



# *Digital Image Processing*

Third Edition

*Rafael C. Gonzalez*

University of Tennessee

*Richard E. Woods*

MedData Interactive



Upper Saddle River, NJ 07458

## Library of Congress Cataloging-in-Publication Data on File

Vice President and Editorial Director, ECS: *Marcia J. Horton*  
 Executive Editor: *Michael McDonald*  
 Associate Editor: *Alice Dworkin*  
 Editorial Assistant: *William Opaluch*  
 Managing Editor: *Scott Disanno*  
 Production Editor: *Rose Kernan*  
 Director of Creative Services: *Paul Belfanti*  
 Creative Director: *Juan Lopez*  
 Art Director: *Heather Scott*  
 Art Editors: *Gregory Dulles* and *Thomas Benfatti*  
 Manufacturing Manager: *Alexis Heydt-Long*  
 Manufacturing Buyer: *Lisa McDowell*  
 Senior Marketing Manager: *Tim Galligan*



© 2008 by Pearson Education, Inc.  
 Pearson Prentice Hall  
 Pearson Education, Inc.  
 Upper Saddle River, New Jersey 07458

All rights reserved. No part of this book may be reproduced, in any form, or by any means, without permission in writing from the publisher.

Pearson Prentice Hall® is a trademark of Pearson Education, Inc.

The authors and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The authors and publisher shall not be liable in any event for incidental or consequential damages with, or arising out of, the furnishing, performance, or use of these programs.

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-168728-x  
 978-0-13-168728-8

Pearson Education Ltd., *London*  
 Pearson Education Australia Pty. Ltd., *Sydney*  
 Pearson Education Singapore, Pte., Ltd.  
 Pearson Education North Asia Ltd., *Hong Kong*  
 Pearson Education Canada, Inc., *Toronto*  
 Pearson Educación de Mexico, S.A. de C.V.  
 Pearson Education—Japan, *Tokyo*  
 Pearson Education Malaysia, Pte. Ltd.  
 Pearson Education, Inc., *Upper Saddle River, New Jersey*



# 2 *Digital Image Fundamentals*

Those who wish to succeed must ask the right preliminary questions.

*Aristotle*

## *Preview*

The purpose of this chapter is to introduce you to a number of basic concepts in digital image processing that are used throughout the book. Section 2.1 summarizes the mechanics of the human visual system, including image formation in the eye and its capabilities for brightness adaptation and discrimination. Section 2.2 discusses light, other components of the electromagnetic spectrum, and their imaging characteristics. Section 2.3 discusses imaging sensors and how they are used to generate digital images. Section 2.4 introduces the concepts of uniform image sampling and intensity quantization. Additional topics discussed in that section include digital image representation, the effects of varying the number of samples and intensity levels in an image, the concepts of spatial and intensity resolution, and the principles of image interpolation. Section 2.5 deals with a variety of basic relationships between pixels. Finally, Section 2.6 is an introduction to the principal mathematical tools we use throughout the book. A second objective of that section is to help you begin developing a “feel” for how these tools are used in a variety of basic image processing tasks. The scope of these tools and their application are expanded as needed in the remainder of the book.

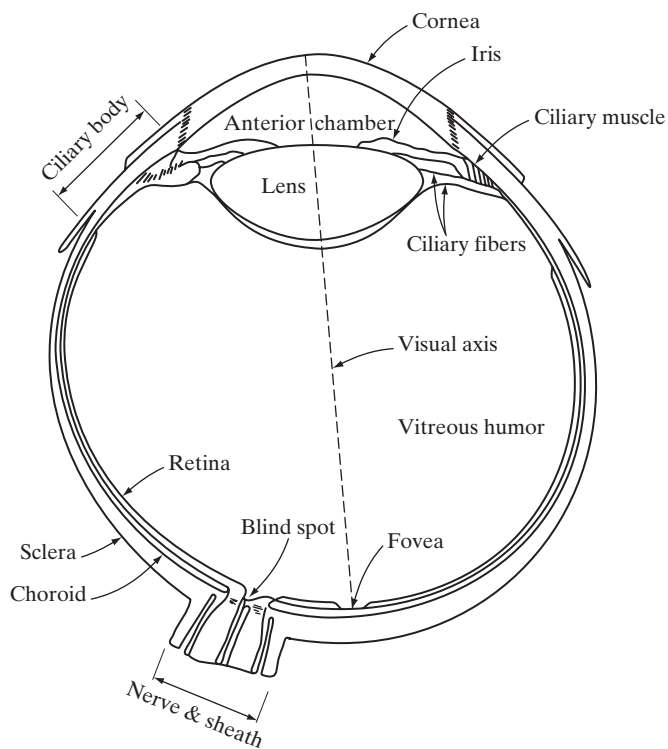
## 2.1 Elements of Visual Perception

Although the field of digital image processing is built on a foundation of mathematical and probabilistic formulations, human intuition and analysis play a central role in the choice of one technique versus another, and this choice often is made based on subjective, visual judgments. Hence, developing a basic understanding of human visual perception as a first step in our journey through this book is appropriate. Given the complexity and breadth of this topic, we can only aspire to cover the most rudimentary aspects of human vision. In particular, our interest is in the mechanics and parameters related to 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. Thus, 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 also are important from a practical point of view.

### 2.1.1 Structure of the Human Eye

Figure 2.1 shows a simplified horizontal cross section of the human eye. The eye is nearly a sphere, with an average diameter of approximately 20 mm. Three membranes enclose the eye: the *cornea* and *sclera* outer cover; the *choroid*; and the *retina*. The cornea is a tough, transparent tissue that covers

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



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 injury to the choroid, often not deemed serious, can lead to severe eye damage as a result of inflammation that restricts blood flow. The choroid coat is heavily pigmented and hence helps to 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.

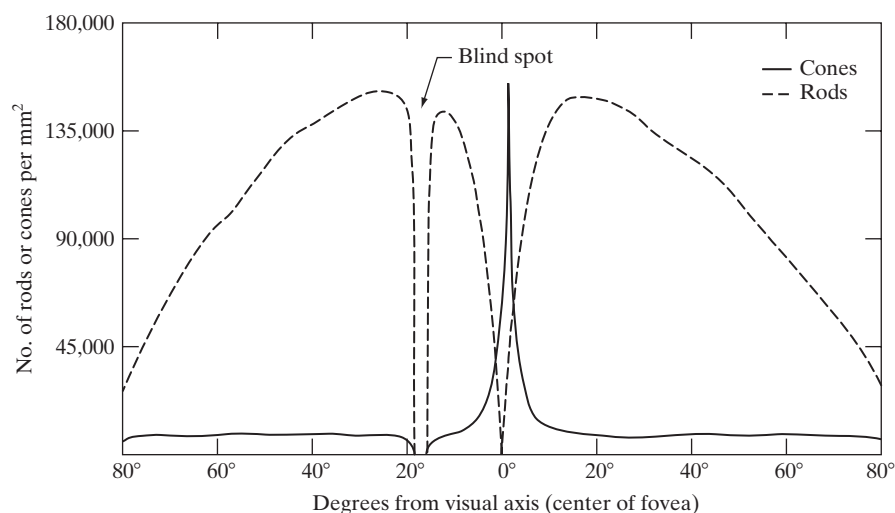
The *lens* is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body. It contains 60 to 70% water, about 6% fat, and more protein than any other tissue in the eye. The lens is colored by a slightly yellow pigmentation that increases with age. In extreme cases, excessive clouding of the lens, caused by the affliction commonly 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 relatively higher absorption at shorter wavelengths. Both infrared and ultraviolet light are absorbed appreciably by proteins within the lens structure 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 properly focused, light from an object outside the eye is imaged on the retina. Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina. There are two classes of receptors: *cones* and *rods*. The cones in each eye number between 6 and 7 million. 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 with these cones largely because each one is connected to its own nerve end. Muscles controlling the eye rotate the eyeball until the image of an object 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 retinal surface. The larger area of distribution and the fact that several rods are connected to a single nerve end reduce the amount of detail discernible by these receptors. Rods serve to give a general, overall picture 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 when seen by moonlight appear as colorless forms 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 of emergence of the optic nerve from the eye. The absence of receptors in this area results in 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

**FIGURE 2.2**  
Distribution of  
rods and cones in  
the retina.

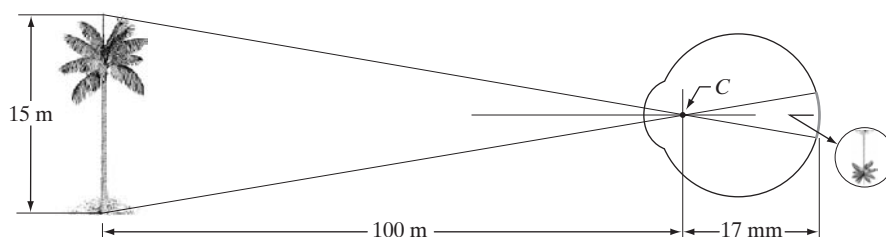


fovea (that is, in degrees off axis, as measured by the angle formed by the visual axis and a line passing through the center of the lens and intersecting the retina). Note in Fig. 2.2 that cones are most dense in the center of the retina (in the center area of the fovea). Note also that rods increase in density from the center out to approximately 20° off axis and then decrease in density out to the extreme periphery of the retina.

The fovea itself is a circular indentation in the retina of about 1.5 mm in diameter. However, in terms of future discussions, talking about square or rectangular arrays of sensing elements is more useful. Thus, by taking some liberty in interpretation, we can view the fovea as a square sensor array of size 1.5 mm × 1.5 mm. The density of cones in that area of the retina is approximately 150,000 elements per mm<sup>2</sup>. Based on these approximations, the number of cones in the region of highest acuity in the eye is about 337,000 elements. Just in terms of raw resolving power, a charge-coupled device (CCD) imaging chip of medium resolution can have this number of elements in a receptor array no larger than 5 mm × 5 mm. While the ability of humans to integrate intelligence and experience with vision makes these types of number comparisons somewhat superficial, keep in mind for future discussions that the basic ability of the eye to resolve detail certainly is comparable to current electronic imaging sensors.

