Digital Image Processing

Lecture # 10

Color Image Processing

ALI JAVED

Lecturer

SOFTWARE ENGINEERING DEPARTMENT

U.E.T TAXILA

Email:: ali.javed@uettaxila.edu.pk

Office Room #:: 7

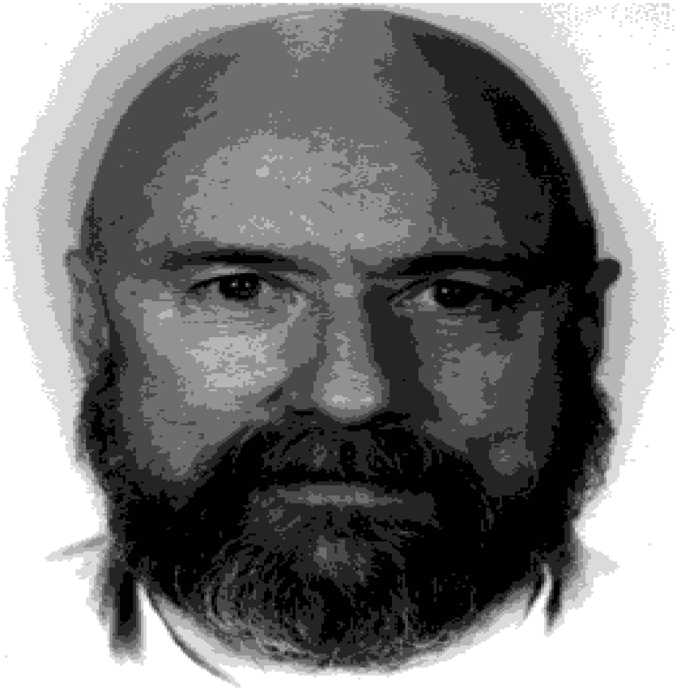
Pseudo-Color (False Color) Image Processing

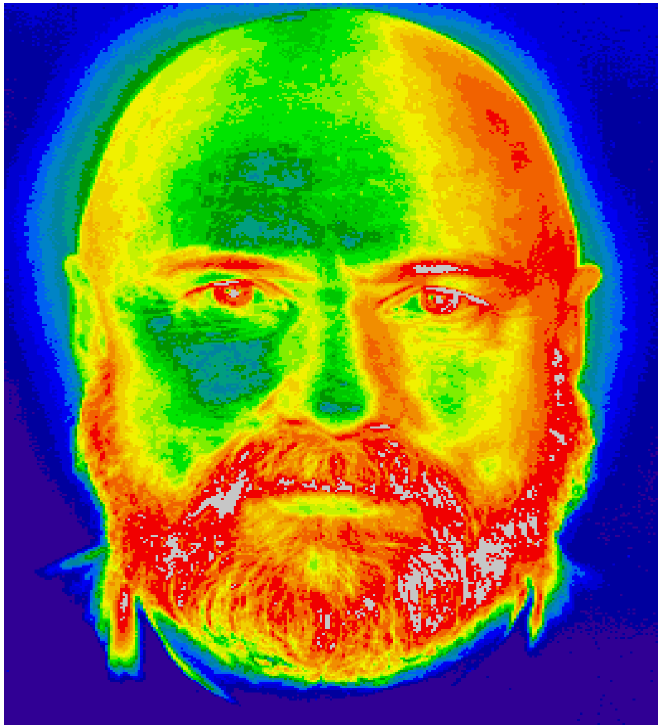
- Pseudo-color Image Processing consists of assigning colors to gray levels based on specific criterion

- Generally, the eye cannot distinguish more than about 2 dozen gray levels in an image. Thus subtle detail can easily be lost in looking at gray scale images. To enhance variations in gray level and make them more obvious, gray scale images are frequently pseudo-colored, where each gray scale (generally at least 256 levels for most displays) are mapped to a color level through a LUT. The eye is extremely sensitive to color and can distinguish thousands of color values in a picture.

Pseudo-Coloring using LUT

- CLUT(Color lookup table):: A mapping of a pixel value to a color value shown on a display device.
  - For example, in a grayscale image with levels 0, 1, 2, 3, and 4, pseudo-coloring is a color lookup table that maps 0 to black, 1 to red, 2 to green, 3 to blue, and 4 to white.





Intensity Slicing

- The technique of intensity slicing or density slicing or color coding is one of the simplest example of Pseudo-color image processing

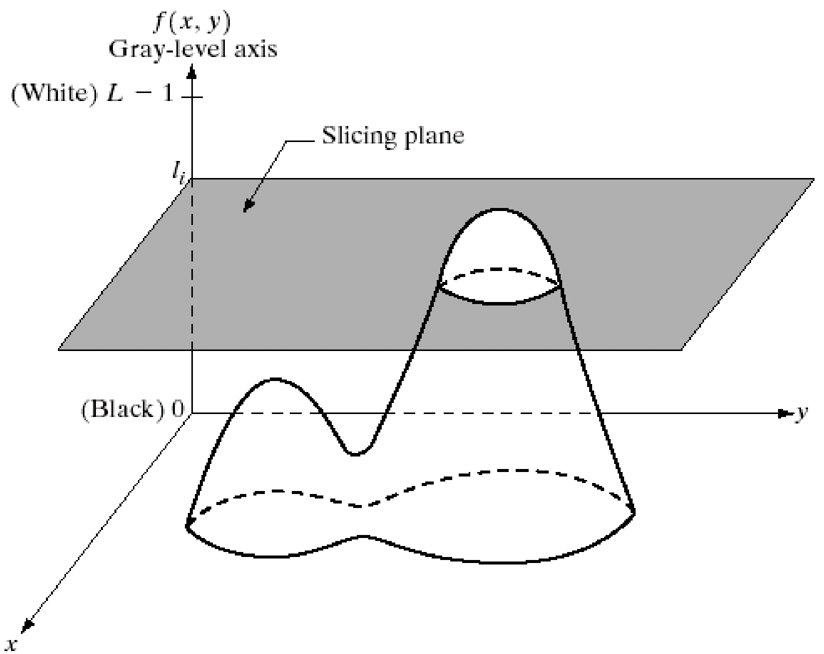


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

Intensity Slicing

- The Gray Scale [0,L-1] is divided into L levels; where l_0 represents Black ($f(x,y)=0$) and l_{L-1} represents white ($f(x,y)=L-1$)
- Suppose that P planes perpendicular to the intensity axis are defined at levels $l_1, l_2 \ldots\ldots, l_p$
- Then assuming that $0<P<L-1$ the P planes partition the gray scale into P+1 intervals, $V_1, V_2 \ldots\ldots V_{p+1}$

