ECE5470-9

ECE 5740 Digital Image Processing

Color Image Processing

Z. Aliyazicioglu

Electrical and Computer Engineering Department
Cal Poly Pomona

Outline

- Color Fundamentals
- Color Models
- The RGB Color Model
- The CMY and CMYK Color Models
- The HSI Color Models
- Converting colors from RGB to HIS
- Pseudocolor Image Processing
- Gray Level to color Transformation
- Basic of Full-Color Image processing
- Color Transformation
- Color Slicing
- Histogram Process
- Color Image Sharpening

Color Fundamentals

- Color image processing divided into two major areas
- Full color: Images are used full color sensor, color scanner
- Pseudo color: assigning a color to a particular monochrome intensity or range of intensity
- Chromatic light spans the electromagnetic spectrum in 400 and 700nm
- Three basic quantities to describe the quality of chromatic:
- Radiance: The total amount of energy that flows from the light source [watts]
- Luminance: The measure of the amount of energy an observer <u>perceives</u> from the light source
- Brightness: express chromatic notion of intensity

Color Fundamentals

- Primary and secondary colors of light and pigments

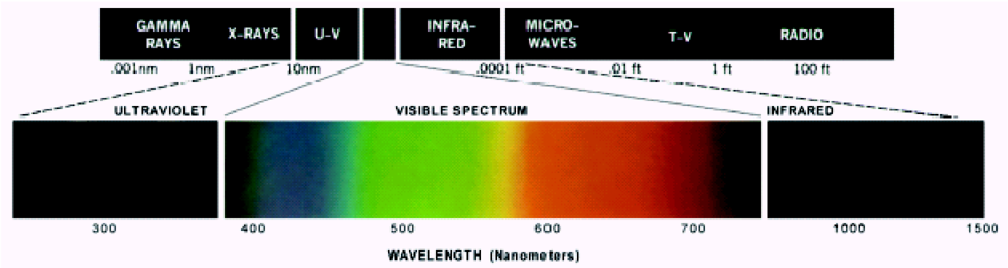
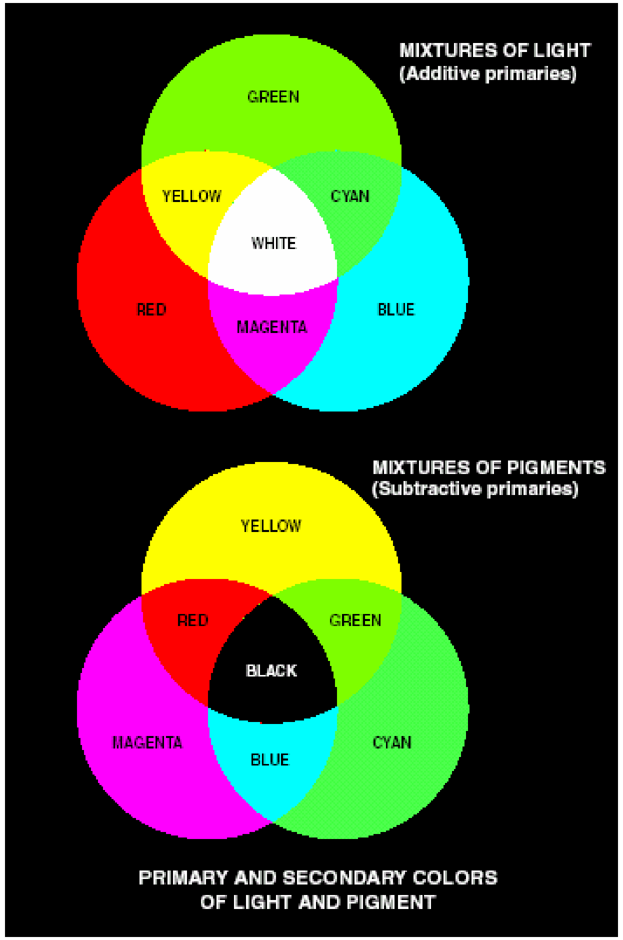
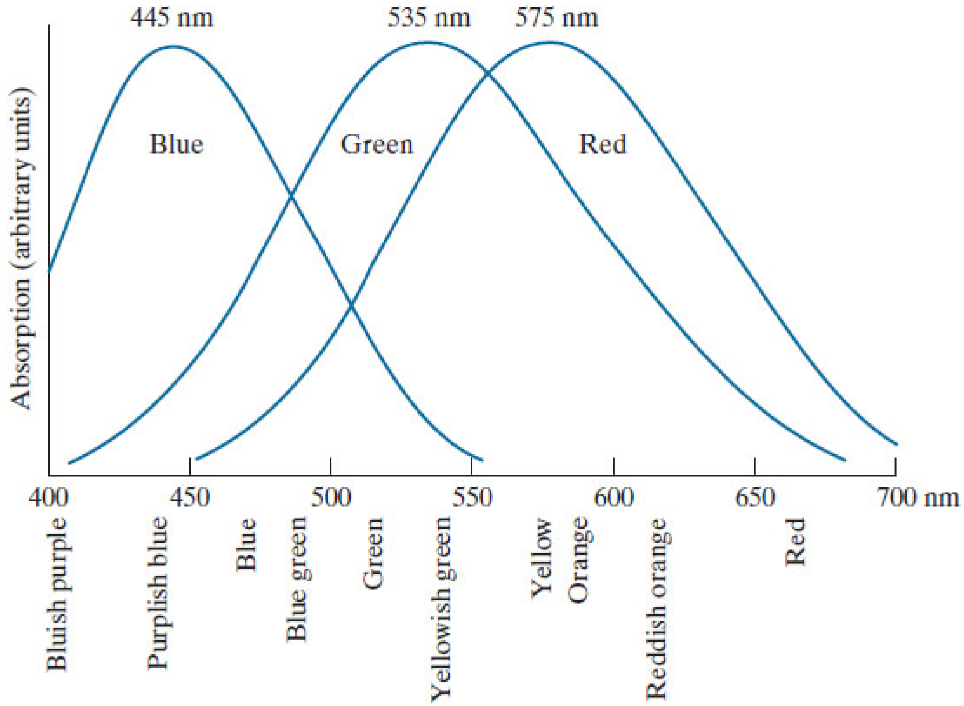


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)



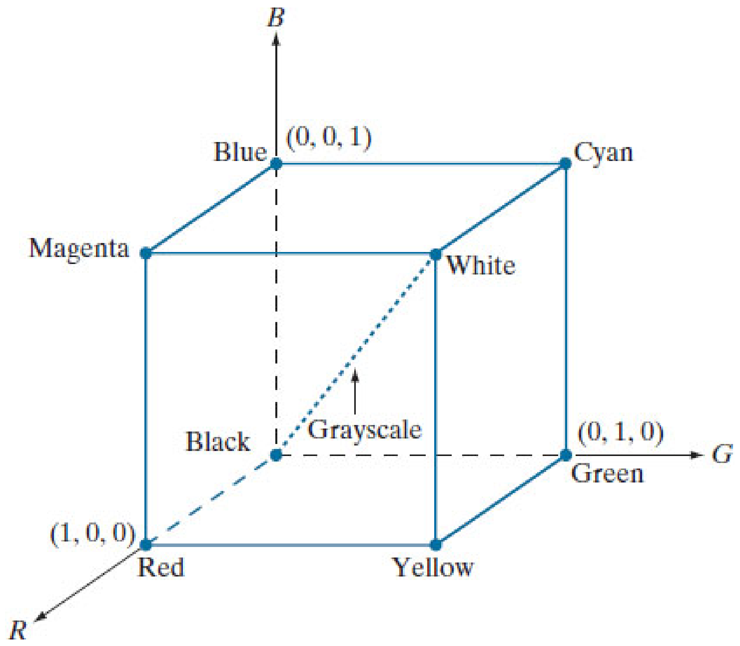
Color Models

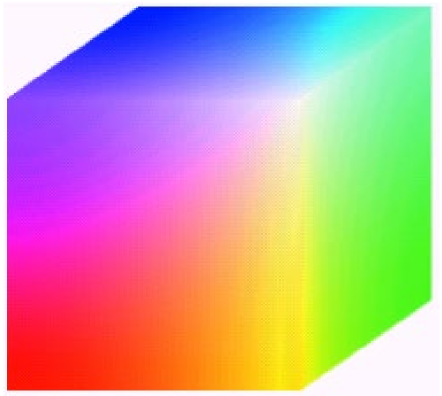
- The most commonly used models:
  - RGB (red, green, blue): The hardware oriented model. (color monitor, video camera)
  - blue=435.8 nm, green=546.1 nm, red = 600nm
  - CMY (cyan, magenta, yellow) and CMYK (cyan, magenta, yellow, black) for color printing
  - HIS (hue, saturation, intensity): has color and gray-scale information in the image



The RGB Color Model

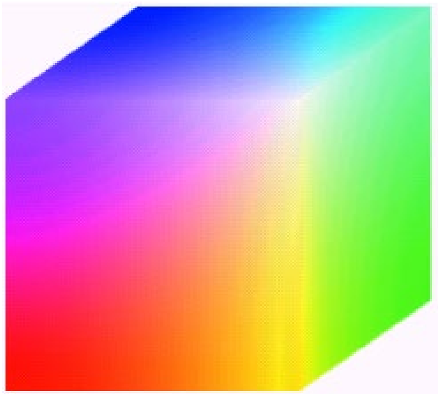
- Each color appears in its primary spectral components
- The model is based on a Cartesian coordinate system.
- RGB values are at the three corners;
- Cyan, magenta, and yellow are at three other corners
- Black is at the origin
- White is at the farthest from the origin
- The gray scale extends from black to white along the line joining these two points





