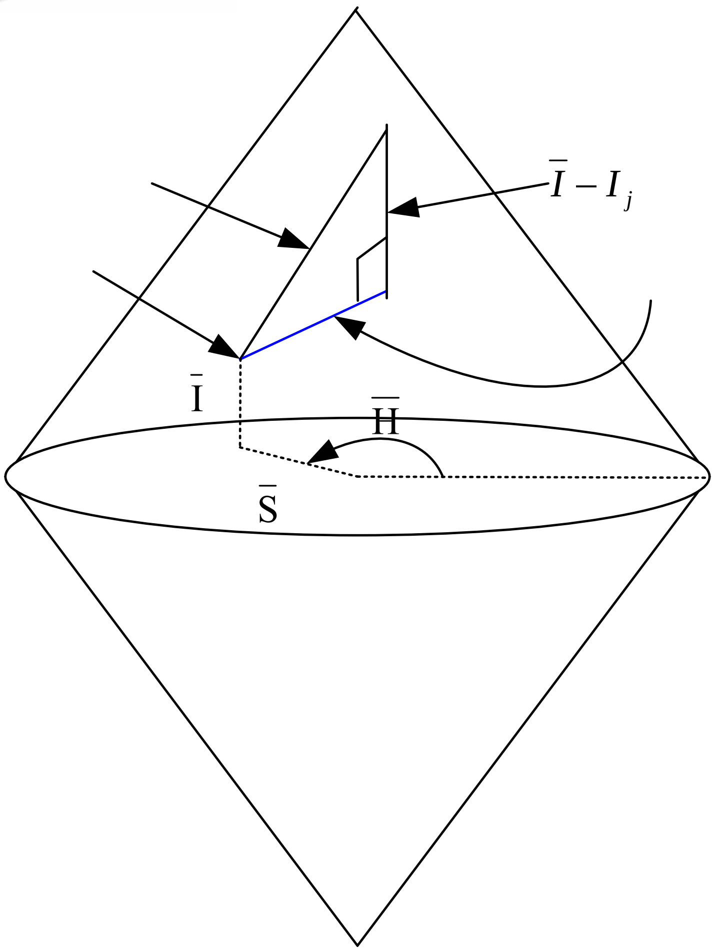
$$d_{\text{chroma}}(j) = \sqrt{(S_j)^2 + (\bar{S})^2 - 2 S_j \bar{S} \cos(\theta(j))}$$

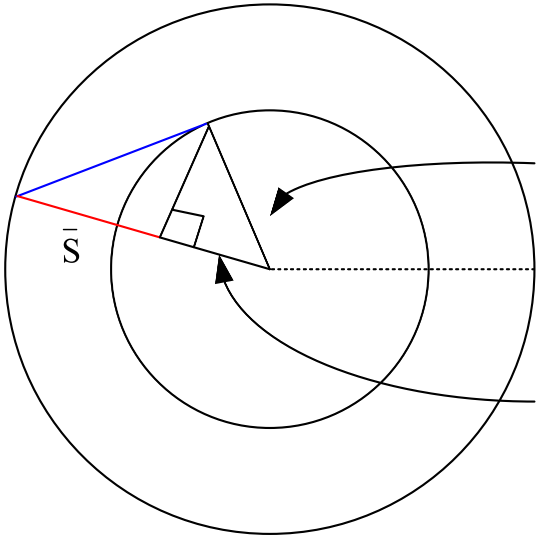
$$d_{cylindiral}(j) = \sqrt{(d_{\text{intensity}}(j))^2 + (d_{\text{chroma}}(j))^2}$$

$$\begin{cases} \Omega(j) & \text{if} \quad \Omega(\text{j}) \le 180^{\circ} \\ 360^{\circ} - \Omega(j) & \text{otherwise} \end{cases}$$

