Digital Image Processing

Chap 6: Color Image Processing

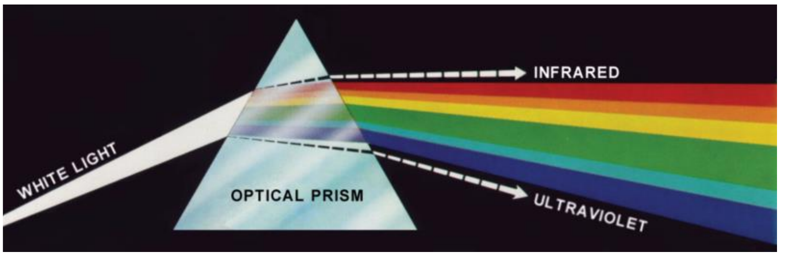
清華大學電機系林嘉文
cwlin@ee.nthu.edu.tw
Tel: 03-5731152

Color Image Processing

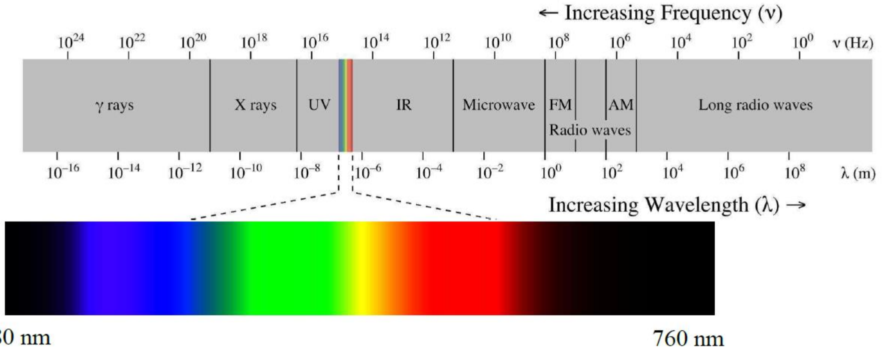
- **Color** is a powerful descriptor
- Human can discern thousands of color shades.
- "color" is more pleasing than "black and white“.
- **Full Color**: color from full-color sensor, i.e., CCD camera
- **Pseudo color**: assign a color to a particular monochromatic intensity.

Color Fundamentals

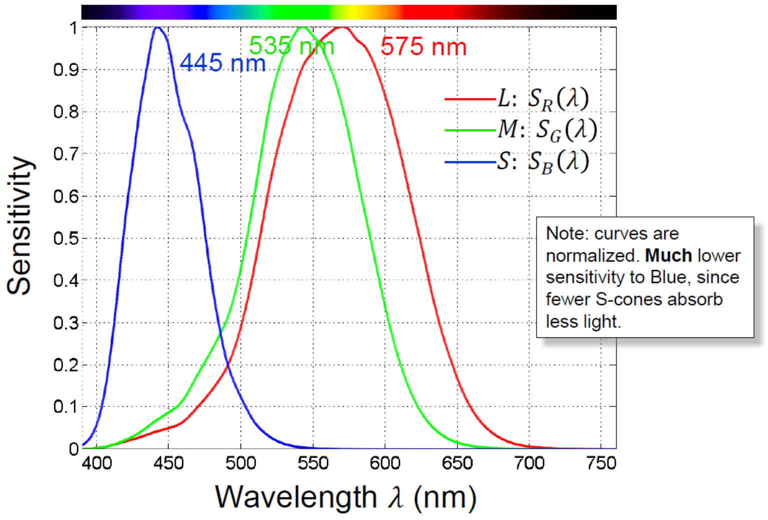
FIGURE 6.1
Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lighting Division.)



Visible Range of Electromagnetic Spectrum



Absorption of Lights in Human Retina



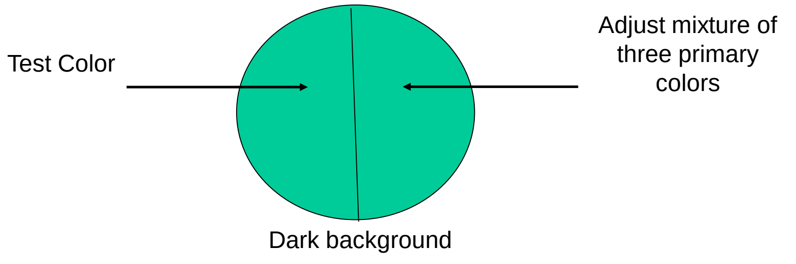
Color Fundamentals

- 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 eyes
- 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.

Color Fundamentals

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



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:
  - Linearity
  - Additivity

Color Fundamentals

- linearity:

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

- additivity :

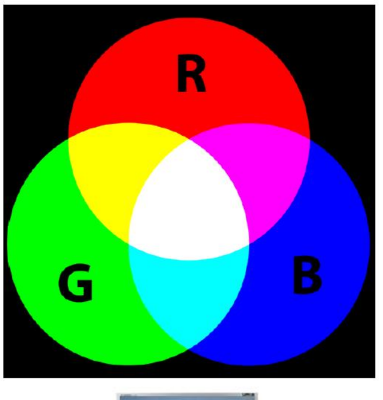
  If $S_1(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_2(\lambda)$ and $S_3(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_4(\lambda)$

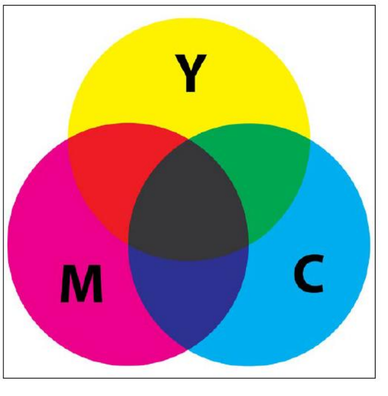
  then $S_{1(\lambda)} + S_3(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_{2(\lambda)} + S_4(\lambda)$

- Color with negative tri-stimulus values:

