UNIT 4
Color Image Processing

Course: Digital Image Processing(EC662)

Course Instructor: Prof. Shashidhar R

Dept.: Electronics and Communication Engg.

CONTENTS

- Color models RGB
- CMY,CMYK,HIS
- Color transformations
- Converting colors from RGB to HIS and HIS to RGB
- Pseudo color image processing

Color Image Processing is divided into two major areas:

- Full-color processing
- Pseudo-color processing

- Full-color processing: images typically are acquired using a full-color sensor, such as a digital camera, or color scanner.
- Pseudo-color processing : The issue is one of assigning color(s) to a particular grayscale intensity or range of intensities.

- Until just a few years ago, most digital color image processing was done at the pseudo- or reduced-color level.
- However, because color sensors and processing hardware have become available at reasonable prices, full-color image processing techniques are now used in a broad range of applications.
- cones are the sensors in the eye responsible for color vision. Detailed experimental evidence has established that the 6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue.
- Approximately 65% of all cones are sensitive to red light, 33% are sensitive to green light, and only about 2% are sensitive to blue.

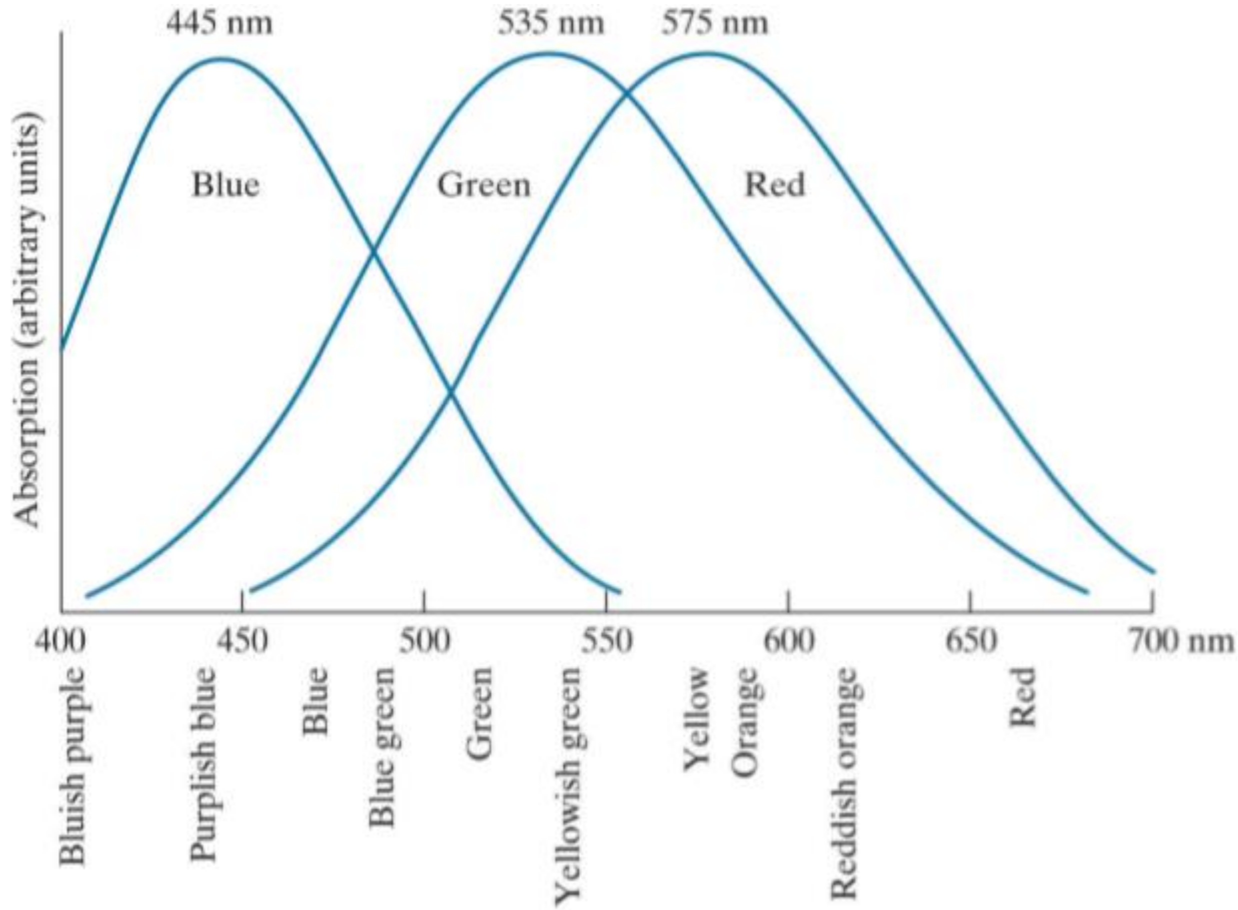


FIGURE 7.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

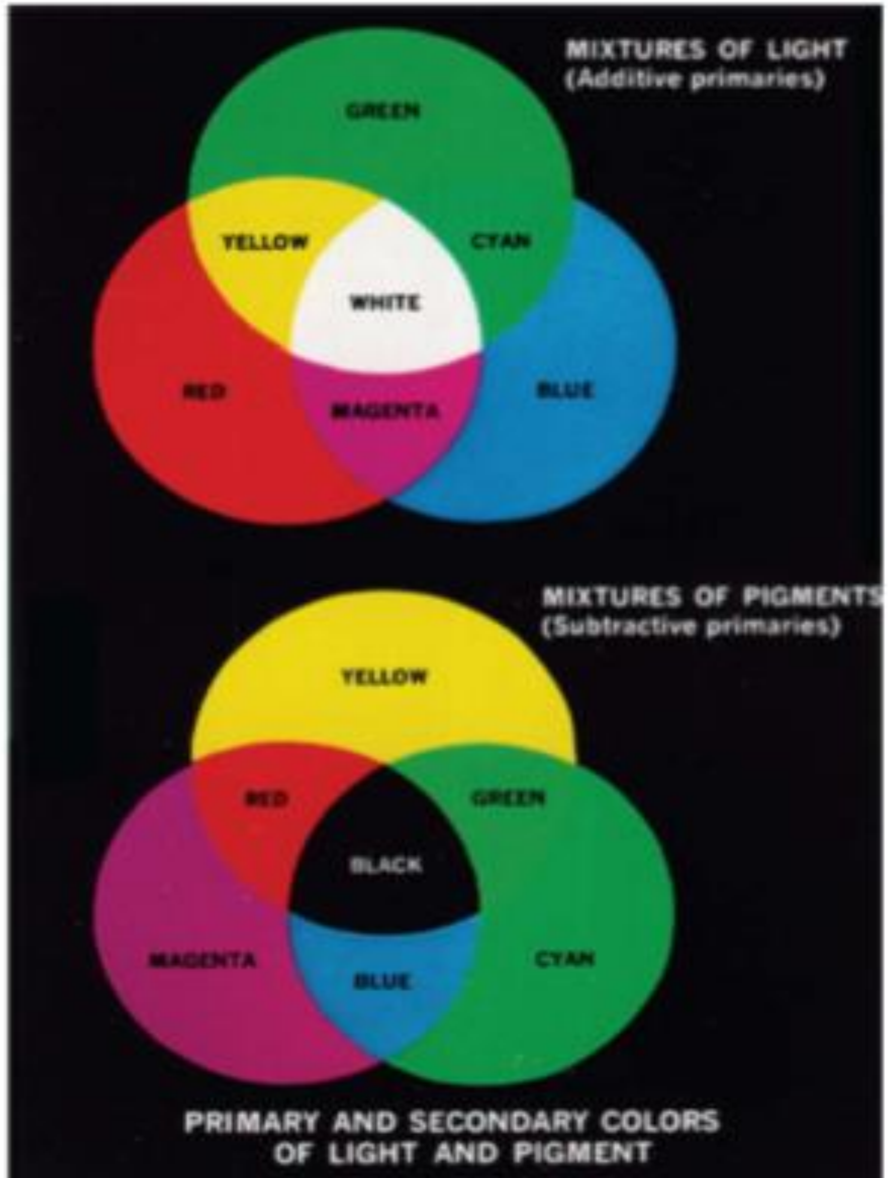


FIGURE 7.4 Primary and secondary colors of light and pigments.

Color Models:

The purpose of a color model (also called a color space or color system) is to facilitate the specification of colors in some standard way.

a color model is a specification of

(1) A coordinate system

(2) a subspace within that system

such that each color in the model is represented by a single point contained in that subspace.

- Most color models in use today are oriented either toward hardware (such as for color monitors and printers) or toward applications, where color manipulation is a goal (the creation of color graphics for animation is an example of the latter).
- In terms of digital image processing, the hardware-oriented models most commonly used in practice are the RGB (red, green, blue) model for color monitors and a broad class of color video cameras;
- CMY (cyan, magenta, yellow)
- CMYK (cyan, magenta, yellow, black) models for color printing;
- the HSI (hue, saturation, intensity) model, which corresponds closely with the way humans describe and interpret color.

The RGB Color Model

- In the RGB model, each color appears in its primary spectral components of red, green, and blue.
- This model is based on a Cartesian coordinate system.
- The color subspace of interest is the cube shown in Fig. 7.7 , in which RGB primary values are at three corners;
- The secondary colors cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin.
- In this model, the grayscale (points of equal RGB values) extends from black to white along the line joining these two points.
- The different colors in this model are points on or inside the cube, and are defined by vectors extending from the origin

