6.4 Basics of Full-Color Image Processing(1)

- Two approaches for full-color image processing
  (1) Processing each component image individually
  (2) Directly processing color pixels with three components

- Color pixels are vectors in direct processing
  - For example, in the RGB color space

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

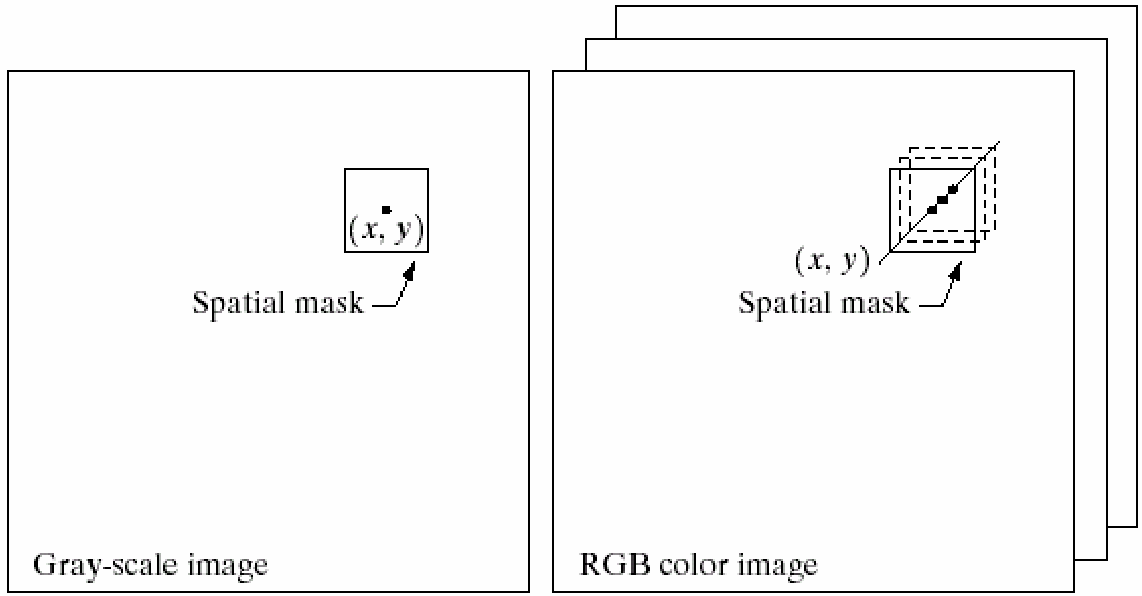
where, ***c*** : an arbitrary vector in RGB space
c(x,y) : a function of coordinate (x,y)

◇ Basics of Full-Color Image Processing(2)

- Processing for spatial mask
  - Fig. 6.29(a) : per-color-component processing
  - Fig. 6.29(b) : vector-based processing

a b

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



(a) Gray-scale image (b) RGB color image

6.5 Color Transformations

- Formulation
- Color Complements
- Color Slicing
- Tone and Color Correction
- Histogram Processing

6.5.1 Formulation (1)

- Color transformation similar to the gray-level transformation in chap. 3

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

where, f(x,y) : color input image
g(x,y) : transformed color output image
T : operator of *f*

- Color transformation form

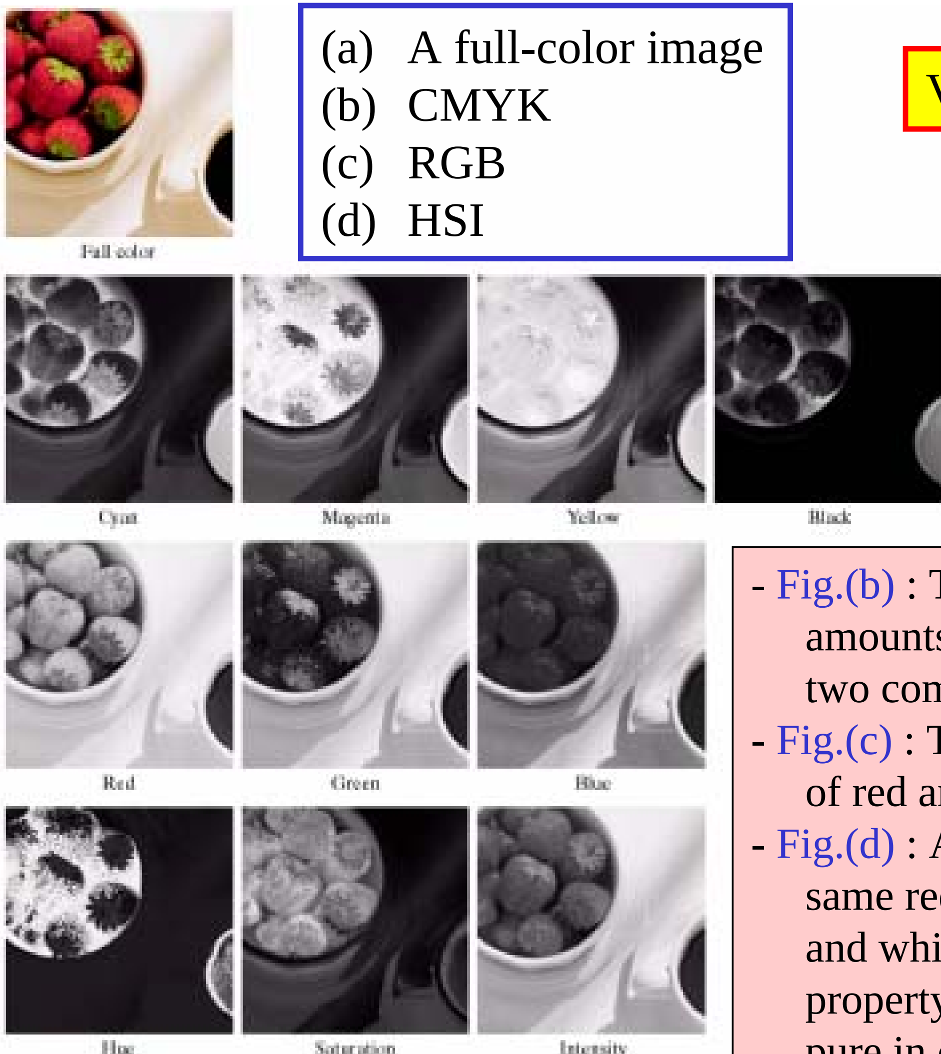
$$s_i = T_i(r_1, r_2, ..., r_n), \quad i = 1, 2, ..., n$$

where, s_i and r_i : normalized color component of *g(x,y)* and *f(x,y)* , respectively
T_i : transformation or color mapping functions

- Examples :

- RGB space : $n = 3$, r_1 = red, r_2 = green, r_3 = blue
- CMYK space : $n = 4$, r_1 = cyan, r_2 = magenta, r_3 = yellow, r_4 = black
- HSI space : $n = 3$, r_1 = hue, r_2 =saturation, r_3 = intensity

◇ Formulation (2)



(a) A full-color image
(b) CMYK
(c) RGB
(d) HSI

Various color-space components

- Fig.(b) : The strawberries are composed of large amounts of magenta and yellow because these two components are brightest
- Fig.(c) : The strawberries contain a large amount of red and very little green and blue
- Fig.(d) : Although the strawberries are almost same red, they have values near both black (0^o) and white (360^o) in hue because of discontinuity property of the hue. They are also relatively pure in color (highest saturation).

