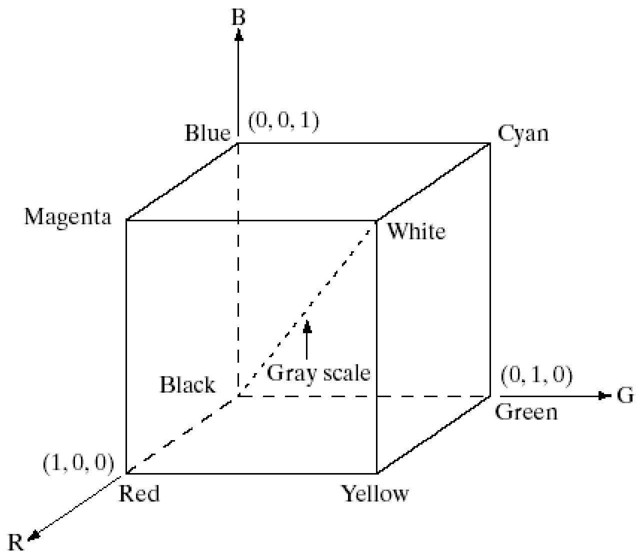
Color Image Processing II

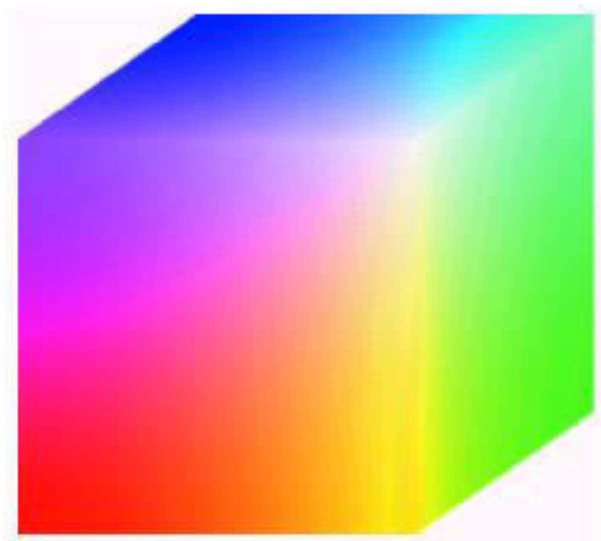
Outline

- Color fundamentals
- Color perception and color matching
- Color models
- Pseudo-color image processing
- Basics of full-color image processing
- Color transformations
- Smoothing and sharpening

Pixel depth

- Pixel depth: the number of bits used to represent each pixel in RGB space
- Full-color image: 24-bit RGB color image
  - (R, G, B) = (8 bits, 8 bits, 8 bits)





Safe RGB colors

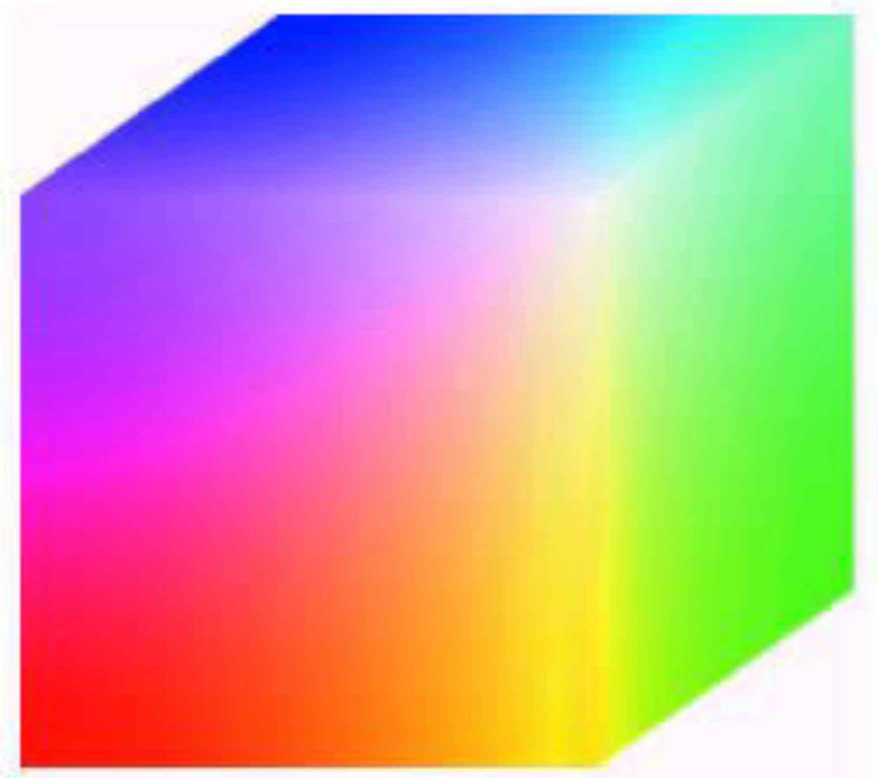
- Subset of colors is enough for some application
- Safe RGB colors (safe Web colors, safe browser colors)

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

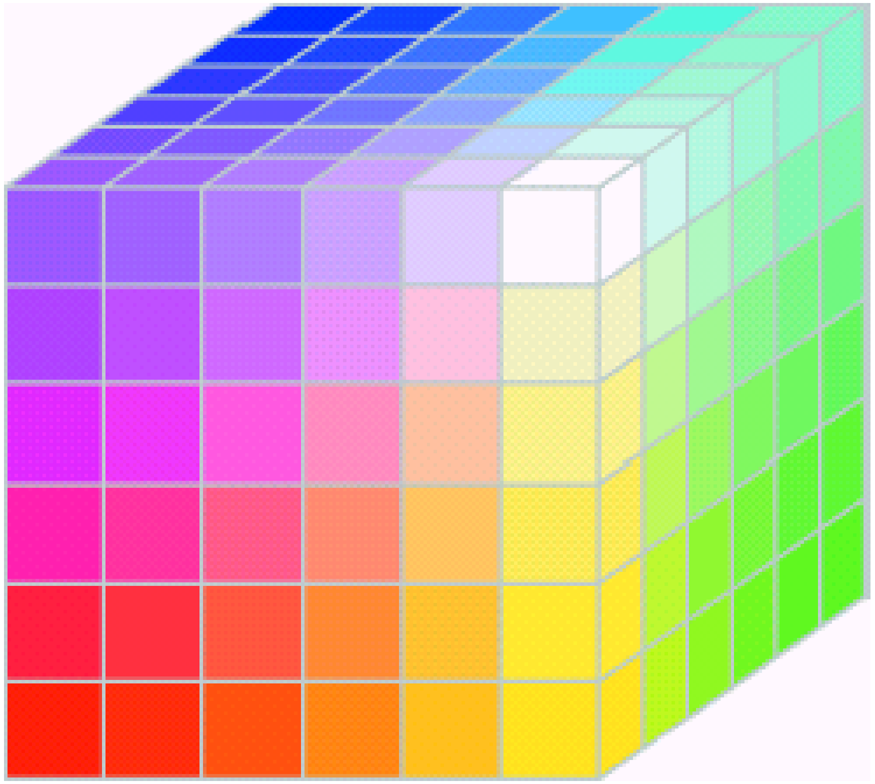
TABLE 6.1 Valid values of each RGB component in a safe color.

$(6)^3 = 216$

Safe RGB color (cont.)



Full color cube

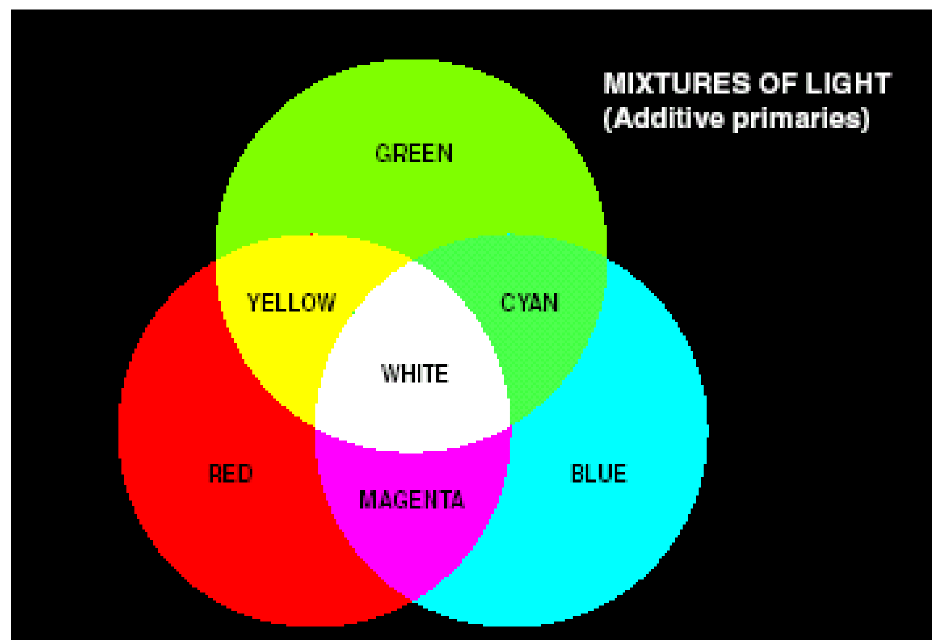


Safe color cube

CMY model (+Black = CMYK)

- CMY: secondary colors of light, or primary colors of pigments
- Used to generate hardcopy output

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



Why black ink is used?

