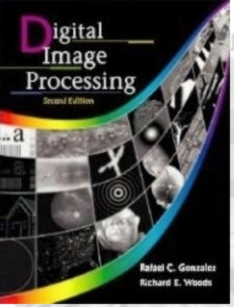


Digital Image Processing

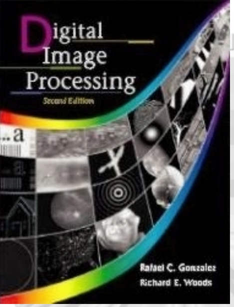
Chapter 6

Color Image Processing

Dr. Kai Shuang

Department of Electronic Engineering
China University of Petroleum
shuangkai@cup.edu.cn

Some slides and illustrations from Dr. Jimin Liang and Dr. Nawapak Eua-Anant



Preview

- In automated image analysis, color is a powerful descriptor, which simplifies object identification and extraction.
- The human eye can distinguish between thousands of color shades and intensities but only about 20-30 shades of gray. Hence, use of color in human image processing would be very effective.
- Color image processing consists of two parts: Pseudo-color processing and Full color processing.
- In *pseudo-color* processing, (false) colors are assigned to monochrome image. For example, objects with different intensity values maybe assigned different colors, which would enable easy identification/recognition by humans.
- In *full-color* processing, image are acquired with full color sensors/cameras. This has become common in the last decade or so, due to the easy and cheap availability of color sensors and hardware.

Color Fundamentals and Color Models

- Color and electromagnetic spectrum
- Primary colors
- Chromaticity diagram
- Color models
  - RGB model
  - CMY model
  - HSI model

Color spectrum

- When passing through a prism, a beam of sunlight is decomposed into a spectrum of colors : violet, blue, green, yellow, orange, red

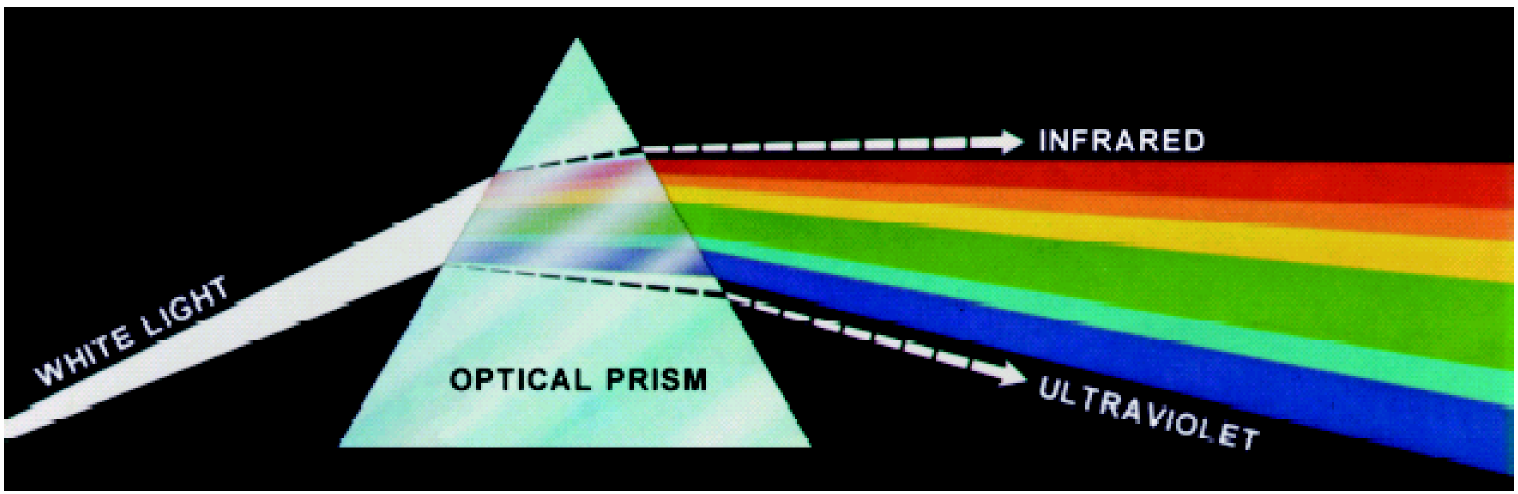


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

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

Electromagnetic spectrum

- Ultraviolet – visible light – infrared
- The longer the wavelength (meter), the lower the frequency (Hz), the lower the energy (electron volts)

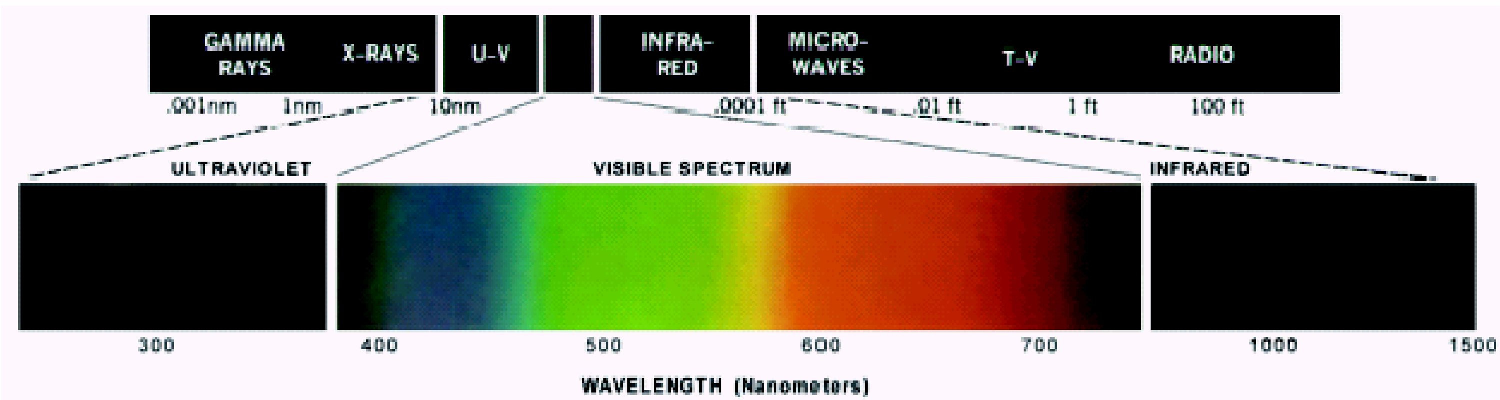


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Multispectral imaging

- LANDSAT
  - The first Landsat satellite was launched in 1972
  - Landsat 7 was launched on April 15, 1999
  - Landsat 7 sensors: Enhanced Thematic Mapper Plus (ETM+)
  - Landsat 7 Home page : http://landsat7.usgs.gov/index.php
  - NASA Landsat 7 Home page : http://landsat.gsfc.nasa.gov/

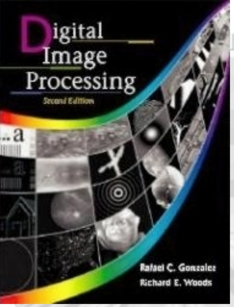
Multispectral imaging - example

Southern California Fires
Acquired on: Oct 26, 2003

Wildfires rage through the San Bernadino and San Diego counties in southern California, with as many as eight separate fires burning in the heavily populated area.

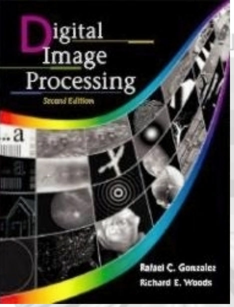
Hyperspectral imaging

- AVIRIS (Airborne Visible-Infrared Imaging Spectrometer)
  - Number of bands : 224
  - Wavelength range (um) : 0.4-2.5
  - Image size : 512 x 614
  - Home page : http://makalu.jpl.nasa.gov/html/overview.html
- Spectrum range
  - Visible light : 0.4 – 0.77 um
  - Near infrared : 0.77 – 1.5 um
  - Medium infrared : 1.6 – 6 um
  - Far infrared : 8 – 40 um



Questions

- What does it mean when we say an object is in certain color?
- Why the primary colors of human vision are red, green, and blue?
- Is it true that different portions of red, green, and blue can produce all the visible color?
- What kind of color model is the most suitable one to describe the human vision?

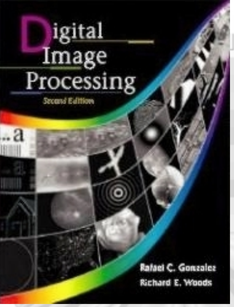


Illuminating and reflecting light

- Light source
  - Emit energy in a range of wavelength
  - Its intensity may vary in both space and time
- Illuminating sources
  - Emit light (e.g. the sun, light bulb, TV monitors)
  - Perceived color depends on the emitted freq.
  - Follows additive rule: R+G+B = White
- Reflecting sources
  - Reflect an incident light (e.g. the color dye, matte surface, cloth)
  - Perceived color depends on reflected freq (=incident freq – absorbed freq.)
  - Follows subtractive rule: R+G+B = Black

Human perception of color

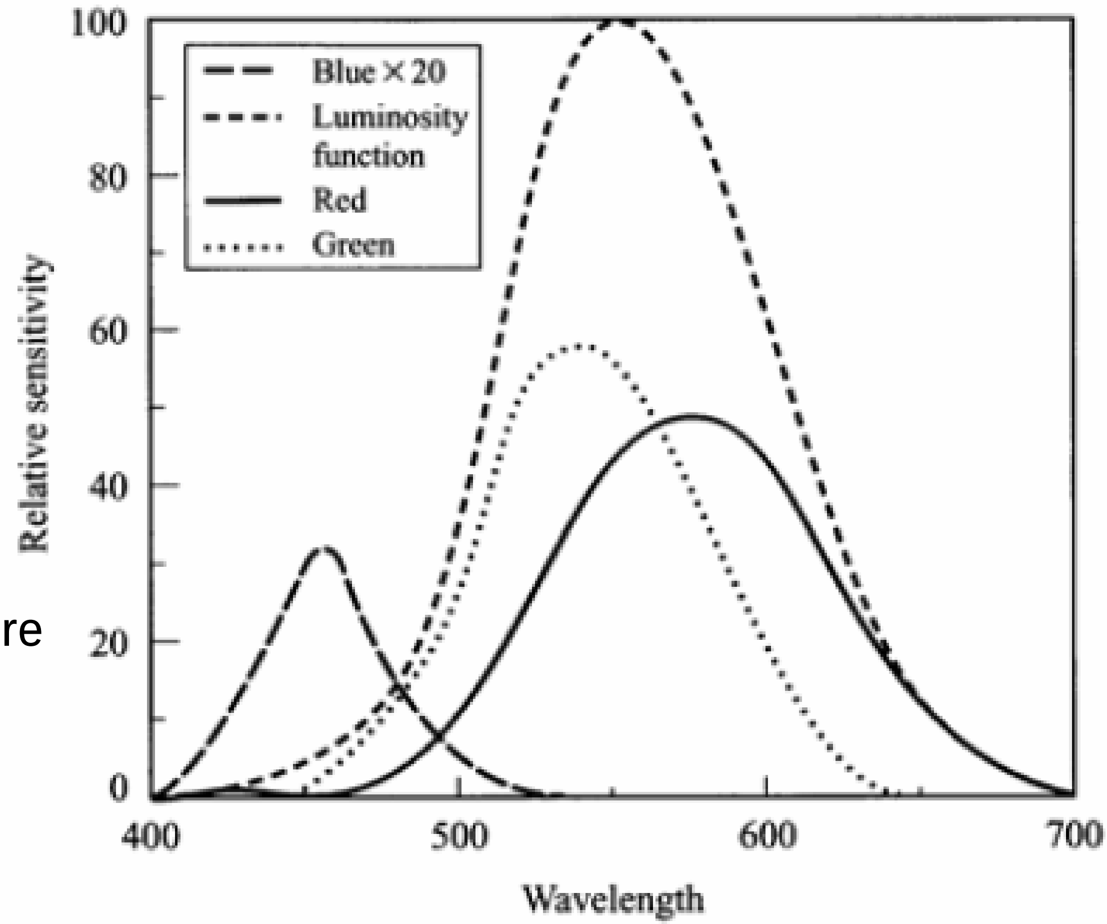
- Retina contains photo receptors
  - Rods: night vision, perceive brightness only
  - Cones : day vision, can perceive color tone
  - Cones are divided into three sensible categories
    - 65 % percent of cones are sensible to red light
    - 33 % percent of cones are sensible to green light
    - 2% percent of cones are sensible to blue light
  - Different cones have different frequency responses
  - Tri-receptor theory of color vision: the perceived color depends only on three numbers, Cr, Cg, Cb, rather than the complete light spectrum.



Human perception of color (cont'd)

$$C_i = \int C(\lambda) a_i(\lambda) d\lambda, i = r, g, b$$

$$Y = \int C(\lambda) a_Y(\lambda) d\lambda$$

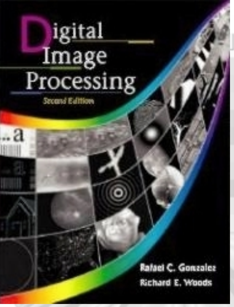


Trichromatic theory of color mixture

$$C = \sum_{k=1,2,3} T_k C_k$$

Maxwell, 1855

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



Primary colors of human vision

- For this reason, red, green, and blue are referred to as the primary colors of human vision. CIE (the international Commission on Illumination) standard designated three specific wavelength to these three colors in 1931.
  - Red : 700 nm
  - Green : 546.1 nm
  - Blue : 435.8 nm

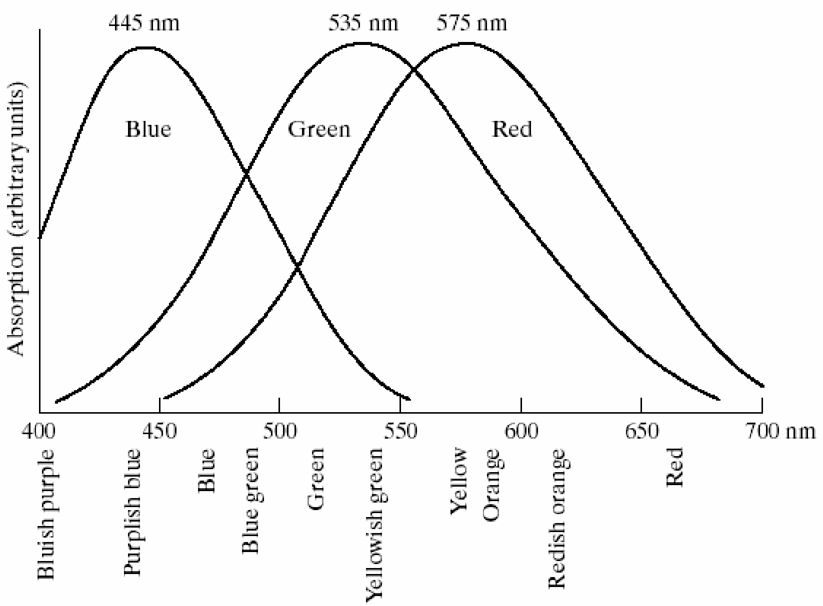


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

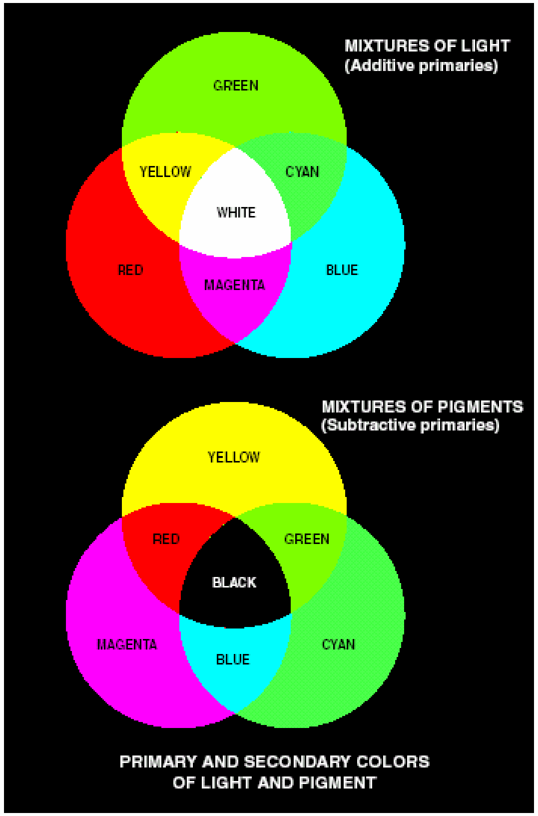
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Some clarification

- No single color may be called red, green, or blue.
- R, G, B are only specified by standard.

Secondary colors

- Magenta (R+B)
- Cyan (G+B)
- Yellow (R+G)



a
b

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

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

Color characterization

- Brightness : chromatic notion of intensity
- Hue : dominant color (dominant wavelength in a mixture of light waves) perceived by an observer
- Saturation : relative purity of the amount of white mixed with a hue

Some clarification

- So when we call an object red, orange, etc., we refer to its hue.