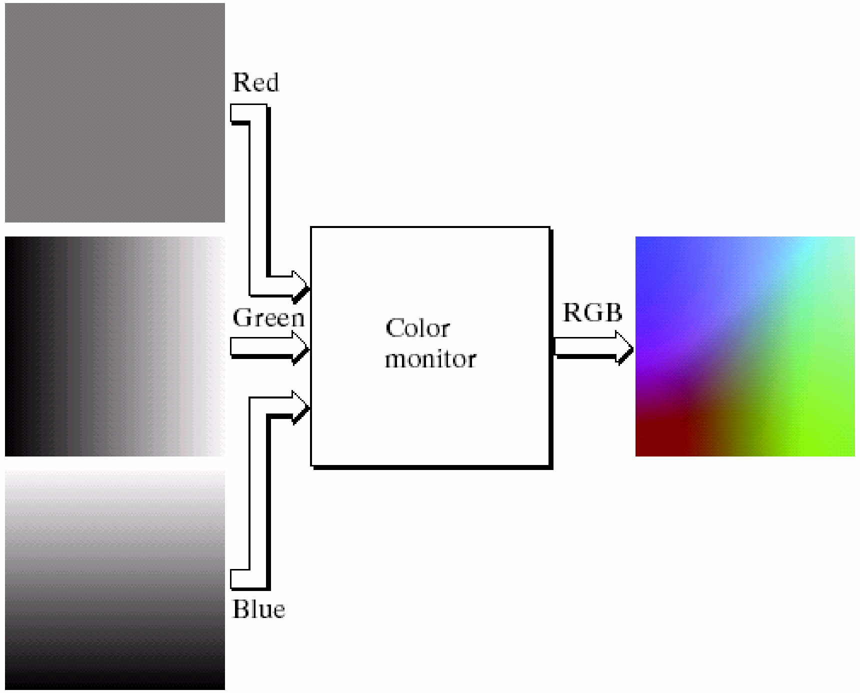
The RGB Color Model

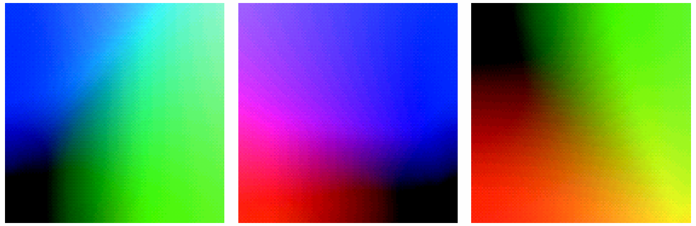
- Image representation in RGB model use three components, one for each primary color.
- The number of bits used to represent each pixel in RGB space is called the pixel depth.
- Consider an RGB image that each color has 8 bits, the pixel depth is 24 bits.
- The total number of color in 24 bit RGB image is 2^{24}=16,777,216.
- Each color has 2^{8}=256 color level



The RGB Color Model

- Generating RGB image of the cross-sectional color plane (127,G,B)
- Three hidden surface planes in the color cube



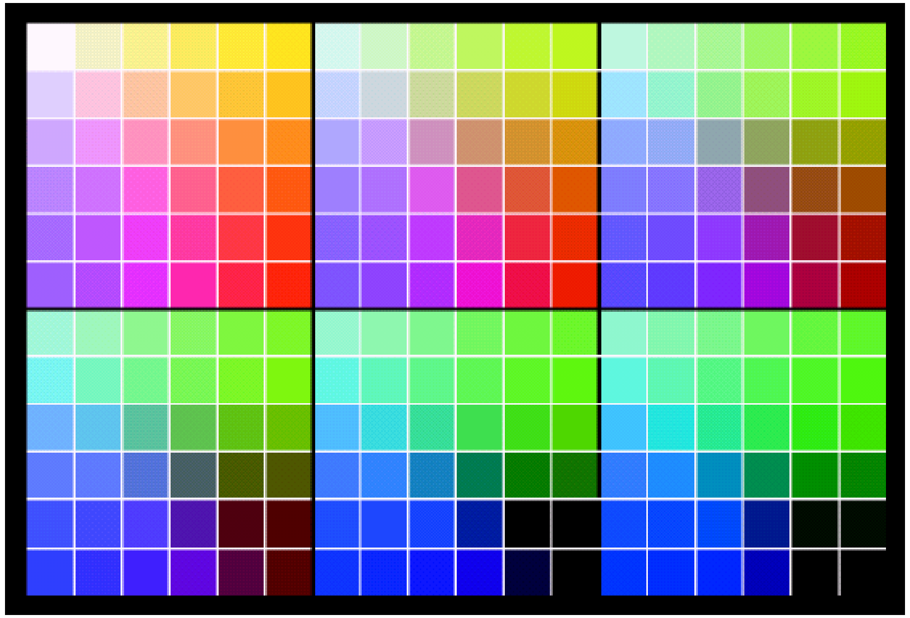


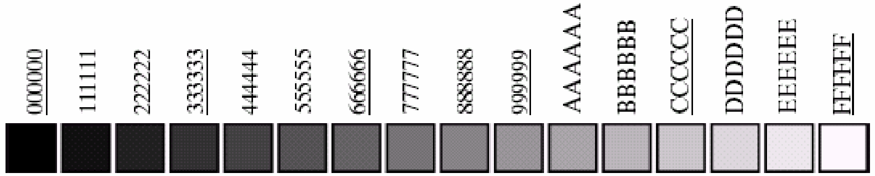
($R = 0$) ($G = 0$) ($B = 0$)

The RGB Color Model

- Many system is used today are limited to 256 colors
- This subset of colors is called the set of safe RGB colors or on internet Safe web color.
- 216 colors are common to most system
- Each color in RGB model can have 0,51,102,153,204,and 255 levels

Number System	Color Equivalents					
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255





The CMY and CMYK Color Models

- Cyan, magenta, and yellow are the secondary colors of light the primary colors.
- The conversion between CMY and RGB

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

- The color values have been normalized to the range [0,1]
- CMY produce black color.
- For printing purpose, forth color, black, is added to give CMYK color model

The HSI Color Models

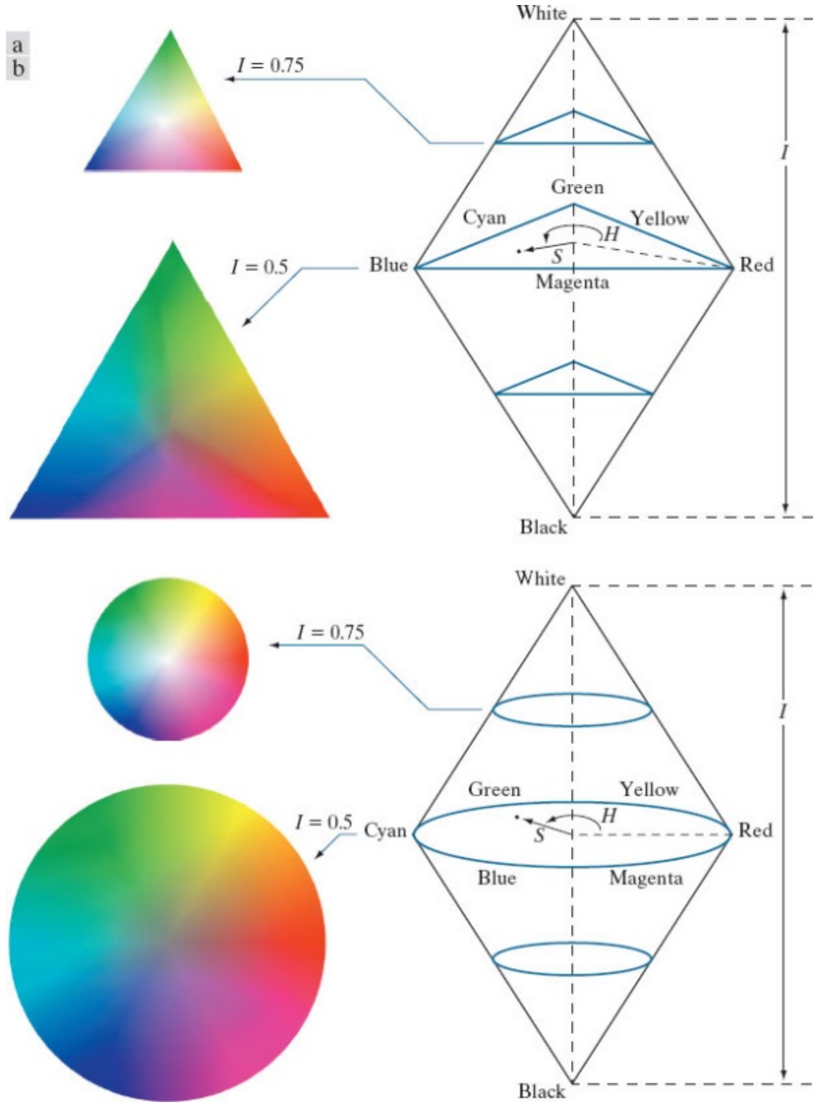
- Humans describes or view a color object by hue, saturation, and intensity.
- Hue is a color attribute that describes a pure color (pure yellow, orange, or red)
- Saturation gives a measure of the degree to which a pure color is diluted by white light
- Intensity describes color sensation and known as gray level
- HIS model is an ideal tool for developing image processing algorithms

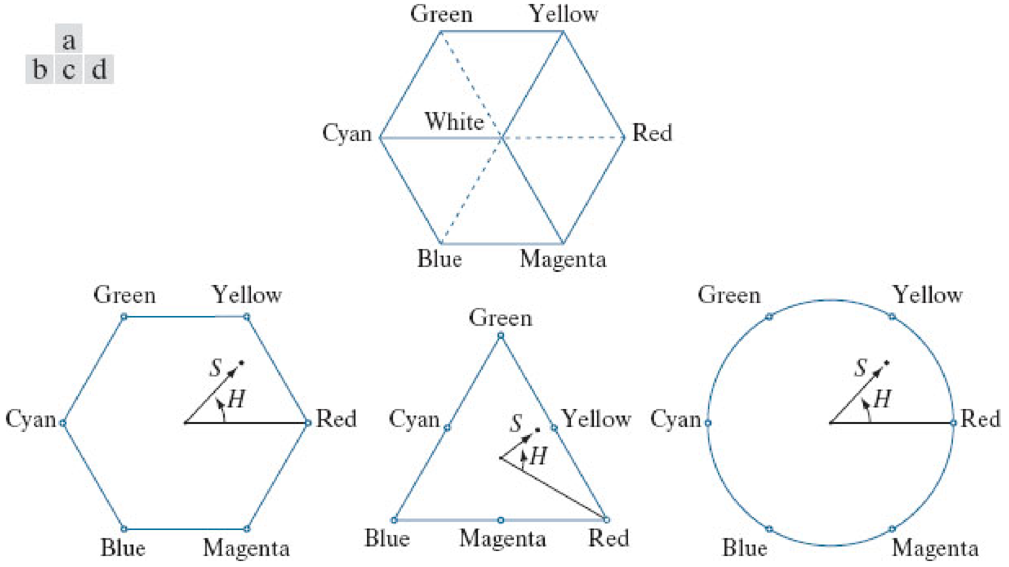
- RGB is ideal for image color generation but limited for color description

The HSI Color Models

- The HIS color model based on
  (a) triangular and (b) circular color planes

- Hue and saturation in the HIS color model





Converting colors from RGB to HSI

- Given an image in RGB color format: HSI components of each pixel are obtained as:

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \qquad \theta = \cos^{-1}\left\{ \frac{0.5\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{0.5}} \right\}$$

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

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

It is assumed that the RGB values have been normalized to the range [0, 1]

θ is measure 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

Converting Colors from HSI to RGB

- Given values of HIS in the interval [0, 1].
- Multiplying H by 360 ° to return the original range [0 °, 360 °]
- There are three sector
- RG sector ($0^\circ \le H \le 120^\circ$)