Intensity Slicing

- Gray level to color assignments are made according to the relation:

$$f(x,y)= c_k \quad \text{if } f(x,y)€v_k$$

- Where c_k is the color associated with the kth intensity interval v_k defined by the partition planes at $l=k-1$ and $l=k$

An Alternative View of Intensity Slicing

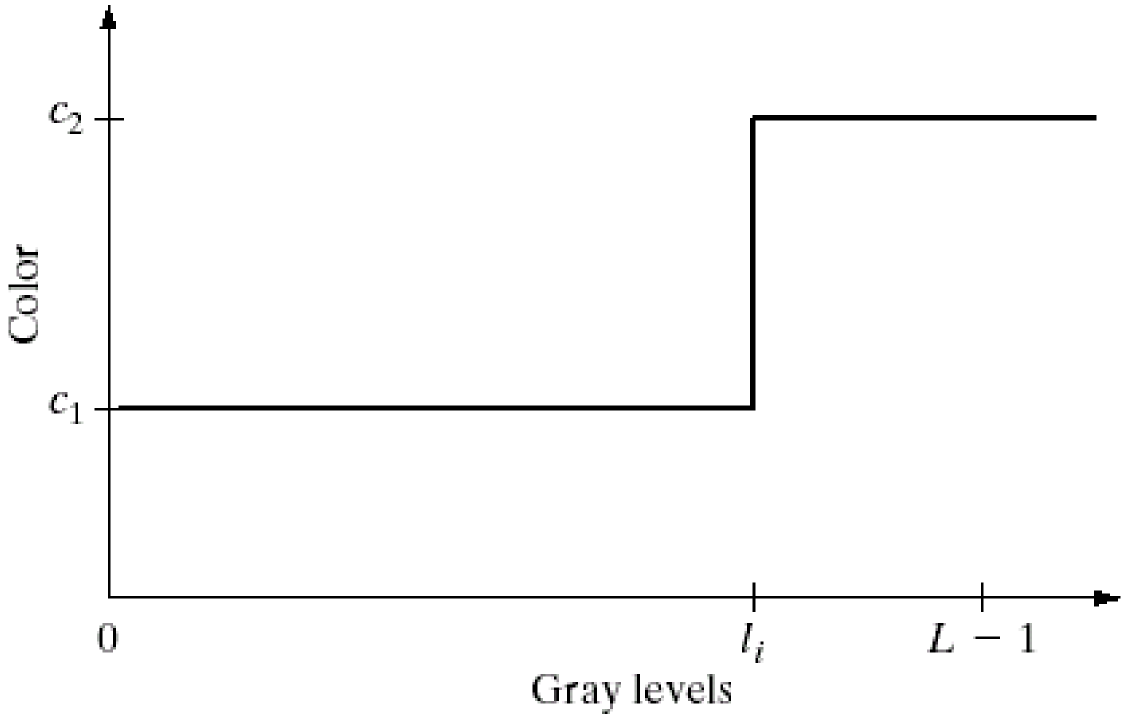
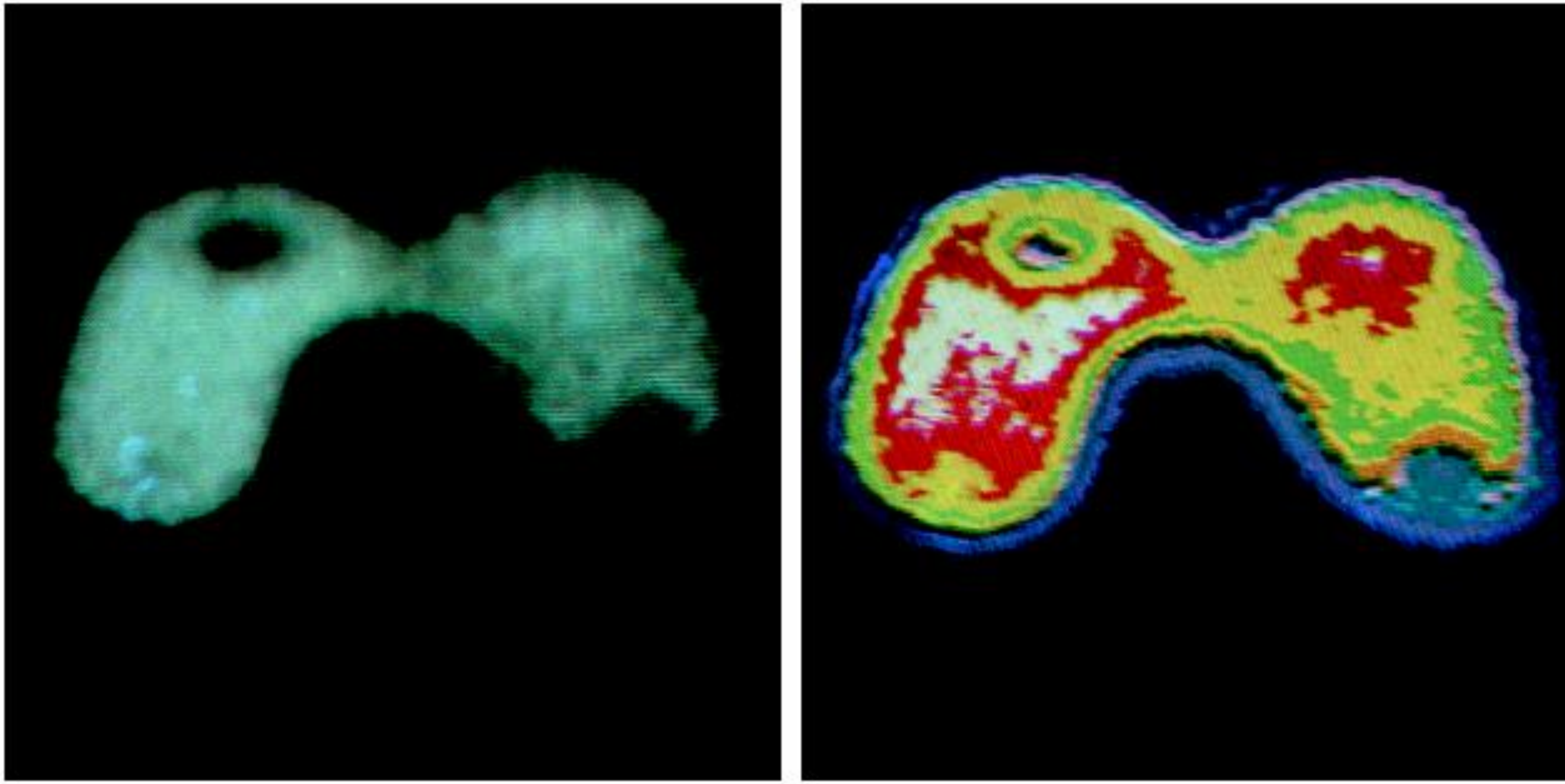
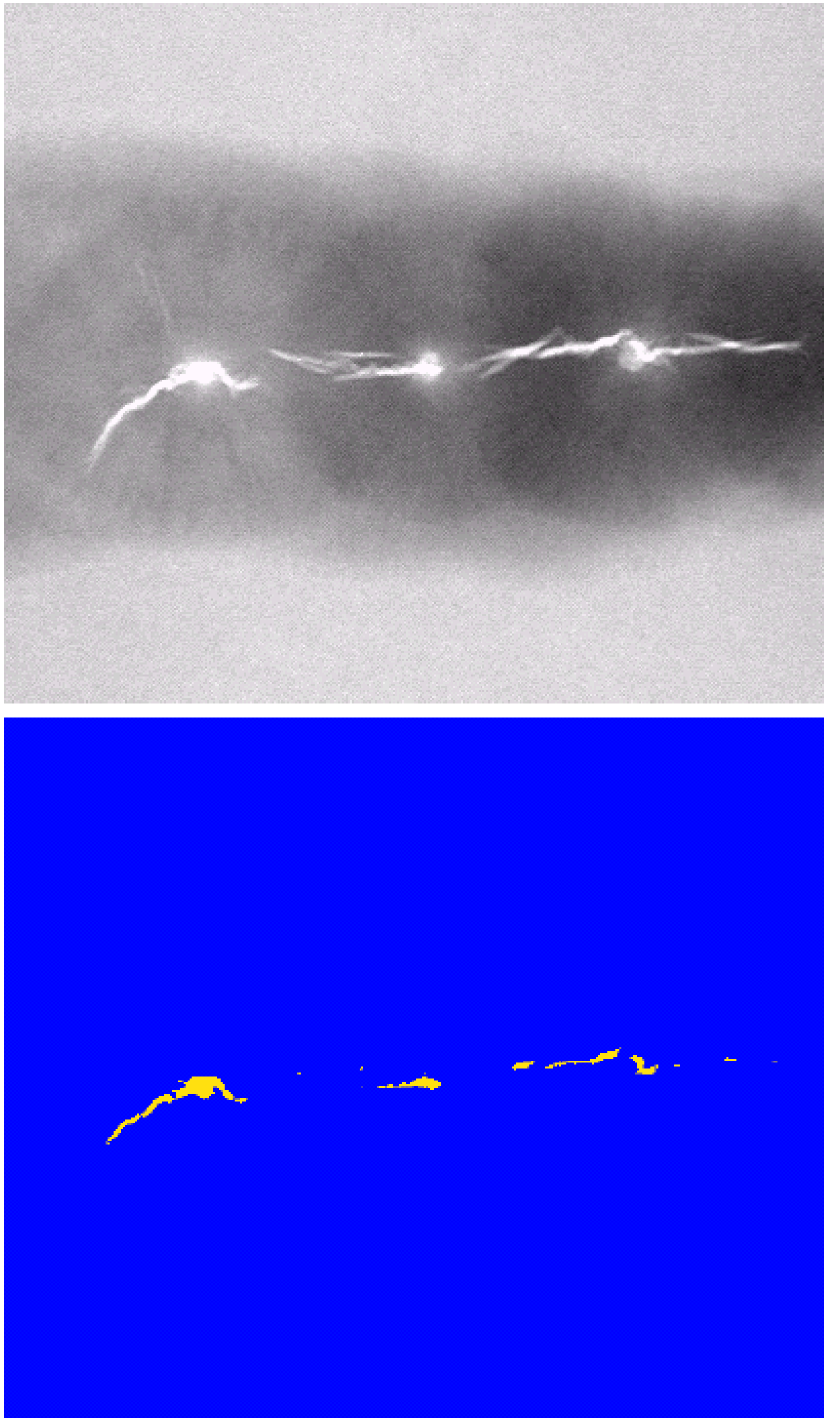