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

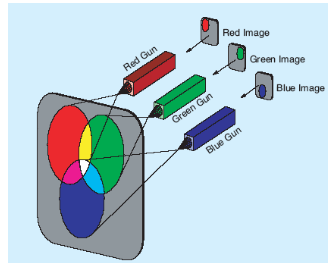
Additive vs. Subtractive Color Mixing

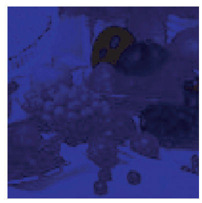




Additive Color System

- Additive Color System
- Primary: RGB

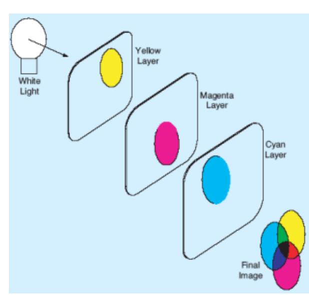


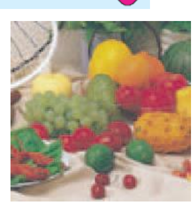
   

Red + Green + Blue = Color

Subtractive Color System

- Subtractive Color System:
- Primary: Cyan (青藍)
  Magenta (紫紅)
  Yellow



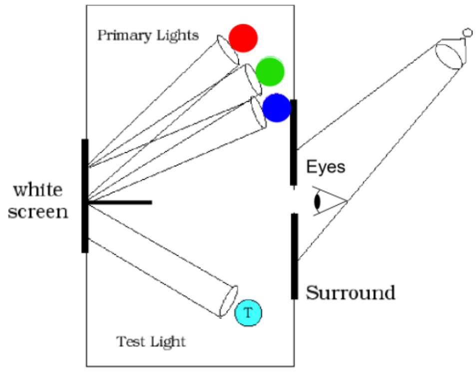
White − Yellow − Cyan − Magenta = Color

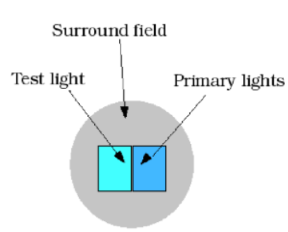
CMY vs. CMYK

CMYK:

- To save the ink cost, as black ink is cheaper
- To produce deeper black tones, unsaturated and dark colors are produced by using black ink instead of the combination of cyan, magenta and yellow

Color Matching Experiment





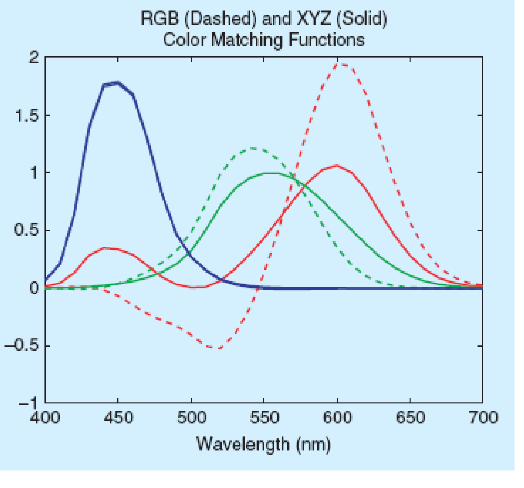
Color Matching Function

- **Color matching function**: the tristimulus values of the **spectral color** with unit intensity light of **single** wavelength.
- The primary colors are the spectral colors of wavelength (CIE 1931):

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

Color Matching Function

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



Color Matching Function

- Any color $S(\lambda)$ 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** $r(\lambda)$, $g(\lambda)$, $b(\lambda)$

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

Color Matching Function

- 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** : $S_1(\lambda) \neq S_2(\lambda), \ S_1(\lambda) \xleftrightarrow[\text{match}]{\text{color}} S_2(\lambda)$
- **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

Color Matching Function

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

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

Since $r+g+b=1$, the 3-D color reduces to 2-D color information → **chromaticity**

The 3rd information is the **luminance**

Color Matching

- Y (luminance) →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)

Color Matching

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

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

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

- Brightness match

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

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

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.

Standard CIE Color System

- CIE also defines three new primaries : *X*, *Y*, *Z*

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

- By matrix inversion, we obtain

$$R_0 = 0.490X + 0.177Y$$
$$G_0 = 0.310X + 0.813Y + 0.01Z$$
$$B_0 = 0.200X + 0.010Y + 0.990Z \quad \text{(b)}$$

- *Y* tristimulus value corresponds to the normalized luminance.

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**:

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

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

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

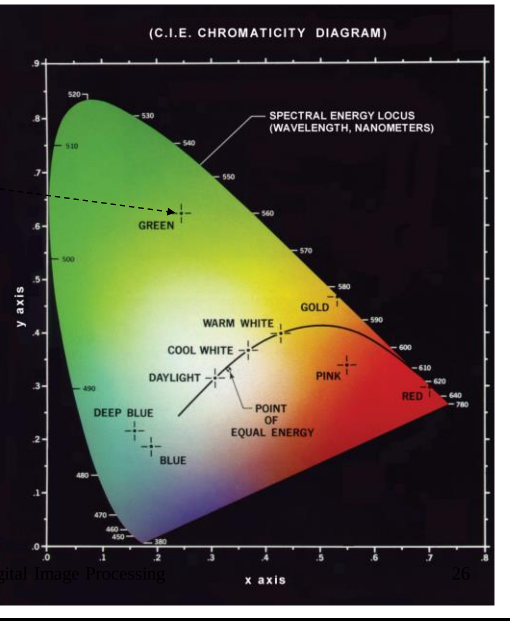
Standard CIE Color System

- 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 ≠ color matching function

Chromaticity Diagram

FIGURE 6.5 The CIE chromaticity diagram. (Courtesy of the General Electric Co., Lighting Division.)

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



Typical Color Gamut

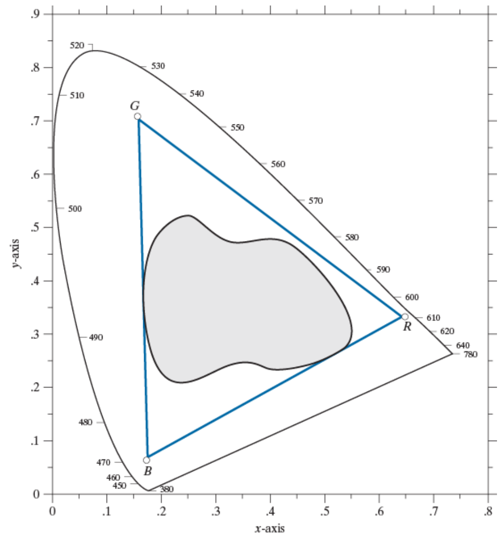
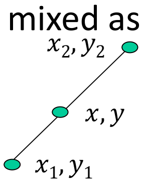


FIGURE 6.6 Illustrative color gamut of color monitors (triangle) and color printing devices (shaded region).

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



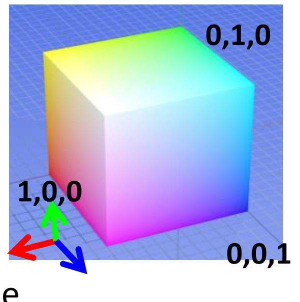
$$x = \frac{x_1(Y_1/y_1) + x_2(Y_2/y_2)}{(Y_1/y_1) + (Y_2/y_2)} \qquad y = \frac{Y_1 + Y_2}{(Y_1/y_1) + (Y_2/y_2)}$$

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, Saturation): decouples the color and gray-scale information.

RGB Color Model

Default color space



RGB cube

- Easy for devices
- But not perceptual
- Where do the grays live?
- Where is hue and saturation?

