DIGITAL IMAGE PROCESSING

UNIT I
Introduction and Image Processing

Digital Image Processing (DIP) is a software which is used to manipulate the digital images by the use of computer system. It is also used to enhance the images, to get some important information from it. **For example:** Adobe Photoshop, MATLAB, etc.

It is also used in the conversion of signals from an image sensor into the digital images.

A certain number of algorithms are used in image processing.

- Digital Image Processing is a software which is used in image processing. For example: computer graphics, signals, photography, camera mechanism, pixels, etc.
- Digital Image Processing provides a platform to perform various operations like image enhancing, processing of analog and digital signals, image signals, voice signals etc.
- It provides images in different formats.

Digital Image Processing allows users the following tasks

- **Image sharpening and restoration:** The common applications of Image sharpening and restoration are zooming, blurring, sharpening, grayscale conversion, edges detecting, Image recognition, and Image retrieval, etc.
- **Medical field:** The common applications of medical field are Gamma-ray imaging, PET scan, X-Ray Imaging, Medical CT, UV imaging, etc.
- **Remote sensing:** It is the process of scanning the earth by the use of satellite and acknowledges all activities of space.
- **Machine/Robot vision:** It works on the vision of robots so that they can see things, identify them, etc.

Characteristics of Digital Image Processing

- It uses software, and some are free of cost.
- It provides clear images.

- Digital Image Processing do image enhancement to recollect the data through images.
- It is used widely everywhere in many fields.
- It reduces the complexity of digital image processing.
- It is used to support a better experience of life.

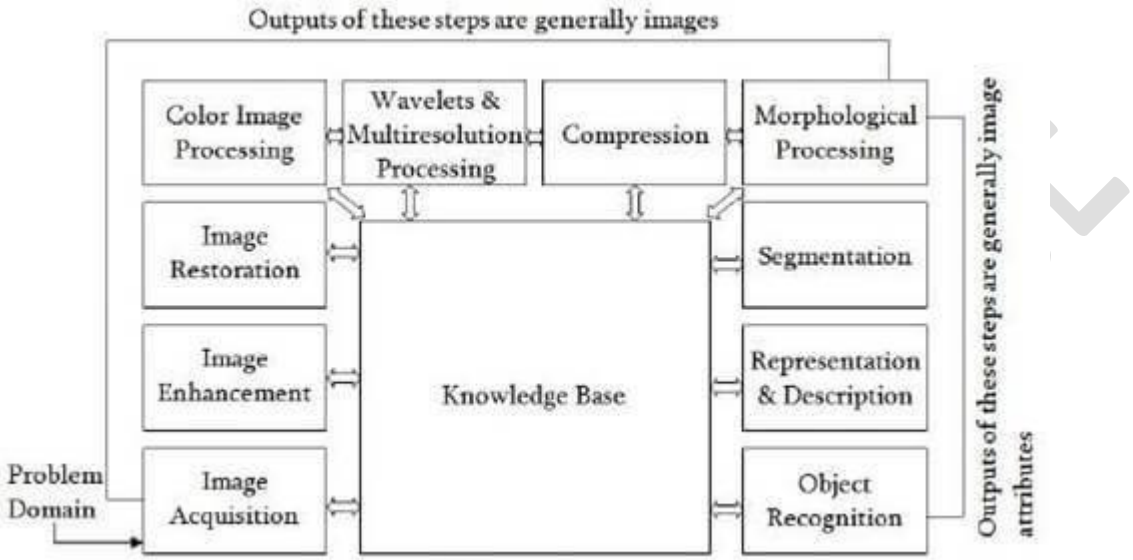
Advantages of Digital Image Processing

- Image reconstruction (CT, MRI, SPECT, PET)
- Image reformatting (Multi-plane, multi-view reconstructions)
- Fast image storage and retrieval
- Fast and high-quality image distribution.
- Controlled viewing (windowing, zooming)

Disadvantages of Digital Image Processing

- It is very much time-consuming.
- It is very much costly depending on the particular system.
- Qualified persons can be used.

Fundamental Steps of Digital Image Processing:



1. Image Acquisition

Image acquisition is the first step of the fundamental steps of DIP. In this stage, an image is given in the digital form. Generally, in this stage, pre-processing such as scaling is done.

2. Image Enhancement

Image enhancement is the simplest and most attractive area of DIP. In this stage details which are not known, or we can say that interesting features of an image is highlighted. Such as brightness, contrast, etc...

3. Image Restoration

Image restoration is the stage in which the appearance of an image is improved.

4. Color Image Processing

Color image processing is a famous area because it has increased the use of digital images on the internet. This includes color modeling, processing in a digital domain, etc....

5. Wavelets and Multi-Resolution Processing

In this stage, an image is represented in various degrees of resolution. Image is divided into smaller regions for data compression and for the pyramidal representation.

6. Compression

Compression is a technique which is used for reducing the requirement of storing an image. It is a very important stage because it is very necessary to compress data for internet use.

7. Morphological Processing

This stage deals with tools which are used for extracting the components of the image, which is useful in the representation and description of shape.

8. Segmentation

In this stage, an image is a partitioned into its objects. Segmentation is the most difficult tasks in DIP. It is a process which takes a lot of time for the successful solution of imaging problems which requires objects to identify individually.

9. Representation and Description

Representation and description follow the output of the segmentation stage. The output is a raw pixel data which has all points of the region itself. To transform the raw data, representation is the only solution. Whereas description is used for extracting information's to differentiate one class of objects from another.

10. Object recognition

In this stage, the label is assigned to the object, which is based on descriptors.

11. Knowledge Base

Knowledge is the last stage in DIP. In this stage, important information of the image is located, which limits the searching processes. The knowledge base is very complex when the image database has a high-resolution satellite.

Applications of Digital Image Processing

Almost in every field, digital image processing puts a live effect on things and is growing with time to time and with new technologies.

1) Image sharpening and restoration

It refers to the process in which we can modify the look and feel of an image. It basically manipulates the images and achieves the desired output. It includes conversion, sharpening, blurring, detecting edges, retrieval, and recognition of images.

2) Medical Field

There are several applications under medical field which depends on the functioning of digital image processing.