FIGURE 6.19 An alternative representation of the intensity-slicing technique.



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.)



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.)

Gray to Color Conversion

Gray to Color Conversion

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

$$f_G(x, y) = 0.33\, f(x, y)$$

$$f_B(x, y) = 0.11\, f(x, y)$$

Gray to Color Conversion

$$f_R(x, y) = 0.33\, f(x, y)$$

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

$$f_B(x, y) = 0.11\, f(x, y)$$

Basics of Full Color Image Processing

- Full color image processing fall into 2 categories.
  - In 1st category we process each component image individually and then form a composite processed color image from the individually processed component.
  - In 2nd category we work with color pixels directly. Because full color images have at least three components, color pixels are really vectors.
  - Let c represent an arbitrary vector in RGB color space:

$$C = \begin{bmatrix} C_r \\ C_g \\ C_b \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Basics of Full Color Image Processing

- Color components are the function of co-ordinates(x,y) so we can write it as:

$$C(x,y) = \begin{pmatrix} C_R(x,y) \\ C_G(x,y) \\ C_B(x,y) \end{pmatrix} = \begin{pmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{pmatrix}$$

- For an image of size MxN there are MN such vectors, c(x,y), for x=0,1,2,…,M-1; y=0,1,2,…,N-1

Color Transformations

- Color transformation can be represented by the expression ::

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

  $f(x,y)$: input image

  $g(x,y)$: processed (output) image

  $T[*]$: an operator on f defined over neighborhood of (x,y).