R
(G=0,B=0)

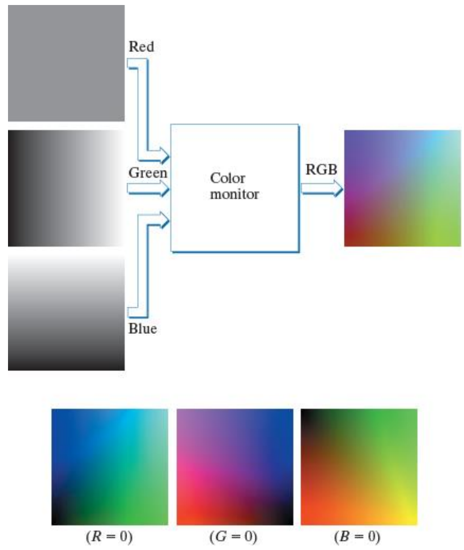
G
(R=0,B=0)

B
(R=0,G=0)

RGB Color Model

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.

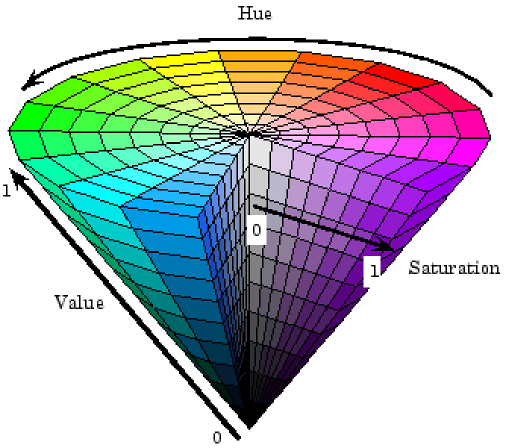


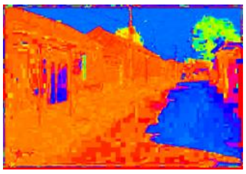
HSI Color Model

- Human describes a color in terms of **Hue (色相、色調)**, **Saturation (色度、飽和度)** and **Brightness (明度)**.
- **Hue**: describes the purity of 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.

HSI Color Model

Intuitive color space





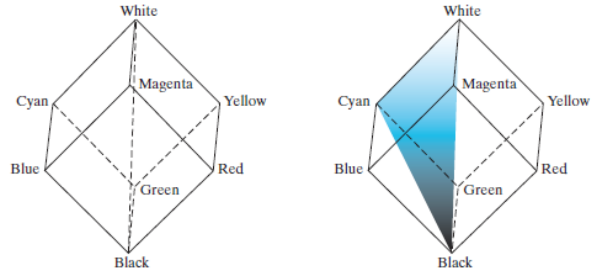
H
(S=1,V=1)

S
(H=1,V=1)

I
(H=1,S=0)

HSI Color Model

a b
FIGURE 6.10 Conceptual relationships between the RGB and HSI color models.



HSI Color Model

- From **RGB** to **HSI**

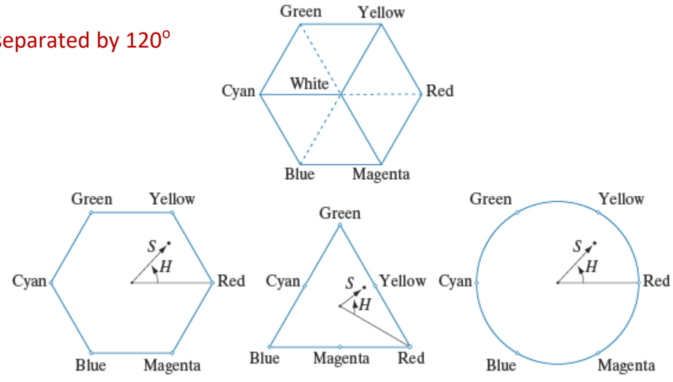
$$H = \begin{cases} \theta & \text{if} \quad B \le G \\ 360^o - \theta & \text{if} \quad B > G \end{cases}$$

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·min(R,G,B) / (R+G+B)]
- I = (R+G+B)/3

HSI Color Model

Primary colors are separated by 120^o



a
b c d

FIGURE 6.11 Hue and saturation in the HSI color model. The dot is any color point. The angle from the red axis gives the hue. 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.

HSI Color Model

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

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

HSI Color Model

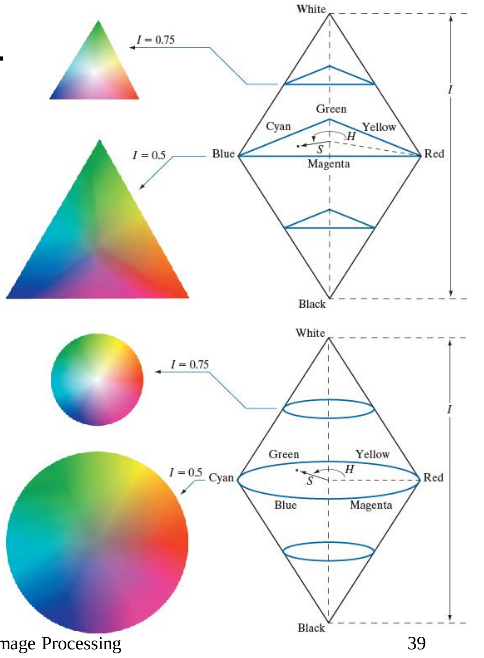
- **G/B sector**
  $(120^o \le H < 240^o)$
- $\min(R,G,B) = R$
- $H = H - 120°$
- $R = I\,(1-S)$
- $G = I\left[1 + \frac{S\cos H}{\cos(60°-H)}\right]$
- $B = 3I - (R+G)$

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

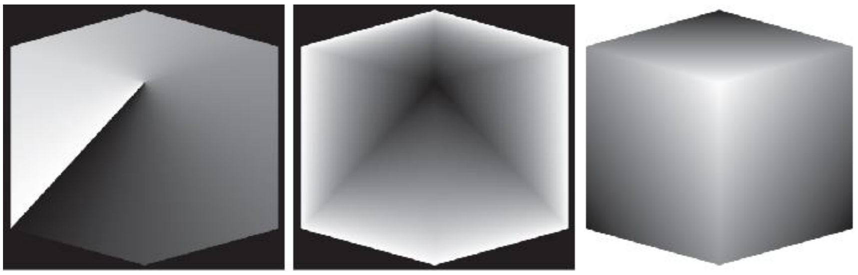
HSI Color Model

a
b

FIGURE 6.12 The HSI color model based on (a) triangular, and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis.