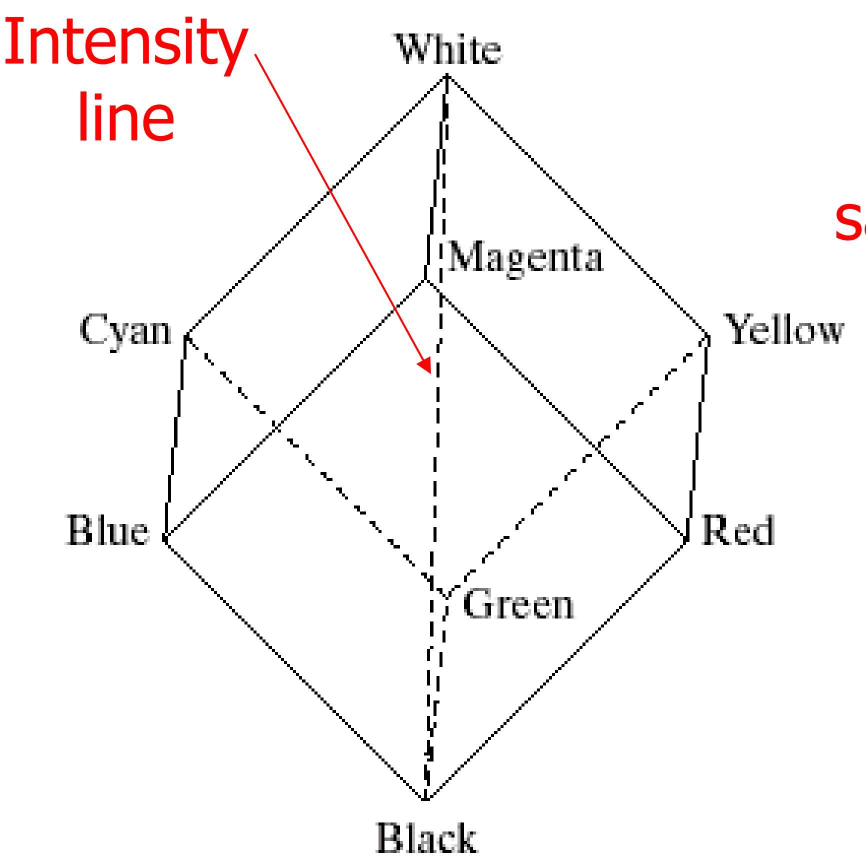
http://en.wikipedia.org/wiki/CMYK

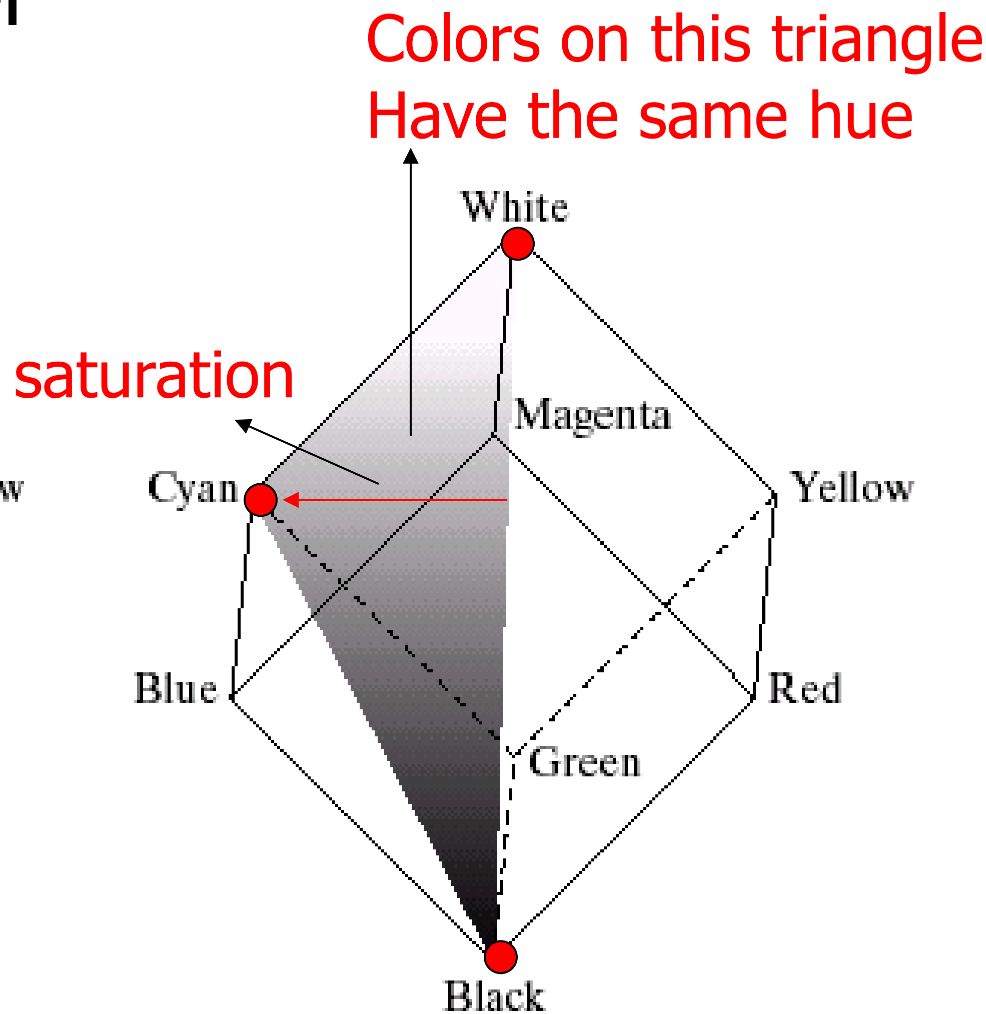
HSI color model

- Can you describe a color precisely using its R, G, B components?
- Human describe a color by its hue, saturation, and brightness
  - Hue 色度: color attribute
  - Saturation: purity of color (white->0, primary color->1)
  - Brightness: achromatic notion of intensity

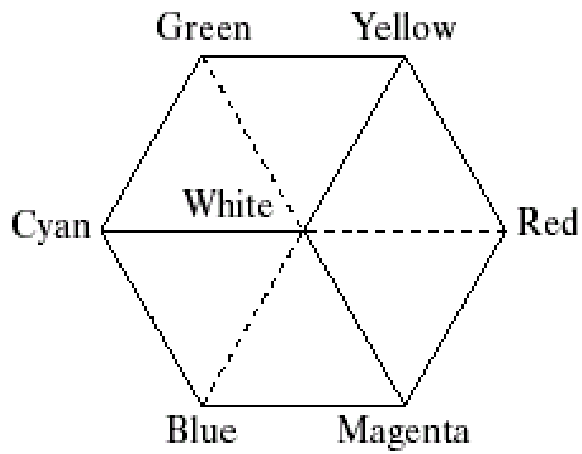
HSI color model (cont.)

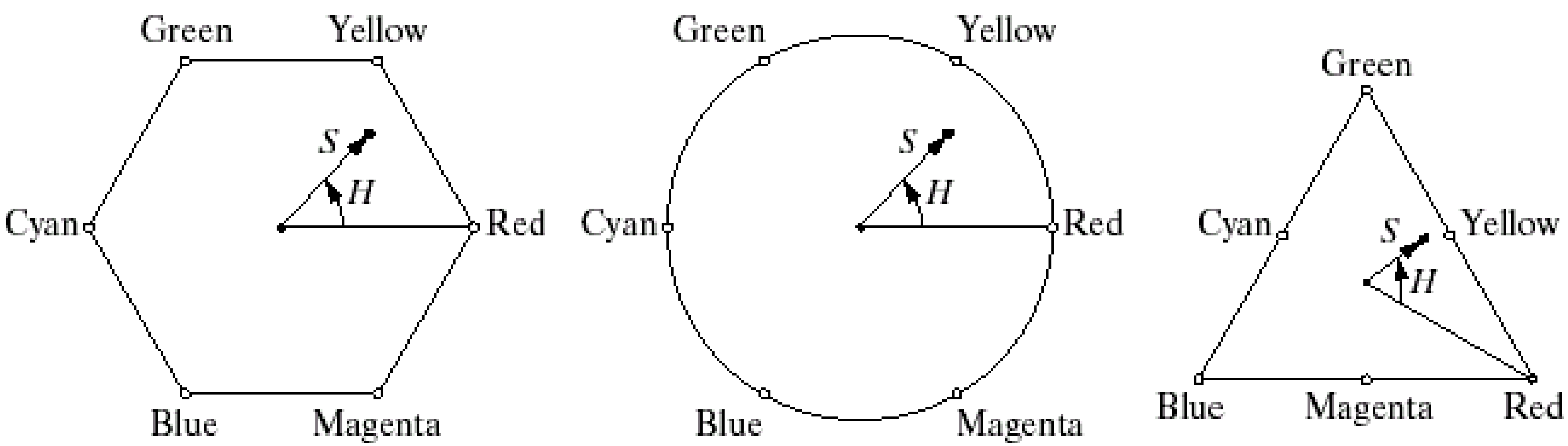
- RGB -> HSI model



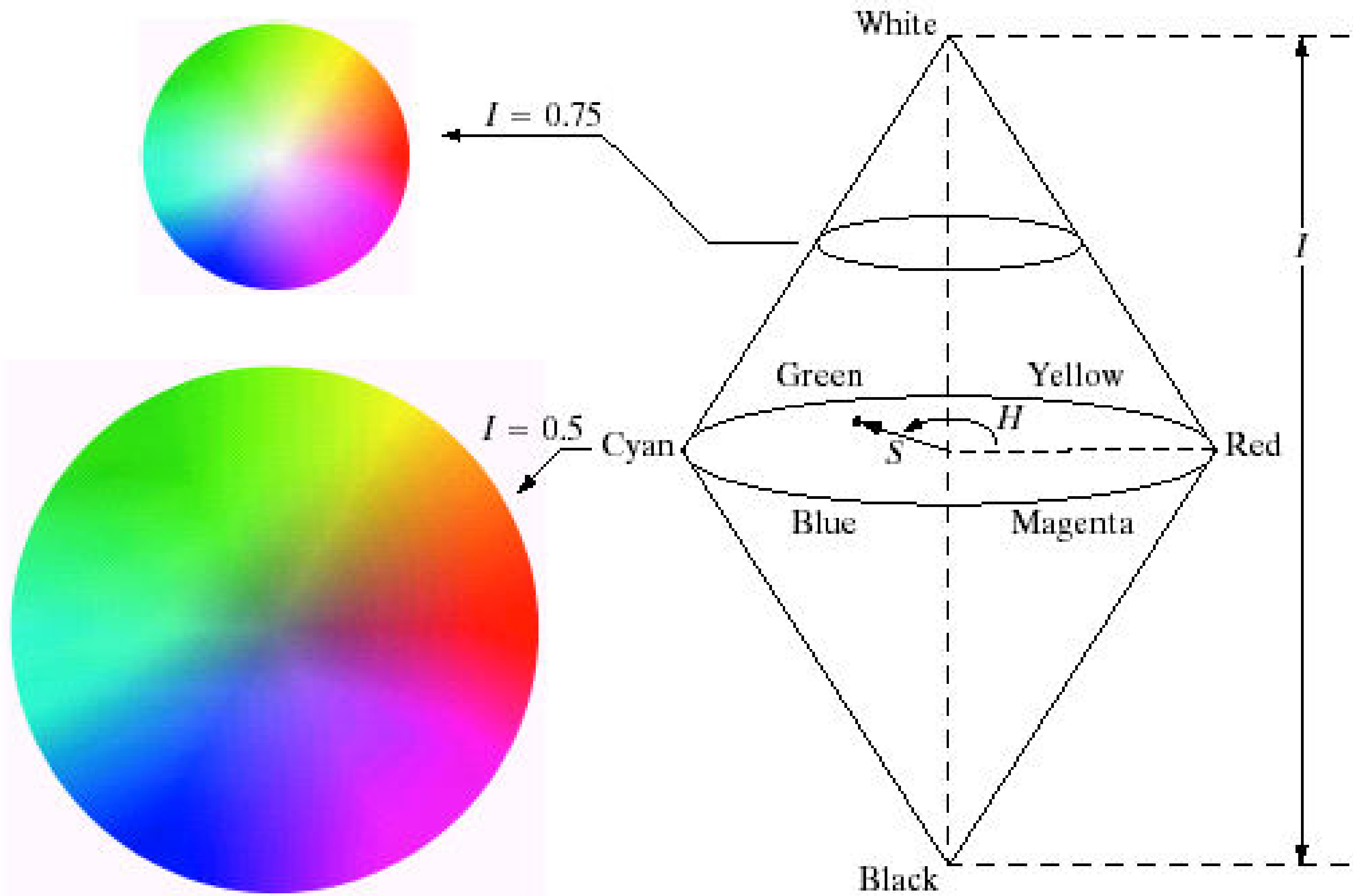


HSI model: hue and saturation



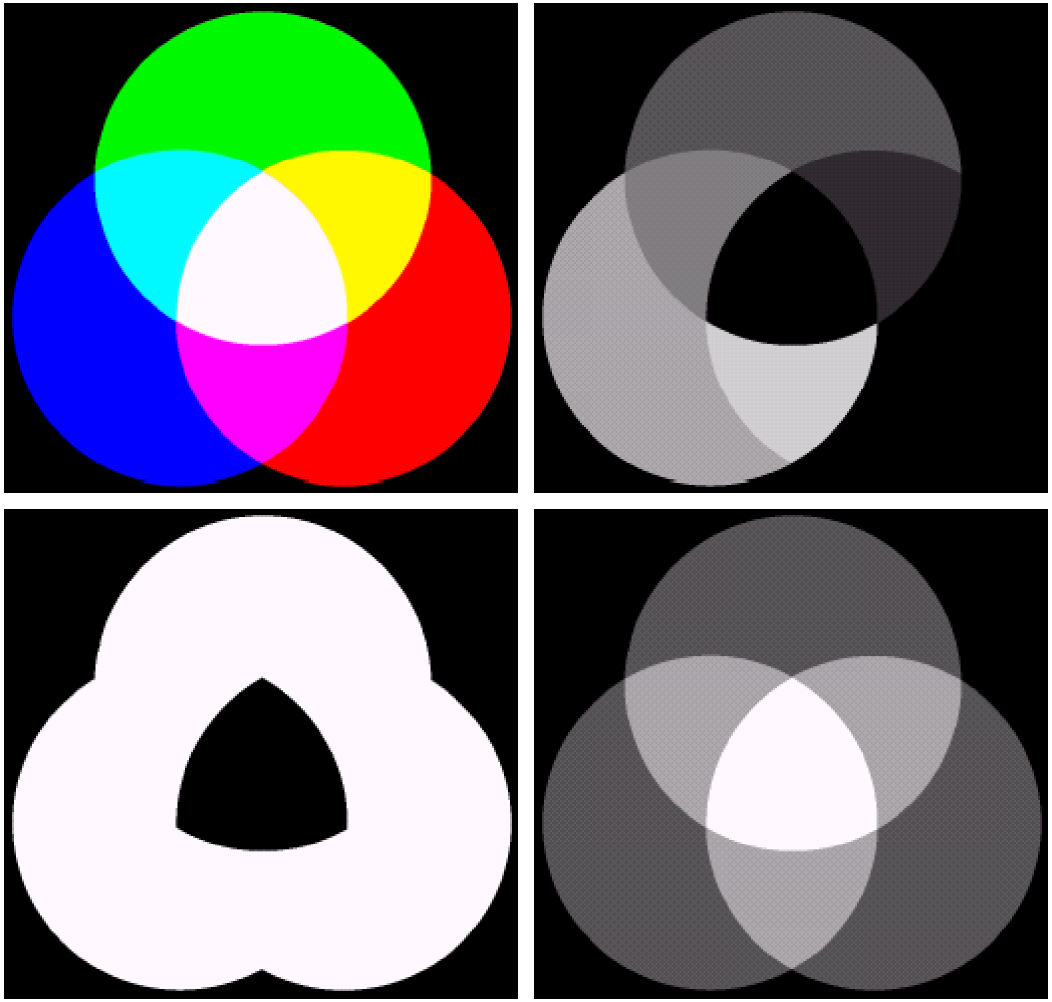


HSI model



HSI component images

R,G,B



Hue

saturation

intensity

Exercise#1: HSI

- x=imread('lily.tif');
- x(:,:,1) is R component
- x(:,:,2) is G component
- x(:,:,3) is B component