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

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

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

GB sector ($120^\circ \le H \le 240^\circ$)

$$H = H - 120^\circ$$

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

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

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

Converting colors from HSI to RGB

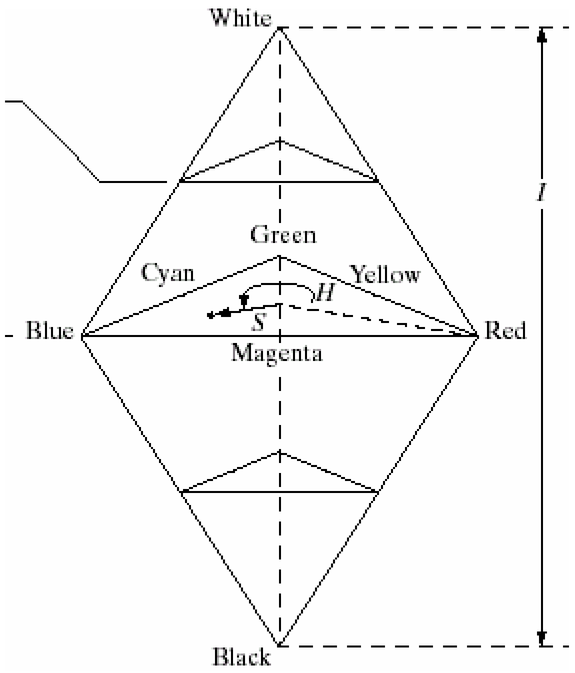
- BR sector $(240° \le H \le 360°)$

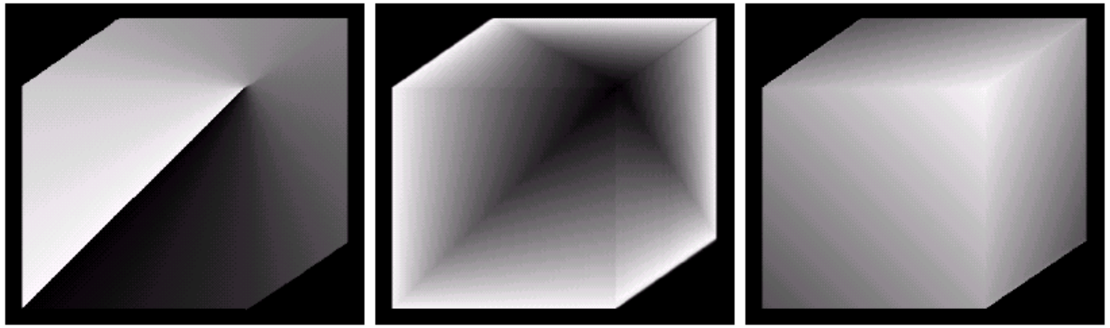
$$H = H - 240^{\circ}$$

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

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

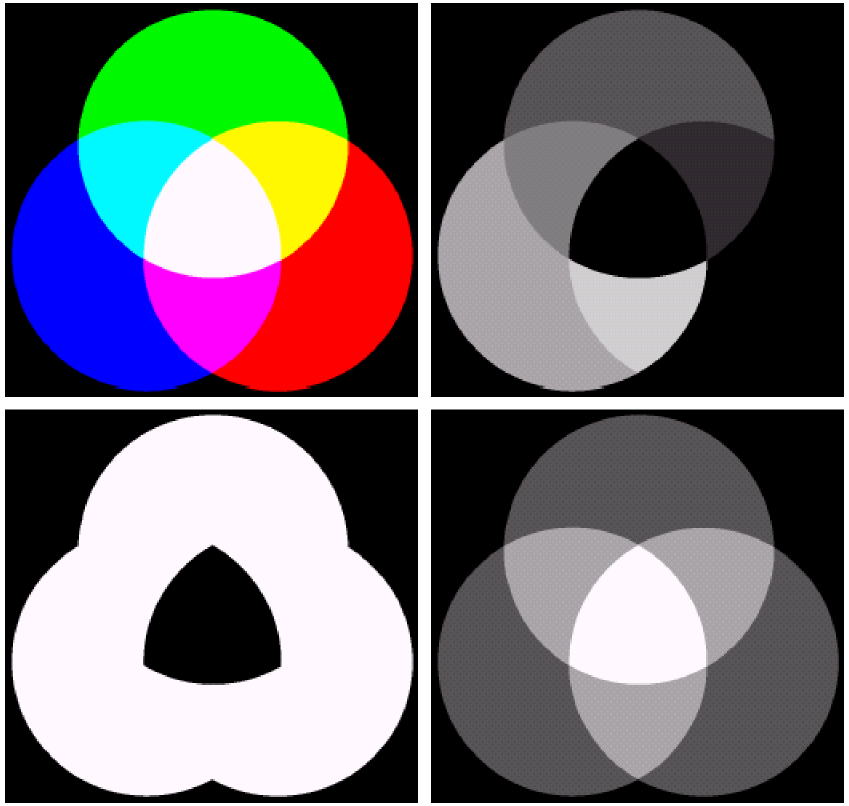
$$R = 3I - (G + B)$$





Converting colors from HSI to RGB

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

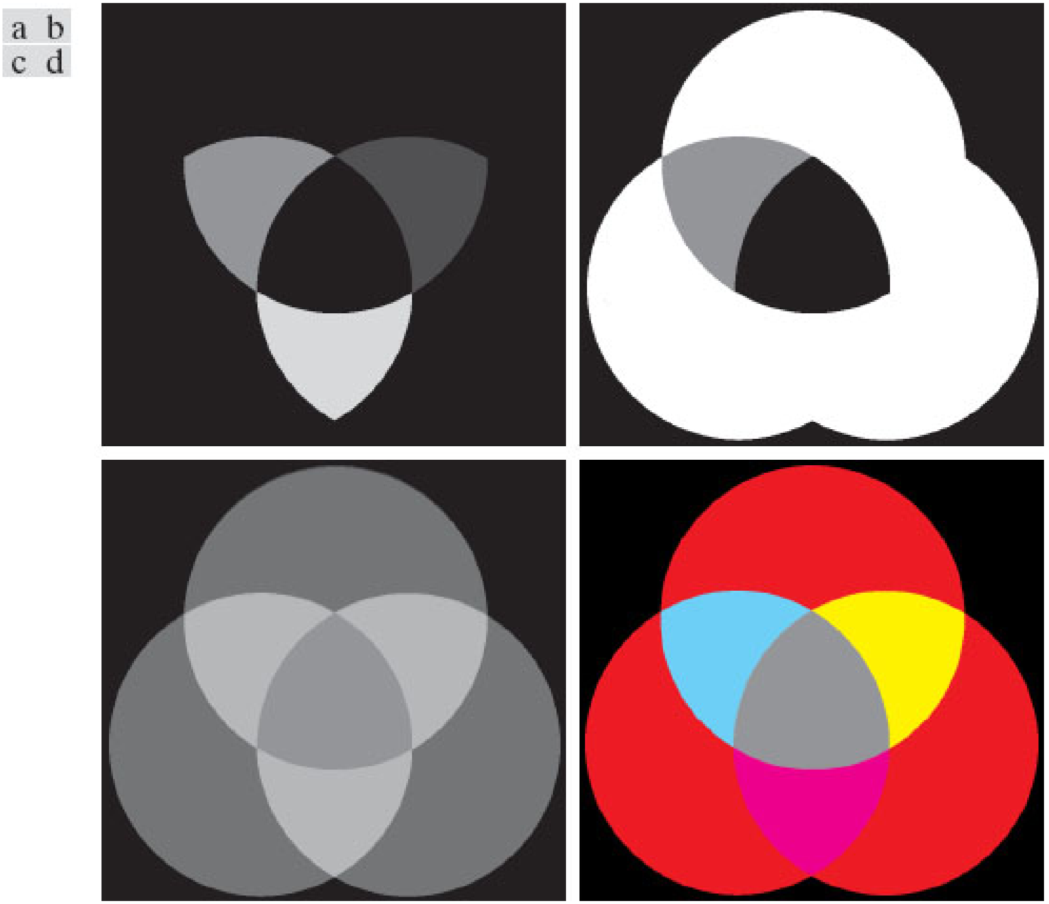


a b
c d

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

Converting colors from HSI to RGB

a)-(c) Modified HSI component images.
(d) Resulting RGB image.
(See Fig. 7.14 for the original HSI images.)



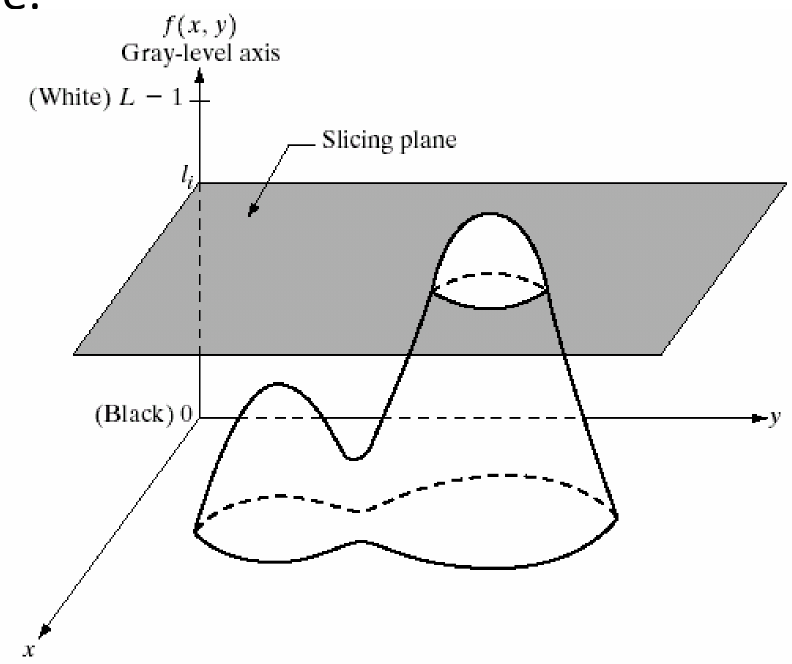
Pseudocolor Image Processing

Intensity Slicing

- An image is interpreted as a 3-D function (intensity and coordinates) and slices the function in the area of intersection
- A different color is assigned to each side of plane.
- If gray level partitioned into P+1 intervals, V_1, V_2, ….,V_{P+1}
- The gray level to color assignments are

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

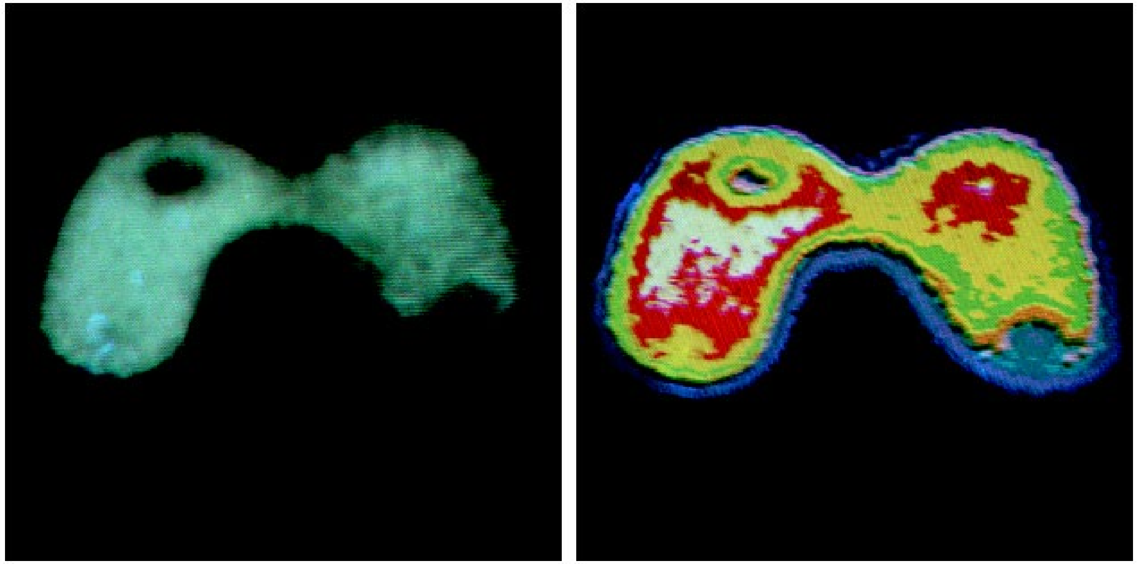
- Where c_k is the color associated with the k^{th} intensity interval V_k



Pseudocolor Image Processing

(a) A monochrome image

(b) The result of intensity slicing the image into eight color regions.