◇ Formulation (3)

Intensity modification using the various color spaces

- For example, modification of the intensity

$$g(x, y) = kf(x, y) \qquad 0 < k < 1$$

- In the HSI color space, one component must be transformed

$$s_i = r_i \qquad i = 1, 2$$

$$s_3 = kr_3 \qquad r_3 : \text{intensity component}$$

- In the RGB color space, three component must be transformed

$$s_i = kr_i, \qquad i = 1, 2, 3$$

- The CMY space requires a similar set of linear transformation

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

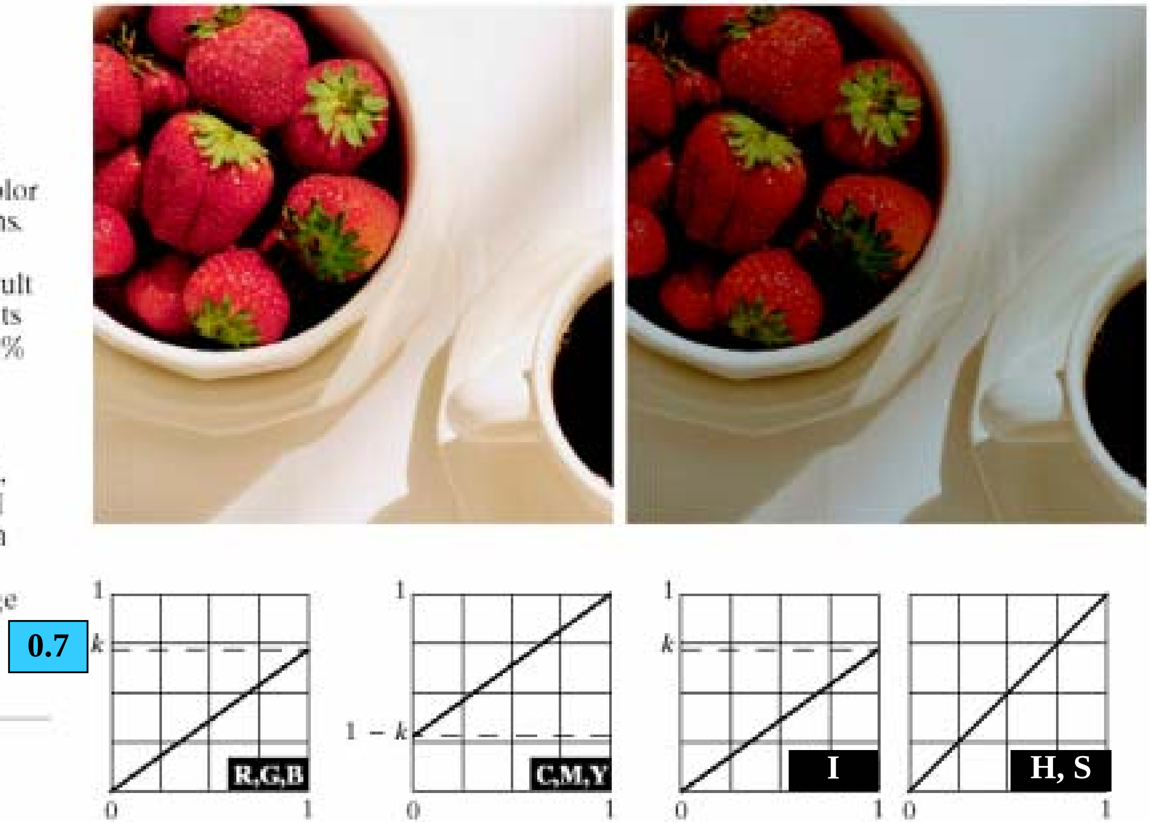
◇ Formulation (4)

Original Image

Transformed image

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.)



- Using k = 0.7, the intensity is transformed
- Although the color spaces are different, the output image is the same

6.5.2 Color Complements (1)

- The hues directly opposite one another on the circle are called *color complements*
- Color complements are useful for enhancing detail that is embedded in dark region of a color image-particularly when the regions are dominant in size

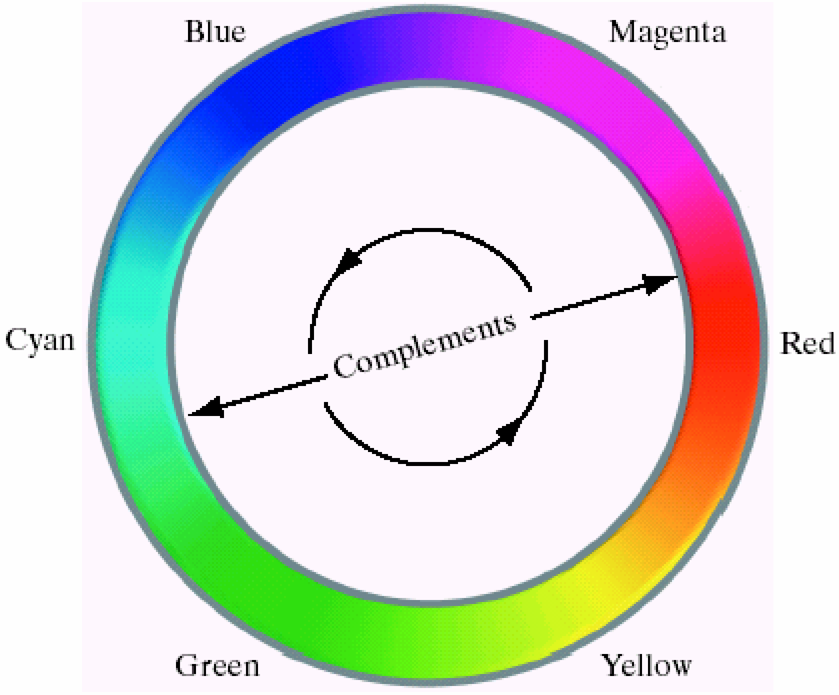
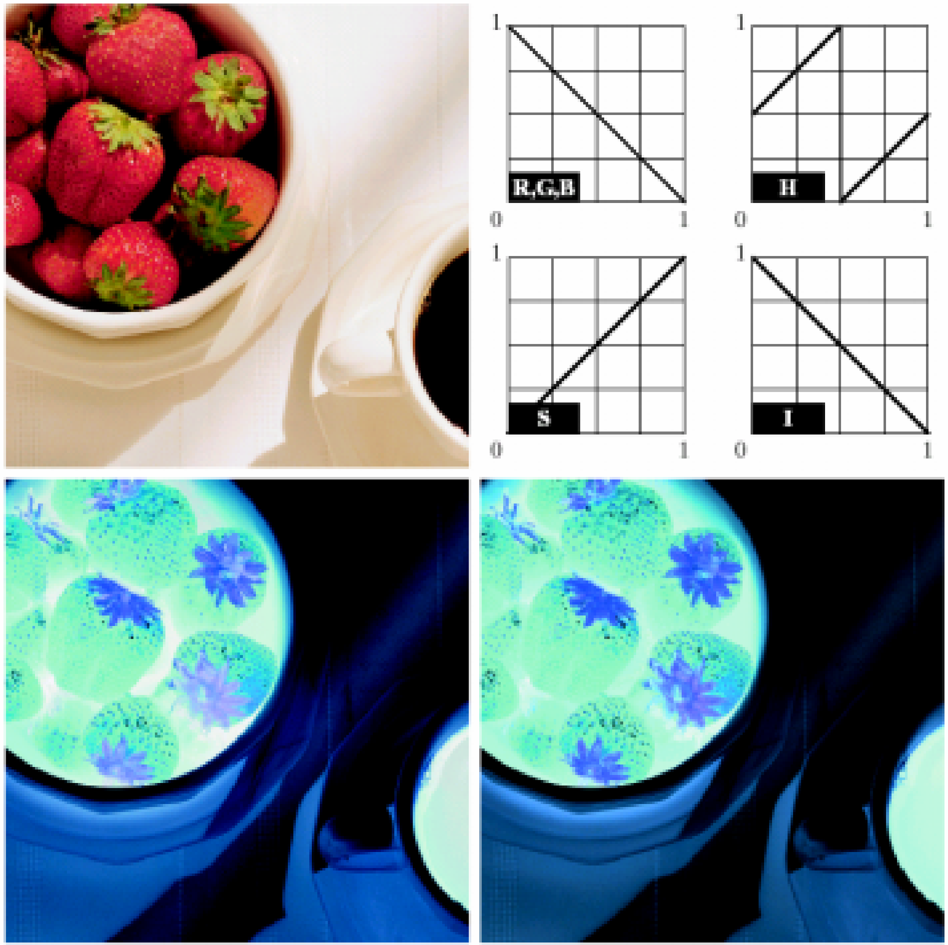


