Chapter 6: Color Image Processing

Chapter Overview:

- ✓ Introduce the basic concept of color and color models.
- ✓ Introduce the pseudo-color image processing concept and its applications.
- ✓ Introduce some basic full-color image image processing and its applications.

Introduction to Color Image Processing

Full-color

Images are acquired with full-color sensor, e.g. CCD camera, color scanner, etc.

Pseudo-color

- Assigning a color to a particular monochrome intensities or range of intensities.
- Most color image processing done in this mode.

Area of applications – publishing, visualization, internet, etc.

Color Fundamentals

- When a beam of sunlight passes through a glass prima, the emerging beam of light is not white but consists of a continuous spectrum of colors.
- The color spectrum may be divided into 6 broad regions: violet, blue, green, yellow, orange, and red.

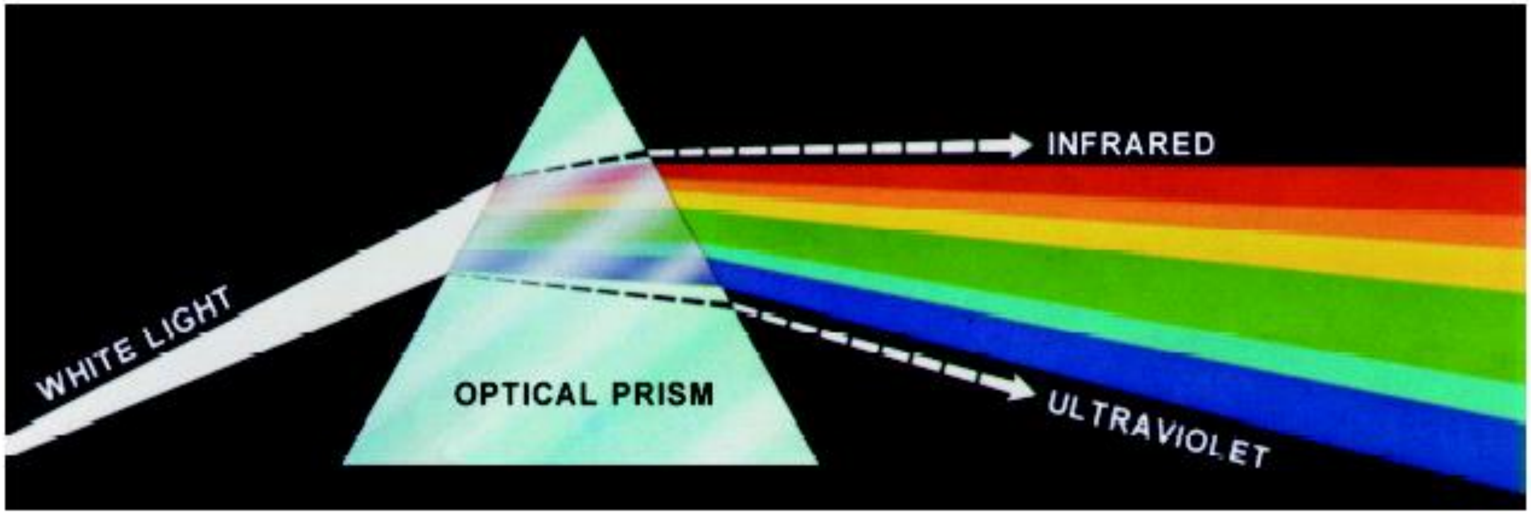


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

Color Fundamentals (cont')

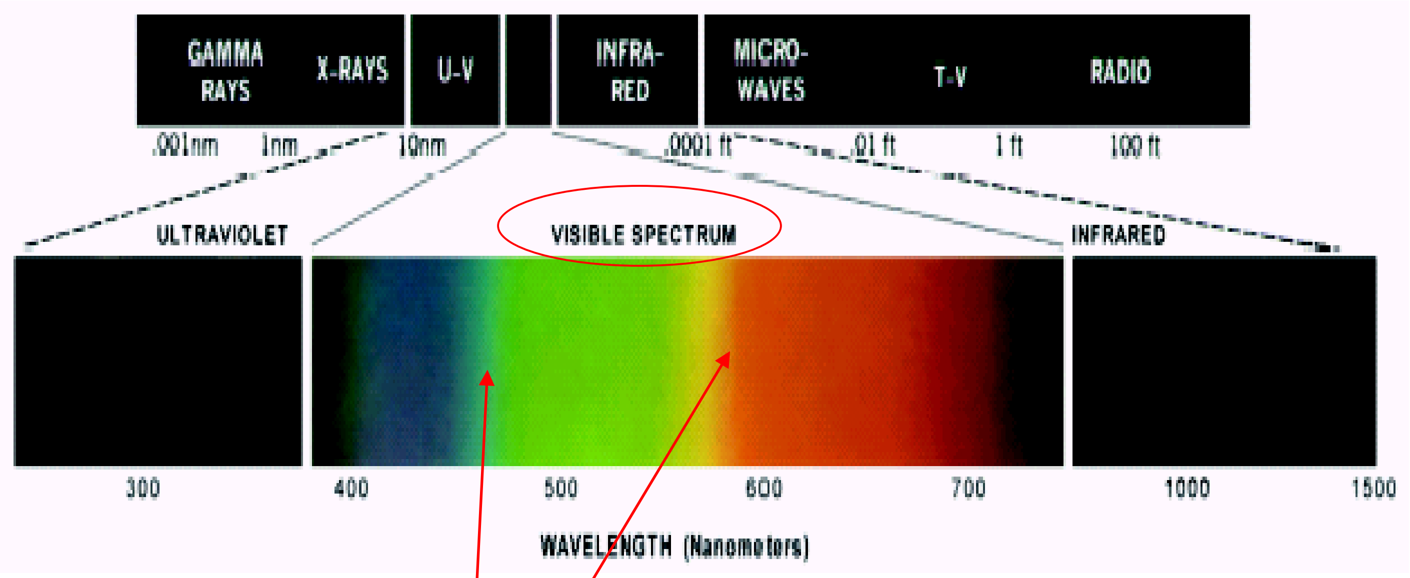


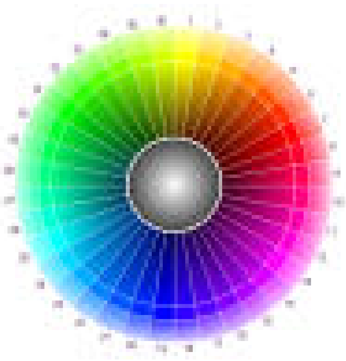
FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Smooth transition of colors

Color Fundamentals (cont')

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

- A body that reflects light that is balanced in all visible wavelengths appears white to the observer.



Color Fundamentals (cont')

3 basic quantities are used to describe the quality of a chromatic light source:

QUANTITY	DESCRIPTION
Radiance	▪ Total amount of energy that flows from the light source. ▪ Usually measured in watts (W).
Luminance	▪ A measure of the amount of energy an observer *perceives* from a light source. ▪ Measured in lumens (lm).
Brightness	▪ Subjective descriptor that is practically impossible to measure. ▪ It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.

Color Fundamentals (cont')

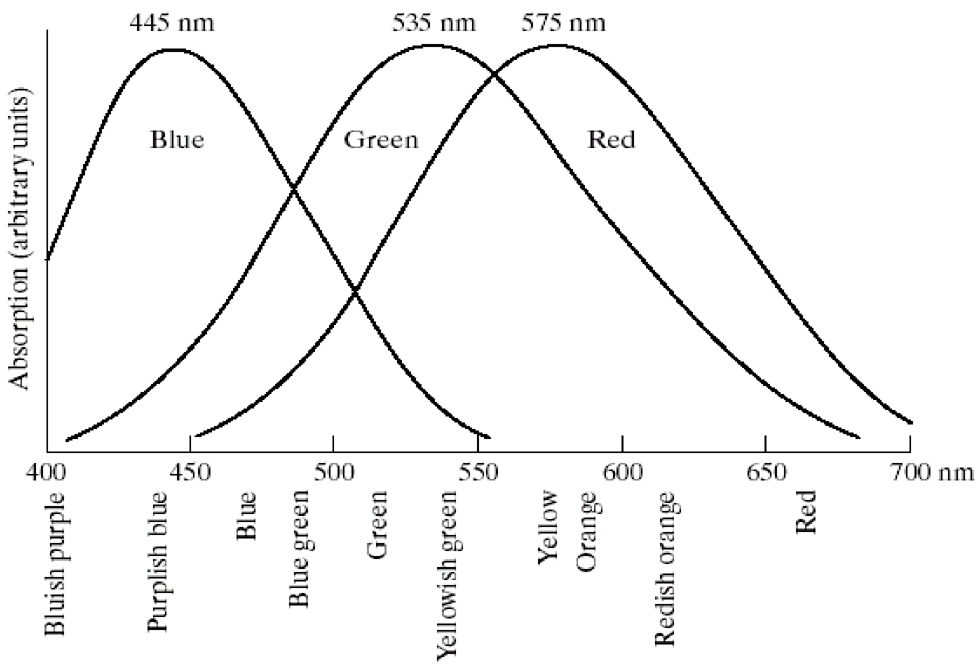


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

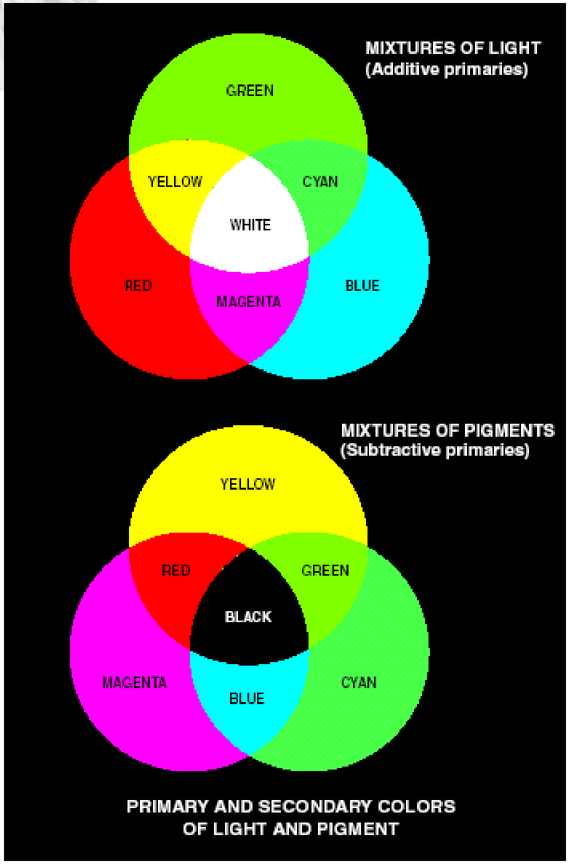
Absorption of light by human eye.

Color Fundamentals (cont')

- Due to these absorption characteristics of the human eye, color are seen as variable combinations of *primary colors* : Red (R), Green (G), and Blue (B).
- For the purpose of standardization, the Commission International de I'Eclairage – the International Commission on Ilumination (CIE) designated in 1931 the following specific wavelength values to the primary colors of light.

PRIMARY COLORS	WAVELENGTH VALUES
RED	700 nm
GREEN	546.1 nm
BLUE	435.8 nm

Color Fundamentals (cont')



a
b

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

- Primary colors can be added to produce the *secondary* colors of light :
  - magenta (red plus blue)
  - cyan (green plus blue)
  - yellow (red plus green).
- Mixing the 3 primaries, or a secondary with its opposite primary color, in the right intensities produces white light.
- A proper combination of the three pigment primaries, or a secondary with its opposite primary, produces black.

Color Fundamentals (cont')

- It is important to differentiate between <u>the primary colors of light</u> and the <u>primary colors of pigments</u> (colorants).
- Primary colors of light: one that subtracts or absorbs a primary color of light and reflects or transmits the other two.

TYPE / COLORS	COLORS OF LIGHT	COLORS OF PIGMENTS
PRIMARY	RED	MAGENTA
	GREEN	CYAN
	BLUE	YELLOW
SECONDARY	MAGENTA	RED
	CYAN	GREEN
	YELLOW	BLUE

Color Fundamentals (cont')

Characteristic used to distinguish a color from another are :

CHARACTERISTIC	DESCRIPTION
BRIGHTNESS / (INTENSITY)	▪ Embodies the chromatic notion of intensity.
HUE	▪ An attribute associated with the dominant wavelength in a mixture of light waves. ▪ It represents dominant color as perceive by an observer. - Thus, when we call an object red, orange, or yellow, we are specifying its hue.
SATURATION	▪ The relative purity or the amount of white light mixed with a hue. ▪ The pure spectrum colors are fully saturated. ▪ Colors such as pink (red and white) and lavender (violet and white) are less saturated, with the degree saturation being inversely proportional to the amount of light white added.

- Hue and saturation taken together are called *chromaticity*, and therefore a color may be characterized by its brightness and chromaticity.

Color Fundamentals (cont')

- *Tristimulus* values : The amounts of red, green and blue needed to form any particular color.
- A color is the specified by its *trichromatic coefficient*

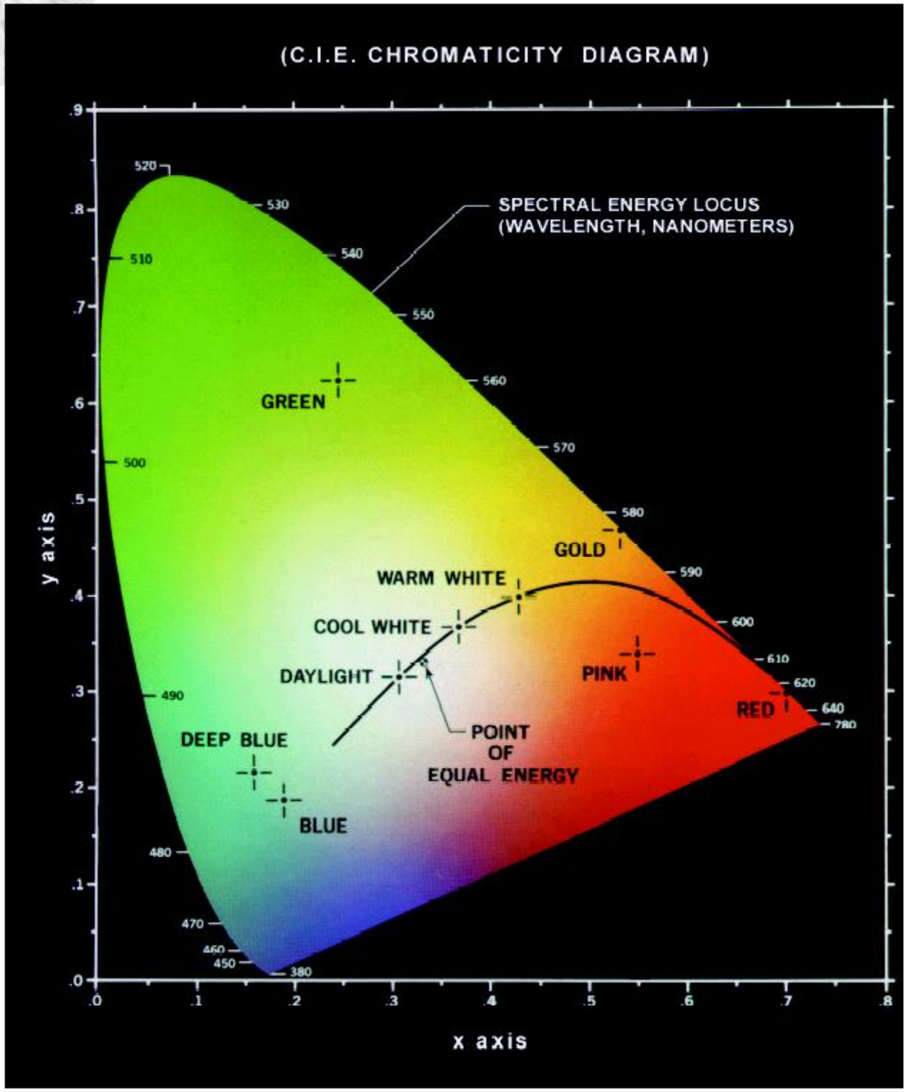
TRISTIMULUS VALUES	TRICHROMATIC COEFFICIENT
X	$$x = \frac{X}{X+Y+Z} \quad (6.1\text{-}1)$$
Y	$$y = \frac{Y}{X+Y+Z} \quad (6.1\text{-}2)$$
Z	$$z = \frac{Z}{X+Y+Z} \quad (6.1\text{-}3)$$

- It is noted from these equations that

$$x + y + z = 1 \quad (6.1\text{-}4)$$

Color Fundamentals (cont')

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



CIE Approach for specifying color

CIE *chromaticity diagram* - shows color composition as a function of *x* (red) and *y* (green).