- Gamma-ray imaging
- PET scan
- X-Ray Imaging
- Medical CT scan
- UV imaging

3) Robot vision

There are several robotic machines which work on the digital image processing. Through image processing technique robot finds their ways, for example, hurdle detection root and line follower robot.

4) Pattern recognition

It involves the study of image processing, it is also combined with artificial intelligence such that computer-aided diagnosis, handwriting recognition and images recognition can be easily implemented. Now a days, image processing is used for pattern recognition.

5) Video processing

It is also one of the applications of digital image processing. A collection of frames or pictures are arranged in such a way that it makes the fast movement of pictures. It involves frame rate conversion, motion detection, reduction of noise and colour space conversion etc.

Human Visual System

Although the field of digital image processing is built on a foundation of mathematics, human intuition and analysis often play a role in the choice of one technique versus another, and this choice often is made based on subjective, visual judgments. Thus, developing an understanding of basic characteristics of human visual perception as a first step in our journey through this book is appropriate. In particular, our interest is in the elementary mechanics of how images are formed and perceived by humans. We are interested in learning the physical limitations of human vision in terms of factors that also are used in our work with digital images. Factors such as how human and electronic imaging devices compare in terms of resolution and ability to adapt to changes in illumination are not only interesting, they are also important from a practical point of view.

STRUCTURE OF THE HUMAN EYE

Figure 2.1 shows a simplified cross section of the human eye. The eye is nearly a sphere (with a diameter of about 20 mm) enclosed by three membranes: the cornea and sclera outer cover; the choroid; and the retina. The cornea is a tough, transparent tissue that covers the anterior surface of the eye. Continuous with the cornea, the sclera is an opaque membrane that encloses the remainder of the optic globe. The choroid lies directly below the sclera. This membrane contains a network of blood vessels that serve as the major source of nutrition to the eye. Even superficial

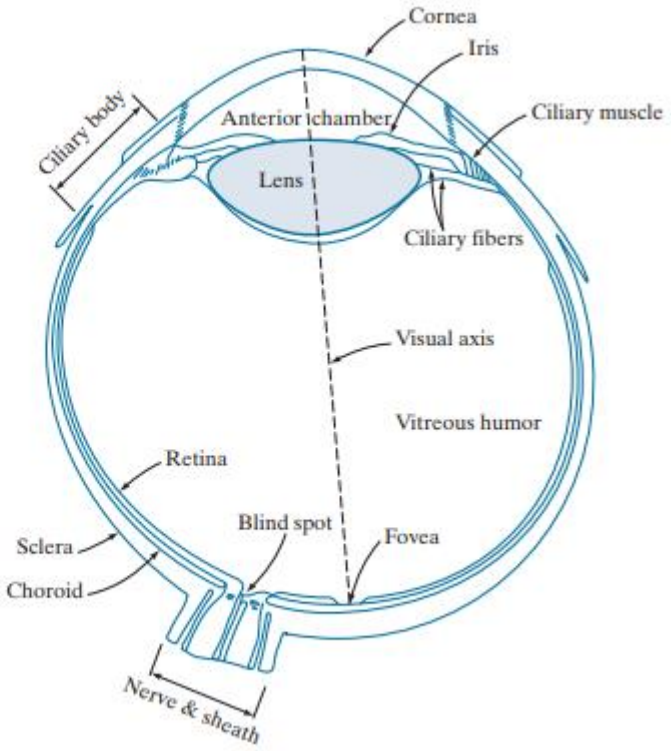


FIGURE 2.1
Simplified diagram of a cross section of the human eye.

injury to the choroid can lead to severe eye damage as a result of inflammation that restricts blood flow. The choroid coat is heavily pigmented, which helps reduce the amount of extraneous light entering the eye and the backscatter within the optic globe. At its anterior extreme, the choroid is divided into the ciliary body and the iris. The latter contracts or expands to control the amount of light that enters the eye. The central opening of the iris (the pupil) varies in diameter from approximately 2 to 8 mm. The front of the iris contains the visible pigment of the eye, whereas the back contains a black pigment. The lens consists of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body. It is composed of 60% to 70% water, about 6% fat, and more protein than any other tissue in the eye. The lens is colored by a slightly yellow pigmentation that increases with age. In extreme cases, excessive clouding of the lens, referred to as cataracts, can lead to poor color discrimination and loss of clear vision. The lens absorbs approximately 8% of the visible light spectrum, with higher absorption at shorter wavelengths. Both infrared and ultraviolet light are absorbed by proteins within the lens and, in excessive amounts, can damage the eye. The innermost membrane of the eye is the retina, which lines the inside of the wall's entire posterior portion. When the eye is focused, light from an object is imaged on the retina. Pattern vision is afforded by discrete light receptors distributed over the surface of the retina. There are two types of receptors: cones and rods. There are between 6 and 7 million cones in each eye. They are located primarily in the central portion of the retina, called the fovea, and are highly sensitive to color. Humans can resolve fine details because each cone is connected to its own nerve end. Muscles rotate the eye until the image of a region of interest falls on the fovea. Cone vision is called photopic or bright-light vision. The number of rods is much

larger: Some 75 to 150 million are distributed over the retina. The larger area of distribution, and the fact that several rods are connected to a single nerve ending, reduces the amount of detail discernible by these receptors. Rods capture an overall image of the field of view. They are not involved in color vision, and are sensitive to low levels of illumination. For example, objects that appear brightly colored in daylight appear as colorless forms in moonlight because only the rods are stimulated. This phenomenon is known as scotopic or dim-light vision. Figure 2.2 shows the density of rods and cones for a cross section of the right eye, passing through the region where the optic nerve emerges from the eye. The absence of receptors in this area causes the so-called blind spot (see Fig. 2.1). Except for this region, the distribution of receptors is radially symmetric about the fovea. Receptor density is measured in degrees from the visual axis. Note in Fig. 2.2 that cones are most dense in the center area of the fovea, and that rods increase in density from the center out to approximately 20° off axis. Then, their density decreases out to the periphery of the retina. The fovea itself is a circular indentation in the retina of about 1.5 mm in diameter, so it has an area of approximately 1.77 mm2 . As Fig. 2.2 shows, the density of cones in that area of the retina is on the order of 150,000 elements per mm2 . Based on these figures, the number of cones in the fovea, which is the region of highest acuity