Chromaticity

- Chromaticity = hue + saturation
- Tristimulus : the amount of R, G, and B needed to form any color (R, G, B)
- Trichromatic coefficients : x, y, z

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

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

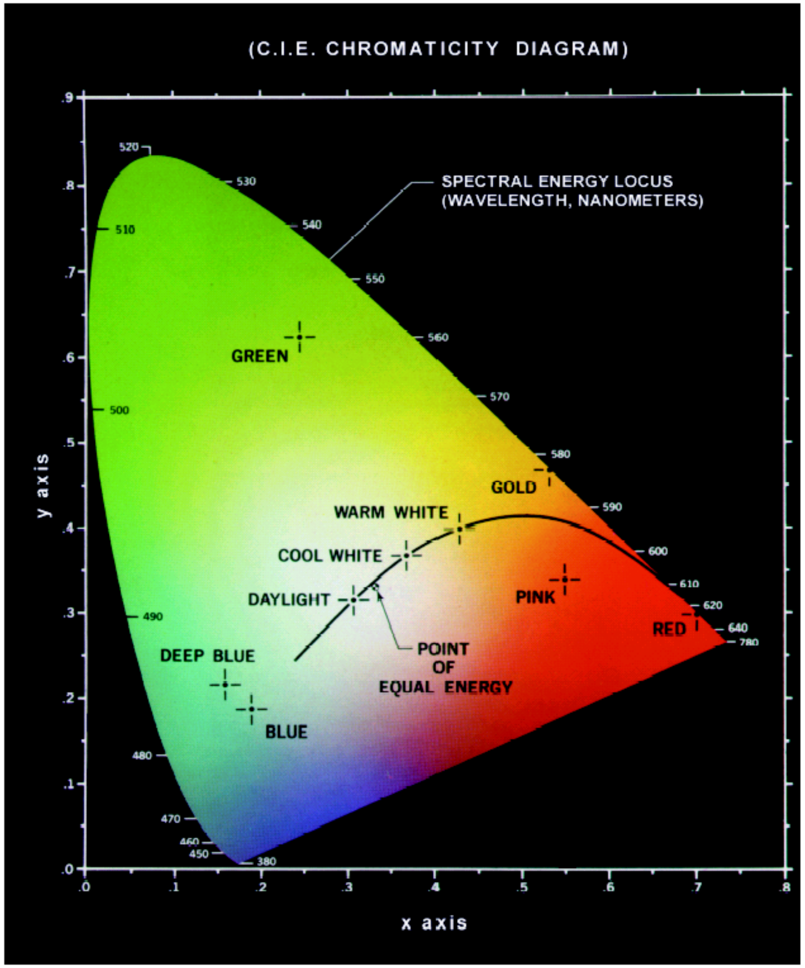
$$z = \frac{Z}{X+Y+Z}$$

$$x + y + z = 1$$

Chromaticity diagram

- CIE standard (1931)
- Shows all the possible colors
- Questions
  - Can different portions of R, G, and B create all the possible colors?
  - Where is the brown in the diagram?

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



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

Answers

- A triangle can never cover the house-shoe shape diagram
- The fixed primary colors can not produce all the visible colors.
- Chromaticity diagram only shows dominant wavelengths

  and the saturation, and is independent of the amount of luminous energy (brightness)

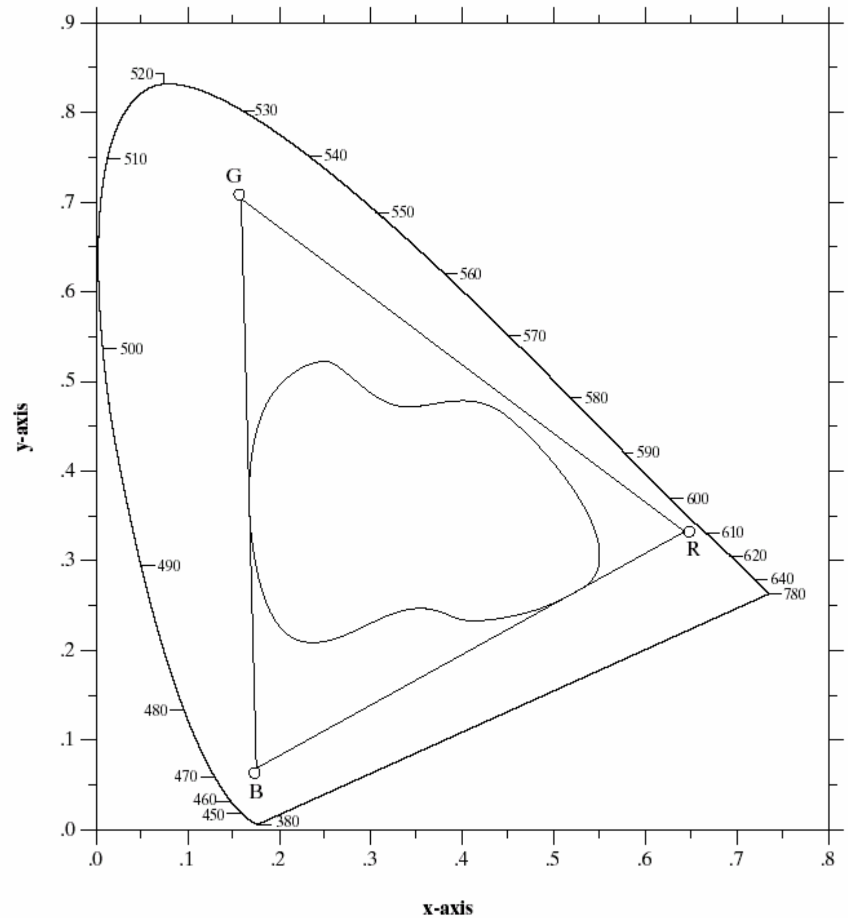


FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).

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

Color models

- RGB model
  - Color monitor, color video cameras
- CMY model
  - Color printer
- HSI model
  - Color image manipulation
- XYZ (CIE standard, Y directly measures the luminance)
- YUV (used in PAL color TV)
- YIQ (used in NTSC color TV)
- YCbCr (used in digital color TV standard BT.601)

RGB model

- Color monitors, color video cameras
- Pixel depth : number of bits used to represent each pixel

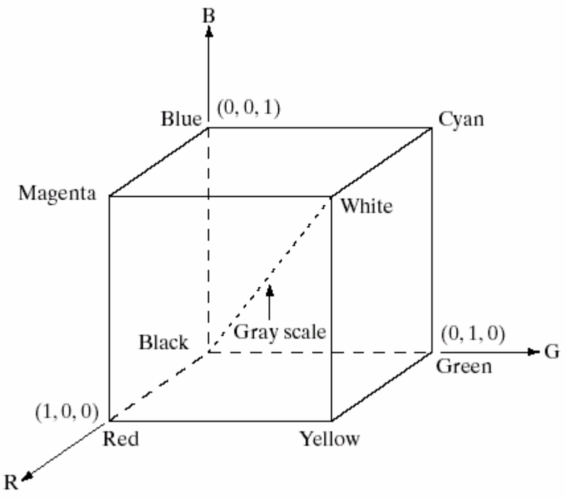


FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

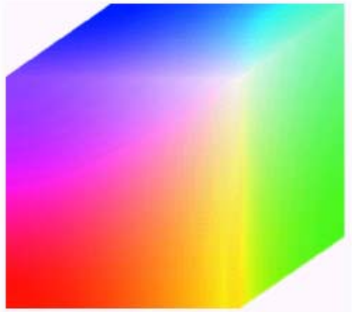
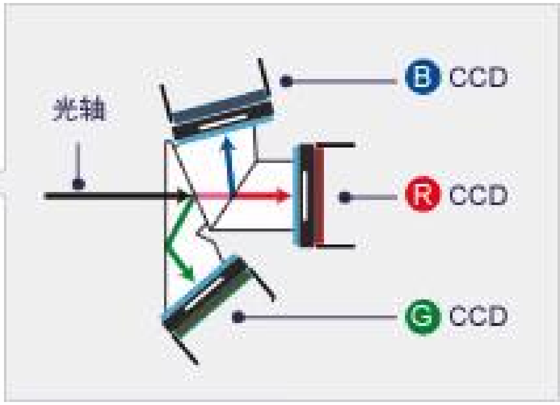


FIGURE 6.8 RGB 24-bit color cube.

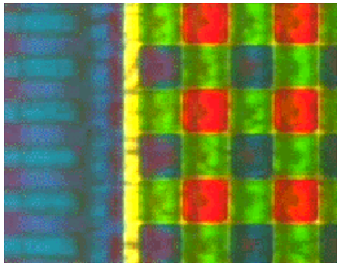
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

RGB model – suitable for imaging and display

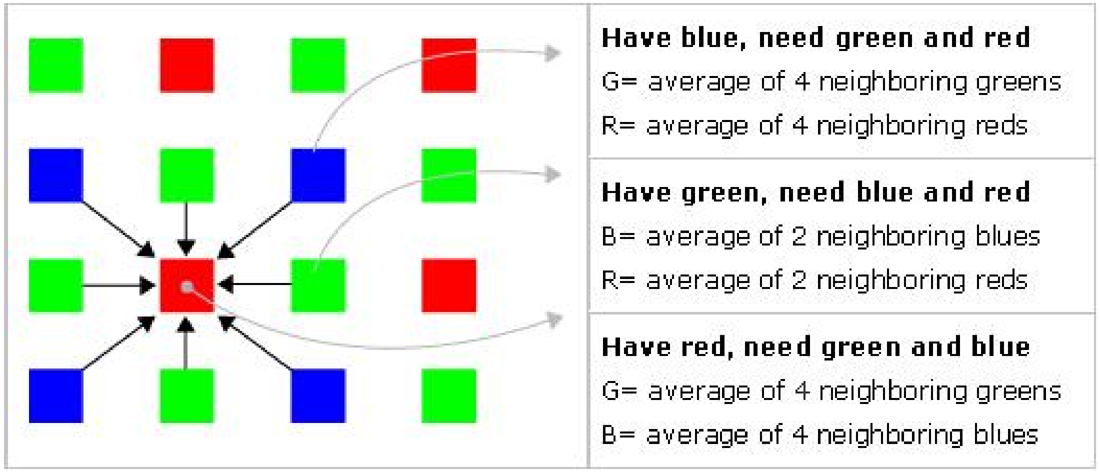
CCD Sensor



CMOS Sensor



HV7131E



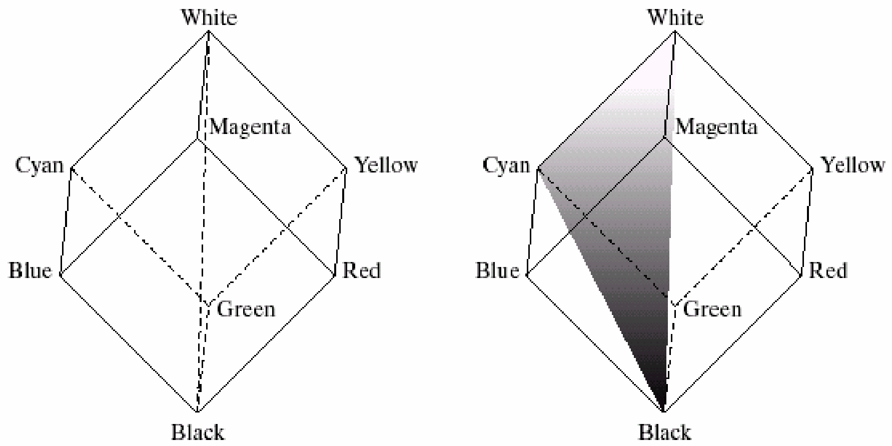
CMY model

- Color printer and copier
- Deposit colored pigment on paper
- Relationship with RGB model

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

HSI model

- The intensity component (I) is decoupled from the color components (H and S), so it is ideal for image processing algorithm development.
- H and S are closely related to the way human visual system perceives colors.

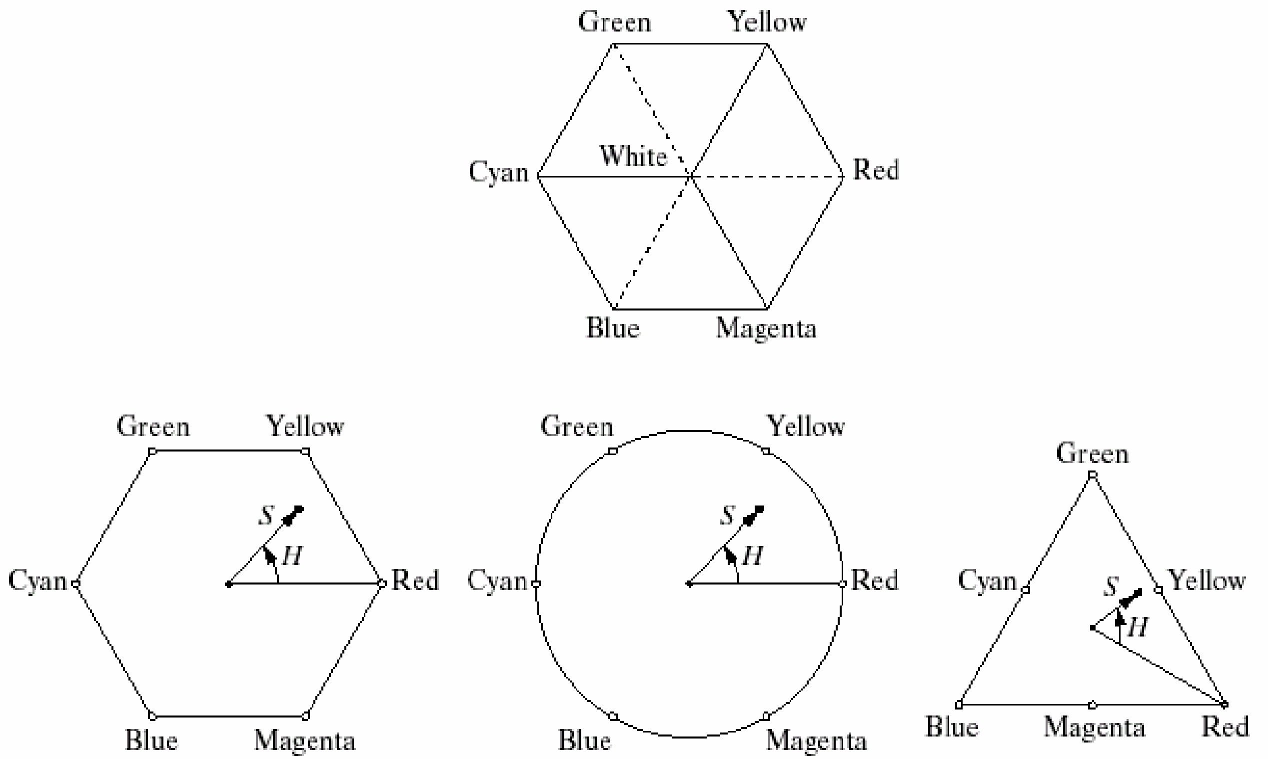


a b

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

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

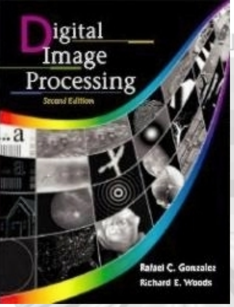
HSI model (cont'd)



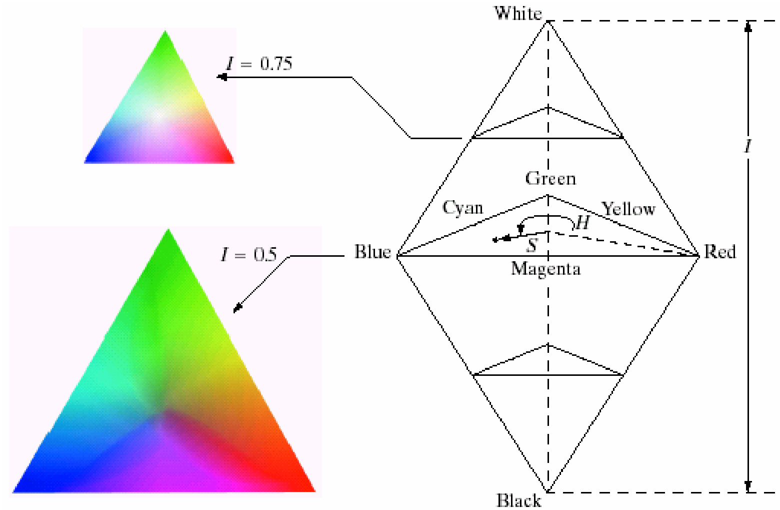
a
b c d

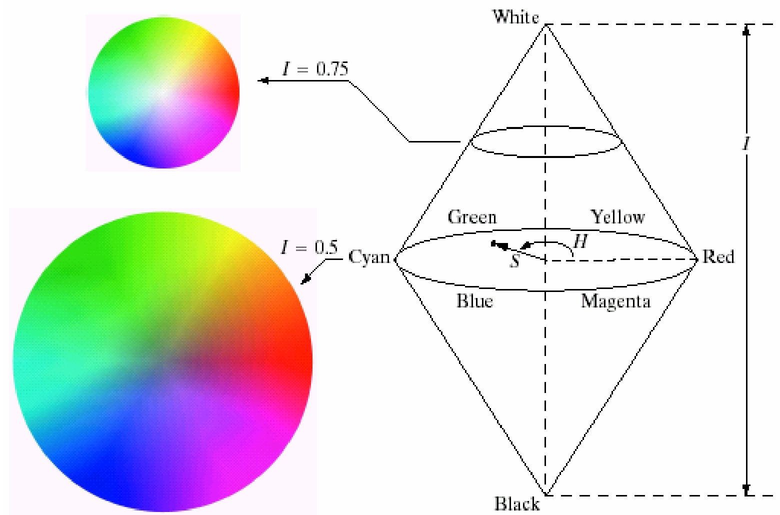
FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