For any value of *x* and *y*, the corresponding value of *z* (blue) is obtained from eq. (6.1-4) by noting that $z = 1 - (x + y)$.

Color Fundamentals (cont')

By drawing a line from 3 color points – any color inside the triangle can be produced by various combination of the 3 initial colors.

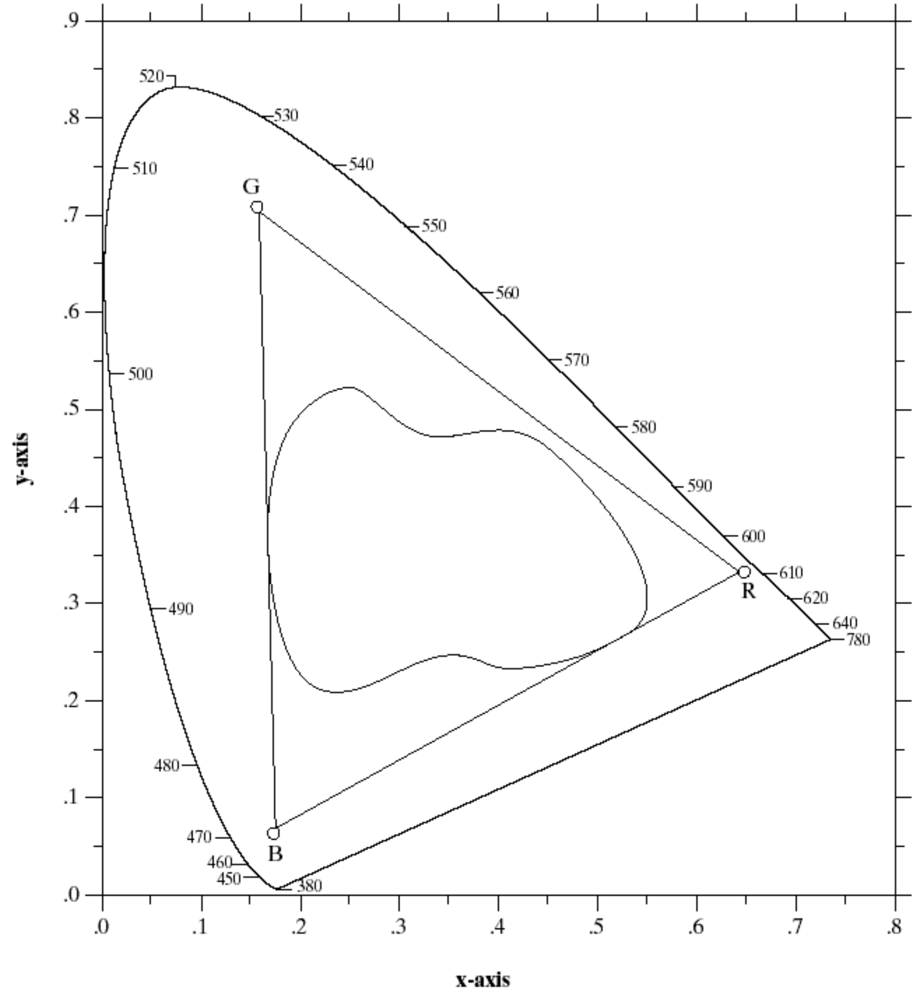


FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

Color Models

- Also called as ***color space/color system***
- Definition - a specification of a coordinate system and a subspace within that system where each color is represented by a single point.
- The purpose: to facilitate the specification of colors in some standard.
- Mostly oriented either toward : - Hardware (e.g for color monitors and printers).

  - Applications (e.g color graphics for animation).
- The most common used hardware oriented models are:

MODELS	DESCRIPTION	USAGE
RGB	RED, GREEN, BLUE	Color monitors, color video cameras
CMY	CYAN, MAGENTA, YELLOW	Color printing
CMYK	CYAN, MAGENTA, YELLOW, BLACK	
HSI	HUE, SATURATION, INTENSITY	Describe and interpret color, decouples the color and gray-scale information in an image

The RGB Color Model

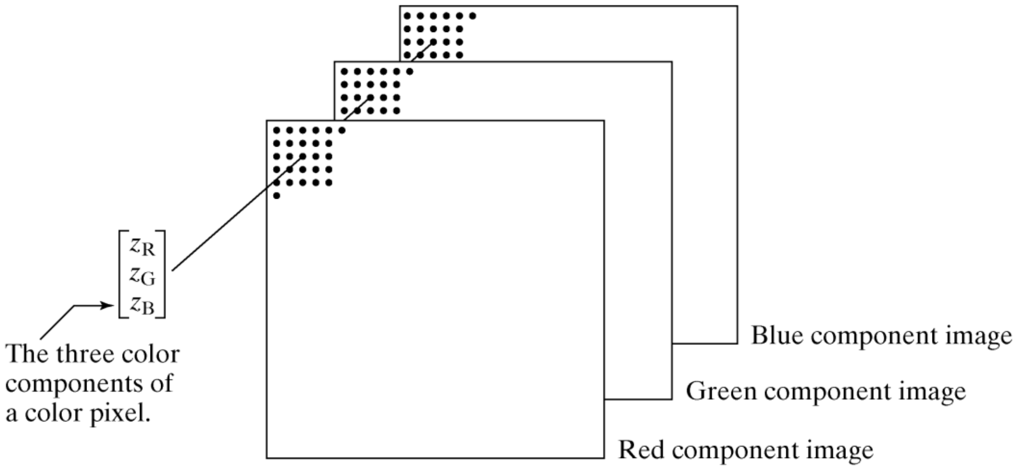
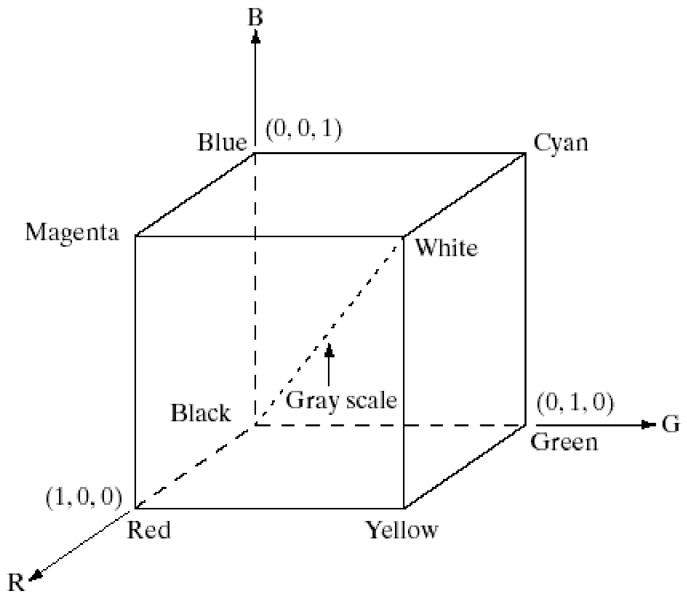


FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

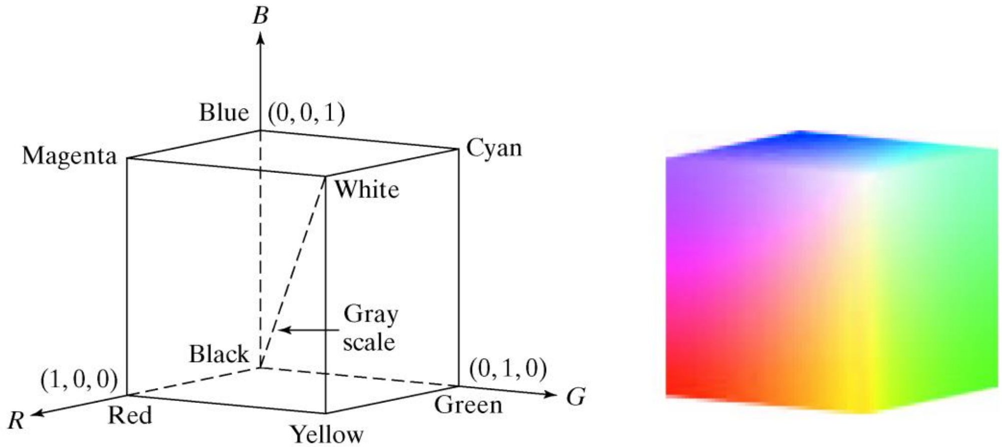
- Based on a Cartesian coordinate system.
- Each color appears in its primary spectral components (red, green, blue)



The RGB Color Model (cont')

Consider an RGB image in which each images is an 8-bit image.

- *Pixel depth* : the number of bits used to represent each pixel in RGB space. Thus, each RGB *color pixel* is said to have a depth of 24-bits. (image planes x no. of bits per plane)
- *Full color* image : often used to denote a 24-bits per plane
- The number of colors in a 24-bit RGB image shown below is $(2^8)^3 = 16,777,216$ colors.



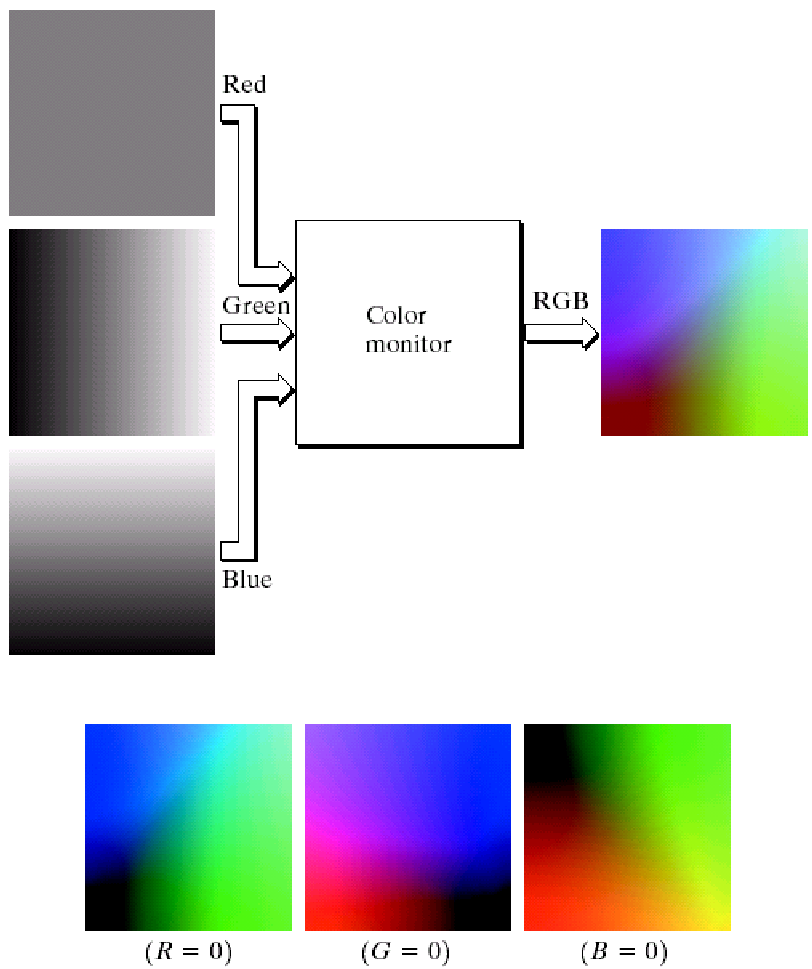
a b

FIGURE 6.2 (a) Schematic of the RGB color cube showing the primary and secondary colors of light at the vertices. Points along the main diagonal have gray values from black at the origin to white at point (1, 1, 1). (b) The RGB color cube.

The RGB Color Model (cont')

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. 6.2



- From Fig. 6.9(a), an image of the cross-sectional plane is viewed simply by feeding the 3 individual component images into a color monitor.
- In the component images, 0 represent black and 255 represent white.
- Fig. 6.9 (b) shows the 3 hidden surface planes of the cube.
- A color image can be acquired by using 3 filters, sensitive to red, green, and blue, respectively which is basically the reverse process in Fig. 6.9

The CMY and CMYK Color Models

- Most devices that deposit colored pigments on paper (e.g color printers, copier) require CMY data input or perform an RGB to CMY conversion internally using the simple operation of:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (6.2\text{-}1)$$

  with assumption that all color values have been normalized to the range [0,1].

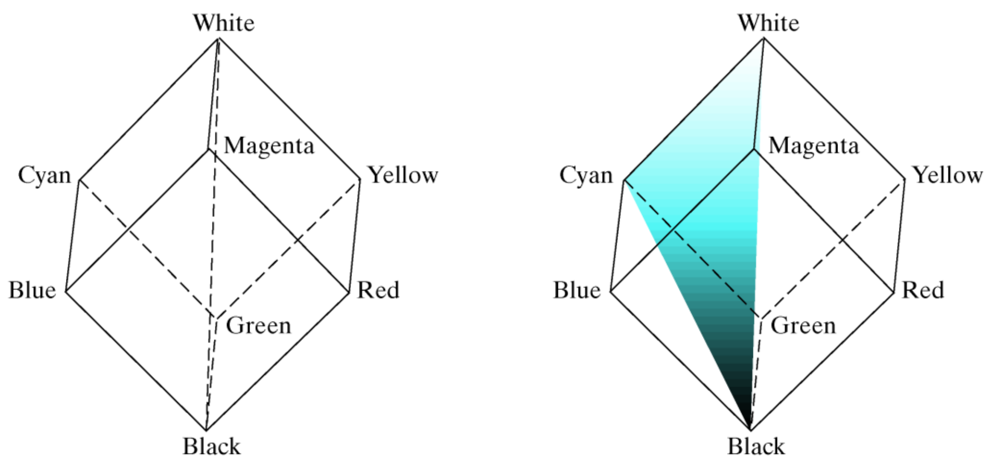
- Eq. 6.2-1 demonstrates that light reflected from a surface coated with pure cyan does not contained red (that is C = 1 - R in the equation)
- Similarly, pure magenta does not reflect green, and pure yellow does not reflect blue.