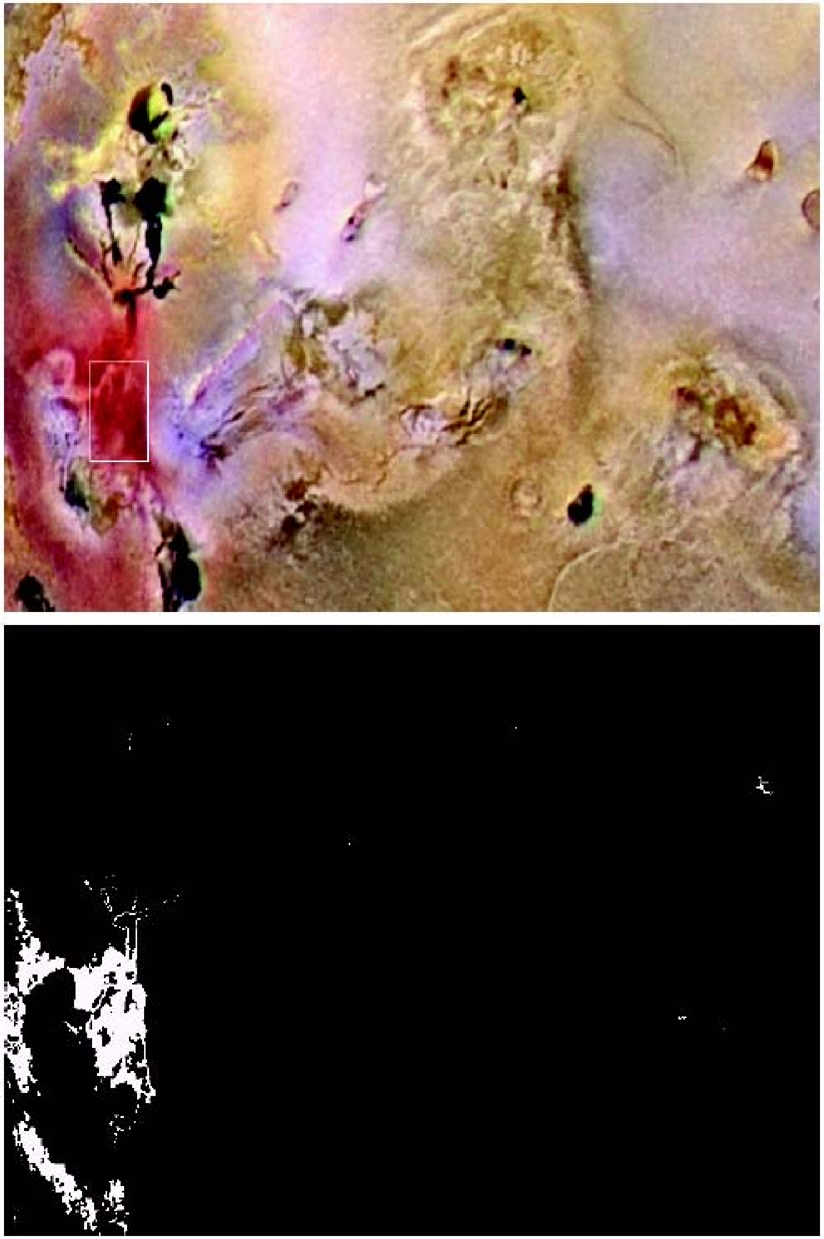
$$\Omega(j) = | \bar{H} - H_j |$$

# 6.7 Color Segmentation



# 6.7 Color Segmentation



a
b

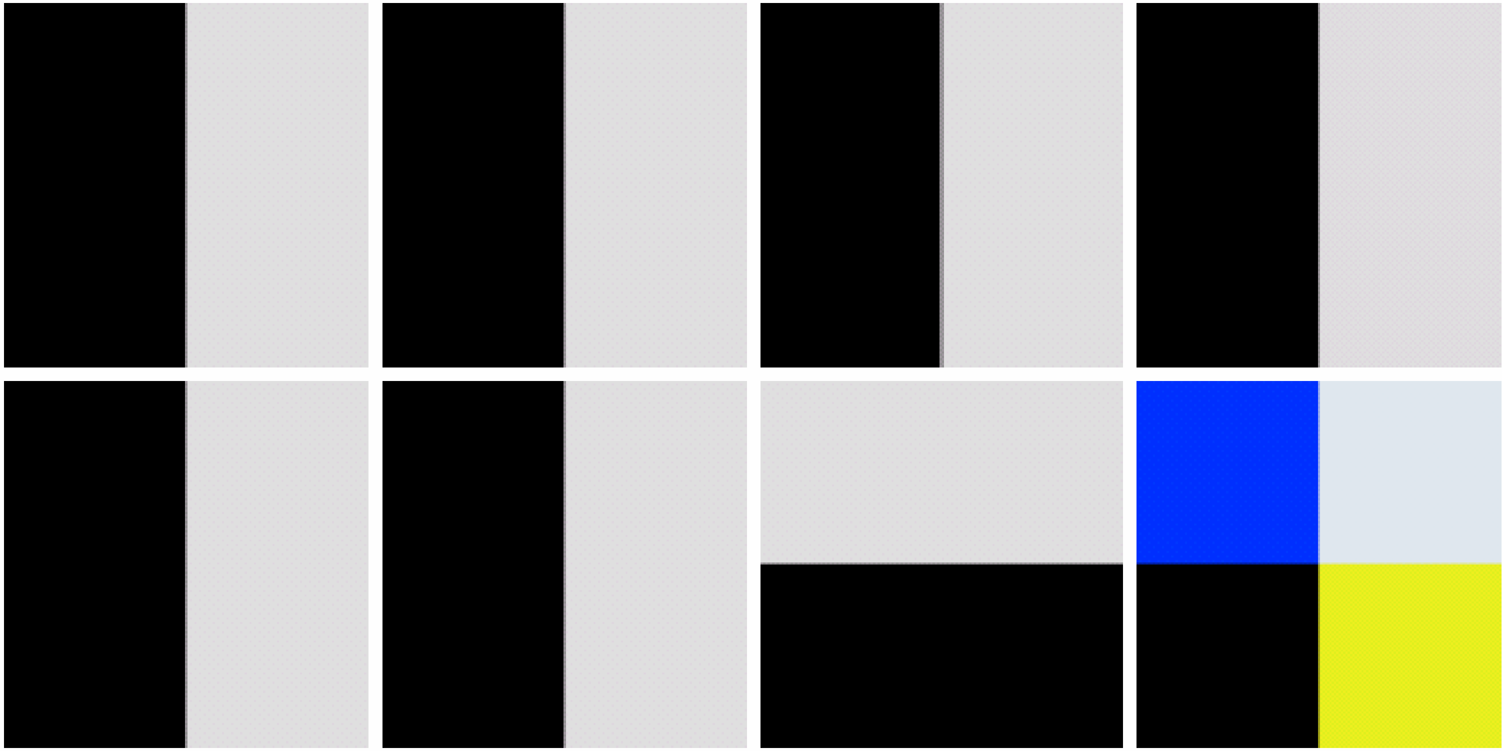
**FIGURE 6.44** Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 6.42(h).

The dimension of the box along R-axis extended from $(a_R\text{-}1.25\sigma_R)$ to $(a_R+1.25\sigma_R)$

# 6.7.3 Color Edge detection

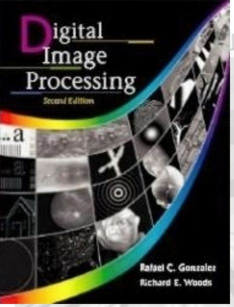
- The gradient operators introduced is effective for scalar image.
- Compute the gradient on individual images and then using the results to form a color image will lead to erroneous results.

# 6.7.3 Color Edge Detection



| | | | |
|---|---|---|---|
| a | b | c | d |
| e | f | g | h |

**FIGURE 6.45** (a)–(c) $R$, $G$, and $B$ component images and (d) resulting RGB color image. (f)–(g) $R$, $G$, and $B$ component images and (h) resulting RGB color image.




# 6.7.3 Color Edge Detection

- Let **r, g, b** be a unit vector along the *R*, *G*, *B* axis and define the unit vector as

$$\mathbf{u} = \frac{\partial R}{\partial x}\mathbf{r} + \frac{\partial G}{\partial x}\mathbf{g} + \frac{\partial B}{\partial x}\mathbf{b}$$

$$\mathbf{v} = \frac{\partial R}{\partial y}\mathbf{r} + \frac{\partial G}{\partial y}\mathbf{g} + \frac{\partial B}{\partial y}\mathbf{b}$$

- $g_{xx} = \mathbf{u} \cdot \mathbf{u} = |\partial R/\partial x|^2 + |\partial G/\partial x|^2 + |\partial B/\partial x|^2$
- $g_{yy} = \mathbf{v} \cdot \mathbf{v} = |\partial R/\partial y|^2 + |\partial G/\partial y|^2 + |\partial B/\partial y|^2$
- $g_{xy} = \mathbf{u} \cdot \mathbf{v} = (\partial R/\partial x)(\partial R/\partial y) + (\partial G/\partial x)(\partial G/\partial y) + (\partial B/\partial x)(\partial B/\partial y)$

# 6.7.3 Color Edge Detection

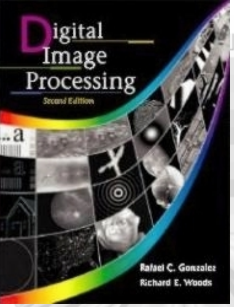
- The direction of maximum rate of change of **c**$(x, y)$ is given by the angle

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2g_{xy}}{(g_{xx} - g_{yy})}\right]$$

- The value of the rate of change at $(x, y)$ in the direction $\theta$ is

$$F(\theta)=\{0.5[(g_{xx}+g_{yy})+(g_{xx}-g_{yy})\cos\theta+2g_{xy}\sin\theta]\}^{1/2}$$

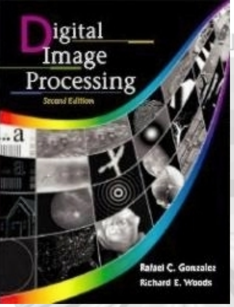
- There are two solved $\theta$ or $\theta + \pi/2$ in orthogonal directions.
- One generate maximum $F$ and the other generate minimum $F$.




# 6.7.3 Color Edge Detection

| a | b |
|---|---|
| c | d |

**FIGURE 6.46** (a) RGB image. (b) Gradient computed in RGB color vector space. (c) Gradients computed on a per-image basis and then added. (d) Difference between (b) and (c).





# 6.7.3 Color Edge Detection

a b c

**FIGURE 6.47** Component gradient images of the color image in Fig. 6.46. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 6.46(c).

# 6.8 Noise in Color Image

- The noise content of a color image has the same characteristics in each color channel.
- It is possible for color channels to be affected differently by noise.
- The fine grain noise (in Figure 6.48) tends to be less visually noticeable in a color image than it is in a monochrome image.