FIGURE 6.32 Complements on the color circle.

◇ Color Complements (2)

Color complement transformation



a b
c d

FIGURE 6.33

(a) Original image
(b) Transformation functions
(c) Complement based RGB transformation function
(d) An approximation of RGB complement using HIS transformations

6.5.3 Color Slicing(1)

- Highlighting a specific range of colors in an image is useful for separating objects from their surroundings.
- Basic idea
  - Display the colors of interest so that they stand out from background or
  - Use the region defined by the colors as a mask for further processing
- One way to slice a color image is to map the colors outside some range of interest to a nonprominent neutral color (s_i = 0.5)
  - If the colors of interest (prototypical color) are enclosed by a (hyper)cube,

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

a_j : center of interest color
W : width of interest color
i = 1, 2, …., n

  - If the colors of interest is specified by a (hyper)sphere

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

R_0: the radius of the sphere
i = 1, 2, …., n

6.5.3 Color Slicing(2)

Prototype red (0.6863, 0.1608, 0.1922) with RGB color coordinate

Cube of W = 0.2549

Sphere of R_0 = 0.1765

- Sphere-based transformation is slightly better in the sense that it includes more of the strawberries' red areas.
- A sphere of radius 0.1765 does not completely enclose a cube of width 0.2549 but is itself not completely enclosed by the cube. Vertices of cube, which is 0.2539 away from the center, are outside the sphere and part of the sphere is outside the cube.

6.5.4 Tone and Color Correction(1)

- Color transformations include tonal adjustment and color correction.
- Their common uses are photo enhancement and color reproduction.
- The computers are the mainstays of high-end color reproduction systems.
- Since these transformations are developed, refined, and evaluated on monitors, it is necessary to maintain a high degree of color consistency between the monitors and the eventual output devices as a printer.
- For color consistency, we need a device-independent color model that relates the color gamuts of the monitors, output devices and any other devices.
- Choice from many color management systems (CMS) is the CIE L*a*b* model for color consistency.
- The degree to which the luminance (lightness) information is separated from the color information in L*a*b* model is greater than in other color models.

◇ Tone and Color Correction(2)

- The L*a*b color components

$$L^* = 116 \cdot h\left(\frac{Y}{Y_W}\right) - 16$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right]$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right]$$

where

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

- X_W, Y_W, Z_W :
  - reference white tristimulus value
  - point of equal energy in the CIE chromaticity diagram (Section 6.1) (x = 0.3127, y = 0.3290, z=1-x-y)
- L* a* b*
  - Intensity : L*
  - Color : a* = R-G, b* = G-B

◇ Tone and Color Correction(3)

- The L*a*b color space is *colorimetric, perceptually uniform* and *device independent*
  - Colorimetric : colors perceived as matching are encoded identically
  - Perceptually uniform : color differences among various hues are perceived uniformly
- Its gamut encompasses the entire visible spectrum and can represent accurately the colors of any display, print or input device
- The L*a*b is an excellent decoupler of intensity (L*) and color (a* and b*), like the HSI system
- Useful in both image manipulations (tone and contrast editing) and image compression applications

◇ Tone and Color Correction(4)

- The principal benefit of calibrated imaging systems is to allow tonal and color imbalances to be corrected interactively and independently.
- Procedures of image calibration

  (1) Correction of the image's tonal range

  (2) Resolving color irregularities like over- and under-saturated colors
- The tonal range of an image (called its *key type*) refers to its general distribution of color intensities.

  - High-key image : concentrated at high (or light) intensities
  - Low-key image : located predominantly at low intensities
  - Middle-key image : lies in between high-key and low-key
- As in the monochrome image, it is desirable to distribute the intensities of color image equally between the highlights and shadows.

◇Tone and Color Correction(5)