Exercise:

1. Apply color transform, rgb2hsv, show the H, S, V component
2. Apply rgb2hsv to your RGB circles image in exercise#2

Outline

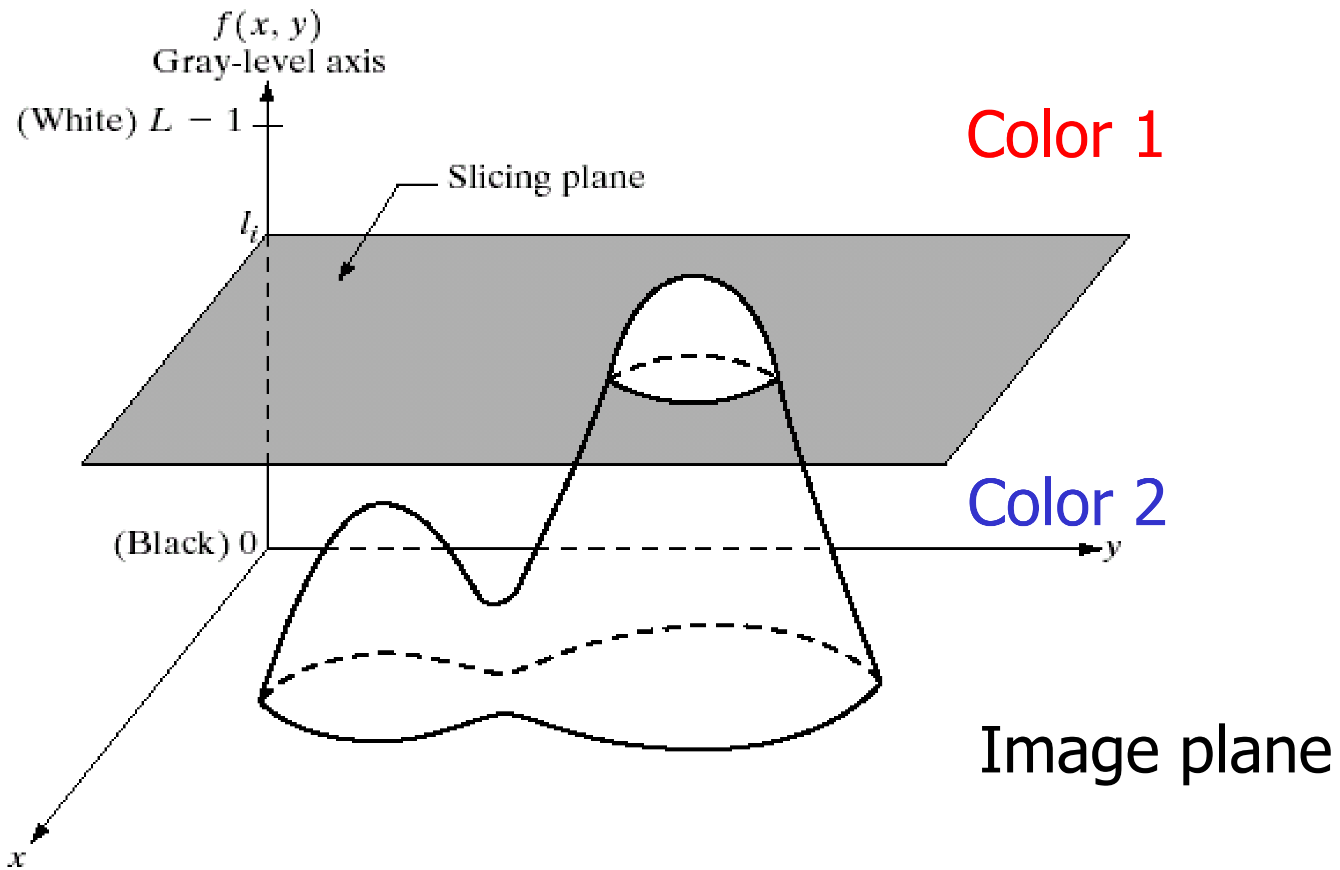
- Color fundamentals
- Color perception and color matching
- Color models
- Pseudo-color image processing
- Basics of full-color image processing
- Color transformations
- Smoothing and sharpening

Pseudo-color image processing

- Assign colors to gray values based on a specified criterion
- For human visualization and interpretation of gray-scale events
- Intensity slicing
- Gray level to color transformations

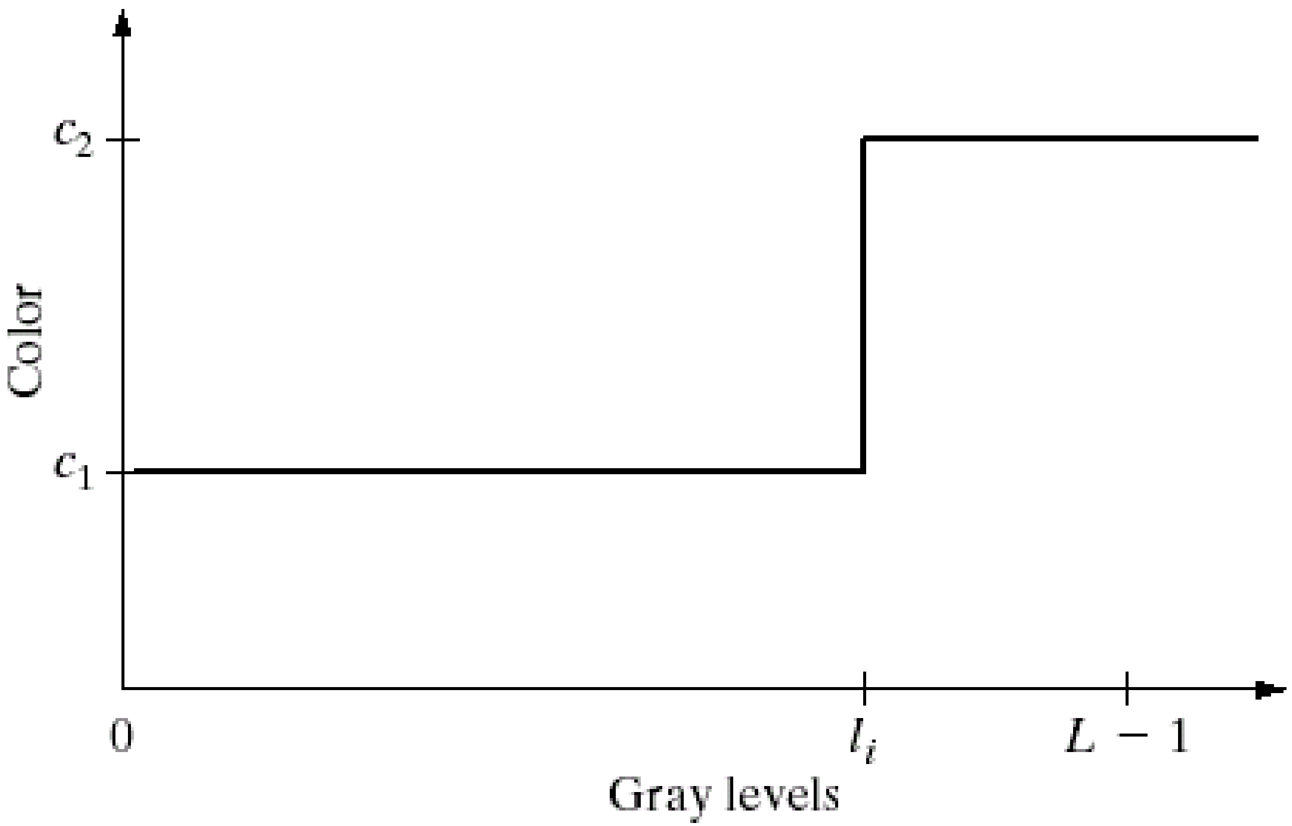
Intensity slicing

- 3-D view of intensity image



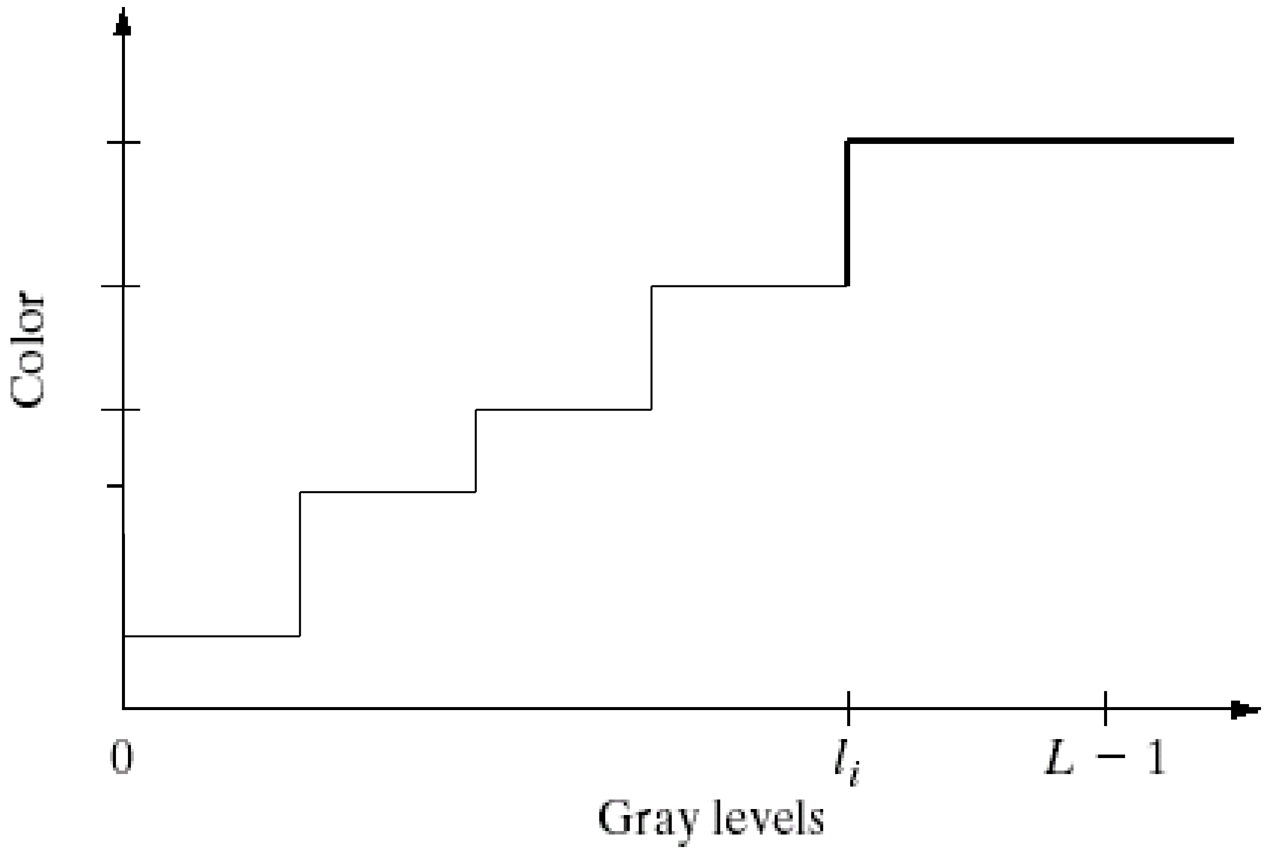
Intensity slicing (cont.)