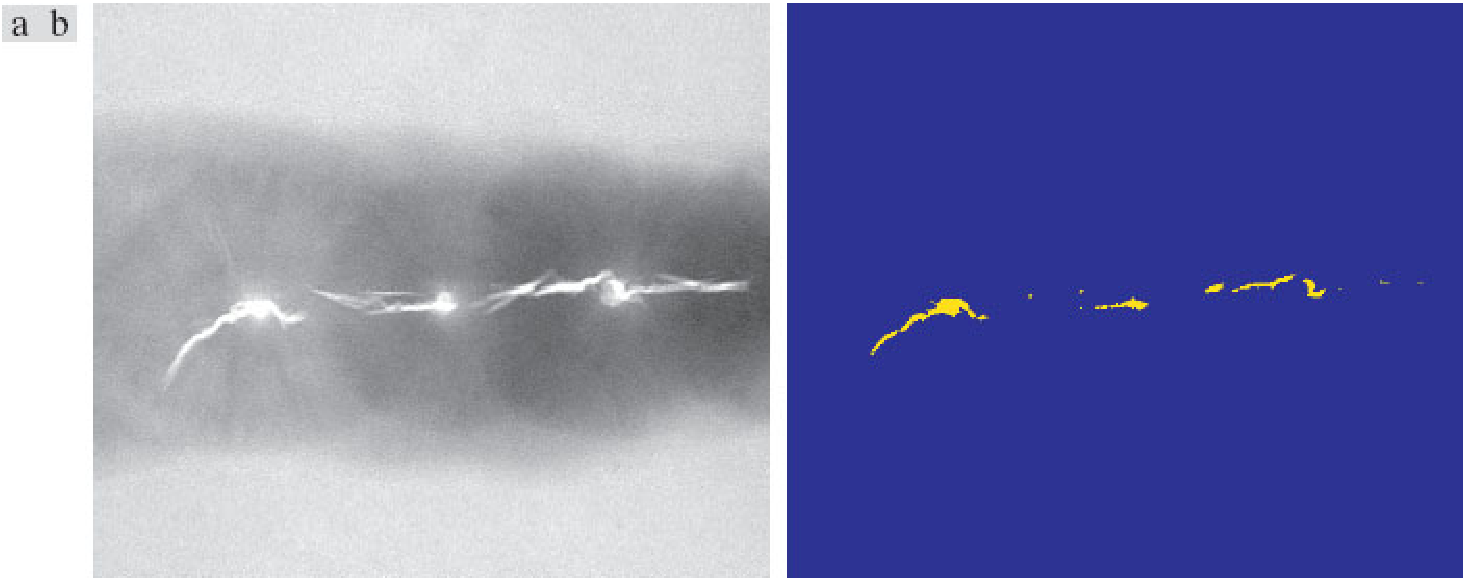


Pseudocolor Image Processing

(a) X-ray image of a weld.

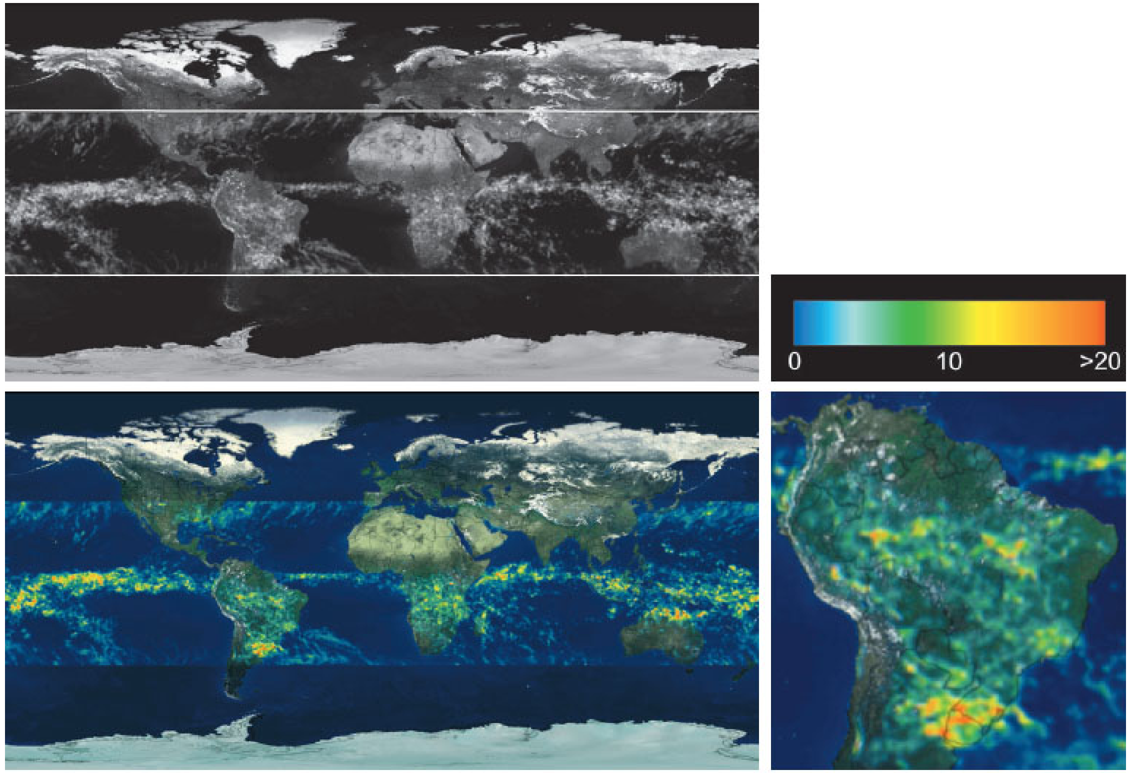
(b) Result of color coding.

(Original image courtesy of X-T E K Systems, Ltd.)



Pseudocolor Image Processing

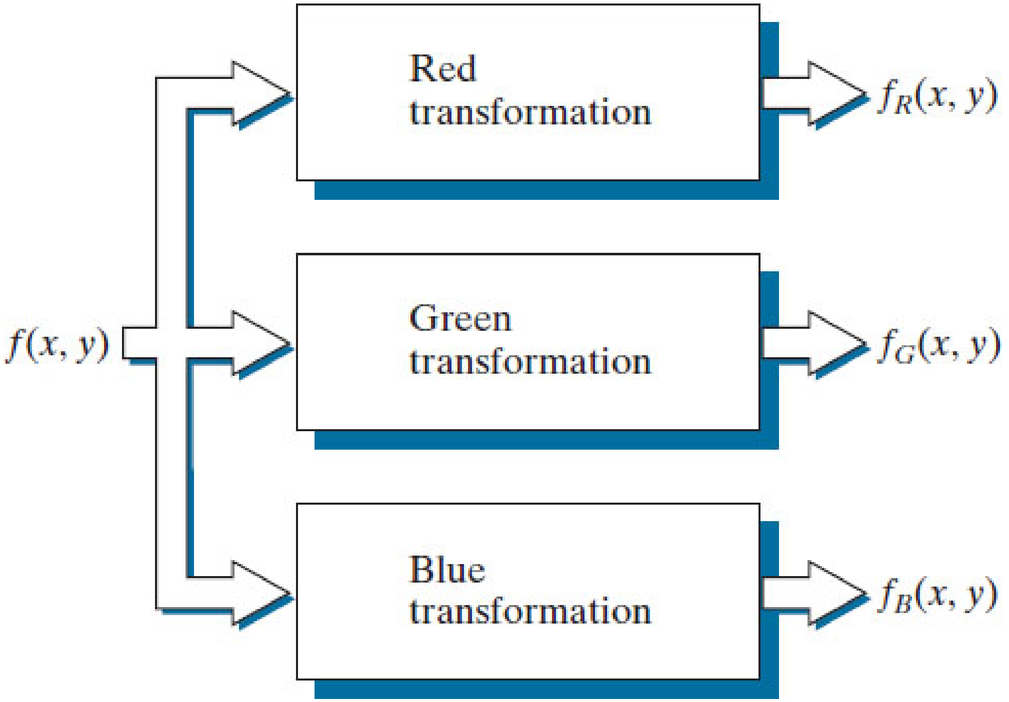
(a) Grayscale image in which intensity (in the horizontal band shown) corresponds to average monthly rainfall.
(b) Colors assigned to intensity values.
(c) Color-coded image.
(d) Zoom of the South American region.
(Courtesy of NASA.)



a b
c d

Pseudocolor Image Processing

- Approach is perform three independent transformations on the gray level of any input pixel
- The three results are the fed separately into the red, green, and blue channels of a color TV