### 2.1.2 Image Formation in the Eye

In an ordinary photographic camera, the lens has a fixed focal length, and 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 lens and the imaging region (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, flattening or thickening the



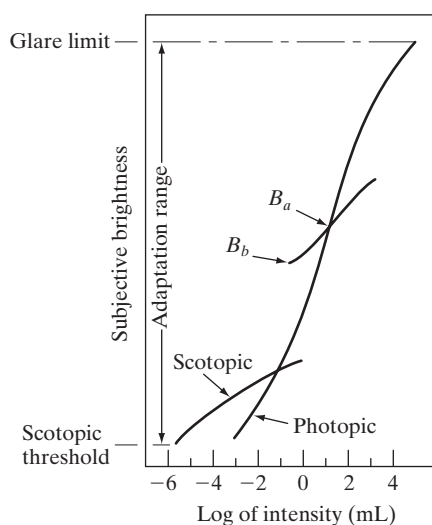
**FIGURE 2.3**  
Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.

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  $15/100 = h/17$  or  $h = 2.55$  mm. As indicated in Section 2.1.1, 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.

### 2.1.3 Brightness Adaptation and Discrimination

Because digital images are displayed as a discrete set of intensities, the eye's ability to discriminate between different intensity levels is an important consideration 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  $10^{10}$ —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



**FIGURE 2.4**  
Range of subjective brightness sensations showing a particular adaptation level.

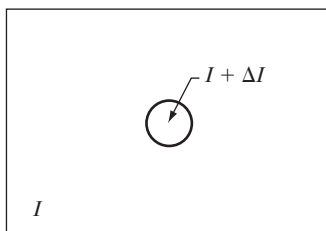
of light intensity versus subjective brightness, illustrates this characteristic. The long solid curve represents the range of intensities to which the visual system can adapt. In photopic vision alone, the range is about  $10^6$ . The transition from scotopic to photopic vision is gradual over the approximate 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.

The essential point in interpreting the impressive dynamic range depicted in Fig. 2.4 is that the visual system cannot operate over such a range *simultaneously*. Rather, it accomplishes this large variation by changing its overall sensitivity, a phenomenon known as *brightness adaptation*. The total range of distinct intensity levels the eye can discriminate simultaneously is rather small when compared with the total adaptation range. For any given set of conditions, the current sensitivity level of the visual system is called the *brightness adaptation level*, which may correspond, for example, to brightness  $B_a$  in Fig. 2.4. The short intersecting curve represents the range of subjective brightness that the eye can perceive when adapted to this level. This range is rather restricted, having a level  $B_b$  at and below which all stimuli are perceived as indistinguishable blacks. The upper portion of the curve is not actually restricted but, if extended too far, loses its meaning because much higher intensities would simply raise the adaptation level higher than  $B_a$ .

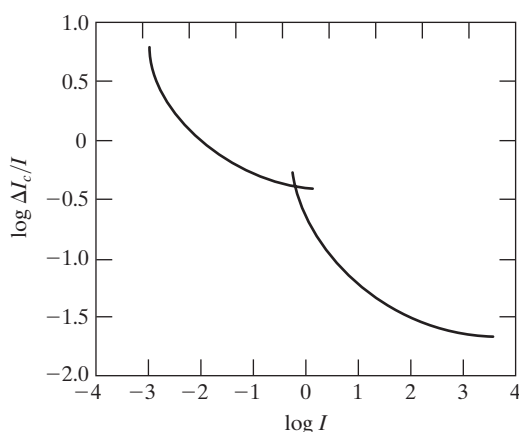
The ability of the eye to discriminate between *changes* in light intensity at any specific adaptation level is also of considerable interest. A classic experiment used to determine the capability of the human visual system for brightness discrimination consists of having a subject look at a flat, uniformly illuminated area large enough to occupy the entire field of view. This area typically is a diffuser, such as opaque glass, that is illuminated from behind by a light source whose intensity,  $I$ , can be varied. To this field is added an increment of illumination,  $\Delta I$ , in the form of a short-duration flash that appears as a circle in the center of the uniformly illuminated field, as Fig. 2.5 shows.

If  $\Delta I$  is not bright enough, the subject says “no,” indicating no perceivable change. As  $\Delta I$  gets stronger, the subject may give a positive response of “yes,” indicating a perceived change. Finally, when  $\Delta I$  is strong enough, the subject will give a response of “yes” all the time. The quantity  $\Delta I_c/I$ , where  $\Delta I_c$  is the increment of illumination discriminable 50% of the time with background illumination  $I$ , is called the *Weber ratio*. A small value of  $\Delta I_c/I$  means that a small percentage change in intensity is discriminable. This represents “good” brightness discrimination. Conversely, a large value of  $\Delta I_c/I$  means that a large percentage change in intensity is required. This represents “poor” brightness discrimination.

**FIGURE 2.5** Basic experimental setup used to characterize brightness discrimination.







**FIGURE 2.6**  
Typical Weber ratio as a function of intensity.

A plot of  $\log \Delta I_c/I$  as a function of  $\log I$  has the general shape shown in Fig. 2.6. This curve shows that brightness discrimination is poor (the Weber ratio is large) at low levels of illumination, and it improves significantly (the Weber ratio decreases) as background illumination increases. The two branches in the curve reflect the fact that at low levels of illumination vision is carried out by the rods, whereas at high levels (showing better discrimination) vision is the function of cones.

If the background illumination is held constant and the intensity of the other source, instead of flashing, is now allowed to vary incrementally from never being perceived to always being perceived, the typical observer can discern a total of one to two dozen different intensity changes. Roughly, this result is related to the number of different intensities a person can see at any one point in a monochrome image. This result does not mean that an image can be represented by such a small number of intensity values because, as the eye roams about the image, the average background changes, thus allowing a *different* set of incremental changes to be detected at each new adaptation level. The net consequence is that the eye is capable of a much broader range of *overall* intensity discrimination. In fact, we show in Section 2.4.3 that the eye is capable of detecting objectionable contouring effects in monochrome images whose overall intensity is represented by fewer than approximately two dozen levels.

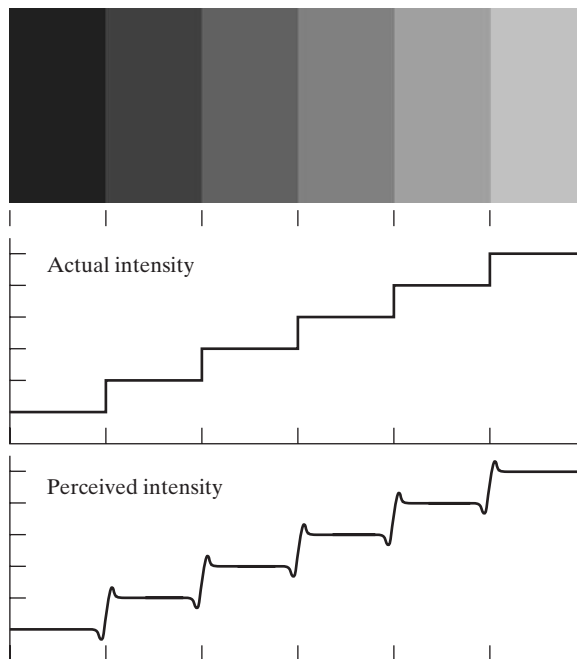
Two phenomena clearly demonstrate that perceived brightness is not a simple function of intensity. The first is based on the fact that the visual system tends to undershoot or overshoot around the boundary of regions of different intensities. Figure 2.7(a) shows a striking example of this phenomenon. Although the intensity of the stripes is constant, we actually perceive a brightness pattern that is strongly scalloped near the boundaries [Fig. 2.7(c)]. These seemingly scalloped bands are called *Mach bands* after Ernst Mach, who first described the phenomenon in 1865.

The second phenomenon, called *simultaneous contrast*, is related to the fact that a region's perceived brightness does not depend simply on its intensity, as Fig. 2.8 demonstrates. All the center squares have exactly the same intensity.

## 42 Chapter 2 ■ Digital Image Fundamentals

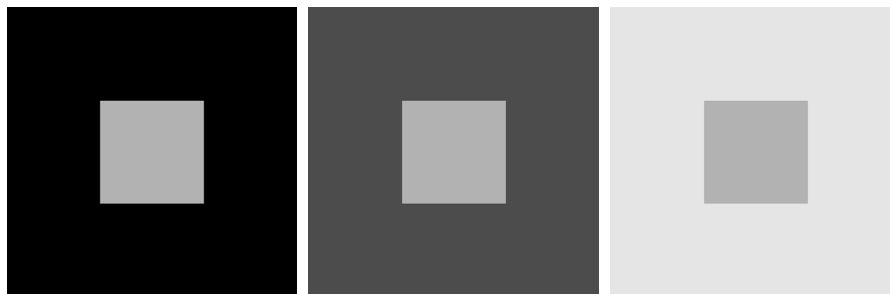
a  
b  
c

**FIGURE 2.7**  
Illustration of the  
Mach band effect.  
Perceived  
intensity is not a  
simple function of  
actual intensity.



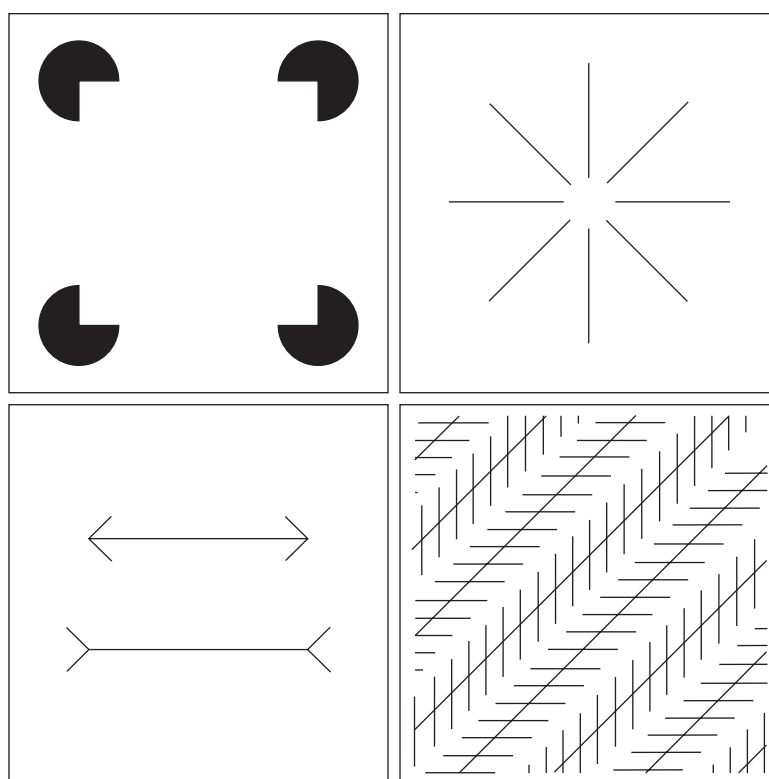
However, they appear to the eye to become darker as the background gets lighter. A more familiar example is a piece of paper that seems white when lying on a desk, but can appear totally black when used to shield the eyes while looking directly at a bright sky.