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

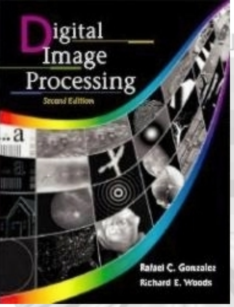


HSI model (cont'd)





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



Model conversion between RGB and HSI

- Converting from RGB to HSI

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

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

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

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

- Converting from HSI to RGB
  - Refer to pp. 299-230

Model conversion between RGB and YCbCr

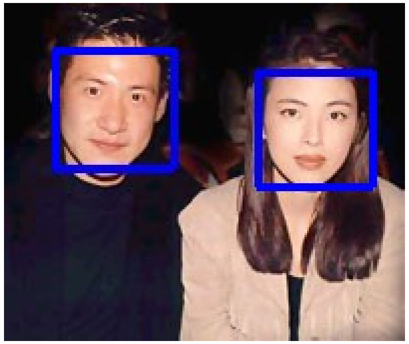
$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

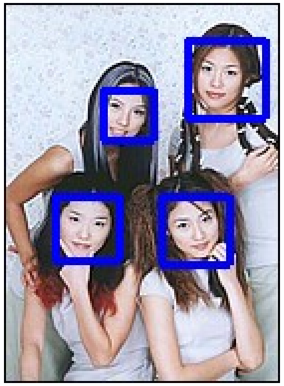
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.598 \\ 1.164 & -0.329 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y-16 \\ Cb-128 \\ Cr-128 \end{bmatrix}$$

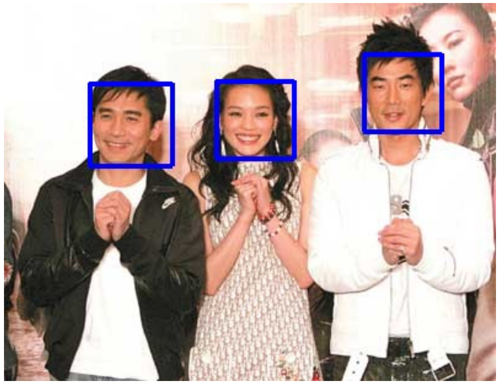
$$Y \in [16,235], Cb, Cr \in [16-240]$$

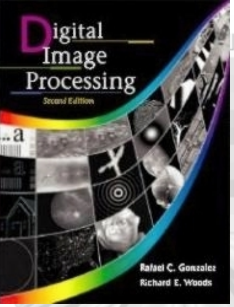
Lab project

- Face detection in color image
- First step: image segmentation based on skin color statistics
  - Select one color space
  - Obtain skin color statistics by using a set of face samples
  - Threshold the image using the skin color statistics





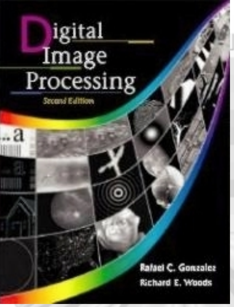




Lab project (cont'd)

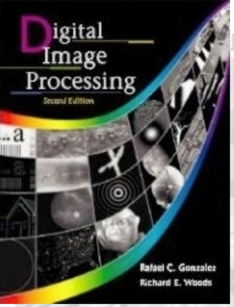
- Color modelling
  - Simple models
    - Model 1: (R,G,B) is classified as skin if $R>95$ and $G>40$ and $B>20$ and $\max\{R,G,B\}-\min\{R,G,B\}>15$ and $|R-G|>15$ and $R>G$ and $R>B$.
    - Model 2: Let $r=R/(R+G+B)$, $g=G/(R+G+B)$, $Y=0.3R+0.59G+0.11B$, (R,G,B) is classified as skin if $0.333<r<0.664$ and $0.246<g<0.398$ and $r>g$ and $g>=0.5-0.5r$. When $Y<40$, (R,G,B) may be hair color.
  - Parametric and non-parametric models

Reference: Vezhnevets V., Sazonov V., Andreeva A., "A Survey on Pixel-Based Skin Color Detection Techniques". Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003



Color Image Processing

- Pseudo-color (false color) image processing
- Full-color image processing



Pseudo-color image processing

- Assign color to monochrome images
- Perform three independent transformations on the gray level of any input pixel
- The three result can then serve as the red, green, and blue component of a color image
- The motivations for using color is that humans can discern thousands of color shades and intensities, compared to only dozen or so shades of gray.

Intensity slicing

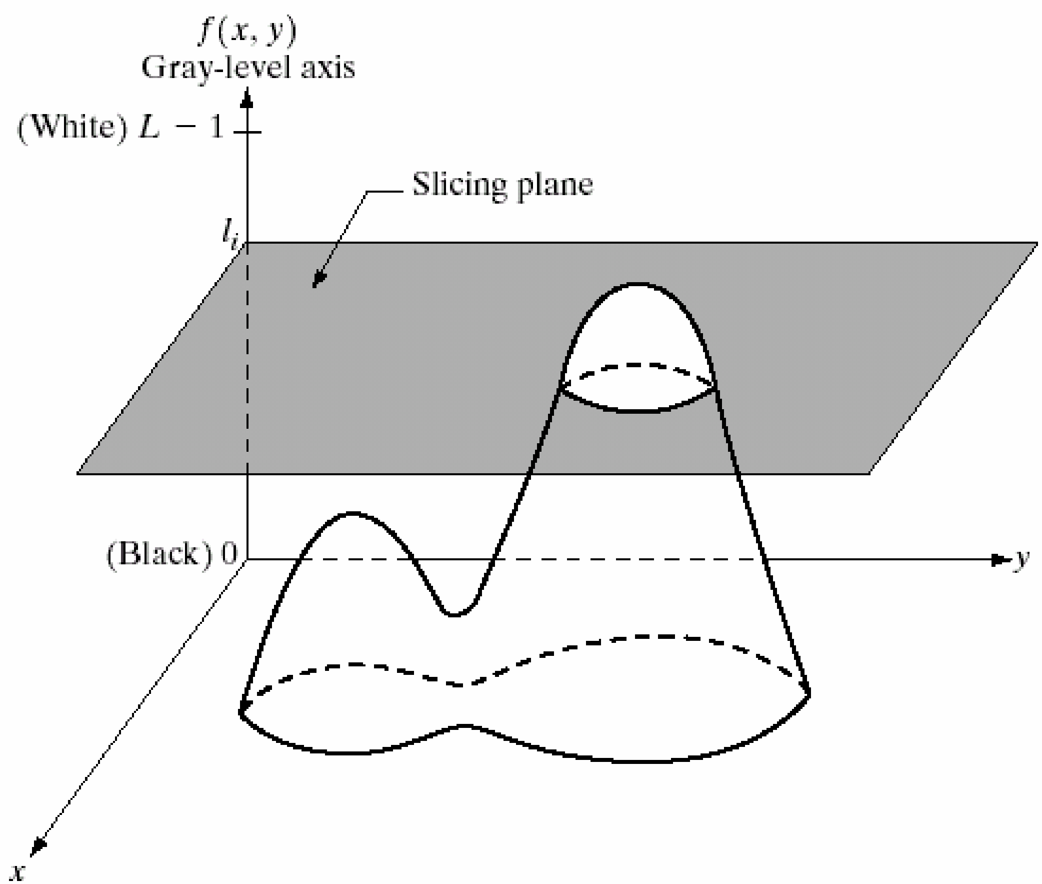
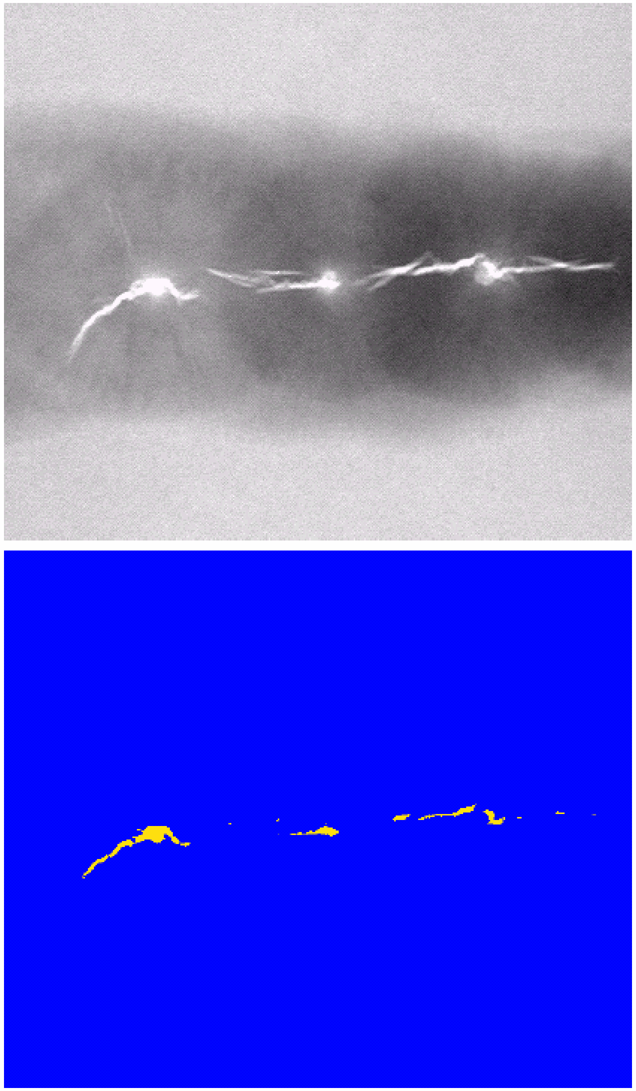


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

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

Intensity slicing (cont'd)

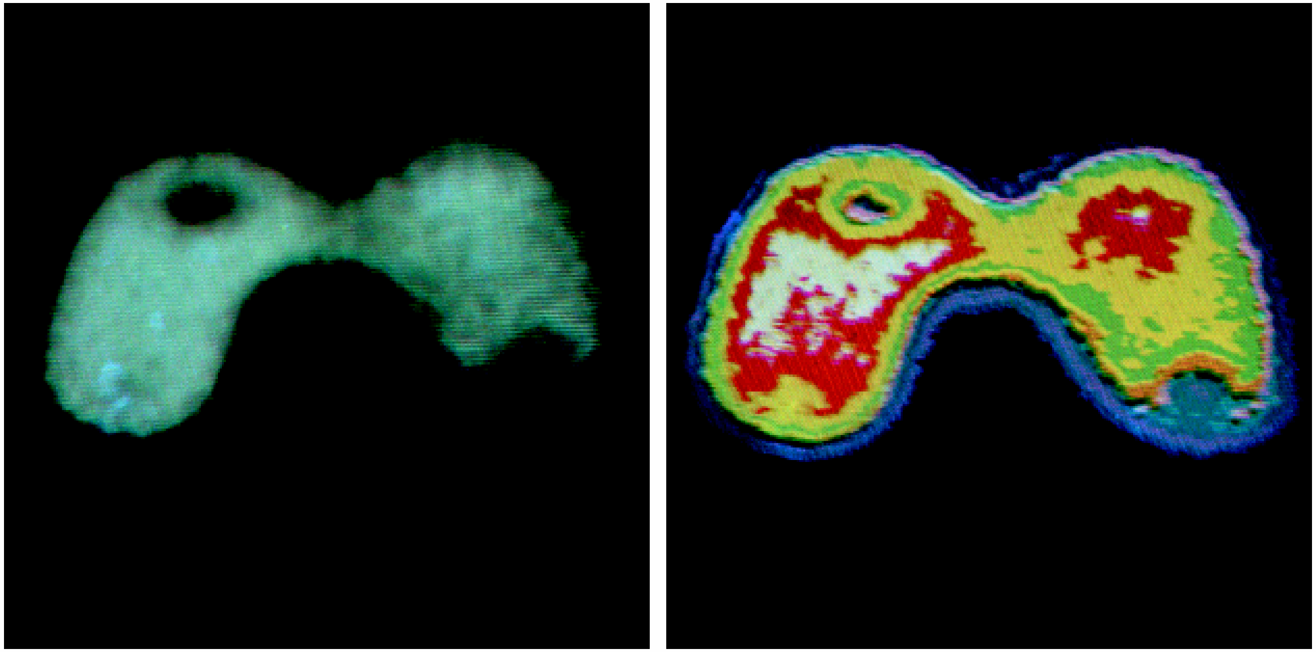


a
b

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

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

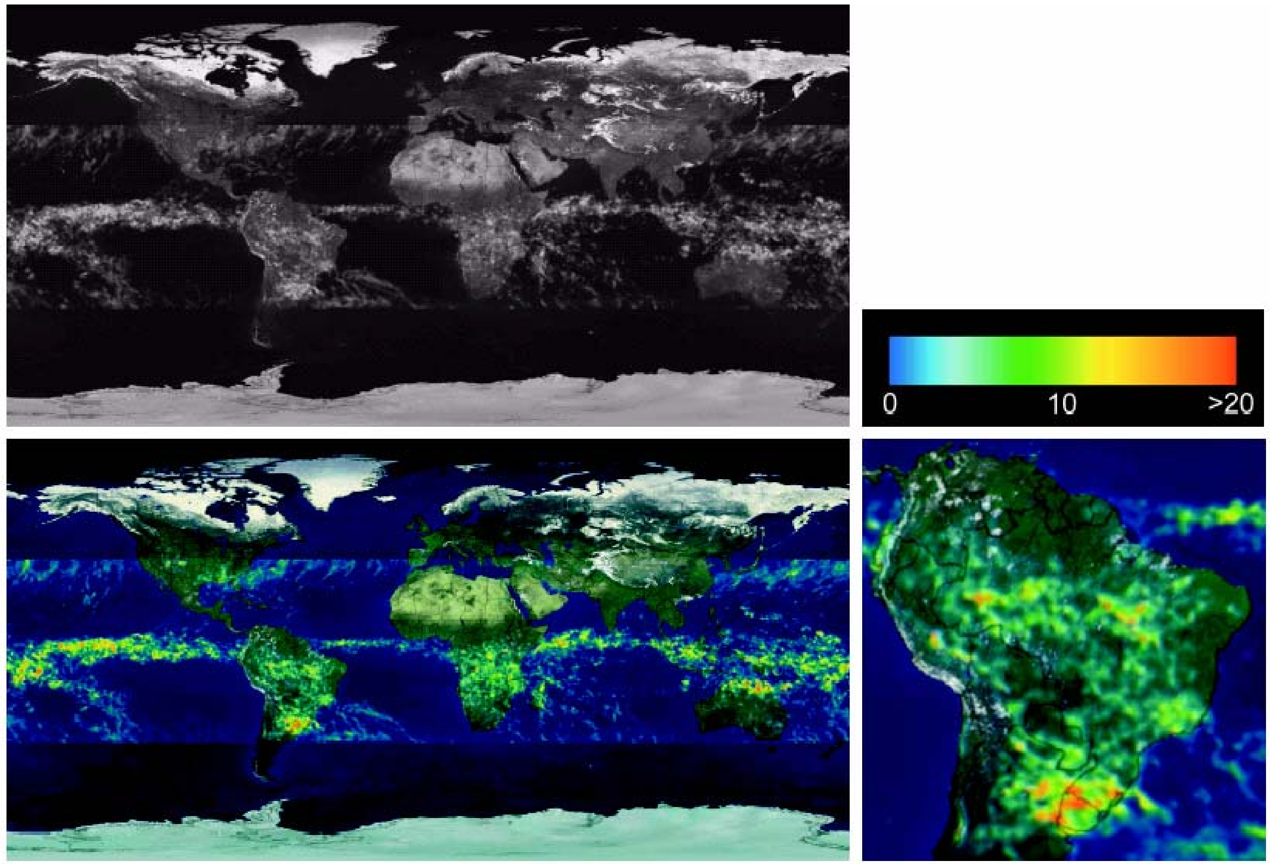
Intensity slicing (cont'd)



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.)

Intensity slicing (cont’d)



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.)