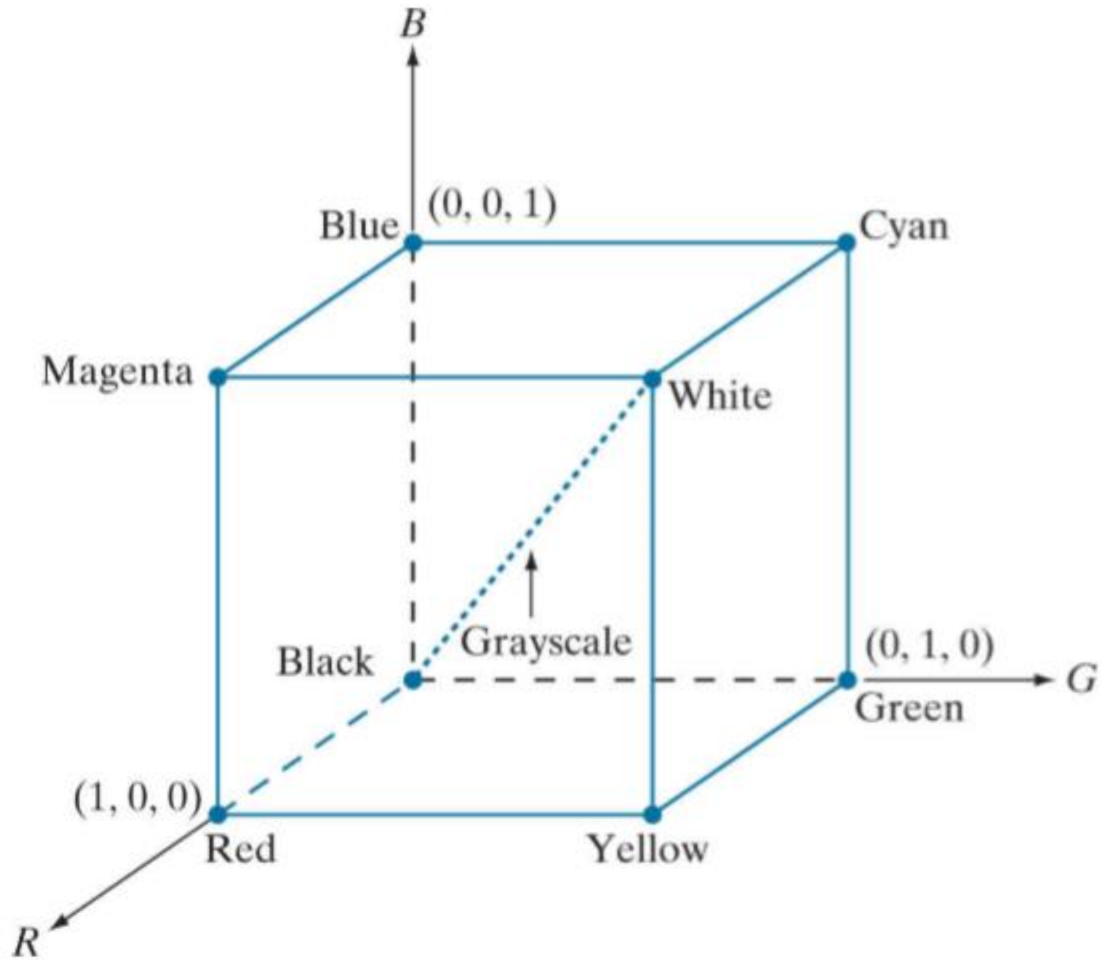


FIGURE 7.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

- For convenience, the assumption is that all color values have been normalized so the cube in Fig.7.7 is the unit cube.
- That is, all values of R, G, and B in this representation are assumed to be in the range [0, 1]. Note that the RGB primaries can be interpreted as unit vectors emanating from the origin of the cube.
- Images represented in the RGB color model consist of three component images, one for each primary color.
- When fed into an RGB monitor, these three images combine on the screen to produce a composite color image.

- The number of bits used to represent each pixel in RGB space is called the pixel depth.
- Consider an RGB image in which each of the red, green, and blue images is an 8-bit image.
- Under these conditions, each RGB color pixel [that is, a triplet of values (R, G, B)] has a depth of 24 bits (3 image planes times the number of bits per plane).
- The term full-color image is used often to denote a 24-bit RGB color image.
- The total number of possible colors in a 24-bit RGB image is $(2^8)^3 = 16{,}777{,}216$

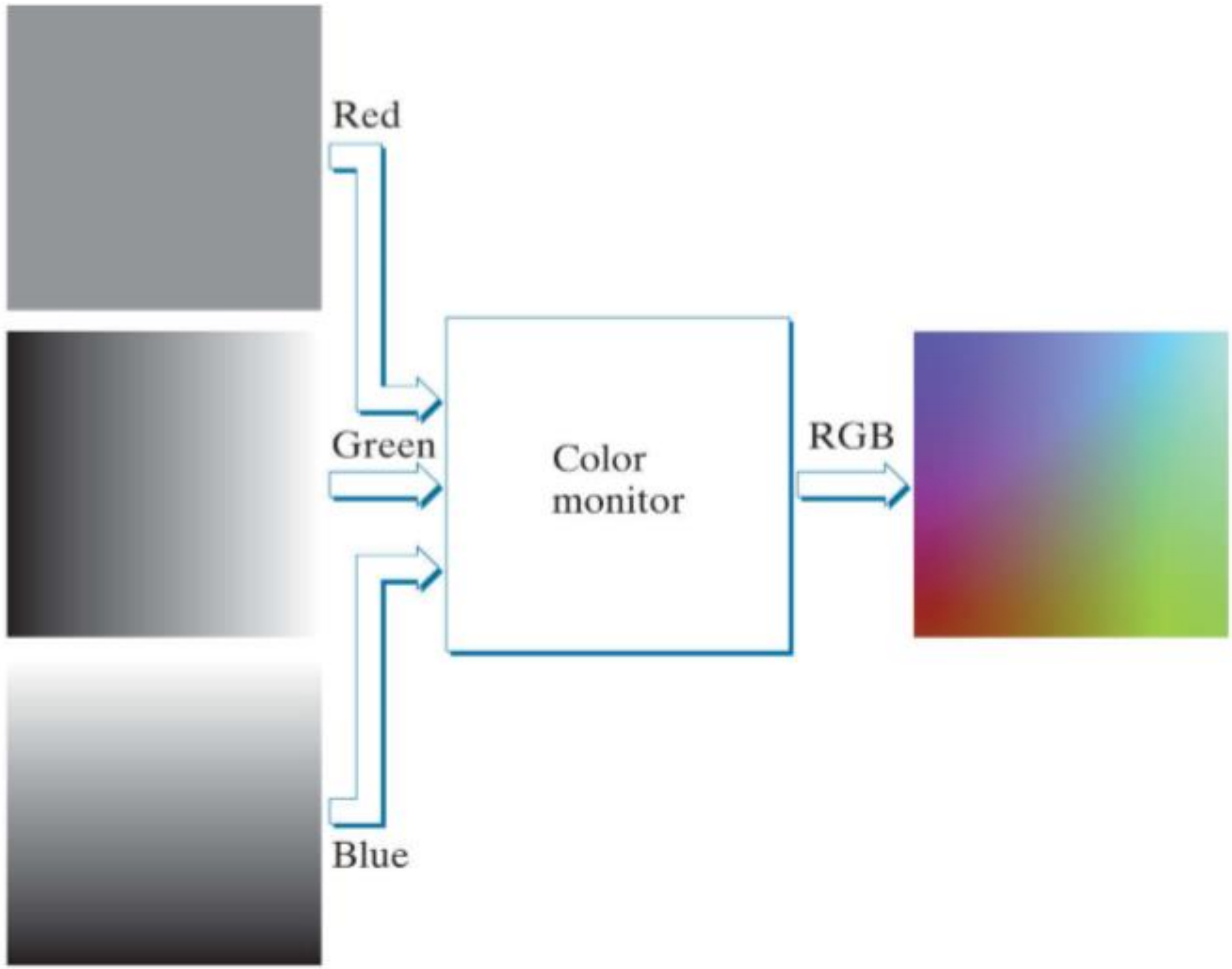
FIGURE 7.8
A 24-bit RGB color cube.

- Figure 7.8 shows the 24-bit RGB color cube corresponding to the diagram in Fig.7.7

- Note also that for digital images, the range of values in the cube are scaled to the numbers representable by the number bits in the images.

- If, as above, the primary images are 8-bit images, the limits of the cube along each axis becomes [0, 255]. Then, for example, white would be at point [255, 255, 255] in the cube.

EXAMPLE 7.1 : Generating a cross-section of the RGB color cube and its three hidden planes.



- FIGURE7.9 (a) Generating the RGB image of the cross-sectional color plane (127, G, B).

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

FIGURE7.9 (b) The three hidden surface planes in the color cube of Fig.7.8 .

- The cube in Fig. 7.8 is a solid, composed of the $(2^8)^3$ colors mentioned in the preceding paragraph.
- A useful way to view these colors is to generate color planes (faces or cross sections of the cube).
- This is done by fixing one of the three colors and allowing the other two to vary. For instance, a cross-sectional plane through the center of the cube and parallel to the GB-plane in Fig. 7.8 is the plane (127, G, B) for G, B = 0,1,2…25

- Figure 7.9(a) shows that an image of this cross-sectional plane is generated by feeding the three individual component images into a color monitor. In the component images, 0 represents black and 255 represents white.
- Observe that each component image into the monitor is a grayscale image.
- The monitor does the job of combining the intensities of these images to generate an RGB image. Figure 7.9(b) shows the three hidden surface planes of the cube in Fig.7.8 , generated in a similar manner.

- Acquiring a color image is the process shown in Fig. 7.9(a) in reverse.
- A color image can be acquired by using three filters, sensitive to red, green, and blue, respectively.
- When we view a color scene with a monochrome camera equipped with one of these filters, the result is a monochrome image whose intensity is proportional to the response of that filter.
- Repeating this process with each filter produces three monochrome images that are the RGB component images of the color scene.
- In practice, RGB color image sensors usually integrate this process into a single device. Clearly, displaying these three RGB component images as in Fig.7.9(a) would yield an RGB color rendition of the original color scene.

The CMY and CMYK Color Models

- cyan, magenta, and yellow are the secondary colors of light or, alternatively, they are the primary colors of pigments. For example, when a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface.
- That is, cyan subtracts red light from reflected white light, which itself is composed of equal amounts of red, green, and blue light.
- Most devices that deposit colored pigments on paper, such as color printers and copiers, require CMY data input or perform an RGB to CMY conversion internally. This conversion is performed using the simple operation

- Equation (7-5) , as well as all other equations in this section, are applied on a pixel-by-pixel basis.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (7\text{-}5)$$

- where the assumption is that all RGB color values have been normalized to the range [0, 1]. Equation (7-5) demonstrates that light reflected from a surface coated with pure cyan does not contain red (that is, in the equation).
- Similarly, pure magenta does not reflect green, and pure yellow does not reflect blue. Equation(7-5) allso reveals that RGB values can be obtained easily from a set of CMY values by subtracting the individual CMY values from 1.

- According to Fig.7.4 , equal amounts of the pigment primaries, cyan, magenta, and yellow, should produce black. In practice, because C, M, and Y inks seldom are pure colors, combining these colors for printing black produces instead a muddy-looking brown.
- So, in order to produce true black (which is the predominant color in printing), a fourth color, black, denoted by K, is added, giving rise to the CMYK color model.
- The black is added in just the proportions needed to produce true black. Thus, when publishers talk about "four-color printing," they are referring to the three CMY colors, plus a portion of black.
- The conversion from CMY to CMYK begins by letting

$$K=\min(C,M,Y) \tag{7-6}$$

- If $K=1$ then we have pure black, with no color contributions, from which it follows that

$$C=0 \quad (7\text{-}7)$$

$$M=0 \quad (7\text{-}8)$$

$$Y=0 \quad (7\text{-}9)$$

The C, M, and Y on the right side of Eqs.(7-6) -(7-12) are in the CMY color system.

The C, M, and Y on the left of Eqs.(7-7) -(7-12) are in the CMYK system.

Otherwise

$$C=(C-K)/(1-K) \quad (7\text{-}10)$$

$$M=(M-K)/(1-K) \quad (7\text{-}11)$$

$$Y=(Y-K)/(1-K) \quad (7\text{-}12)$$

- where all values are assumed to be in the range [0, 1]. The conversions from CMYK back to CMY are:

$$C = C*(1-K)+K \qquad (7\text{-}13)$$

$$M = M*(1-K)+K \qquad (7\text{-}14)$$

$$Y = Y*(1-K)+K \qquad (7\text{-}15)$$

The C, M, Y, and K on the right side of Eqs.(7-13) -(7-15) are in the CMYK color system. The C, M, and Y on the left of these equations are in the CMY system.

The HSI Color Model

- Creating colors in the RGB, CMY, and CMYK models, and changing from one model to the other, is straightforward.
- These color systems are ideally suited for hardware implementations.
- The RGB system matches nicely with the fact that the human eye is strongly perceptive to red, green, and blue primaries.
- Unfortunately, the RGB, CMY, and other similar color models are not well suited for describing colors in terms that are practical for human interpretation.
- For example, one does not refer to the color of an automobile by giving the percentage of each of the primaries composing its color.