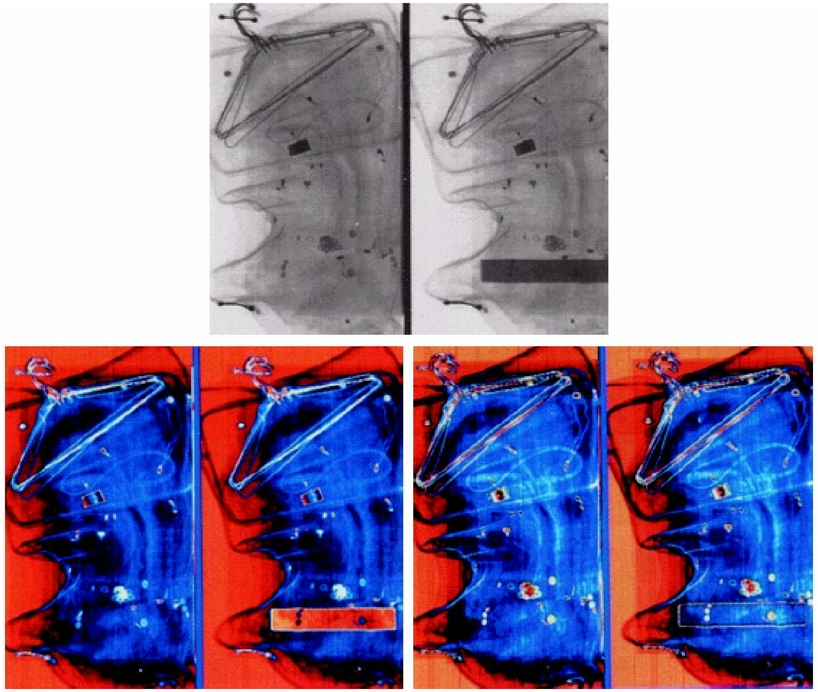


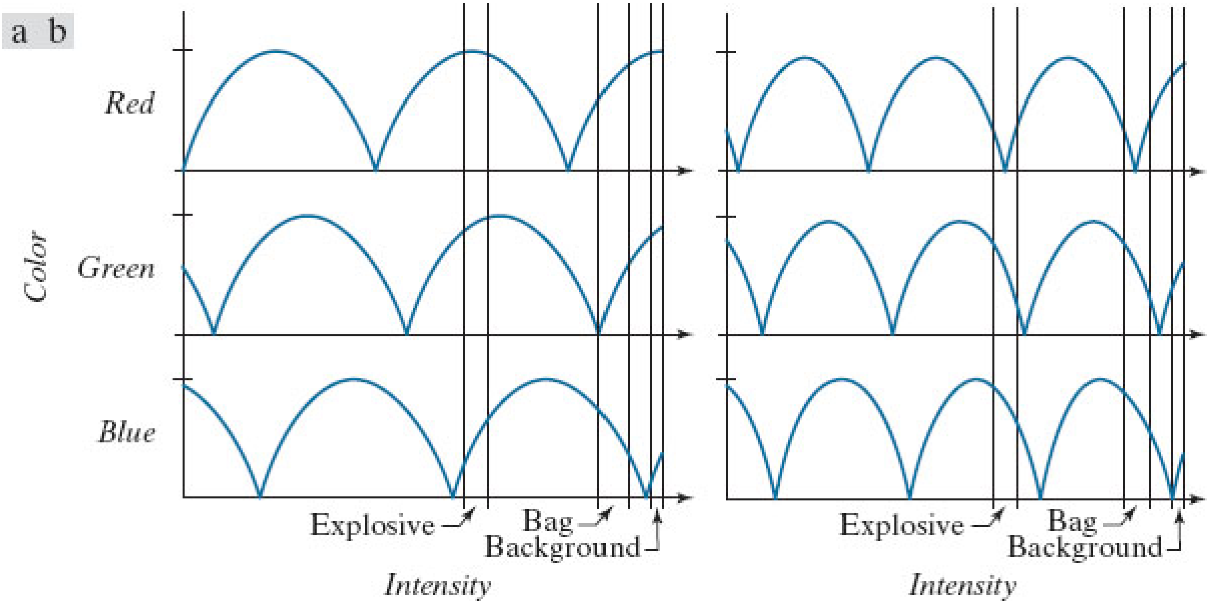
Functional block diagram for pseudocolor image processing.

Images f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Example:

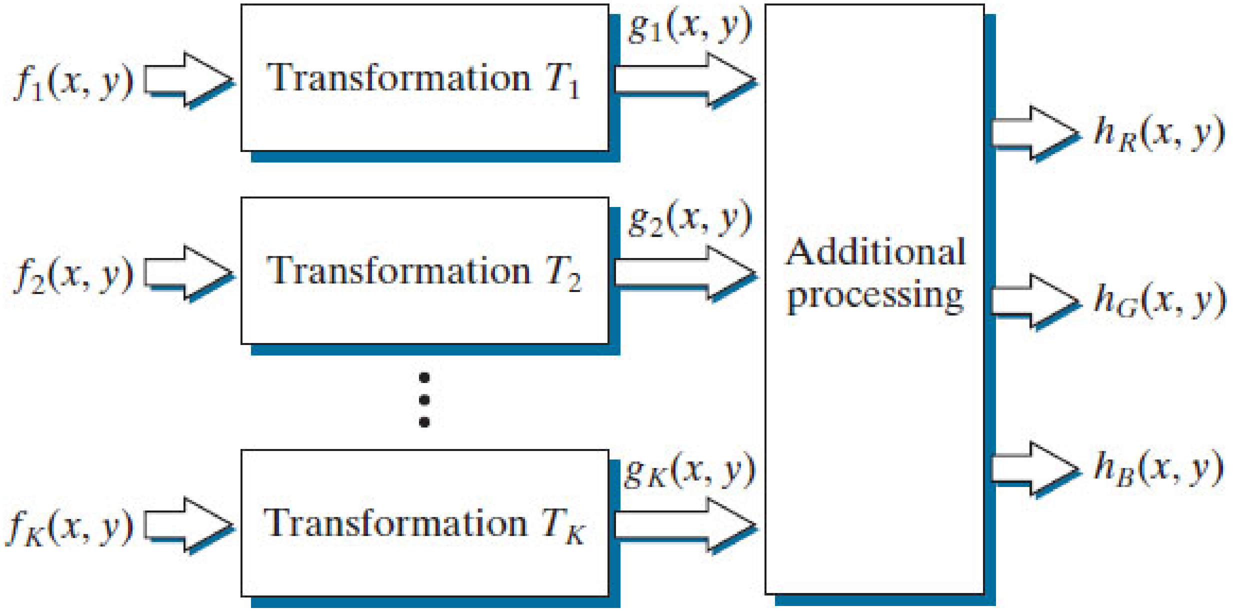
- Psuedocolor enhancement by using gray-level to color transformation
- The transformation function used to obtain the pseudocolor images





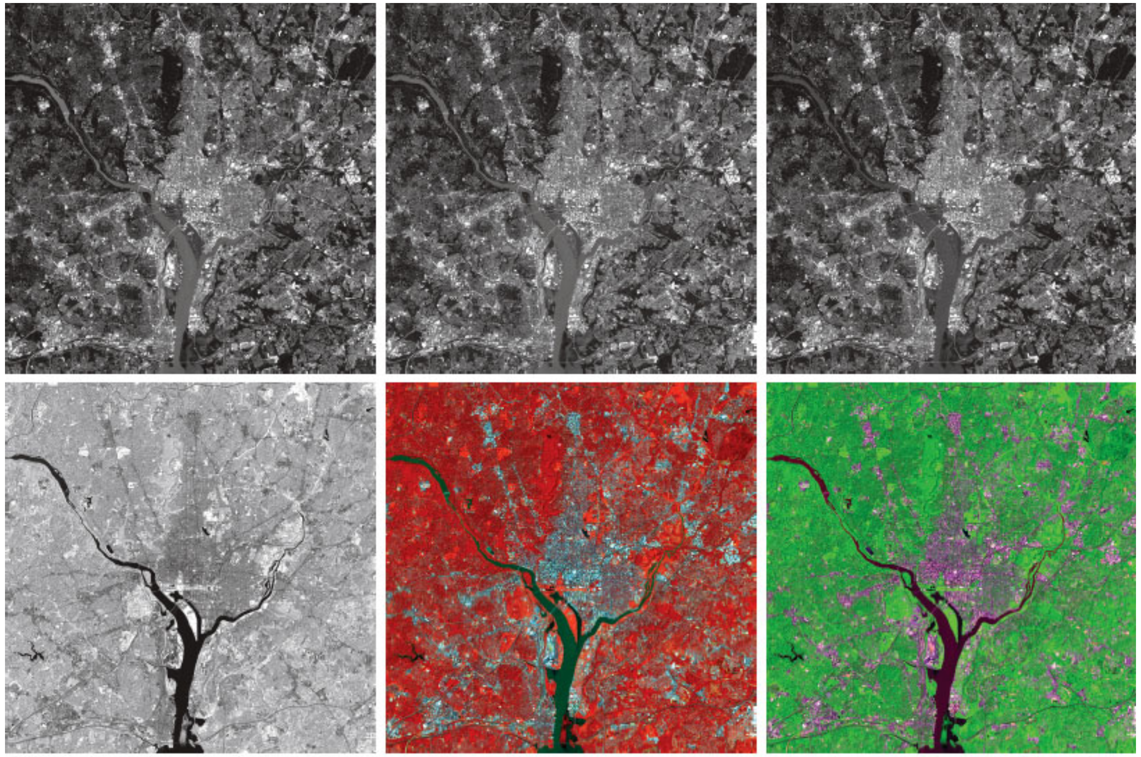
Basic of Full-Color Image processing

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



Pseudocolor Image Processing

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



a b c
d e f

Basic of Full-Color Image processing

- Full color image processing approaches fall into two major categories
  - We process each component image individually and the for a composite processed color image from each component
  - We work with color pixels directly. Each color pixels can be interpreted as a vector
  - Let c represent an arbitrary vector in RGB color space

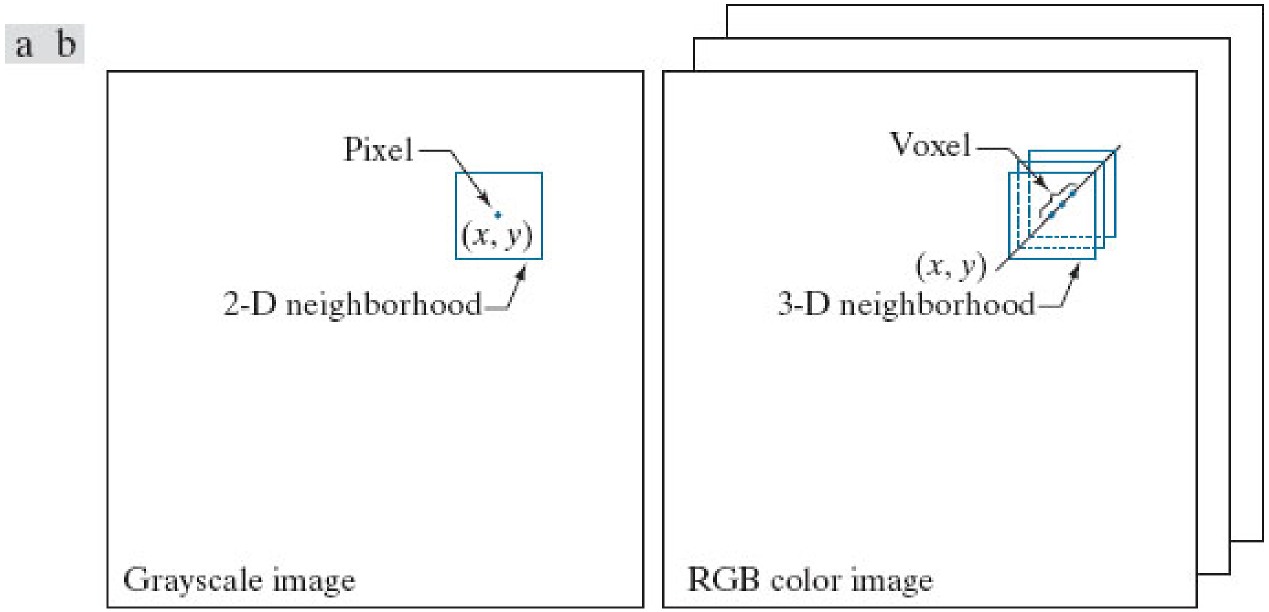
$$c(x,y)=\begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix}=\begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

  - For image size MxN there are MN such vectors c(x,y) for x=0,1,…,M-1, and y=0,1,…,N-1

Color Transformation

Spatial neighborhoods for grayscale and RGB color images.