- Alternative representation of intensity slicing

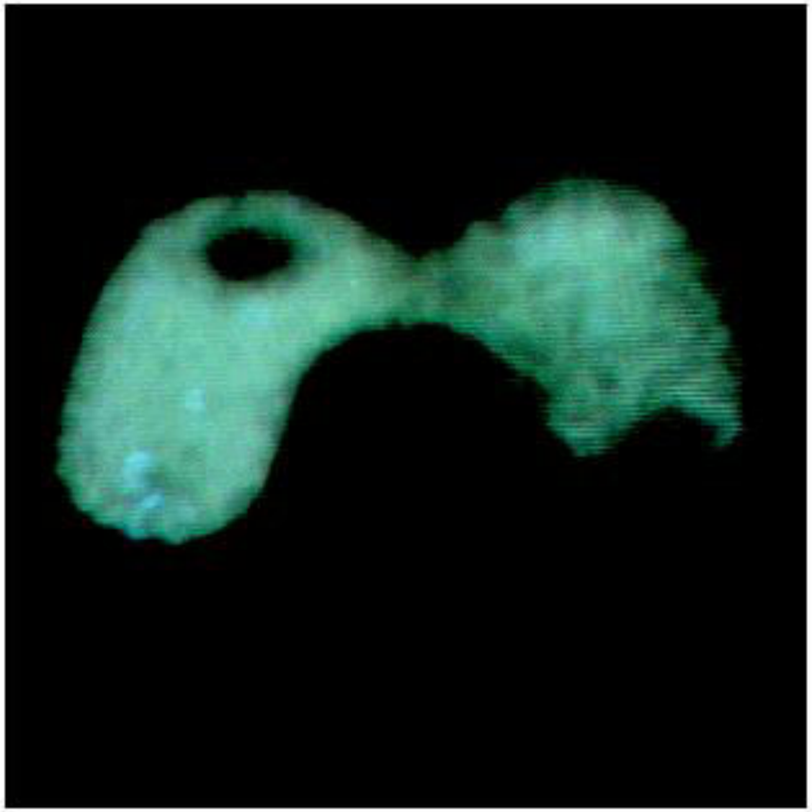


Intensity slicing (cont.)