- Furthermore, we do not think of color images as being composed of three primary images that combine to form a single image.
- When humans view a color object, we describe it by its hue, saturation, and brightness.
- hue is a color attribute that describes a pure color (pure yellow, orange, or red), whereas saturation gives a measure of the degree to which a pure color is diluted by white light.
- Brightness is a subjective descriptor that is practically impossible to measure.
- It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.
- We do know that intensity (gray level) is a most useful descriptor of achromatic images.

- This quantity definitely is measurable and easily interpretable.
- The model we are about to present, called the HSI (hue, saturation, intensity) color model, decouples the intensity component from the color-carrying information (hue and saturation) in a color image.
- As a result, the HSI model is a useful tool for developing image processing algorithms based on color descriptions that are natural and intuitive to humans, who, after all, are the developers and users of these algorithms.
- We can summarize by saying that RGB is ideal for image color generation (as in image capture by a color camera or image display on a monitor screen), but its use for color description is much more limited.

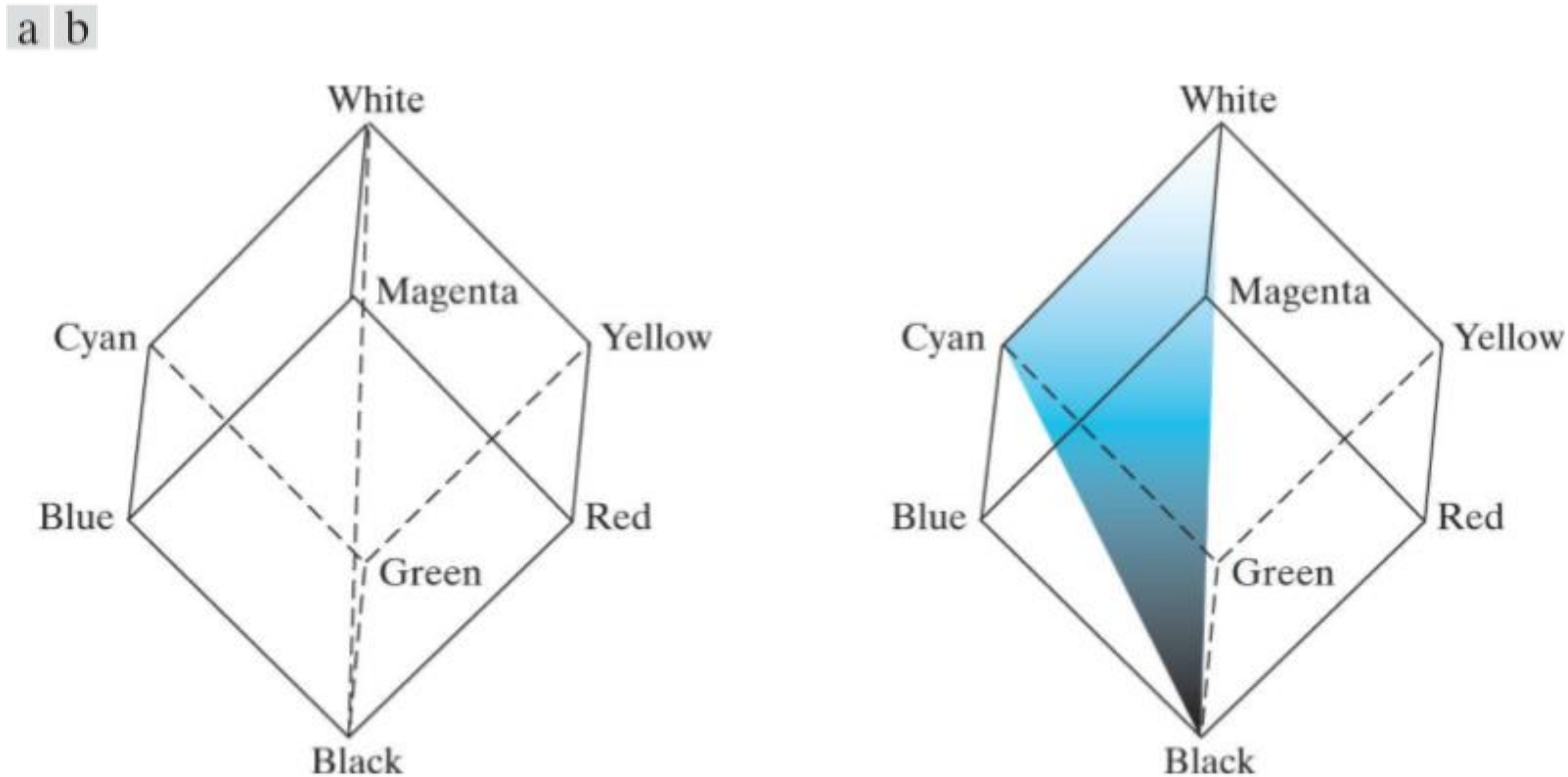


FIGURE 7.10 Conceptual relationships between the RGB and HSI color models.

- We know from Example7.10 that an RGB color image is composed three gray-scale intensity images (representing red, green, and blue), so it should come as no surprise that we can to
- This becomes clear if we take the color cube from Fig. 7.7 and stand it on the black, (0, 0, 0), vertex, with the white, (1, 1, 1), vertex directly above it [see Fig.7.10(a)]. extract intensity from an RGB image.
- As noted in our discussion of Fig.7.7 , the intensity (gray) scale is along the line joining these two vertices. In Figs.7.10(a) and (b), the line (intensity axis) joining the black and white vertices is vertical.
- Thus, if we wanted to determine the intensity component of any color point in Fig. 7.10 ,

- we would simply define a plane that contains the color point and, at the same time, is perpendicular to the intensity axis.
- The intersection of the plane with the intensity axis would give us a point with intensity value in the range [0, 1].
- A little thought would reveal that the saturation (purity) of a color increases as a function of distance from the intensity axis.
- In fact, the saturation of points on the intensity axis is zero, as evidenced by the fact that all points along this axis are gray.
- Hue can be determined from an RGB value also.
- To see how, consider Fig.7.10(b) , which shows a plane defined by three points (black, white, and cyan).

- The fact that the black and white points are contained in the plane tells us that the intensity axis also is contained in the plane.
- Furthermore, we see that all points contained in the plane segment defined by the intensity axis and the boundaries of the cube have the same hue (cyan in this case).
- all colors generated by three colors lie in the triangle defined by those colors.
- If two of those points are black and white, and the third is a color point, all points on the triangle would have the same hue, because the black and white components cannot change the hue (of course, the intensity and saturation of points in this triangle would be different).

- By rotating the shaded plane about the vertical intensity axis, we would obtain different hues. From these concepts, we arrive at the conclusion that the hue, saturation, and intensity values required to form the HSI space can be obtained from the RGB color cube.
- That is, we can convert any RGB point to a corresponding point in the HSI color space by working out the formulas that describe the reasoning outlined in the preceding discussion.
- The key point regarding the cube arrangement in Fig. 7.10 , and its corresponding HSI color space, is that the HSI space is represented by a vertical intensity axis, and the locus of color points that lie on planes perpendicular to that axis.

- As the planes move up and down the intensity axis, the boundaries defined by the intersection of each plane with the faces of the cube have either a triangular or a hexagonal shape.
- This can be visualized much more readily by looking at the cube straight down its grayscale axis, as shown in Fig. 7.11(a) .
- We see that the primary colors are separated by 120 degree. The secondary colors are 60 degree from the primaries, which means that the angle between secondaries is 120 degree also
- Figure 7.11(b) shows the same hexagonal shape and an arbitrary color point (shown as a dot).

- The hue of the point is determined by an angle from some reference point.
- Usually (but not always) an angle of from the red axis designates 0 hue, and the hue increases counterclockwise from there.
- The saturation (distance from the vertical axis) is the length of the vector from the origin to the point.
- Note that the origin is defined by the intersection of the color plane with the vertical intensity axis.
- The important components of the HSI color space are the vertical intensity axis, the length of the vector to a color point, and the angle this vector makes with the red axis.
- Therefore, it is not unusual to see the HSI planes defined in terms of the hexagon just discussed, a triangle, or even a circle, as Figs.7.11(c) and (d) show.

- The shape chosen does not matter because any one of these shapes can be warped into one of the other two by a geometric transformation.
- Figure7.12 shows the HSI model based on color triangles, and on circles.

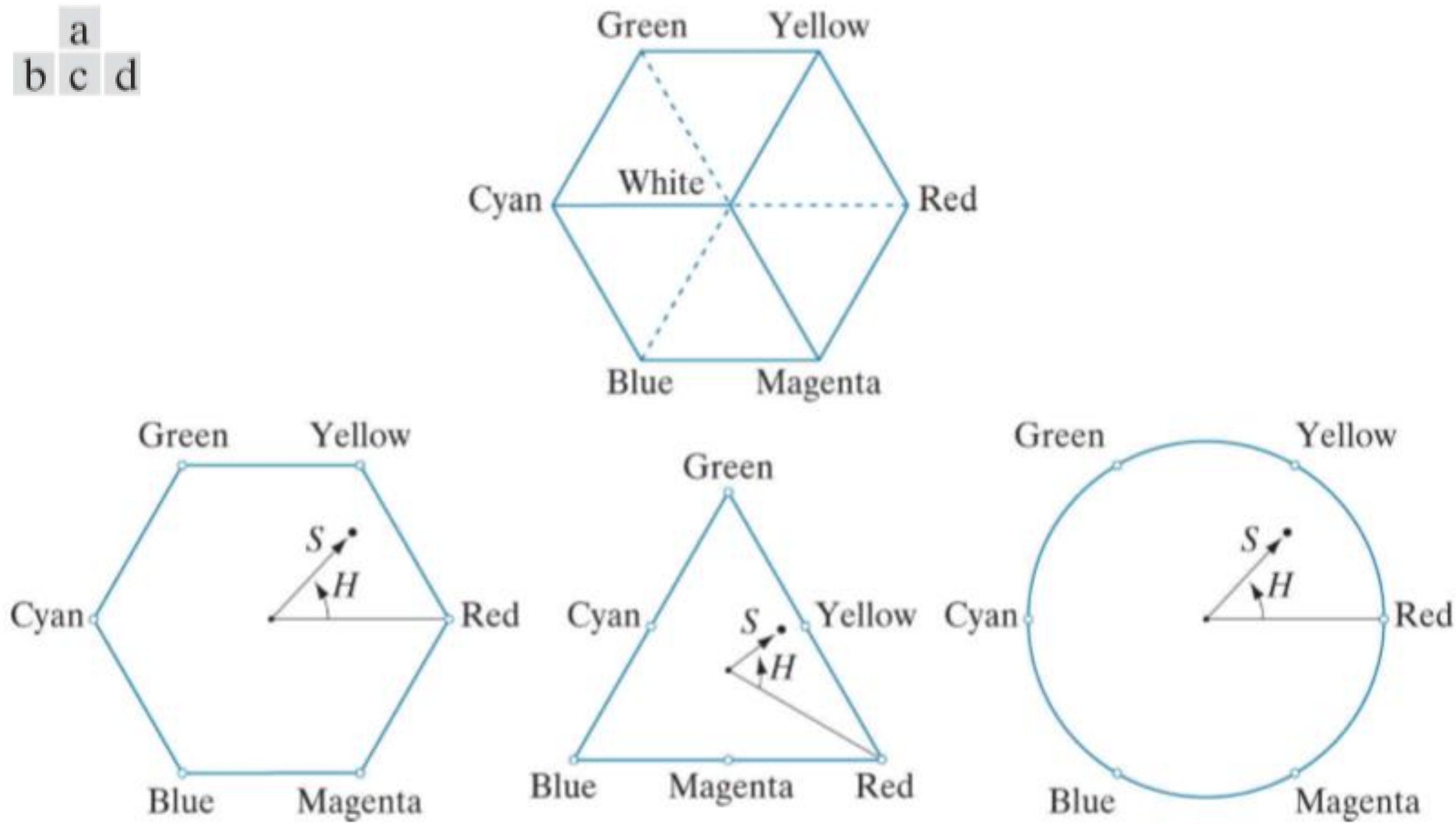


FIGURE 7.11 Hue and saturation in the HSI color model. The dot is any color point. The angle from the red axis gives the hue. The length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