Observe in (b) that a single pair of spatial coordinates, (x,y), addresses the same spatial location in all three images.



Color Transformation

Formulation:

- Model color transformation

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

- Where f(x,y) is color input image, g(x,y) is the transformed of processed color output image, and T is an operator on f

- Or color transformation of the form

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

- Where r_i and s_i variables denoting the color components of f(x,y) and g(x,y) at any point (x,y).
- n is the number of color components
- $\{T_1, T_2, \ldots, T_n\}$ is set of transformation or color mapping function

Example:

a) A full color image and its various color-space components

Example:

Adjusting the intensity of an image using color transformations.

(a) Original image.

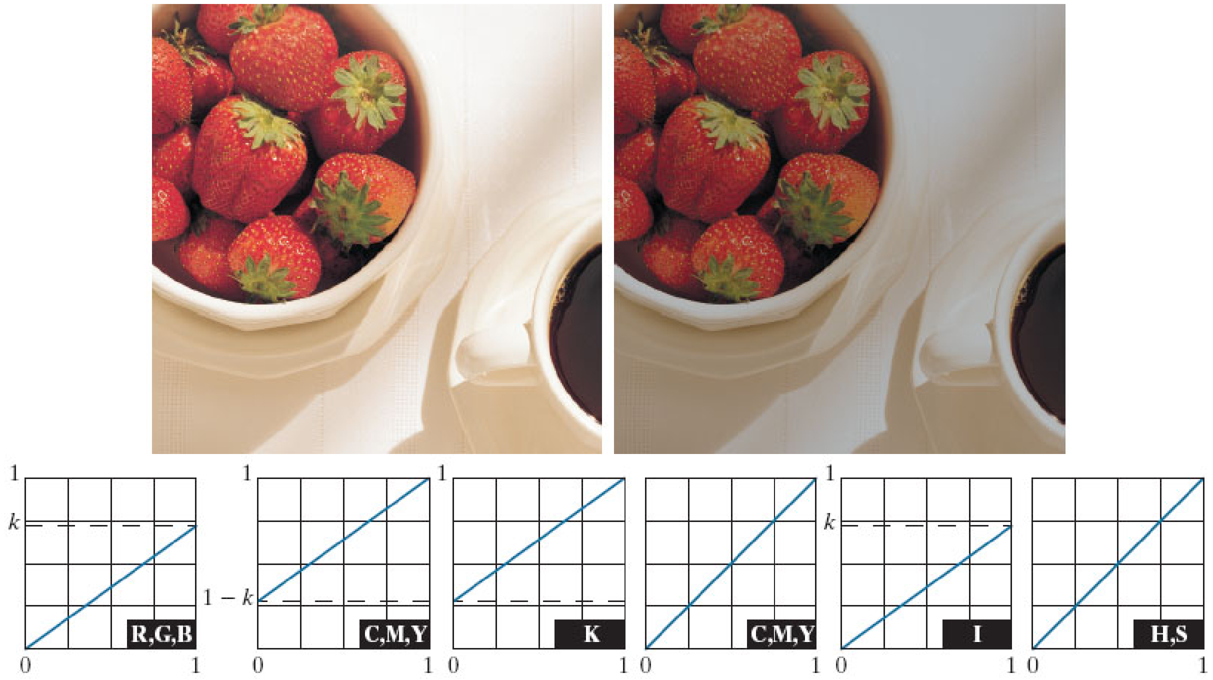
(b) Result of decreasing its intensity by 30% (i.e., letting k = 0.7).

(c) The required RGB mapping function.

(d) (e) The required CMYK mapping functions.

(f) The required CMY mapping function.

(g)–(h) The required HSI mapping functions.

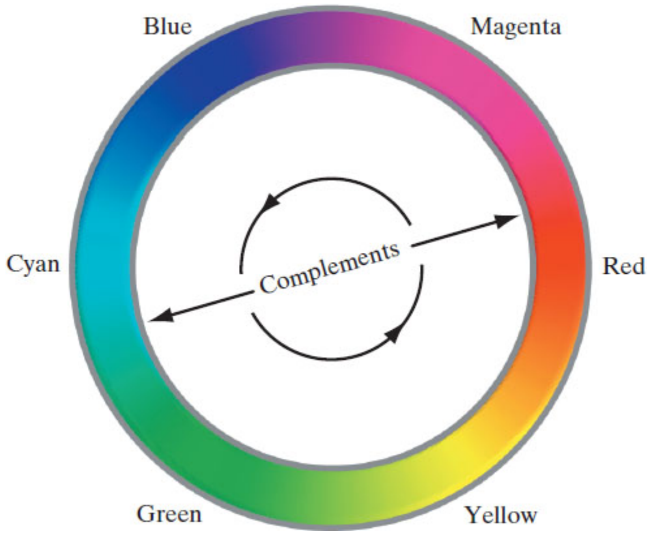


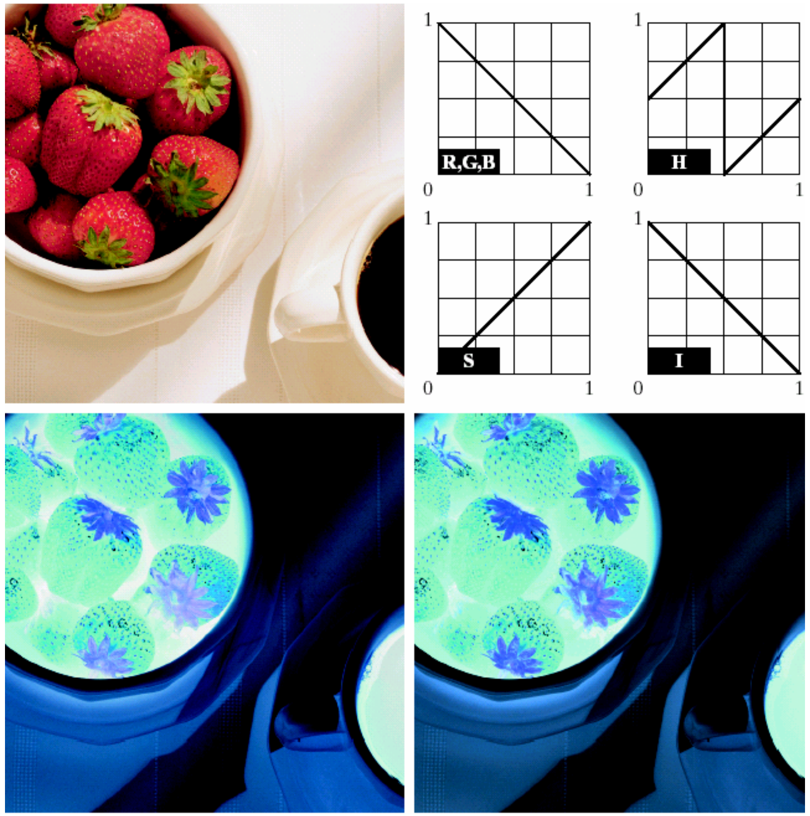
a b
c d e f g h

Color Transformation

Color Complement

- The hues directly opposite one another on the color circle of the following figure are called complements.





Original image

Complement transformation

Complement of (a) based on the RGB mapping

Complement of HSI transformation

Color Slicing

- Idea is:
  - Display the color of interest so that they stand out from the background
  - Use the region defined by the colors as mask for further processing
- If the colors of interest are enclosed by n>3 of width W and color components ($a_1, a_2, ..a_n$), Transformation

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

To specify the color of interest

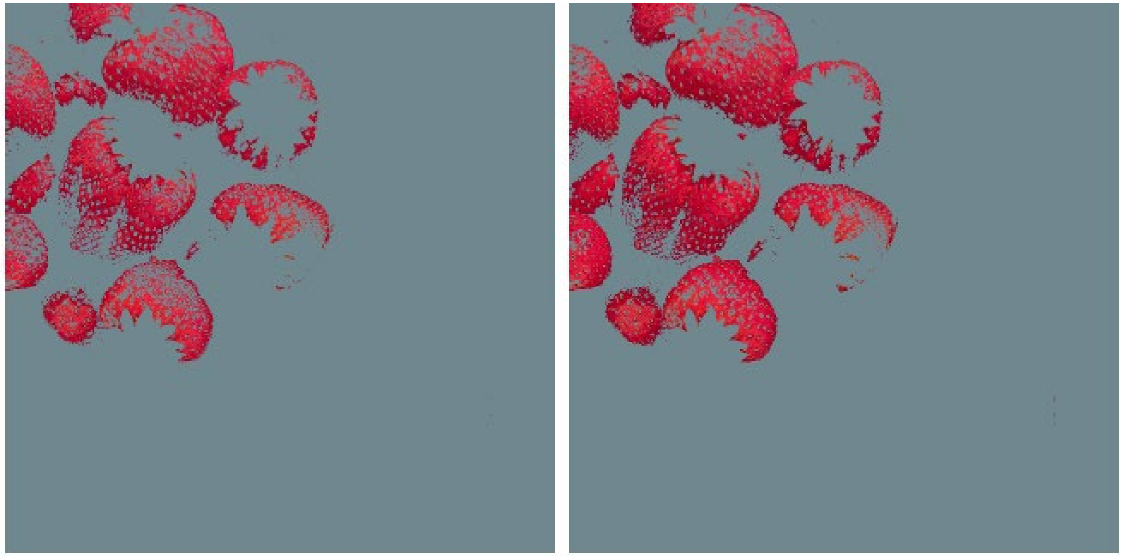
$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^{n} (r_j - a_j)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases}$$

Example:

Color slicing transformation that detect
(a) red within an RGB cube of width W=0.2549 centered at (0.6863, 0.1608, 0.1922)