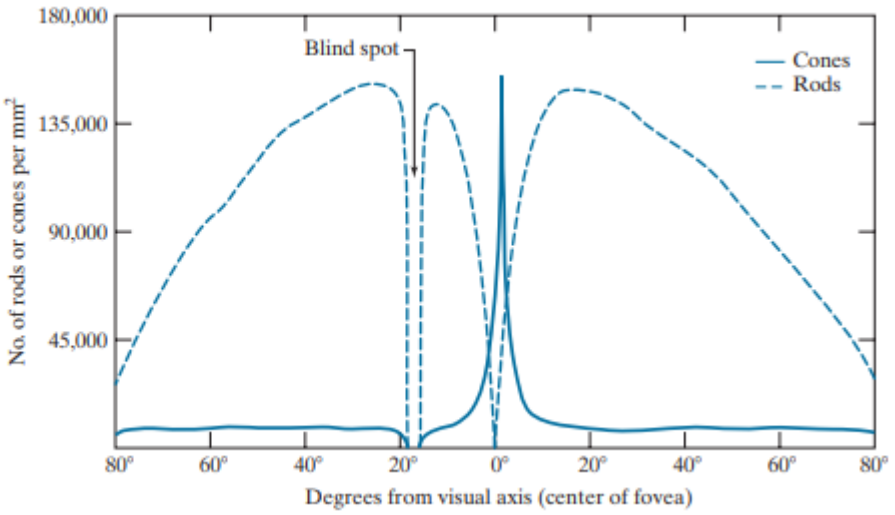


FIGURE 2.2 Distribution of rods and cones in the retina.

in the eye, is about 265,000 elements. Modern electronic imaging chips exceed this number by a large factor. While the ability of humans to integrate intelligence and experience with vision makes purely quantitative comparisons somewhat superficial, keep in mind for future discussions that electronic imaging sensors can easily exceed the capability of the eye in resolving image detail.

IMAGE FORMATION IN THE EYE

In an ordinary photographic camera, the lens has a fixed focal length. Focusing at various distances is achieved by varying the distance between the lens and the imaging plane, where the film (or imaging chip in the case of a digital camera) is located. In the human eye, the converse is true; the distance between the center of the lens and the imaging sensor (the retina) is fixed, and the focal length needed to achieve proper focus is obtained by varying the

shape of the lens. The fibers in the ciliary body accomplish this by flattening or thickening the lens for distant or near objects, respectively. The distance between the center of the lens and the retina along the visual axis is approximately 17 mm. The range of focal lengths is approximately 14 mm to 17 mm, the latter taking place when the eye is relaxed and focused at distances greater than about 3 m. The geometry in Fig. 2.3 illustrates how to obtain the dimensions of an image formed on the retina. For example, suppose that a person is looking at a tree 15 m high at a distance of 100 m. Letting h denote the height of that object in the retinal image, the geometry of Fig. 2.3 yields 15 100 17 = h or h = 2 5. mm. As indicated earlier in this section, the retinal image is focused primarily on the region of the fovea. Perception then takes place by the relative excitation of light receptors, which transform radiant energy into electrical impulses that ultimately are decoded by the brain.

BRIGHTNESS ADAPTATION AND DISCRIMINATION

Because digital images are displayed as sets of discrete intensities, the eye's ability to discriminate between different intensity levels is an important consideration

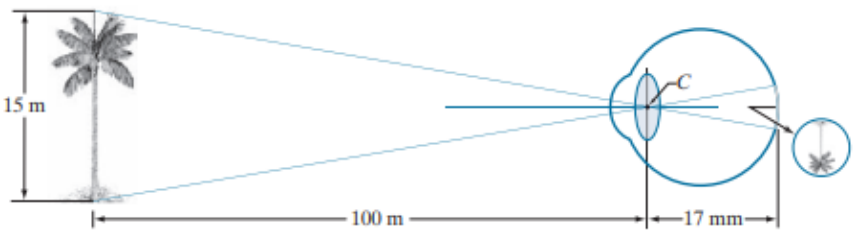


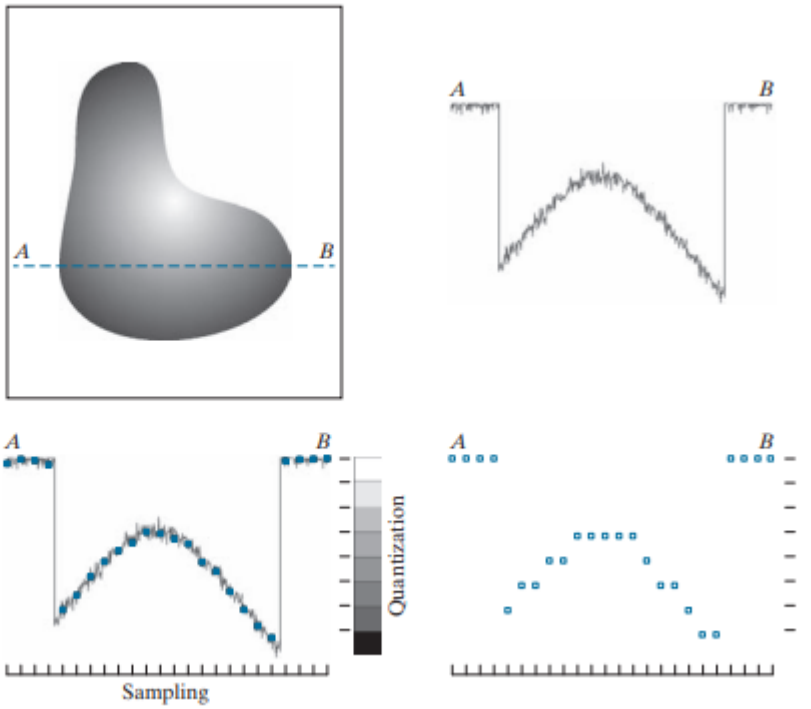
FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point *C* is the focal center of the lens.

in presenting image processing results. The range of light intensity levels to which the human visual system can adapt is enormous—on the order of 1010— from the scotopic threshold to the glare limit. Experimental evidence indicates that subjective brightness (intensity as perceived by the human visual system) is a logarithmic function of the light intensity incident on the eye. Figure 2.4, a plot of light intensity versus subjective brightness, illustrates this characteristic. The long solid curve represents the range of intensities to which the visual system can adapt. In photopic vision alone, the range is about 106 . The transition from scotopic to photopic vision is gradual over the approximate range from 0.001 to 0.1 millilambert (−3 to −1 mL in the log scale), as the double branches of the adaptation curve in this range show.