HSI Color Model



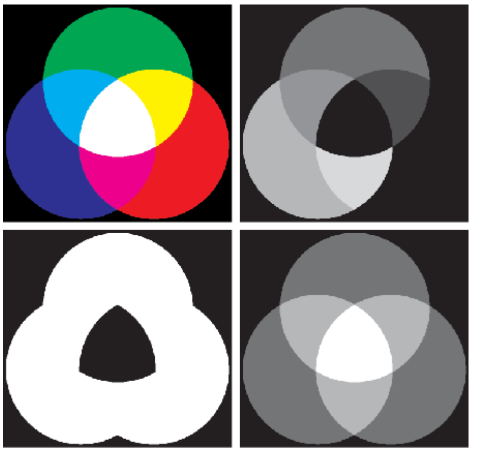
a b c

FIGURE 6.13 HSI components of the image in Fig. 6.8: (a) hue, (b) saturation, and (c) intensity images.

HSI Color Model

a b
c d

FIGURE 6.14 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.



HSI Color Model

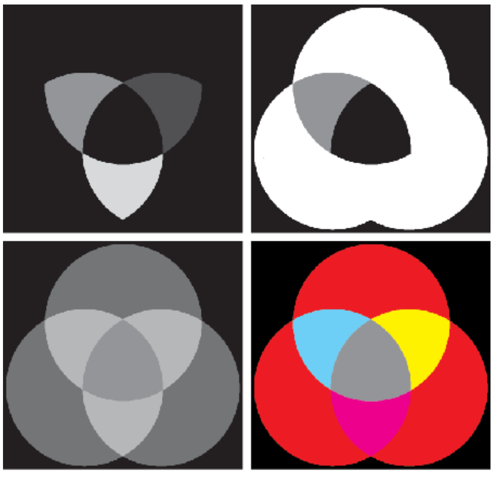
a b
c d

FIGURE 6.15 (a)-(c) Modified HSI component images. (d) Resulting RGB image. (See Fig. 6.14 for the original HSI images.)

6.14(b)→6.15(a)

6.14(c)→6.15(b)

6.14(d)→6.15(c)



Pseudo Image Processing - Intensity Slicing

- 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 (Fig. 6.16).
- In general, we assume that P planes perpendicular to the intensity axis defined at level l_i $l = 1, 2, \dots, P$.
- These P planes partition the gray level in to $P + 1$ intervals:

$$V_k, k = 1, 2, \dots, P + 1$$

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

where c_k is the color associated with the kth intensity interval V_k defined by the partition planes at $l = k–1$ and $l = k$.

- From Fig. 6.16; if more levels are used, the mapping function takes on a staircase form.

Intensity Slicing

FIGURE 6.16 Graphical interpretation of the intensity-slicing technique.

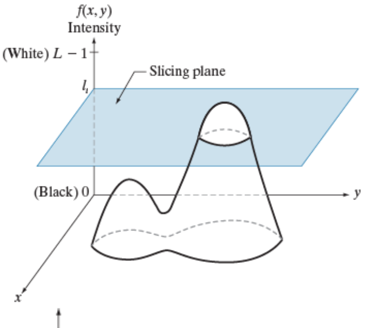
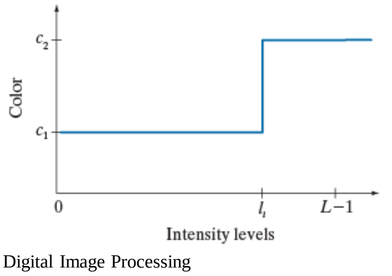


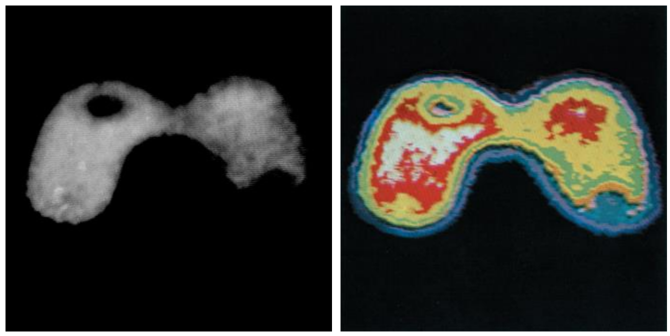
FIGURE 6.17 An alternative representation of the intensity-slicing technique.



Intensity Slicing

a b

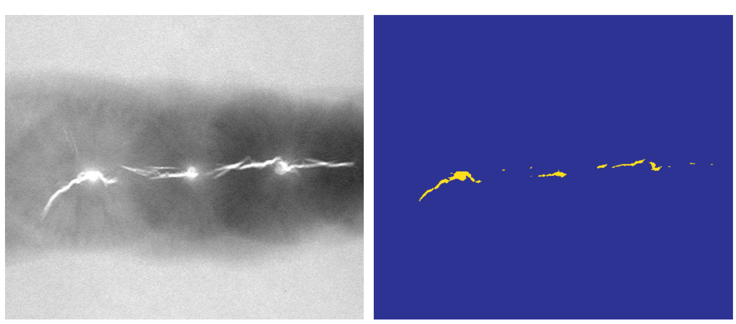
FIGURE 6.18 (a) Grayscale image of the Picker Thyroid Phantom. (b) Result of intensity slicing using eight colors. (Courtesy of Dr. J. L. Blankenship, Oak Ridge National Laboratory.)



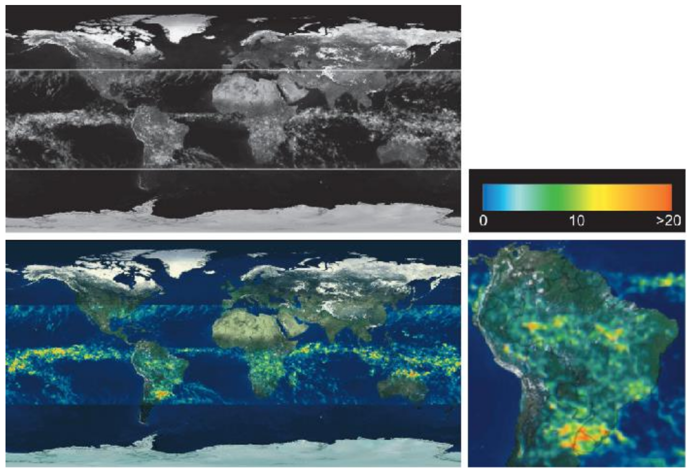
Intensity Slicing

a b

FIGURE 6.19 (a) X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)



Intensity Slicing



a b
c d

FIGURE 6.20 (a) Grayscale image in which intensity (in the horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South American region. (Courtesy of NASA.)

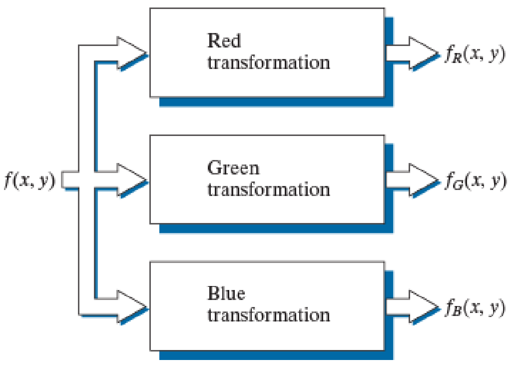
Intensity to Color Transformation

- Three independent transformation functions on the gray-level of each pixel.