# 6.8 Noise in Color Image

a b
c d

**FIGURE 6.48** (a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

# 6.8 Noise in Color Image

a b c

**FIGURE 6.49** HSI components of the noisy color image in Fig. 6.48(d). (a) Hue. (b) Saturation. (c) Intensity.

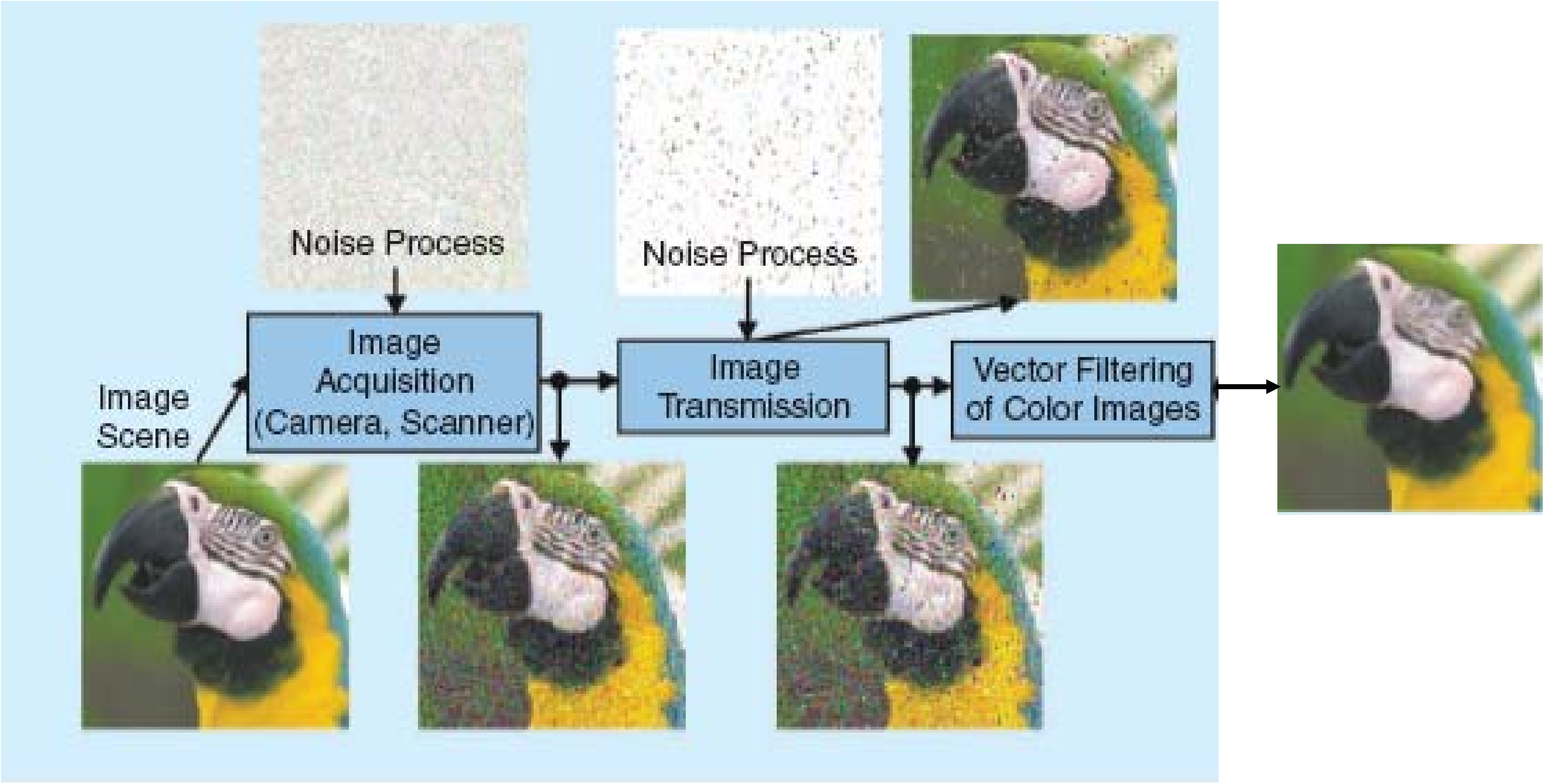
# 6.8 Noise in Color Image

a b
c d

**FIGURE 6.50** (a) RGB image with green plane corrupted by salt-and-pepper noise. (b) Hue component of HSI image. (c) Saturation component. (d) Intensity component.