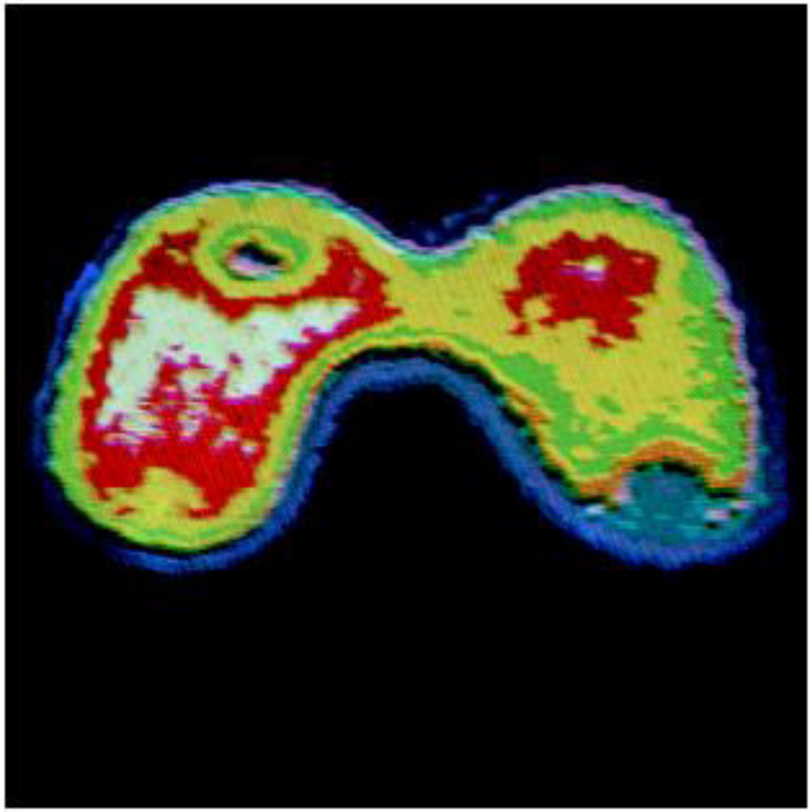
- More slicing plane, more colors



Application 1



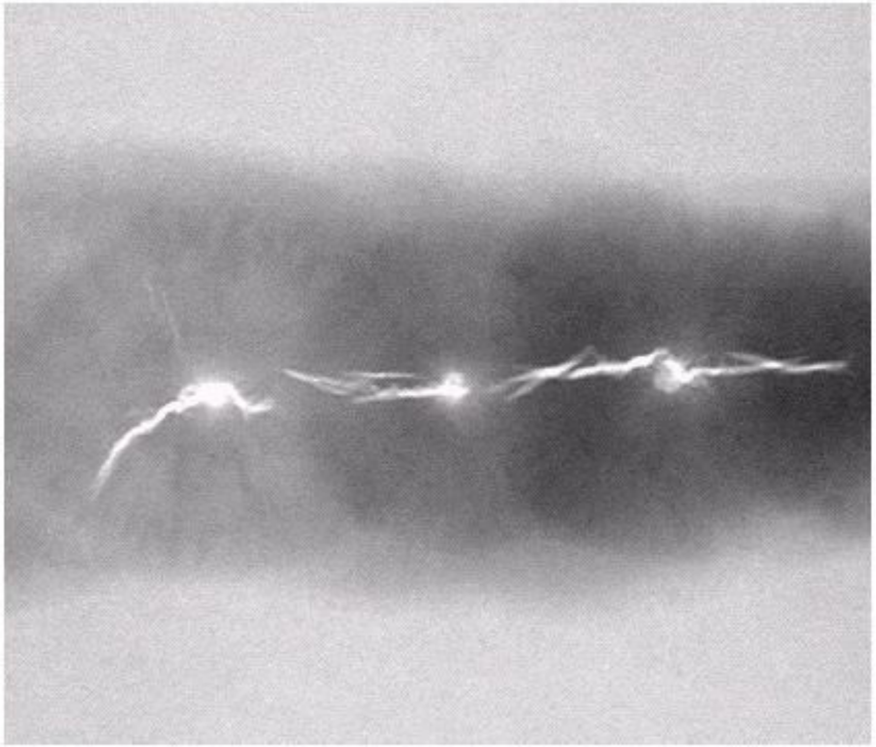
Radiation test pattern

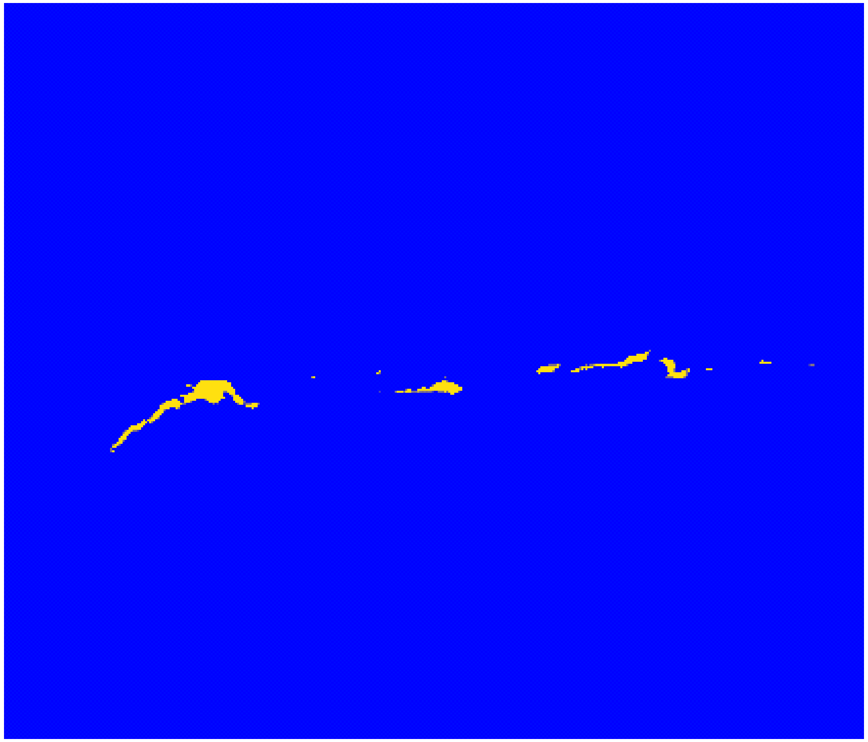


8 color regions

* See the gradual gray-level changes

Application 2

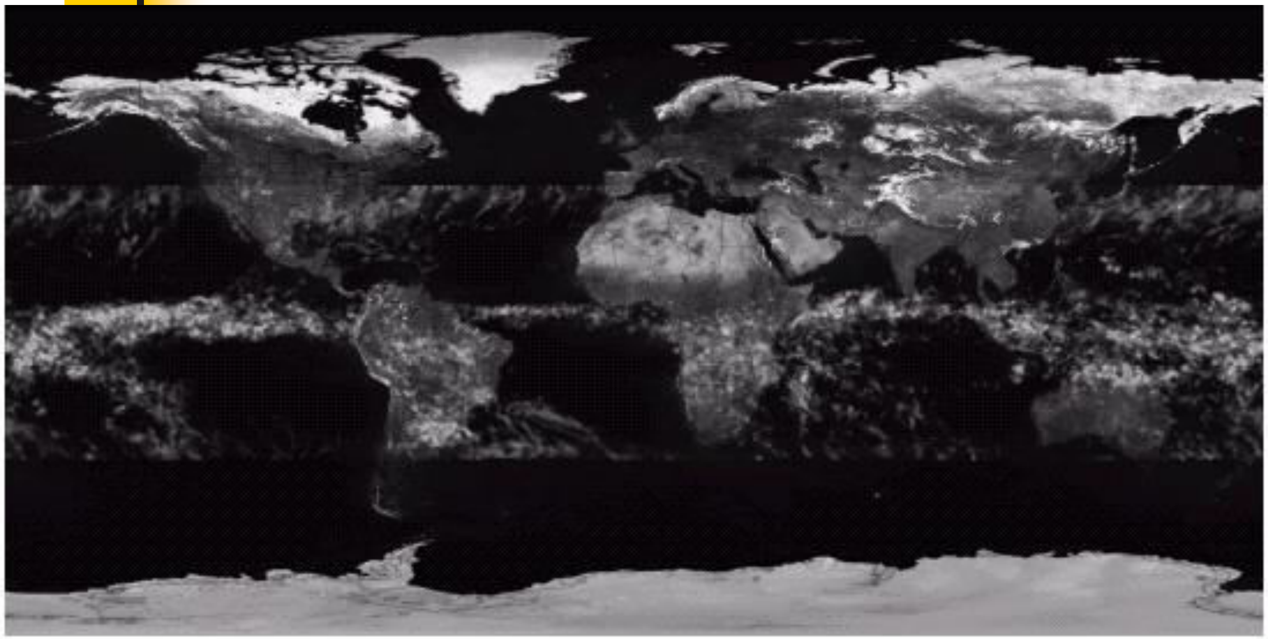




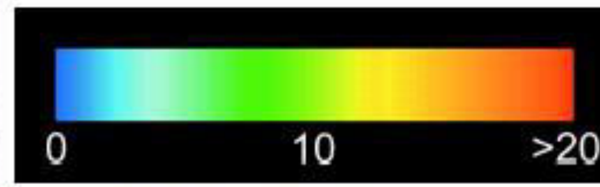
X-ray image of a weld

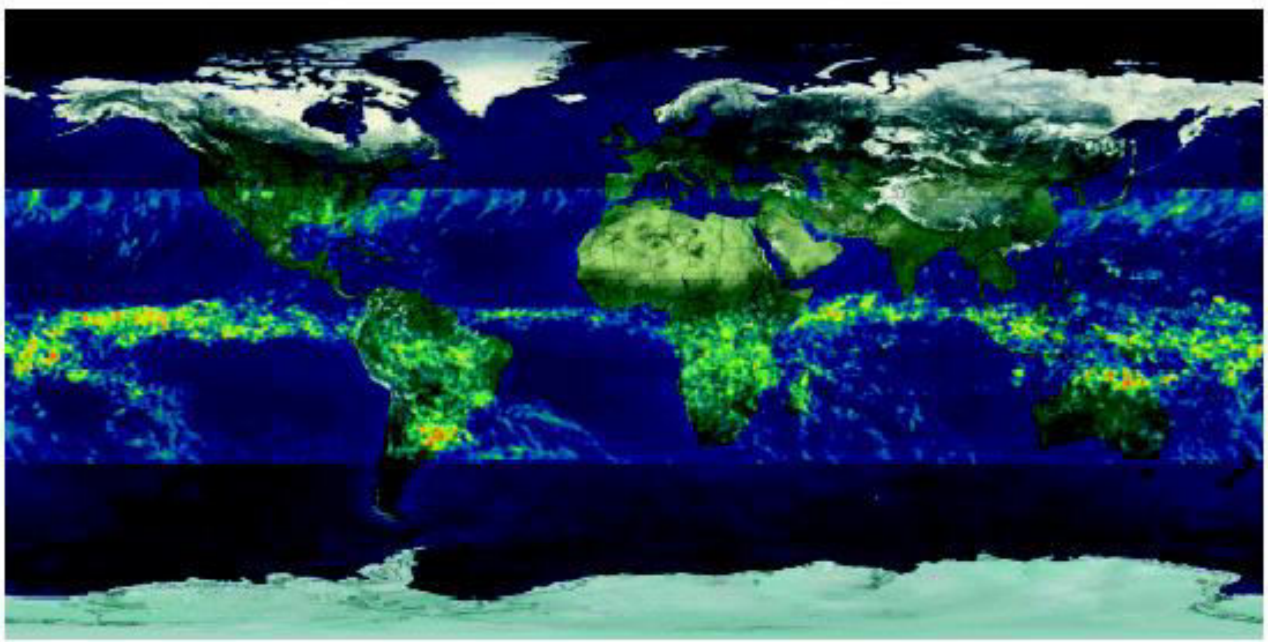
焊接物

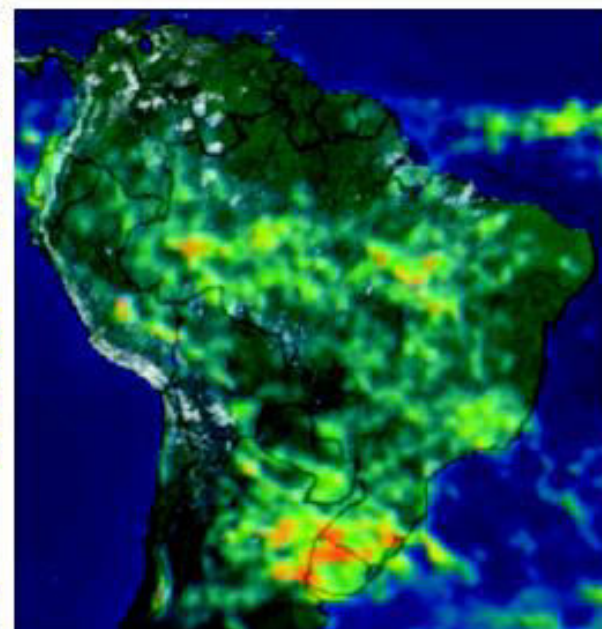
Application 3



Rainfall statistics







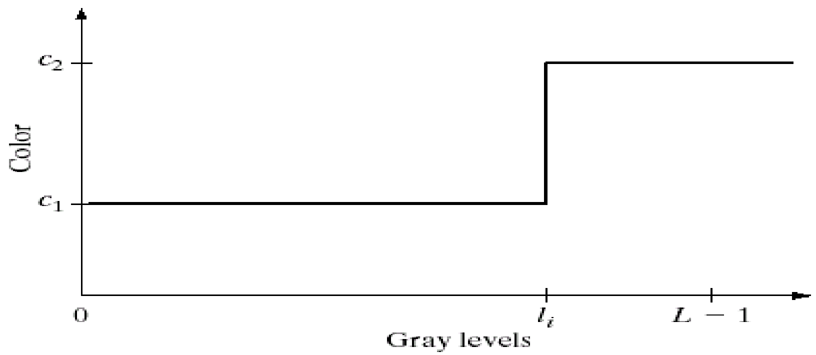
Exercise#2: Gray to color transformations

- b=imread(‘blocks.tif’);
- imshow(b, colormap(jet(256)));
- colorbar

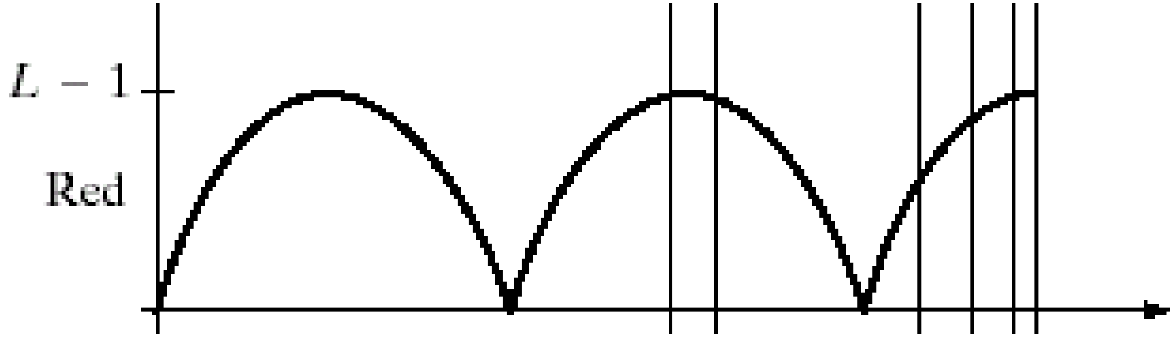
Exercise: Try any other 2 colormaps
See doc colormap

Gray level to color transformation

- Intensity slicing: piecewise linear transformation



- General Gray level to color transformation



Gray level to color transformation

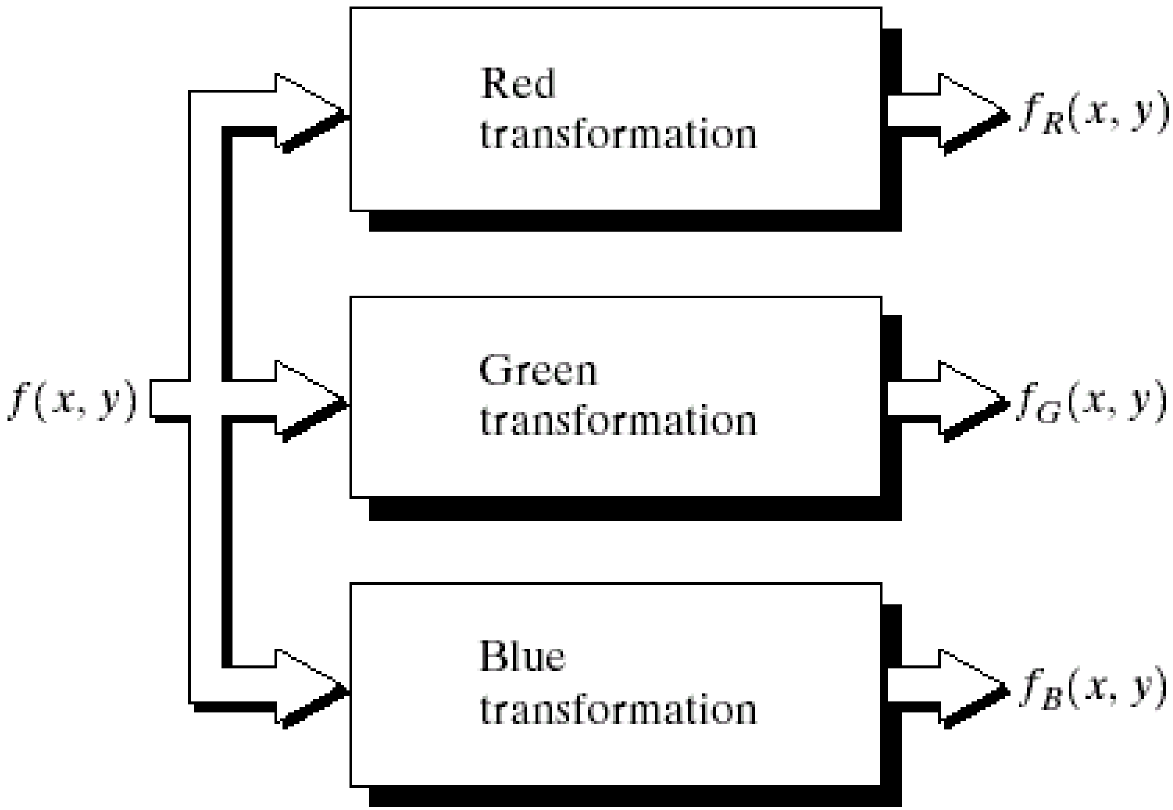
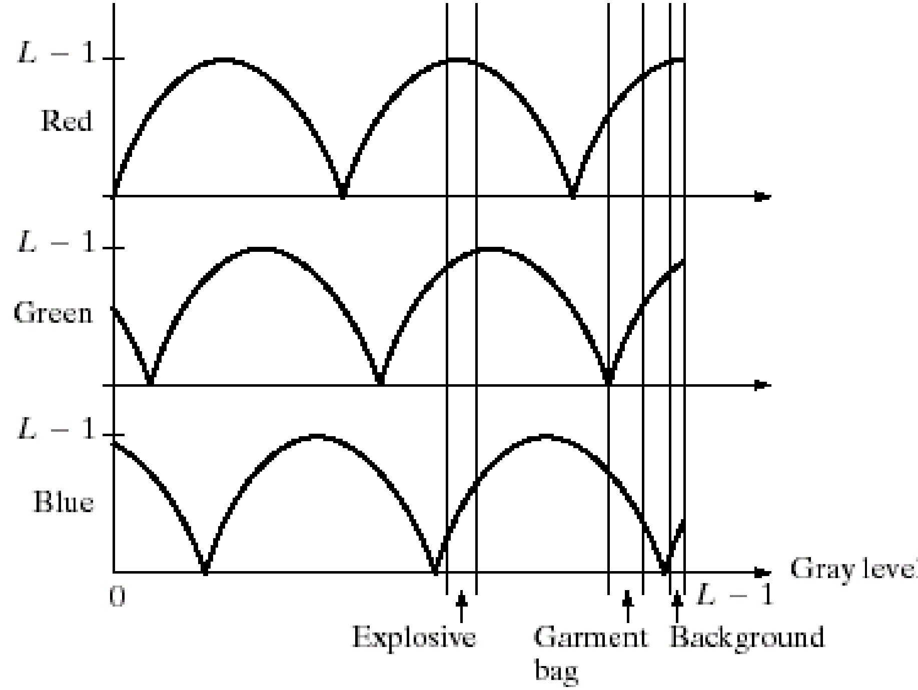
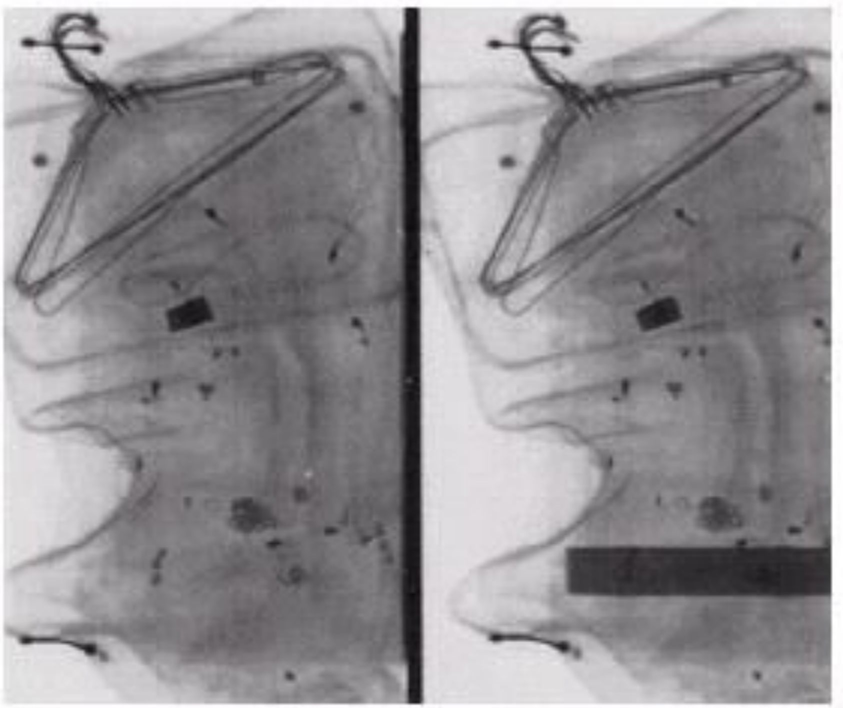
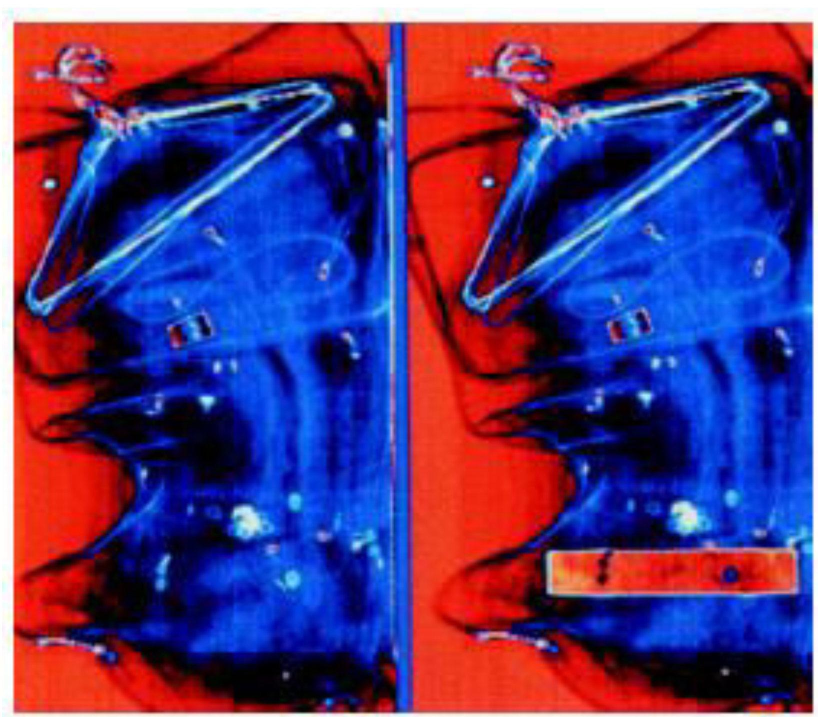