IMAGE SAMPLING AND QUANTIZATION

As discussed in the previous section, there are numerous ways to acquire images, but our objective in all is the same: to generate digital images from sensed data. The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed. To create a digital image, we need to convert the continuous sensed data into a digital format. This requires two processes: sampling and

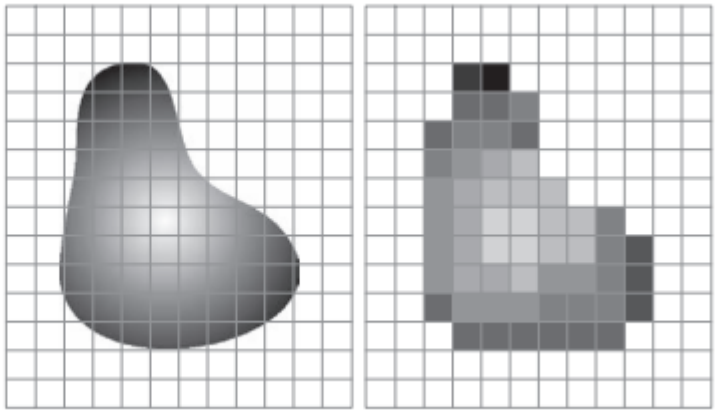
quantization. Figure 2.16(a) shows a continuous image f that we want to convert to digital form. An image may be continuous with respect to the x- and y-coordinates, and also in



a b
c d

FIGURE 2.16
(a) Continuous image. (b) A scan line showing intensity variations along line *AB* in the continuous image. (c) Sampling and quantization. (d) Digital scan line. (The black border in (a) is included for clarity. It is not part of the image).

amplitude. To digitize it, we have to sample the function in both coordinates and also in amplitude. Digitizing the coordinate values is called sampling. Digitizing the amplitude values is called quantization. The one-dimensional function in Fig. 2.16(b) is a plot of amplitude (intensity level) values of the continuous image along the line segment AB in Fig. 2.16(a). The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Fig. 2.16(c). The samples are shown as small dark squares superimposed on the function, and their (discrete) spatial locations are indicated by corresponding tick marks in the bottom of the figure. The set of dark squares constitute the sampled function. However, the values of the samples still span (vertically) a continuous range of intensity values. In order to form a digital function, the intensity values also must be converted (quantized) into discrete quantities. The vertical gray bar in Fig. 2.16(c) depicts the intensity scale divided into eight discrete intervals, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight intensity intervals. The continuous intensity levels are quantized by assigning one of the eight values to each sample, depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown as white squares in Fig. 2.16(d). Starting at the top of the continuous image and carrying out this procedure downward, line by line, produces a two-dimensional digital image. It is implied in Fig. 2.16 that, in addition to the number of discrete levels used, the accuracy achieved in quantization is highly dependent on the noise content of the sampled signal.



a b

FIGURE 2.17
(a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single sensing element combined with mechanical motion, as in Fig. 2.13, the output of the sensor is quantized in the manner described above. However, spatial sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be very exact so, in principle, there is almost no limit on how fine we can sample an image using this approach. In practice, limits on sampling accuracy are determined by other factors, such as the quality of the optical components used in the system. When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the samples in the resulting image in one direction, and mechanical motion establishes the number of samples in the other. Quantization of the sensor outputs completes the process of generating a digital image. When a sensing array is used for image acquisition, no motion is required. The number of sensors in the array establishes the limits of sampling in both directions. Quantization of the sensor outputs is as explained above. Figure 2.17 illustrates this concept. Figure 2.17(a) shows a continuous image projected onto the plane of a 2-D sensor. Figure 2.17(b) shows the image after sampling and quantization. The quality of a digital image is determined to a large degree by the number of samples and discrete intensity levels used in sampling and quantization. However, as we will show later in this section, image content also plays a role in the choice of these parameters.

REPRESENTING DIGITAL IMAGES

Let f (s ,t) represent a continuous image function of two continuous variables, s and t. We convert this function into a digital image by sampling and quantization, as explained in the previous section. Suppose that we sample the continuous image into a digital image, f xy (,), containing M rows and N columns, where (,) x y are discrete coordinates. For notational clarity and convenience, we use integer values for these discrete coordinates: x M = − 012 ,,, , … 1 and y N = − 012 ,,, , … 1. Thus, for example, the value of the digital image at the origin is f(, 0 0), and its value at the next coordinates along the first row is f(, 0 1). Here, the notation (0, 1) is used to denote the second sample along the first row. It does not mean that these are the values of the physical coordinates when the image was sampled. In general, the value of a digital image at any coordinates (,) x y is denoted f xy (,), where x and y are