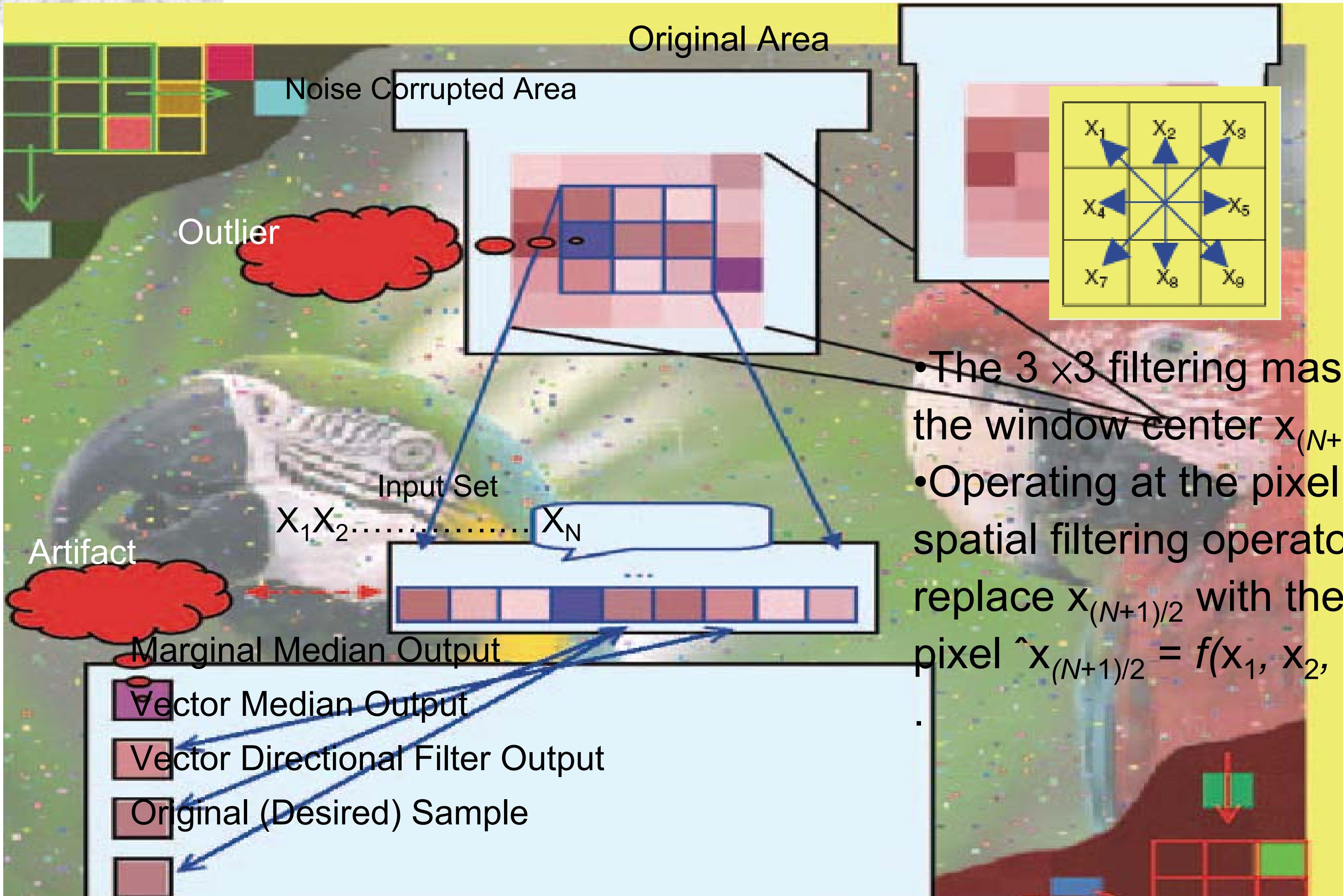
# 6.8 Noise in Color Image

- Vector filtering



# 6.8 Noise in Color Image

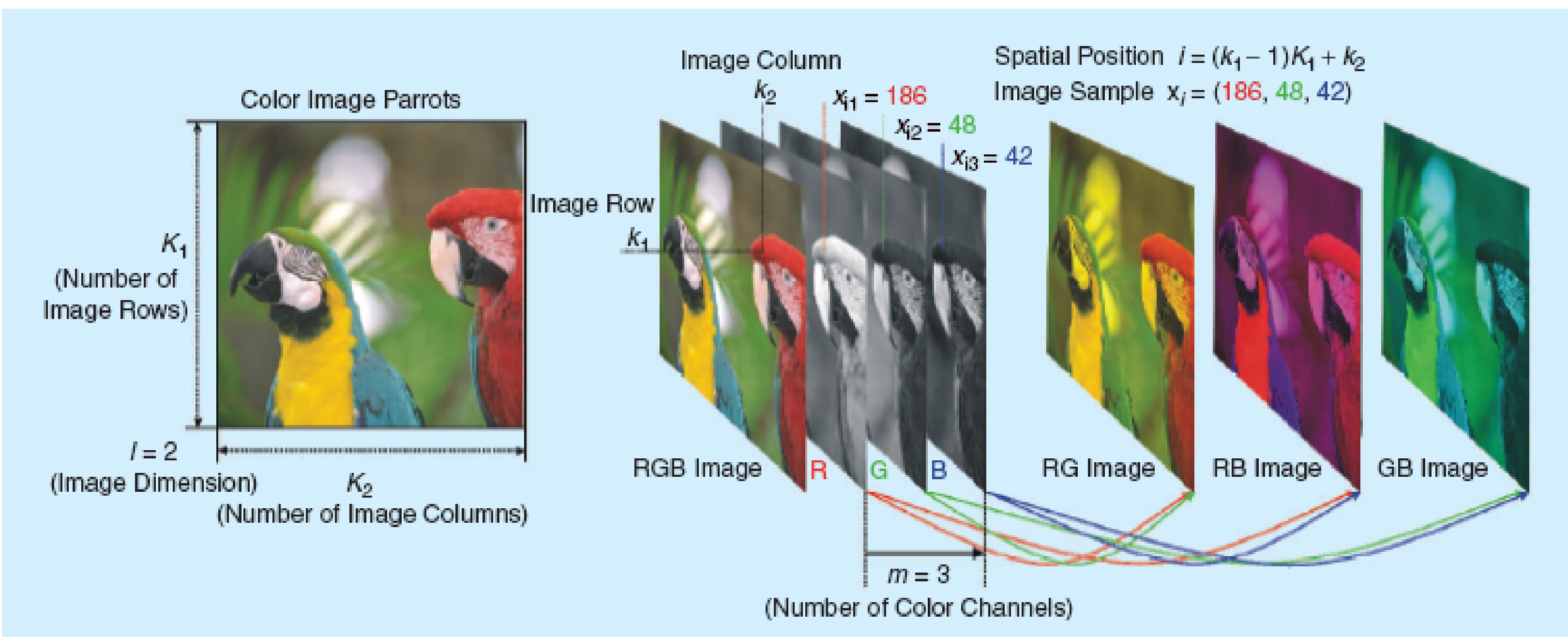
• w

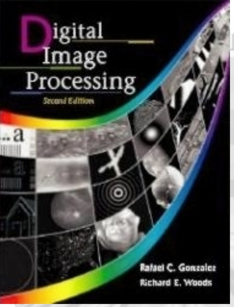



- The 3 × 3 filtering mask with the window center $x_{(N+1)/2} = x_5$.
- Operating at the pixel level, spatial filtering operators replace $x_{(N+1)/2}$ with the output pixel $\hat{x}_{(N+1)/2} = f(x_1, x_2, \ldots, x_N)$ .

# 6.8 Noise in Color Image

- Vector filtering techniques that treat the color image as a vector field are more appropriate.
- The filter output $\hat{\mathbf{x}}_{(N+1)/2}$ is a function of the vectorial inputs $\mathbf{x}_1$, $\mathbf{x}_2$, . . . , $\mathbf{x}_N$ located within the supporting window $W$ .
- A color red, green, blue (RGB) image $\mathbf{x} : Z^2 \rightarrow Z^3$, each pixel $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ represents a three-component vector in a color space








# 6.8 Noise in Color Image

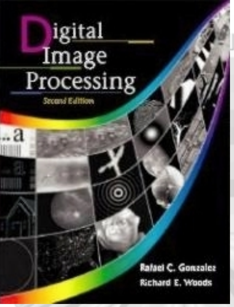
- The color image $\mathbf{x}$ is a vector array or a 2-D matrix of three component samples $\mathbf{x}_i$ with $x_{ik}$ denoting the R ($k = 1$), G ($k = 2$), or B component ($k = 3$).
- The chromatic properties of $\mathbf{x}_i$ is related to its magnitude

$$M\mathbf{x}i = \| \mathbf{x}_i \| = [(x_{i1})^2 + (x_{i2})^2 + (x_{i3})^2]^{1/2}$$

and direction (orientation in the vector space)