- The pixel values here are triplets or quartets (i.e group of 3 or 4 values)

Color Transformations

- $S_i = T_i(r_1, r_2, \ldots, r_n) \qquad i = 1, 2, 3, \ldots, n$

  - r_i and S_i are variables denoting the color components of f(x,y) and g(x,y) at any point (x,y).

  - n is the no of color components

  - $\{T_1, T_2, \ldots, T_n\}$ is a set of transformation or color mapping functions.

- Note that n transformations combine to produce a single transformation T

Color Transformations

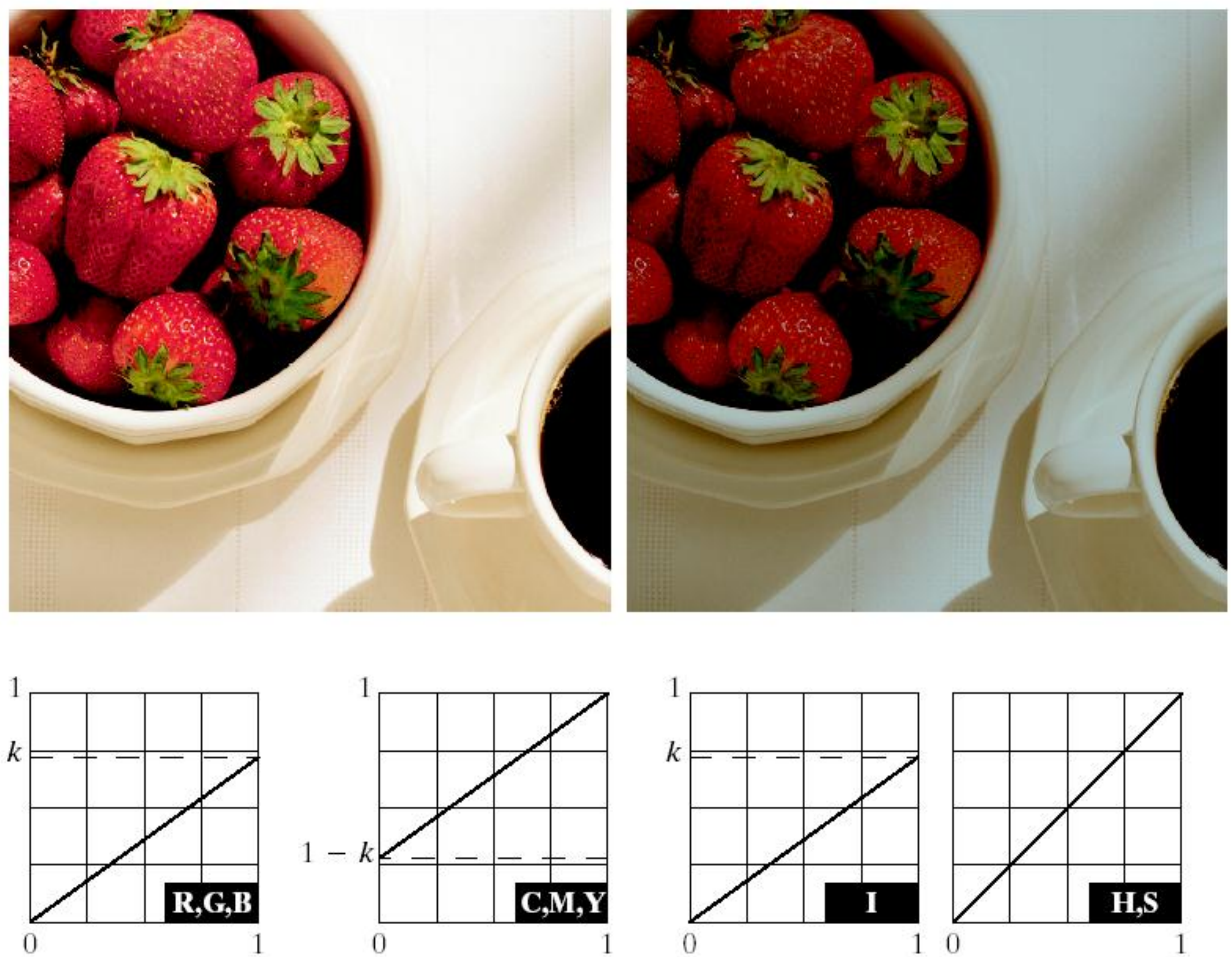
- The color space chosen determine the value of n.
- If RGB color space is selected then n=3 & r1,r2,r3 denotes the red, blue and green components of the image.
- If CMYK color space is selected then n=4 & r1,r2,r3,r4 denotes the cyan, hue, magenta and black components of the image.
- Suppose we want to modify the intensity of the given image
- using $g(x,y)=k*f(x,y)$ where $0<k<1$

Full color

Color Transformations

- In HSI color space this can be done with the simple transformation
  - $s3=k*r3$
  - where $s1=r1$ and $s2=r2$
  - Only intensity component r3 is modified.
- In RGB color space 3 components must be transformed:
  - $si=k*ri$ $\qquad$ $i=1,2,3$.
- In CMY color space 3 components must be transformed:
  - $si=k*ri + (1-k)$ $\qquad$ $i=1,2,3$.
- Using $k=0.7$ the intensity of an image is decreased by 30%

Color Transformations



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.)

Color Complements

- The hues opposite to one another on the Color Circle are called Complements.