  Red, **Green**, **Blue** transform
- Piecewise linear function
- Smooth non-linear function

Intensity to Color Transformation

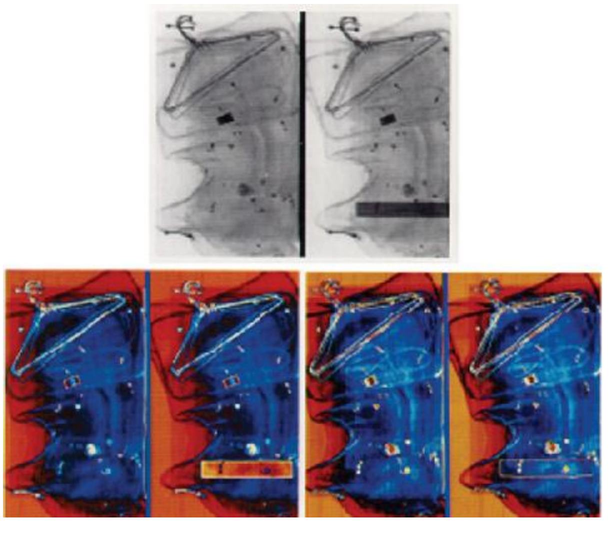
FIGURE 6.21 Functional block diagram for pseudocolor image processing. Images f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.



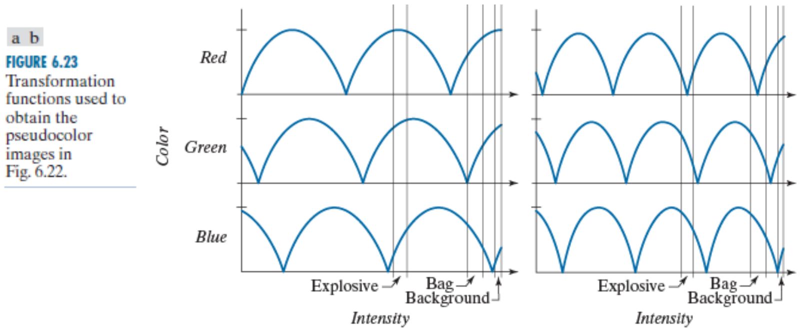
Intensity to Color Transformation

a
b c

FIGURE 6.22 Pseudocolor enhancement by using the gray level to color transformations in Fig. 6.23. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)



Intensity to Color Transformation



Intensity to Color Transformation

- Change the **phase** and **frequency** of each sinusoid can emphasize (in color) ranges in the gray scale.

  Peak → constant color region.

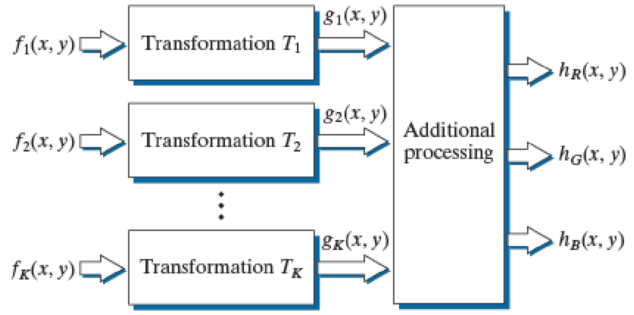
  Valley → rapid changed color region.

- 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.

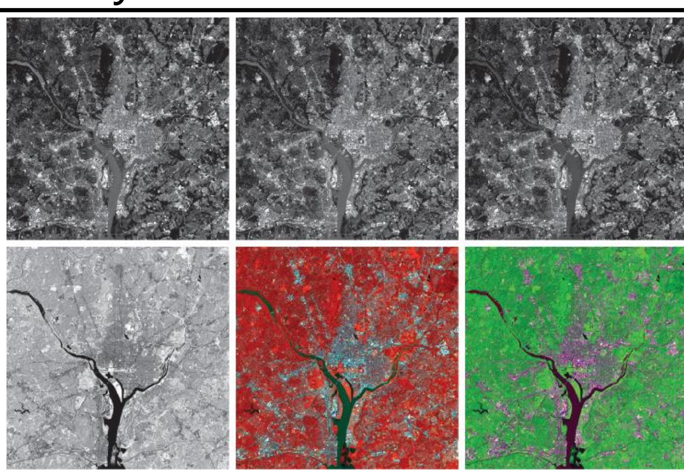
Intensity to Color Transformation

Combine several monochrome images into a single color image.

FIGURE 6.24 A pseudocolor coding approach using multiple grayscale images. The inputs are grayscale images. The outputs are the three components of an RGB composite image.



Intensity to Color Transformation



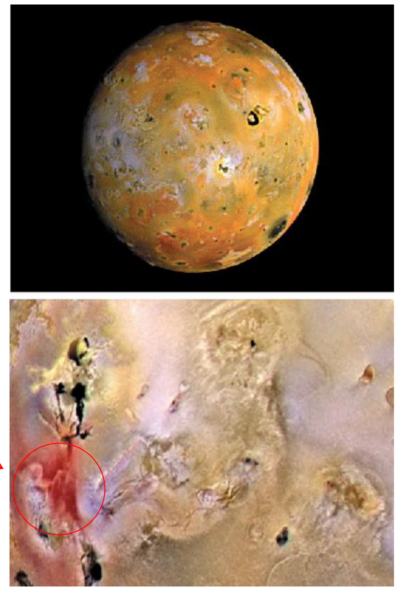
a b c
d e f

FIGURE 6.25 (a)–(d) Red (R), green (G), blue (B), and near-infrared (IR) components of a LANDSAT multispectral image of the Washington, D.C. area. (e) RGB color composite image obtained using the IR, G, and B component images. (f) RGB color composite image obtained using the R, IR, and B component images. (Original multispectral images courtesy of NASA.)

Intensity to Color Transformation

a
b

FIGURE 6.26
(a) Pseudocolor rendition of Jupiter Moon Io. (b) A close-up. (Courtesy of NASA.)



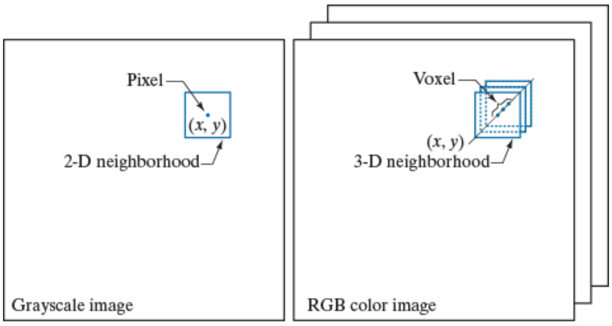
material ejected from the active volcano

Full-Color Image Processing

- Color processing in two categories:
  1. Process each **component** (R/G/B) individually and then form a **composite** processed color image from the **components**.
  2. Work with color pixels directly.