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Application 1

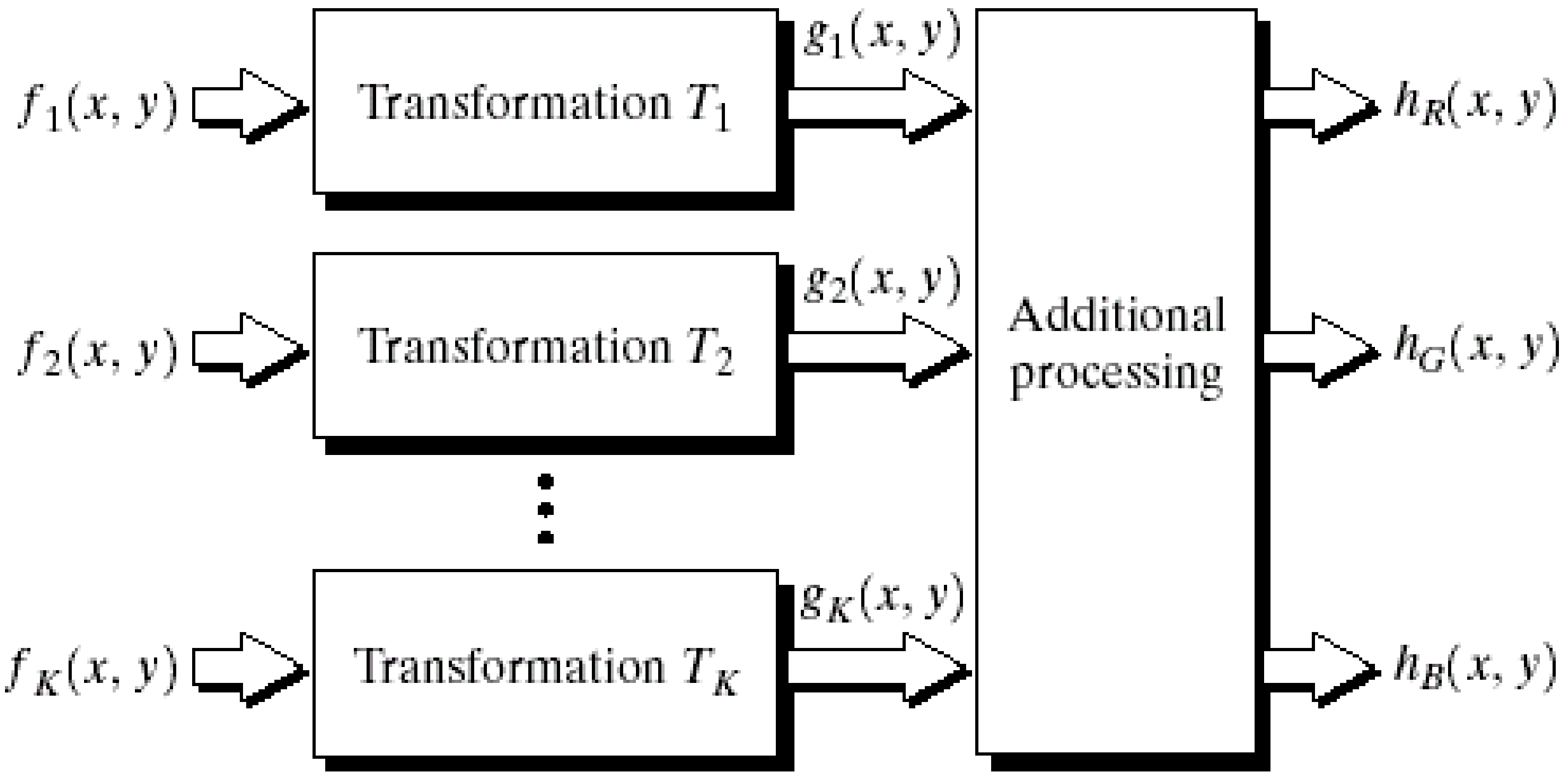




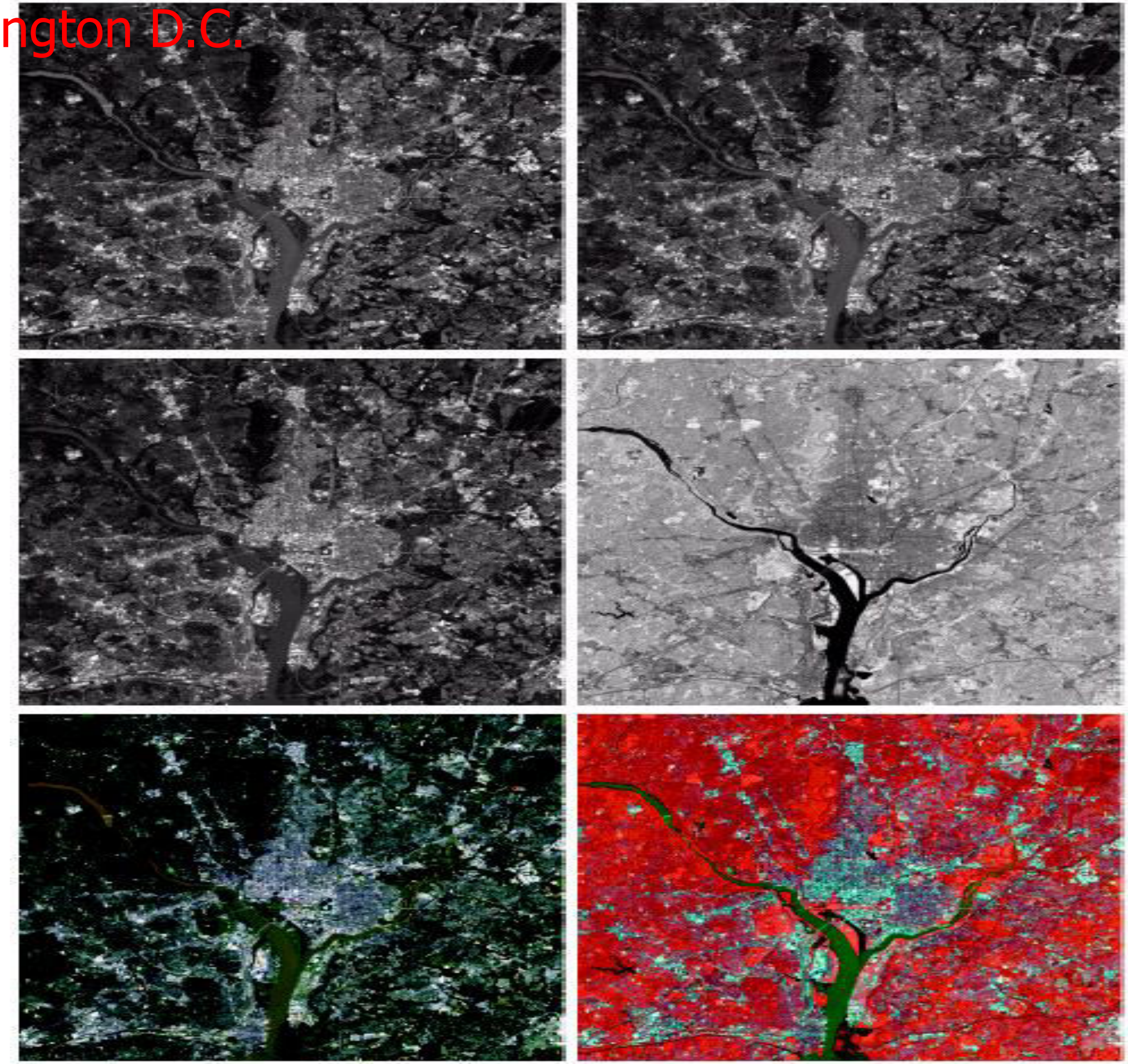


Combine several monochrome images

Example: multi-spectral images

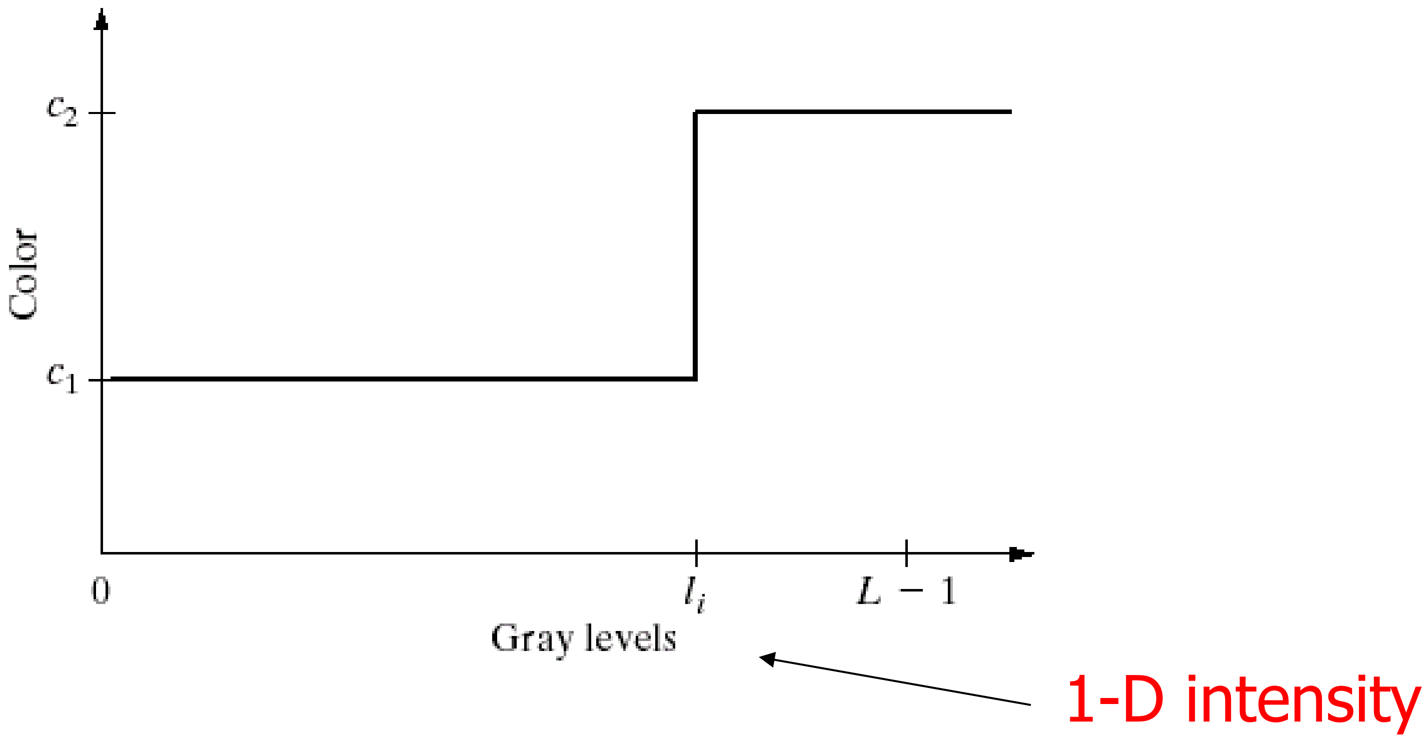


Washington D.C.