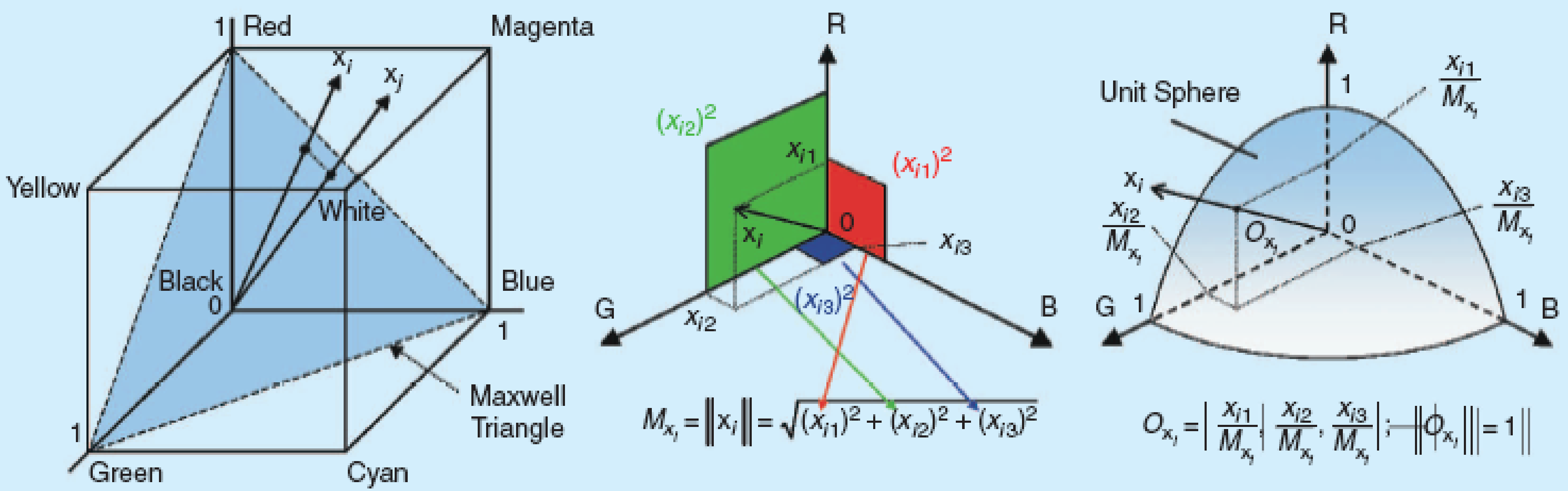
$$O\mathbf{x}i = \mathbf{x}_i / \| \mathbf{x}_i \| = \mathbf{x}_i / M\mathbf{x}i \text{ , with } \| O\mathbf{x}_i \| = 1.$$

- Both the magnitude and the direction can be used in classifying the differences between two vectorial inputs.




# 6.8 Noise in Color Image

- (a) RGB color cube and (b), (c) the basic parameters related to the RGB color vector $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ .
- (b) The magnitude $M_{\mathbf{x}i}$ .
- (c) The orientation defined as the point $O_{\mathbf{x}i}$ on unit sphere.




(a) (b) (c)

# 6.8 Noise in Color Image

- Distance and similarity measures
- The distance between two color vectors $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ and $\mathbf{x}_j = [x_{j1}, x_{j2}, x_{j3}]^T$ in the magnitude domain is the ***generalized weighted Minkowski metric***

$$d(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i - \mathbf{x}_j \|_L = c\left(\sum_{k=1}^{3} \xi_k \left|\mathbf{x}_{ik} - \mathbf{x}_{jk}\right|^L\right)^{1/L}$$

- The nonnegative scaling parameter $c$ is a measure of the overall discrimination power.
- The exponent $L$ defines the nature of the distance metric, *i.e.*, $L = 1$ (city-block distance), $L = 2$ (Euclidean distance), $L \to \infty$ (The chess-board distance)
- The distance between the two 3-D vectors is considered equal to the maximum distance among their components.
- The parameter $\xi_k$ measures the proportion of attention allocated to the dimensional component $k$ and, therefore, $\Sigma_k \xi_k = 1$.
- Vectors having a range of values greater than a desirable threshold can be scaled down by the use of the weighting function $\xi$ .

# 6.8 Noise in Color Image

- Opposite to distance measures, a similarity measure $s(\mathbf{x}_i, \mathbf{x}_j)$ is defined as a symmetric function whose value is large when the vectorial inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar.
- Similarity in orientation is expressed through the normalized inner product $s(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \mathbf{x}_j^T / (|\mathbf{x}_i||\mathbf{x}_j|) \rightarrow$ the cosine of the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- Since similar colors have almost parallel orientations and significantly different colors point in different overall directions in a 3-D color space, the normalized inner product, or equivalently the angular distance $\theta = \arccos(\mathbf{x}_i \mathbf{x}_j^T / (|\mathbf{x}_i||\mathbf{x}_j|)$, is used to quantify the dissimilarity (here the orientation difference) between the two vectors.

# 6.8 Noise in Color Image

- Vector median filters (VMF)
- The VMF is a vector processing operator that has been introduced as an extension of the scalar median filter.
- The generalized Minkowski metric $\| \mathbf{x}_i - \mathbf{x}_j \|_L$ is used to quantify the distance between two color pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ in the magnitude domain.
- The VMF output is the sample $\mathbf{x}_{(1)} \in W$ that minimizes the distance to the other samples inside $W$ as

$$\mathbf{x}_{(1)} = \arg\min_{\mathbf{x}_i \in W} \sum_{j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|_L .$$

# 6.8 Noise in Color Image

- Vector directional filters (VDFs)
- VDF represents a different type of vector processing filter.
- VDF operates on the directions of image vectors, aiming at eliminating vectors with ***atypical directions*** in the color space.
- The Basic VDF (BVDF) operates in the directional domain of a color image, its output is the color vector $\mathbf{x}_{(1)} \in W$ whose direction is the *MLE* of directions of the input vectors.
- The BVDF output $\mathbf{x}_{(1)}$ minimizes the angular ordering criteria to other samples inside the sliding filtering window *W*:

$$\mathbf{x}_{(1)} = \arg\min_{\mathbf{x}_i \in W} \sum_{j=1}^{N} \theta(\mathbf{x}_i, \mathbf{x}_j).$$

where $\theta(\mathbf{x}_i, \mathbf{x}_j)$ represents the angle between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

# 6.8 Noise in Color Image

- **Algorithm of VMF or BVDF outputting the lowest ranked vector.**

Inputs: $NumberOfRows \times NumberOfColumns$ image
Window size $N$
Moving window spawning the input set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
Output: $NumberOfRows \times NumberOfColumns$ image

For $a = 1$ to $NumberOfRows$
For $b = 1$ to $NumberOfColumns$
Determine the input set $W(a,b) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
For $i = 1$ to $N$
Let the aggregated distance $D_i = d(\mathbf{x}_i, \mathbf{x}_1) + d(\mathbf{x}_i, \mathbf{x}_2) + \ldots + d(\mathbf{x}_i, \mathbf{x}_N)$
End
Sort scalars $D_1, D_2, \ldots, D_N$ to the ordered set $D_{(1)} \le D_{(2)} \le \ldots \le D_{(N)}$
Apply the same ordering scheme to the vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$
Store the ordered sequence as $\mathbf{x}_{(1)} \le \mathbf{x}_{(2)} \le \ldots \le \mathbf{x}_{(N)}$
Let the filter output $\mathbf{y}(a,b) = \mathbf{x}_{(1)}$
End
End

# 6.8 Noise in Color Image

- Data Adaptive Filter
- The general form of the data-dependent filter is given as a fuzzy weighted average of the input vectors inside the supporting window $W$

$$\hat{\mathbf{x}}_{(N+1)/2} = f\left(\sum_{i=1}^{N} w_i^* \mathbf{x}_i\right) = f\left(\sum_{i=1}^{N} w_i \mathbf{x}_i \Big/ \sum_{i=1}^{N} w_i\right)$$

- where $f(\cdot)$ is a nonlinear function that operates over the weighted average of the input set, and $w_i$ is the filter weight equivalent to the fuzzy membership function associated with the input color vector $\mathbf{x}_i$. with the constraints $w_i^* \geq 0$ and $\Sigma w_i^* = 1$.
- The weights $w_i$ are determined adaptively using functions of a distance criterion between the input vectors as

  $w_i = \beta\left(1 + \exp\left\{\sum_{j=1}^{N} d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right\}\right)^{-r}$, where $r$ is a parameter adjusting the weighting effect of the membership function, and $\beta$ is a normalizing constant.