- Color Complement transformation is equivalent to image negative in Grayscale images

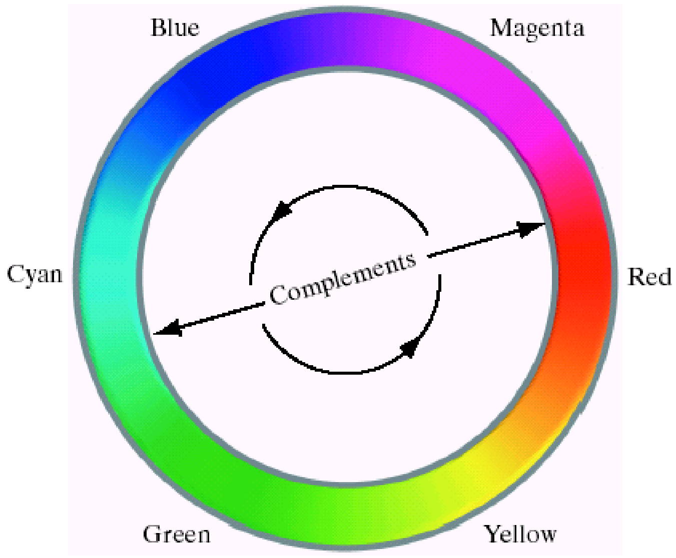


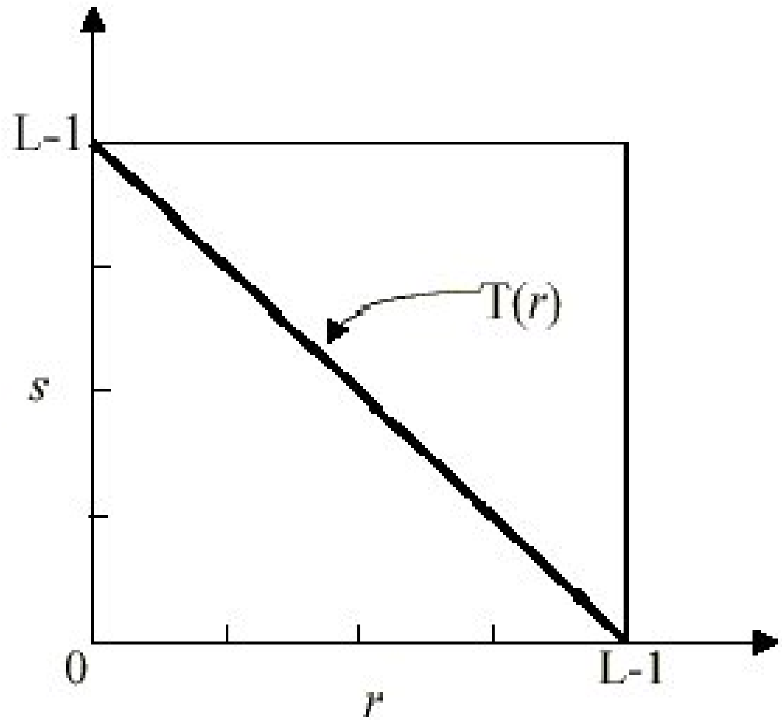
FIGURE 6.32 Complements on the color circle.

Color Complements

- $S=T(r)= L-1-r$ For Gray scale image

- $S_i=T(r_i)=L-1-r_i$ For Color image

Where i=1,2,3



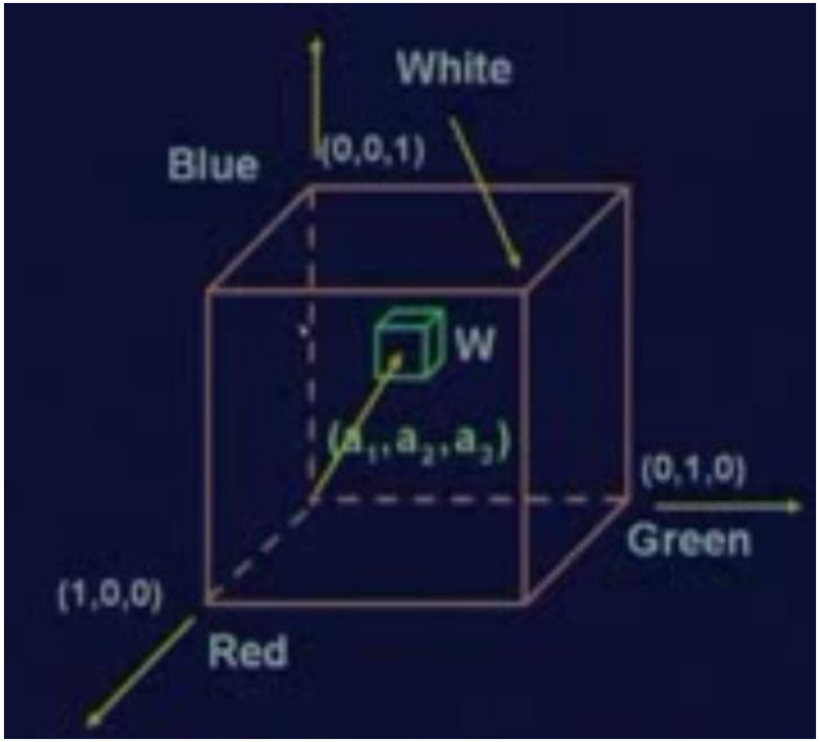
Color Complements

Color Slicing

- Highlighting a specific range of colors in an image is useful for separating objects from their surroundings.

- Display the colors of interest so that they are distinguished from background.

- One way to slice a color image is to map the color outside some range of interest to a non prominent neutral color.