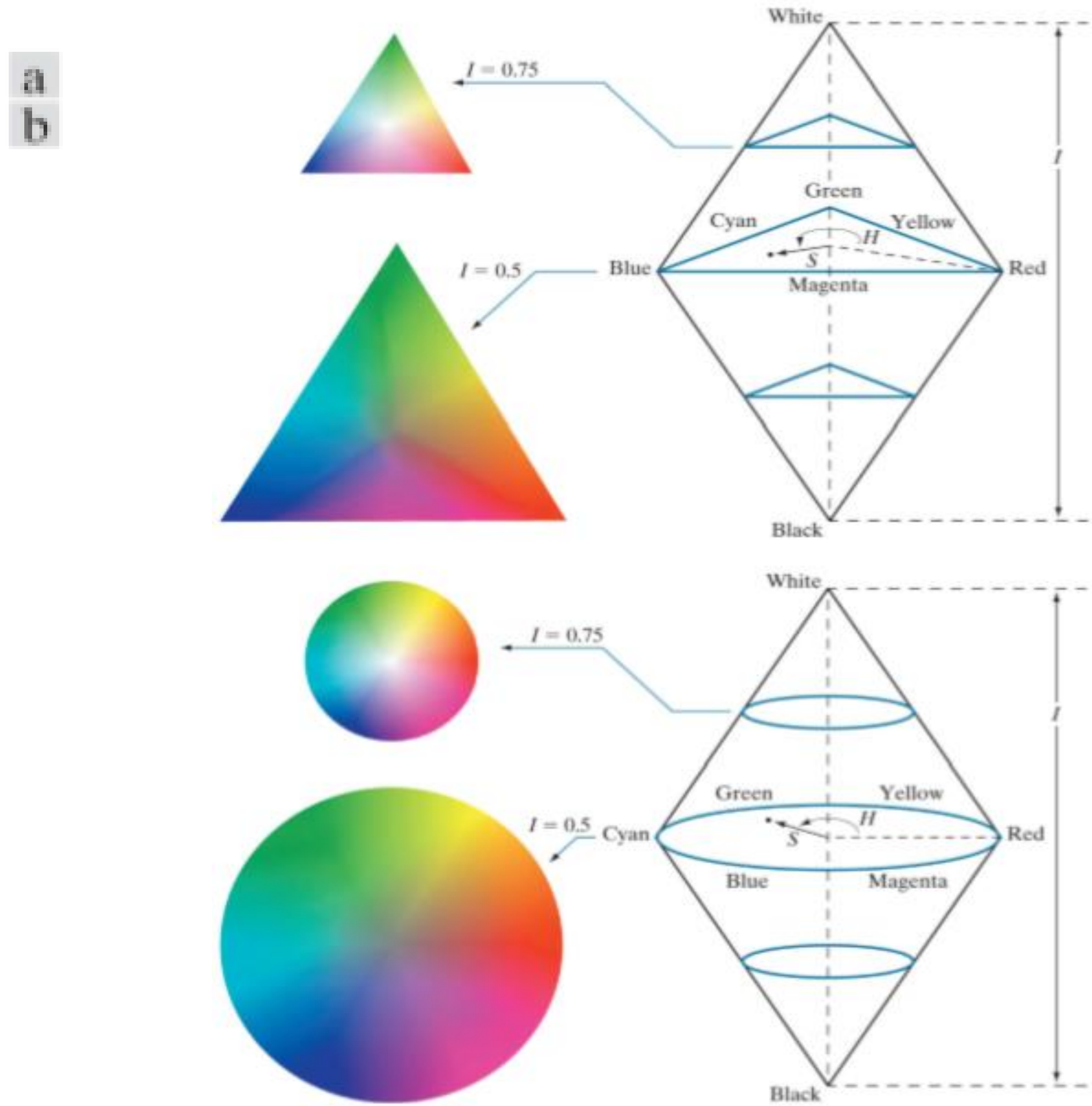


FIGURE 7.12 The HSI color model based on (a) triangular, and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis.

Converting Colors from RGB to HSI

- Given an image in RGB color format, the H component of each RGB pixel is obtained using the equation
- $$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B\ \ > G \end{cases} \qquad (7\text{-}16)$$

- With It is good practice to add a small number in the denominator of this expression to avoid dividing by 0 when R=G=B.
- Note that when all RGB components are equal, Eq. (7-18) gives In addition, the conversion from HSI back to RGB in Eqs.(7-20) through (7-30) will give expected, because, when we are dealing with a grayscale image.

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\} \qquad (7\text{-}17)$$

- The saturation component is given by

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \qquad (7\text{-}18)$$

- Finally, the intensity component is obtained from the equation

$$I = \frac{1}{3}(R+G+B) \qquad (7\text{-}19)$$

- These equations assume that the RGB values have been normalized to the range [0, 1], and that angle u is measured with respect to the red axis of the HSI space, as in Fig. 7.11
- Hue can be normalized to the range [0, 1] by dividing by 360^0 all values resulting from Eq. (7-16) . The other two HSI components already are in this range if the given RGB values are in the interval [0, 1].
- The results in Eqs. (7-16) through (7-19) can be derived from the geometry in Figs.7.10 and 7.11.The derivation is tedious and would not add significantly to the present discussion.

Converting Colors from HSI to RGB

- Given values of HSI in the interval [0, 1], we now want to find the corresponding RGB values in the same range.
- The applicable equations depend on the values of H.
- There are three sectors of interest, corresponding to the 120 degree intervals in the separation of primaries (see Fig. 7.11)
- We begin by multiplying H by 360^0 which returns the hue to its original range of $[0^0, 360^0]$
- **RG Sector (**$0^0 \leq H < 120^0$): When H is in this sector, the RGB components are given by the equations

$$B = I(1 - S) \tag{7-20}$$

$$R = I\left[1 + \frac{S \cos H}{\cos(60^\circ - H)}\right] \tag{7-21}$$

and

$$G = 3I - (R + B) \tag{7-22}$$

GB Sector: $120^0 \le H < 240^0$

If the given value of H is in this sector, we first subtract 120^0 from it.

$$H = H - 120° \tag{7-23}$$

Then, the RGB components are

$$R = I(1 - S) \tag{7-24}$$

$$G = I\left[1 + \frac{S \cos H}{\cos(60° - H)}\right] \tag{7-25}$$

and

$$B = 3I - (R + G) \tag{7-26}$$

BR sector $(240° \le H \le 360°)$: Finally, if *H* is in this range, we subtract 240° from it:

$$H = H - 240° \tag{7-27}$$

Then, the RGB components are

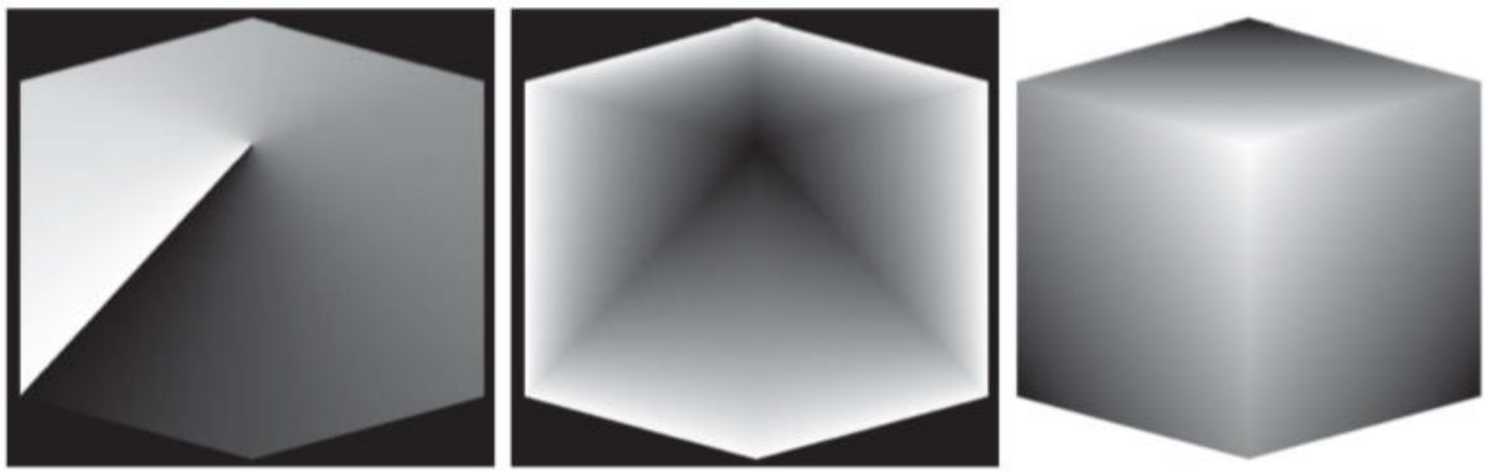
$$G = I(1 - S) \tag{7-28}$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right] \tag{7-29}$$

and

$$R = 3I - (G + B) \tag{7-30}$$

EXAMPLE 7.2: The HSI values corresponding to the image of the RGB color cube.



a b c

FIGURE7.13 HSI components of the image in Fig.7.8 : (a) hue, (b) saturation, and (c) intensity images

- Figure 7.13 shows the hue, saturation, and intensity images for the RGB values in Fig.7.8
- Figure 7.13(a) is the hue image. Its most distinguishing feature is the discontinuity in value along a 45 degree line in the front (red) plane of the cube.
- To understand the reason for this discontinuity refer to Fig.7.8 , draw a line from the red to the white vertices of the cube, and select a point in the middle of this line.
- Starting at that point, draw a path to the right, following the cube around until you return to the starting point.
- The major colors encountered in this path are yellow, green, cyan, blue, magenta, and back to red. According to Fig.7.11
- The values of hue along this path should increase from 0 degree to 360 degree (i.e., from the lowest to highest possible values of hue).

- This is precisely what Fig.7.13(a) shows, because the lowest value is represented as black and the highest value as white in the grayscale. In fact, the hue image was originally normalized to the range [0, 1] and then scaled to 8 bits; that is, we converted it to the range [0, 255], for display.
- The saturation image in Fig.7.13(b) shows progressively darker values toward the white vertex of the RGB cube, indicating that colors become less and less saturated as they approach white. Finally, every pixel in the intensity image shown in Fig. 7.13(c) is the average of the RGB values at the corresponding pixel in Fig.7.8

Pseudocolor Image Processing

- Pseudocolor (sometimes called false color) image processing consists of assigning colors to gray values based on a specified criterion.
- The term pseudo or false color is used to differentiate the process of assigning colors to achromatic images from the processes associated with true color images,
- The principal use of pseudocolor is for human visualization and interpretation of grayscale events in an image or sequence of images.
- As noted at the beginning of this chapter, one of the principal motivations for using color is the fact that humans can discern thousands of color shades and intensities, compared to less than two dozen shades of gray.