Other examples of human perception phenomena are optical illusions, in which the eye fills in nonexistent information or wrongly perceives geometrical properties of objects. Figure 2.9 shows some examples. In Fig. 2.9(a), the outline of a square is seen clearly, despite the fact that no lines defining such a figure are part of the image. The same effect, this time with a circle, can be seen in Fig. 2.9(b); note how just a few lines are sufficient to give the illusion of a



a b c

**FIGURE 2.8** Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.



a b  
c d

**FIGURE 2.9** Some well-known optical illusions.

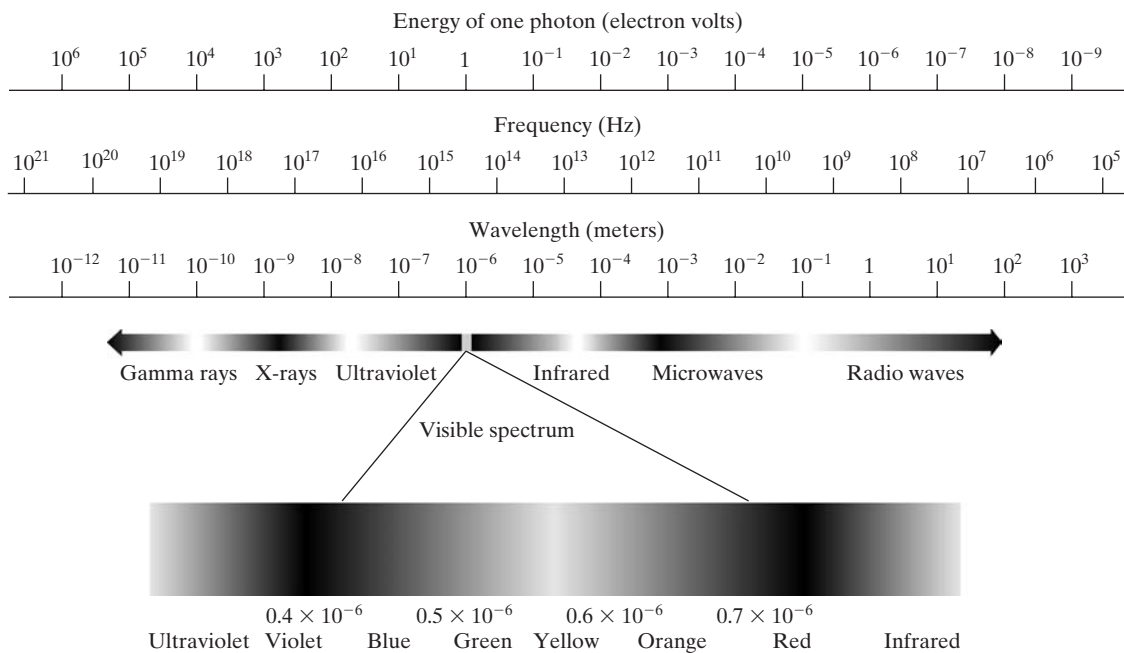
complete circle. The two horizontal line segments in Fig. 2.9(c) are of the same length, but one appears shorter than the other. Finally, all lines in Fig. 2.9(d) that are oriented at  $45^\circ$  are equidistant and parallel. Yet the crosshatching creates the illusion that those lines are far from being parallel. Optical illusions are a characteristic of the human visual system that is not fully understood.

## 2.2 Light and the Electromagnetic Spectrum

The electromagnetic spectrum was introduced in Section 1.3. We now consider this topic in more detail. In 1666, Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other. As Fig. 2.10 shows, the range of colors we perceive in visible light represents a very small portion of the electromagnetic spectrum. On one end of the spectrum are radio waves with wavelengths billions of times longer than those of visible light. On the other end of the spectrum are gamma rays with wavelengths millions of times smaller than those of visible light. The electromagnetic spectrum can be expressed in terms of wavelength, frequency, or energy. Wavelength ( $\lambda$ ) and frequency ( $\nu$ ) are related by the expression

$$\lambda = \frac{c}{\nu} \quad (2.2-1)$$

## 44 Chapter 2 ■ Digital Image Fundamentals



**FIGURE 2.10** The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

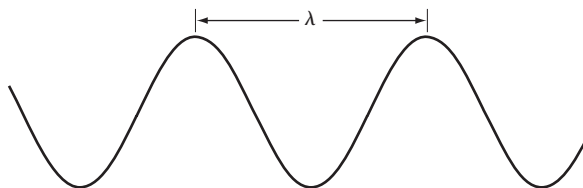
where  $c$  is the speed of light ( $2.998 \times 10^8$  m/s). The energy of the various components of the electromagnetic spectrum is given by the expression

$$E = h\nu \quad (2.2-2)$$

where  $h$  is Planck's constant. The units of wavelength are meters, with the terms *microns* (denoted  $\mu\text{m}$  and equal to  $10^{-6}$  m) and *nanometers* (denoted nm and equal to  $10^{-9}$  m) being used just as frequently. Frequency is measured in Hertz (Hz), with one Hertz being equal to one cycle of a sinusoidal wave per second. A commonly used unit of energy is the electron-volt.

Electromagnetic waves can be visualized as propagating sinusoidal waves with wavelength  $\lambda$  (Fig. 2.11), or they can be thought of as a stream of massless particles, each traveling in a wavelike pattern and moving at the speed of light. Each massless particle contains a certain amount (or bundle) of energy. Each

**FIGURE 2.11**  
Graphical representation of one wavelength.



bundle of energy is called a *photon*. We see from Eq. (2.2-2) that energy is proportional to frequency, so the higher-frequency (shorter wavelength) electromagnetic phenomena carry more energy per photon. Thus, radio waves have photons with low energies, microwaves have more energy than radio waves, infrared still more, then visible, ultraviolet, X-rays, and finally gamma rays, the most energetic of all. This is the reason why gamma rays are so dangerous to living organisms.

Light is a particular type of electromagnetic radiation that can be sensed by the human eye. The visible (color) spectrum is shown expanded in Fig. 2.10 for the purpose of discussion (we consider color in much more detail in Chapter 6). The visible band of the electromagnetic spectrum spans the range from approximately  $0.43 \mu\text{m}$  (violet) to about  $0.79 \mu\text{m}$  (red). For convenience, the color spectrum is divided into six broad regions: violet, blue, green, yellow, orange, and red. No color (or other component of the electromagnetic spectrum) ends abruptly, but rather each range blends smoothly into the next, as shown in Fig. 2.10.

The colors that humans perceive in an object are determined by the nature of the light *reflected* from the object. A body that reflects light relatively balanced in all visible wavelengths appears white to the observer. However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color. For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths.

Light that is void of color is called *monochromatic* (or *achromatic*) light. The only attribute of monochromatic light is its *intensity* or amount. Because the intensity of monochromatic light is perceived to vary from black to grays and finally to white, the term *gray level* is used commonly to denote monochromatic intensity. We use the terms *intensity* and *gray level* interchangeably in subsequent discussions. The range of measured values of monochromatic light from black to white is usually called the *gray scale*, and monochromatic images are frequently referred to as *gray-scale images*.

*Chromatic (color) light* spans the electromagnetic energy spectrum from approximately  $0.43$  to  $0.79 \mu\text{m}$ , as noted previously. In addition to frequency, three basic quantities are used to describe the quality of a chromatic light source: radiance, luminance, and brightness. *Radiance* is the total amount of energy that flows from the light source, and it is usually measured in watts (W). *Luminance*, measured in lumens (lm), gives a measure of the amount of energy an observer *perceives* from a light source. For example, light emitted from a source operating in the far infrared region of the spectrum could have significant energy (radiance), but an observer would hardly perceive it; its luminance would be almost zero. Finally, as discussed in Section 2.1, *brightness* is a subjective descriptor of light perception that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.

Continuing with the discussion of Fig. 2.10, we note that at the short-wavelength end of the electromagnetic spectrum, we have gamma rays and X-rays. As discussed in Section 1.3.1, gamma radiation is important for medical and astronomical imaging, and for imaging radiation in nuclear environments.

Hard (high-energy) X-rays are used in industrial applications. Chest and dental X-rays are in the lower energy (soft) end of the X-ray band. The soft X-ray band transitions into the far ultraviolet light region, which in turn blends with the visible spectrum at longer wavelengths. Moving still higher in wavelength, we encounter the infrared band, which radiates heat, a fact that makes it useful in imaging applications that rely on “heat signatures.” The part of the infrared band close to the visible spectrum is called the *near-infrared* region. The opposite end of this band is called the *far-infrared* region. This latter region blends with the microwave band. This band is well known as the source of energy in microwave ovens, but it has many other uses, including communication and radar. Finally, the radio wave band encompasses television as well as AM and FM radio. In the higher energies, radio signals emanating from certain stellar bodies are useful in astronomical observations. Examples of images in most of the bands just discussed are given in Section 1.3.