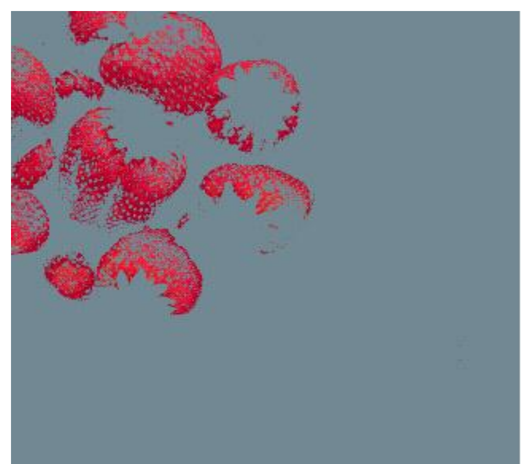
Color Slicing

- $S_i = 0.5$ if $[\,| r_j - a_j | > W/2\,]$ for $1 <= j <= 3$

- $S_i = r_i$ otherwise

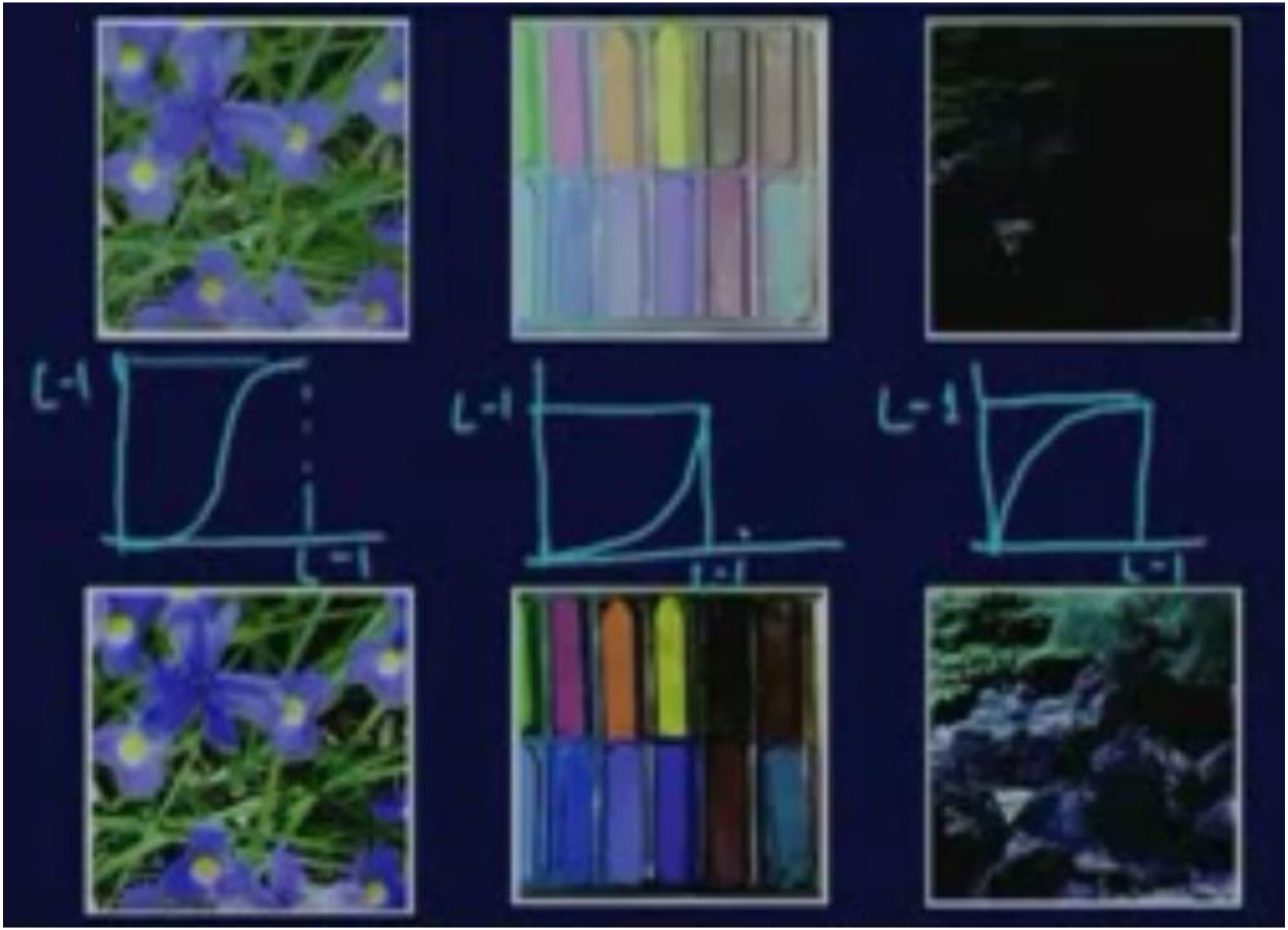
  Where i=1,2,3


Full color



Tone Correction

Flat Light Dark



Color Correction

- When a color imbalance is noted, there are a variety of ways to correct it

- When adjusting the color components of an image it is important to realize that every action affects the overall color balance of the image (Perception of one color is affected by its surrounding colors)

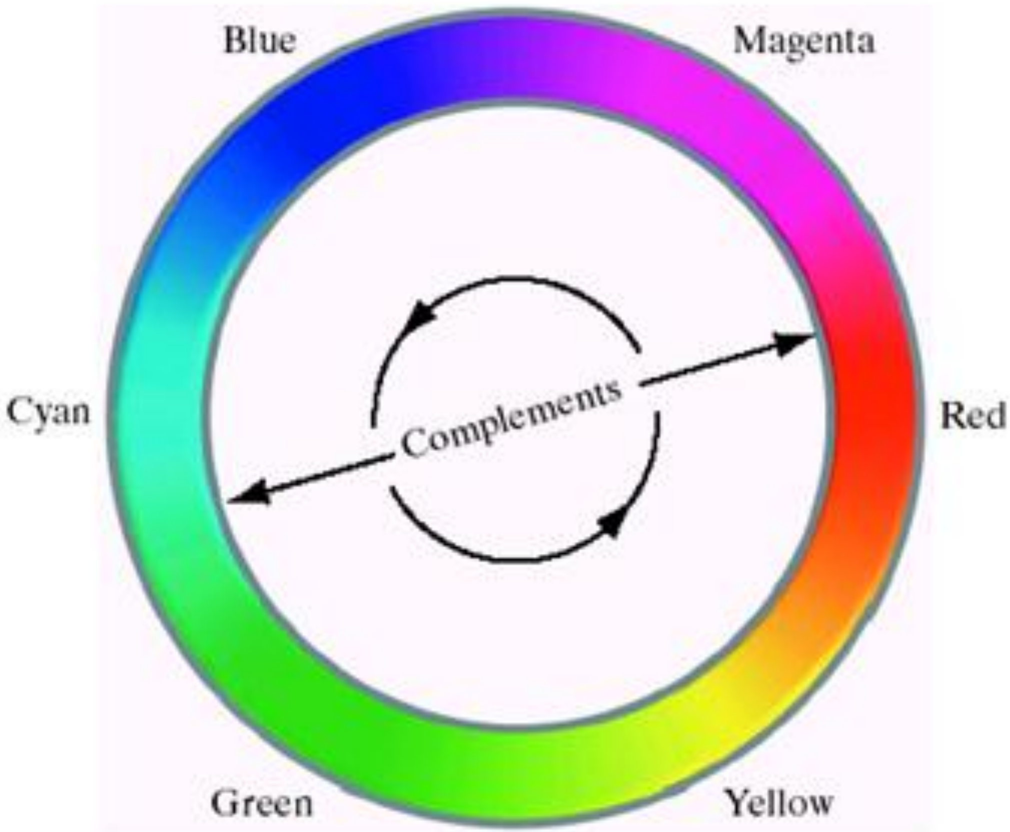
- Based on the color wheel, the proportion of any color can be increased by decreasing the amount of the opposite color in the image