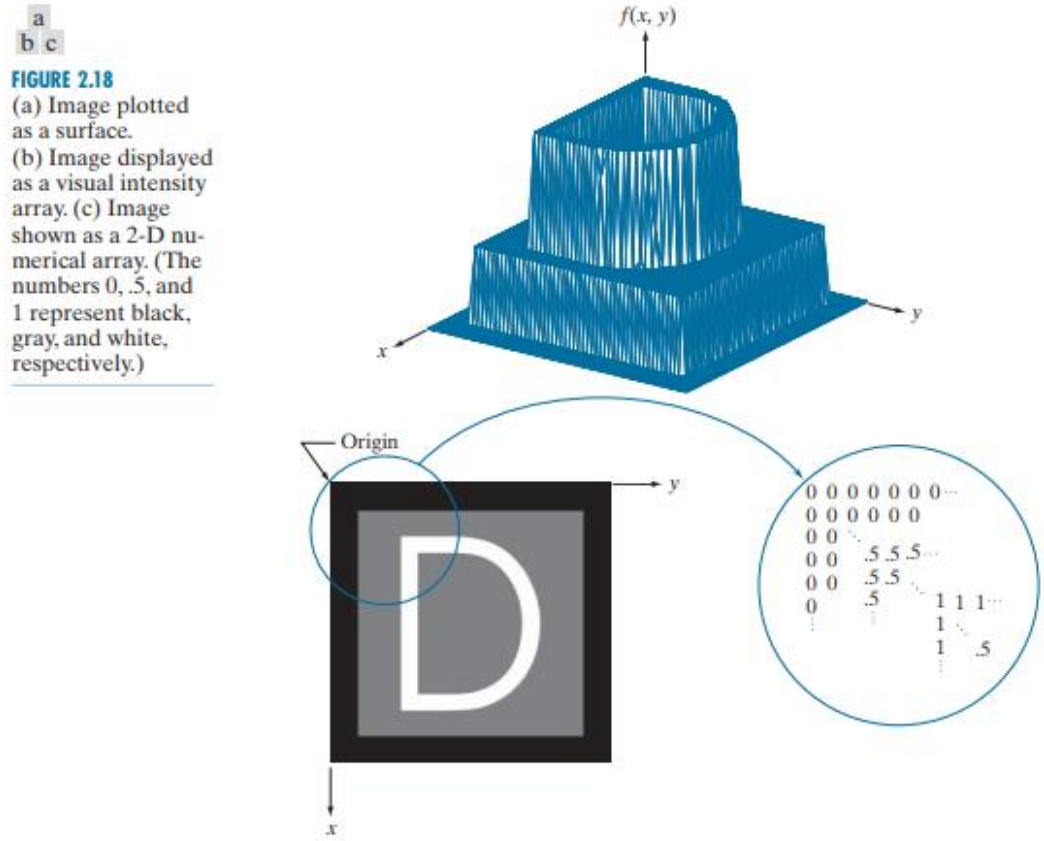
integers. When we need to refer to specific coordinates (,) i j , we use the notation fij (,), where the arguments are integers. The section of the real plane spanned by the coordinates of an image is called the spatial domain, with x and y being referred to as spatial variables or spatial coordinates. Figure 2.18 shows three ways of representing f xy (,). Figure 2.18(a) is a plot of the function, with two axes determining spatial location and the third axis being the values of f as a function of x and y. This representation is useful when working with grayscale sets whose elements are expressed as triplets of the form (,,) x y z , where x and y are spatial coordinates and z is the value of f at coordinates (,). x y We will work with this representation briefly in Section 2.6. The representation in Fig. 2.18(b) is more common, and it shows f xy (,) as it would appear on a computer display or photograph. Here, the intensity of each point in the display is proportional to the value of f at that point. In this figure, there are only three equally spaced intensity values. If the intensity is normalized to the interval [,], 0 1 then each point in the image has the value 0, 0.5, or 1. A monitor or printer converts these three values to black, gray, or white, respectively, as in Fig. 2.18(b). This type of representation includes color images, and allows us to view results at a glance. As Fig. 2.18(c) shows, the third representation is an array (matrix) composed of the numerical values of f xy (,). This is the representation used for computer processing. In equation form, we write the representation of an M N* numerical array as

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \qquad (2\text{-}9)$$

The right side of this equation is a digital image represented as an array of real numbers. Each element of this array is called an image element, picture element, pixel, or pel. We use the terms image and pixel throughout the book to denote a digital image and its elements. Figure 2.19 shows a graphical representation of an image array, where the x- and y-axis are used to denote the rows and columns of the array. Specific pixels are values of the array at a fixed pair of coordinates. As mentioned earlier, we generally use fij (,) when referring to a pixel with coordinates (,). i j We can also represent a digital image in a traditional matrix form:

$$\mathbf{A}=\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \qquad (2\text{-}10)$$

Clearly, a= fij(i,j) so Eqs. (2-9) and (2-10) denote identical arrays.



a
b c

FIGURE 2.18
(a) Image plotted as a surface. (b) Image displayed as a visual intensity array. (c) Image shown as a 2-D numerical array. (The numbers 0, .5, and 1 represent black, gray, and white, respectively.)

As Fig. 2.19 shows, we define the origin of an image at the top left corner. This is a convention based on the fact that many image displays (e.g., TV monitors) sweep an image starting at the top left and moving to the right, one row at a time. More important is the fact that the first element of a matrix is by convention at the top left of the array. Choosing the origin of f xy (,) at that point makes sense mathematically because digital images in reality are matrices. In fact, as you will see, sometimes we use x and y interchangeably in equations with the rows (r) and columns (c) of a matrix. It is important to note that the representation in Fig. 2.19, in which the positive x-axis extends downward and the positive y-axis extends to the right, is precisely the right-handed Cartesian coordinate system with which you are familiar,† but shown rotated by 90° so that the origin appears on the top, left.

Spatial and grey level resolution

SPATIAL AND INTENSITY RESOLUTION

Intuitively, spatial resolution is a measure of the smallest discernible detail in an image. Quantitatively, spatial resolution can be stated in several ways, with line pairs per unit distance, and dots (pixels) per unit distance being common measures. Suppose that we construct a chart with alternating black and white vertical lines, each of width W units (W can be less than 1). The width of a line pair is thus 2W, and there are W 2 line pairs per unit distance. For example, if the width of a line is 0.1 mm, there are 5 line pairs per unit distance