In principle, if a sensor can be developed that is capable of detecting energy radiated by a band of the electromagnetic spectrum, we can image events of interest in that band. It is important to note, however, that the wavelength of an electromagnetic wave required to “see” an object must be of the same size as or smaller than the object. For example, a water molecule has a diameter on the order of  $10^{-10}$  m. Thus, to study molecules, we would need a source capable of emitting in the far ultraviolet or soft X-ray region. This limitation, along with the physical properties of the sensor material, establishes the fundamental limits on the capability of imaging sensors, such as visible, infrared, and other sensors in use today.

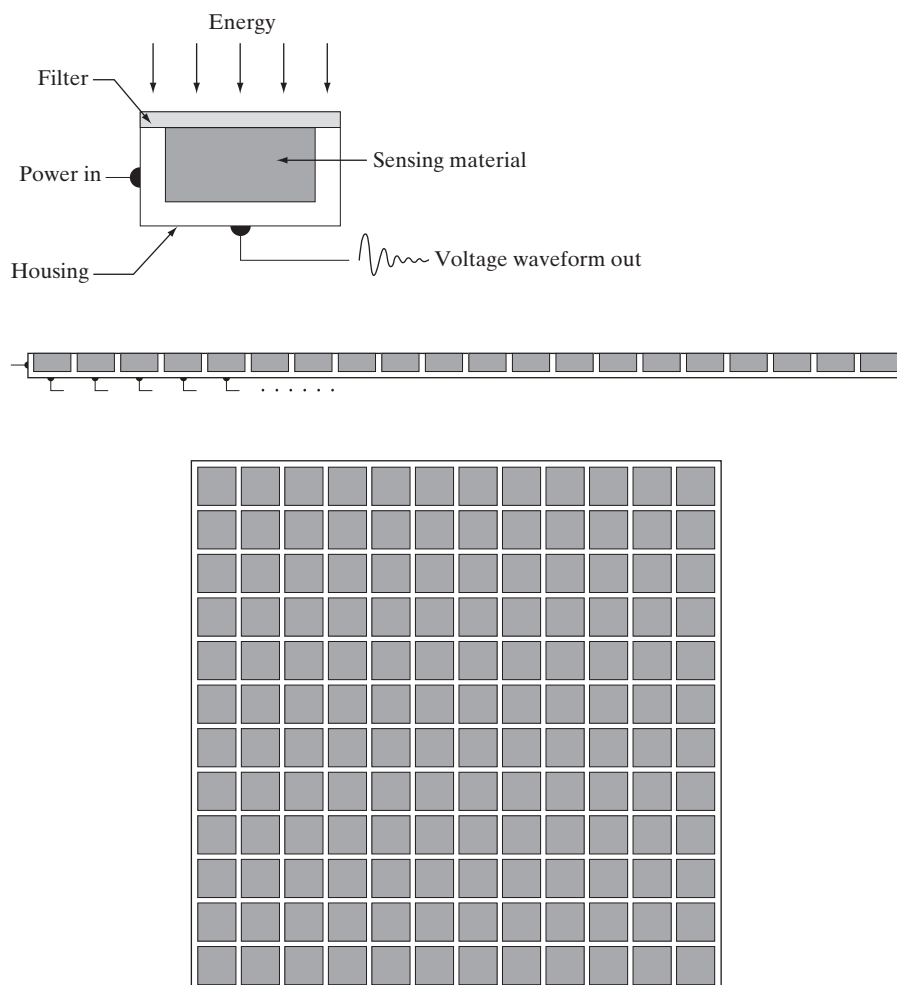
Although imaging is based predominantly on energy radiated by electromagnetic waves, this is not the only method for image generation. For example, as discussed in Section 1.3.7, sound reflected from objects can be used to form ultrasonic images. Other major sources of digital images are electron beams for electron microscopy and synthetic images used in graphics and visualization.

### 2.3 Image Sensing and Acquisition

Most of the images in which we are interested are generated by the combination of an “illumination” source and the reflection or absorption of energy from that source by the elements of the “scene” being imaged. We enclose *illumination* and *scene* in quotes to emphasize the fact that they are considerably more general than the familiar situation in which a visible light source illuminates a common everyday 3-D (three-dimensional) scene. For example, the illumination may originate from a source of electromagnetic energy such as radar, infrared, or X-ray system. But, as noted earlier, it could originate from less traditional sources, such as ultrasound or even a computer-generated illumination pattern. Similarly, the scene elements could be familiar objects, but they can just as easily be molecules, buried rock formations, or a human brain. Depending on the nature of the source, illumination energy is reflected from, or transmitted through, objects. An example in the first category is light

reflected from a planar surface. An example in the second category is when X-rays pass through a patient's body for the purpose of generating a diagnostic X-ray film. In some applications, the reflected or transmitted energy is focused onto a photoconverter (e.g., a phosphor screen), which converts the energy into visible light. Electron microscopy and some applications of gamma imaging use this approach.

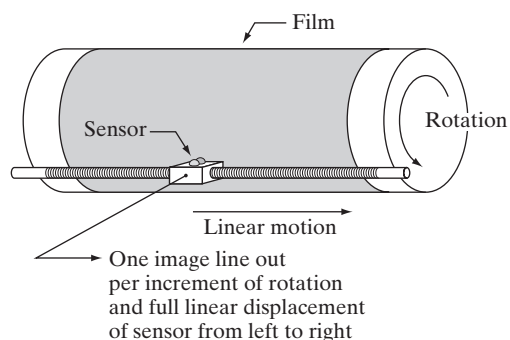
Figure 2.12 shows the three principal sensor arrangements used to transform illumination energy into digital images. The idea is simple: Incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to the particular type of energy being detected. The output voltage waveform is the response of the sensor(s), and a digital quantity is obtained from each sensor by digitizing its response. In this section, we look at the principal modalities for image sensing and generation. Image digitizing is discussed in Section 2.4.



a  
b  
c

**FIGURE 2.12**  
(a) Single imaging sensor.  
(b) Line sensor.  
(c) Array sensor.

**FIGURE 2.13**  
Combining a single sensor with motion to generate a 2-D image.



### 2.3.1 Image Acquisition Using a Single Sensor

Figure 2.12(a) shows the components of a single sensor. Perhaps the most familiar sensor of this type is the photodiode, which is constructed of silicon materials and whose output voltage waveform is proportional to light. The use of a filter in front of a sensor improves selectivity. For example, a green (pass) filter in front of a light sensor favors light in the green band of the color spectrum. As a consequence, the sensor output will be stronger for green light than for other components in the visible spectrum.

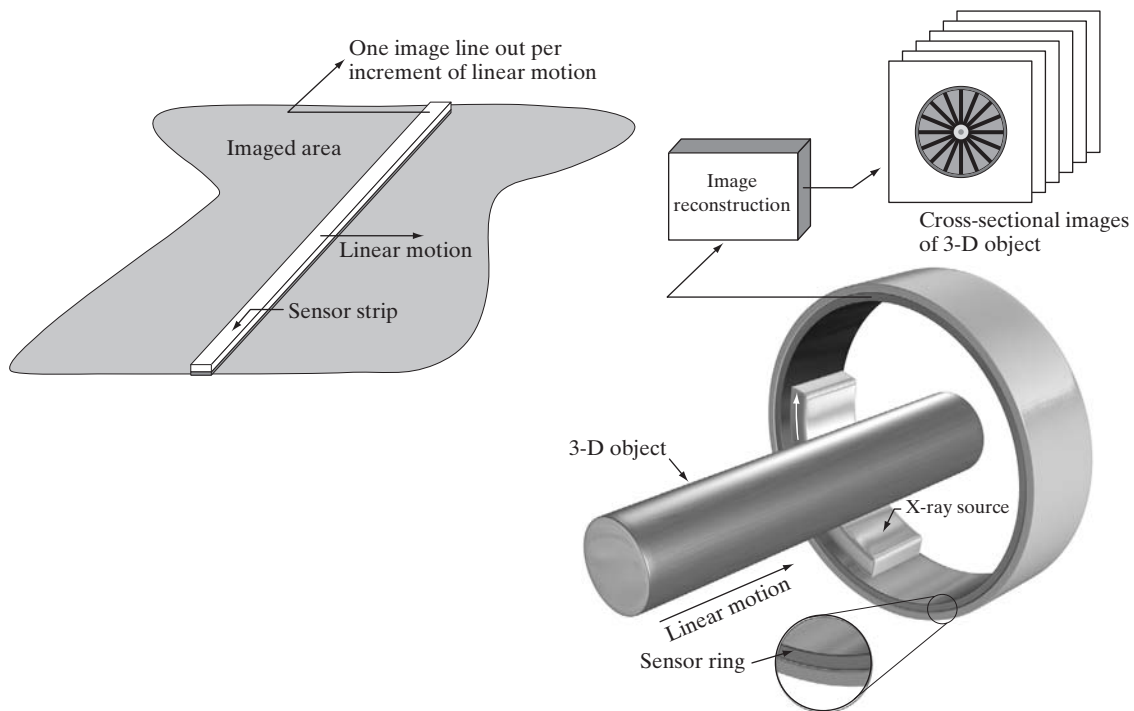
In order to generate a 2-D image using a single sensor, there has to be relative displacements in both the  $x$ - and  $y$ -directions between the sensor and the area to be imaged. Figure 2.13 shows an arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension. The single sensor is mounted on a lead screw that provides motion in the perpendicular direction. Because mechanical motion can be controlled with high precision, this method is an inexpensive (but slow) way to obtain high-resolution images. Other similar mechanical arrangements use a flat bed, with the sensor moving in two linear directions. These types of mechanical digitizers sometimes are referred to as *microdensitometers*.

Another example of imaging with a single sensor places a laser source coincident with the sensor. Moving mirrors are used to control the outgoing beam in a scanning pattern and to direct the reflected laser signal onto the sensor. This arrangement can be used also to acquire images using strip and array sensors, which are discussed in the following two sections.