The CMY and CMYK Color Models (cont')

- Eq. 6.2-1 reveals that RGB values can be obtained easily from a set of CMY values by subtracting the individual CMY values from 1.
- In practice, combining equal amount of the pigment primaries; cyan, magenta and yellow should produce a muddy-looking black.
- In order to produce true black, a fourth color, *black* is added to the CMYK color model.
- Most publishers refer the 3 color of the CMY color model plus black as "four-color printing"

The HSI Color Models

- RGB model is ideal for image color generation (e.g image capture/image display), but it is not well suited for *describing* colors in terms that are practical for human interpretation.
- The HSI color model, decouples the intensity component from the color-carrying information (hue and saturation) in a color image, making it as an ideal tool for developing image processing algorithms based on color descriptions that are natural and intuitive to humans.
- To determine the intensity component of any color point in Fig. 6.12, pass a plane perpendicular to the intensity axis and containing the color point.



a b

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

The HSI Color Models (cont’)

The shape does not matter

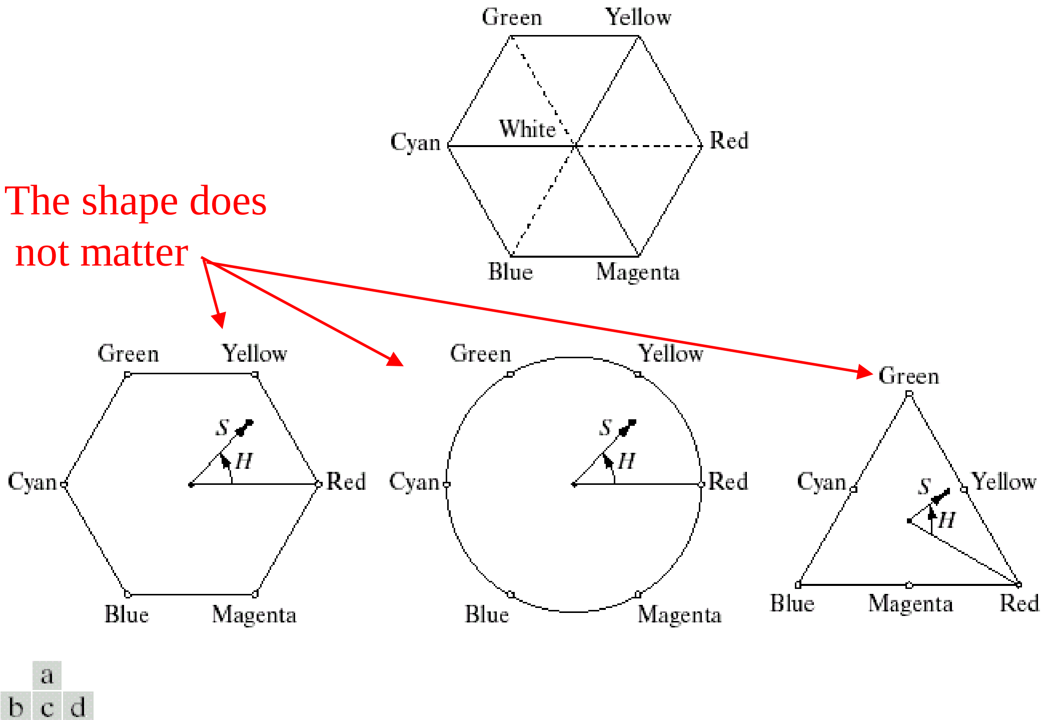


FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

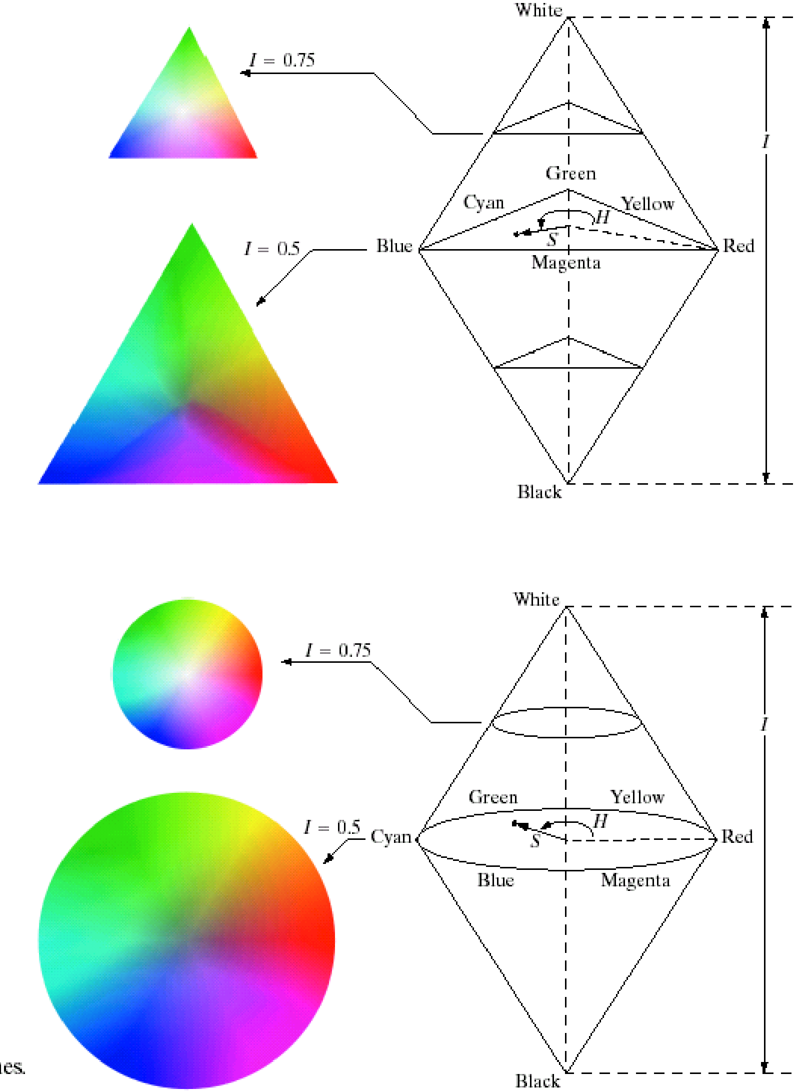
- Fig. 6.13(a) shows the cube down its gray-scale axis in which :
  - Each primary color are separated by 120°
  - secondary colors are 120° from each other/60° from the primaries
- Fig 6.13(b) shows the same hexagonal shape and an arbitrary color point.
- The *hue* of the point is determined by an angle from some reference point
- The *saturation* is the length of the vector from the origin to the point

The HSI Color Models (cont’)

- Origin : intersection of the color plane to the vertical intensity axis.
- important components of HSI color space are:

  - The vertical intensity axis.
  - The length of the vector to the color point
  - The angle this vector makes with the red axis

- The shape chosen as shown in Fig. 6.13(c) and (d) are not important, since any one of these shapes can be warped into one of the other two by a geometric transformation.
- Fig. 6.14 shows the HSI model based on color triangles and also on circles.



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.

Converting RGB to HSI

- For an image in RGB color format, the H component of each RGB pixel is obtained using equation :

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \qquad (6.2\text{-}2)$$

with:

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

- The saturation component is given by :

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \qquad (6.2\text{-}3)$$

Converting RGB to HSI (cont')

- The intensity component is given by :

$$I = \frac{1}{3}(R+G+B) \qquad (6.2\text{-}4)$$

- With assumption :

  - the RGB values have been normalized to the range [0,1]

  - the angle θ is measured with respect to the red axis of the HSI space

- Hue can be normalized to the range [0,1] by dividing by 360° all values resulting from Eq. (6.2.2)

- The other 2 HSI components already are in this range if the given RGB values are in the interval [0,1]

Converting HSI to RGB (cont')

- For values of HSI in the interval [0,1], the corresponding RGB values for the same range can be obtained by multiplying *H* by 360°, which returns the hue to its original range of [0°, 360°]:

RG sector	GB sector	BR sector
$(0° \le H < 120°)$	$(120° \le H < 240°)$	$(240° \le H < 360°)$
	$H = H - 120°$ (6.2-8)	$H = H - 240°$ (6.2-12)
$B = I(1 - S)$ (6.2-5)	$R = I(1 - S)$ (6.2-9)	$G = I(1 - S)$ (6.2-13)
$R = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$ (6.2-6)	$G = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$ (6.2-10)	$B = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$ (6.2-14)
$G = 1 - (R + B)$ (6.2-7)	$B = 1 - (R + G)$ (6.2-11)	$R = 1 - (G + B)$ (6.2-15)

Manipulating HSI Component Images

(a)

An image composed of the primary and secondary RGB colors.

(b)

The gray-level values correspond to angles (as red correspond 0°, the red region in (a) mapped to a black region in the hue image)

(c)

Gray level correspond to saturation (they were scaled to [0,255] for display)

(d)

Average intensities.

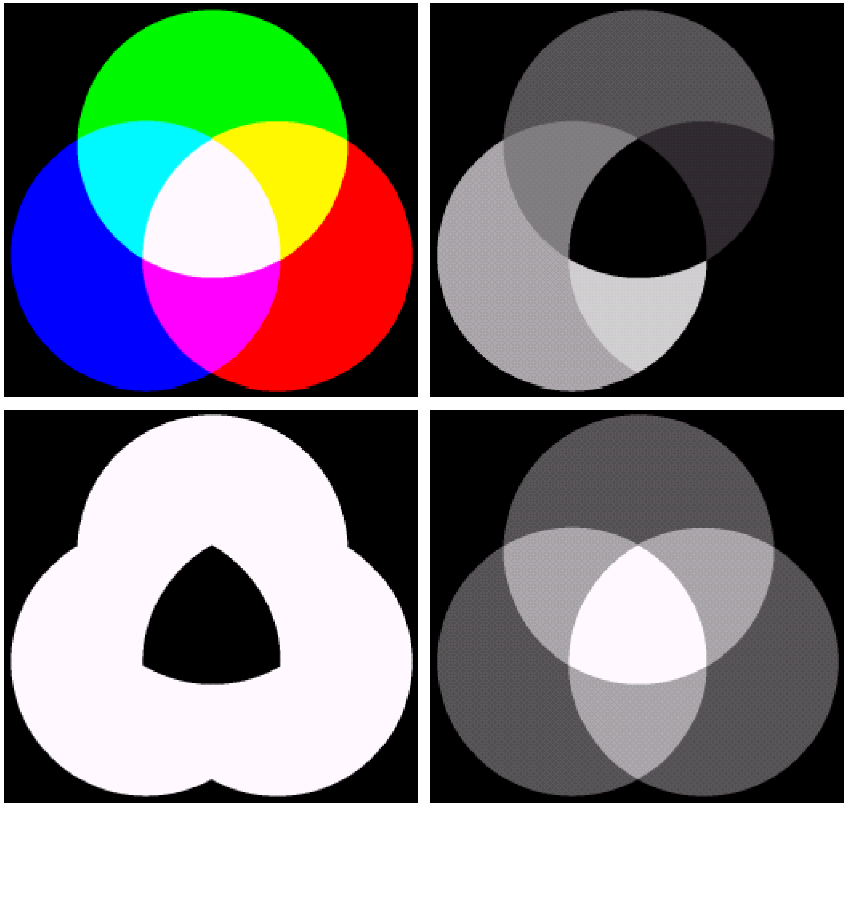


FIGURE 6.16 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

Manipulating HSI Component Images (cont')

(a)

obtained by changing to 0 the pixels corresponding to blue and green regions in (b)

(b)

Reduced by half the saturation of the cyan region in component image S from (c)

(c)

Reduced by half the intensity of the central white region in the intensity image (d)

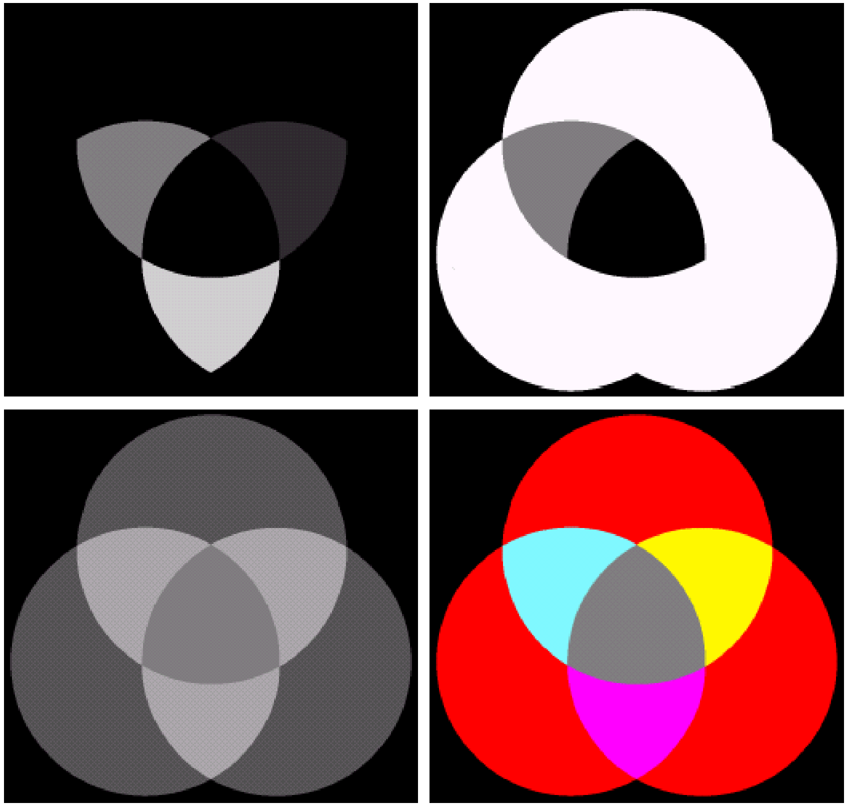
(d)

The result of converting this modified HIS image back to RGB

The outer portions of all circles = red

The purity of the cyan region was diminished

The central region became gray rather than white



a b
c d

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

Pseudocolor Image Processing

- Pseudocolor (also called false color) image processing consists of assigning colors to gray values based on a specified criterion.
- The term *pseudo/false* color is used to differentiate the process of assigning colors to monochrome images from the process associated with true color images (will be discussed in later section).
- The principal use of pseudocolor is for human visualization and interpretation of gray-scale events in an image or sequence of images – human can discern thousand of color shades and intensities, compared to 2 dozen or so shades of gray.

Intensity Slicing

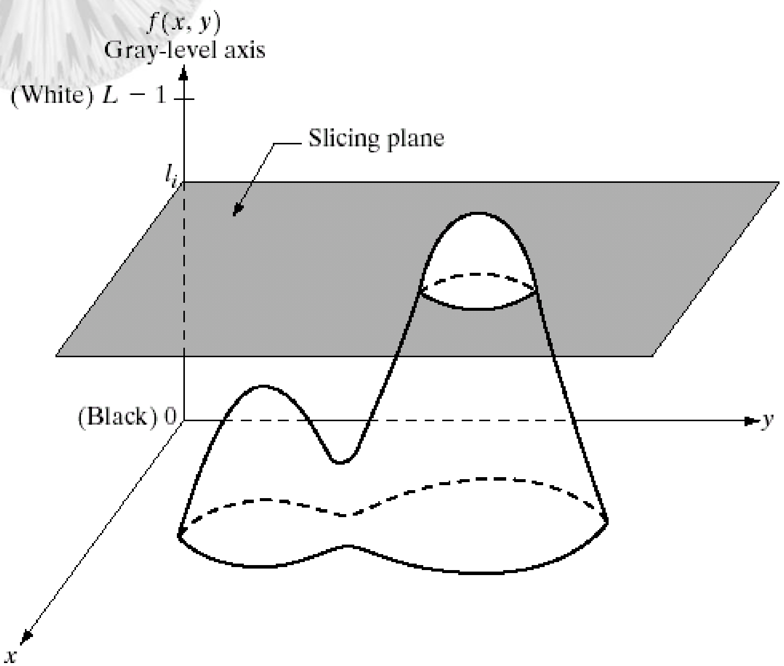


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

Fig 6.18

shows an example of using a plane at f(x,y) = l_i to slice the image function into 2 levels.

- Consider an image to be interpreted as a 3-D function (intensity-spatial coordinates).
- If different color is assigned to each side of the plane, any pixel whose gray level is:

  - above the plane: coded with one color.
  - below the plane: coded with the other.

- Levels that lie on the plane itself may be arbitrarily assigned one of the 2 colors.
- The result = a two-color image whose relative appearance can be controlled by moving the slicing plane up and down the gray-level axis.