  In RGB system, each color point can be interpreted as a vector, i.e., $\mathbf{c}(x, y) = [c_R(x, y), c_G(x, y), c_B(x, y)]$

Full-Color Image Processing



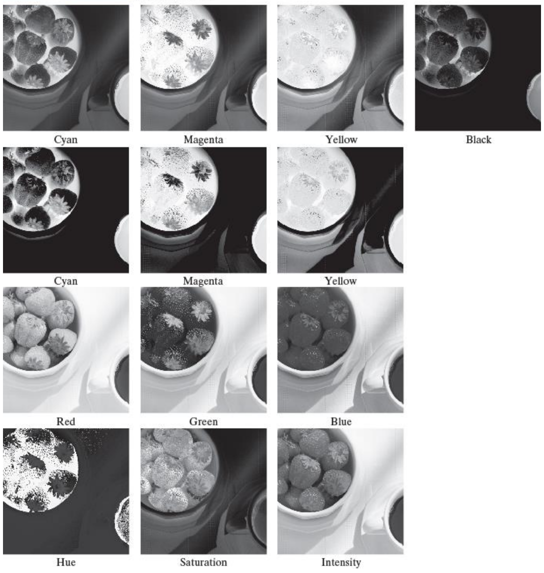
a b
FIGURE 6.27 Spatial neighborhoods for grayscale and RGB color images. Observe in (b) that a *single* pair of spatial coordinates, (x, y), addresses the same spatial location in all three images.

Color Transformations

- Gray-level transformation: $g(x,y) = T[f(x,y)]$
- Color transformation: $s_i = T_i(r_1, r_2, \cdots, r_n)$ $I = 1,2,\cdots,n$

where r_i and s_i denote the **color component** of $f(x,y)$ and $g(x,y)$ at any point (x,y), i.e., R/G/B. n is the number of color components, ($n = 3$). $\{T_i\}$ is a set of transformation or color mapping functions.

Color Transformations



Color Transformations

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

Color Transformations

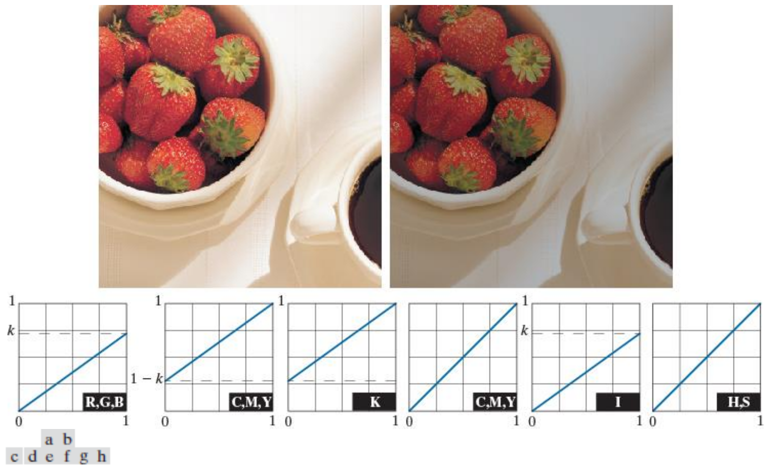


FIGURE 6.29 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c) The required RGB mapping function. (d)–(e) The required CMYK mapping functions. (f) The required CMY mapping function. (g)–(h) The required HSI mapping functions. (Original image courtesy of MedData Interactive.)

Color Complements

- The hues directly opposite one another on the color circle are called **complements** (互補色)
- Color complements are useful for enhancing detail that is embedded in **dark regions** of a color image.
- The **saturation** component of the **complement** of a color image cannot be computed from the saturation of the input image alone.
- In Fig. 6.30, the **saturation** component of the input image is unaltered.

Color Complements

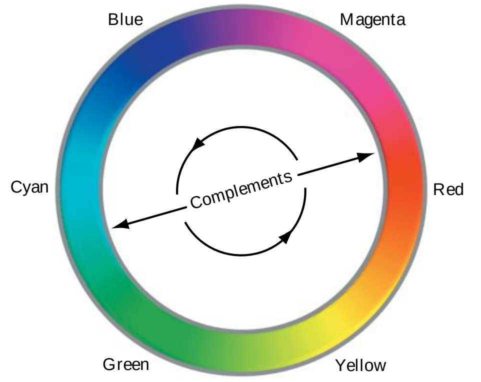
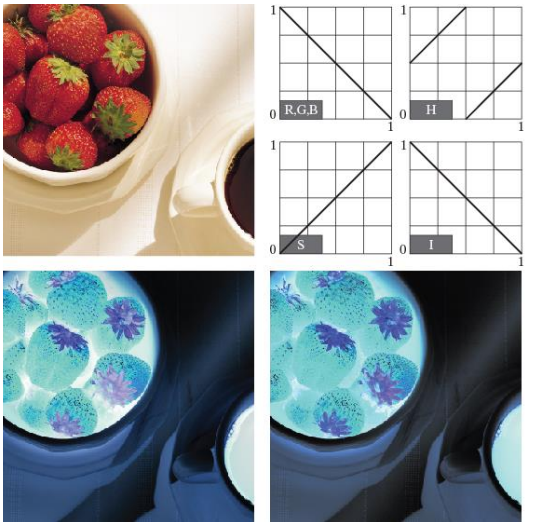


FIGURE 6.30 Color complements on the color circle.

Color Complements



a b
c d

FIGURE 6.31 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 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 non-prominent **neutral color**, $(R, G, B) = (0.5, 0.5, 0.5)$.
- 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

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

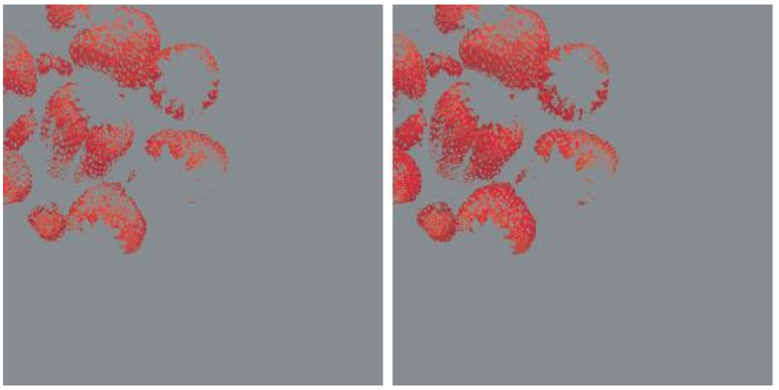
Color Slicing

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

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

- 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)

Color Slicing



a b

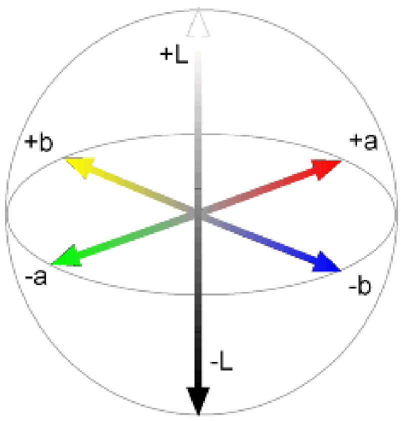
FIGURE 6.32 Color-slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at $(0.6863, 0.1608, 0.1922)$, and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color $(0.5, 0.5, 0.5)$.