### 2.3.2 Image Acquisition Using Sensor Strips

A geometry that is used much more frequently than single sensors consists of an in-line arrangement of sensors in the form of a sensor strip, as Fig. 2.12(b) shows. The strip provides imaging elements in one direction. Motion perpendicular to the strip provides imaging in the other direction, as shown in Fig. 2.14(a). This is the type of arrangement used in most flat bed scanners. Sensing devices with 4000 or more in-line sensors are possible. In-line sensors are used routinely in airborne imaging applications, in which the imaging system is mounted on an aircraft that





a b

**FIGURE 2.14** (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

flies at a constant altitude and speed over the geographical area to be imaged. One-dimensional imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted perpendicular to the direction of flight. The imaging strip gives one line of an image at a time, and the motion of the strip completes the other dimension of a two-dimensional image. Lenses or other focusing schemes are used to project the area to be scanned onto the sensors.

Sensor strips mounted in a ring configuration are used in medical and industrial imaging to obtain cross-sectional (“slice”) images of 3-D objects, as Fig. 2.14(b) shows. A rotating X-ray source provides illumination and the sensors opposite the source collect the X-ray energy that passes through the object (the sensors obviously have to be sensitive to X-ray energy). This is the basis for medical and industrial computerized axial tomography (CAT) imaging as indicated in Sections 1.2 and 1.3.2. It is important to note that the output of the sensors must be processed by reconstruction algorithms whose objective is to transform the sensed data into meaningful cross-sectional images (see Section 5.11). In other words, images are not obtained directly from the sensors by motion alone; they require extensive processing. A 3-D digital volume consisting of stacked images is generated as the object is moved in a direction

perpendicular to the sensor ring. Other modalities of imaging based on the CAT principle include magnetic resonance imaging (MRI) and positron emission tomography (PET). The illumination sources, sensors, and types of images are different, but conceptually they are very similar to the basic imaging approach shown in Fig. 2.14(b).

### 2.3.3 Image Acquisition Using Sensor Arrays

Figure 2.12(c) shows individual sensors arranged in the form of a 2-D array. Numerous electromagnetic and some ultrasonic sensing devices frequently are arranged in an array format. This is also the predominant arrangement found in digital cameras. A typical sensor for these cameras is a CCD array, which can be manufactured with a broad range of sensing properties and can be packaged in rugged arrays of  $4000 \times 4000$  elements or more. CCD sensors are used widely in digital cameras and other light sensing instruments. The response of each sensor is proportional to the integral of the light energy projected onto the surface of the sensor, a property that is used in astronomical and other applications requiring low noise images. Noise reduction is achieved by letting the sensor integrate the input light signal over minutes or even hours. Because the sensor array in Fig. 2.12(c) is two-dimensional, its key advantage is that a complete image can be obtained by focusing the energy pattern onto the surface of the array. Motion obviously is not necessary, as is the case with the sensor arrangements discussed in the preceding two sections.

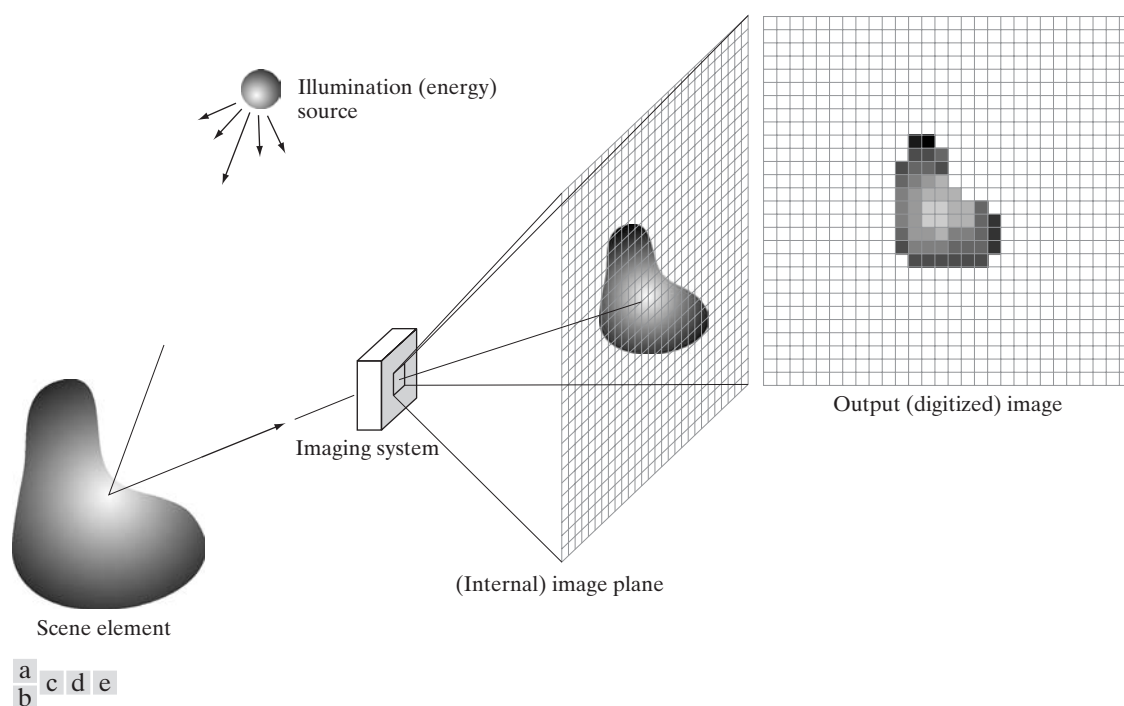
The principal manner in which array sensors are used is shown in Fig. 2.15. This figure shows the energy from an illumination source being reflected from a scene element (as mentioned at the beginning of this section, the energy also could be transmitted through the scene elements). The first function performed by the imaging system in Fig. 2.15(c) is to collect the incoming energy and focus it onto an image plane. If the illumination is light, the front end of the imaging system is an optical lens that projects the viewed scene onto the lens focal plane, as Fig. 2.15(d) shows. The sensor array, which is coincident with the focal plane, produces outputs proportional to the integral of the light received at each sensor. Digital and analog circuitry sweep these outputs and convert them to an analog signal, which is then digitized by another section of the imaging system. The output is a digital image, as shown diagrammatically in Fig. 2.15(e). Conversion of an image into digital form is the topic of Section 2.4.

### 2.3.4 A Simple Image Formation Model

As introduced in Section 1.1, we denote images by two-dimensional functions of the form  $f(x, y)$ . The value or amplitude of  $f$  at spatial coordinates  $(x, y)$  is a positive scalar quantity whose physical meaning is determined by the source of the image. When an image is generated from a physical process, its intensity values are proportional to energy radiated by a physical source (e.g., electromagnetic waves). As a consequence,  $f(x, y)$  must be nonzero

In some cases, we image the source directly, as in obtaining images of the sun.

Image intensities can become negative during processing or as a result of interpretation. For example, in radar images objects moving toward a radar system often are interpreted as having negative velocities while objects moving away are interpreted as having positive velocities. Thus, a velocity image might be coded as having both positive and negative values. When storing and displaying images, we normally scale the intensities so that the smallest negative value becomes 0 (see Section 2.6.3 regarding intensity scaling).



**FIGURE 2.15** An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

and finite; that is,

$$0 < f(x, y) < \infty \quad (2.3-1)$$

The function  $f(x, y)$  may be characterized by two components: (1) the amount of source illumination incident on the scene being viewed, and (2) the amount of illumination reflected by the objects in the scene. Appropriately, these are called the *illumination* and *reflectance* components and are denoted by  $i(x, y)$  and  $r(x, y)$ , respectively. The two functions combine as a product to form  $f(x, y)$ :

$$f(x, y) = i(x, y)r(x, y) \quad (2.3-2)$$

where

$$0 < i(x, y) < \infty \quad (2.3-3)$$

and

$$0 < r(x, y) < 1 \quad (2.3-4)$$

Equation (2.3-4) indicates that reflectance is bounded by 0 (total absorption) and 1 (total reflectance). The nature of  $i(x, y)$  is determined by the illumination source, and  $r(x, y)$  is determined by the characteristics of the imaged objects. It is noted that these expressions also are applicable to images formed via transmission of the illumination through a medium, such as a chest X-ray.

In this case, we would deal with a *transmissivity* instead of a *reflectivity* function, but the limits would be the same as in Eq. (2.3-4), and the image function formed would be modeled as the product in Eq. (2.3-2).

**EXAMPLE 2.1:** Some typical values of illumination and reflectance.

■ The values given in Eqs. (2.3-3) and (2.3-4) are theoretical bounds. The following *average* numerical figures illustrate some typical ranges of  $i(x, y)$  for visible light. On a clear day, the sun may produce in excess of 90,000 lm/m<sup>2</sup> of illumination on the surface of the Earth. This figure decreases to less than 10,000 lm/m<sup>2</sup> on a cloudy day. On a clear evening, a full moon yields about 0.1 lm/m<sup>2</sup> of illumination. The typical illumination level in a commercial office is about 1000 lm/m<sup>2</sup>. Similarly, the following are typical values of  $r(x, y)$ : 0.01 for black velvet, 0.65 for stainless steel, 0.80 for flat-white wall paint, 0.90 for silver-plated metal, and 0.93 for snow. ■

Let the intensity (gray level) of a monochrome image at any coordinates  $(x_0, y_0)$  be denoted by

$$\ell = f(x_0, y_0) \quad (2.3-5)$$

From Eqs. (2.3-2) through (2.3-4), it is evident that  $\ell$  lies in the range

$$L_{\min} \leq \ell \leq L_{\max} \quad (2.3-6)$$

In theory, the only requirement on  $L_{\min}$  is that it be positive, and on  $L_{\max}$  that it be finite. In practice,  $L_{\min} = i_{\min} r_{\min}$  and  $L_{\max} = i_{\max} r_{\max}$ . Using the preceding average office illumination and range of reflectance values as guidelines, we may expect  $L_{\min} \approx 10$  and  $L_{\max} \approx 1000$  to be typical limits for indoor values in the absence of additional illumination.

The interval  $[L_{\min}, L_{\max}]$  is called the *gray* (or *intensity*) *scale*. Common practice is to shift this interval numerically to the interval  $[0, L - 1]$ , where  $\ell = 0$  is considered black and  $\ell = L - 1$  is considered white on the gray scale. All intermediate values are shades of gray varying from black to white.