Intensity Slicing (cont')

- Generally, the technique can be summarize as follows:

  - let $[0, L-1]$ represent the gray scale
  - let l_0 represent black $[f(x,y) = 0]$
  - level l_{L-1} represent white $[f(x,y) = L-1]$

- Suppose that P planes perpendicular to the intensity axis are defined at levels $l_1, l_2, \ldots, l_p$.
- Assumption: $0 < P < L-1$, the P planes partition the gray-scale into $P + 1$ intervals, $V_1, V_2, \ldots, V_{P+1}$.
- Gray-level to color assignments are made according to the relation:

$$f(x,y) = c_k \quad \text{if} \quad f(x,y) \in V_k \tag{6.2-2}$$

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

Intensity Slicing (cont')

- The idea of planes is useful primarily for a geometric interpretation of the intensity-slicing technique.

Fig 6.19

- An alternative representation that defines the same mapping as in Fig 6.18
- Any input gray level assigned one of 2 colors, depending on whether it is above or below the value of l_i
- When more level is used, the mapping function takes on a staircase form

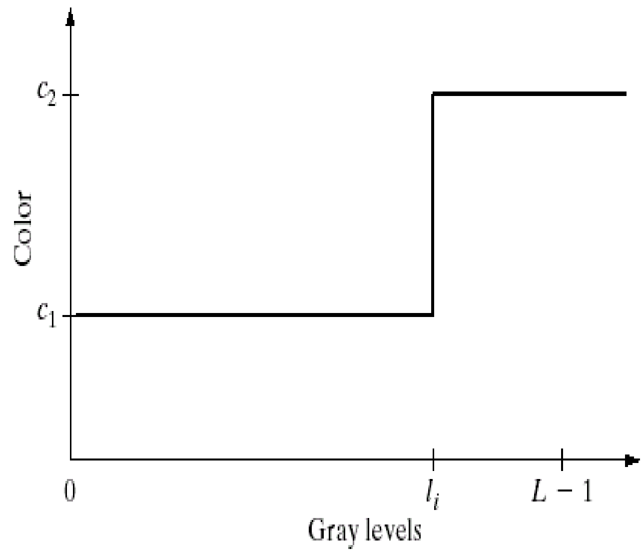
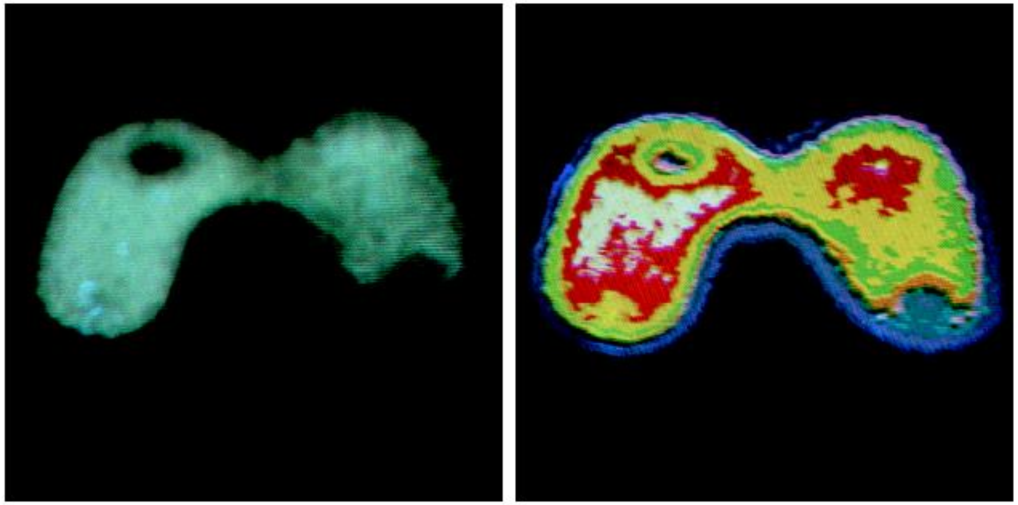


FIGURE 6.19 An alternative representation of the intensity-slicing technique.

Application of Intensity Slicing – Radiation Test Pattern



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.)

Fig. 6.20

Shows a simple, but practical use of intensity slicing

(a)

Picker Thyroid Phantom (a radiation test pattern)

(b)

Result: 8 color regions

The regions that appear of constant intensity in the monochrome image are quite variable (various colors in the sliced image)

By contrast, the color image shows 8 different regions of constant intensity, one for each of the colors used.

Application of Intensity Slicing - Detection of a Crack in Weld

a
b

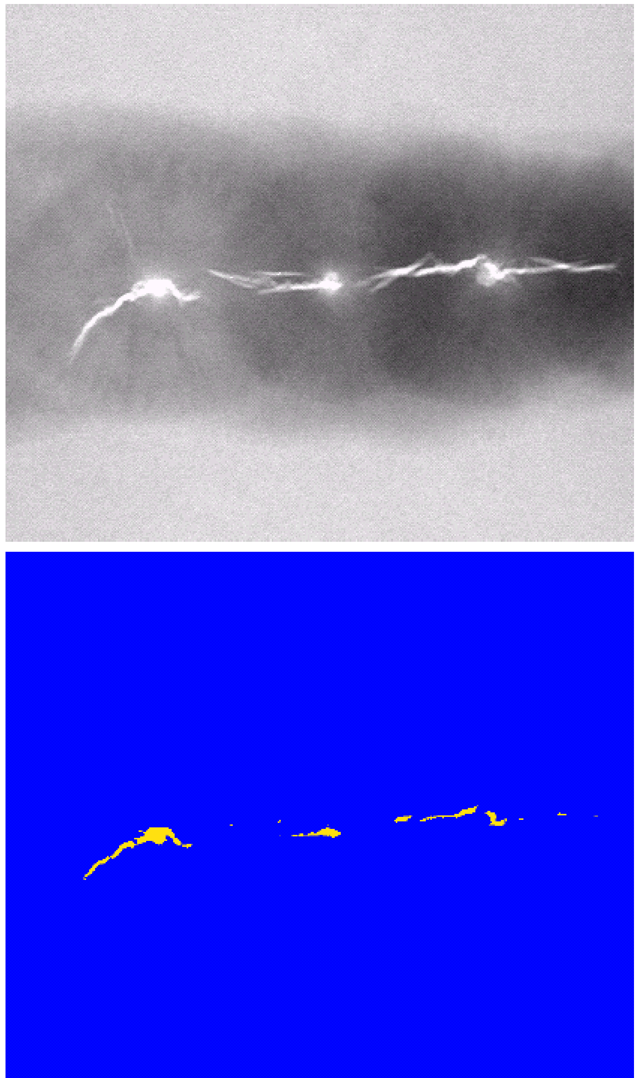
FIGURE 6.21 (a) Monochrome X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)

Fig 6.21

Intensity slicing assumes a much more meaningful and useful role when subdivision of the gray scale is based on physical characteristics of the image.

The gray scale was divided into intervals.

A different color was assigned to each region.



(a)

Shows an X-ray image of a weld (the horizontal dark region) containing several cracks and porosities (the bright, white streaks running horizontally through the middle of the image)

When there is a porosity/crack in a weld, the full strength of the X-rays going through the object saturates the imaging sensor on the other side of the object.

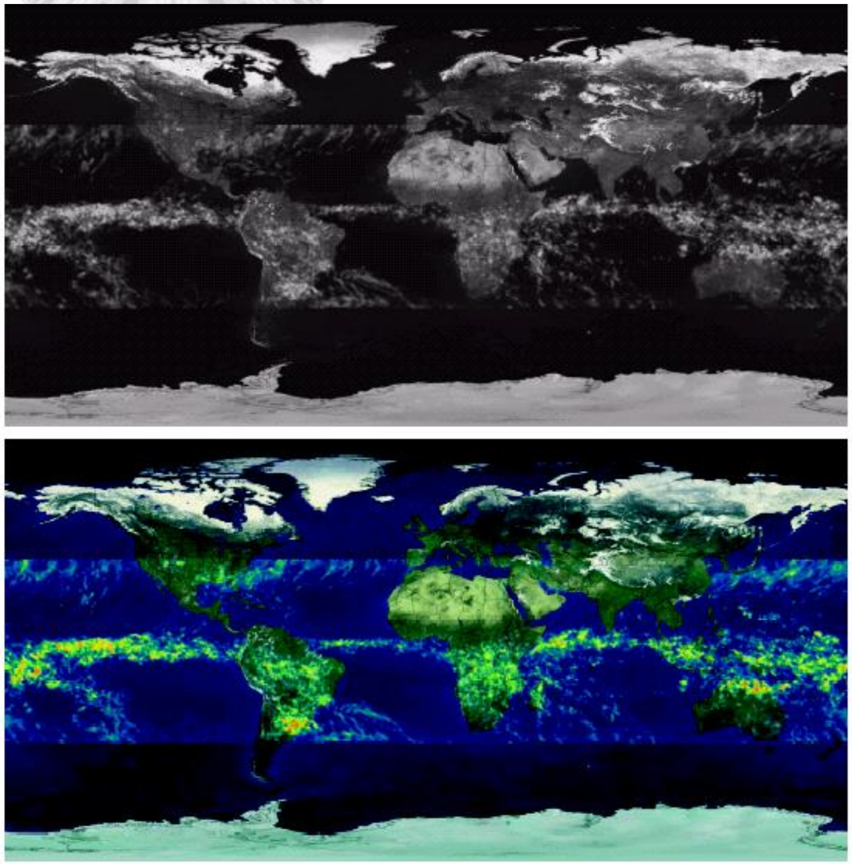
Thus, the gray levels of value 255 in an 8-bit image coming from such a system automatically imply a problem with the weld.

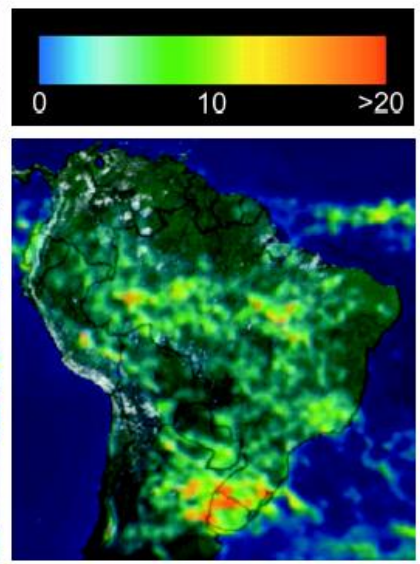
(b)

Human error rates would be lower

If the exact values of gray levels are known, intensity slicing is a simple but powerful aid in visualization, especially if numerous images are involved.

Application of Intensity Slicing - Highlighting Rainfall Levels





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.)

(a)

An intensity image, where the area monitored by the satellite is the slightly lighter horizontal band in the middle one-third of the picture.

The rainfall values are average monthly values over a 3 year period.

(b)

The colors shown are used to code gray levels from 0 to 255.

Values toward blue signify low values of rainfall, with the opposite being true for red.

(c)

The result of color coding the gray image with the color map.

(d)

Zoomed area

Gray Level to Color Transformation

- To perform 3 independent transformations on the gray level of any input pixel.
- The 3 result are fed separately into the red, green, and blue channels of a color television monitor.
- It produces a composite image whose color content is modulated by the nature of the transformation functions.
- This method can be based on smooth, nonlinear functions, which gives the technique considerable flexibility.

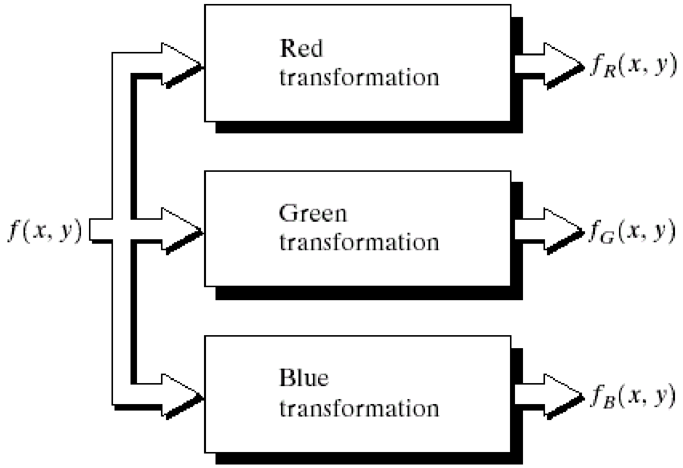
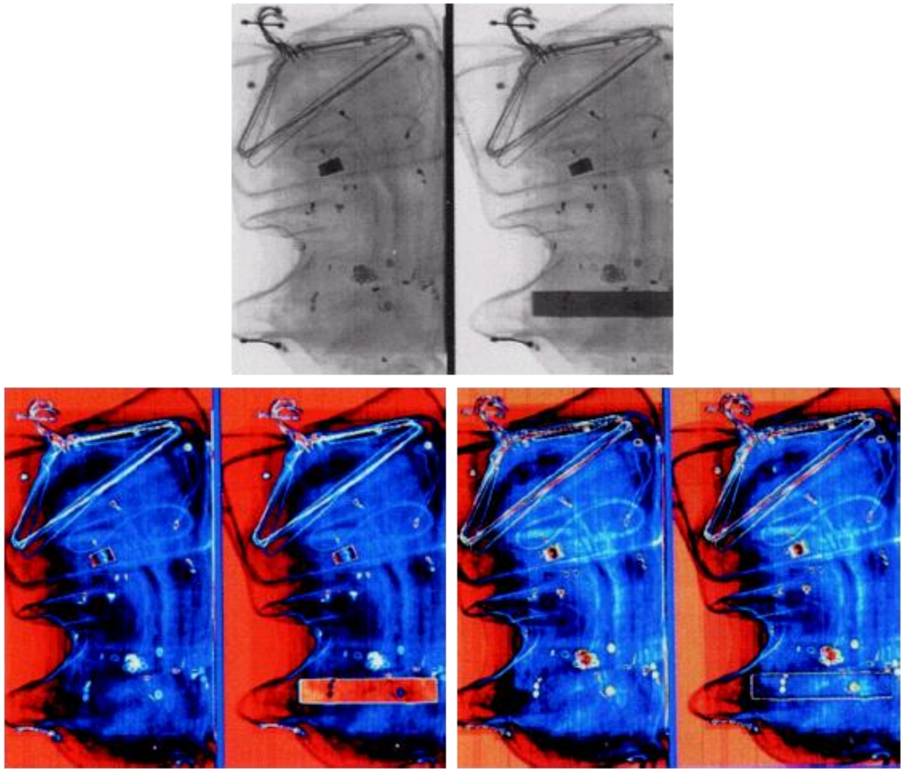


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Example: Use of Pseudocolor for Highlighting Explosive Contained in Luggage



a
b c

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.)

(a)

Shows 2 monochrome images of luggage obtained from an airport X-ray scanning system

Left: contains ordinary articles

Right: contains the same articles and a block of simulated plastic explosives.

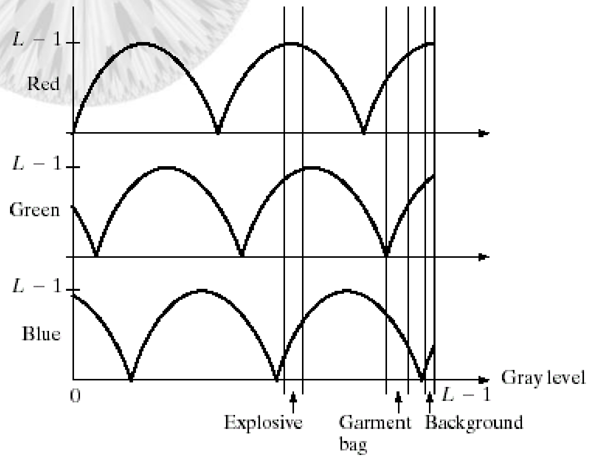
(b)

Obtained from the transformation functions in Fig. 6.25(a).