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

Gray level to color transformations

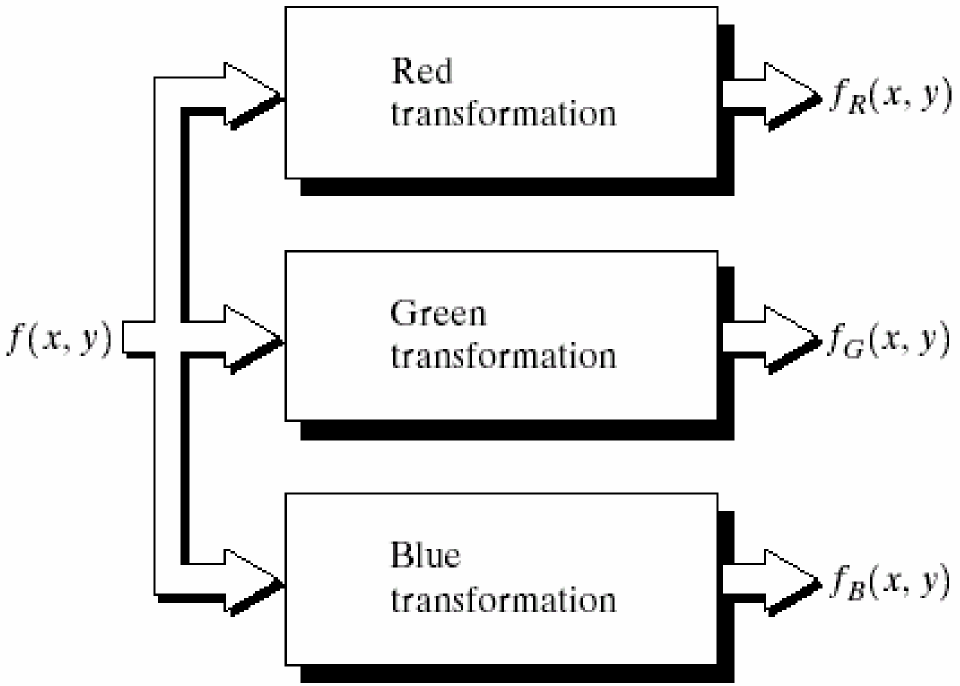


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

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

Gray level to color transformations (cont'd)

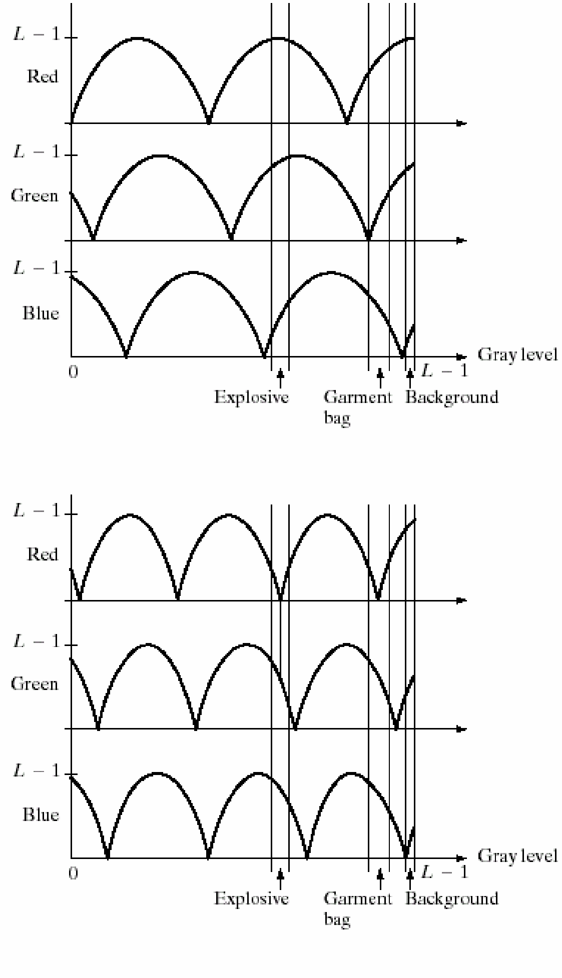


FIGURE 6.25 Transformation functions used to obtain the images in Fig. 6.24.

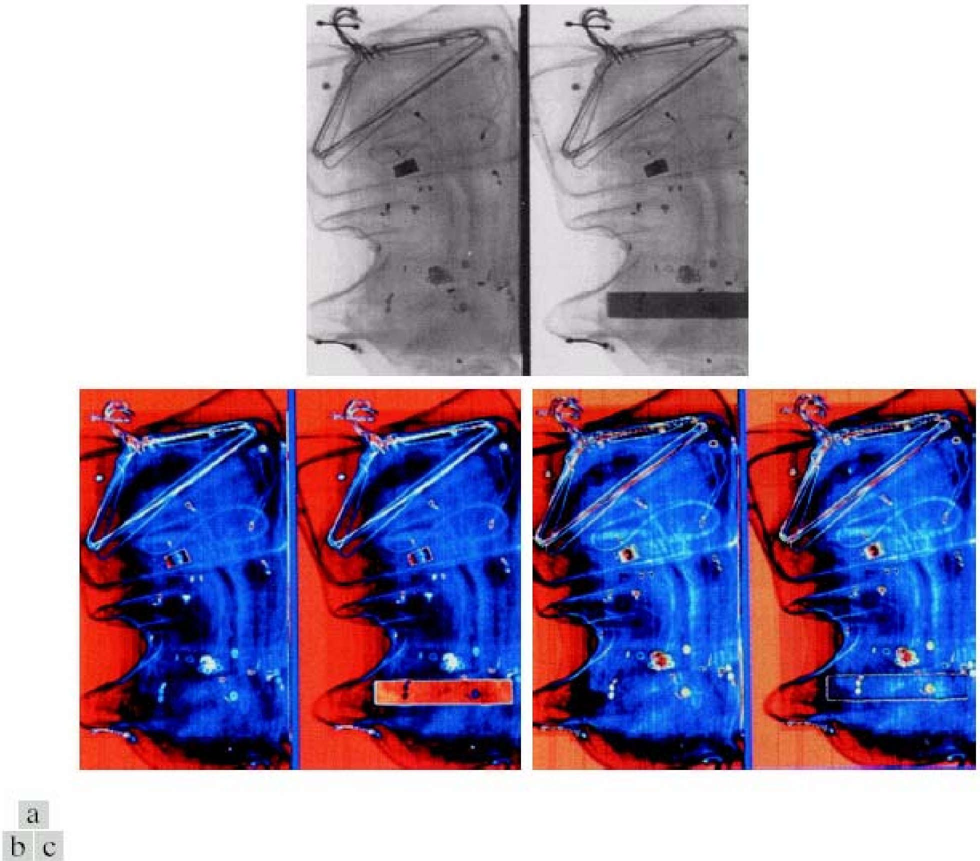
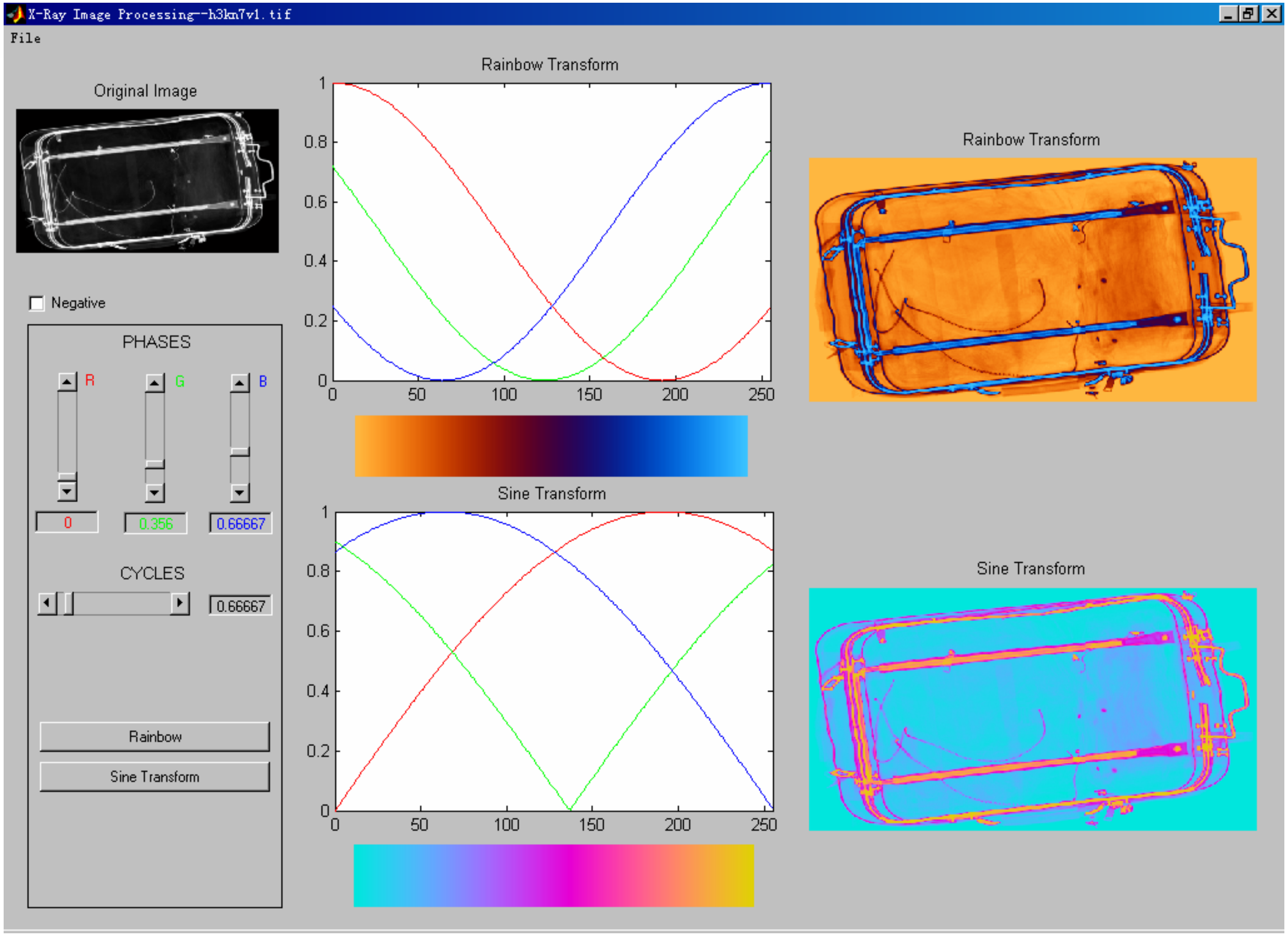


FIGURE 6.24 Pseudocolor enhancement by using the gray-level to color transformations in Fig. 6.25. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)

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

Gray level to color transformations (cont'd)



Multiple monochrome images

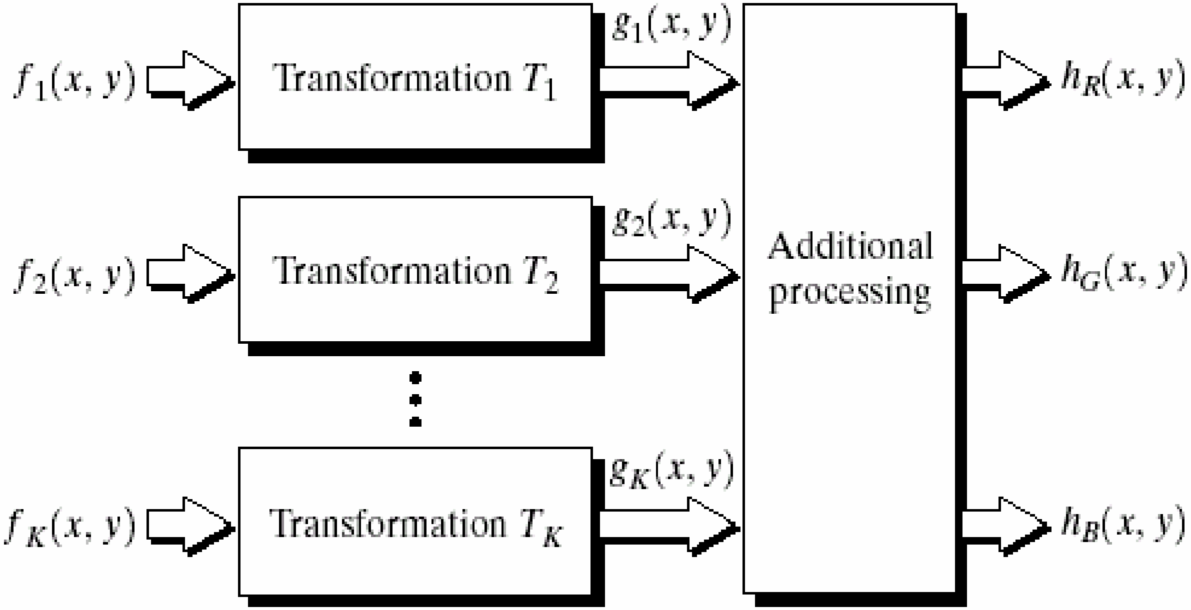
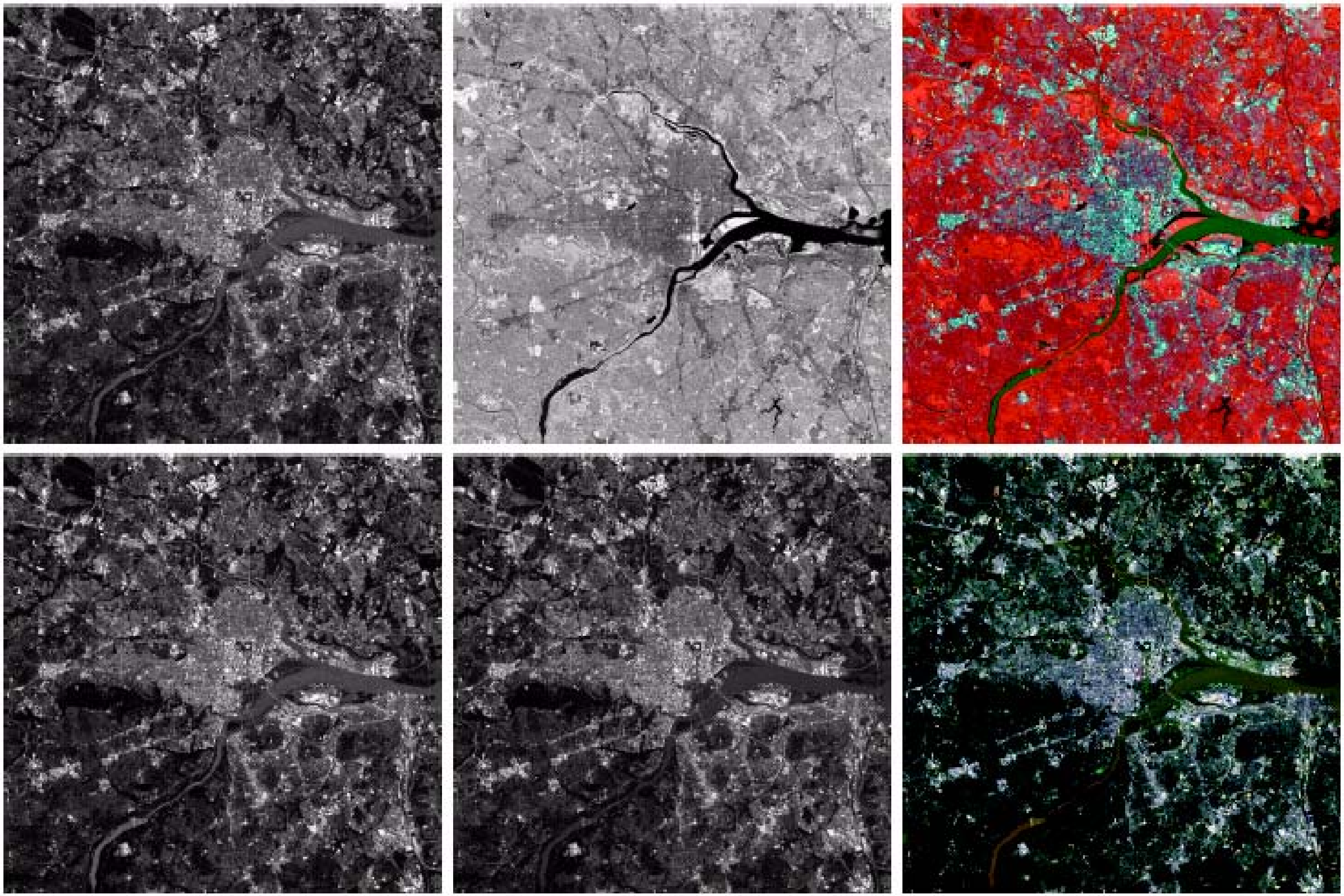


FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

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

Multiple monochrome images (cont'd)



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.)