# 6.8 Noise in Color Image

- Based on the difference between the observation (noisy) color vector $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ and the original (desired) sample $\mathbf{o}_i = [o_{i1}, o_{i2}, o_{i3}]^T$, the noise corruption is modeled via the additive noise model defined as follows:

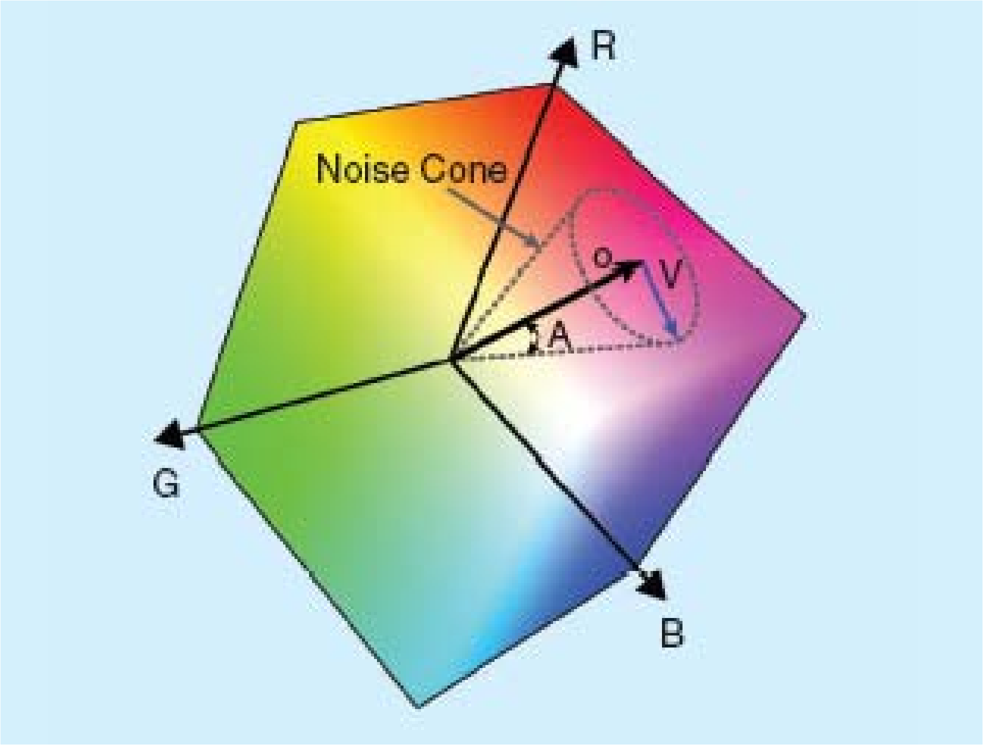
$$\mathbf{x}_i = \mathbf{o}_i + \mathbf{v}_i$$

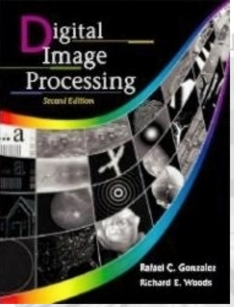
where $\mathbf{v}_i = [v_{i1}, v_{i2}, v_{i3}]^T$ is the vector describing the noise process and $i$ denotes the spatial position of the samples in the image. Note that $\mathbf{v}_i$ can describe either signal-dependent or independent noise.

- Considering the likely presence of many noise sources, it is reasonable to assume that the overall noise process can be modeled as a zero mean white Gaussian, affecting each color component and pixel position independently.
- The noise variance $\sigma$ is the same for all three color components in a correlated color space, such as RGB.

# 6.8 Noise in Color Image

- Angular noise margins for a color signal.








# 6.8 Noise in Color Image

- The noise can be reduced to a scalar perturbation, the magnitude of the noise vector $p_i = \| \mathbf{v}_i \| = \sqrt{v_{i1}^2 + v_{i2}^2 + v_{i3}^2}$
- It follows that the distribution of the $p_i$s is:

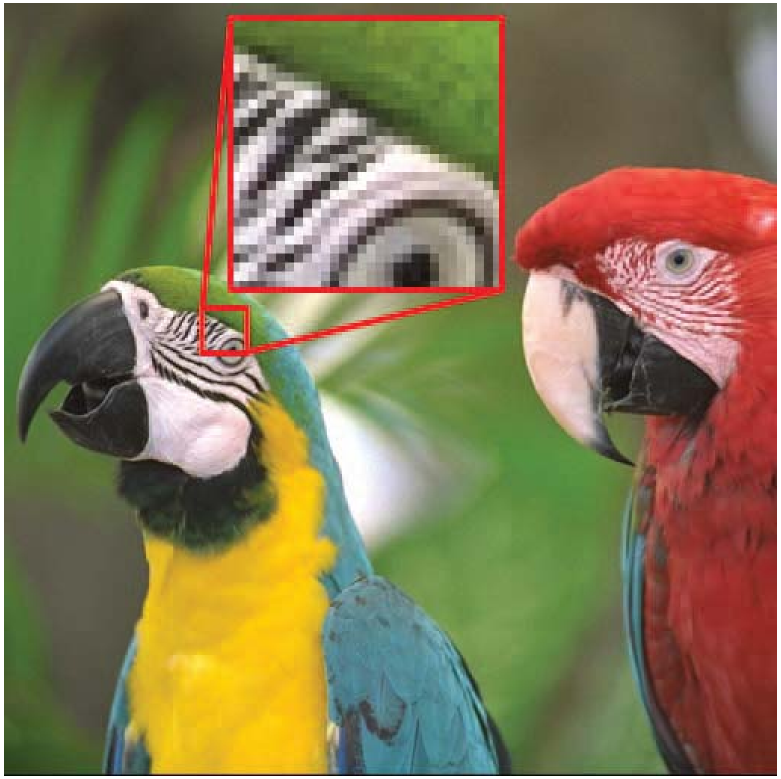
$$Pr(p) = \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^3 4\pi p^2 e^{-\frac{p^2}{2\sigma^2}}$$

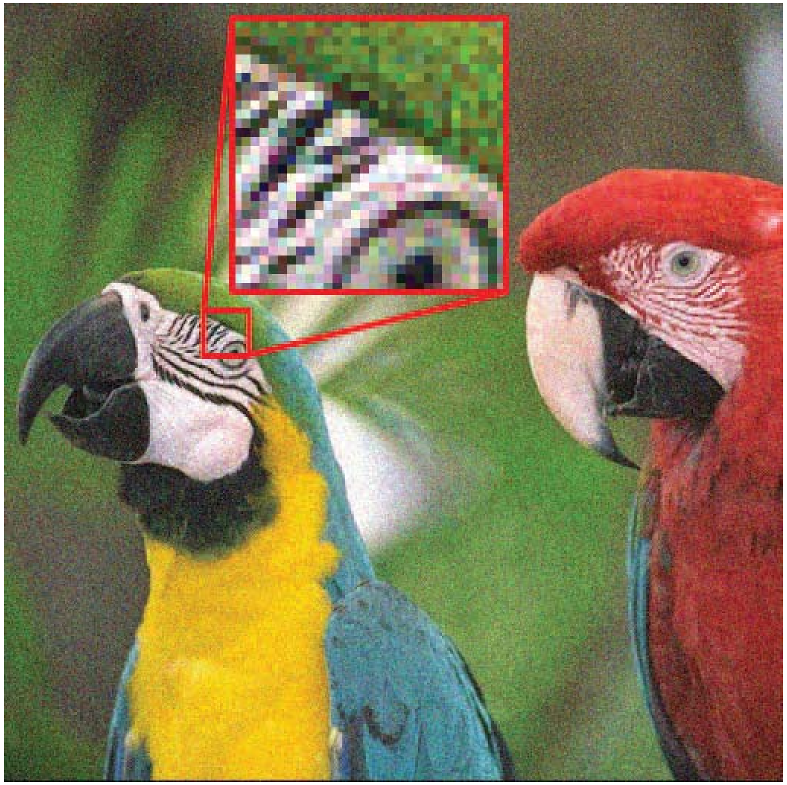
- This perturbation results in a "noise cone" in the RGB color space.
- This vector magnitude perturbation can be translated into an angular perturbation $A$.
- Assuming $\| \mathbf{o}_i \| \gg \sigma$, $A$ can be approximated to have the distribution