(b)

Obtained from the transformation functions in Fig. 6.25(b).

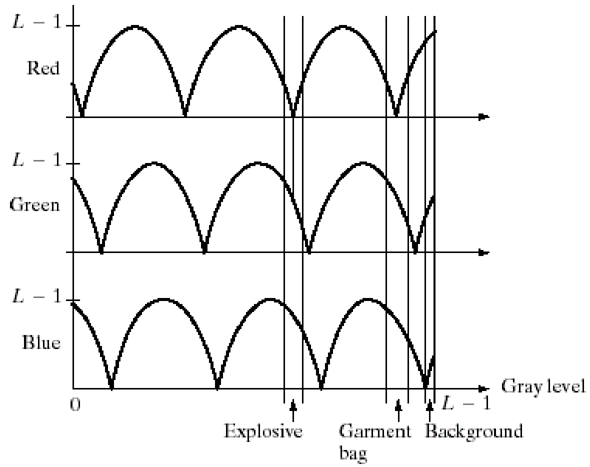
Example: Use of Pseudocolor for Highlighting Explosive Contained in Luggage (cont')



(a)

Shows the gray-level bands corresponding to the explosive, garment bag, and background, respectively.

The explosive and background have quite different gray levels, but they both were coded with approximately the same color as a result of the periodicity of the sine waves



(b)

The explosives and garment bag intensity bands were mapped by similar transformations and thus received essentially the same color assignments

This mapping allows an observer to "see" through the explosives

a
b

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

Gray Level to Color Transformation (cont')

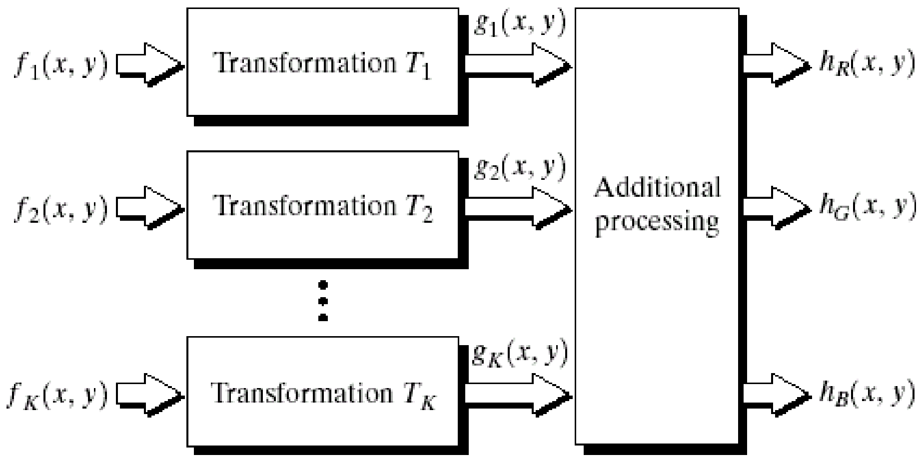
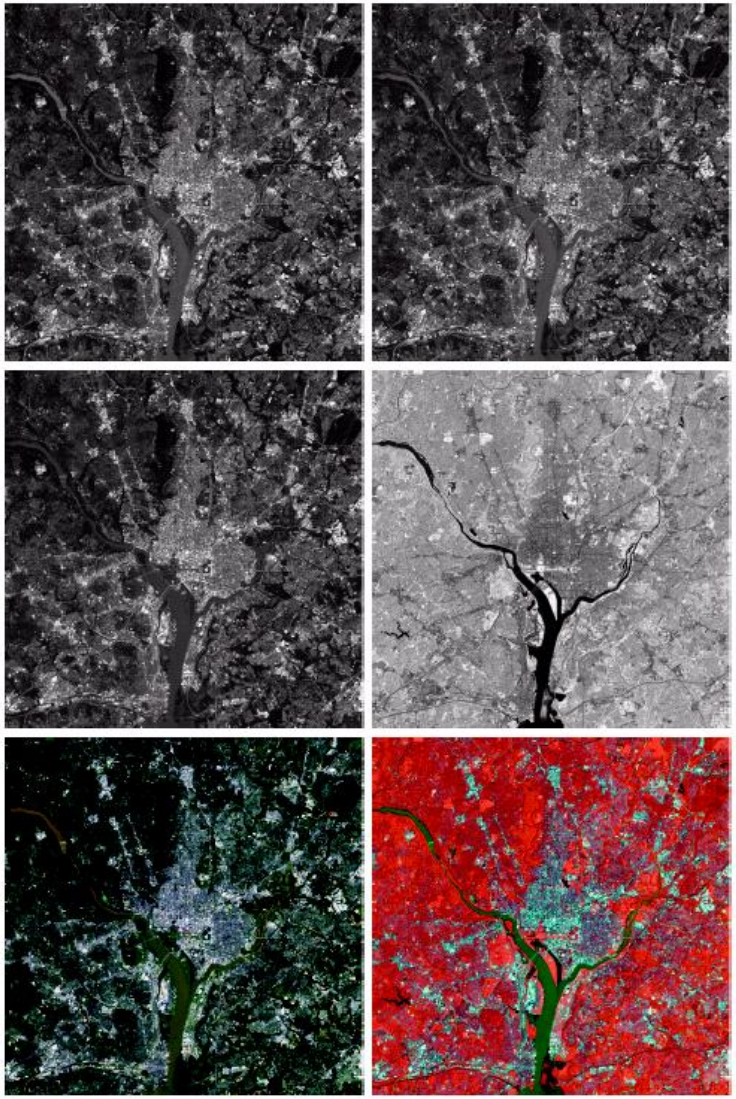


FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

Fig. 6.26
Combine several monochrome images into a single color composite.
Frequently used in multispectral image processing, where different sensors produce individual monochrome images, each image a different spectral band.
Show types of additional processes such as color balancing, combining images, and selecting the 3 images for display based on knowledge about response characteristics of the sensor used to generate the images.

Example: Color Coding of a Multispectral Images



(a) – (d)
Shows 4 spectral satellite images of Washington D.C, including part of the Potomac River.
image (a) – (c) : in the visible red, green, and blue.
Image (d) : in the near infrared.
(e)
The full color image obtained by combining the image (a), (b), and (c) into an RGB image.
Full color images of dense areas are difficult to interpret, but one notable feature of this image is the difference in color in various part of the Potomac River.
(f)
Formed by replacing the red component in (e) with the near infrared image
Shows the difference between biomass (in red) and the human-made features in the scene, composed primarily of concrete and asphalt, which appear bluish in the image

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.)

Example: Visualizing Event of Interest in Complex Image

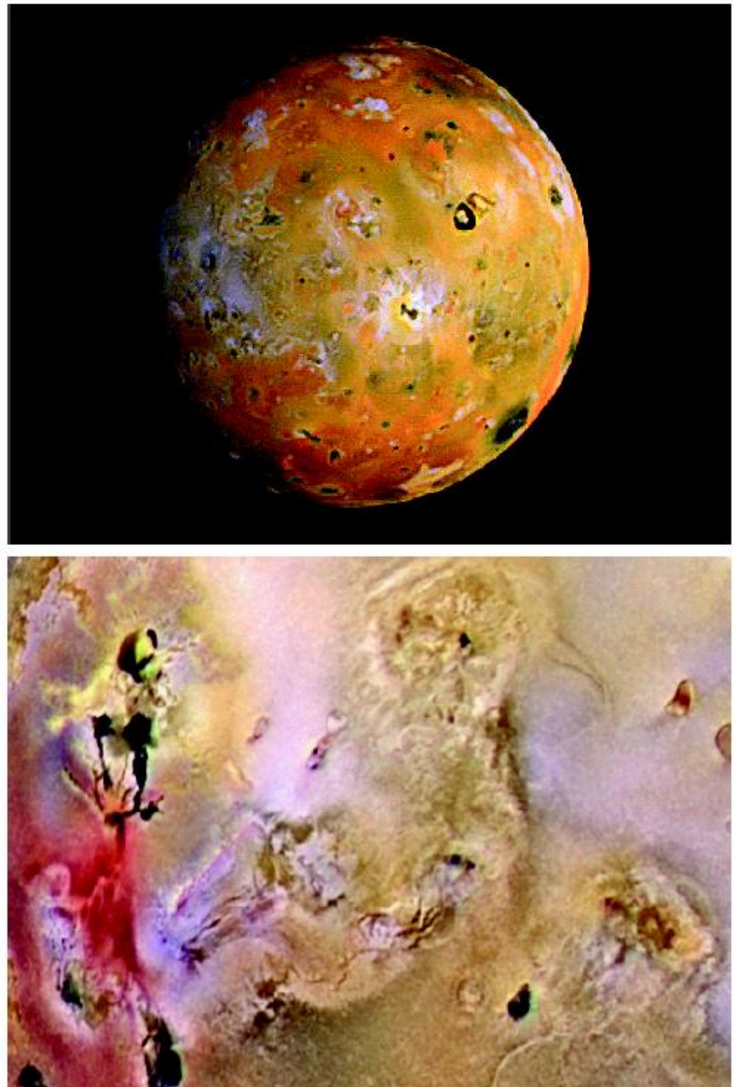
Fig. 6.28

Powerful technique in helping visualize events of interest in complex images, especially when those events are beyond our normal sensing capabilities.

(a)

Shows images of the Jupiter moon Io (in pseudocolor by combining several of the sensor images from *Galileo* spacecraft, some of which are in spectral regions not visible to the eye).

It is possible to combine the sensed image into meaningful pseudocolor map.



a
b

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

(b)

Bright red depicts material newly ejected from an active volcano on Io, and the surrounding yellow materials are older sulfur deposits

The image conveys these characteristics much more readily than would be possible by analyzing the component images individually

Basic of Full-Color Image Processing

- 2 major categories of full-color image processing:

  - process each component image individually and then form a composite processed color image from the individually processed components

  - work with color pixels directly as full-color images have at least 3 components.

- Let ***c*** represent an arbitrary vector in RGB color space: (component ***c*** are simply the RGB components of a color image at a point).

$$c = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (6.4\text{-}1)$$

- Take into account that the color components are a function of coordinates (*x*,*y*) by using the notation:

$$c(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix} \qquad (6.4\text{-}2)$$

* For an image of size M x N, there are MN such vectors, **c** (x,y), for $x = 0, 1, 2, \ldots, M-1$; $y = 0, 1, 2, \ldots, N-1$.

Basic of Full-Color Image Processing (cont')

- Image processing techniques formulated in *x* and Image processing techniques formulated in x and *y* allows us to process a color image by processing each of its component images separately, using standard gray-scale image processing methods.
- But the result of (1) <u>individual color component processing</u> are not always equivalent to (2) <u>direct processing</u> in color vector space.
- 2 conditions have to be satisfied in order for both techniques (1 & 2) to be equivalent:

  - the process has to be applicable to both vectors and scalars

  - the operation on each component of a vector must be independent of the other components

Basic of Full-Color Image Processing (cont’)

a b

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

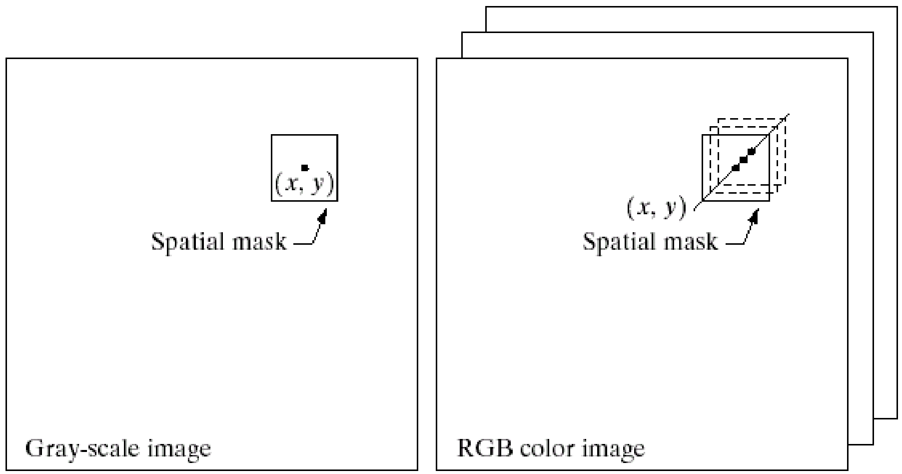


Fig. 6.29
Shows neighborhood spatial processing of gray-scale and full-color images.

e.g. neighbourhood averaging process

(a)

$$\text{Average} = \frac{\text{Sums of gray levels of all the pixels in the neighborhood}}{\text{the total number of pixels in the neighborhood}}$$

(b)

$$\text{Average} = \frac{\text{Each component of sums of all the vectors in the neighborhood}}{\text{the total number of vectors in the neighborhood}}$$

Color Transformations

- Deal with processing the components of a color image within the context of a *single* color model, as opposed to the conversion of those components between models

- Model color transformation using:

$$g(x, y) = T[f(x, y)] \qquad (6.5\text{-}1)$$

where:

$f(x,y)$: a color input image

$g(x,y)$: the transformed/processed color output image.

T : an operator on f over a spatial neighborhood in (x,y)

$$s_i = T_i(r_1, r_2, \ldots, r_n), \quad i = 1,2,\ldots,n \qquad (6.5\text{-}2)$$

where:

r_i and s_i : variables denoting the color components of $f(x,y)$ and $g(x,y)$ at any point

n : the number of color components

$\{T_1, T_2, \ldots, T_n\}$: a set of transformation/color mapping functions

Full-Color Image & Its Various Color-Space Components

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

Fig 6.30 (a)

Shows a high resolution color image of a bowl of strawberries and a cup of coffee that was digitized from a large format (4” X 5”) color negative.

2nd row

Contains the components of the initial CMYK scan

Black represent 0 and white represent 1

3rd row

CMYK image is converted to RGB

contain a large amount of red and very little green and blue

Last row

Shows the HSI components of Fig 6.30 (a)

The intensity components is a monochrome rendition of the full color original

6.5.1 :Formulation – cont.