Intensity Slicing and Color Coding

- The techniques of intensity (sometimes called density) slicing and color coding are the simplest and earliest examples of pseudocolor processing of digital images.
- If an image is interpreted as a 3-D function the method can be viewed as one of placing planes parallel to the coordinate plane of the image; each plane then "slices" the function in the area of intersection.
- Figure 7.16 shows an example of using a plane at $f(x,y)=l_i$ to slice the image intensity function into two levels.
- If a different color is assigned to each side of the plane in Fig. 7.16 , any pixel whose intensity level is above the plane will be coded with one color, and any pixel below the plane will be coded with the other.

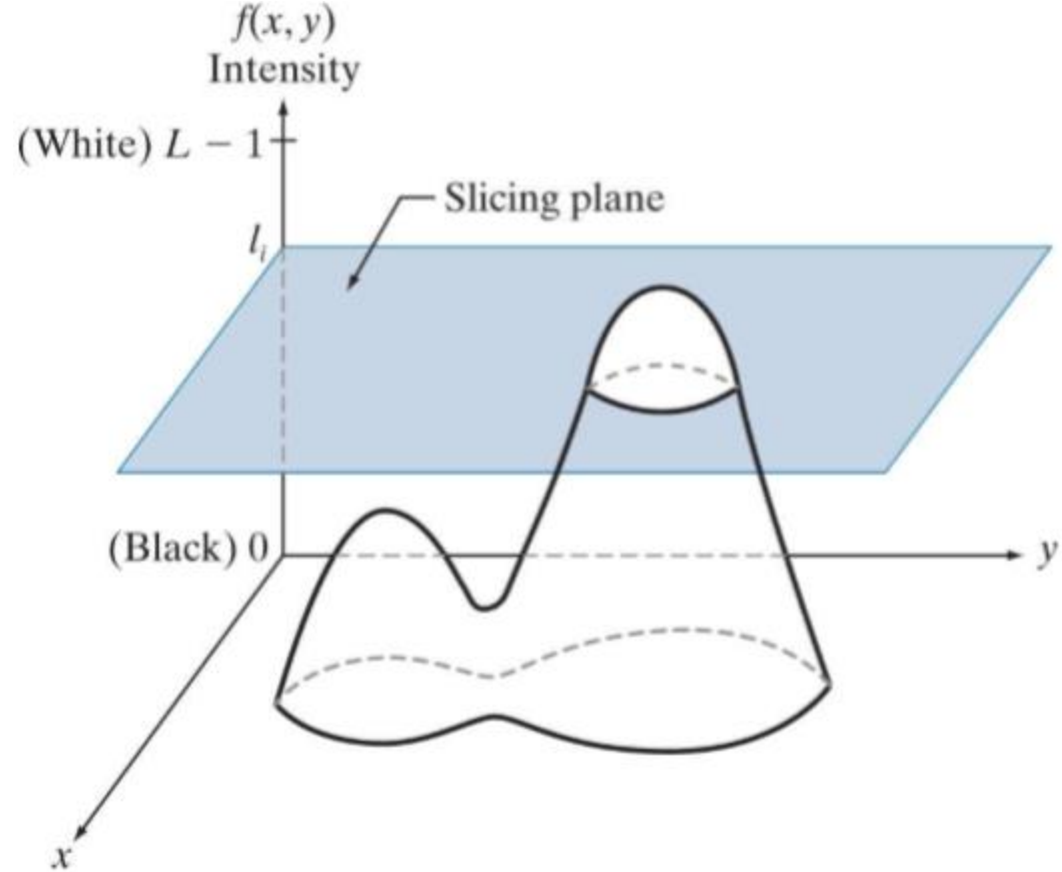


FIGURE 7.16

Graphical interpretation of the intensity-slicing technique

- Levels that lie on the plane itself may be arbitrarily assigned one of the two colors, or they could be given a third color to highlight all the pixels at that level.
- The result is a two- (or three-) color image whose relative appearance can be controlled by moving the slicing plane up and down the intensity axis.
- In general, the technique for multiple colors may be summarized as follows.
- Let [0,L-1] represent the grayscale, let level represent black [f(x,y) = 0] and level l_{L-1} represent white [f(x,y) = L-1].
- Suppose that P planes perpendicular to the intensity axis are defined at levels $l_1, l_2, l_3 \ldots l_p$.
-

- Then, assuming that 0<p<L-1. the P planes partition the grayscale into P+1 intervals,I1,I2,I3…IP+1 Intensity to color assignments at each pixel location (x, y) are made according to the equation,

$$\text{if } f(x,y) \in I_k, \text{ let } f(x,y) = c_k \qquad (7\text{-}35)$$

- where C_k is the color associated with the kth intensity interval I_k, defined by the planes at l=k-1 and l=k
- Figure 7.16 is not the only way to visualize the method just described. Figure 7.17 shows an equivalent approach. According to the mapping in this figure, any image intensity below level is assigned one color, and any level above is assigned another. When more partitioning levels are used, the mapping function takes on a staircase form.

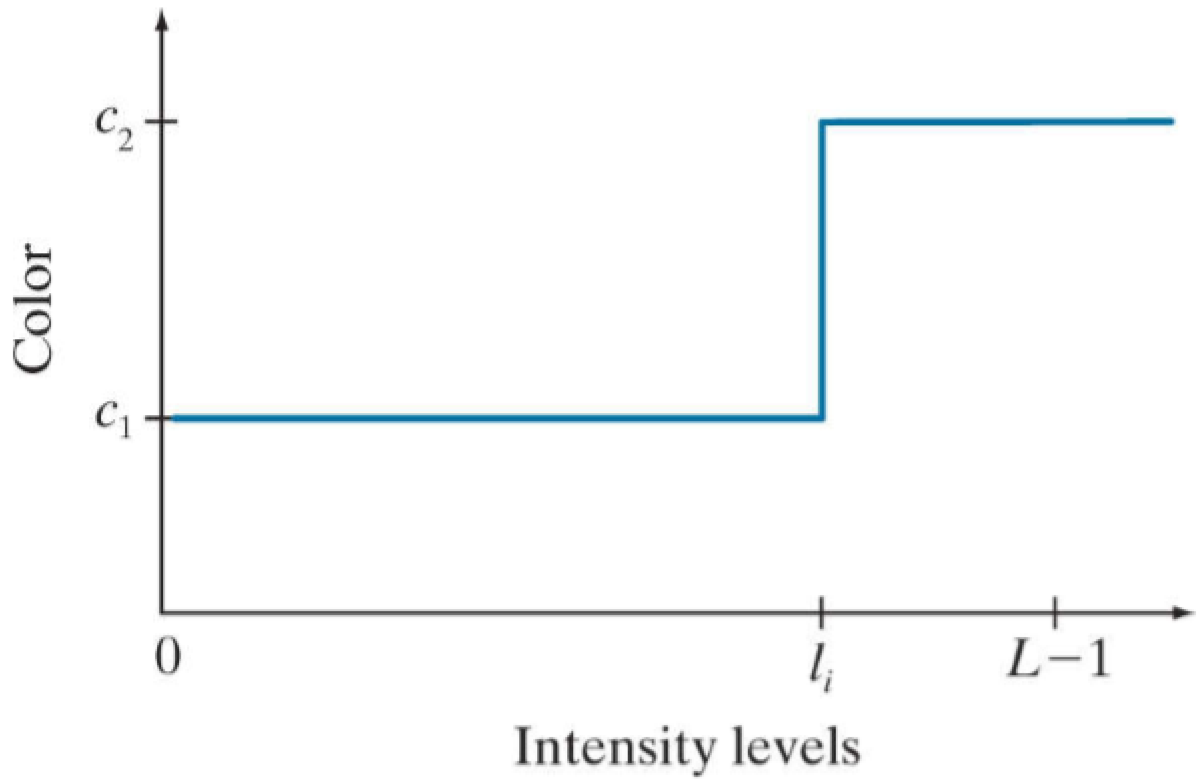


FIGURE 7.17

An alternative representation of the intensity-slicing technique.

EXAMPLE 7.3: Intensity slicing and color coding.

A simple but practical use of intensity slicing is shown in Fig.7.18 . Figure7.18(a) is a grayscale image of the Picker Thyroid Phantom (a radiation test pattern), and Fig.7.18(b) is the result of intensity slicing this image into eight colors.

a b

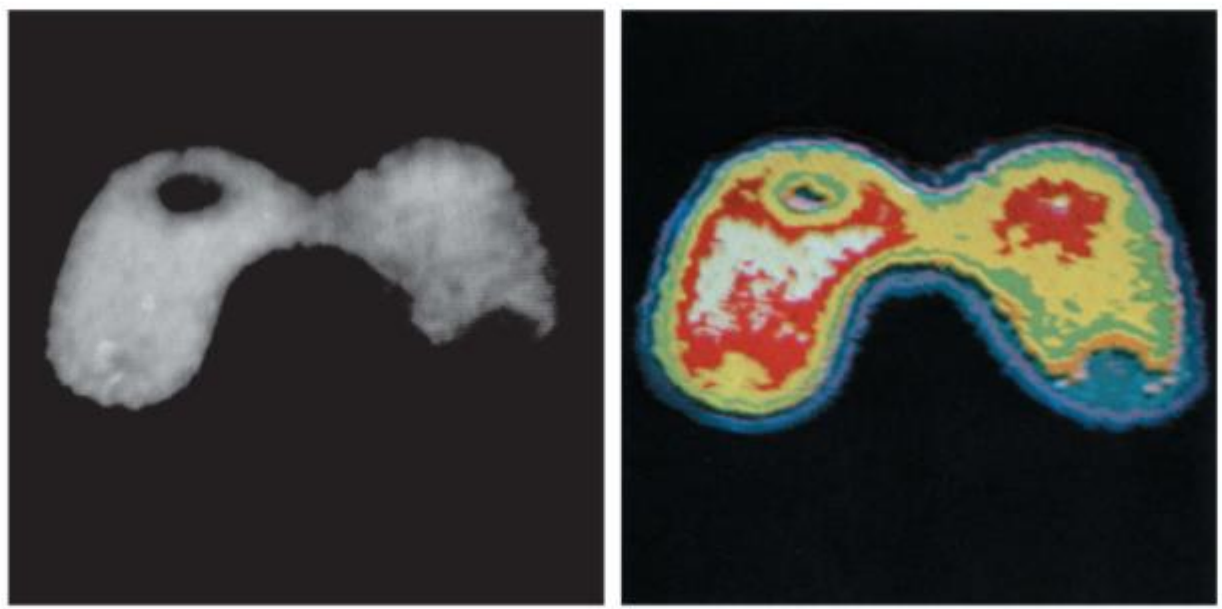


FIGURE7.18 (a) Grayscale image of the Picker Thyroid Phantom. (b) Result of intensity slicing using eight colors.