Correction of tonal range

(a) Original

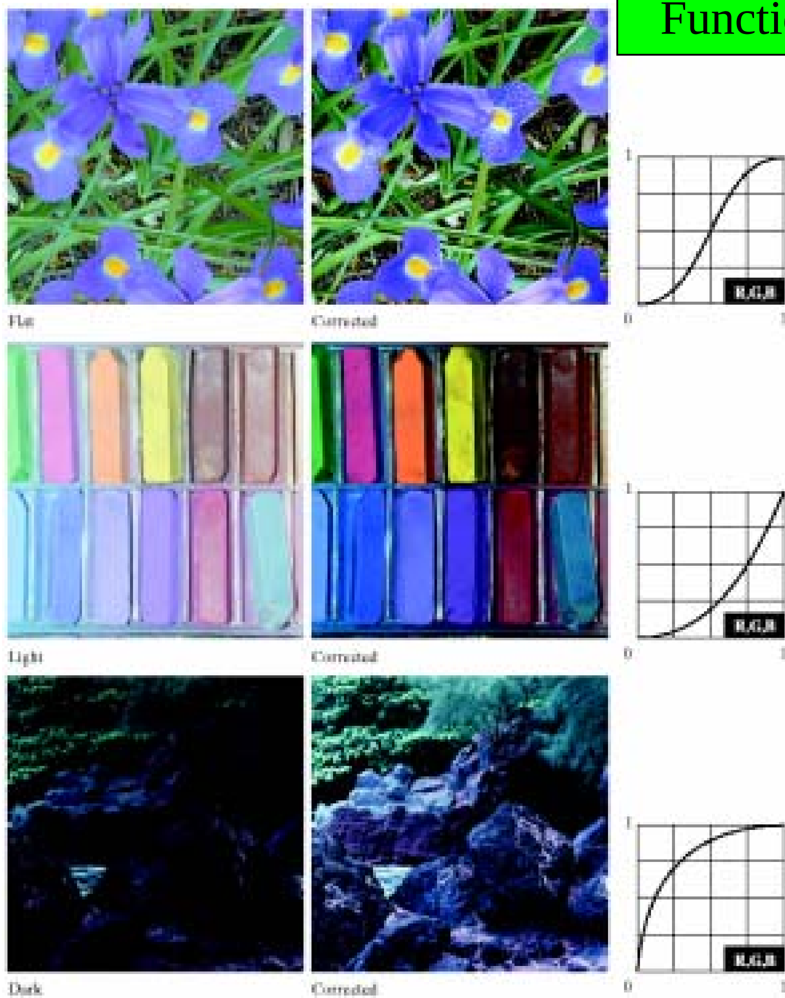
(b)Corrected

(c) Applied Transformation Function to get corrected

Flat

Light

Dark

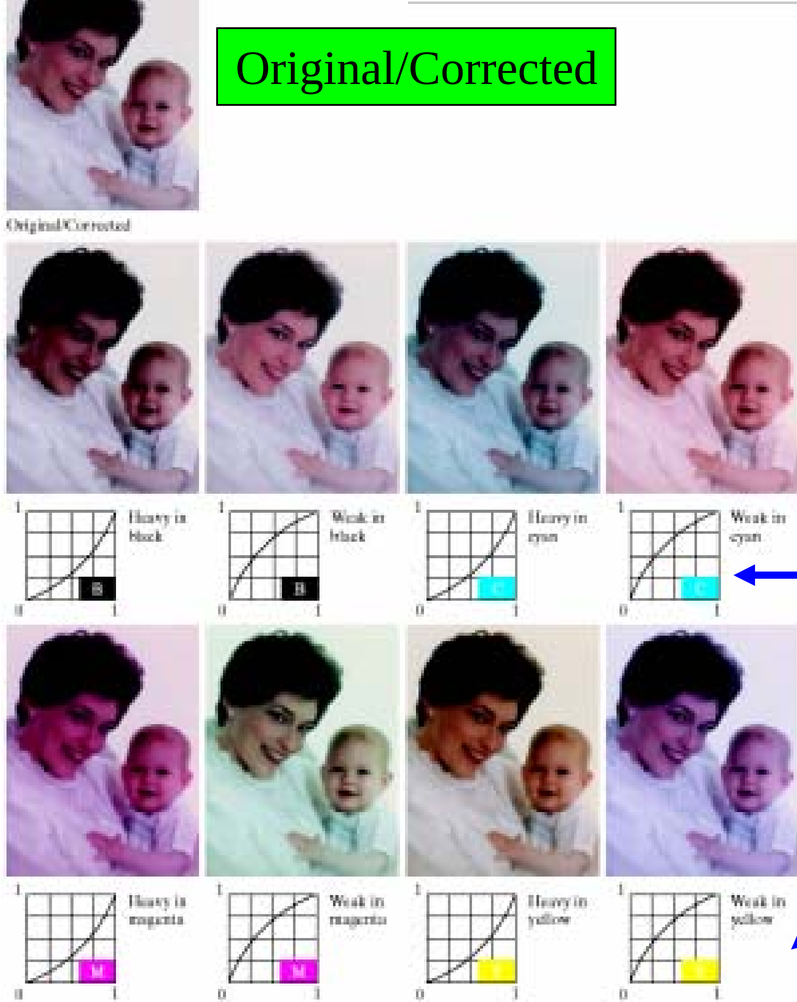


R,G,B

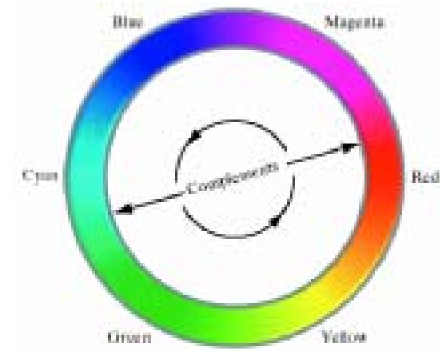
◇ Tone and Color Correction (6)

Correction of color imbalance in CMYK

FIGURE 6.36 Color balancing corrections for CMYK color images.



Original/Corrected



Color imbalance

Too much black

Excessive magenta

-Transformation function needed to get corrected.
-Inverse of these were applied to get imbalanced

◇ Tone and Color Correction (7)

• Visual assessment in color correction
  - Using objective analysis of a known color with a color spectrometer
  - Using white areas where the RGB or CMY(K) components should be equal
  - Using the color that human is highly perceptive, such as skin colors

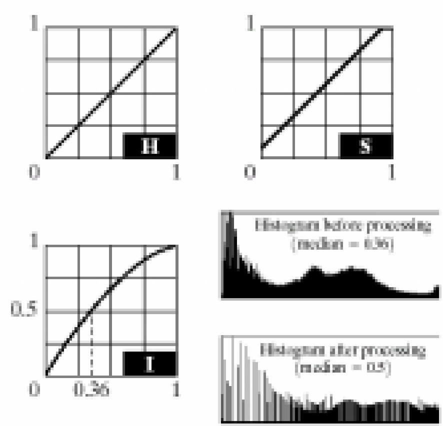
• Adjusting the color component
  - Important to realize that every action affects the overall color balance of the image : the perception of one color is affected by its surrounding color
  - The color wheel (see section 6.5.2) can be used to predict how one color component will affect others
    * For example, the proportion of any color can be increased by decreasing the amount of the opposite (or complementary) color
    * An abundance of magenta can be also decreased by
      (1) removing both red and blue or
      (2) adding green

6.5.5 Histogram Processing(1)

• Histogram equalization of the color intensities

(a) Original

(b) Transform. function



Saturation correcting

Histogram before processing
(median = 0.36)

Histogram after processing
(median = 0.50)

(c) Histogram equal.

(d) Correcting saturation

◇ Histogram Processing(2)

- An approach of the histogram processing in color image is to spread the color intensities uniformly, leaving the colors themselves (e.g., hues) unchanged.
- Fig.(c) : significantly bright and the grain in the caster is now visible. In particular, note that the loss of vibrancy in the oil and vinegar in the cruets.
- Fig.(d) : The results for correcting Fig.(c) partially by increasing saturation component. This type of adjustment is common when working with the intensity component in HSI space, because changes in intensity usually affect the relative appearance of colors.