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.

L*a*b* Color Space

"Perceptually uniform" color space



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

Tone and Color Correction

- The model of 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

$$L^* = 116 \cdot h(Y/Y_W) - 16$$
$$a^* = 500[h(X/X_W) - h(Y/Y_W)]$$
$$b^* = 200[h(Y/Y_W) - h(Z/Z_W)]$$

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

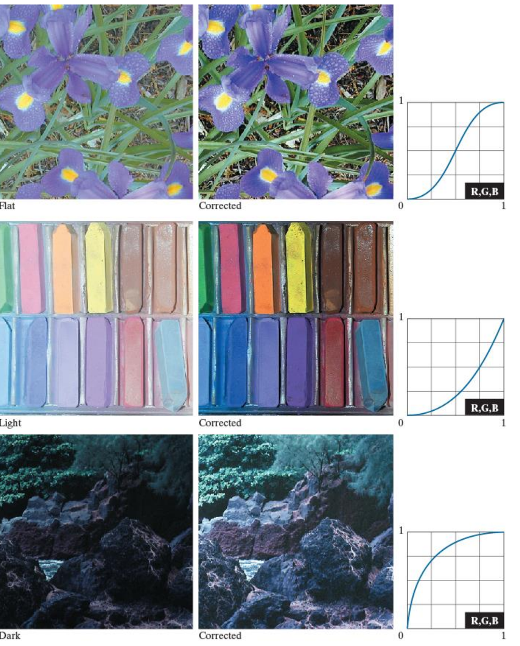
Tone and Color Correction

- 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^*)

Tone and Color Correction

- **Tonal** and **color** imbalances are corrected interactively.
- **Tonal correction** and then **color correction**.
- The **tonal range** of an image (*key type*) refers to its general distribution of color intensity.
- **High-key image**: the color is concentrated at high (or light) intensity.
- **Low-key image**: the color is concentrated at low intensity.
- **Middle-key image** lies in between high-key and low-key images.
- It is desirable to distribute the intensities of a color image equally between the **highlights** and the **shadows**

Tone and Color Correction



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

FIGURE 6.33 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

Tone and Color Correction

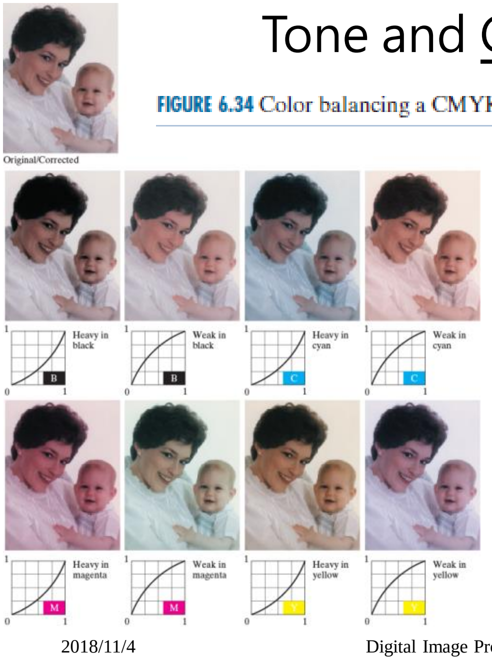
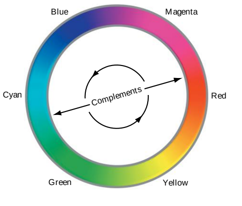


FIGURE 6.34 Color balancing a CMYK image.



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.

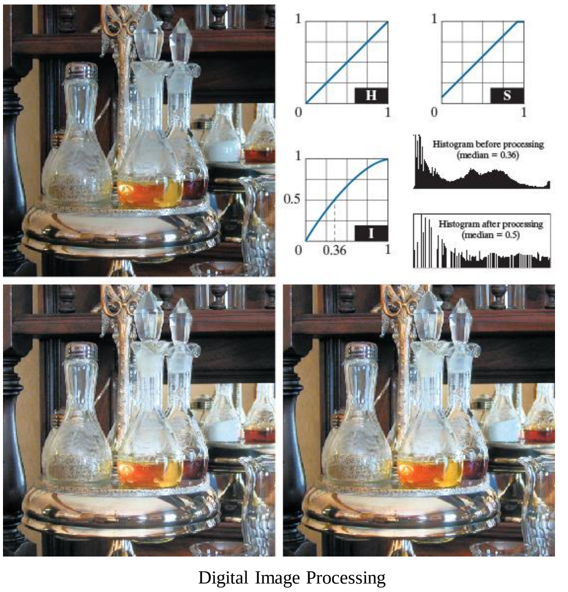
Histogram Processing

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

Histogram Processing

Mean=0.36

Mean=0.5



Saturation adjustment

a b
c d

FIGURE 6.35 Histogram equalization (followed by saturation adjustment) in the HSI color space.

Smoothing and Sharpening

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

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

Smoothing and Sharpening

a b
c d

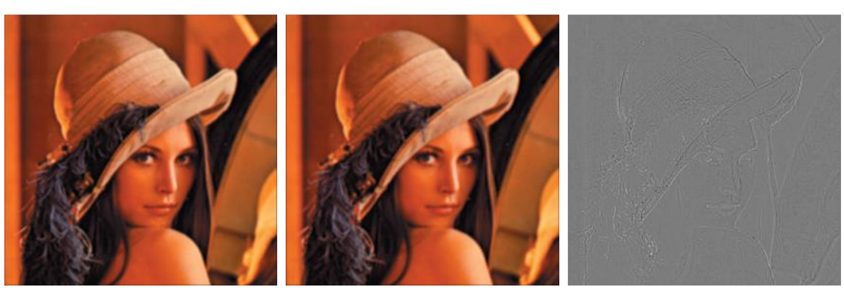
FIGURE 6.36
(a) RGB image.
(b) Red component image.
(c)Green component.
(d) Blue component.

Smoothing and Sharpening

a b c

FIGURE 6.37 HSI components of the RGB color image in Fig. 6.36(a). (a) Hue. (b) Saturation. (c) Intensity.

Smoothing and Sharpening



a b c

FIGURE 6.38 Image smoothing with a 5×5 averaging kernel. (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.