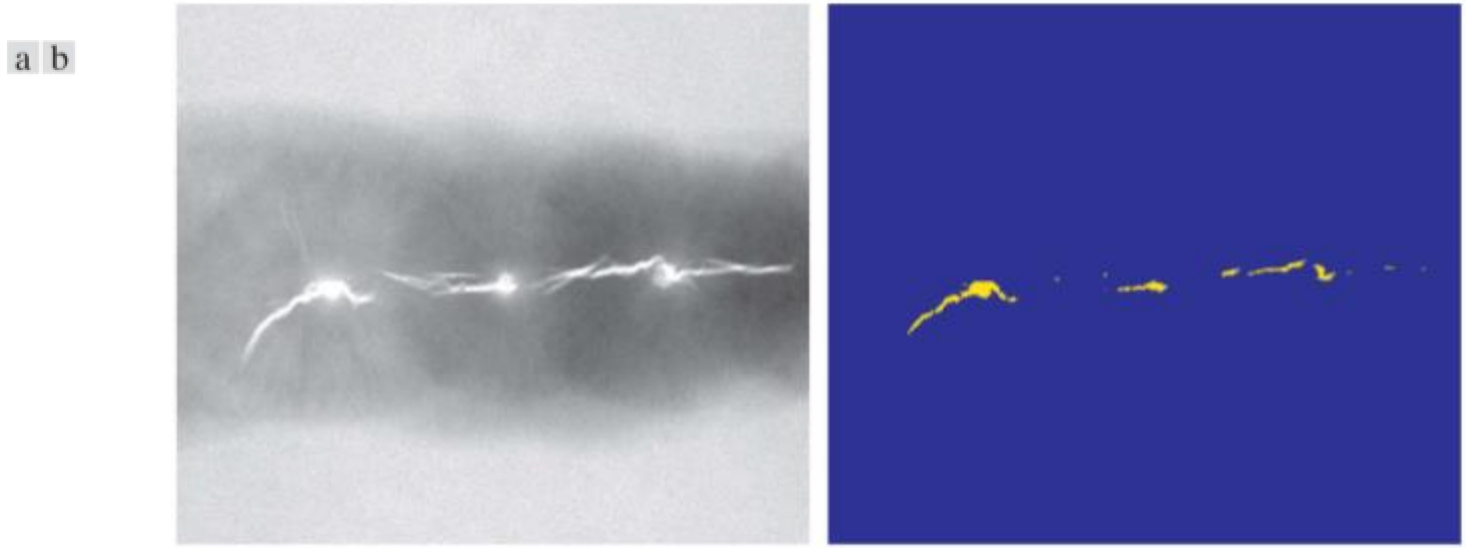


FIGURE7.19 (a) X-ray image of a weld. (b) Result of color coding

- Fig.7.19(a) shows an X-ray image of a weld (the broad, horizontal dark region) containing several cracks and porosities (the bright streaks running horizontally through the middle of the image).
- When there is a porosity or crack in a weld, the full strength of the X-rays going through the object saturates the imaging sensor on the other side of the object.

- Thus, intensity values of 255 in an 8-bit image coming from such a system automatically imply a problem with the weld.
- If human visual analysis is used to inspect welds (still a common procedure today), a simple color coding that assigns one color to level 255 and another to all other intensity levels can simplify the inspector's job considerably.
- Figure7.19(b) shows the result
- No explanation is required to arrive at the conclusion that human error rates would be lower if images were displayed in the form of Fig.7.19(b) , instead of the form in Fig.7.19(a)
- In other words, if an intensity value, or range of values, one is looking for is known, intensity slicing is a simple but powerful aid in visualization, especially if numerous images have to be inspected on a routine basis.

EXAMPLE 7.4: Use of color to highlight rainfall levels.

- Measurement of rainfall levels, especially in the tropical regions of the Earth, is of interest in diverse applications dealing with the environment.
- Accurate measurements using ground-based sensors are difficult and expensive to acquire, and total rainfall figures are even more difficult to obtain because a significant portion of precipitation occurs over the ocean.
- One approach for obtaining rainfall figures remotely is to use satellites.
- The TRMM (Tropical Rainfall Measuring Mission) satellite utilizes, among others, three sensors specially designed to detect rain: a precipitation radar, a microwave imager, and a visible and infrared scanner

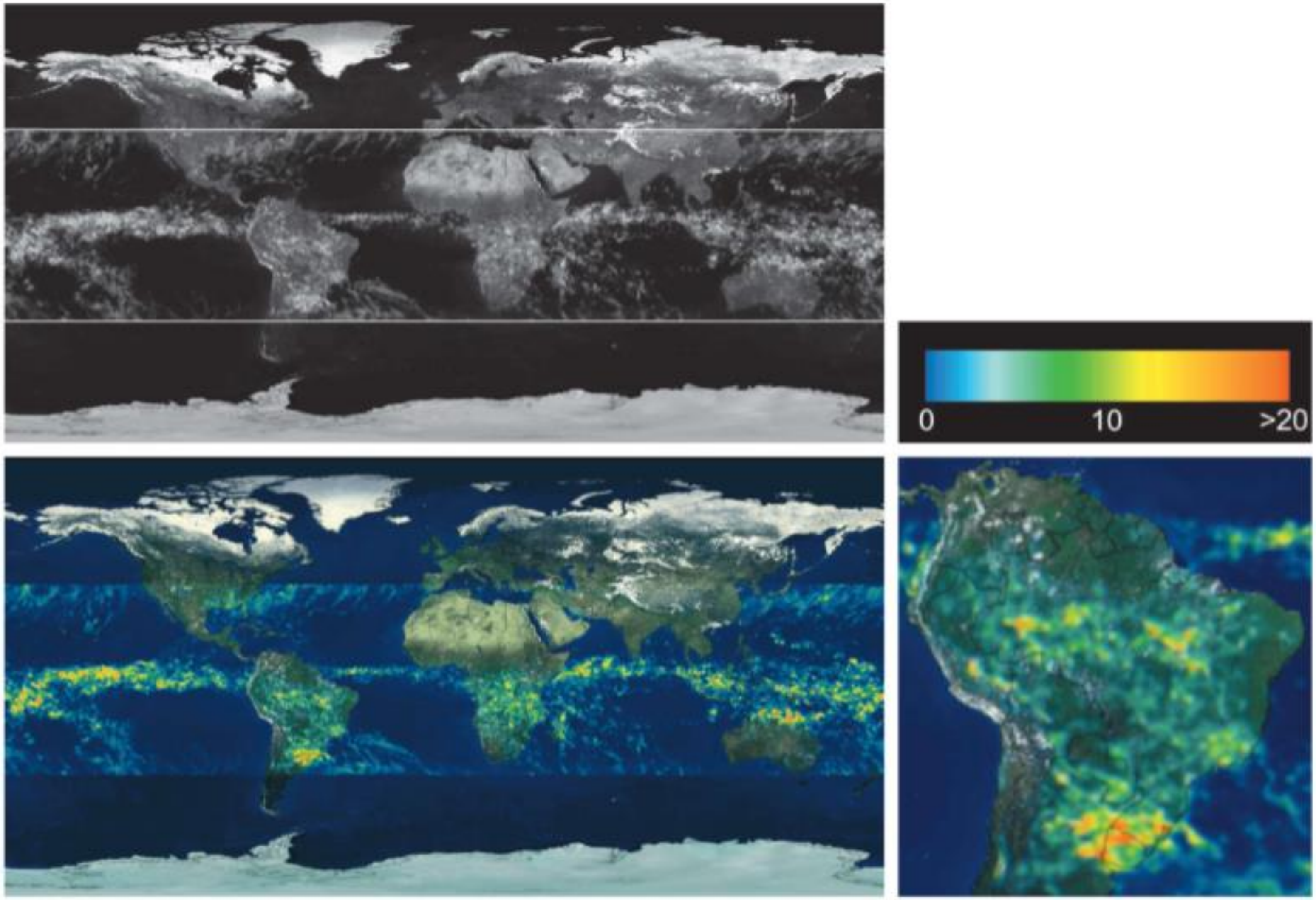


FIGURE 7.20

(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.

- The results from the various rain sensors are processed, resulting in estimates of average rainfall over a given time period in the area monitored by the sensors.
- From these estimates, it is not difficult to generate grayscale images whose intensity values correspond directly to rainfall, with each pixel representing a physical land area whose size depends on the resolution of the sensors.
- Such an intensity image is shown in Fig. 7.20(a) , where the area monitored by the satellite is the horizontal band highlighted in the middle of the picture (these are tropical regions).
- In this particular example, the rainfall values are monthly averages (in inches) over a three-year period.
- Visual examination of this picture for rainfall patterns is difficult and prone to error. However, suppose that we code intensity levels from 0 to 255 using the colors shown in Fig.7.20(b)

- In this mode of intensity slicing, each slice is one of the colors in the color band. Values toward the blues signify low values of rainfall, with the opposite being true for red. Note that the scale tops out at pure red for values of rainfall greater than 20 inches.
- Figure 7.20(c) shows the result of color coding the grayscale image
- The results are much easier to interpret, as shown in this figure and in the zoomed area of Fig.7.20(d) . with the color map just discussed.
- In addition to providing global coverage, this type of data allows meteorologists to calibrate ground-based rain monitoring systems with greater precision than ever before.

Intensity to Color Transformations

- Other types of transformations are more general, and thus are capable of achieving a wider range of pseudocolor enhancement results than the simple slicing technique discussed in the preceding section.
- Figure 7.21 shows an approach that is particularly attractive. Basically, the idea underlying this approach is to perform three independent transformations on the intensity of input pixels.
- The three results are then fed separately into the red, green, and blue channels of a color monitor.
- This method produces a composite image whose color content is modulated by the nature of the transformation functions.

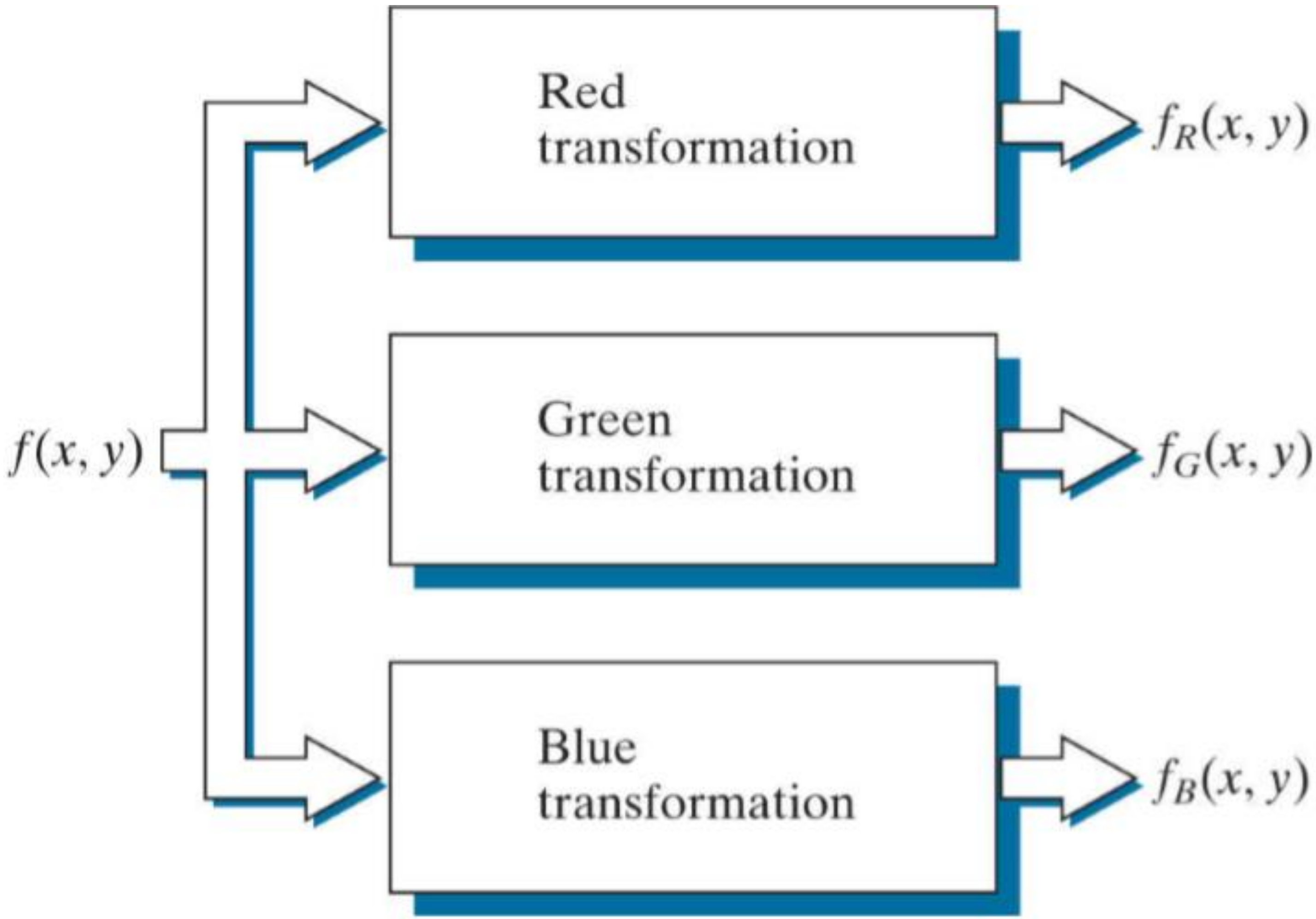


FIGURE 7.21

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

EXAMPLE 7.5: Using pseudocolor to highlight explosives in X-ray images.