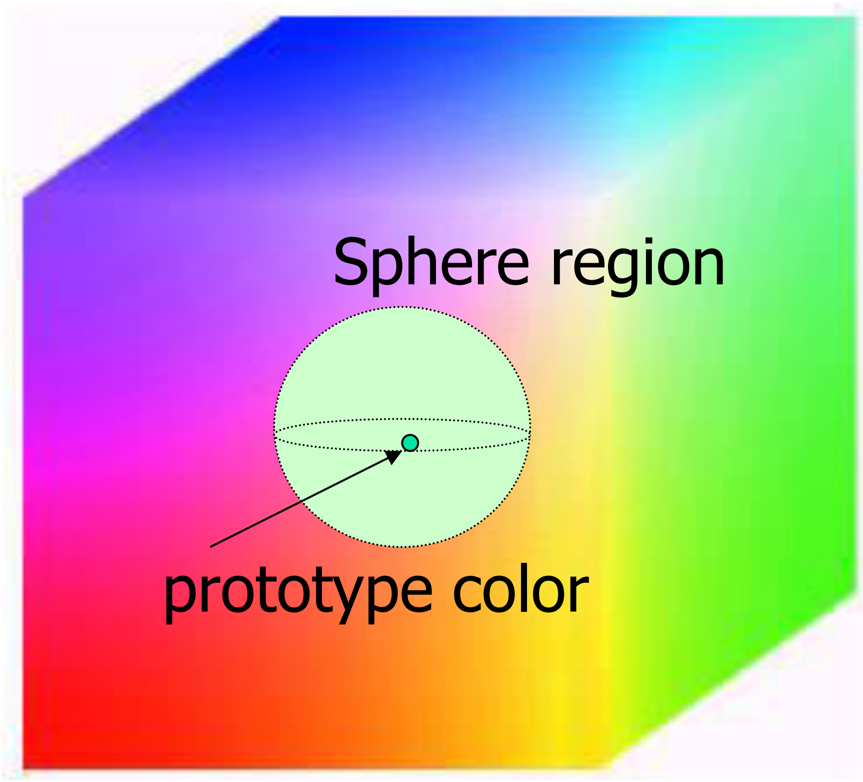
Color slicing

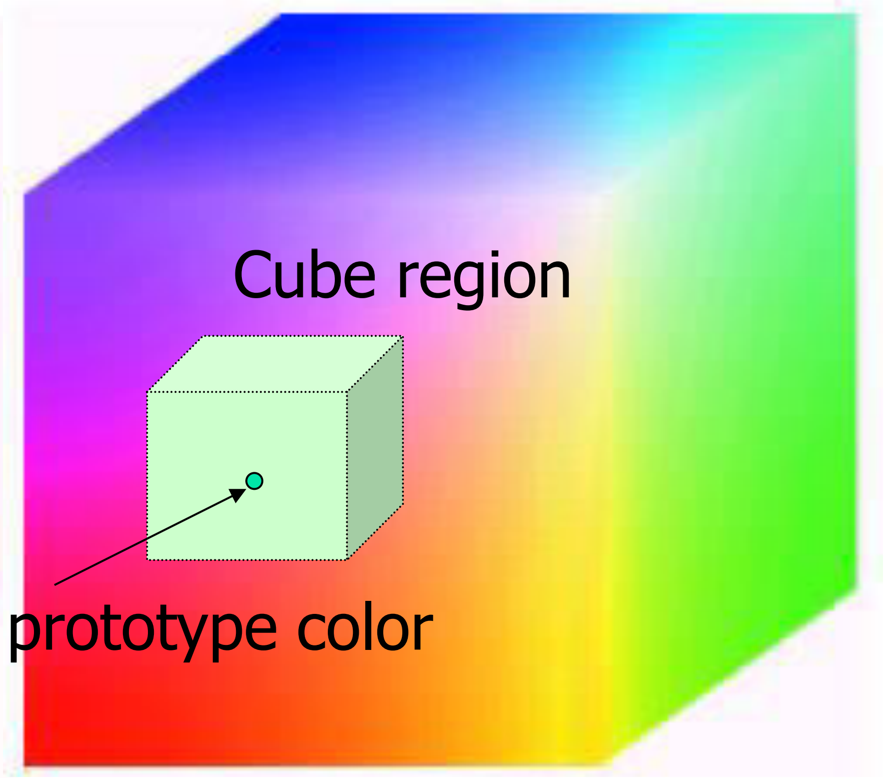
- Recall the pseudo-color intensity slicing



Color slicing

- How to take a region of colors of interest?





Application

Full color

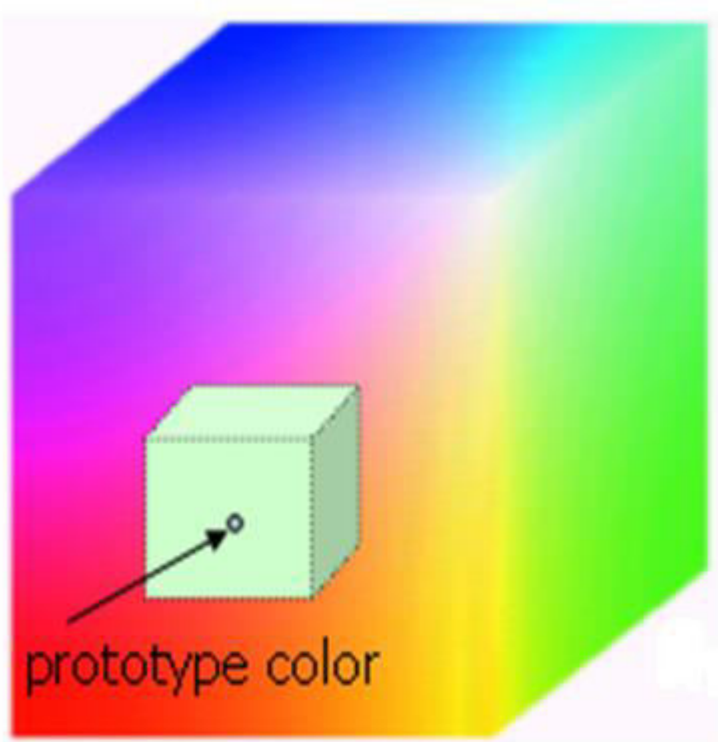
cube

sphere

Matlab: Color slicing

- Exercise#3: Get strawberry image

- Perform the color slicing using the cubic region. Cubic slicing is shown as follows:



Cube region

Prototype color (a_1, a_2, a_3)
$= (175, 41, 49)$

Original pixel's color (r_1, r_2, r_3)

New pixel's color (s_1, s_2, s_3)

$(s_1, s_2, s_3) = (r_1, r_2, r_3)$

if $|r_1 - a_1| < w$ & $|r_2 - a_2| < w$ & $|r_3 - a_3| < w$

$(s_1, s_2, s_3) = (127, 127, 127)$ otherwise

Outline

- Color fundamentals
- Color perception and color matching
- Color models
- Pseudo-color image processing
- Basics of full-color image processing
- Color transformations
- Smoothing and sharpening

Color pixel

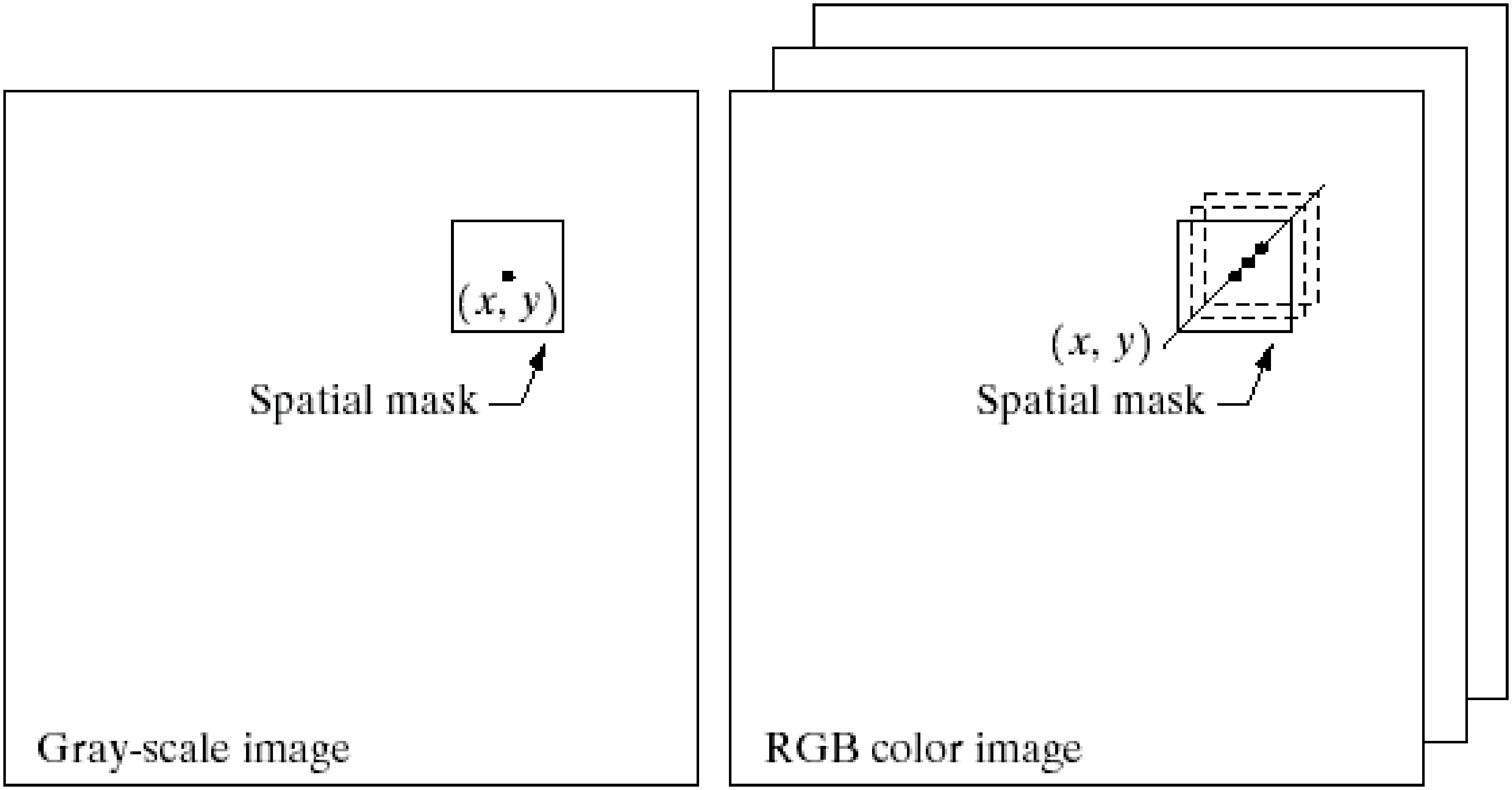
- A pixel at (x,y) is a vector in the color space
  - RGB color space

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

c.f. gray-scale image

$$f(x,y) = I(x,y)$$

Example: spatial mask



How to deal with color vector?

- Per-color-component processing
  - Process each color component
- Vector-based processing
  - Process the color vector of each pixel
- When can the above methods be equivalent?
  - Process can be applied to both scalars and vectors
  - Operation on each component of a vector must be independent of the other component

Two spatial processing categories

- Similar to gray scale processing studied before, we have to major categories
  - Pixel-wise processing: color transformation
  - Neighborhood processing: smoothing and sharpening filtering

Outline

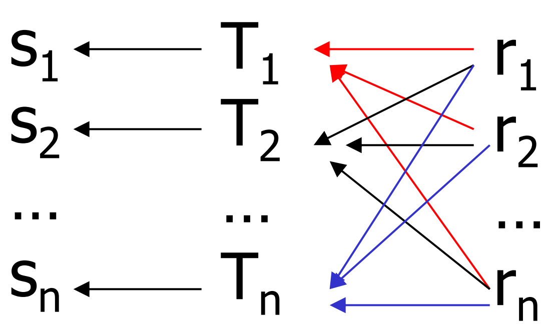
- Color fundamentals
- Color perception and color matching
- Color models
- Pseudo-color image processing
- Basics of full-color image processing
- Color transformations
- Smoothing and sharpening

Color transformation

- Similar to gray scale transformation
  - g(x,y)=T[f(x,y)]
- Color transformation

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

output vector input vector



Use which color model in color transformation?