(b) reds within an RGB sphere of radius 0.1765 centered at the same point

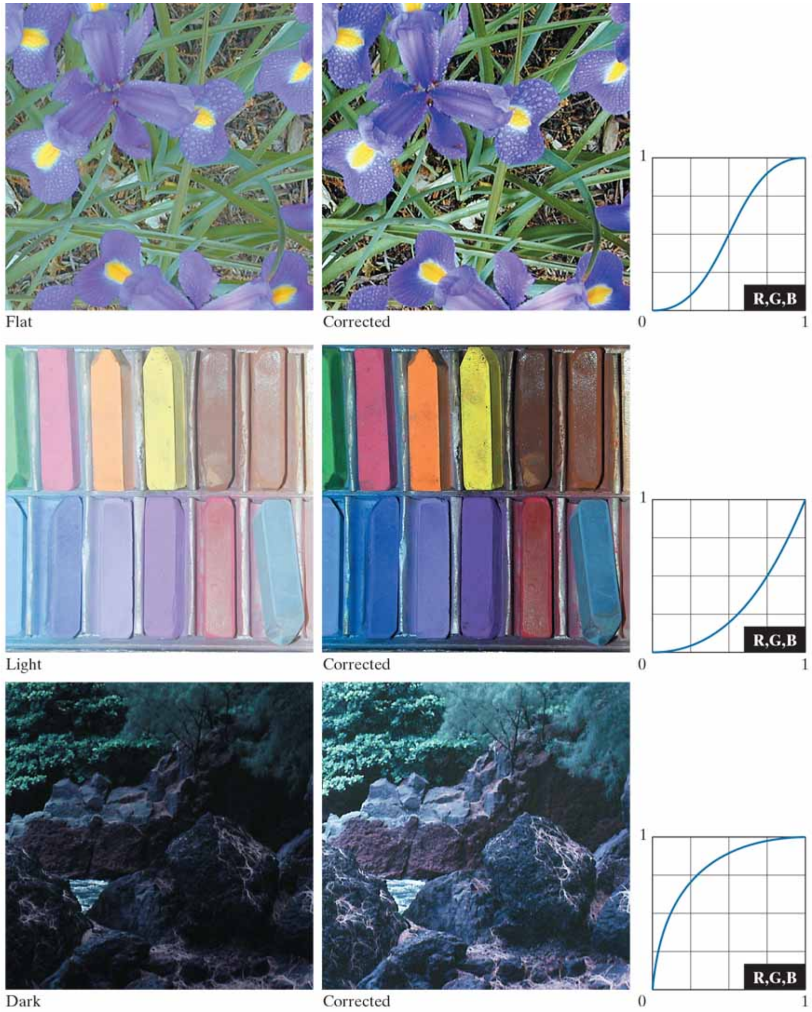
The pixels outside the cube and sphere were replaces by color (0.5, 0.5, 0.5)



Tone and Color Correction

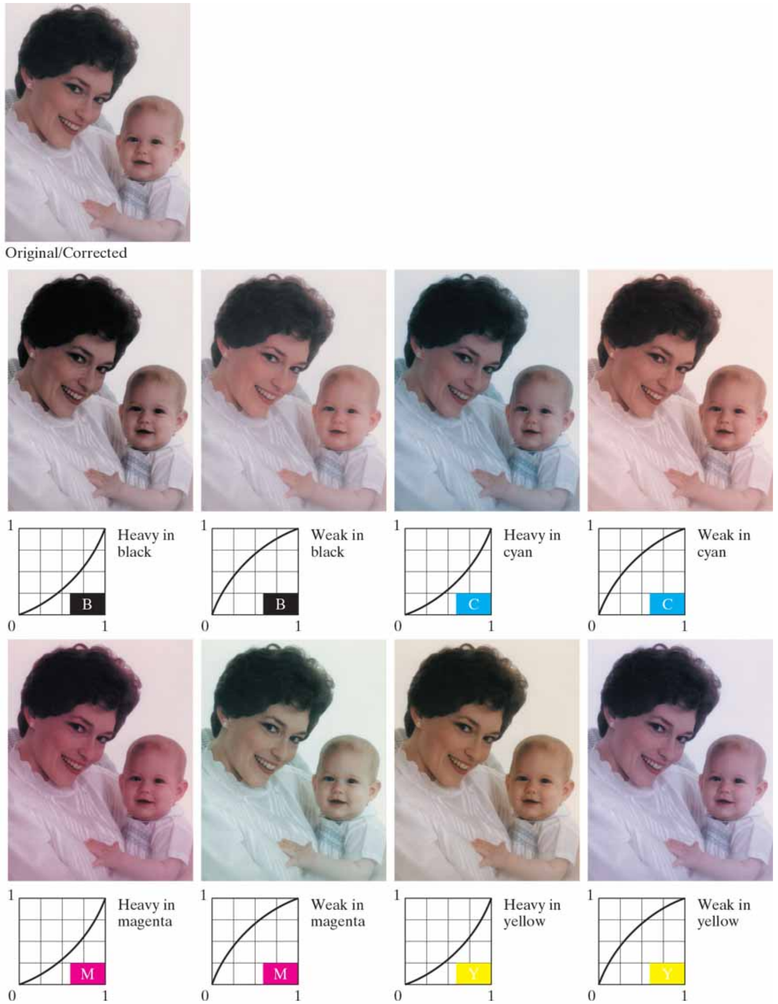
Tonal corrections for flat, light (high key), and dark (low key) color images.

Adjusting the red, green, and blue components equally does not always alter the image hues significantly.



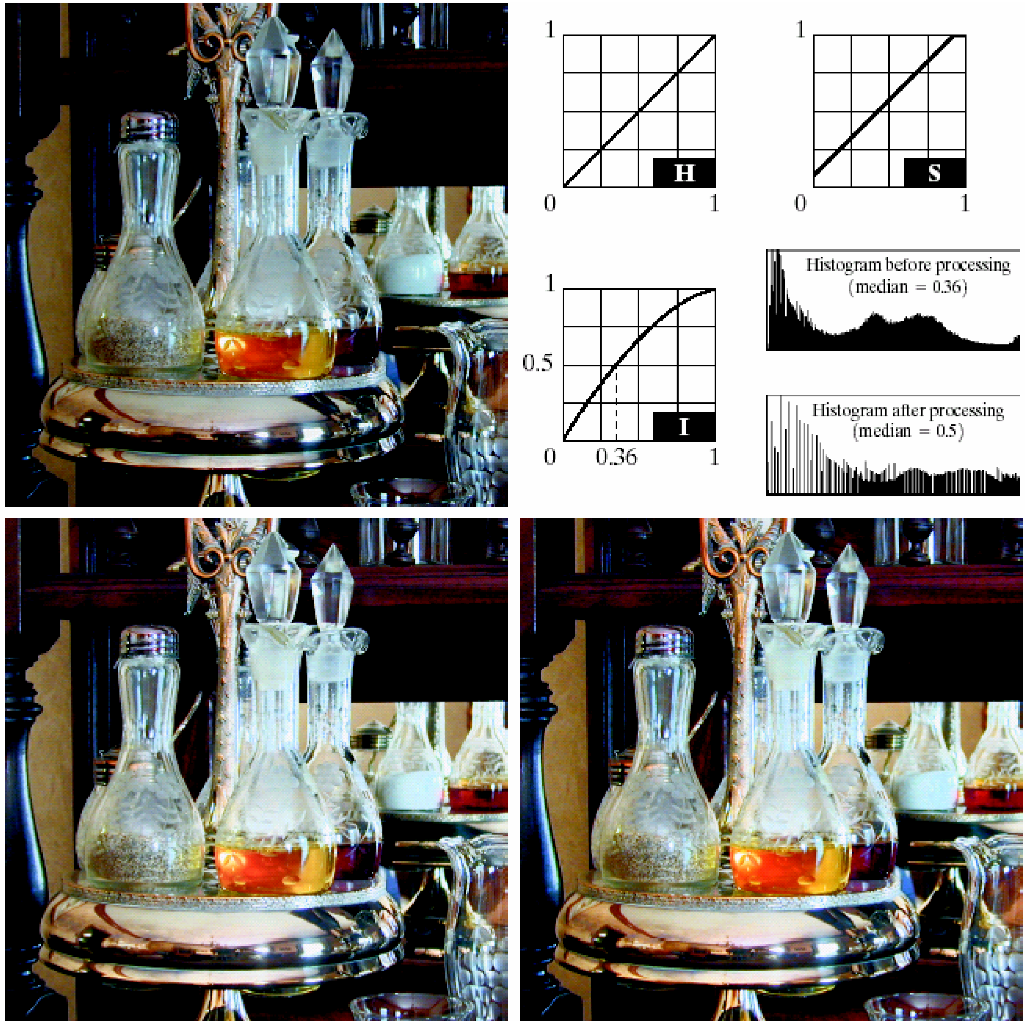
Tone and Color Correction

Color balancing a CMYK image.



Tone and Color Correction

- Histogram equalization automatically determines a transformation that seeks to produce an image with a uniform histogram of intensity value.



Smoothing and Sharpening

- Let S_{xy} denotes the set of coordinates defining a neighborhood centered at (x,y) in an RGB color image.
- The average of the RGB components

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

Example:

- RGB Image
- Red component image
- Green component image
- Blue component image

Example:

HSI components of the RGB color image in Fig. 7.36(a).

(a) Hue.

(b) Saturation.

(c) Intensity.

a b c

Example:

- Image smoothing with a 5x5 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 b c

Color Image Sharpening

- Consider image sharpening using the Laplacian
- In RGB system, the Laplacian of vector c

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

- Compute Laplacian of a full color image by computing the Laplacian of each component image separately

Example

Image sharpening using the Laplacian.

(a) Result of processing each RGB channel.

(b) Result of processing the HSI intensity component and converting to RGB.

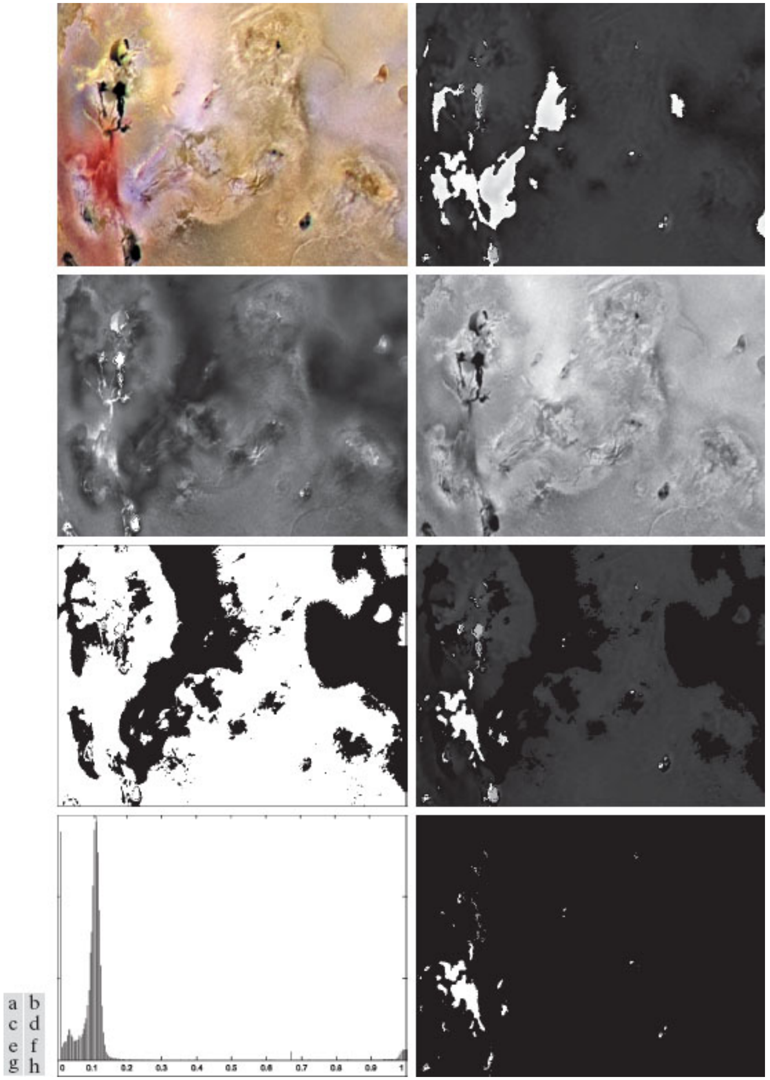
(c) Difference between the two results.

a b c

Using Color in Image Segmentation

- Chapter 10-11 covers it in detail.
- Briefly,
- Segment an image is based on color
- Image segmentation in HSI space.