(i.e., per mm). A widely used definition of image resolution is the largest number of discernible line pairs per unit distance (e.g., 100 line pairs per mm). Dots per unit distance is a measure of image resolution used in the printing and publishing industry. In the U.S., this measure usually is expressed as dots per inch (dpi). To give you an idea of quality, newspapers are printed with a resolution of 75 dpi, magazines at 133 dpi, glossy brochures at 175 dpi, and the book page at which you are presently looking was printed at 2400 dpi. To be meaningful, measures of spatial resolution must be stated with respect to spatial units. Image size by itself does not tell the complete story. For example, to say that an image has a resolution of 1024 1024 * pixels is not a meaningful statement without stating the spatial dimensions encompassed by the image. Size by itself is helpful only in making comparisons between imaging capabilities. For instance, a digital camera with a 20-megapixel CCD imaging chip can be expected to have a higher capability to resolve detail than an 8-megapixel camera, assuming that both cameras are equipped with comparable lenses and the comparison images are taken at the same distance. Intensity resolution similarly refers to the smallest discernible change in intensity level. We have considerable discretion regarding the number of spatial samples (pixels) used to generate a digital image, but this is not true regarding the number of intensity levels. Based on hardware considerations, the number of intensity levels usually is an integer power of two, as we mentioned when discussing Eq. (2-11). The most common number is 8 bits, with 16 bits being used in some applications in which enhancement of specific intensity ranges is necessary. Intensity quantization using 32 bits is rare. Sometimes one finds systems that can digitize the intensity levels of an image using 10 or 12 bits, but these are not as common. Unlike spatial resolution, which must be based on a per-unit-of-distance basis to be meaningful, it is common practice to refer to the number of bits used to quantize intensity as the "intensity resolution." For example, it is common to say that an image whose intensity is quantized into 256 levels has 8 bits of intensity resolution. However, keep in mind that discernible changes in intensity are influenced also by noise and saturation values, and by the capabilities of human perception to analyze and interpret details in the context of an entire scene (see Section 2.1). The following two examples illustrate the effects of spatial and intensity resolution on discernible detail. Later in this section, we will discuss how these two parameters interact in determining perceived image quality.

a b
c d

FIGURE 2.23
Effects of reducing spatial resolution. The images shown are at:
(a) 930 dpi,
(b) 300 dpi,
(c) 150 dpi, and
(d) 72 dpi.

SOME BASIC RELATIONSHIPS BETWEEN PIXELS

In this section, we discuss several important relationships between pixels in a digital image. When referring in the following discussion to particular pixels, we use lowercase letters, such as p and q.

NEIGHBORS OF A PIXEL

A pixel p at coordinates (x, y) has two horizontal and two vertical neighbors with coordinates

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

This set of pixels, called the 4-*neighbors* of p, is denoted $N_4(p)$.

The four *diagonal* neighbors of p have coordinates

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

and are denoted $N_D(p)$. These neighbors, together with the 4-neighbors, are called the 8-*neighbors* of p, denoted by $N_8(p)$. The set of image locations of the neighbors of a point p is called the *neighborhood* of p. The neighborhood is said to be *closed* if it contains p. Otherwise, the neighborhood is said to be *open*.

ADJACENCY, CONNECTIVITY, REGIONS, AND BOUNDARIES

Let V be the set of intensity values used to define adjacency. In a binary image, $V = \{1\}$ if we are referring to adjacency of pixels with value 1. In a grayscale image, the idea is the same, but set V typically contains more elements. For example, if we are dealing with the adjacency of pixels whose values are in the range 0 to 255, set V could be any subset of these 256 values. We consider three types of adjacency:

1. 4-*adjacency*. Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.
2. 8-*adjacency*. Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.
3. *m-adjacency* (also called *mixed adjacency*). Two pixels p and q with values from V are m-adjacent if

(a) q is in $N_4(p)$, *or*

(b) q is in $N_D(p)$ *and* the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

We use the symbols $\cap$ and $\cup$ to denote set intersection and union, respectively. Given sets A and B, recall that their intersection is the set of elements that are members of both A and B. The union of these two sets is the set of elements that are members of A, of B, or of both. We will discuss sets in more detail in Section 2.6.

Mixed adjacency is a modification of 8-adjacency, and is introduced to eliminate the ambiguities that may result from using 8-adjacency. For example, consider the pixel arrangement in Fig. 2.28(a) and let $V = \{1\}$. The three pixels at the top of Fig. 2.28(b) show multiple (ambiguous) 8-adjacency, as indicated by the dashed lines. This ambiguity is removed by using m-adjacency, as in Fig. 2.28(c). In other words, the center and upper-right diagonal pixels are not m-adjacent because they do not satisfy condition (b).

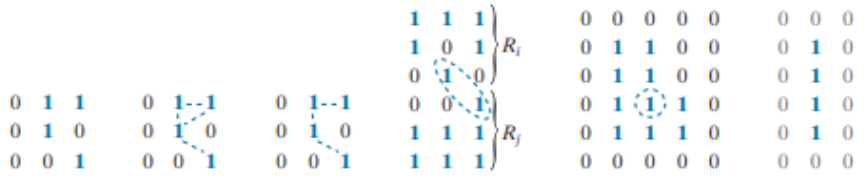
A *digital path* (or *curve*) from pixel p with coordinates (x_0, y_0) to pixel q with coordinates (x_n, y_n) is a sequence of distinct pixels with coordinates

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

where points (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \le i \le n$. In this case, n is the *length* of the path. If $(x_0, y_0) = (x_n, y_n)$ the path is a *closed* path. We can define 4-, 8-, or m-paths, depending on the type of adjacency specified. For example, the paths in Fig. 2.28(b) between the top right and bottom right points are 8-paths, and the path in Fig. 2.28(c) is an m-path.

Let S represent a subset of pixels in an image. Two pixels p and q are said to be *connected in S* if there exists a path between them consisting entirely of pixels in S. For any pixel p in S, the set of pixels that are connected to it in S is called a *connected component* of S. If it only has one component, and that component is connected, then S is called a *connected set*.

Let R represent a subset of pixels in an image. We call R a *region* of the image if R is a connected set. Two regions, R_i and R_j are said to be *adjacent* if their union forms a connected set. Regions that are not adjacent are said to be *disjoint*. We consider 4- and 8-adjacency when referring to regions. For our definition to make sense, the type of adjacency used must be specified. For example, the two regions of 1's in Fig. 2.28(d) are adjacent only if 8-adjacency is used (according to the definition in the previous



a b c d e f

FIGURE 2.28 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines). (c) m-adjacency. (d) Two regions (of 1's) that are 8-adjacent. (e) The circled point is on the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

paragraph, a 4-path between the two regions does not exist, so their union is not a connected set).