- RGB ⇔CMY(K) ⇔ HSI
- Theoretically, any transformation can be performed in any color model
- Practically, some operations are better suited to specific color model

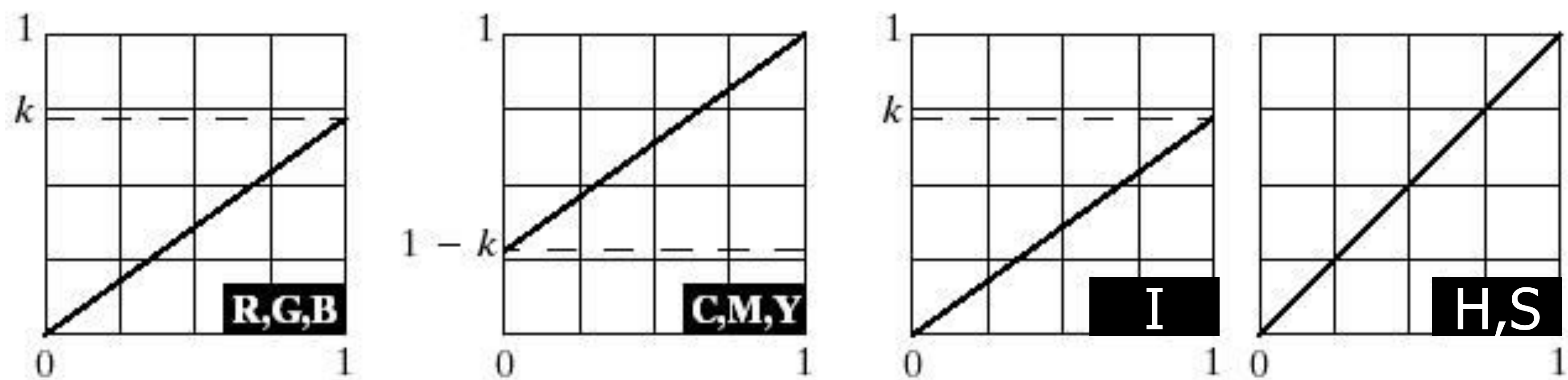
Example: modify intensity of a color image

- **Example:** $g(x,y)=k\,f(x,y)$, $0<k<1$
- HSI color space
  - Intensity: $s_3 = k\,r_3$
- RGB color space
  - For each R,G,B component: $s_i = k\,r_i$
- CMY color space
  - For each C,M,Y component: $s_i = k\,r_i + (1-k)$

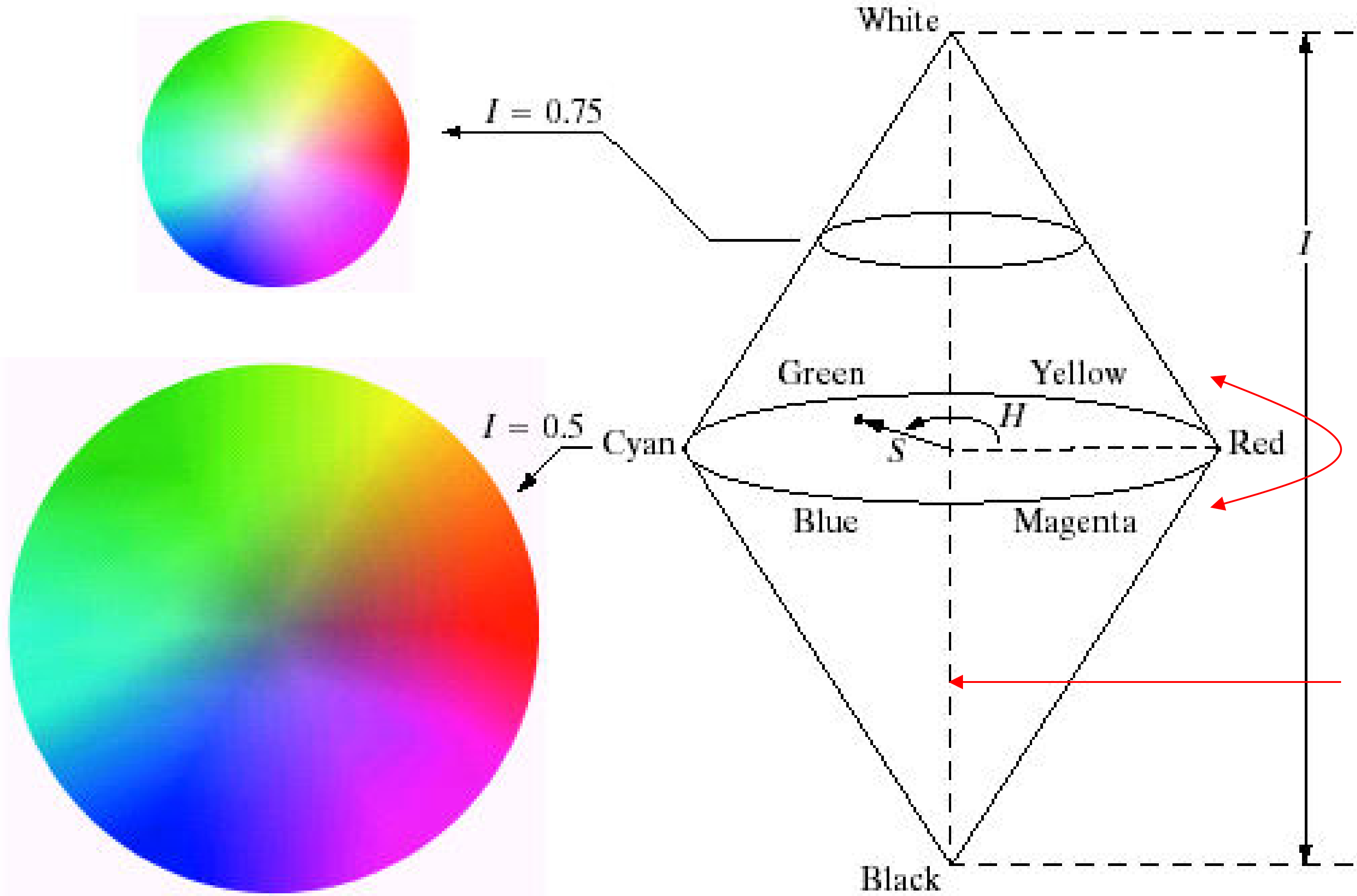
Processing results are the same. Which operations is the fastest?

Original image

k=0.7 (reduce intensity)



Problem of using Hue component



dis-continuous

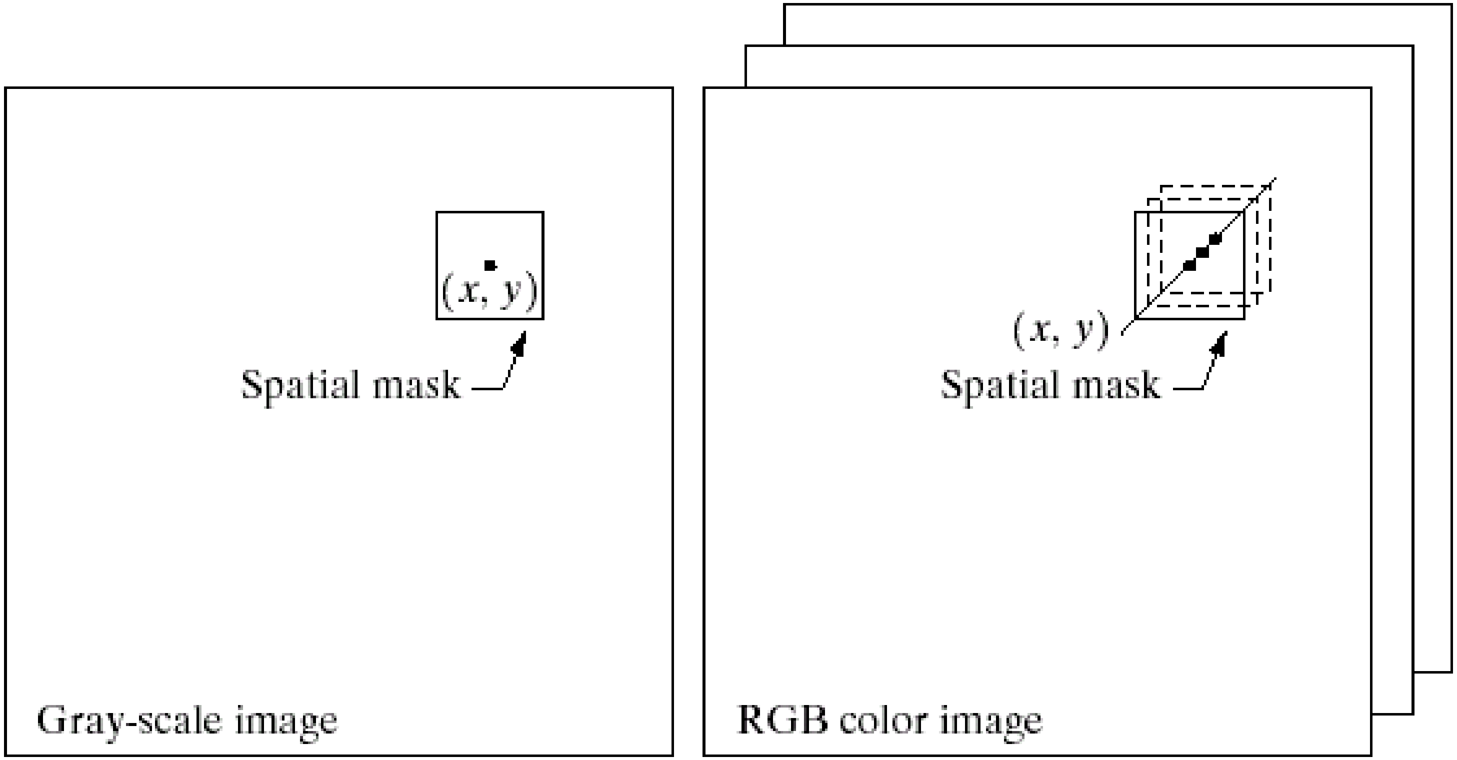
Un-defined over gray axis

Outline

- Color fundamentals
- Color perception and color matching
- Color models
- Pseudo-color image processing
- Basics of full-color image processing
- Color transformations
- Smoothing and sharpening

Color image smoothing

- Neighborhood processing



Color image smoothing: averaging mask

$$\bar{\mathbf{c}}(x, y) = \frac{1}{K} \sum_{(x,y)\in S_{xy}} \mathbf{c}(x, y)$$

vector processing

Neighborhood Centered at (x,y)

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

per-component processing

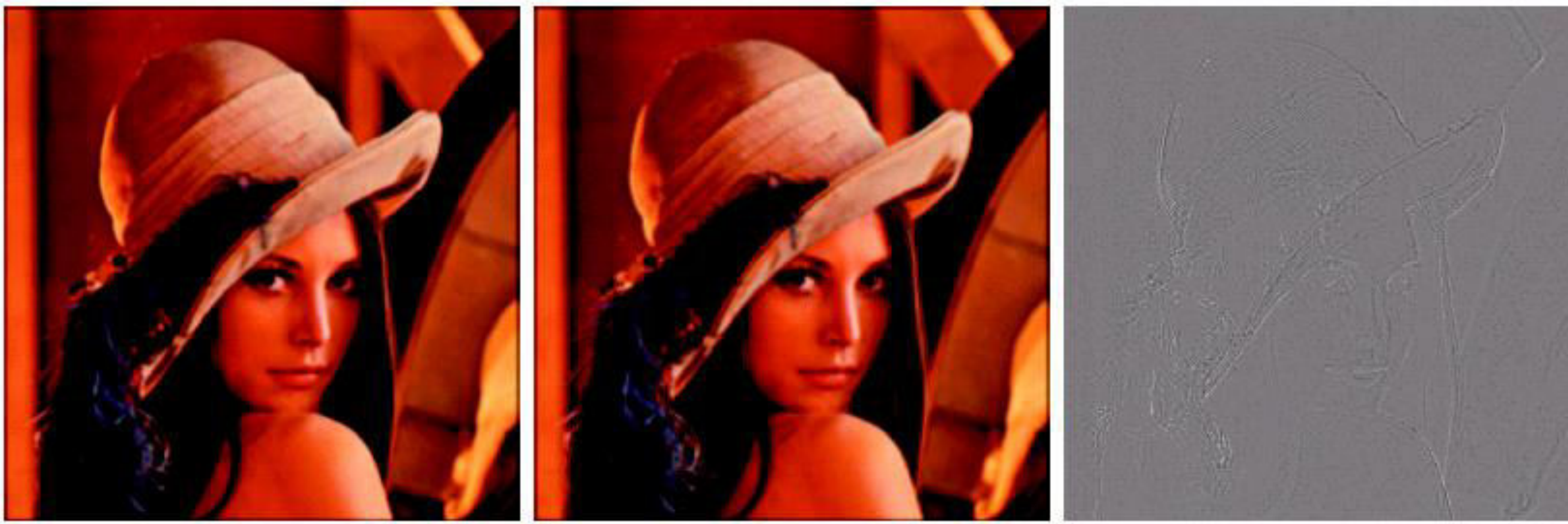

original
R
G
G
H
S
I

Example: 5x5 smoothing mask

Smooth each component in RGB model

Smooth I in HSI model

difference



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.

Exercise#4: Smoothing color image

- Download lena_RGB.tif
- Smoothing with 10x10 average filter in
  - RGB domain
  - HSI domain (smoothing intensity only)