6.6 Smoothing and Sharpening

- Color Image Smoothing
- Color Image Sharpening

6.6.1 Color Image Smoothing (1)

- The average of the RGB component vectors in the neighborhood

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

S_{xy} : the set of coordinates defining a neighborhood centered at (x,y)

- For each component of RGB vector

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

◇ Color Image Smoothing (2)

- RGB components of color image

(a) Color image (b) Red component

a b
c d

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

(b) Green component (d) Blue component

◇ Color Image Smoothing (3)

• HSI components of the color image

(a) Hue (b) Saturation (c) Intensity

a b c

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

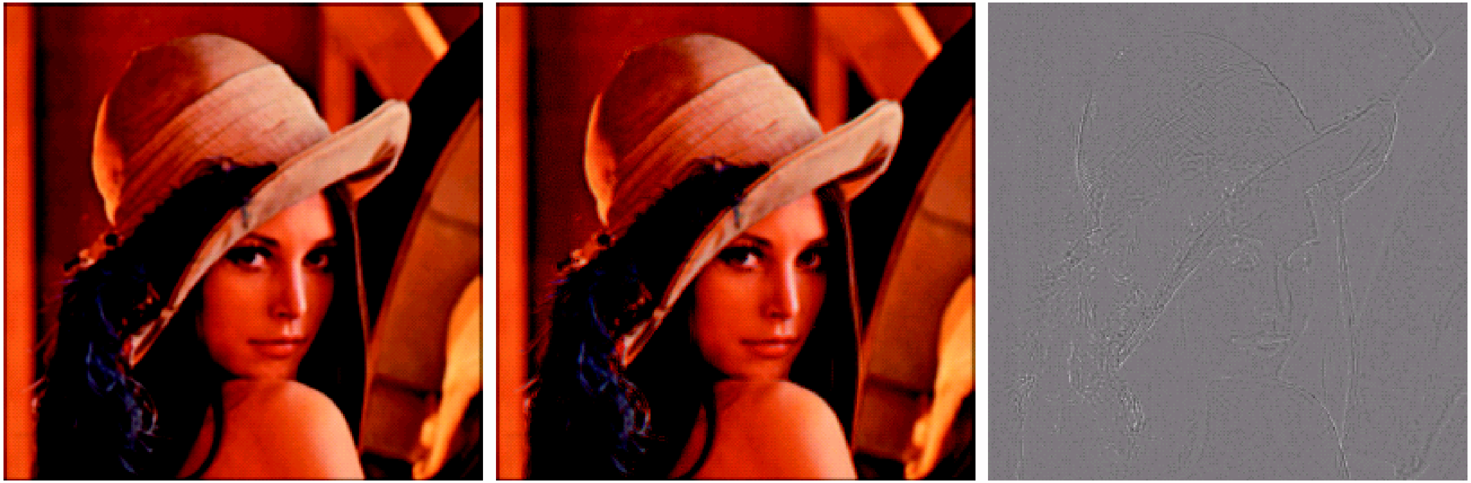
◇ Color Image Smoothing (4)

• Image smoothing with a 5 x 5 averaging mask

(a) RGB component

(b) Intensity component of HSI

(c) Difference between two results



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.

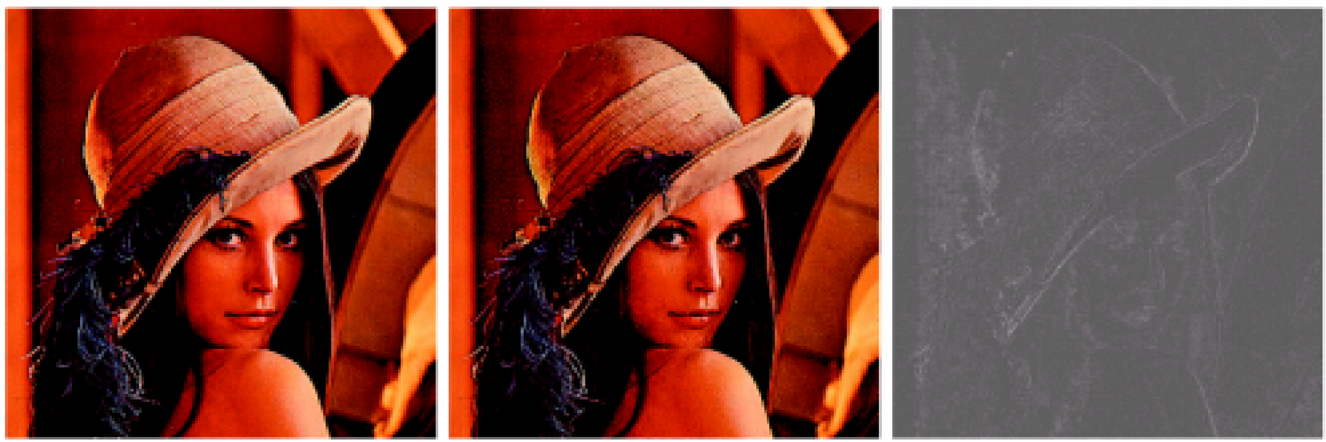
Fig. 6.40(b) : By smoothing only the intensity image, the pixels maintain their original hue and saturation

6.6.2 Color Image Sharpening

- In the RGB color system, the Laplacian of vector c

$$\nabla^2[c(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix} \qquad \text{Eq.(6.6-3)}$$

(a) RGB component (b) Intensity component of HSI (c) Difference between two results



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.

6.7 Color Segmentation

- Segmentation in HSI Color Space
- Segmentation in RGB Vector Space
- Color Edge Detection

6.7.1 Segmentation in HSI Color Space(1)

- The hue image is used frequently in segmentation of color images because it represents color conveniently.
- Saturation is used as a masking image in order to isolate further regions of interest in the hue images.
- The intensity image is used less frequently because it carries no color information.

◇ Segmentation in HSI Color Space(2)

• Suppose that it is of interest to segment the reddish region in Fig. 6.42(a)

(a) Color image

Reddish region (not pure red, but magenta and blue side red, thus high hue values)

(b) Hue

Relatively high values

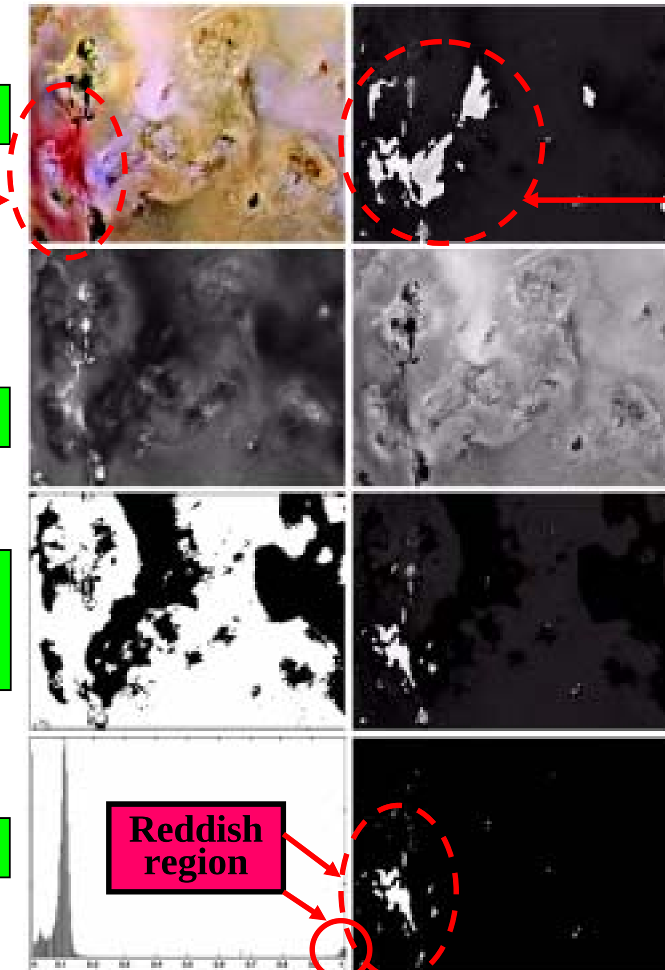
(c) Saturation

(d) Intensity

(e) Binary saturation mask (Th. = 0.1of max.)