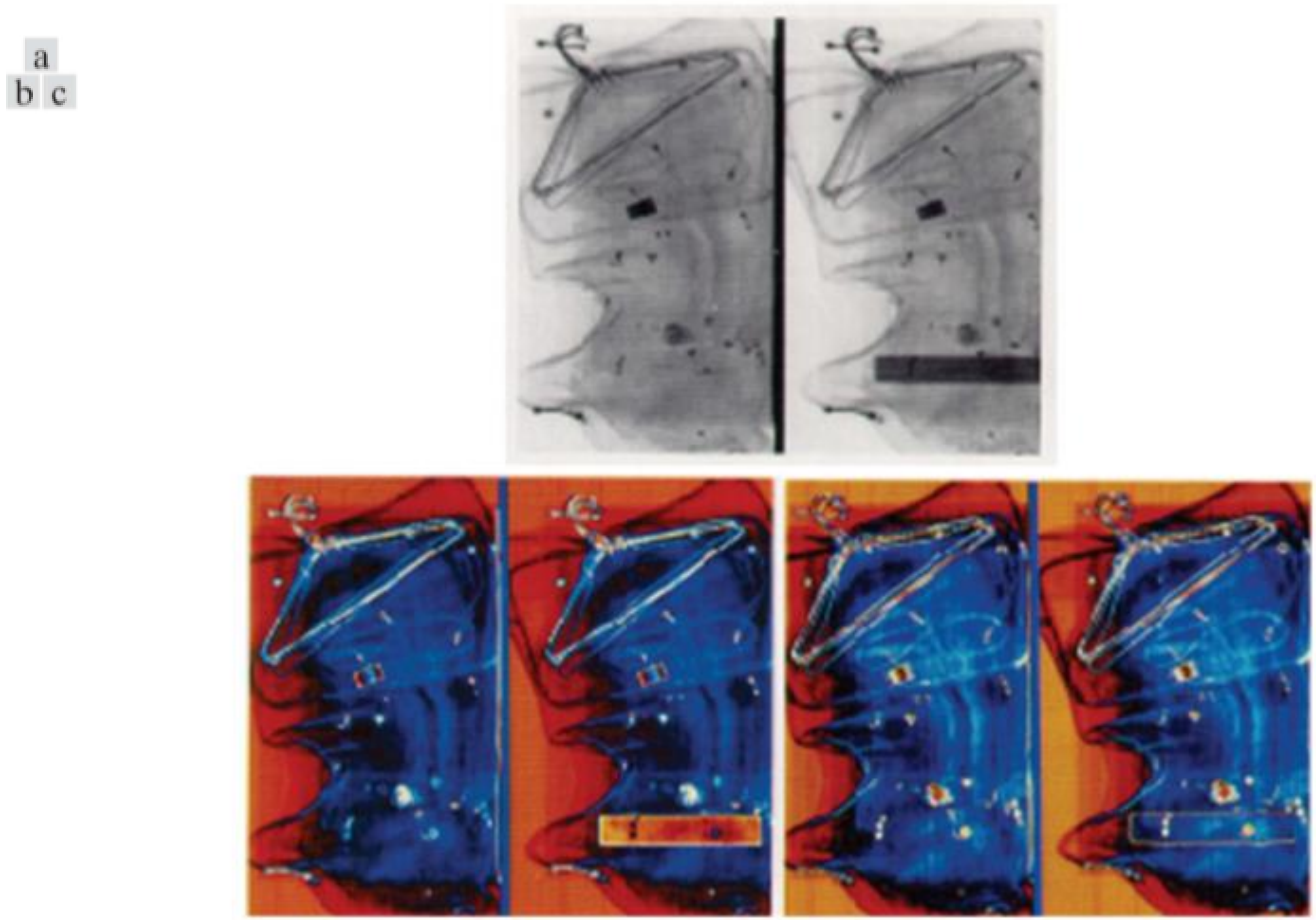


FIGURE7.22 Pseudocolor enhancement by using the gray level to color transformations in Fig.7.23

- Figure 7.22(a) shows two monochrome images of luggage obtained from an airport X-ray scanning system.
- The image on the left contains ordinary articles.
- The image on the right contains the same articles, as well as a block of simulated plastic explosives.
- The purpose of this example is to illustrate the use of intensity to color transformations to facilitate detection of the explosives.
- Figure 7.23 shows the transformation functions used.
- These sinusoidal functions contain regions of relatively constant value around the peaks as well as regions that change rapidly near the valleys.

- Changing the phase and frequency of each sinusoid can emphasize (in color) ranges in the grayscale.
- For instance, if all three transformations have the same phase and frequency, the output will be a grayscale image.
- A small change in the phase between the three transformations produces little change in pixels whose intensities correspond to peaks in the sinusoids, especially if the sinusoids have broad profiles (low frequencies).
- Pixels with intensity values in the steep section of the sinusoids are assigned a much stronger color content as a result of significant differences between the amplitudes of the three sinusoids caused by the phase displacement between them.

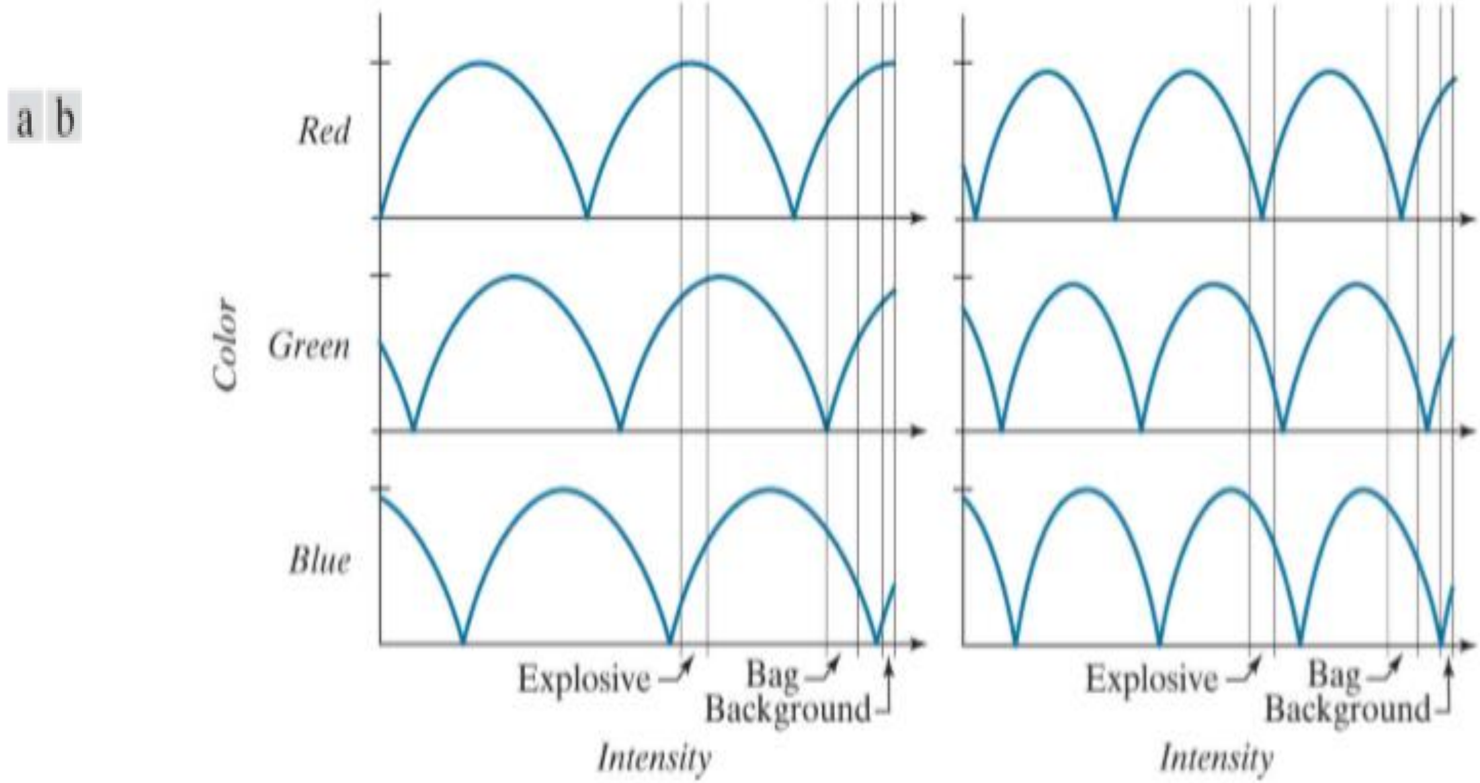


FIGURE7.23

Transformation functions used to obtain the pseudocolor images in Fig. 7.22 .

- The image in Fig. 7.22(b) was obtained using the transformation functions in Fig.7.23(a) , which shows the gray-level bands corresponding to the explosive, garment bag, and background, respectively.
- Note that the explosive and background have quite different intensity levels, but they were both coded with approximately the same color as a result of the periodicity of the sine waves.
- The image in Fig. 7.22(c) was obtained with the transformation functions in Fig.7.23(b).
- In this case, the explosives and garment bag intensity bands were mapped by similar transformations, and thus received essentially the same color assignments.
- Note that this mapping allows an observer to “see” through the explosives. The background mappings were about the same as those used for Fig. 7.22(b) , producing almost identical color assignments for the two pseudocolor images.