(a) Original.
(b) Hue.
(c) Saturation.
(d) Intensity.
(e) Binary saturation mask (black = 0).
(f) Product of (b) and (e).
(g) Histogram of (f).
(h) Segmentation of red components from (a).



Using Color in Image Segmentation

- Image segmentation in RGB space may get better results by using RGB colors vector.
- Obtain an estimate of a the "average" color that we wish to segment.
- The average color denoted by the RGB vector **a**
- Each RGB pixel in a given image as having a color in the specified range or not.
- The simplest measurement is the Euclidean distance
- Let **z** denote an arbitrary points in RGB space
- **z** is similar to a if the distance between them is less than a specified threshold D_0.

- $D(\boldsymbol{z}, \boldsymbol{a}) = \|\boldsymbol{z} - \boldsymbol{a}\|$
- $= [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{1/2}$

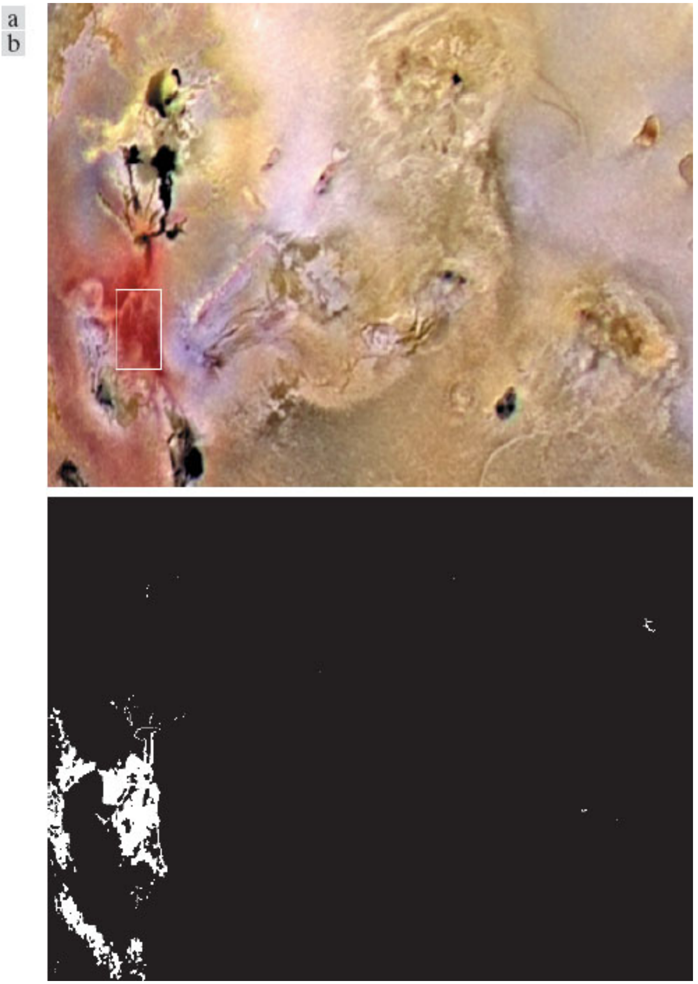
If $D(\boldsymbol{z}, \boldsymbol{a}) \leq D_0$

Using Color in Image Segmentation

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. 7.40(h).



Color Edge Detection

(a) RGB image.

(b) Gradient computed in RGB color vector space.

(c) Gradient image formed by the elementwise sum of three individual gradient images, each computed using the Sobel operators.

(d) Difference between (b) and (c).

Color Edge Detection

Component gradient images of the color image in Fig. 7.44.

(a) Red component,

(b) green component, and

(c) blue component. These three images were added and scaled to produce the image in Fig. 7.44(c).

a b c

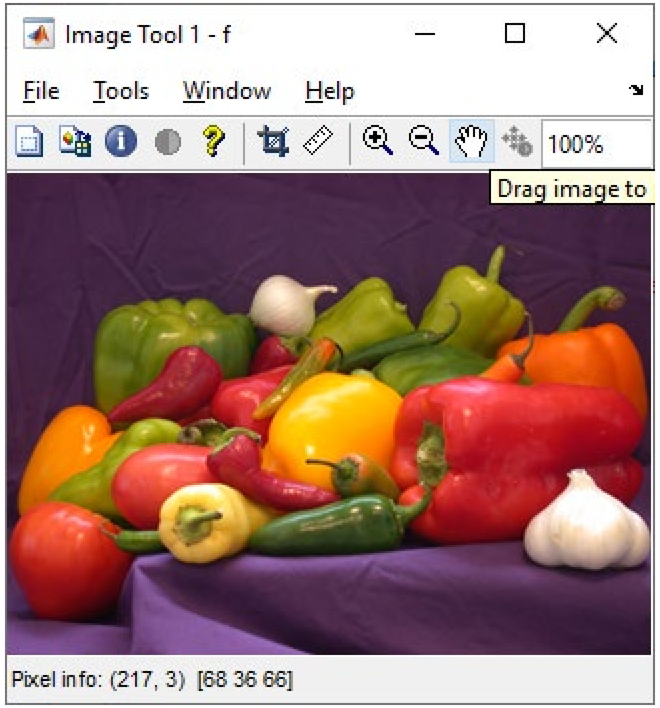
MATLAB

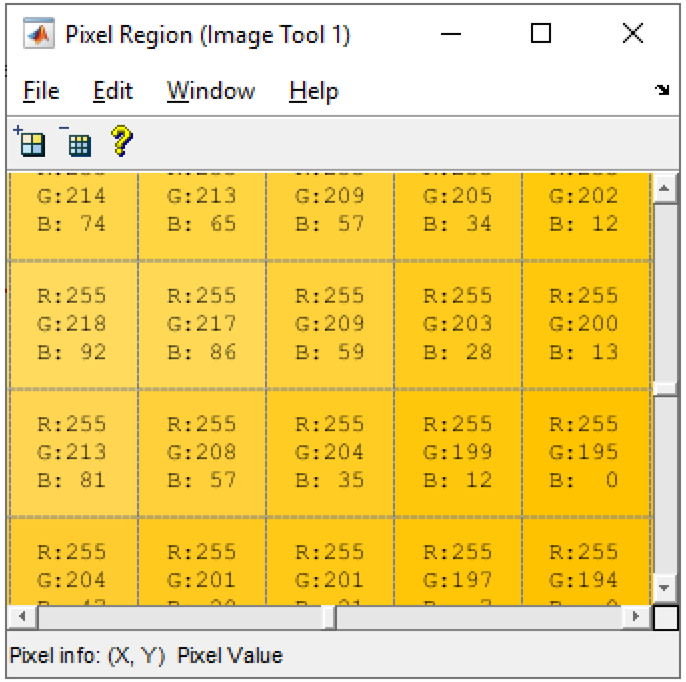
% Displaying RGB Images

RGB = imread(`peppers.png');

imshow(RGB)

or imtool(RGB)

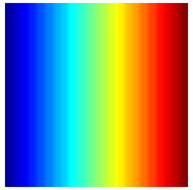


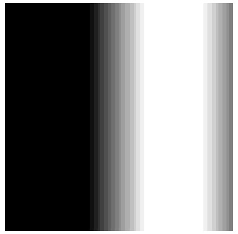


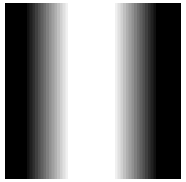
MATLAB

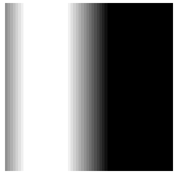
```
%displays each color plane image separately

RGB=reshape(ones(64,1)*reshape(jet(64),1,192),[64,64,3]);
imshow(RGB)
R=RGB(:,:,1);
G=RGB(:,:,2);
B=RGB(:,:,3);
figure, imshow(R)
figure, imshow(G)
figure, imshow(B)
```









MATLAB

% Reducing the Number of Colors in an Image

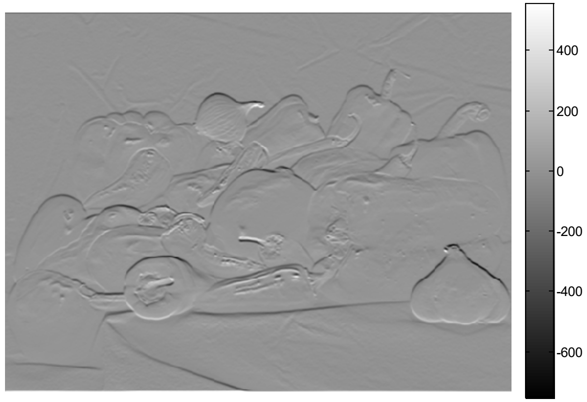
```
RGB = imread('peppers.png');
imshow(RGB);
[X,map] = rgb2ind(RGB,128);
figure, imshow(X,map)
```

[X,map] = rgb2ind(RGB,n) converts the RGB image to an indexed image X using minimum variance quantization. map contains at most n colors. n must be less than or equal to 65536.

MATLAB

% Converting Gray scale and filtering

```
>> RGB = imread('peppers.png');
>> imshow(RGB)
>> I = rgb2gray(RGB);
>> figure, imshow(I)
>> h = [1 2 1; 0 0 0; -1 -2 -1];
>> I2 = filter2(h,I);
>>figure, imshow(I2,[]), colorbar
```



MATLAB

% NTSC Color Space (Conver gray scale)

```
rgb_img = imread('ngc6543a.jpg'); % Load the image
image(rgb_img) % Display the RGB image
Y = .2989*rgb_img(:,:,1)...
   +.5870*rgb_img(:,:,2)...
   +.1140*rgb_img(:,:,3); %the NTSC standard
min(Y(:))
max(Y(:))
figure; colormap(gray(256)); image(Y)
```

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

Y: Luminance

I :Hue

Q: Saturation