(f) Product of (b) and (e)

(g) Histogram of (f)



(h) Segmentation of red component in (a) (Th. = 0.9)

◇ Segmentation in HSI Color Space(3)

- The white points in Fig. 6.42(h) identifies the reddish regions of the interest.
- But this was far from a perfect segmentation because the white points did not identify all regions with reddish hue in Fig. 6.42(a).

6.7.2 Segmentation in RGB Vector Space(1)

- Segmentation is one area in which better results generally are obtained by using RGB color vectors.
- The objective of segmentation is to classify objects of a specified color range in an RGB image.
- Given a set of sample color points (representatives of the colors of interest), we obtain an estimate of the “average” color.
- In order to classify whether each pixel has a color in the specified range or not, it is necessary to have a measure of similarity.

◇ Segmentation in RGB Vector Space(2)

The similarity measures

- The measure for Euclidean distance (Sphere)

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

$$= [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{1/2} < D_0$$

a : the average vector of given color samples
z : arbitrary point
D_0 : threshold value

- Generalization of the distance measure (Elliptic)

$$D(z,a) = [(z-a)^T C^{-1} (z-a)]^{1/2} \quad < D_0$$

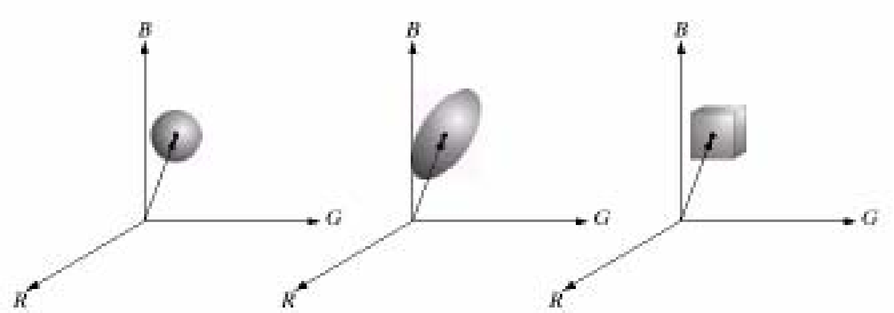
C : the covariance matrix of the given color samples

- Distance measure using bounding box

$$|z_R - a_R| < k\sigma_R, \quad |z_G - a_G| < k\sigma_G, \quad |z_B - a_B| < k\sigma_B$$

σ : standard deviation of given color samples

k : constant



(a) Sphere (b) Elliptic (c) Bounding box

◇ Segmentation in RGB Vector Space(3)

- Distance measure of the sphere form
  - One of the simplest measures is the Euclidean distance.

- Distance measure of the elliptic form
  - **C** is the covariance matrix of the samples (representatives of the color).
  - The locus of points such that $D(z,a) =< D_0$ describes a solid 3-D elliptical body.
  - Its principal axes are oriented in the direction of maximum data spread.
  - When **C** = **I** (identity matrix), this measure becomes the sphere form.

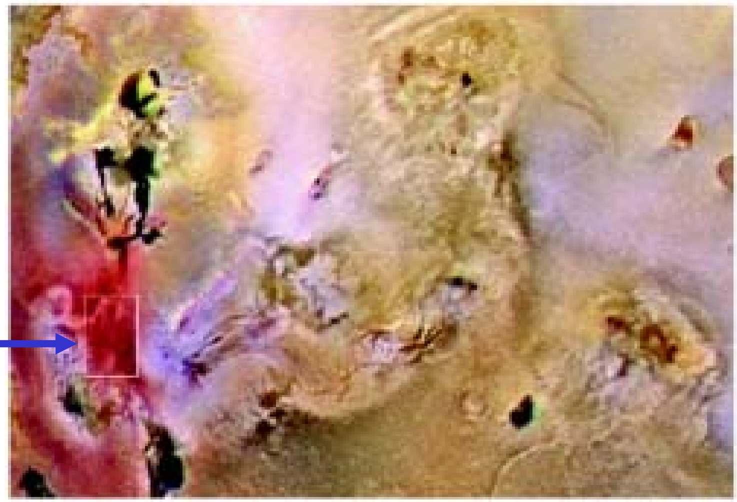
- Distance measure of the bounding box form
  - Using the squared root in the above measures are computationally expensive.
  - A compromise is to use a bounding box.
  - Its dimensions along each of the color axes are proportional to the standard deviation of the given samples.

◇ Segmentation in RGB Vector Space (4)

• Using bounding box as the distance measure

(a) Original image

samples of reddish colors to compute a's



(b) Result of segmentation

Bounding box size

$$|z_R - a_R| < 1.25\sigma_R,$$
$$|z_G - a_G| < 1.25\sigma_G,$$
$$|z_B - a_B| < 1.25\sigma_B$$

Much more accurate than previous results

6.7.3 Color Edge Detection (1)

- Edge detection is an important tool for image segmentation.
- We are interested in the issue of computing edges on an individual-image basis versus computing edges directly in color vector space.
- Edge detection is performed by gradient operators.
- Computing the gradient on individual images and then using the results to form a color image will lead to erroneous results.
- What is the reason ?