DISTANCE MEASURES

For pixels p, q, and s, with coordinates (x, y), (u, v), and (w, z), respectively, D is a *distance function* or *metric* if

(a) $D(p,q) \geq 0 \quad (D(p,q) = 0 \text{ iff } p = q)$,
(b) $D(p,q) = D(q, p)$, and
(c) $D(p,s) \leq D(p, q) + D(q,s)$.

The *Euclidean distance* between p and q is defined as

$$D_e(p, q) = \left[(x - u)^2 + (y - v)^2\right]^{\frac{1}{2}} \tag{2-19}$$

For this distance measure, the pixels having a distance less than or equal to some value r from (x, y) are the points contained in a disk of radius r centered at (x, y).

The D_4 *distance*, (called the *city-block distance*) between p and q is defined as

$$D_4(p, q) = |x - u| + |y - v| \tag{2-20}$$

In this case, pixels having a D_4 distance from (x, y) that is less than or equal to some value d form a diamond centered at (x, y). For example, the pixels with D_4 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

The pixels with $D_4 = 1$ are the 4-neighbors of (x, y).

The D_8 *distance* (called the *chessboard distance*) between p and q is defined as

$$D_8(p,q) = \max(|x - u|, |y - v|) \tag{2-21}$$

In this case, the pixels with D_8 distance from (x, y) less than or equal to some value d form a square centered at (x, y). For example, the pixels with D_8 distance ≤ 2 form the following contours of constant distance:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

The pixels with $D_8 = 1$ are the 8-neighbors of the pixel at (x, y).

Note that the D_4 and D_8 distances between p and q are independent of any paths that might exist between these points because these distances involve only the coordinates of the points. In the case of m-adjacency, however, the D_m distance between two points is defined as the shortest m-path between the points. In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors. For instance, consider the following arrangement of pixels and assume that p, p_2, and p_4 have a value of 1, and that p_1 and p_3 can be 0 or 1:

$$\begin{matrix} & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \end{matrix}$$

Suppose that we consider adjacency of pixels valued 1 (i.e., $V = \{1\}$). If p_1 and p_3 are 0, the length of the shortest m-path (the D_m distance) between p and p_4 is 2. If p_1 is 1, then p_2 and p will no longer be m-adjacent (see the definition of m-adjacency given earlier) and the length of the shortest m-path becomes 3 (the path goes through the points $pp_1p_2p_4$). Similar comments apply if p_3 is 1 (and p_1 is 0); in this case, the length of the shortest m-path also is 3. Finally, if both p_1 and p_3 are 1, the length of the shortest m-path between p and p_4 is 4. In this case, the path goes through the sequence of points $pp_1p_2p_3p_4$.

Color Fundamentals

In this chapter we discuss fundamentals of color image processing using the Image Processing Toolbox and extend some of its functionality by developing additional color generation and transformation functions.

Color Image Representation in MATLAB As noted in Section 2.6, the Image Processing Toolbox handles color images either as indexed images or RGB (red, green, blue) images. In this section we discuss these two image types in some detail. 7.1.1 RGB Images An RGB color image is an M X N X 3 array of color pixels, where each color pixel is a triplet corresponding to the red, green, and blue components of an RGB image at a specific spatial location (see Fig. 7.1). An RGB image may be viewed as a "stack" of three gray-scale images that, when fed into the red, green, and blue inputs of a color monitor, produce a color image on the screen. By convention, the three images forming an RGB color image are referred to as the red, green, and blue component images. The data class of the component images determines their range of values. If an RGB image is of class double, the range of values is [O, 1]. Similarly, the range of values is [O, 255] or [O, 65535] for RGB images of class uint8 or uint 1 6, respectively. The number of bits used to represent the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8-bit image, the corresponding RGB image is said to be 24 bits deep. Generally, the number of bits in all component images is the same. In this case, the number of possible colors in an RGB image is (2") 3 where b is the number of bits in each component image. For an 8-bit image, the number is 16,777,216 colors. Let fR, fG, and fB represent three RGB component images. An RGB image is formed from these images by using the cat (concatenate) operator to stack the images:

rgb_image = cat (3, fR , fG , fB)

The order in which the images are placed in the operand matters. In general, cat (dim , A 1 , A2 , . . .) concatenates the arrays (which must be of the same size) along the dimension specified by dim. For example, if dim = 1, the arrays are arranged vertically, if dim = 2, they are arranged horizontally, and, if dim = 3, they are stacked in the third dimension, as in Fig. 7.1. If all component images are identical, the result is a gray-scale image. Let rgb_image denote an RGB image. The following commands extract the three component images:

```
>> fR rgb_image (: ,  :,  1 ) ;
>> fG rgb_image (: ,  :,  2 ) ;
>> fB rgb_image (: , :,  3 ) ;
```

The RGB color space usually is shown graphically as an RGB color cube, as depicted in Fig. 7.2. The vertices of the cube are the primary (red, green, and blue) and secondary (cyan, magenta, and yellow) colors of light. To view the color cube from any perspective, use custom function rgbcube:

rgbcube (vx, vy , vz)

Typing rgbcube (vx , vy , vz) at the prompt produces an RGB cube on the MATLAB desktop, viewed from point (vx , vy , vz). The resulting image can be saved to disk using function print, discussed in Section 2.4. The code for function rgbcube follows.