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

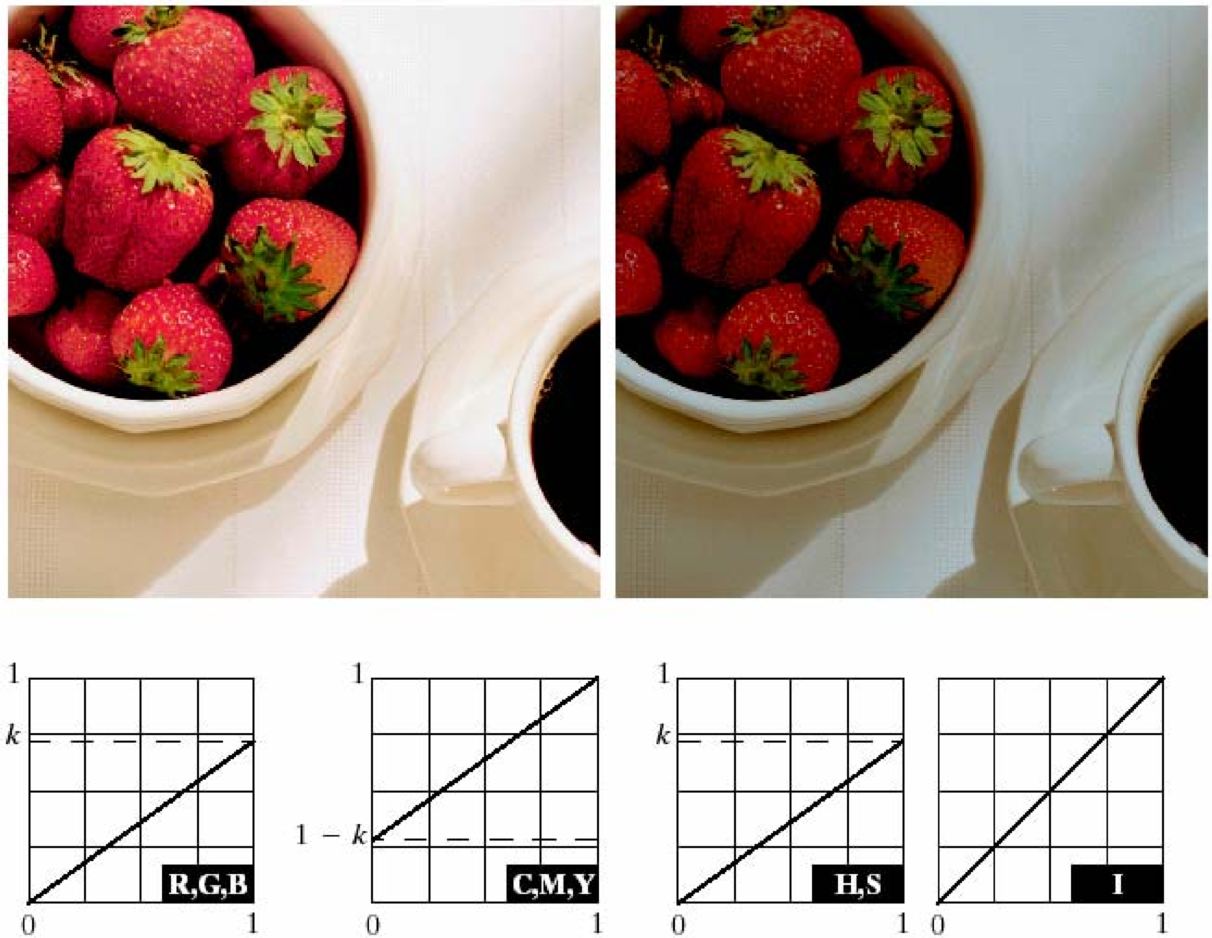
Full-color image processing

- Approach 1: process each component image individually and then form a composite processed color image
- Approach 2: work with color pixels directly
- They are not always equivalent

Color transform

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

Color transform

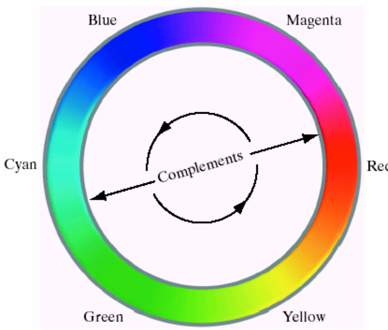


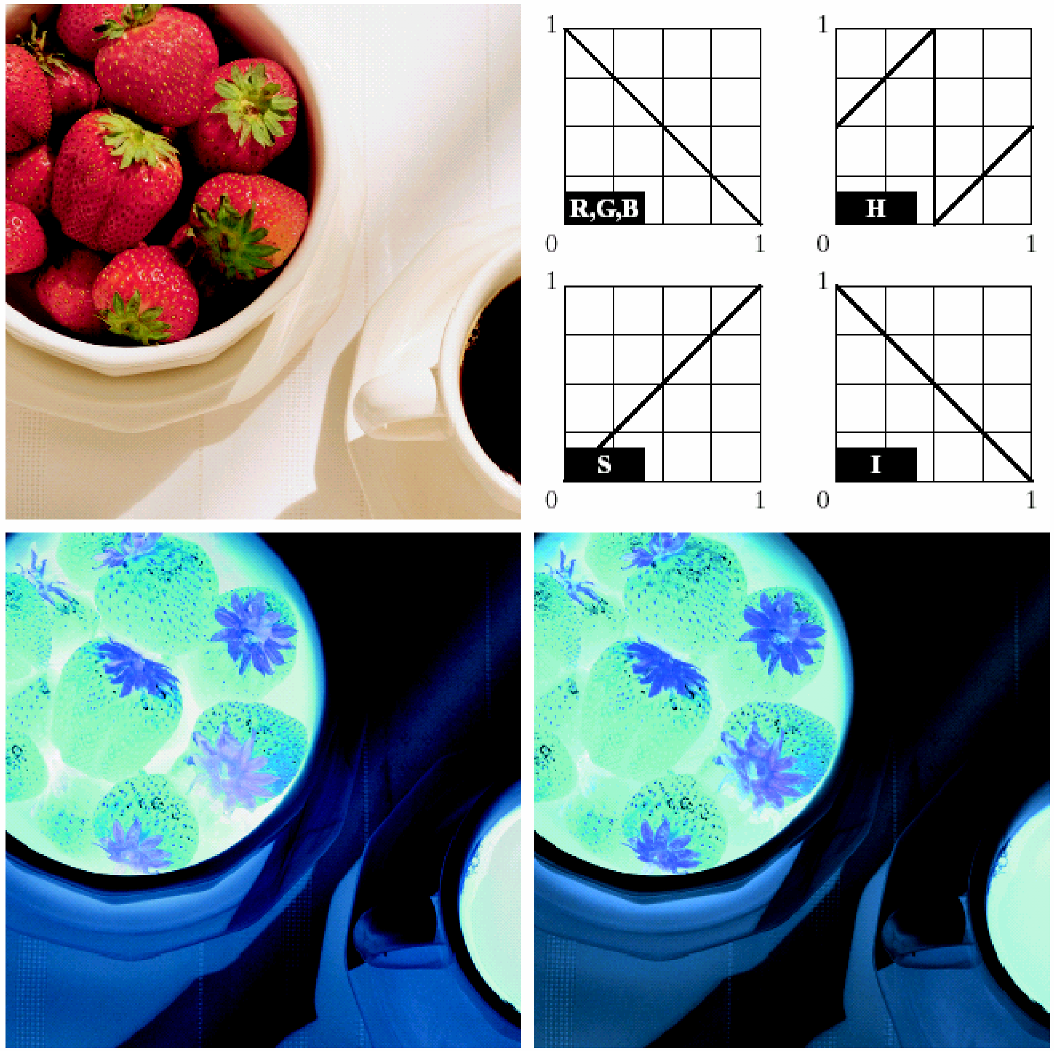
a b
c d e

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.)

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

Color complements

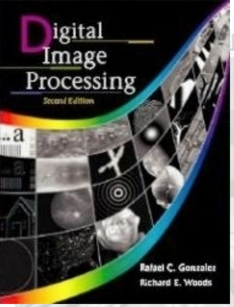




a b
c d

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

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



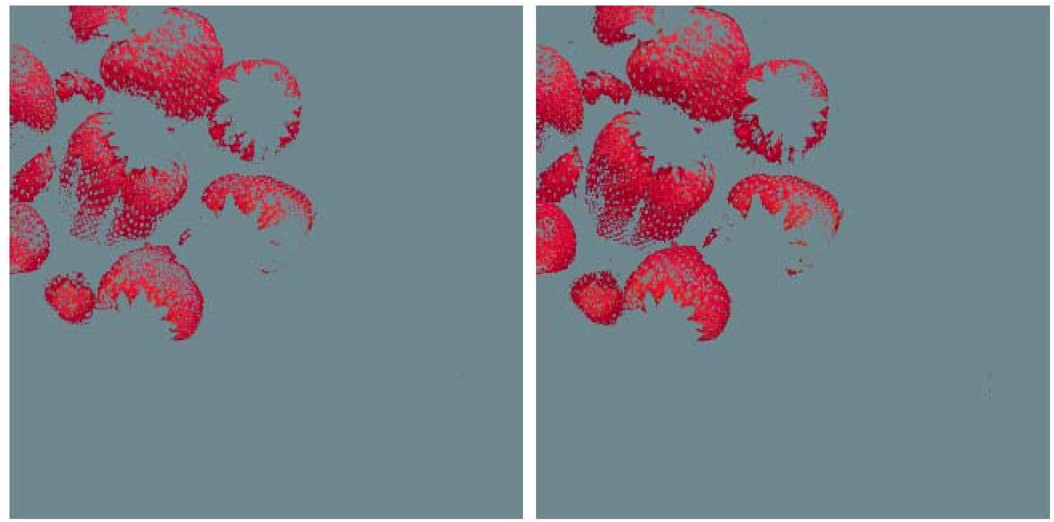
Color slicing

- Slicing using color cube

$$s_i = \begin{cases} 0.5 & if \left| r_j - a_j \right| > \dfrac{W}{2} \\ r_i & else \end{cases}$$

- Slicing using color sphere

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

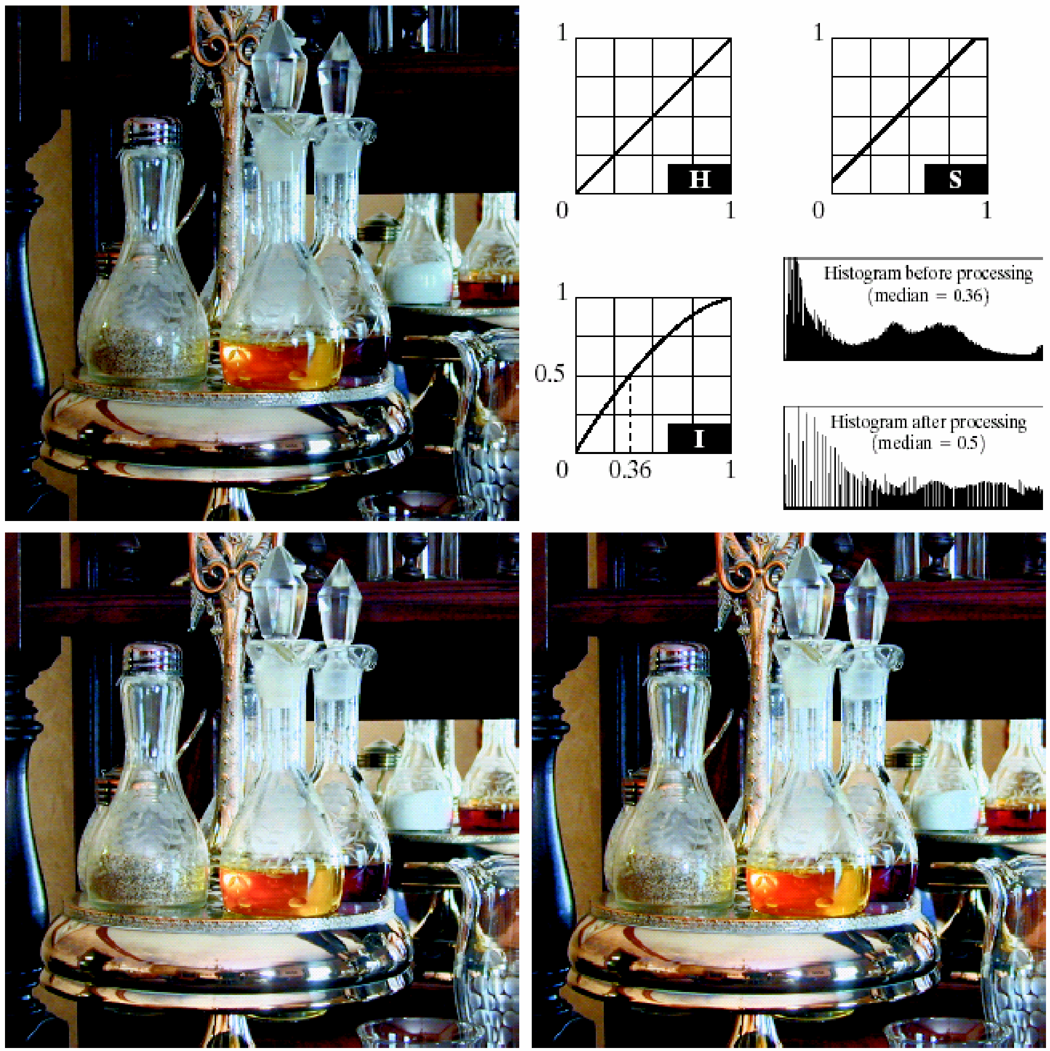


a b

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).

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

Histogram processing

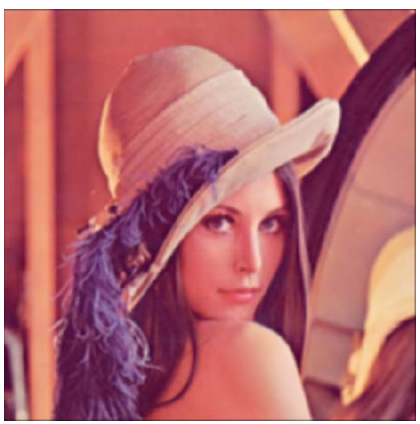


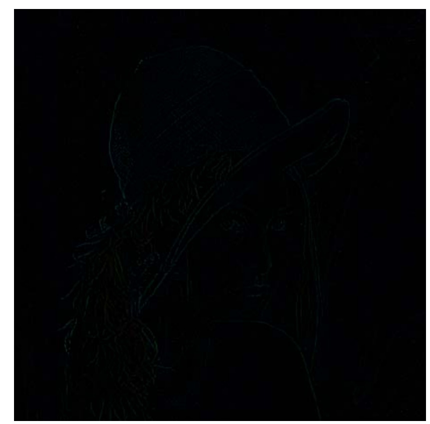
a b
c d

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

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

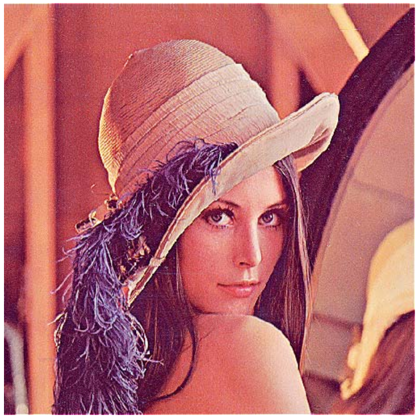
Color image smoothing

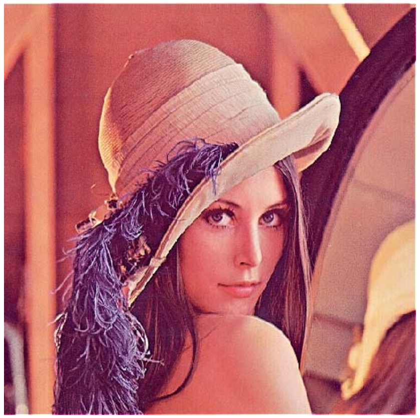




(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Color image sharpening





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

Color Image Processing

Implement in MATLAB

RGB images composition & separation

- Concatenate from the three separated fR, fG and fB component images
  - Function cat