- Similarly it can increase by raising the proportion of two immediately adjacent colors or decreasing the percentage of the two colors adjacent to the complement

- Suppose for example there is an abundance of magenta in an RGB image, it can decreased by

    - Reducing both red and blue or
    - Adding Green

Color Correction

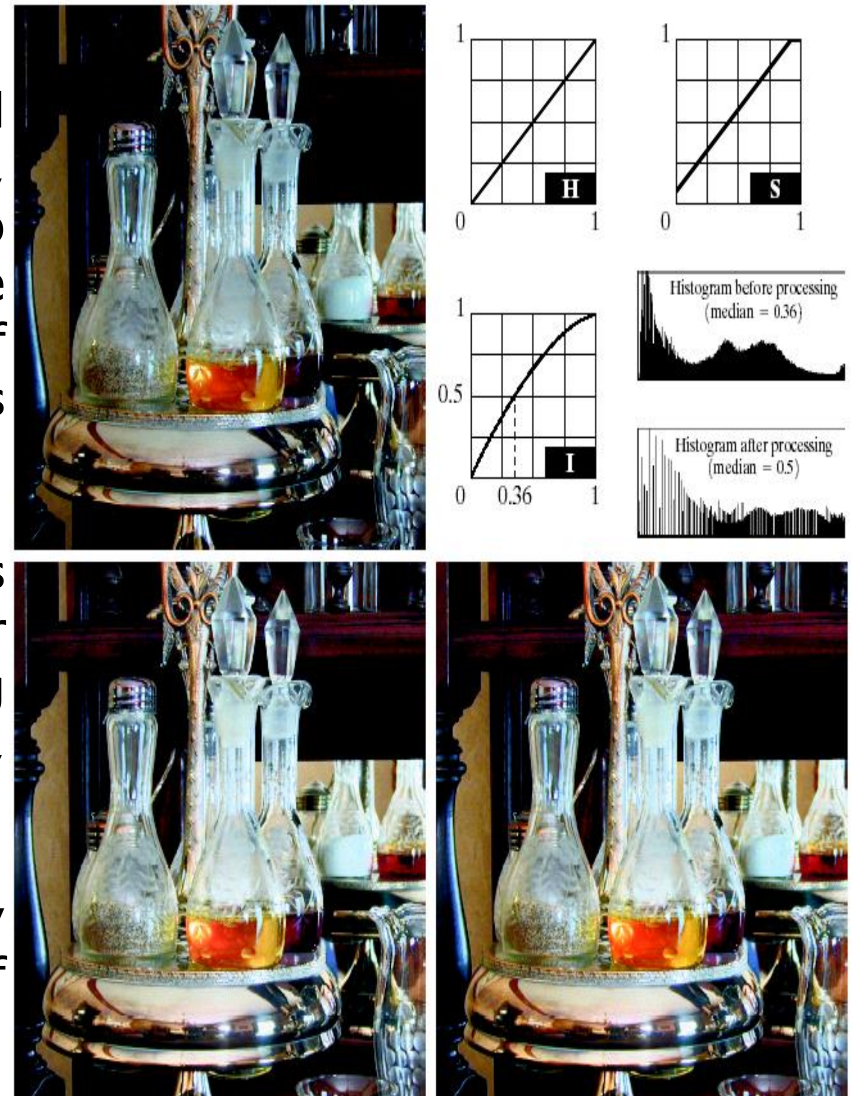


Histogram Processing

- Color images are composed of multiple components, however it is not suitable to process each plane independently in case of histogram equalization. This results in erroneous color.

- A more logical approach is to spread the color intensities uniformly, leaving the colors themselves(hue, saturation) unchanged.

- HSI approach is ideally suited to this type of approach.



a b
c d

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

Color Image Smoothing

- Color images can be smoothed in the same way as gray scale images, the difference is that instead of scalar gray level values we must deal with component vectors of the following form:

$$C(x,y) = \begin{pmatrix} C_R(x,y) \\ C_G(x,y) \\ C_B(x,y) \end{pmatrix} = \begin{pmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{pmatrix}$$

- The average of the RGB component vector in this neighborhood is:

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

Color Image Smoothing

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

- We recognize the components of this vector as the scalar images that would be obtained by independently smoothing each plane of the starting RGB image using conventional gray scale neighborhood processing.

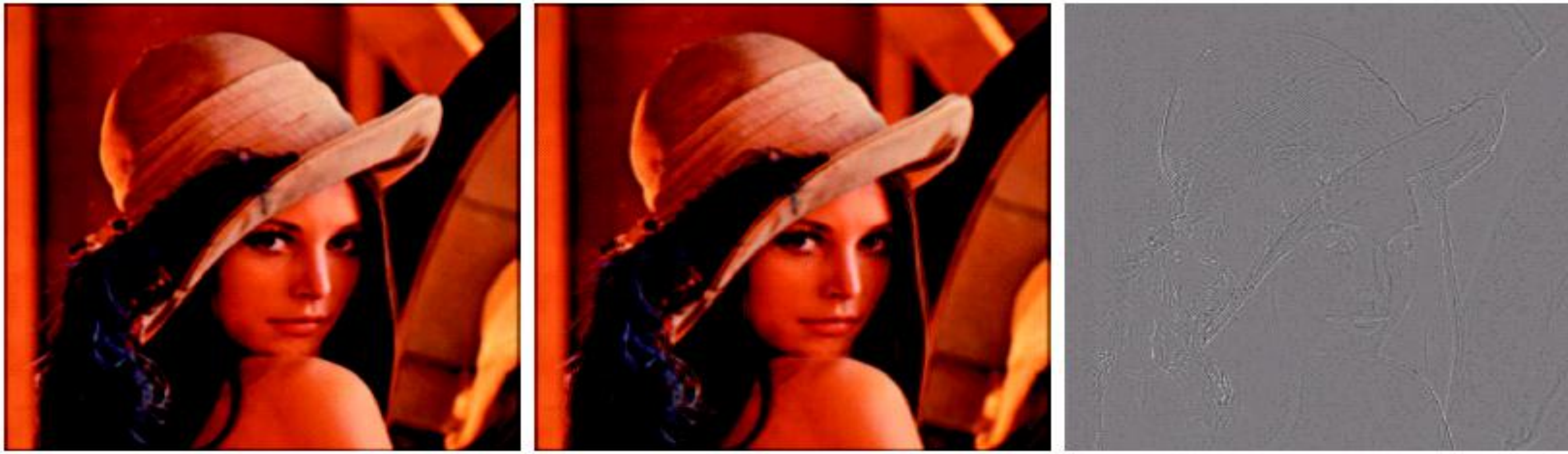
- Thus we conclude that smoothing by neighborhood averaging can be carried out on a per color plane basis.