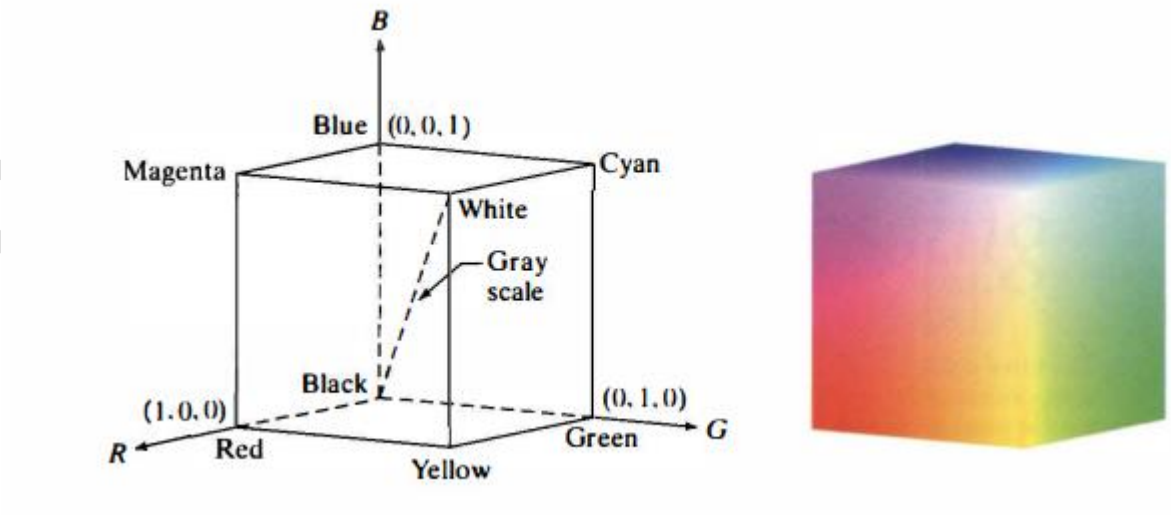


FIGURE 7.2 (a) Schematic of the RGB color cube showing the primary and secondary colors of light at the vertices. Points along the main diagonal have gray values from black at the origin to white at point (1 . 1 , 1). (b)The RGB color cube.

Converting Between Color Spaces (Color Model)

As explained in the previous section, the toolbox represents colors as RGB values, directly in an RGB image, or indirectly in an indexed image, where the color map is stored in RGB format. However, there are other color spaces (also called color models) whose use in some applications may be more convenient and/or meaningful than RGB. These models are transformations of the RGB model and include the NTSC, YCbCr, HSY, CMY, CMYK, and HSI color spaces. The toolbox provides conversion functions from RGB to the NTSC, YCbCr, HSY and CMY color spaces, and back.

NTSC Color Space The NTSC color system is used in analog television. One of the main advantages of this format is that gray-scale information is separate from color data, so the same signal can be used for both color and monochrome television sets. In the NTSC format, image data consists of three components: luminance (Y), hue (!), and saturation (Q), where the choice of the letters YIQ is conventional. The luminance component represents gray-scale information, and the other two components carry the color information of a TV signal. The YIQ components are obtained from the RGB components of an image using the linear transformation

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

Note that the elements of the first row sum to 1 and the elements of the next two rows sum to 0. This is as expected because for a grayscale image all the RGB components are equal, so the I and Q components should be 0 for such an image. Function rgb2ntsc performs the preceding transformation:

yiq_image = rgb2ntsc (rgb_image)

where the input RGB image can be of class uint8, uint1 6, or double. The output image is an M X N X 3 array of class double. Component image

yiq_image (:, : , 1) is the luminance,

yiq_image (: , : , 2) is the hue, and

yiq_image (: , : , 3) is the saturation image.

The YCbCr Color Space

The YCbCr color space is used extensively in digital video. In this format, luminance information is represented by a single component, Y, and color information is stored as two color-difference components, Cb and Cr. Component Cb is the difference between the blue component and a reference value, and component Cr is the difference between the red component and a reference value (Poynton (1 996]). The transformation used by the toolbox to convert from RGB to YCbCr is

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The conversion function is
ycbc r_image = rgb2ycbcr (rgb_image)

The input RGB image can be of class uintB, uint 16, or double. The output image is of the same class as the input. A similar transformation converts from YCbCr back to RGB:

rgb_image = ycbr2rgb(ycbcr_image)

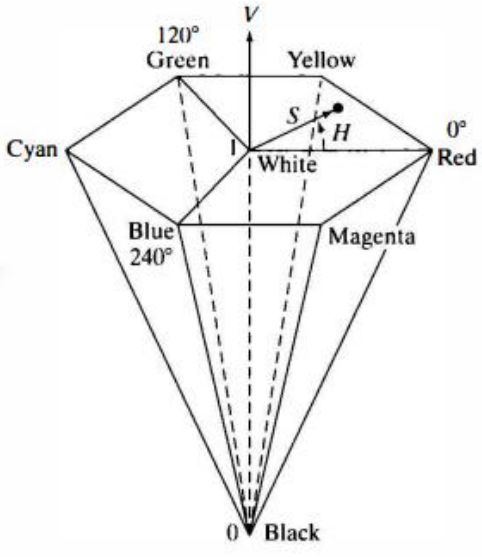
The input YCbCr image can be of class uint8, uint 1 6, or double. The output image is of the same class as the input.

The HSV Color Space

HSY (hue, saturation, value) is one of several color systems used by people to select colors (e.g., of paints or inks) from a color wheel or palette. This color system is considerably closer than the RGB system to the way in which humans experience and describe color sensations. In artists' terminology, hue, saturation, and value refer approximately to tint, shade, and tone. The HSY color space is formulated by looking at the RGB color cube along its gray axis (the axis joining the black and white vertices), which results in the hexagonally shaped color palette shown in Fig. 7.5(a). As we move along the vertical (gray) axis in Fig. 7.5(b), the size of the hexagonal plane that is perpendicular to the axis changes, yielding the volume depicted in the figure. Hue is expressed as an angle around a color hexagon, typically using the red axis as the reference (0°) axis. The value component is measured along the axis of the cone.

a b

FIGURE 7.5
(a) The HSV color hexagon.
(b) The HSV hexagonal cone.



in the center of the full color hexagon in Fig. 7.5(a). Thus, this axis represents all shades of gray. Saturation (purity of the color) is measured as the distance from the V axis. The HSY color system is based on cylindrical coordinates. Converting from RGB to HSY entails developing the equations to map RGB values (which are in Cartesian coordinates) to cylindrical coordinates. This topic is treated in detail in most texts on computer graphics so

we do not develop the equations here. The MATLAB function for converting from RGB to HSY is rgb2hsv, whose syntax is

hsv_image = rgb2hsv (rgb_image)