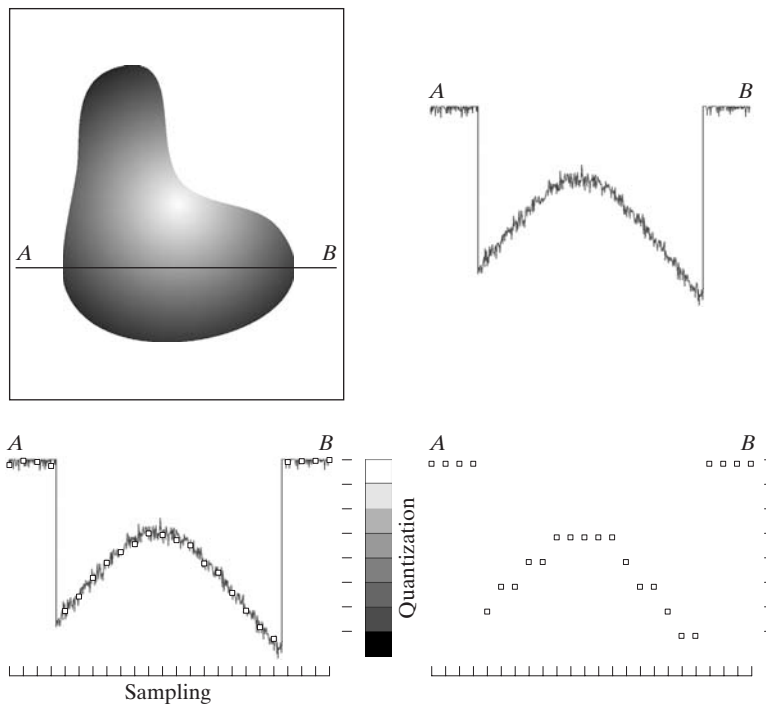
## 2.4 Image Sampling and Quantization

From the discussion in the preceding section, we see that 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 digital form. This involves two processes: *sampling* and *quantization*.

### 2.4.1 Basic Concepts in Sampling and Quantization

The basic idea behind sampling and quantization is illustrated in Fig. 2.16. 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 amplitude. To convert it to digital form, we have to sample the

The discussion of sampling in this section is of an intuitive nature. We consider this topic in depth in Chapter 4.



a	b
c	d

**FIGURE 2.16**

Generating a digital image. (a) Continuous image. (b) A scan line from  $A$  to  $B$  in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

function in both coordinates and in amplitude. Digitizing the coordinate values is called *sampling*. Digitizing the amplitude values is called *quantization*.

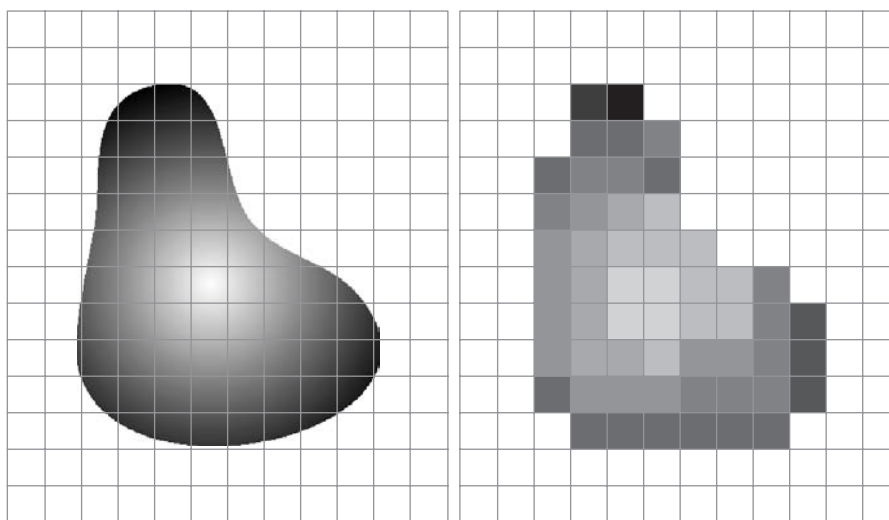
The one-dimensional function in Fig. 2.16(b) is a plot of amplitude (intensity level) values of the continuous image along the line segment  $AB$  in Fig. 2.16(a). The random variations are due to image noise. To sample this function, we take equally spaced samples along line  $AB$ , as shown in Fig. 2.16(c). The spatial location of each sample is indicated by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives 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 right side of Fig. 2.16(c) shows 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. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Fig. 2.16(d). Starting at the top of the image and carrying out this procedure 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.

Sampling in the manner just described assumes that we have a continuous image in both coordinate directions as well as in amplitude. 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 made very exact so, in principle, there is almost no limit as to 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 of the system.

When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors 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, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Quantization of the sensor outputs is as before. Figure 2.17 illustrates this concept. Figure 2.17(a) shows a continuous image projected onto the plane of an array sensor. Figure 2.17(b) shows the image after sampling and quantization. Clearly, 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 show in Section 2.4.3, image content is also an important consideration in choosing these parameters.



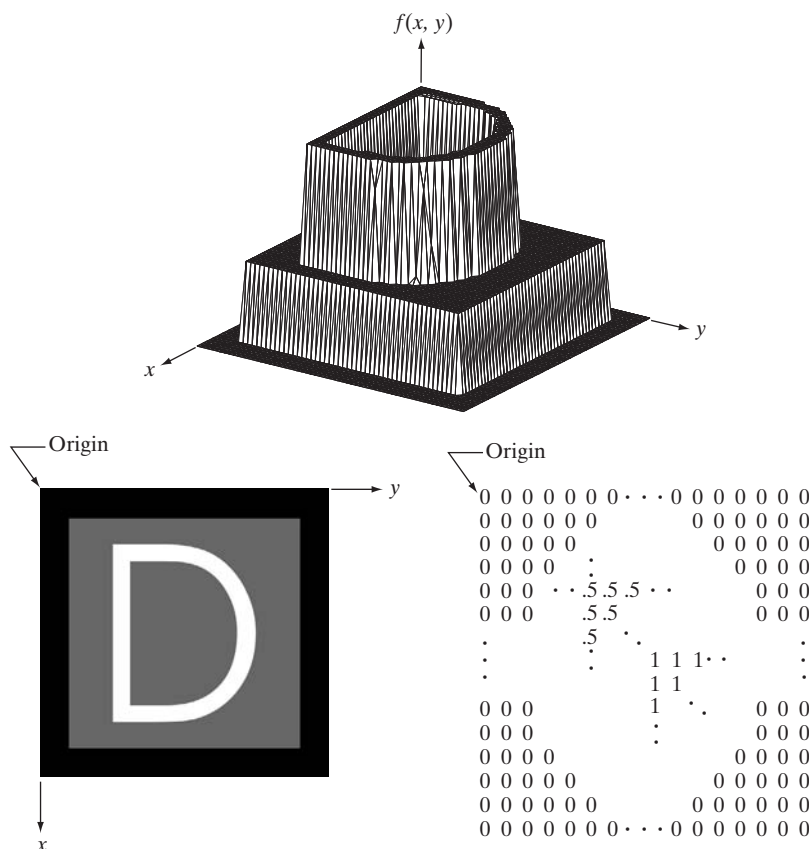
a b

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

## 2.4.2 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 2-D array,  $f(x, y)$ , 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:  $x = 0, 1, 2, \dots, M - 1$  and  $y = 0, 1, 2, \dots, N - 1$ . Thus, for example, the value of the digital image at the origin is  $f(0, 0)$ , and the next coordinate value along the first row is  $f(0, 1)$ . Here, the notation  $(0, 1)$  is used to signify 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 the image at any coordinates  $(x, y)$  is denoted  $f(x, y)$ , where  $x$  and  $y$  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*.

As Fig. 2.18 shows, there are three basic ways to represent  $f(x, y)$ . Figure 2.18(a) is a plot of the function, with two axes determining spatial location



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 (0, .5, and 1 represent black, gray, and white, respectively).

and the third axis being the values of  $f$  (intensities) as a function of the two spatial variables  $x$  and  $y$ . Although we can infer the structure of the image in this example by looking at the plot, complex images generally are too detailed and difficult to interpret from such plots. This representation is useful when working with gray-scale 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 work with this representation in Section 2.6.4.

The representation in Fig. 2.18(b) is much more common. It shows  $f(x, y)$  as it would appear on a monitor or photograph. Here, the intensity of each point 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 simply converts these three values to black, gray, or white, respectively, as Fig. 2.18(b) shows. The third representation is simply to display the numerical values of  $f(x, y)$  as an array (matrix). In this example,  $f$  is of size  $600 \times 600$  elements, or 360,000 numbers. Clearly, printing the complete array would be cumbersome and convey little information. When developing algorithms, however, this representation is quite useful when only parts of the image are printed and analyzed as numerical values. Figure 2.18(c) conveys this concept graphically.

We conclude from the previous paragraph that the representations in Figs. 2.18(b) and (c) are the most useful. Image displays allow us to view results at a glance. Numerical arrays are used for processing and algorithm development. In equation form, we write the representation of an  $M \times N$  numerical array as

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix} \quad (2.4-1)$$

Both sides of this equation are equivalent ways of expressing a digital image quantitatively. The right side is a matrix of real numbers. Each element of this matrix is called an *image element*, *picture element*, *pixel*, or *pel*. The terms *image* and *pixel* are used throughout the book to denote a digital image and its elements.

In some discussions it is advantageous to use a more traditional matrix notation to denote a digital image and its elements:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (2.4-2)$$



Clearly,  $a_{ij} = f(x = i, y = j) = f(i, j)$ , so Eqs. (2.4-1) and (2.4-2) are identical matrices. We can even represent an image as a vector,  $\mathbf{v}$ . For example, a column vector of size  $MN \times 1$  is formed by letting the first  $M$  elements of  $\mathbf{v}$  be the first column of  $\mathbf{A}$ , the next  $M$  elements be the second column, and so on. Alternatively, we can use the rows instead of the columns of  $\mathbf{A}$  to form such a vector. Either representation is valid, as long as we are consistent.