- Any of the color space components in Fig 6.30 can be used in conjunction with Eq. (6.5-2)
- For a given transformation, the cost of converting between representations must be factored into the decision regarding the color space in which to implement it.
- E.g.: to modify the intensity of the image in Fig 6.30 (a):

$$g(x,y) = k\, f(x,y) \quad where\ 0 < k < 1 \tag{6.5-3}$$

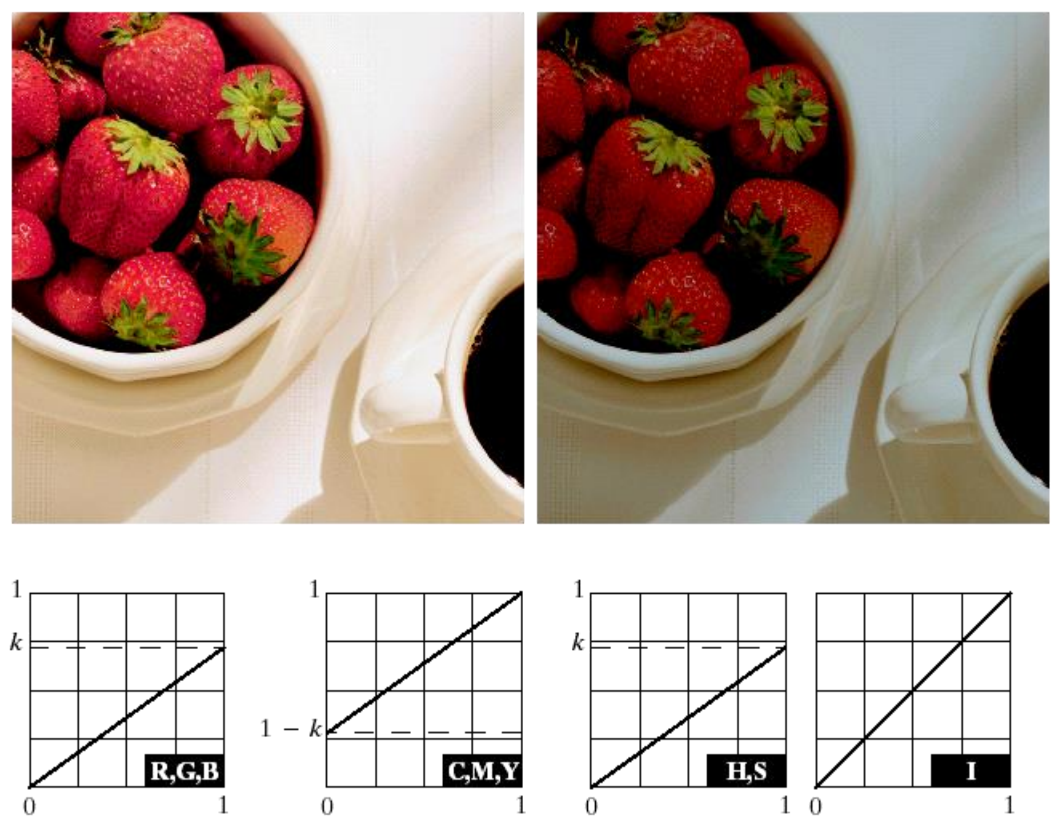
In the HSI color space	In the RGB color space	In the CMY color space
$s_3 = kr_3 \quad where \ s_1 = r_1$ $s_2 = r_2$ * Only HSI intensity component r_3 is modified.	$s_i = kr_i \quad i = 1,2,3.$	$s_i = kr_i + (1-k) \quad i = 1,2,3.$
(6.5-4)	(6.5-5)	(6.5.6)

Example: Adjusting Intensity using Color Transformation

- Each transformation defined in eq. (6.5-4) – (6.5-6) depends only on one component within its color space.
- This type of transformation is among the simplest, most used color processing tools and can be carried out on a per-color-component basis.

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.)



(b)

Shows the result of applying any of the transformation in Eq. (6.5-4) – (6.5-6) to the image of Fig 6.31 (a)

(c) – (e)

The mapping function are depicted graphically

Color Complements

- Complements : the hues directly opposite one another on the color circle. – analogous to gray scale negatives.
- Useful for enhancing detail that is embedded in dark regions of a color image, particularly when the regions are dominant in size.
- The color circle can be used to predict each of the hues in the complement image from the original image.

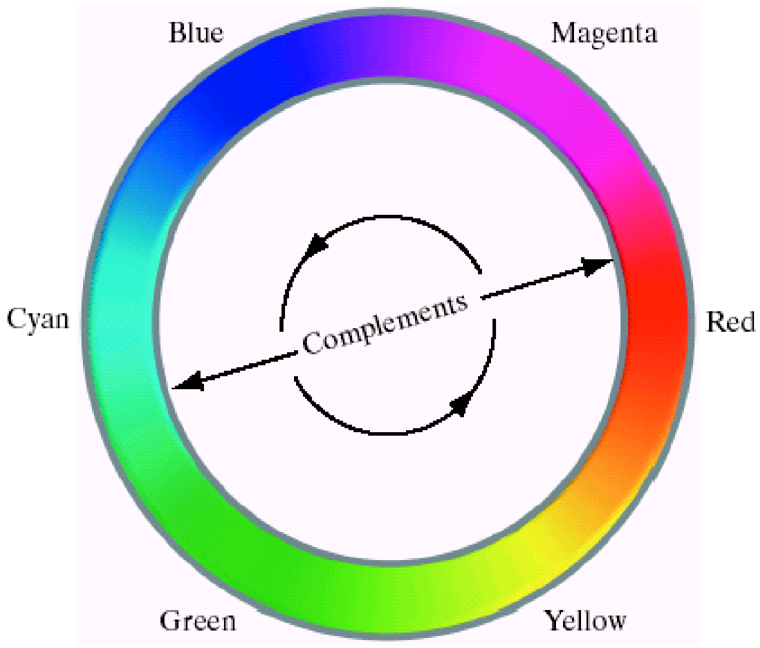


FIGURE 6.32 Complements on the color circle.

Example: Computing Color Image Complement

(b)

The RGB transformation used to compute the complement

(c)

Color complement for an image in (a)

The computed complement is reminiscent of conventional photographic color film negatives

Reds from the original image are replaced by cyan and white are replaced by black

Each of the RGB component transforms is a function of only the corresponding input color component

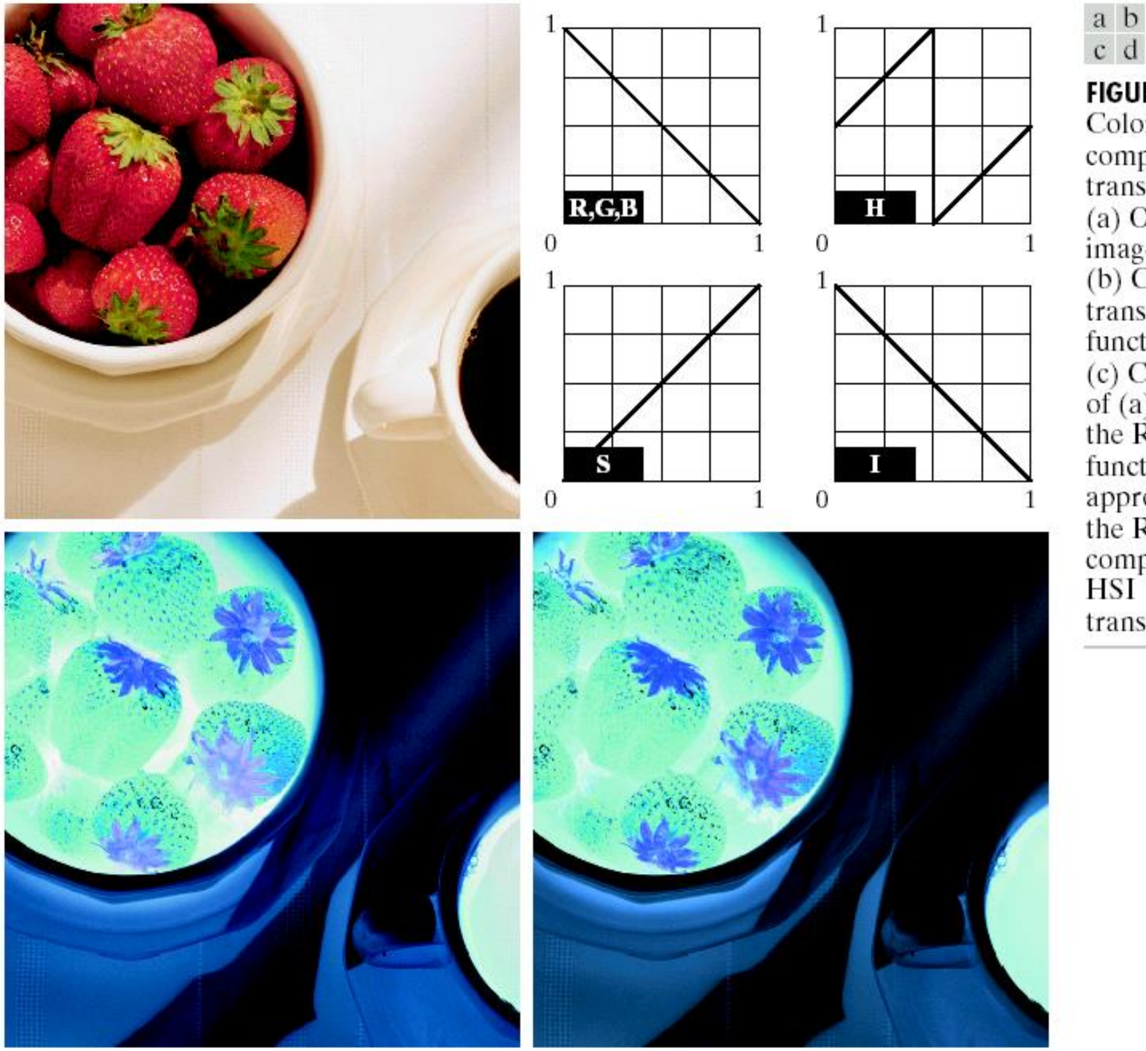


FIGURE 6.33 Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.

Color Slicing

- Basic idea of highlighting a specific range of colors in an image (useful to separate objects from their surrounding):

  - <u>display the colors of interest</u> so that they stand out from the background

  - <u>use the region defined by the colors</u> as a mask for further processing

- Simplest way to slice a color image – to map the colors outside some range of interest to a nonprominent neutral color.

- The set of transformation:

$$s_i = \begin{cases} 0.5 & if \left[\left| r_j - a_j \right| > \frac{W}{2} \right]_{any 1 \leq j \leq n,} \\ r_i & otherwise \end{cases} \quad i = 1,2,\ldots,n. \qquad (6.5\text{-}7)$$

* for the colors of interest that are enclosed by a cube/hypercube for $n > 3$, of width W and centered at a prototypical (e.g., color) color with components $(a_1, a_2, \ldots, a_n)$

Color Slicing (cont')

- To specify the color of interest for a sphere:

$$s_i = \begin{cases} 0.5 & \textbf{if } \sum_{j-1}^{n} \left(r_j - a_j\right)^2 > R_0^2 \\ r_i & \textbf{otherwise} \end{cases} \qquad i = 1,2,\ldots,n. \tag{6.5-8}$$

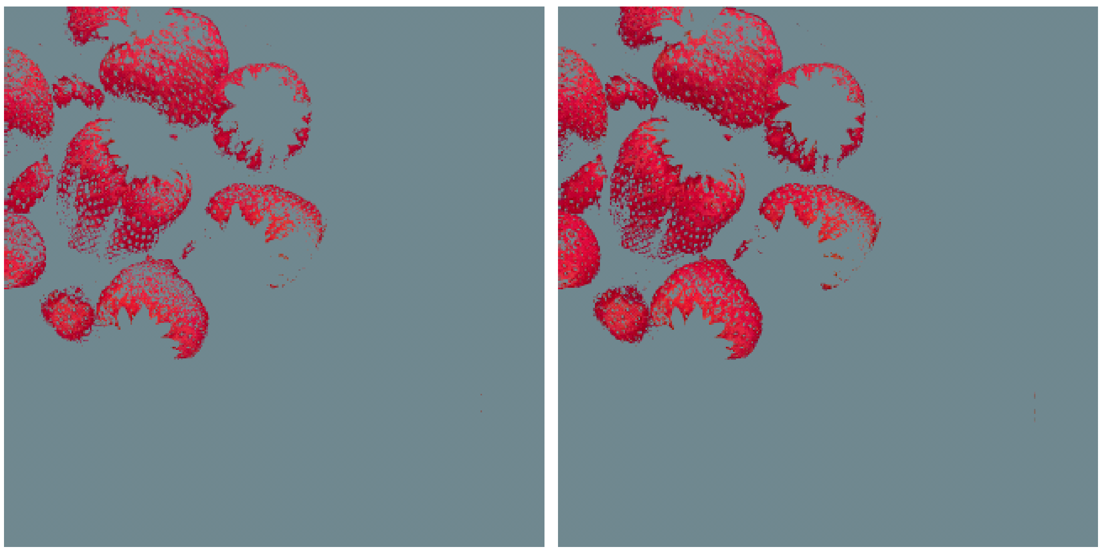
where:

- R_0 : the radius of the enclosing sphere/hypersphere for $n > 3$
- $(a_1, a_2, \ldots, a_n)$: the components of its center (the prototypical color)

- Eq. (6.5.7) and (6.5.8) can be used to implement multiple color prototypes and reduce the intensity of the colors outside the region of interest.

Example: Color Slicing

Eq. 6.5-7 & 6.5-8 can be used to separate the edible part of the strawberries from background cup, bowl, coffee and table.



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).

Fig 6.34

Show the results of applying both transformations

A prototype red with RGB color coordinate (0.6863, 0.1608, 0.19922) was selected from the most prominent strawberry

W and R_0 were chosen so that the highlighted region would not expand to undesirable portions of the image

Tone and Color Corrections

- Color transformation performed with digital cameras allows <u>tonal adjustments and color corrections.</u>

- *A device independent color model* – <u>allows a monitor/output devices to represent accurately any digitally scanned source images and final printed output</u> by relating the color gamuts of the monitors and output devices to one another.

- The success of this approach is a function of the quality of the *color profiles* used to map each device to the model and the model itself. An example is CIE *L*a*b** model or sometimes referred to as CIELAB.

- <u>CIE *L*a*b** model</u> :
  - an excellent decoupler of intensity (*L** : lightness) and color (*a** : red minus green, *b** : green minus blue)
  - useful in both image manipulation (tone and contrast editing) and image compression applications.

Tone and Color Corrections (Cont')

- The L^*a^*b components are:

$$L^* = 116 \bullet h\left(\frac{Y}{Y_w}\right) - 16 \qquad (6.5\text{-}9)$$

$$a^* = 500 \bullet \left[h\left(\frac{Y}{X_w}\right) - h\left(\frac{Y}{Y_w}\right)\right] \qquad (6.5\text{-}10)$$

$$b^* = 200 \bullet \left[h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right)\right] \qquad (6.5\text{-}11)$$

where :

X_W, Y_W, Z_W : reference white tristimulus values

and

$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.787q + 16/116 & q \le 0.008856 \end{cases} \qquad (6.5\text{-}12)$$

Example :Tonal Transformations

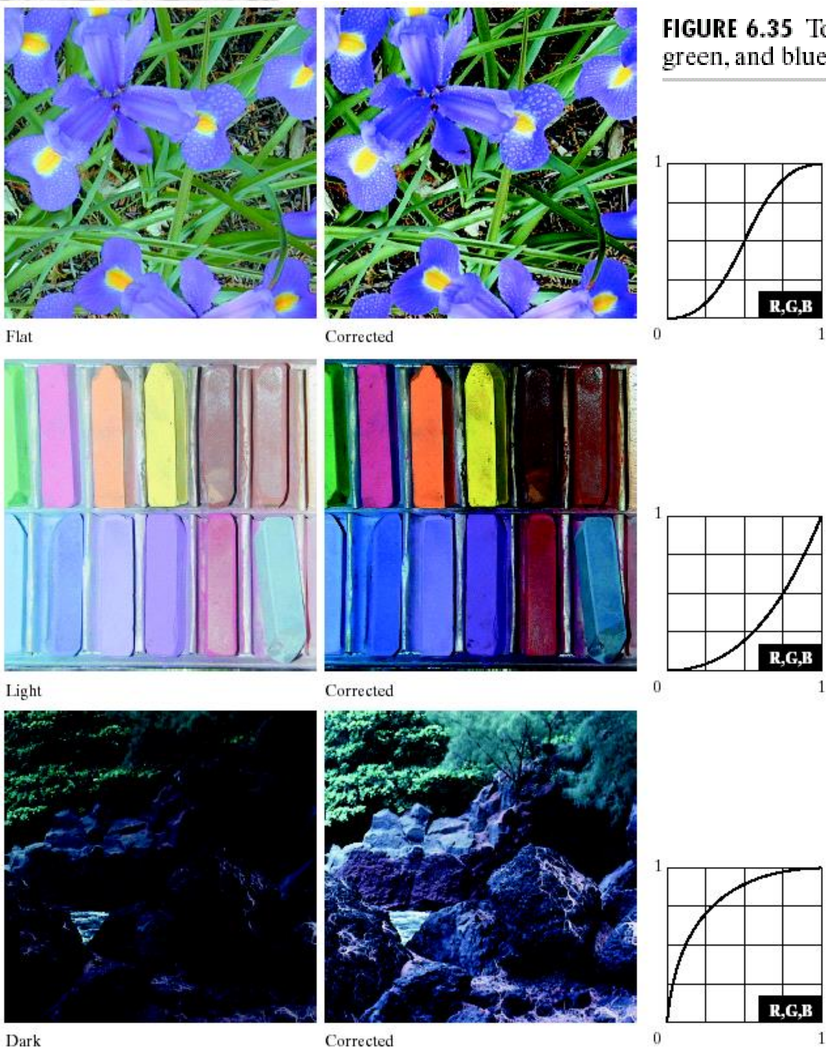


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.