Smoothing and Sharpening

- Image sharpening using Laplacian

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

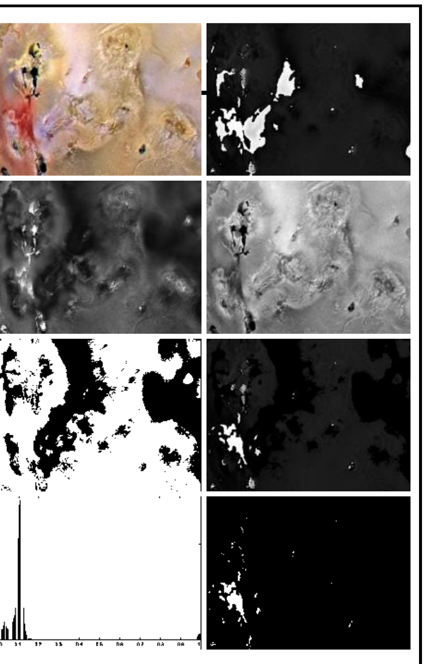
a b c

FIGURE 6.39 Image sharpening using the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the HSI intensity component and converting to RGB. (c) Difference between the two results.

Color Segmentation

- Partition an image into **regions**.
- Segmentation in HSI 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.

Color Segmentation



a b c d e f g h **FIGURE 6.40** 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 from (a).

Color Segmentation in RGB

- Segmentation in RGB color space
- The measurement of color similarity is the **Euclidean distance** between two colors $\mathbf{z}$, and $\mathbf{a}$, (i.e., Fig. 6.41(a)),

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

$$= \left[(z_R-a_R)^2+(z_G-a_G)^2+(z_B-a_B)^2\right]^{1/2}$$

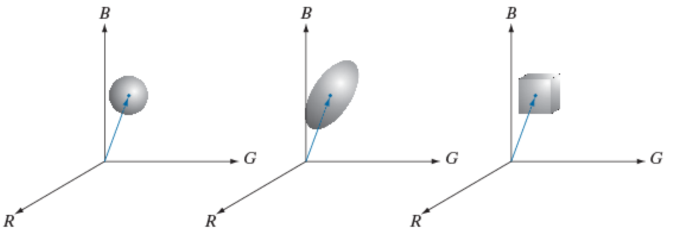
- A generalization of distance measure is

$$D(\mathbf{z},\boldsymbol{a}) = \|\mathbf{z}-\mathbf{a}\| = \left[(\mathbf{z}-\mathbf{a})^T\mathbf{C}^{-1}(\mathbf{z}-\mathbf{a})\right]^{1/2}{}_2$$

- Where $\mathbf{C}$ is the covariance matrix of the **samples representative** of the color we want to segment.
- Fig. 6.41(b) describes the **solid elliptical body** with the **principal axes** oriented in the direction of maximum data spread.

Color Segmentation in RGB

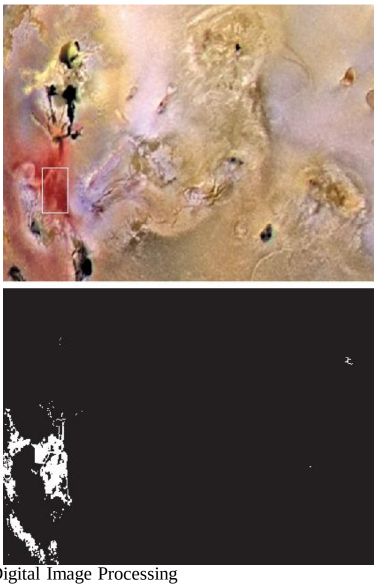
a b c
FIGURE 6.41 Three approaches for enclosing data regions for RGB vector segmentation.



Distance square without square root operation.

Color Segmentation in RGB

a
b
FIGURE 6.42 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. 6.40(h).

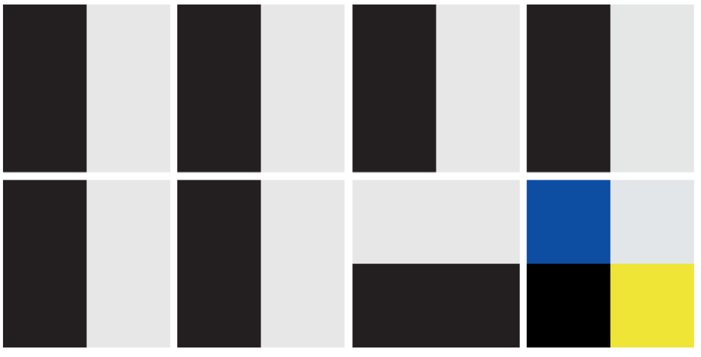


The dimension of the box along R-axis extended from (a_R–1.25σ_R) to (a_R+1.25σ_R)

Color Edge Detection

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

Color Edge Detection



a b c d
e f g h

FIGURE 6.43 (a)–(c) R, G, and B component images, and (d) resulting RGB color image. (e)–(g) R, G, and B component images, and (h) resulting RGB color image.

Color Edge Detection

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

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

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

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

Color Edge Detection

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

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

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

$$F_\theta(x,y) = \left\{0.5\left(g_{xx} + g_{yy}\right) + \left(g_{xx} - g_{yy}\right)\cos 2\theta + 2g_{xy}\sin 2\theta\right\}^{1/2}$$

- There are two solution θ_0 or $\theta_0 + \pi/2$ in orthogonal directions.

$$F_\theta(x,y) = F_{\theta+\pi}(x,y)$$

- One generates maximum F and the other generates minimum F.

Color Edge Detection

a b
c d

FIGURE 6.44
(a) RGB image.
(b) Gradient computed in RGB color vector space.
(c) Gradient image formed by the elementwise sum of three individual gradient images, each computed using the Sobel operators.
(d) Difference between (b) and (c).

Color Edge Detection

a b c

FIGURE 6.45 Component gradient images of the color image in Fig. 6.44. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 6.44(c).

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 Fig. 6.46) tends to be less visually noticeable in a color image than it is in a monochrome image.

Noise in Color Image

a b
c d

FIGURE 6.46 (a)–(c) Red, green, and blue 8-bit component images corrupted by additive Gaussian noise of mean 0 and standard deviation of 28 intensity levels. (d) Resulting RGB image. [Compare (d) with Fig. 6.44(a).]

Noise in Color Image

a b c

FIGURE 6.47 HSI components of the noisy color image in Fig. 6.46(d). (a) Hue. (b) Saturation. (c) Intensity.

Noise in Color Image

a b
c d

FIGURE 6.48
(a) RGB image with green plane corrupted by salt-and-pepper noise.
(b) Hue component of HSI image.
(c) Saturation component.
(d) Intensity component.

Color Image Compression

a
b

FIGURE 6.49 Color image compression. (a) Original RGB image. (b) Result of compressing, then decompressing the image in (a).

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