Returning briefly to Fig. 2.18, note that the origin of a digital image is at the top left, with the positive  $x$ -axis extending downward and the positive  $y$ -axis extending to the right. This is a conventional representation 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, so choosing the origin of  $f(x, y)$  at that point makes sense mathematically. Keep in mind that this representation is the standard right-handed Cartesian coordinate system with which you are familiar.<sup>†</sup> We simply show the axes pointing downward and to the right, instead of to the right and up.

Expressing sampling and quantization in more formal mathematical terms can be useful at times. Let  $Z$  and  $R$  denote the set of integers and the set of real numbers, respectively. The sampling process may be viewed as partitioning the  $xy$ -plane into a grid, with the coordinates of the center of each cell in the grid being a pair of elements from the Cartesian product  $Z^2$ , which is the set of all ordered pairs of elements  $(z_i, z_j)$ , with  $z_i$  and  $z_j$  being integers from  $Z$ . Hence,  $f(x, y)$  is a digital image if  $(x, y)$  are integers from  $Z^2$  and  $f$  is a function that assigns an intensity value (that is, a real number from the set of real numbers,  $R$ ) to each distinct pair of coordinates  $(x, y)$ . This functional assignment is the quantization process described earlier. If the intensity levels also are integers (as usually is the case in this and subsequent chapters),  $Z$  replaces  $R$ , and a digital image then becomes a 2-D function whose coordinates and amplitude values are integers.

This digitization process requires that decisions be made regarding the values for  $M$ ,  $N$ , and for the number,  $L$ , of discrete intensity levels. There are no restrictions placed on  $M$  and  $N$ , other than they have to be positive integers. However, due to storage and quantizing hardware considerations, the number of intensity levels typically is an integer power of 2:

$$L = 2^k \quad (2.4-3)$$

We assume that the discrete levels are equally spaced and that they are integers in the interval  $[0, L - 1]$ . Sometimes, the range of values spanned by the gray scale is referred to informally as the dynamic range. This is a term used in different ways in different fields. Here, we define the *dynamic range* of an imaging system to be the ratio of the maximum measurable intensity to the minimum

Often, it is useful for computation or for algorithm development purposes to scale the  $L$  intensity values to the range  $[0, 1]$ , in which case they cease to be integers. However, in most cases these values are scaled back to the integer range  $[0, L - 1]$  for image storage and display.

<sup>†</sup>Recall that a right-handed coordinate system is such that, when the index of the right hand points in the direction of the positive  $x$ -axis and the middle finger points in the (perpendicular) direction of the positive  $y$ -axis, the thumb points up. As Fig. 2.18(a) shows, this indeed is the case in our image coordinate system.

**FIGURE 2.19** An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, *constant* intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions of an image (e.g., the stem of the rose) masks the lowest detectable true intensity level.



detectable intensity level in the system. As a rule, the upper limit is determined by *saturation* and the lower limit by *noise* (see Fig. 2.19). Basically, dynamic range establishes the lowest and highest intensity levels that a system can represent and, consequently, that an image can have. Closely associated with this concept is image *contrast*, which we define as the difference in intensity between the highest and lowest intensity levels in an image. When an appreciable number of pixels in an image have a high dynamic range, we can expect the image to have high contrast. Conversely, an image with low dynamic range typically has a dull, washed-out gray look. We discuss these concepts in more detail in Chapter 3.

The number,  $b$ , of bits required to store a digitized image is

$$b = M \times N \times k \quad (2.4-4)$$

When  $M = N$ , this equation becomes

$$b = N^2 k \quad (2.4-5)$$

Table 2.1 shows the number of bits required to store square images with various values of  $N$  and  $k$ . The number of intensity levels corresponding to each value of  $k$  is shown in parentheses. When an image can have  $2^k$  intensity levels, it is common practice to refer to the image as a “ $k$ -bit image.” For example, an image with 256 possible discrete intensity values is called an 8-bit image. Note that storage requirements for 8-bit images of size  $1024 \times 1024$  and higher are not insignificant.

**TABLE 2.1**

Number of storage bits for various values of  $N$  and  $k$ .  $L$  is the number of intensity levels.

$N/k$	1 ( $L = 2$ )	2 ( $L = 4$ )	3 ( $L = 8$ )	4 ( $L = 16$ )	5 ( $L = 32$ )	6 ( $L = 64$ )	7 ( $L = 128$ )	8 ( $L = 256$ )
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

### 2.4.3 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 a number of ways, with *line pairs per unit distance*, and *dots (pixels) per unit distance* being among the most 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  $1/2W$  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 (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 commonly 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 is printed at 2400 dpi.

The key point in the preceding paragraph is that, 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. To say that an image has, say, a resolution  $1024 \times 1024$  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 example, 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 samples used to generate a digital image, but this is not true regarding the number

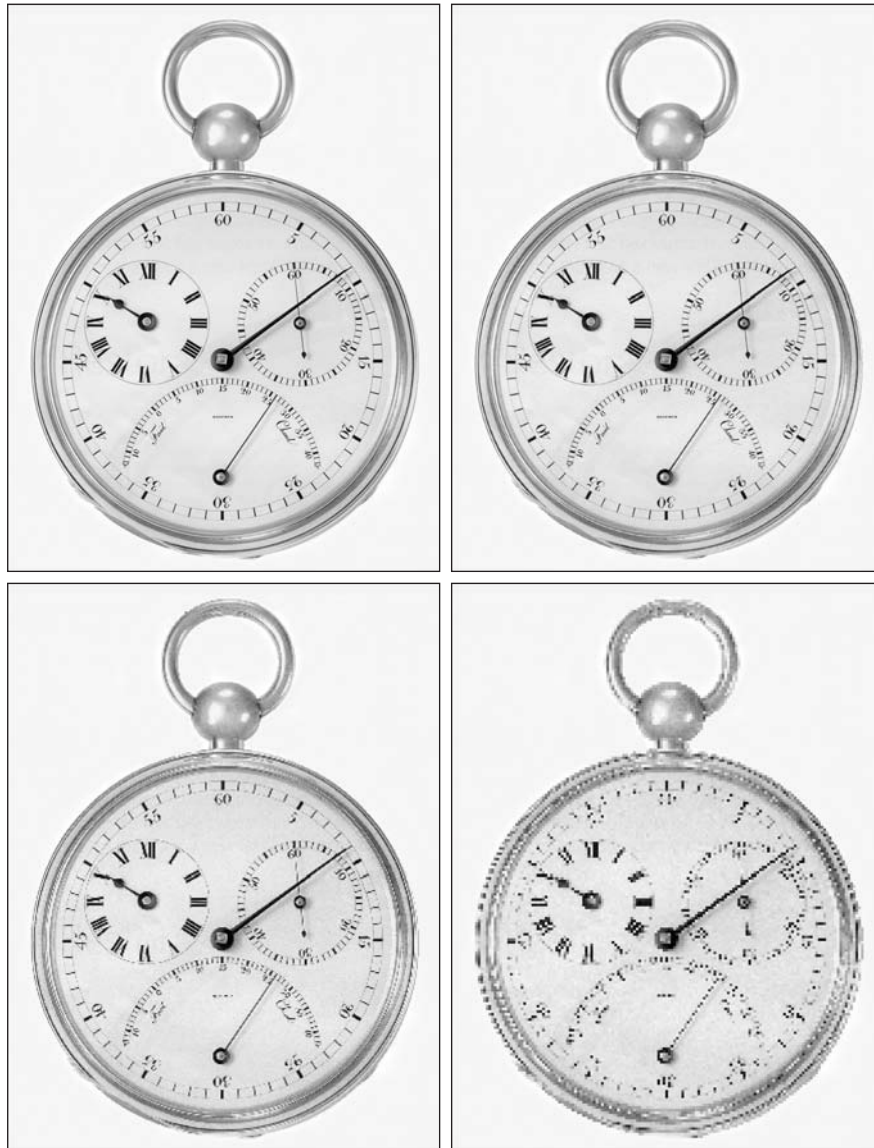
of intensity levels. Based on hardware considerations, the number of intensity levels usually is an integer power of two, as mentioned in the previous section. The most common number is 8 bits, with 16 bits being used in some applications in which enhancement of specific intensity ranges is necessary. 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 the exception, rather than the rule. 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. Because true discernible changes in intensity are influenced not only by noise and saturation values but also by the capabilities of human perception (see Section 2.1), saying that an image has 8 bits of intensity resolution is nothing more than a statement regarding the ability of an 8-bit system to quantize intensity in fixed increments of  $1/256$  units of intensity amplitude.

The following two examples illustrate individually the comparative effects of image size and intensity resolution on discernable detail. Later in this section, we discuss how these two parameters interact in determining perceived image quality.

**EXAMPLE 2.2:**  
Illustration of the effects of reducing image spatial resolution.

■ Figure 2.20 shows the effects of reducing spatial resolution in an image. The images in Figs. 2.20(a) through (d) are shown in 1250, 300, 150, and 72 dpi, respectively. Naturally, the lower resolution images are smaller than the original. For example, the original image is of size  $3692 \times 2812$  pixels, but the 72 dpi image is an array of size  $213 \times 162$ . In order to facilitate comparisons, all the smaller images were zoomed back to the original size (the method used for zooming is discussed in Section 2.4.4). This is somewhat equivalent to “getting closer” to the smaller images so that we can make comparable statements about visible details.