◇ Color Edge Detection (2)

Simple examples of two M x M color images (M : odd)

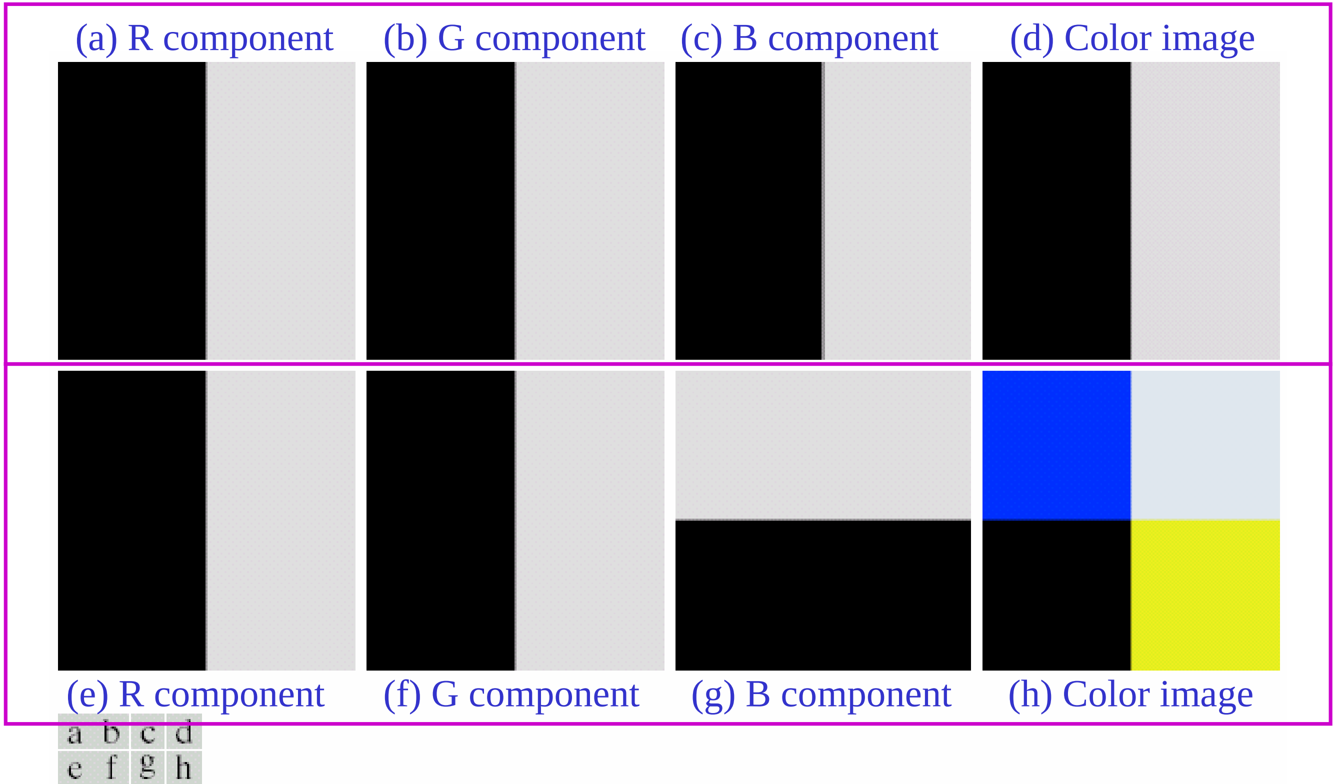


FIGURE 6.45 (a)–(c) R, G, and B component images and (d) resulting RGB color image. (f)–(g) R, G, and B component images and (h) resulting RGB color image.

◇ Color Edge Detection (3)

- The two M x M color images (M odd) in Fig. (d) and (h)
  - Fig.(d) : Composite images obtained by adding R, G and B components
  - Fig.(h) : Different B component is used to generate Fig.(h)
- A Example for computation of the gradient
  - Compute the gradient image from each component image
  - Add the results to form the two corresponding RGB gradient images
- Our intuitive expectation of the gradient at point [(M+1)/2, (M+1)/2]
  - Fig. (d) : The edges of the R, G, and B images are in the same direction at that point.
  - Fig. (h) : Only two of the edges (R, G) are in the same direction.
  - We expect that Fig. (d) will have stronger edge at that point than Fig.(h).
  - However, the values of the gradients are the same in both cases.
- Conclusion
  - Processing the three individual plans to form a composite gradient image can yield erroneous results.
  - We need obviously a new definition of the gradient applicable to the color image.

◇ Color Edge Detection (4)

- At first, recall the gradient of the scalar function in Section 3.7.3
- In scalar function, the gradient is defined as a vector representing direction and magnitude of maximum rate of change of *f(x,y)*

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

direction

$$\theta = \tan^{-1}\left[\frac{G_y}{G_x}\right] = \tan^{-1}\left[\frac{\partial f / \partial y}{\partial f / \partial x}\right],$$

magnitude

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

- This concept can be extended to the gradient of the vector function
- A color vector *c* at any pixel point (x, y) in the RGB color space

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

- The gradient vectors for each component of the vector **c**

$$u = \frac{\partial R}{\partial x} r + \frac{\partial G}{\partial x} g + \frac{\partial B}{\partial x} b \quad \textit{and} \quad v = \frac{\partial R}{\partial y} r + \frac{\partial G}{\partial y} g + \frac{\partial B}{\partial y} b$$

where, ***r***, ***g*** and ***b*** are unit vectors along the R,G, and B axis, respectively

◇ Color Edge Detection (5)

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

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

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

- A new definition for the gradient of the vector **c** is the maximum rate of change of the color pixel ***c****(x,y)* shown below. (Refer to Di Zenzo [1986] for proof.)

  - The *direction* of the gradient of the vector ***c***

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

  - The *magnitude* of the gradient of the vector ***c***

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

◇ Color Edge Detection (6)

Comparison of two gradient methods

- The edge detail of **Fig. 6.46(b) is more complete** than the detail in Fig. 6.46(c)
- See the detail around **the right eye**
- Note that **both approaches yielded reasonable results.**
- The extra detail in Fig. 6.46(b) is worth the added computational burden

(a) A color image

(b) The gradient using the vector method.

(c) Added gradients of RGB components

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

◇ Color Edge Detection (7)

• **Fig. 6.47 shows the three gradient images for R, G, and B components that were scaled and added to produce Fig. 6.46(c).**

a b c (a) R component (b) G component (c) B component

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).

6.8 Noise in Color Images (1)

- Usually the same noise characteristic is prevailed in each color channel.
- However, it is possible for each channel to be affected differently by noise.
  - One possibility : Electronics affect a particular channel to malfunction.
  - Other possibility : Difference of illumination strengths for channels affects. (Usually low strength channel is noisier.)
- In color images, it is interesting to check how noise carries over when converting from one color model to another.

◇ Noise in Color Images (2)

Adding noisy R, G, and, B components to get color image

- R, G and B components are corrupted by Gaussian noise of mean 0 and variance 800
- Fine grain noise is less visually noticeable in a color image than in any component image.
- Recall effects of the image averaging in the Gaussian noise (Section 3.4.2)

◇ Noise in Color Images (3)

Converting the RGB image in Fig. 6.48(d) to HSI

- The hue and saturation components are significantly degraded
  - This is due to the nonlinearity of the cos and min operations in converting equation (Section 6.2.3)
- The intensity component is slightly smoother than any of the three noisy RGB component images.
  - This is due to the fact that the intensity image is the average of the RGB images.

a b c (a) Hue (b) Saturation (c) Intensity

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 (4)

Another example for converting the RGB image to HSI

(a) RGB image with noisy G component

(b) Hue

(c) Saturation

(d) Intensity

◇ Noise in Color Images (5)

- An example for the noise spread in conversion of RGB to HSI
- Fig. 6.50(a) : Only G channel is affected by salt-and-pepper noise
  - The probability of either salt or pepper is 0.05.
- Fig.6.50(b) through (d) : show clearly how the noise spreads from the green RGB channel to all the HSI images.
- This is not unexpected because computation of the HSI components makes use of all RGB components.