  $Pr(A) \approx A(\| \mathbf{o} \|^2 / \sigma^2)\exp\{-(\| \mathbf{o} \|^2 A^2/(2\sigma^2))\}$.
- This is a Rayleigh distribution with mean $\overline{A} \approx \sqrt{\sigma^2 \pi / (2\|\mathbf{o}\|^2)}$
- Using this concept of color noise as an angular perturbation of the original color vector represented in a correlated vector color space, the effect of the median operator can be roughly derived.

# 6.8 Noise in Color Image

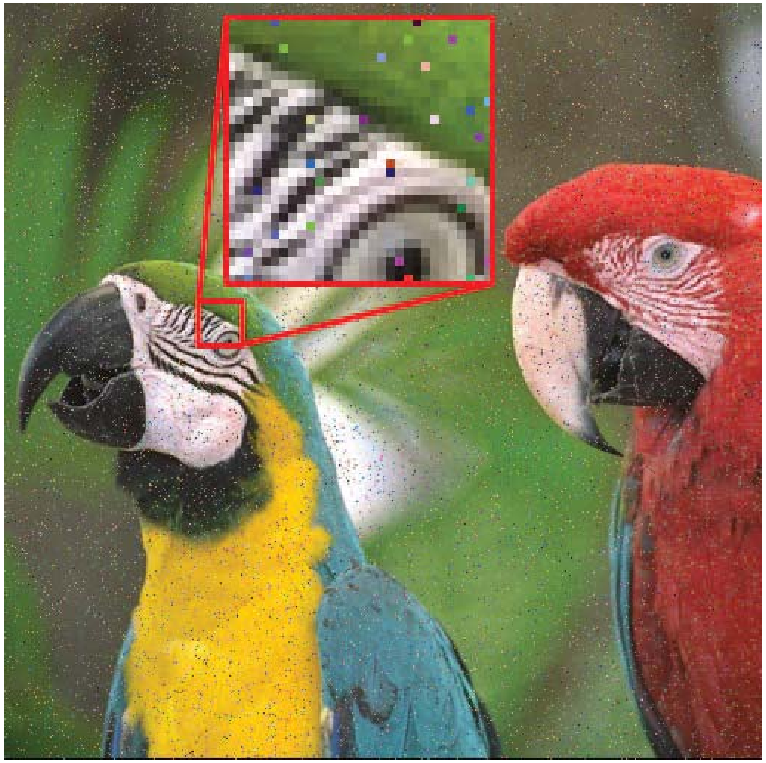
- Test image Parrots (512 $\times$ 512) corrupted by different kinds of noise: (a) original image, (b) additive Gaussian noise with $\sigma$ = 20,

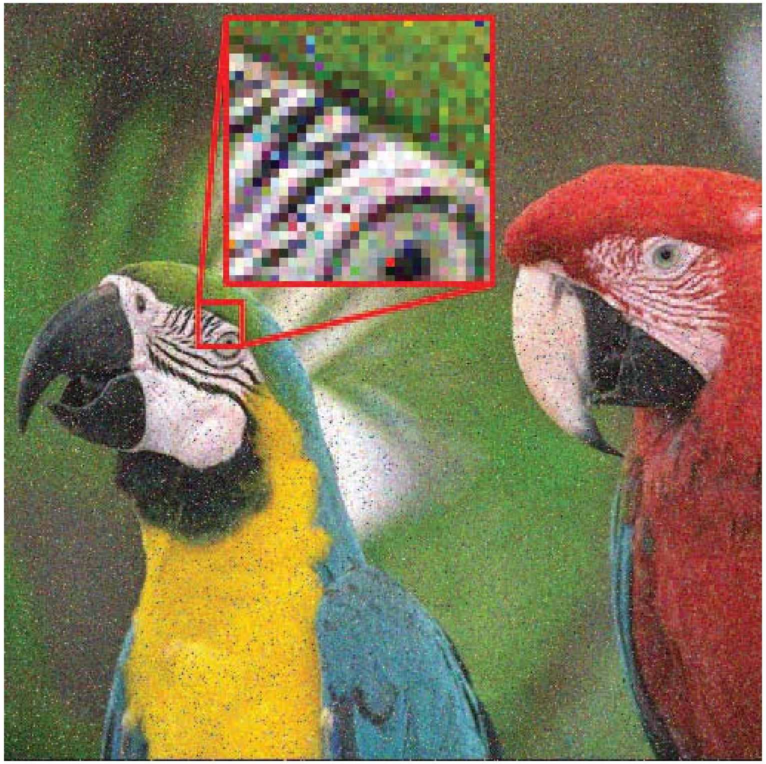


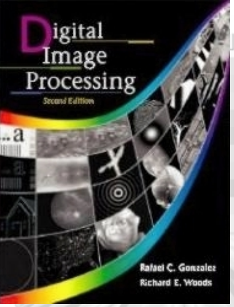


# 6.8 Noise in Color Image

- (c) 5% impulsive noise, (d) mixed noise (additive Gaussian noise of $\sigma$ = 20 followed by 5% impulsive noise).