  >> **rgb_image = cat(3, fR, fG, fB)**

- Extract the three component images from an color image

  >> **fR = rgb_image (:, :, 1);**

  >> **fG = rgb_image (:, :, 2);**

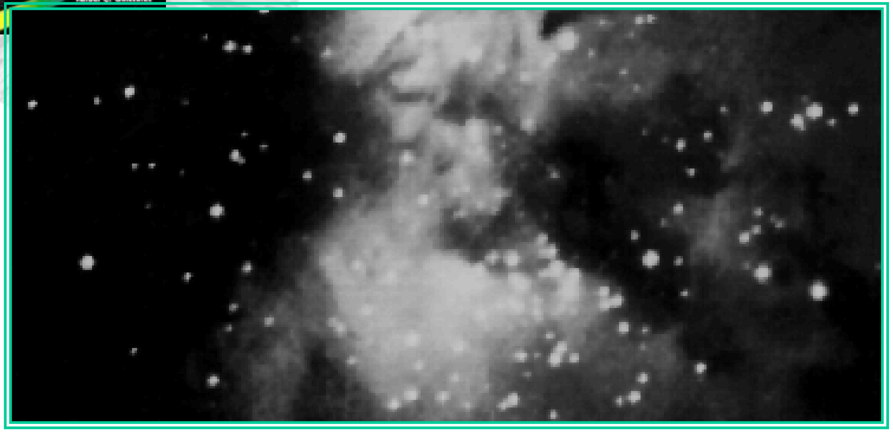
  >> **fB = rgb_image (:, :, 3);**

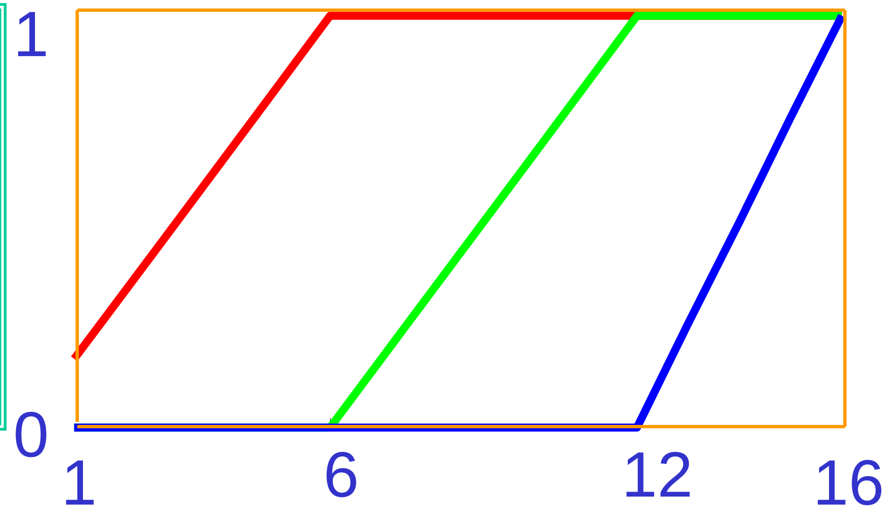
Pseudo-color Processing

1.Gray slicing

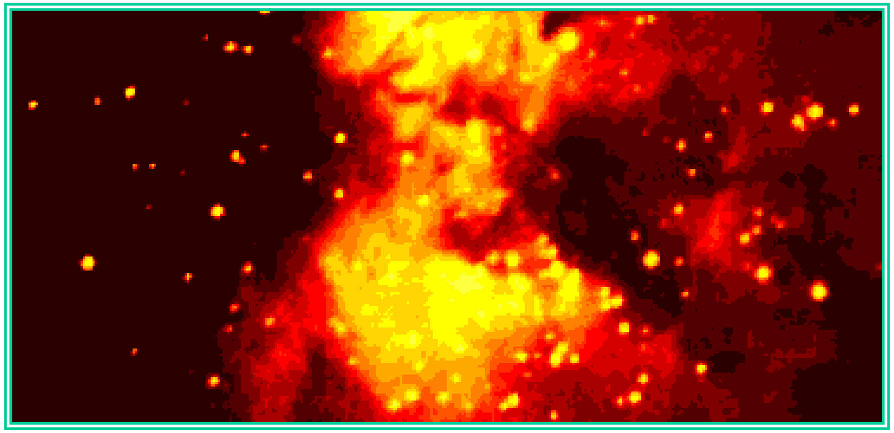
$$f(x,y)=C_i, \quad l_{i-1}\le f(x,y)\le l_i, \quad i=1,2,\cdots,k$$

Gray slicing - example






```
I = imread ('ngc4024m.tif');
X = grayslice (I,16);
imshow ( X, hot (16) )
```



Pseudo-color Processing

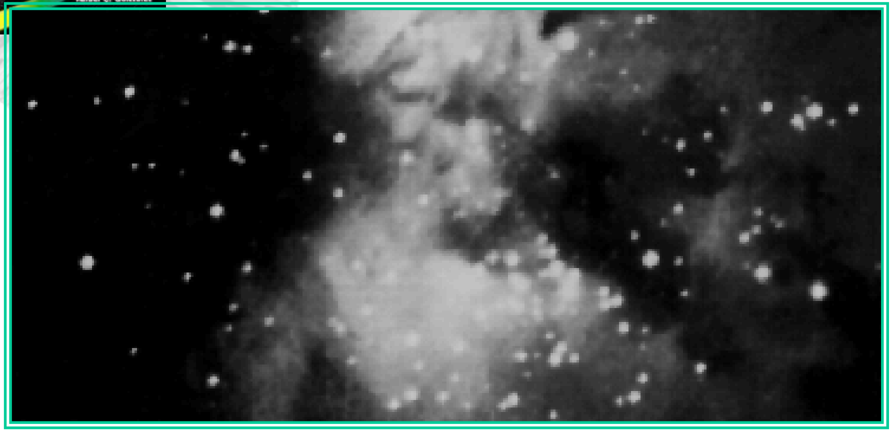
2.Gray level to color transformations

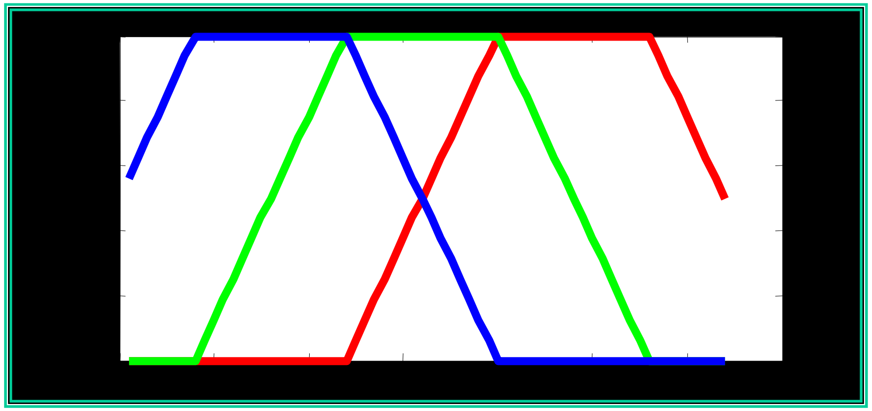
$$R(x,y)=T_R[f(x,y)]$$

$$G(x,y)=T_G[f(x,y)]$$

$$B(x,y)=T_B[f(x,y)]$$

Color transformations - example

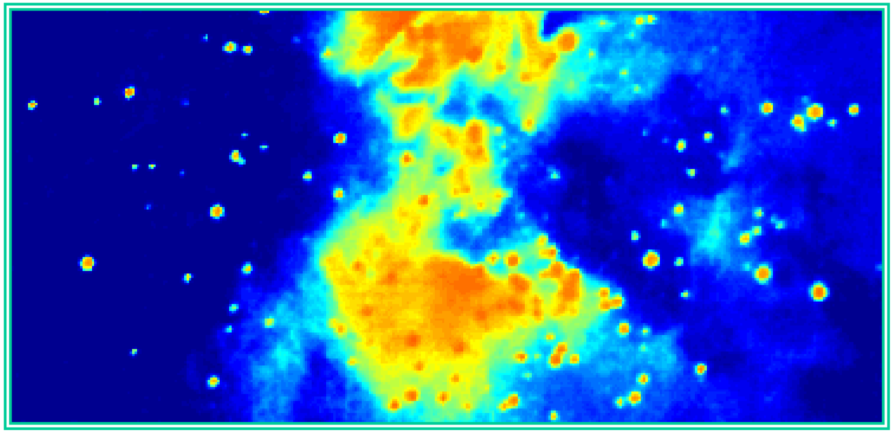




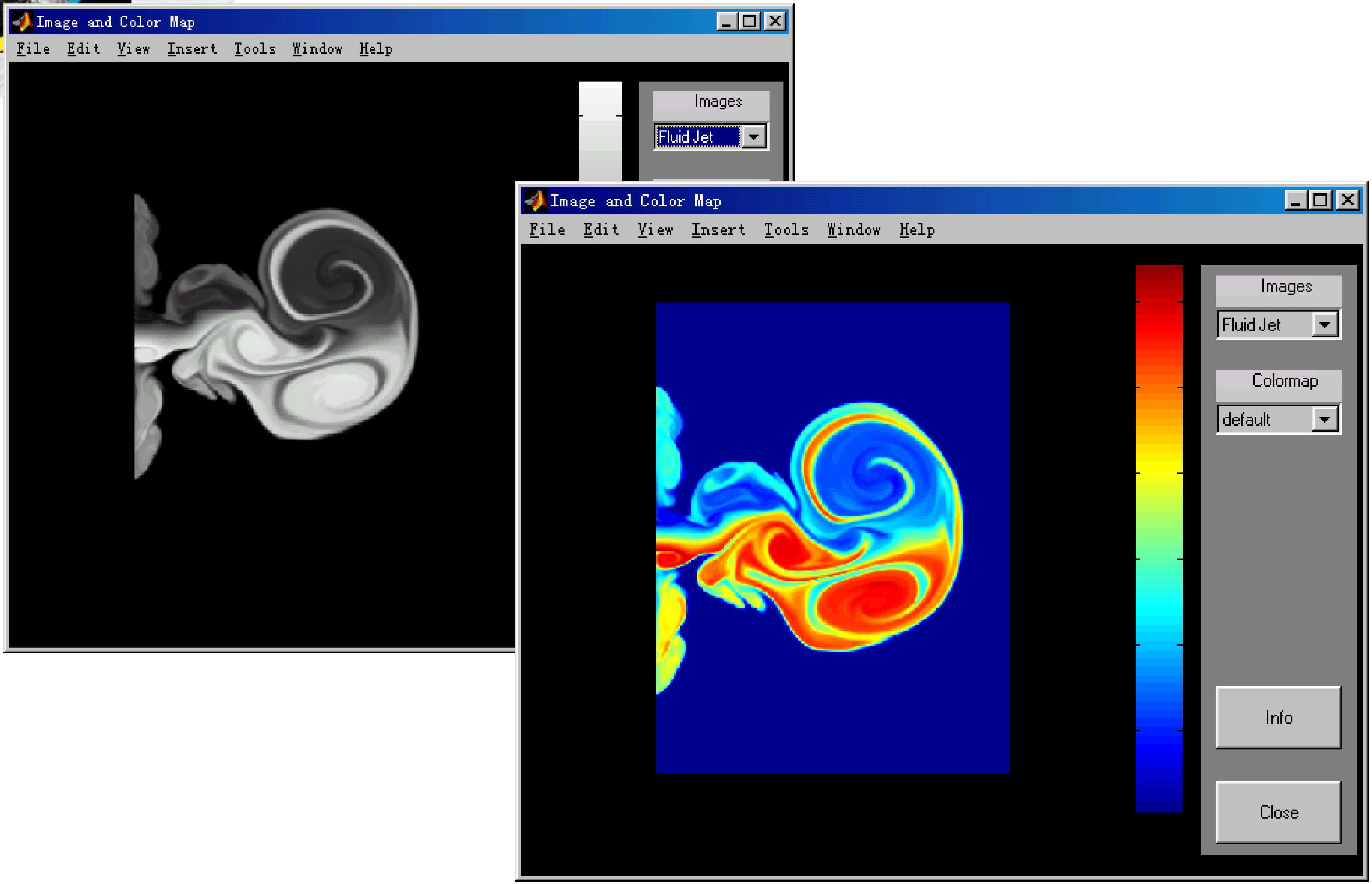
I = imread ('ngc4024m.tif');

imshow (I)

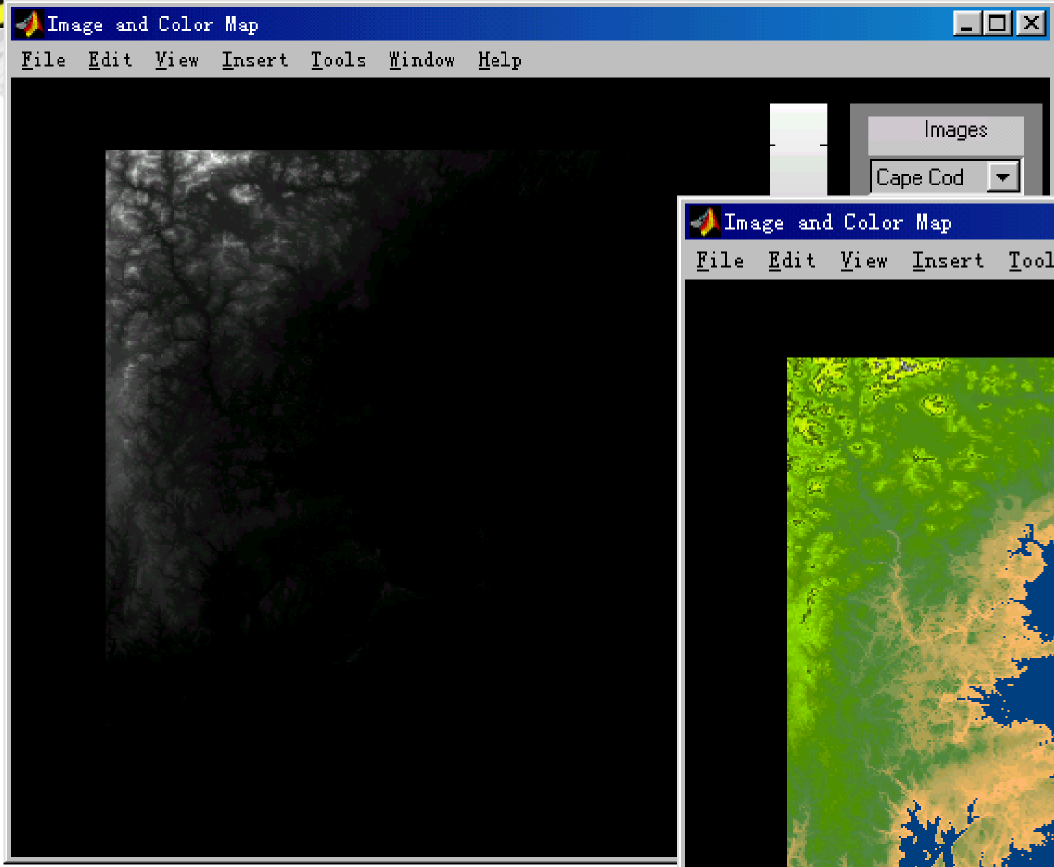
colormap (Jet)

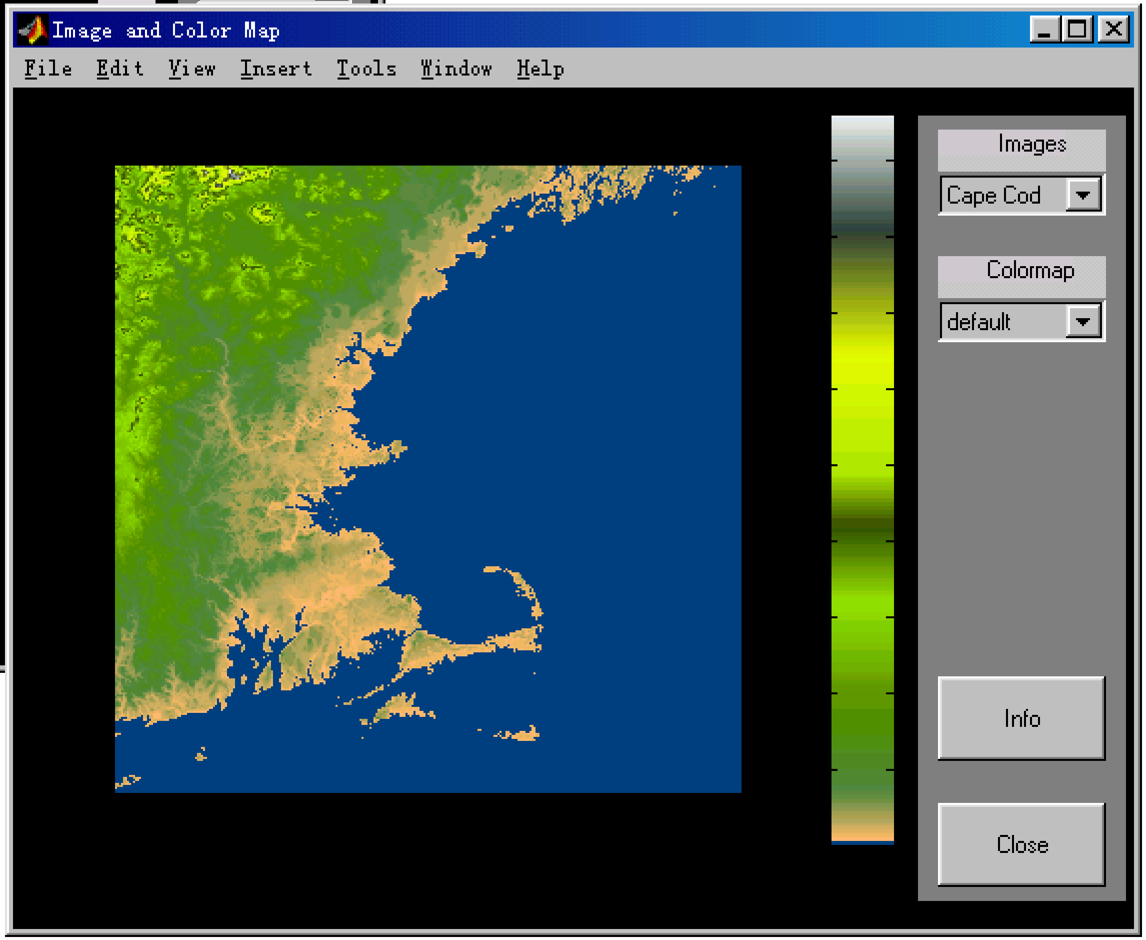


Example – Pseudo-color enhancement



Example – Pseudo-color enhancement





Converting an intensity image using dithering

- Dither an intensity image

```
>> bw = dither ( f );
>> imshow (bw)
```

Convert an color image using dithering

```
>> [X1,map1] = rgb2ind (fc , 8, 'dither');
```

compare with

```
>> [X2,map2] = rgb2ind(fc, 8, 'nodither');
```

Converting to Other Color Spaces

- NTSC Color Space (Television System in US)