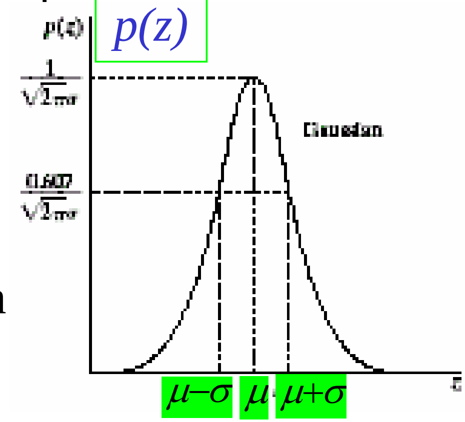
◇ Noise Models(1)

- **Gaussian noise (normal noise) (Section 5.2)**
  - Arises in an image due to factors such as electronic circuit noise and sensor noise under poor illumination and/or high temperature.
  - Happens frequently in practice.

PDF (probability density function) of noise

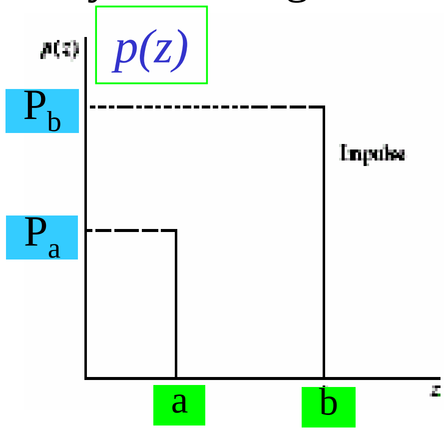
$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$

z : gray level
μ : the mean of z
σ : standard deviation

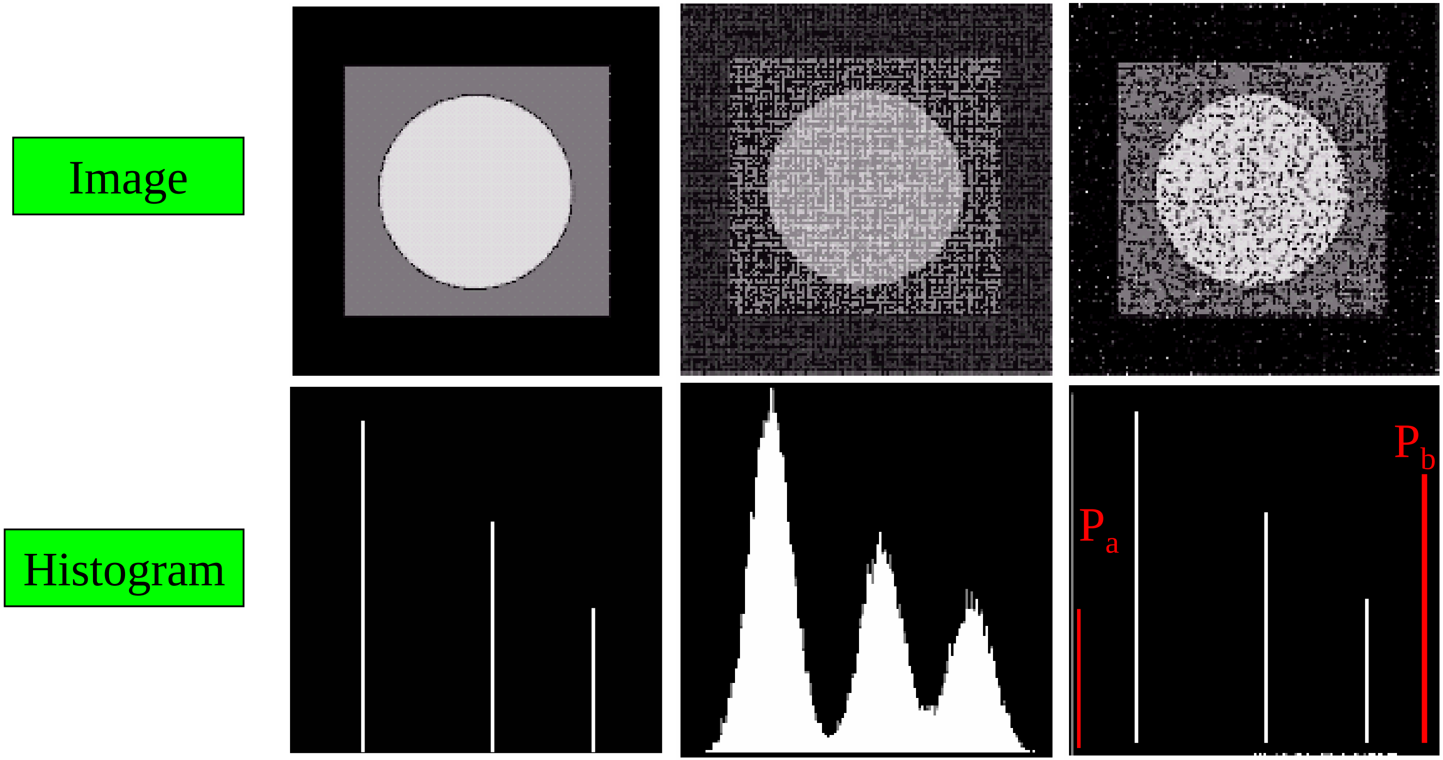


- **Impulse (salt-and-pepper) noise**
  - Found in situations where quick transients, such as faulty switching during imaging

$$p(z) = \begin{cases} P_a & \text{for} \quad z = a \quad \text{(a dark dot)} \\ P_b & \text{for} \quad z = b \quad \text{(a light dot)} \\ 0 & \text{otherwise} \end{cases}$$



◇ Noise Models(2)



(a) Noise free (b) Gaussian noise (c) Salt & Pepper noise

6.9 Color Image Compression(1)

- The number of bits required to represent colors is typically three to four times greater than that required in gray levels.
- Data compression plays a central role in the storage and transmission of color images.
- Compression is the process of reducing or eliminating redundant and/or irrelevant data. (Chap. 8)

◇ Color Image Compression(2)

Comparison between the original image and the compressed image

(a) Original image

Where can you find the difference between the two images ?

(b) Compressed image

Compression ratio = 230 : 1

◇ Color Image Compression(3)

- Fig. 6.51(a) : 24-bit RGB full-color image of an iris
- Fig. 6.51(b) : The image is reconstructed from a compressed version of Fig. 6.51(a).
- The JPEG 2000 compression algorithm is used to generate Fig. 6.51(b). and is a recently introduced standard (Section 8.6.2)
- The compressed image contains only 1 bit for every 230 bits in the original image.
  - Compression ratio is 230 : 1
  - Time required in transmission, for example on internet :
    * The compressed image : 1 minute
    * Original image : 4 hours
- Fig. 6.51(b) is an approximation of Fig. 6.51(a) and is slightly blurred.
  - Blurring is a characteristic of many lossy compression techniques.
  - It can be reduced or eliminated by altering the level of compression.