Fig 6.35
Show typical transformations used to correct 3 common tonal imbalances (flat, light, and dark images)
Determined using the histograms of the images' color components

1st row
The S-shaped curve is ideal for boosting contrast
Its midpoint is anchored so that highlight and shadow areas can be lightened and darkened, respectively.
The inverse curve ca be used to correct excessive contrast

2nd and 3rd row
The transformations correct light and dark images
Reminiscent of the power-law transformation.

Example: Color Balancing

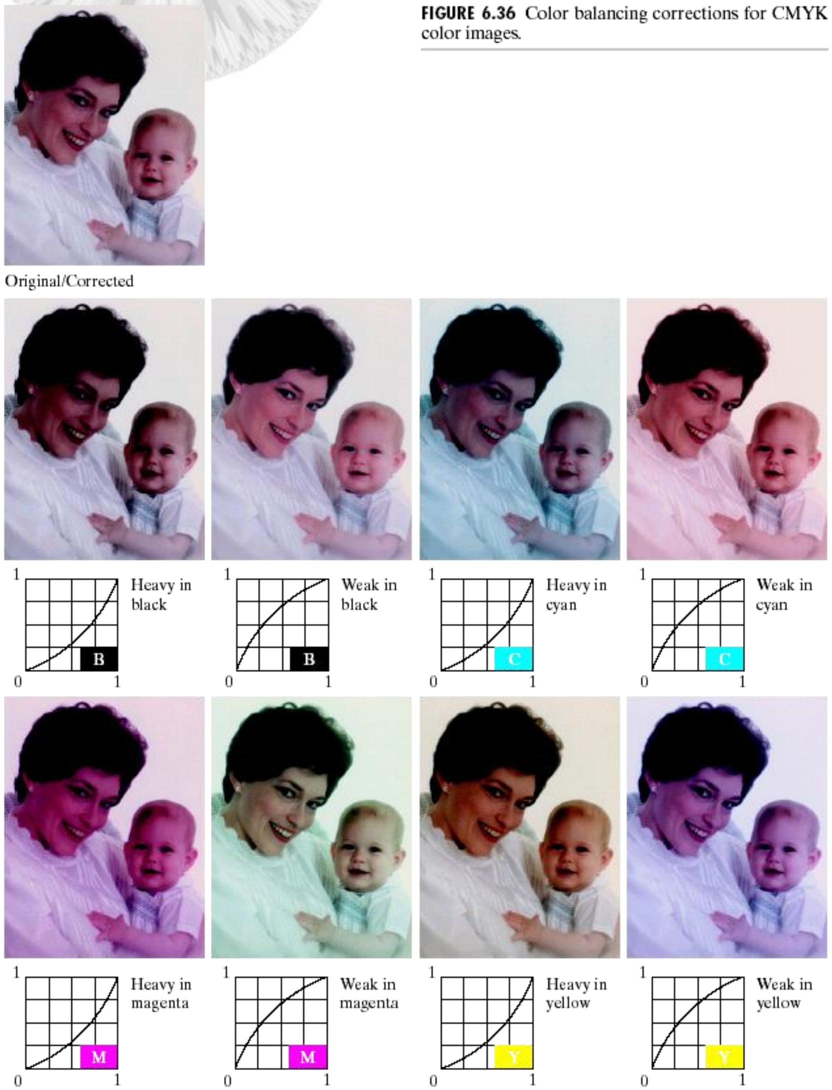


FIGURE 6.36 Color balancing corrections for CMYK color images.

Fig 6.36
Skin tones are excellent subjects for visual color assessments are possible when white areas, where the RGB/CMY(K) components should be equal, are present
Vivid colors (e.g: bright red objects) are little value when it comes to visual color assessment
Shows the transformations used to correct simple CMYK output imbalances.
The transformations depicted are the functions required for correcting the images
The inverses of these functions were used to generate the associated color imbalances
Thus, the images are analogous to a color ring-around print of a darkroom environment and are useful as a reference for identifying color printing problems
2nd row (right) : too little cyan
3rd row (left) : too much red due to excessive magenta

Histogram Processing

- The gray-level histogram processing transformations can be applied to color images in an automated way.
- Histogram equalization automatically determines a transformation that seeks to produce an image with a uniform histogram of intensity values
- Since color images are composed of multiple components, consideration must be given to adapting the gray-scale technique to more than one component and/or histogram.
- An erroneous color is produced when a histogram equalize the components of a color image independently.
- Solution: spread the color intensities uniformly, leaving the colors themselves unchanged.
- Even the values of hue and saturation are unchanged, the process did impact the overall perception.

Example: Histogram Equalization in HIS Color Space

(a)

Shows a color image of a caster stand containing cruets and shaker

(b)

Shows the intensity transformation used to equalize the intensity component.

(c)

Result of histogram equalizing the intensity component (without altering the hue and saturation).

(d)

Result of correcting this partially by increasing the image's saturation component

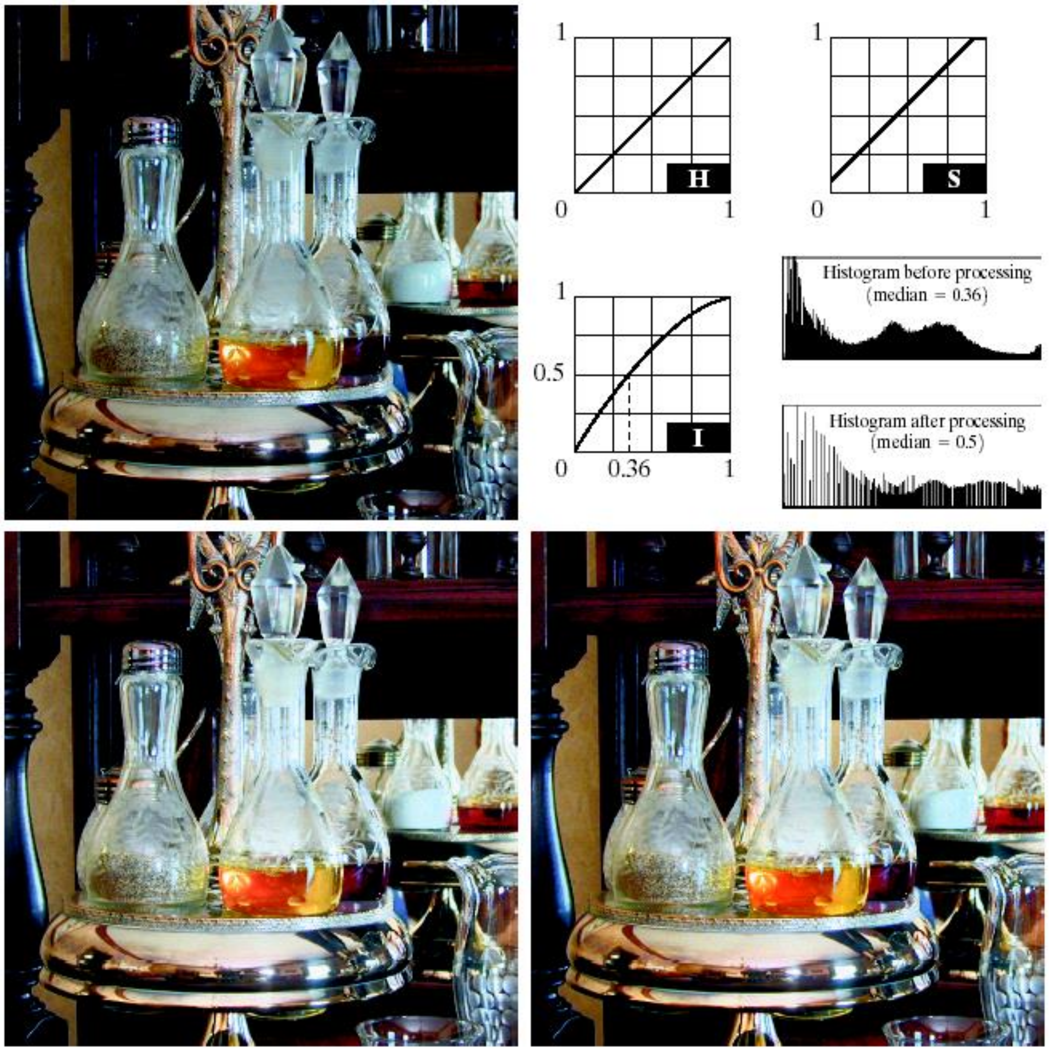


FIGURE 6.37 Histogram equalization (followed by saturation adjustment) in the HSI color space.

Color Image Smoothing

- Gray-scale image smoothing can be viewed as a <u>spatial filtering operation</u> (the coefficients of the filtering mask are all 1's)
- When the mask is slid across the image to be smoothed, each pixel is replaced by the average of the pixels in the neighborhood defined by the mask.
- Let S_{xy} denote the set of coordinates defining a neighborhood centered at (x, y) in an RGB color image.
- The average of the RGB component vectors in this neighborhood:

$$\bar{c}(x,y) = \frac{1}{K} \sum_{(x,y)\in S_{xy}} c(x,y) \qquad (6.6\text{-}1)$$

Color Image Smoothing (Cont')

- The properties of vector addition:

$$c(x,y) = \begin{bmatrix} \frac{1}{K}\sum_{(x,y)\in S_{xy}} R(x,y) \\ \frac{1}{K}\sum_{(x,y)\in S_{xy}} G(x,y) \\ \frac{1}{K}\sum_{(x,y)\in S_{xy}} B(x,y) \end{bmatrix} \qquad (6.6\text{-}2)$$

- The components of this vector is recognize as the scalar images that would be obtained by independently smoothing each plane of the starting RGB image using conventional gray-scale neighborhood processing
- Conclusion: smoothing by neighborhood averaging can be carried out on a per-color-plane basis/using RGB color vectors.

Example: Color Image Smoothing by Neighborhood Averaging

a	b
c	d

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

Fig. 6.38 (a)

The RGB color image

(a)

Red planes of the image

(b)

Green planes of the image

(d)

Blue planes of the image

Example: Color Image Smoothing by Neighborhood Averaging

- HSI color model are suitable for many gray-scale processing techniques.
- It is more efficient to smooth only the intensity component of the HSI representation

Fig. 6.39
The RGB image in Fig 6.38 (a) can be smoothen by using the 5 X 5 gray-level averaging mask
Simply smooth independently each of the RGB color planes and then combine the processed planes to form a smoothed full-color result

(a) - (c)
Show the image's HSI components

a b c

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

Example: Color Image Smoothing by Neighborhood Averaging

Fig 6.40
Smooth only the intensity component (do not modified the hue and saturation) and convert the processed result to an RGB image for display.
(a) and (b) looks similar, but not identical
This due to the fact that the average of 2 pixels of differing color is a mixture of 2 colors.

(b)
The smoothed color image
By smoothing only the intensity image, the pixels maintain their original hue and saturation, thus their original color
The difference between the smoothed result would increased as the size of the smoothing mask increases

a b c

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

(a) - (c)
Show the image's HSI components

Color Image Sharpening

- Considering image sharpening using the Laplacian.
- The Laplacian of a vector is defined as a vector whose components are equal to the Laplacian of the individual scalar components of the input vector.
- In the RGB color system, the Laplacian vector:

$$\nabla^2\left[c(x,y)\right]=\begin{bmatrix}\nabla^2 R(x,y)\\ \nabla^2 G(x,y)\\ \nabla^2 B(x,y)\end{bmatrix} \tag{6.6-3}$$

- The Laplacian of a full-color image can be compute by computing the Laplacian of each image separately.

Example: Color Image Sharpening

(a)

Obtained using Eq. (3.7-6) to compute the Laplacians of the RGB component images in Fig 6.38 and combining them to produce the sharpened full-color result

(b)

Shows a similar sharpened image based on HSI component in Fig 6.39

Generated by combining the Laplacian of the intensity component with the unchanged hue and saturation components

(c)

Shows the difference between the RGB and HIS-based

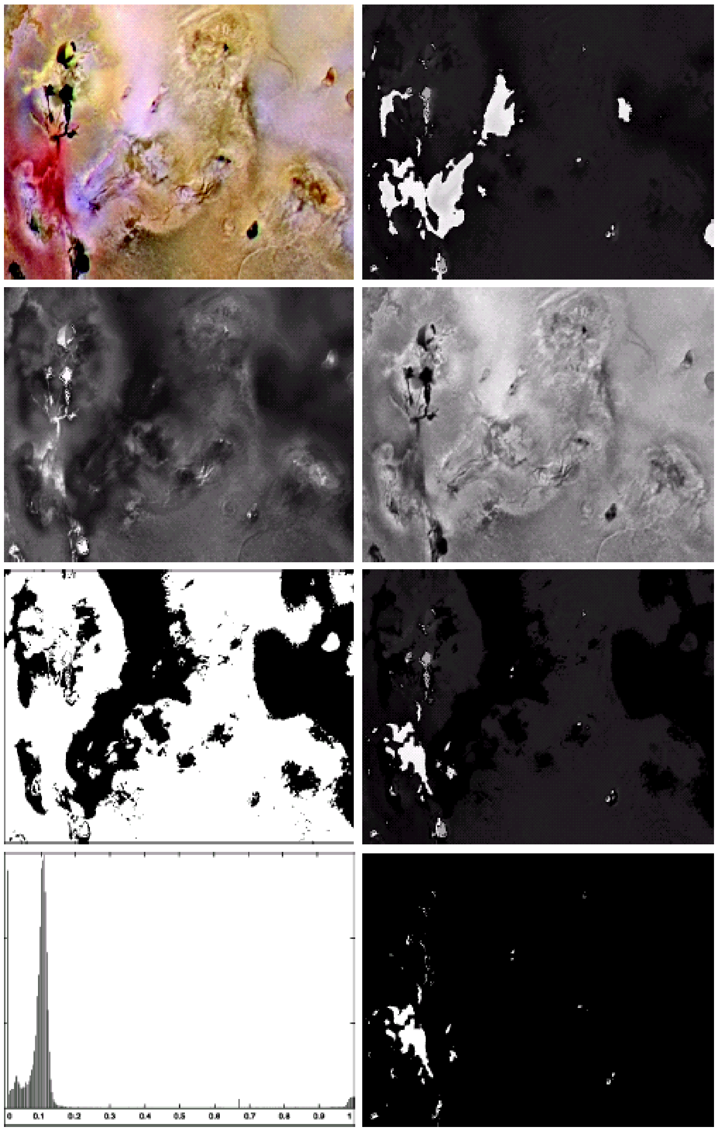
a b c

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.

Color Segmentation in HSI Color Space

- If we want to segment an image based on both the color and individual planes, consider the HSI space because color is conveniently represented in the hue image.
- Saturation is used as a masking image in order to isolate further regions of interest in the hue image.
- The intensity image is used less frequently for segmentation of color images because it carries no color information

Example: Segmentation in HSI Color Space



(a)

Interest: to segment the reddish region in the lower left

Even it was generated by pseudocolor methods, it still can be segmented as a full-color image without loss of generality

(b) – (d)

Its HSI component images

(b) shows that the target region has relatively high values of hue (the colors are on the blue-magenta side of red)

(e)

Shows a binary mask generated by thresholding the saturation image with a threshold = 10% of the maximum value in the saturation image.

Any pixel value > threshold was set to 1 (white)

Other values was set to 0 (black)

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).