Color Image Smoothing

a b
c d

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

Color Image Smoothing



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.

- The result of RGB and HSI are not identical as shown in the difference image of the RGB and HSI processed image
- This is due to the fact that average of the two pixels of differing color is a mixture of two colors, not either of the original colors (case of RGB)
- By Smoothing only the intensity component, the pixels maintain their original hue and saturation and thus their original colors

Color Image Sharpening

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.

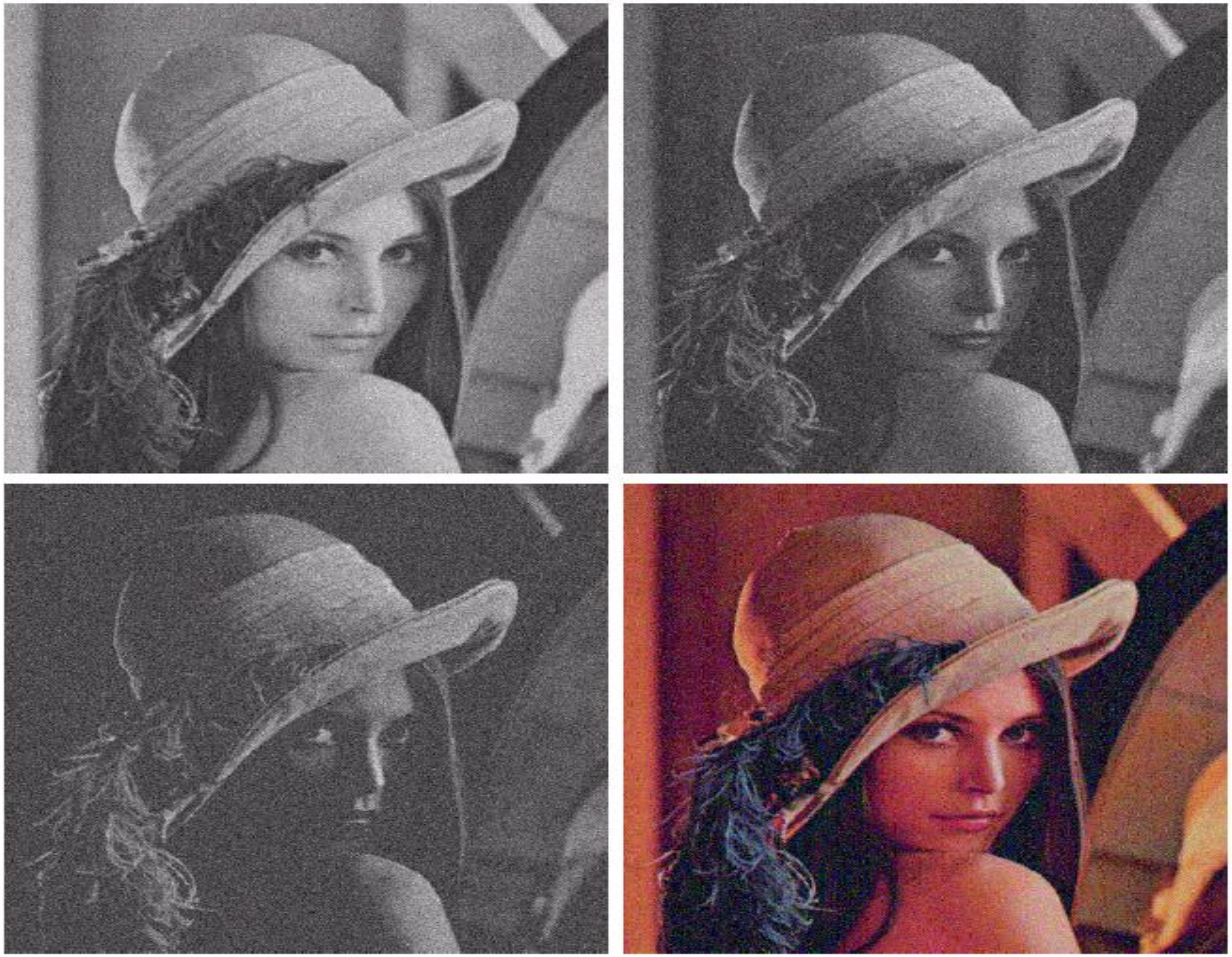
Noise in Color Images

- Noise in color images can be removed through various noise models which we use in Image Restoration in case the noise content of a color image has the same characteristics in each color channel.

- But it is possible for color channels to be affected differently by noise so in this case noise are removed from the image by independently processing each plane

- Remove noise by applying smoothing filters (e.g gaussian, average, median) to each plane individually and then combine the result.

Noise in Color Images



a b
c d

FIGURE 6.48
(a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

Color Image Compression

- Compression is the process of reducing or eliminating redundant and/or irrelevant information

- A compressed image is not directly displayable it must be decompressed before input to a color monitor.

- In case if in a compressed image 1 bit of data represents 230 bits of data in the original image, then compressed image could be transmitted over internet in 1 minute as compared to original image which will take 4 hours to transmit.

a b
c d

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

Any question