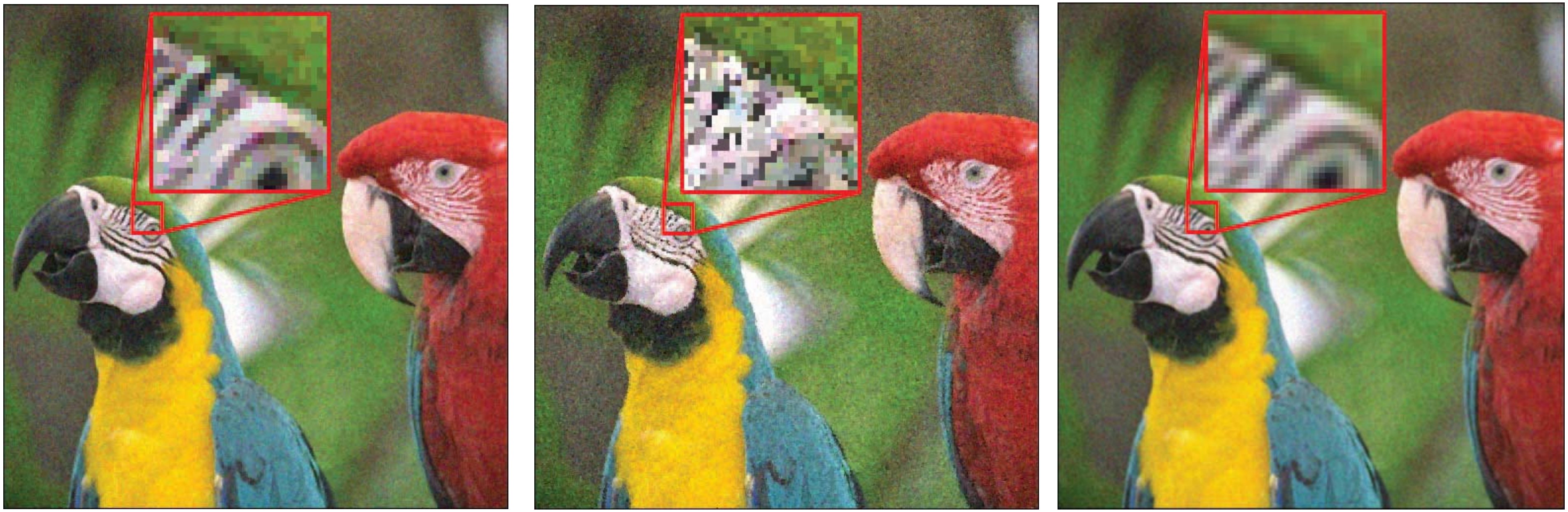






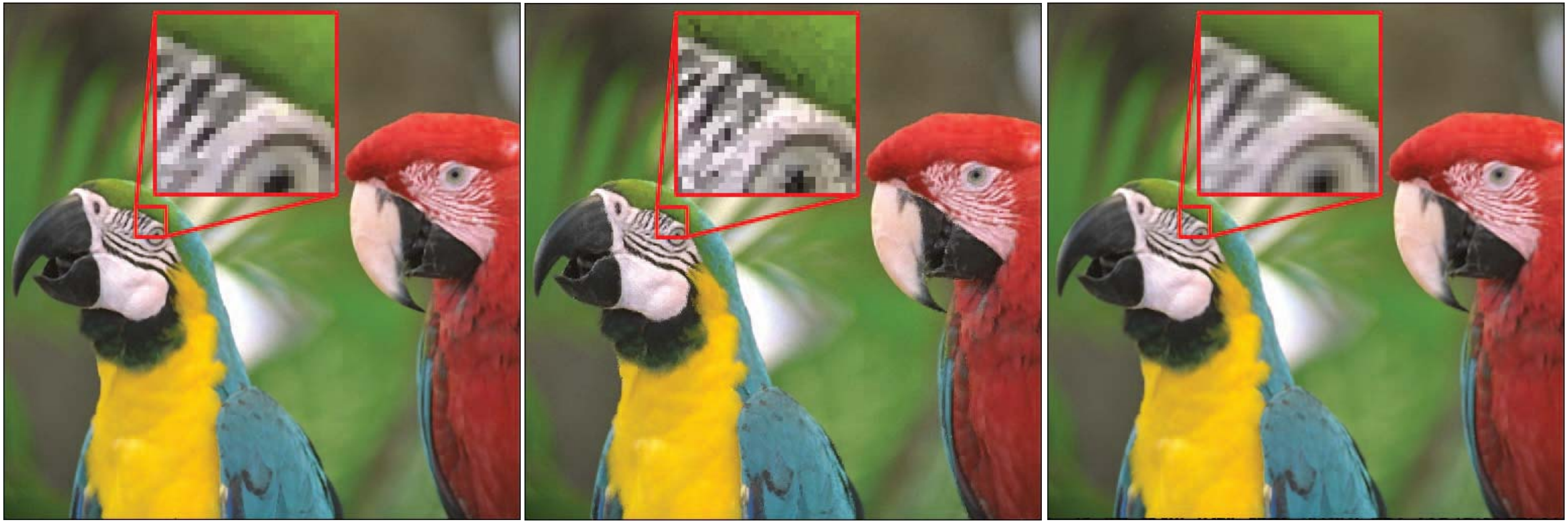
# 6.8 Noise in Color Image

- Additive Gaussian noise ( $\sigma$ = 20) filtered output. (a) VMF, (b) BVDF, and (c) data adaptive filter utilizing the angular distance measure.



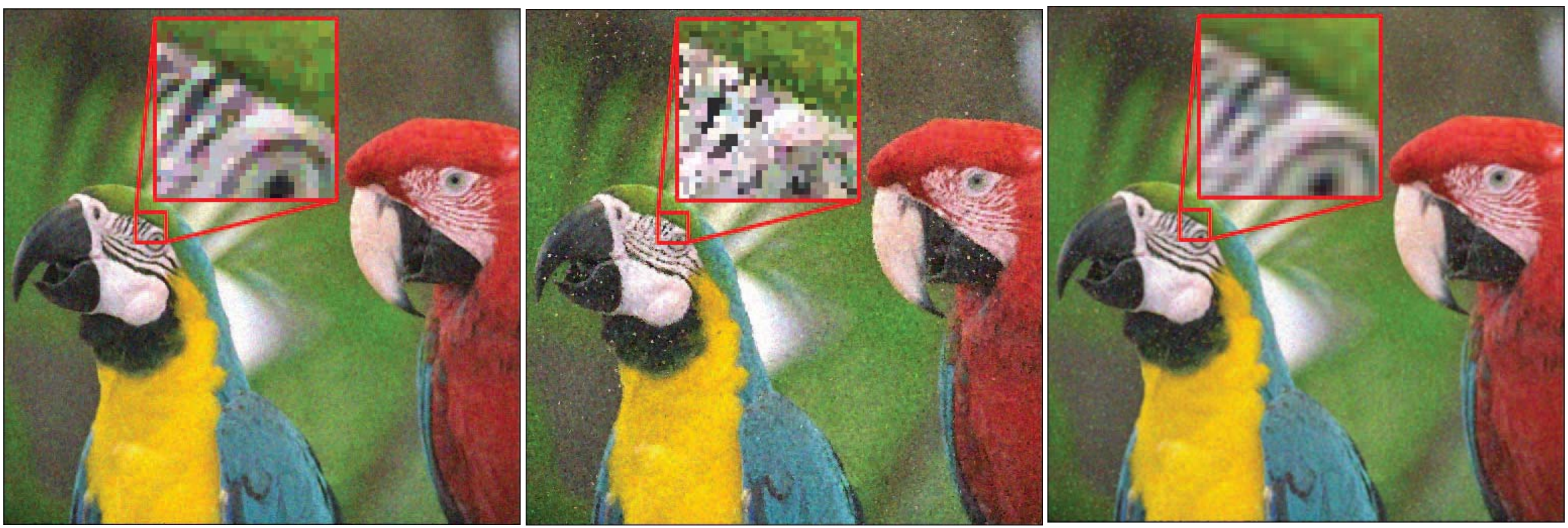
# 6.8 Noise in Color Image

- 5% impulsive noise filtered output. (a) VMF, (b) BVDF, and (c) data adaptive filter utilizing the angular distance measure.



# 6.8 Noise in Color Image

- Mixed noise filtered output (Gaussian with $\sigma$ = 20 and 5% impulsive noise. (a) VMF, (b) BVDF, and (c) data adaptive filter utilizing the angular distance measure.



# 6.9 Color Image Compression

Using JPEG 2000, the compression ratio is 1:230.

a b
c d

**FIGURE 6.51** Color image compression. (a) Original RGB image. (b) Result of compressing and decompressing the image in (a).