Example: Segmentation in HSI Color Space (cont')

(f)
The product of the mask with the hue image

(g)
The histogram of the product image
The gray-scale is in the range[0, 1]
High values (target region) are grouped at the very high end of the gray scale (near 0.1)

(h)
The result of thresholding the product image with threshold value of 0.9 (a binary image)
The spatial location of the white points identifies the points in the original image that have the reddish hue of interest.
The region shown in white are about the best that this method can do in identifying the reddish components of the original image

Segmentation in RGB Vector Space

- Better result for segmentation are obtained by using RGB color vectors.
- Objective of segmentation: to classify each RGB pixel in a given image as having a color in the specified range or not.
- The Euclidean distance between **z** and **a**:

$$\begin{aligned} D(z,a) &= \| z - a \| \\ &= \left[(z-a)^T (z-a) \right]^{\frac{1}{2}} \\ &= \left[(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2 \right]^{\frac{1}{2}} \end{aligned} \tag{6.7-1}$$

where:

subscripts R,G, B : the RGB components of vectors **a** and **z**.
a : the RGB vector for average color.
z : an arbitrary point in RGB space.
z is similar **a** if the distance between them is lest than a specified threshold, D_0.

Segmentation in RGB Vector Space (cont')

- A generalization of eq. (6.7-1) produces a distance measure:

$$D(z,a) = \left[(z-a)^T C^{-1}(z-a)\right]^{\frac{1}{2}} \qquad (6.7\text{-}2)$$

where:

C : the covariance matrix of the samples representatives of the color to segment

- When **C** = **I**, the 3X3 identity matrix in Eq. (6.7-2) reduces to Eq. (6.7-1)
- As distance are positive and monotonic, one can work with the distance squared, avoiding root computations.
- Implementing both equation is computationally expensive for images of practical size, even if the square roots are not computed.
- Solution: use a bounding box in Fig 6.43 (c)

Segmentation in RGB Vector Space (cont')

(a)

The locus points such that D (**z**, **a**) $\leq D_0$ is a solid sphere of radius D_0.

Points within/on the surface of the sphere satisfy the specified color criterion.

(b)

The locus points such that D (**z**, **a**) $\leq D_0$ describes a solid 3-D elliptical body with the important property that its principal axes are oriented in the direction of maximum data spread.

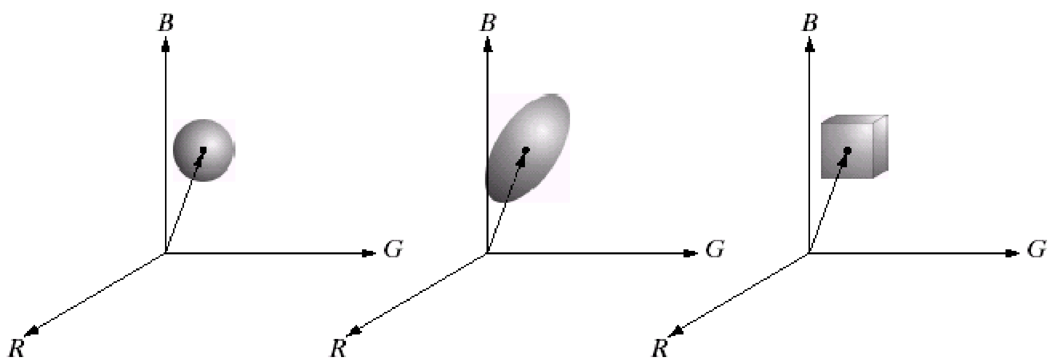
(c)

The box is centered on **a**, and its dimensions along each of the color axes is chosen proportional to the standard deviation of the samples along each of the axes

Computation of the standard deviations is done only using sample color data.

With an arbitrary color point, one can segment it by determining whether or not it is on the surface/inside the box, as with the distance formulations

Determining whether a color point is inside/outside the box is much simpler computationally when compared t spherical/elliptical enclosure



a b c

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

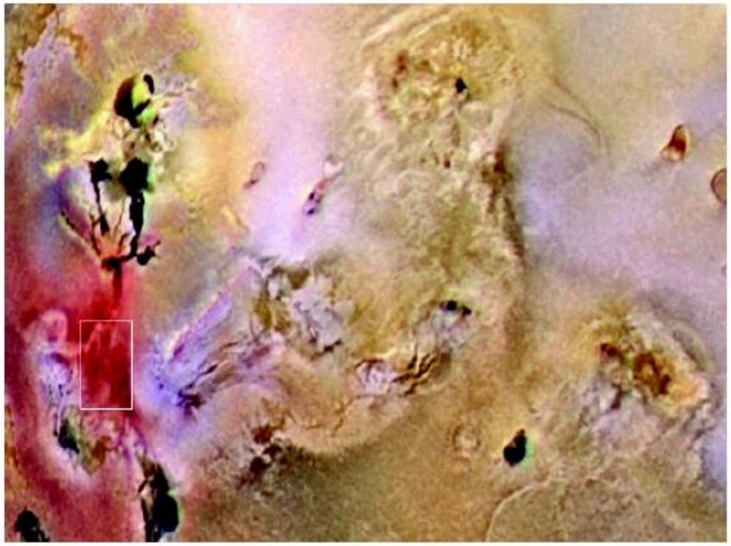
Example: Segmentation in RGB Vector Space

(a)

Shows a rectangular region that contains samples of reddish color to be segment out of the color image

Approach: 1) compute the mean vector **a** using the color points contained within the rectangle

2) Compute the standard deviation of the red, green and blue values of those samples



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).

(b)

Shows the result of coding each point in the entire color image as

i) white if it was on the surface/inside the box

ii) Black for otherwise

Segmentation in RGB vector space yielded more accurate results

Color Edge Detection

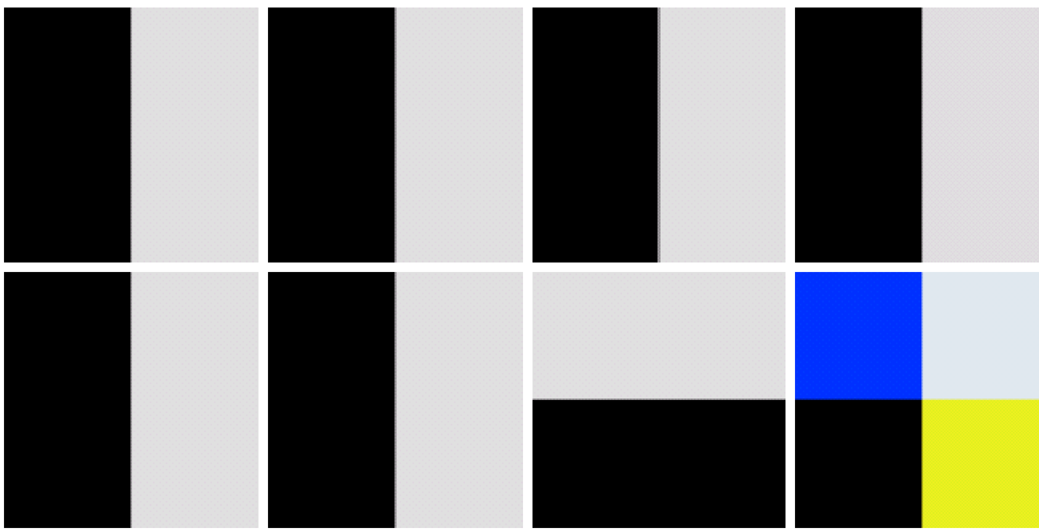
- Edge detection: important tool for image segmentation.
- Interest : to compute edges on an individual-image basis vs computing edges directly in color vector space
- Edge detection by gradient operators:

  - the gradient discussed earlier in sec 3.7.3 is not defined for vector quantities.
  - consider the two *M x M* (M odd) color images

  - if one compute the gradient image of each the component images and add the results to form the 2 corresponding RGB gradient images, the value of the gradient at point [(M+1)/2, (M+1)/2] would be the same in both cases.

  - the gradient at that point in Fig. 6.45 (d) was supposed to be stronger than Fig 6.45 (h) because all 3 edges in (d) are in the same direction compared to (h) with only 2 edges are in the same direction.
  - thus, computing the gradient on individual images and then using the results to form a color image will lead to erroneous results

Color Edge Detection (Cont')



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.

(d)
Composed of the 3 component images in (a) – (c)
Expectation: the gradient at that point to be stronger because the edges of the R, G, and B images are in the same direction in that image

(h)
Composed of the 3 component images in (e) – (g)
Expectation: the gradient at that point to be weaker because only 2 of the edges are in the same direction.

Color Edge Detection (Cont')

- Let **r**, **g**, and **b** be unit vectors along the R, G, and B axis of RGB color space:

$$\mathbf{u} = \frac{\partial R}{\partial x}\mathbf{r} + \frac{\partial G}{\partial x}\mathbf{g} + \frac{\partial B}{\partial x}\mathbf{b} \qquad (6.7\text{-}3)$$

$$\mathbf{v} = \frac{\partial R}{\partial y}\mathbf{r} + \frac{\partial G}{\partial y}\mathbf{g} + \frac{\partial B}{\partial y}\mathbf{b} \qquad (6.7\text{-}4)$$

- Let g_{xx}, g_{yy} and g_{xy} be defined in terms of the dot product of these vectors:

$$\mathbf{g}_{\mathbf{xx}} = \mathbf{u.u} = \mathbf{u}^T\mathbf{u} = \left|\frac{\partial R}{\partial x}\right|^2 + \left|\frac{\partial G}{\partial x}\right|^2 + \left|\frac{\partial B}{\partial x}\right|^2 \qquad (6.7\text{-}5)$$

$$\mathbf{g}_{\mathbf{yy}} = \mathbf{v.v} = \mathbf{v}^T\mathbf{v} = \left|\frac{\partial R}{\partial y}\right|^2 + \left|\frac{\partial G}{\partial y}\right|^2 + \left|\frac{\partial B}{\partial y}\right|^2 \qquad (6.7\text{-}6)$$

$$\mathbf{g}_{\mathbf{xy}} = \mathbf{u.v} = \mathbf{u}^T\mathbf{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y} \qquad (6.7\text{-}7)$$

Color Edge Detection (Cont')

- The direction of maximum rate of change of **c** (x, y) :

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2\mathbf{g}_{\mathbf{xy}}}{\left(\mathbf{g}_{\mathbf{xx}} - \mathbf{g}_{\mathbf{yy}}\right)}\right] \qquad (6.7\text{-}8)$$

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

$$F(\theta) = \left\{\frac{1}{2}\left[\left(\mathbf{g}_{\mathbf{xx}} + \mathbf{g}_{\mathbf{yy}}\right) + \left(\mathbf{g}_{\mathbf{xx}} - \mathbf{g}_{\mathbf{yy}}\right)\cos 2\theta + 2\mathbf{g}_{\mathbf{xy}}\sin 2\theta\right]\right\}^{\frac{1}{2}} \qquad (6.7\text{-}9)$$

Example: Color Edge Detection

(b)

The gradient of the image in (a)

Obtained using the vector method

Contains extra details that is worth the added computational burden

(c)

Shows the image obtained by computing the gradient of ach RGB component image and forming a composite gradient image by adding the corresponding values of the 3 component images at each coordinate

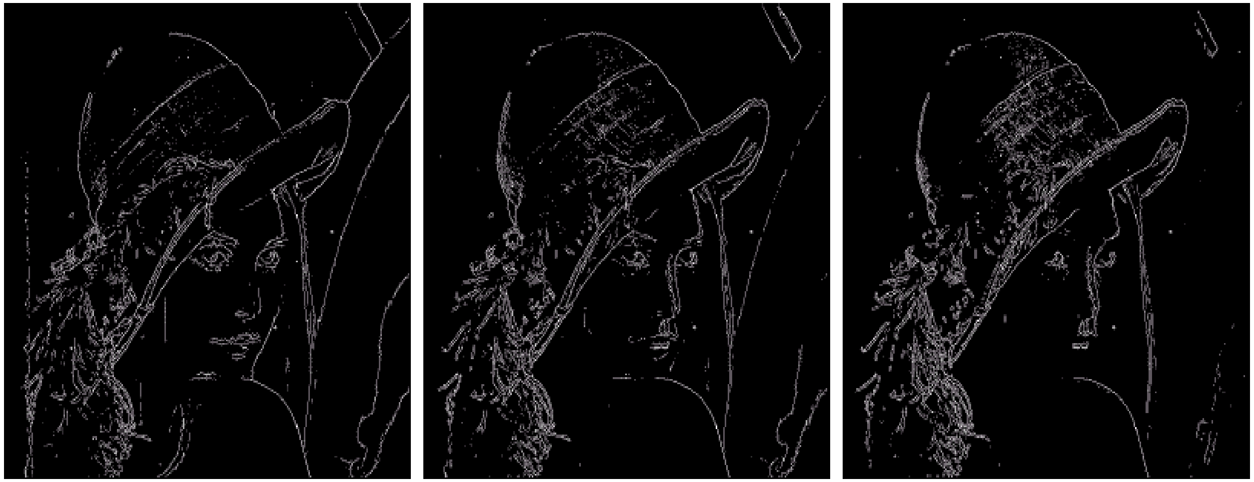
(d)

Shows the difference between the 2 gradient images at each point (x,y)

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).

Example: Color Edge Detection (cont')



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).

(b)

Shows the 3 component gradient images

this 3 component can be added and scaled to obtain Fig 6.46 (c)

Noise in Color Images

- Usually, the noise content of a color image has the same characteristics in each color channel.
- However, it is possible for color channels to be affected differently by noise.
- E.g.: the possibility for the electronics of a particular channel to malfunction
- Different noise levels are more likely to be caused by differences in the relative strength of illumination available to each of the color channels.
- E.g.: the use of a red (reject) filter in a CCD camera will reduce the strength of illumination available to the red sensor.
- CCD sensors are noisier at lower levels of illumination, so the resulting red component of an RGB image would tend to be noisier than the other 2 component images in this example.

Noise in Color Images (cont')

Fig. 6.48

A brief look at noise in color images and how noise carries over when converting from one color model to another

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).]

(a) – (c)

Show the 3 color planes of an RGB image corrupted by Gaussian noise

(d)

The composite RGB image.

The fine grain noise such as this tends to be less visually noticeable in a color image than it is in a monochrome image.

Noise in Color Images (cont')

(a) – (c)
Show the result of converting the RGB image in Fig 6.48 (d) to HSI.
Significantly degraded the hue and saturation components of the noisy image
This is due to the fact that the intensity image is average of the RGB images. (i.e image averaging reduces random noise)

a b c

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

Noise in Color Images (cont')

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.

* If only one RGB channel is affected by noise, conversion to HS spreads the noise to all HSI component images.

(a)
Show an RGB image whose green image is corrupted by salt-and-pepper noise (probability of either salt/pepper = 0.05)

(b) - (d)
Show how the noise spread from the green RGB channel to all the HSI images

Color Image Compression

- Data compression :

  - play a central role in the storage and transmission of color images
  - reason: the number of bits required to represent color is typically 3 to 4 times greater than the number employed in the representation of gray levels.

- Data :

  - the components of each color pixel
  - e.g: the red, green, and blue components of the pixels in an RGB image.
  - they are the means by which the color information is conveyed

- Compression :

  - the process of reducing/eliminating redundant and/or irrelevant data.

Color Image Compression (cont')

a b
c d

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

(a)
Show a 24-bit RGB full-color image of an iris
8 bits each are used to represent the red, green, and blue components.

(b) - (d)
constructed from a compressed version of the image (a)
A compressed and subsequently decompressed approximation of (a)
Contains only 1 data bit (1 storage bit) for every 230 bits of data in the original image
The compressed image is not directly displayable, it must be decompressed before input to a color monitor
This reconstructed approximation image is slightly blurred
Lossy compression techniques: can be reduced/eliminated by altering the level of compression.