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Y luminance
I Hue
Q Saturation

\>> yiq_image = **rgb2ntsc** (rgb_image);

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

\>> rgb_image = **ntsc2rgb** (yiq_image);

Converting to Other Color Spaces

- The YCbCr Color Space (used in digital video)

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 64 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

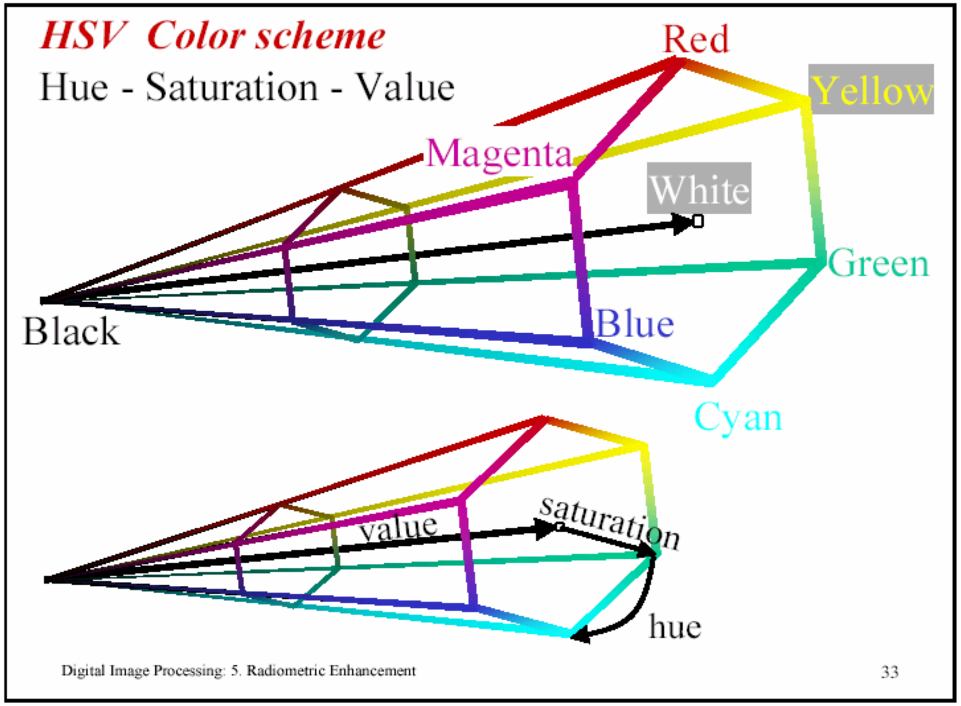
>> ycbcr_image = **rgb2ycbcr** (rgb_image);

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.00456621 & 0 & 0.00626893 \\ 0.00456621 & -0.00153632 & -0.00318811 \\ 0.00456621 & 0 & 0.00791071 \end{bmatrix} \begin{bmatrix} Y - 64 \\ Cb - 128 \\ Cr - 128 \end{bmatrix}$$

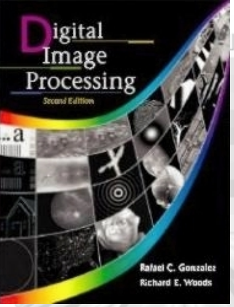
>> rgb_image = **ycbcr2rgb** (ycbcr_image);

Converting to Other Color Spaces

- The HSV Color Space (hue, saturation, value)



>> hsv_image = **rgb2hsv** (rgb_image);
>> rgb_image = **hsv2rgb** (hsv_image);



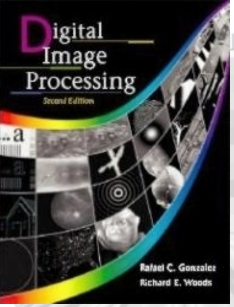
Converting to Other Color Spaces

- The CMY and CMYK Color Spaces

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

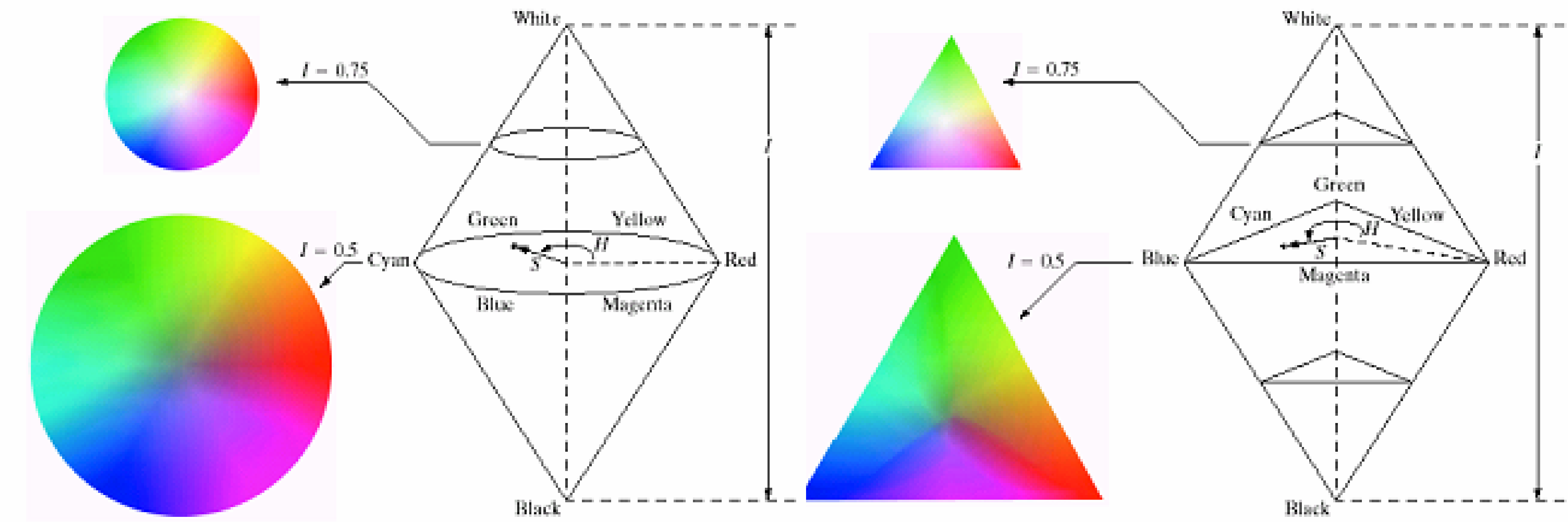
>> cmy_image = **imcomplement** (rgb_image);

>> rgb_image = **imcomplement** (cmy_image);



Converting to Other Color Spaces

- The HSI Color Space
  - The HSV color space is somewhat similar, but its focus is on presenting colors that are meaningful when interpreted in terms of a color artist's palette.



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

Converting to Other Color Spaces

- Converting Colors from RGB to HSI

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

with

$$\theta = \cos^{-1}\left\{ \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right\}$$

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

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

Converting to Other Color Spaces

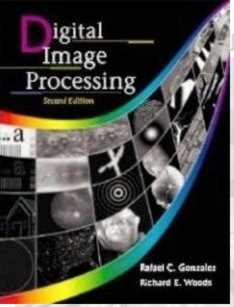
- Converting Colors from HSI to RGB
  - RG sector (0° ≤ H < 120°)

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

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

and

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



Converting to Other Color Spaces

- Converting Colors from HSI to RGB
  - GB sector ($120° \leqslant H < 240°$)
  - If the given value of H is in this sector, subtract 120 ° first

$$H = H - 120°$$

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

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

and

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

Converting to Other Color Spaces

- Converting Colors from HSI to RGB
  - BR sector ($240^\circ \leqslant H \leqslant 360^\circ$)
  - If H is in this range, we subtract 240° from it:

$$H = H - 240^\circ$$

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

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

and

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

Exercise

Prompt: Clip to color range [0, 1], try the statement

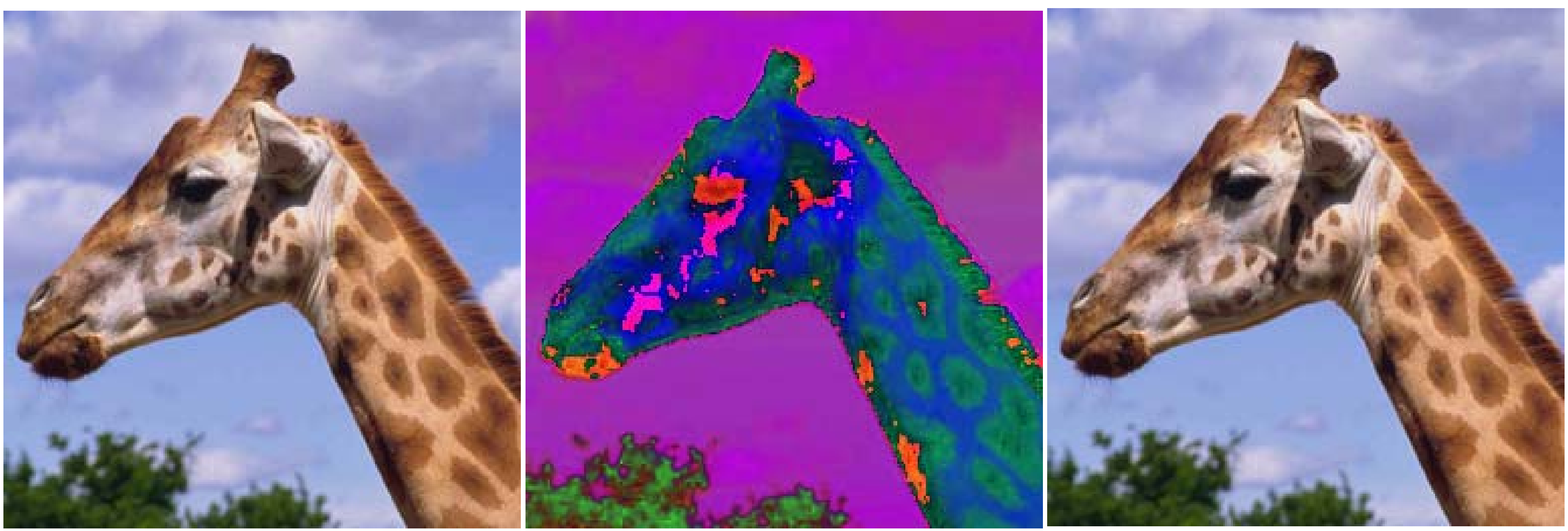
rgb = max(min(rgb, 1), 0);

1. Write an M-function for Converting from RGB to HSI

   hsi_image = rgb2hsi (rgb_image)

2. Write an M-function for Converting from HSI to RGB

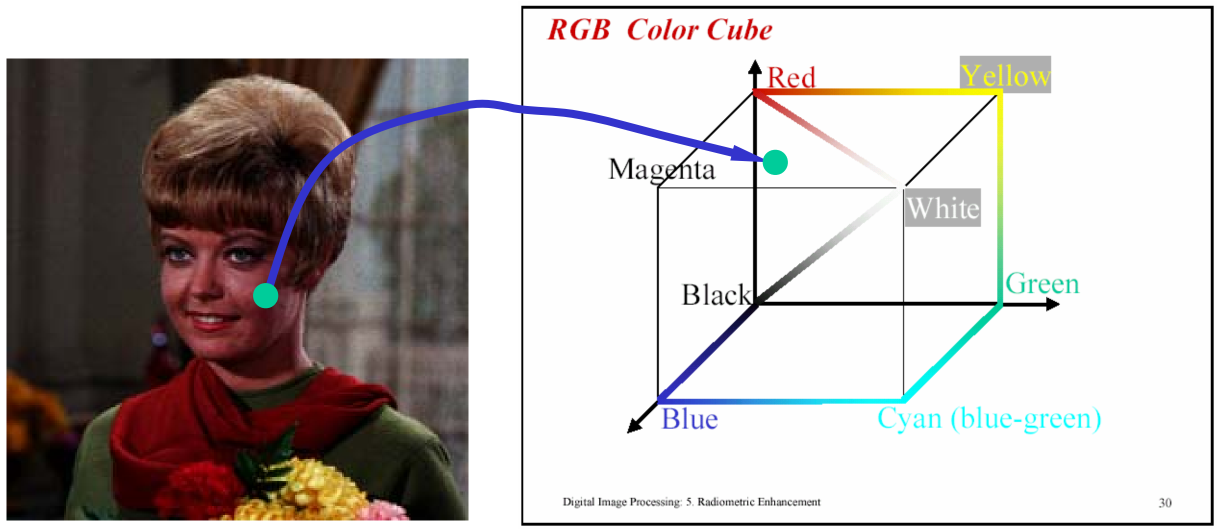
   rgb_image = rgb2hsi (hsi_image)



The Basics of Color Image Processing

- Each color point can be interpreted as a vector extended from the origin to that point in the RGB coordinate system
- M-by-N Vectors in a color image

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



Spatial Filtering of Color Images

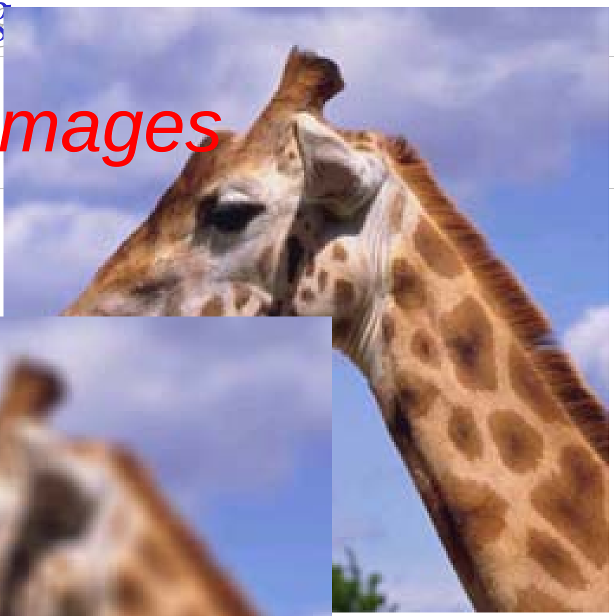
$$\bar{c}(x,y)=\begin{bmatrix} \frac{1}{K}\sum_{(s,t)\in S_{xy}} R(s,t) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} G(s,t) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} B(s,t) \end{bmatrix}$$

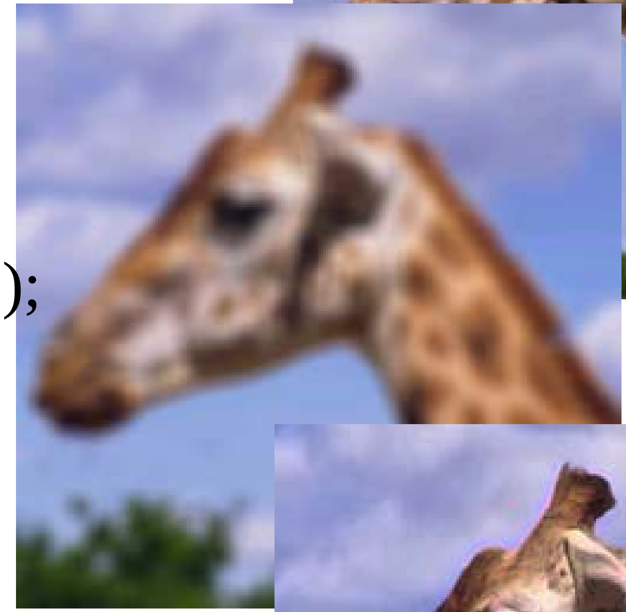
where K is the number of pixels in the neighborhood

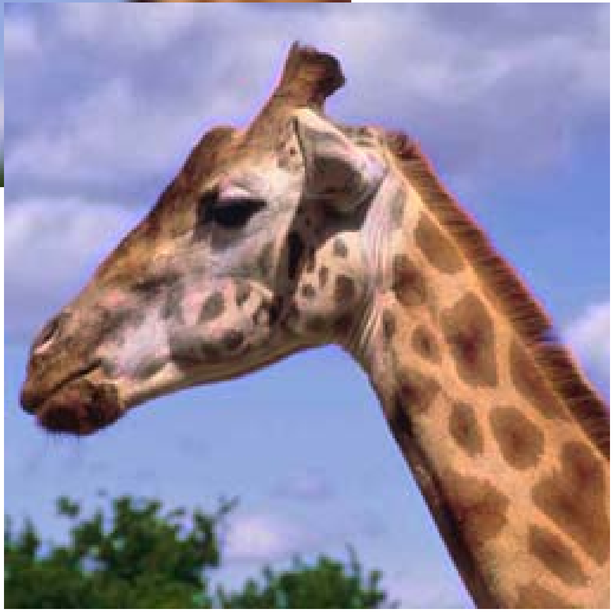
Spatial Filtering of Color Images

- Color Image Smoothing

```
>> w = fspecial('average', 15);
>> filtered = imfilter(fc, w, 'replicate');
    or
>> h = rgb2hsi(fc);
>> H = h(:,:,1);S = h(:,:,2);I = h(:,:,3);
>>S1 =imfilter(S, w, 'replicate');
>> h = cat(3, H, S1, I);
>> f = hsi2rgb (h);
```







Color Image Sharpening

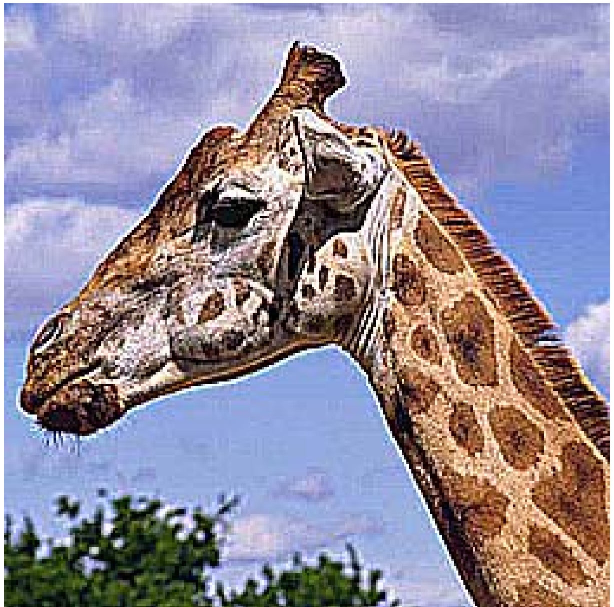
$$\nabla^2[c(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}$$

```
>> lapmask = [1 1 1;1 –8 1;1 1 1];
>> fen = imsubtract (fc, imfilter(fc, lapmask, 'replicate'));
>> imshow(fen)
```

TrueColor CData contains element out of range 0.0 <= value <= 1.0.