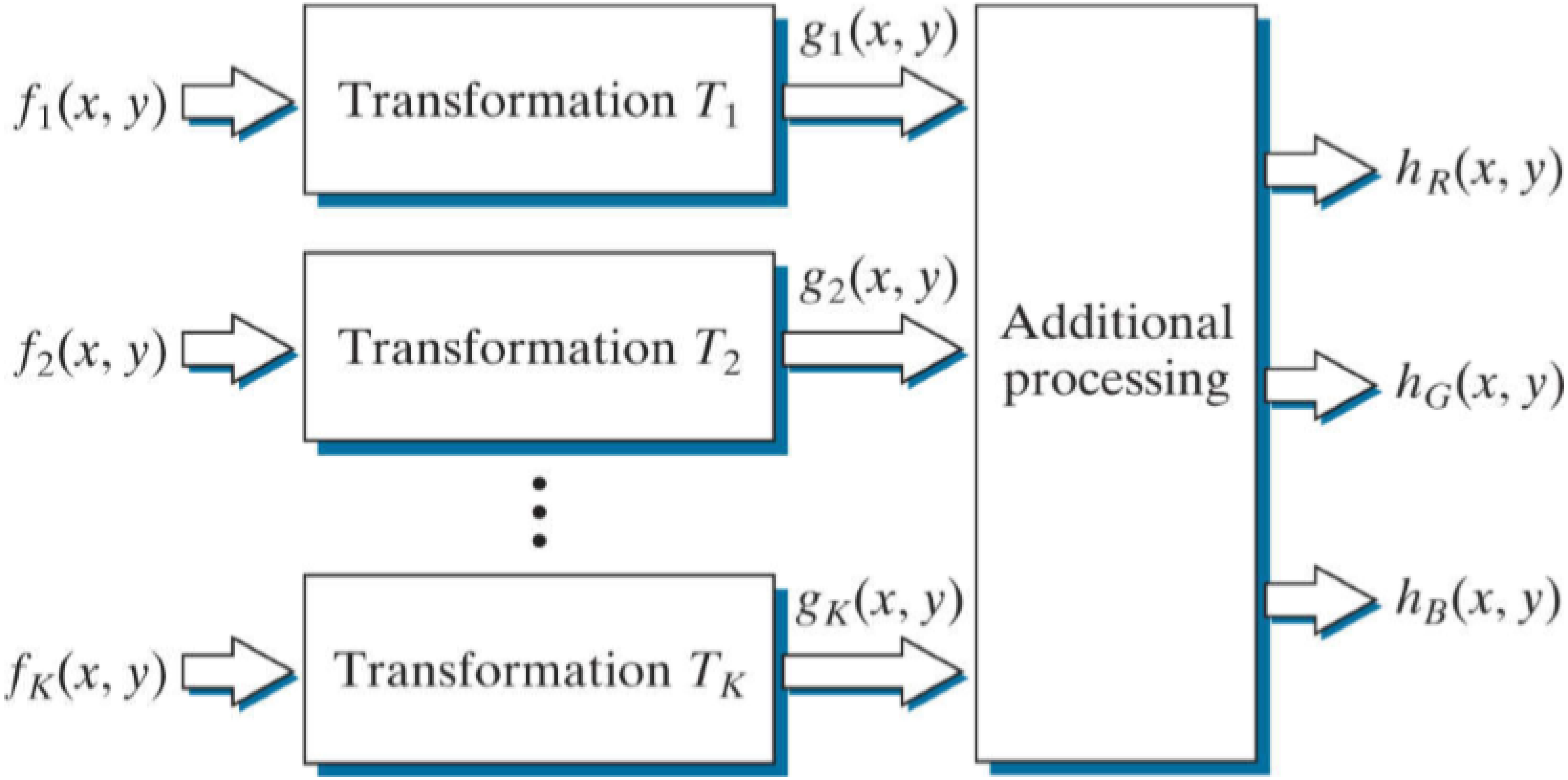


FIGURE 7.24

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

EXAMPLE 7.6: Color coding of multispectral images

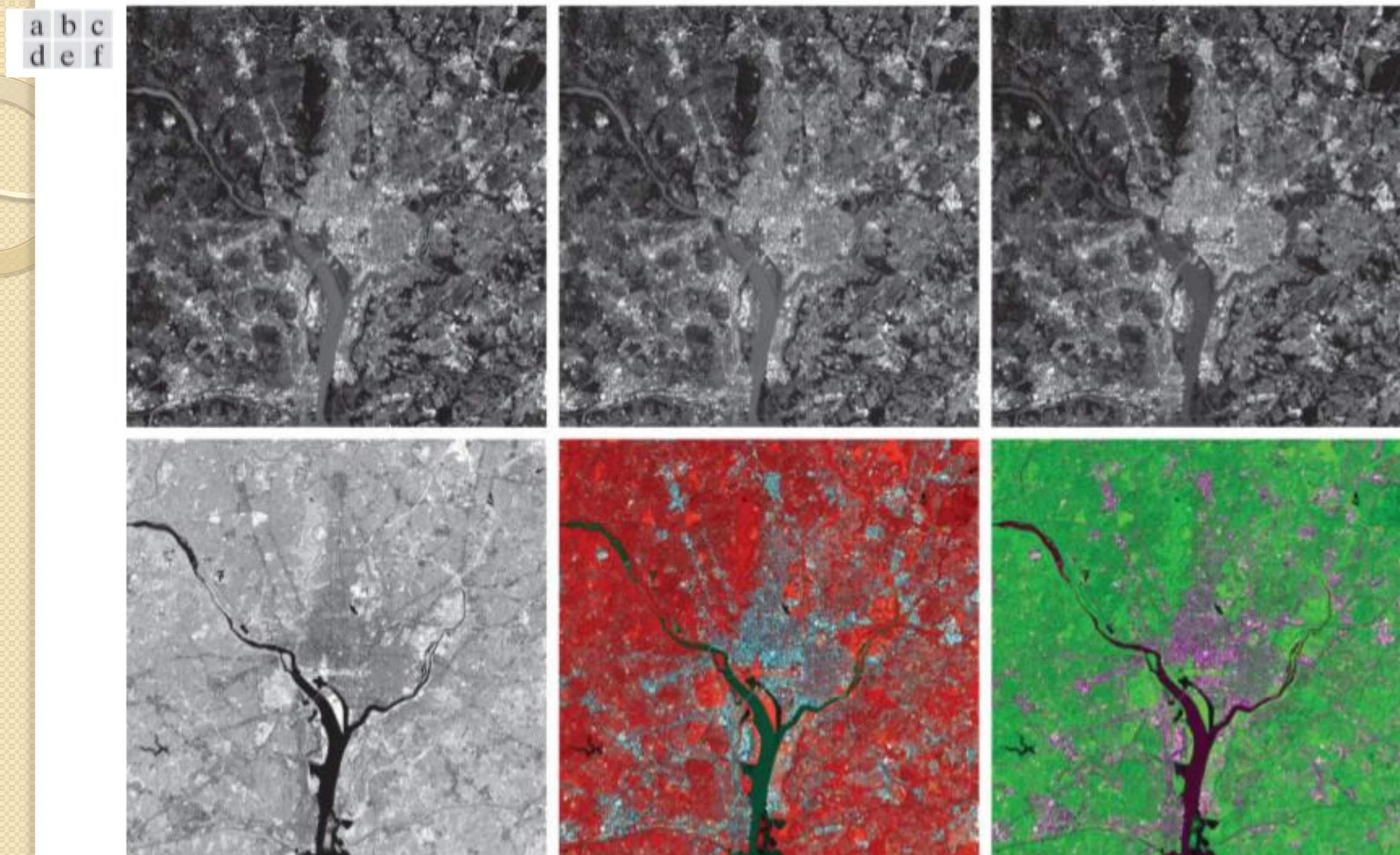


FIGURE7.25 (a)–(d) Red (R), green (G), blue (B), and near-infrared (IR) components of a LANDSAT multispectral image of the Washington, D.C. area. (e) RGB color composite image obtained using the IR, G, and B component images. (f) RGB color composite image obtained using the R, IR, and B component images.

- Figures7.25(a) through (d) show four satellite images of the Washington, D.C., area, including part of the Potomac River.
- The first three images are in the visible red (R), green (G), and blue (B) bands, and the fourth is in the near infrared (IR) band.
- The latter band is responsive to the biomass content of a scene, and we want to use this fact to create a composite RGB color image in which vegetation is emphasized and the other components of the scene are displayed in more muted tones.
- Figure 7.25(e) is an RGB composite obtained by replacing the red image by infrared.
- As you see, vegetation shows as a bright red, and the other components of the scene, which had a weaker response in the near-infrared band, show in pale shades of blue green.

- Figure 7.25(f) is a similar image, but with the green replaced by infrared.
- Here, vegetation shows in a bright green color, and the other components of the scene show in purplish color shades, indicating that their major components are in the red and blue bands.
- Although the last two images do not introduce any new physical information, these images are much easier to interpret visually once it is known that the dominant component of the images are pixels of areas heavily populated by vegetation.

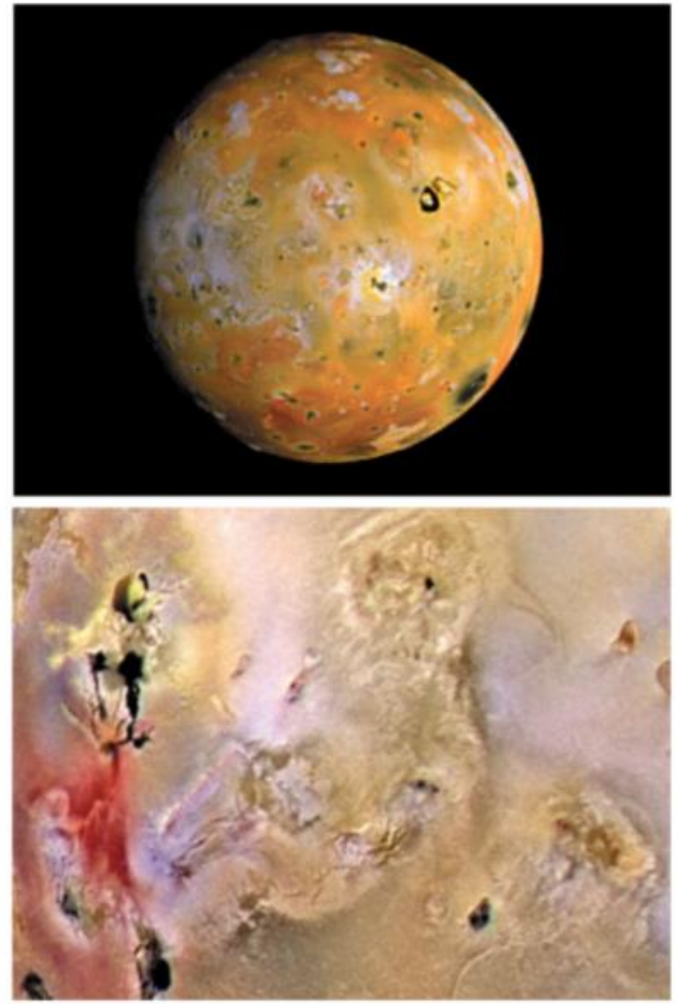


FIGURE7.26

(a) Pseudocolor rendition of Jupiter Moon Io. (b) A close-up.

- The type of processing just illustrated uses the physical characteristics of a single band in a multi-spectral image to emphasize areas of interest.
- The same approach can help visualize events of interest in complex images in which the events are beyond human visual sensing capabilities.
- Figure 7.26 is an excellent illustration of this. These are images of the Jupiter moon Io, shown in pseudocolor by combining several of the sensor images from the Galileo spacecraft, some of which are in spectral regions not visible to the eye.
- However, by understanding the physical and chemical processes likely to affect sensor response, it is possible to combine the sensed images into a meaningful pseudocolor map.
- One way to combine the sensed image data is by how they show either differences in surface chemical composition or changes in the way the surface reflects sunlight

- For example, in the pseudocolor image in Fig.7.26(b) , bright red depicts material newly ejected from an active volcano on Io, and the surrounding yellow materials are older sulfur deposits.
- This image conveys these characteristics much more readily than would be possible by analyzing the component images individually.