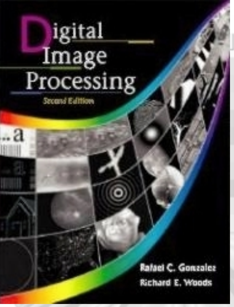
Display the Result of Sharpening

```
>> ma = max(fen(:));
>> mi = min(fen(:));
>> fen1 = (fen-mi)/(ma-mi);
>> imshow(fen1)

>> fen2 = max(min(fen,1),0);

>> imshow(fen2)
```



Working Directly in RGB Vector Space

- Two important applications in color image processing:
  - **Color edge detection**
  - **Region segmentation**



Color Edge Detection Using the Gradient

Review

Gradient Definition

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of this vector is

$$\nabla f = mag(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2 \right]^{\frac{1}{2}}$$

$$= \left[(\partial f / \partial x)^2 + (\partial f / \partial y)^2 \right]^{\frac{1}{2}}$$

Color Edge Detection Using the Gradient

Often, the magnitude of gradient vector is approximated as

$$\nabla f \approx |G_x| + |G_y|$$

The angle of gradient vector is

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Color Edge Detection Using the Gradient

Compute the derivatives approximately

$$G_x = (\mathbf{z}_7+2\mathbf{z}_8+\mathbf{z}_9) - (\mathbf{z}_2+2\mathbf{z}_2+\mathbf{z}_3)$$

$$G_y = (\mathbf{z}_3+2\mathbf{z}_6+\mathbf{z}_9) - (\mathbf{z}_1+2\mathbf{z}_4+\mathbf{z}_7)$$

Small neighborhood

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel mask (vertical)

-1	-2	-1
0	0	0
1	2	1

Sobel mask (horizontal)

-1	0	1
-2	0	2
-1	0	1

Color Edge Detection Using the Gradient

To compute the gradient in RGB color space, Extend the concept of a gradient to vector functions.

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

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

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

Color Edge Detection Using the Gradient

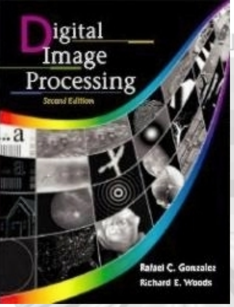
Let the quantities g_{xx}, g_{yy}, and g_{xy} be defined in terms of the dot product of these vectors, as follows

$$g_{xx} = \mathbf{u} \cdot \mathbf{u} = \mathbf{u}^T \mathbf{u} = \left|\frac{\partial R}{\partial x}\right|^2 + \left|\frac{\partial G}{\partial x}\right|^2 + \left|\frac{\partial B}{\partial x}\right|^2$$

$$g_{yy} = \mathbf{v} \cdot \mathbf{v} = \mathbf{v}^T \mathbf{v} = \left|\frac{\partial R}{\partial y}\right|^2 + \left|\frac{\partial G}{\partial y}\right|^2 + \left|\frac{\partial B}{\partial y}\right|^2$$

and

$$g_{xy} = \mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}$$



Color Edge Detection Using the Gradient

The direction of maximum rate of change of c(x,y) as a function (x, y) is given by the angle

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

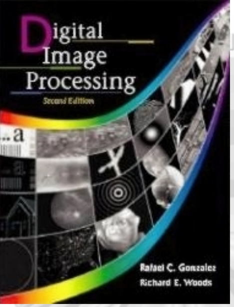
The value of the rate of change in the directions given by the elements $\theta\,(x, y)$ of is given by

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

Color Edge Detection Using the Gradient

To complete the computation $\frac{\partial R}{\partial x}, \frac{\partial G}{\partial x}, \frac{\partial B}{\partial x}, \frac{\partial R}{\partial y}, \frac{\partial G}{\partial y}, \frac{\partial B}{\partial y}$, write the statements in MATLAB

```
>> sh = fspecial('sobel');
>> sv = sh';
>> Rx = imfilter(double(f(:, :, 1)), sh, 'replicate');
>> Ry = imfilter(double(f(:, :, 1)), sv, 'replicate');
>> Gx = imfilter(double(f(:, :, 2)), sh, 'replicate');
>> Gy = imfilter(double(f(:, :, 2)), sv, 'replicate');
>> Bx = imfilter(double(f(:, :, 3)), sh, 'replicate');
>> By = imfilter(double(f(:, :, 3)), sv, 'replicate');
```



Color Edge Detection Using the Gradient

$$g_{xx} = \mathbf{u} \cdot \mathbf{u} = \mathbf{u}^T \mathbf{u} = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2$$

```
>> gxx = Rx.^2 + Gx.^2 + Bx.^2;
```

$$g_{yy} = \mathbf{v} \cdot \mathbf{v} = \mathbf{v}^T \mathbf{v} = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2$$

```
>> gyy = Ry.^2 + Gy.^2 + By.^2;
```

$$g_{xy} = \mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}$$

```
>> gxy = Rx.*Ry + Gx.*Gy + Bx.*By;
```

Color Edge Detection Using the Gradient

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

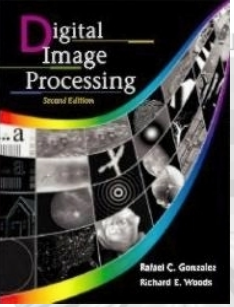
```
>> A = 0.5*(atan(2*gxy./(gxx - gyy + eps)));
```

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

```
>> G1 = 0.5*((gxx + gyy) + (gxx - gyy).*cos(2*A) + 2*gxy.*sin(2*A));
```

The function implements the color gradient for RGB images was listed in Appendix

Color Edge Detection Using the Gradient

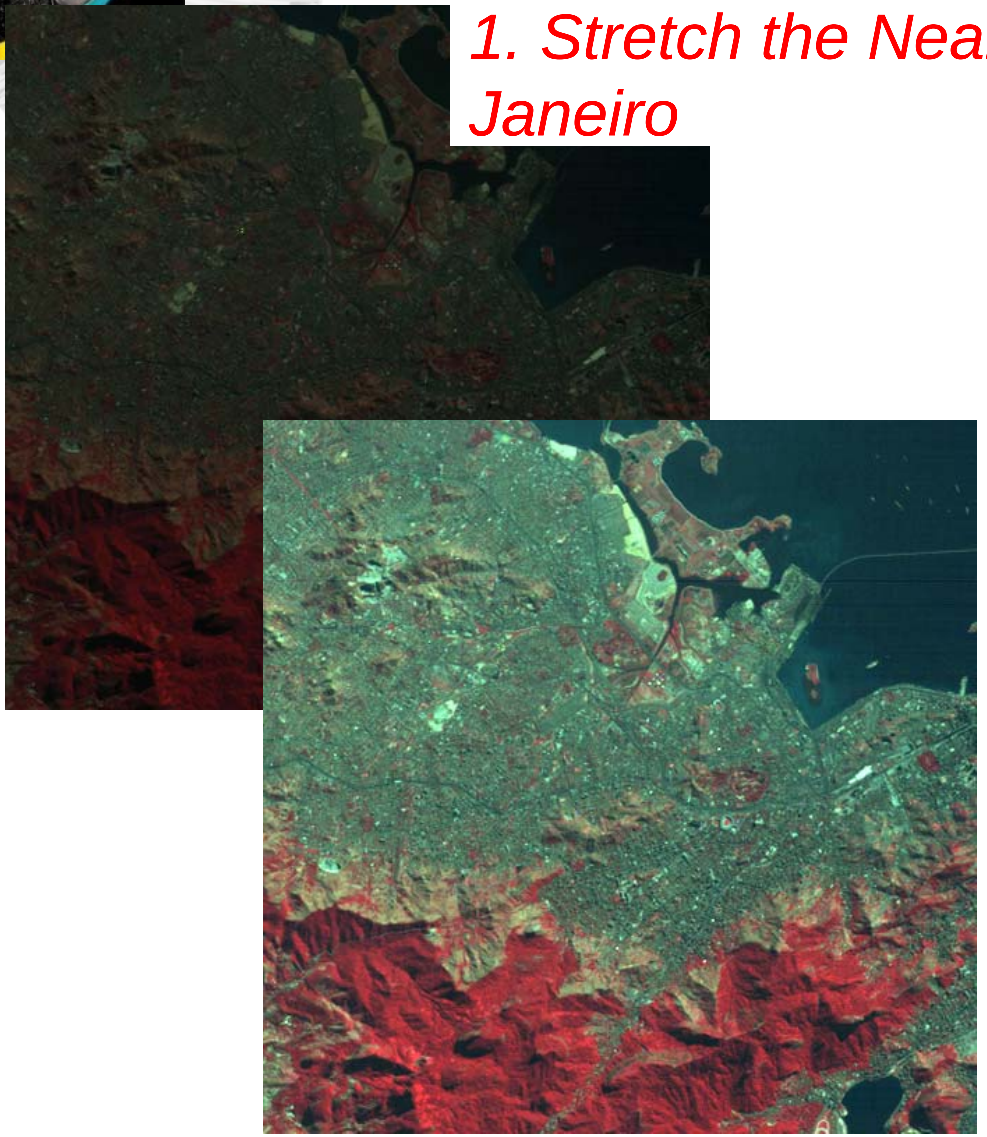


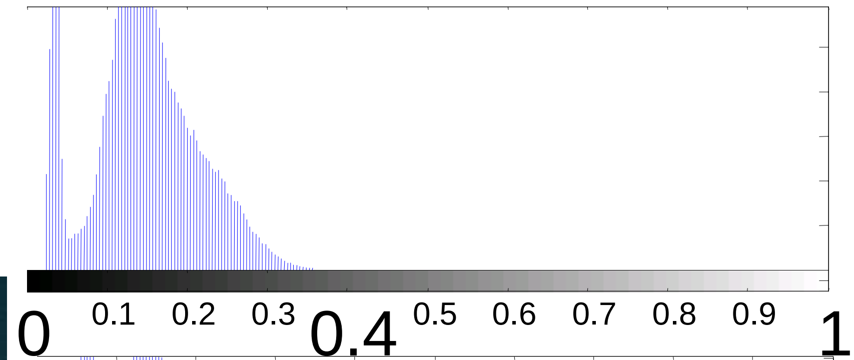
Working Directly in RGB Vector Space

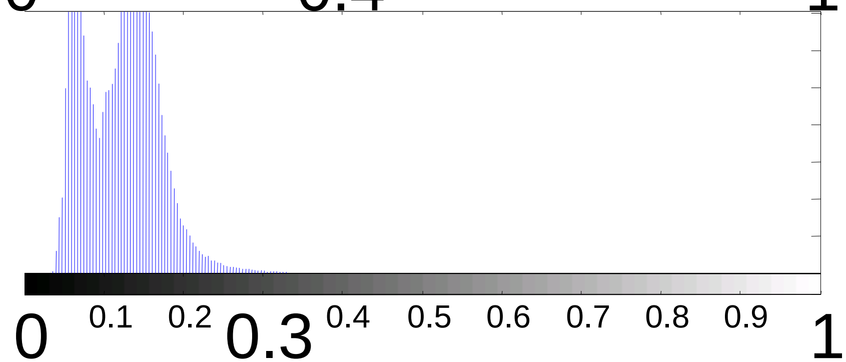
- Two important applications in color image processing:
  - **Color edge detection**
  - **Region segmentation**
    - **Data is Rio de Janeiro**

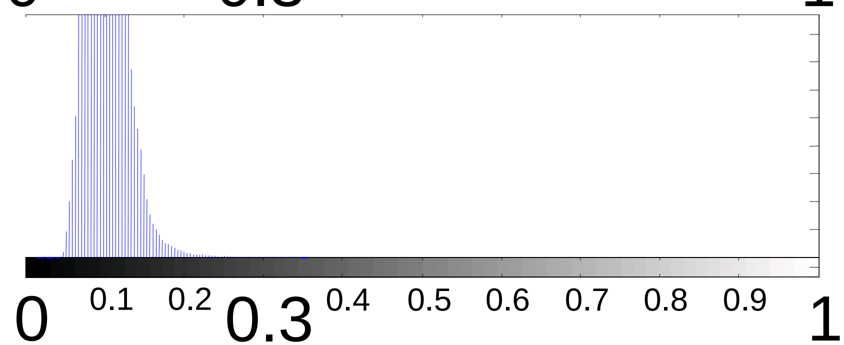
Region segmentation

1. Stretch the Near Infrared Image of Rio de Janeiro

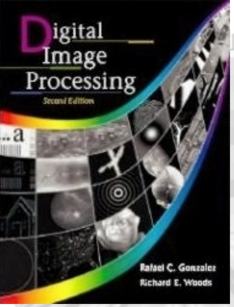








Images from MATLAB image toolbox.



Region segmentation

2 . Select a sample region with interactive function ROIPOLY

```
>> mask = roipoly(f);
>> red = immultiply(mask, f(:,:,1));
>> green = immultiply(mask, f(:,:,2));
>> blue = immultiply(mask, f(:,:,3));
>> g = cat(3, red, green, blue);
```

Region segmentation

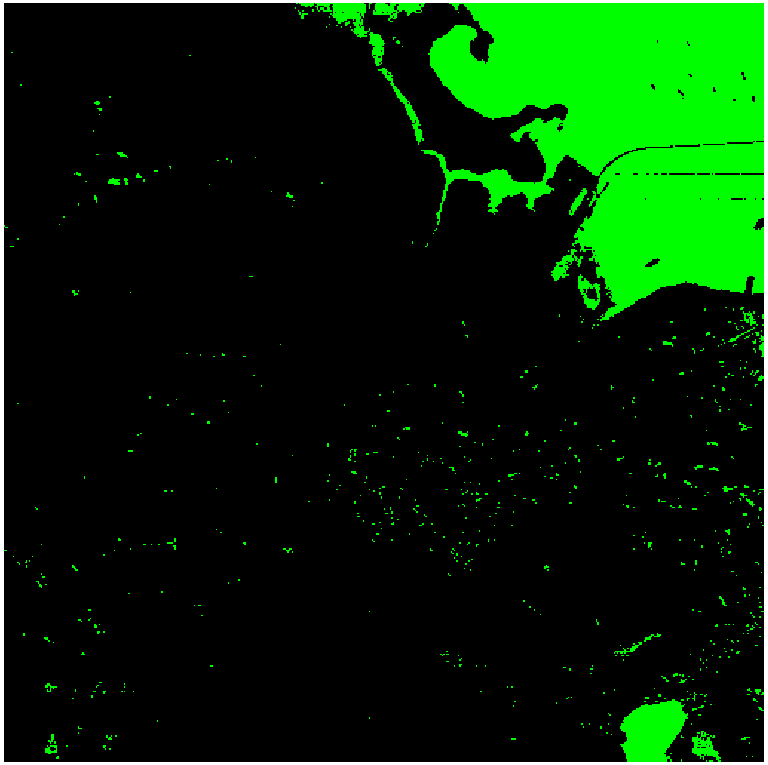
3. Find the statistics parameters from the sample region

```
>> idx = find(mask);
>> I = double(I(idx, 1:3));
>> [C, m]=covmatrix(I);
```

Region segmentation

4. Segment

E25 = colorseg('euclidean', f, 25, m);



Images from MATLAB image toolbox.