There are some small visual differences between Figs. 2.20(a) and (b), the most notable being a slight distortion in the large black needle. For the most part, however, Fig. 2.20(b) is quite acceptable. In fact, 300 dpi is the typical minimum image spatial resolution used for book publishing, so one would not expect to see much difference here. Figure 2.20(c) begins to show visible degradation (see, for example, the round edges of the chronometer and the small needle pointing to 60 on the right side). Figure 2.20(d) shows degradation that is visible in most features of the image. As we discuss in Section 4.5.4, when printing at such low resolutions, the printing and publishing industry uses a number of “tricks” (such as locally varying the pixel size) to produce much better results than those in Fig. 2.20(d). Also, as we show in Section 2.4.4, it is possible to improve on the results of Fig. 2.20 by the choice of interpolation method used. ■



a b  
c d

**FIGURE 2.20** Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

## 62 Chapter 2 ■ Digital Image Fundamentals

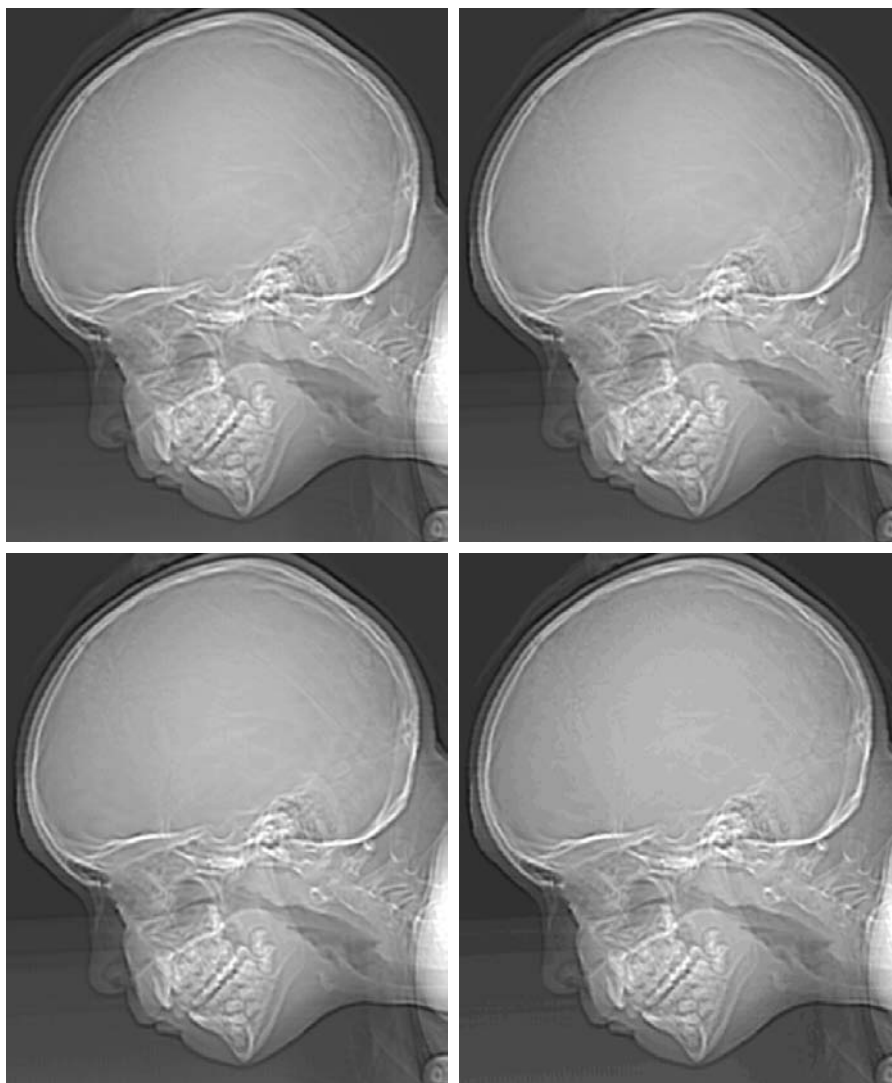
**EXAMPLE 2.3:** Typical effects of varying the number of intensity levels in a digital image.

■ In this example, we keep the number of samples constant and reduce the number of intensity levels from 256 to 2, in integer powers of 2. Figure 2.21(a) is a  $452 \times 374$  CT projection image, displayed with  $k = 8$  (256 intensity levels). Images such as this are obtained by fixing the X-ray source in one position, thus producing a 2-D image in any desired direction. Projection images are used as guides to set up the parameters for a CT scanner, including tilt, number of slices, and range.

Figures 2.21(b) through (h) were obtained by reducing the number of bits from  $k = 7$  to  $k = 1$  while keeping the image size constant at  $452 \times 374$  pixels. The 256-, 128-, and 64-level images are visually identical for all practical purposes. The 32-level image in Fig. 2.21(d), however, has an imperceptible set of

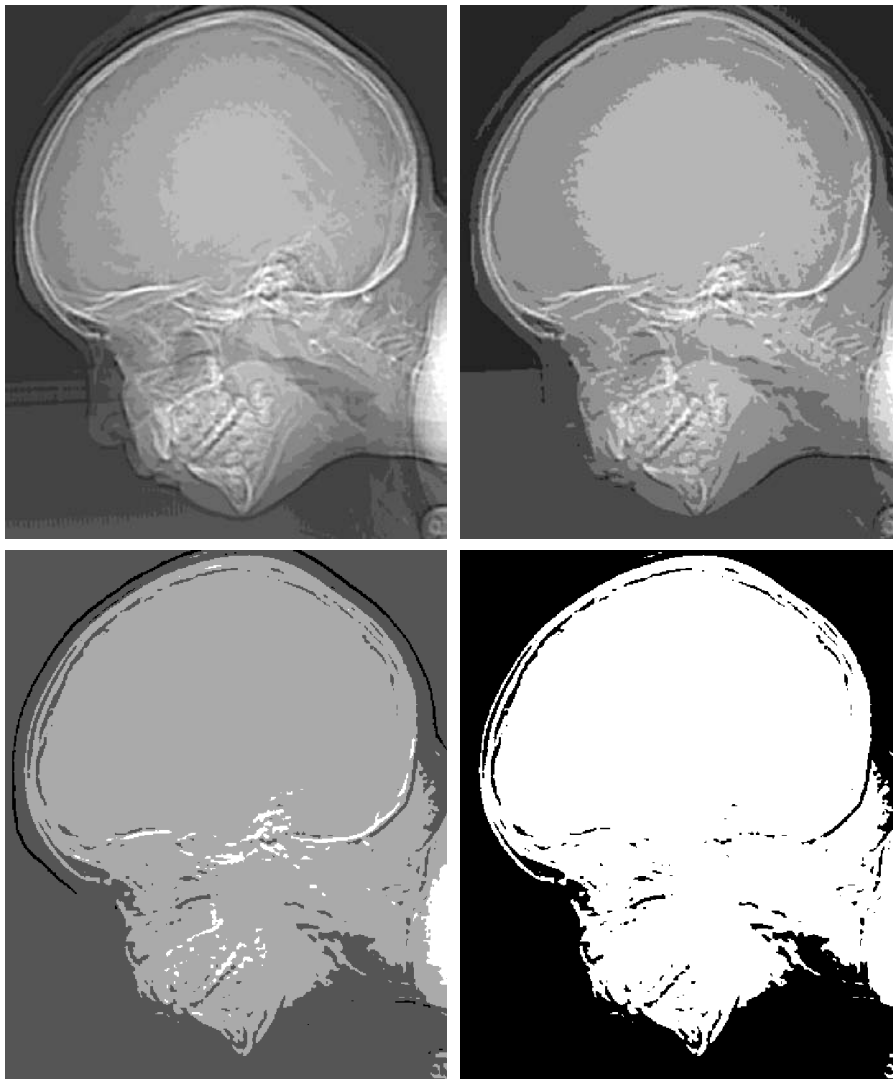
a b  
c d

**FIGURE 2.21**  
(a)  $452 \times 374$ ,  
256-level image.  
(b)–(d) Image  
displayed in 128,  
64, and 32  
intensity levels,  
while keeping the  
image size  
constant.



very fine ridge-like structures in areas of constant or nearly constant intensity (particularly in the skull). This effect, caused by the use of an insufficient number of intensity levels in smooth areas of a digital image, is called *false contouring*, so called because the ridges resemble topographic contours in a map. False contouring generally is quite visible in images displayed using 16 or less uniformly spaced intensity levels, as the images in Figs. 2.21(e) through (h) show.

As a very rough rule of thumb, and assuming integer powers of 2 for convenience, images of size  $256 \times 256$  pixels with 64 intensity levels and printed on a size format on the order of  $5 \times 5$  cm are about the lowest spatial and intensity resolution images that can be expected to be reasonably free of objectionable sampling checkerboards and false contouring. ■



e f  
g h

**FIGURE 2.21**  
(Continued)  
(e)–(h) Image displayed in 16, 8, 4, and 2 intensity levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



a b c

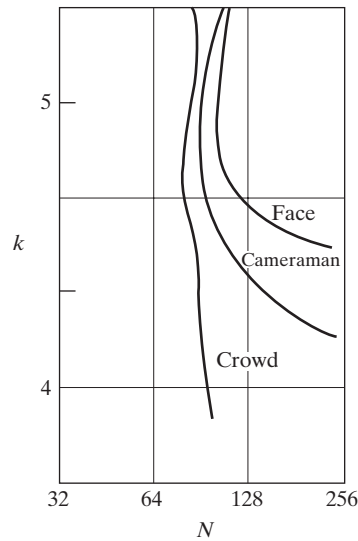
**FIGURE 2.22** (a) Image with a low level of detail. (b) Image with a medium level of detail. (c) Image with a relatively large amount of detail. (Image (b) courtesy of the Massachusetts Institute of Technology.)

The results in Examples 2.2 and 2.3 illustrate the effects produced on image quality by varying  $N$  and  $k$  independently. However, these results only partially answer the question of how varying  $N$  and  $k$  affects images because we have not considered yet any relationships that might exist between these two parameters. An early study by Huang [1965] attempted to quantify experimentally the effects on image quality produced by varying  $N$  and  $k$  simultaneously. The experiment consisted of a set of subjective tests. Images similar to those shown in Fig. 2.22 were used. The woman's face is representative of an image with relatively little detail; the picture of the cameraman contains an intermediate amount of detail; and the crowd picture contains, by comparison, a large amount of detail.

Sets of these three types of images were generated by varying  $N$  and  $k$ , and observers were then asked to rank them according to their subjective quality. Results were summarized in the form of so-called *isopreference curves* in the  $Nk$ -plane. (Figure 2.23 shows average isopreference curves representative of curves corresponding to the images in Fig. 2.22.) Each point in the  $Nk$ -plane represents an image having values of  $N$  and  $k$  equal to the coordinates of that point. Points lying on an isopreference curve correspond to images of equal subjective quality. It was found in the course of the experiments that the isopreference curves tended to shift right and upward, but their shapes in each of the three image categories were similar to those in Fig. 2.23. This is not unexpected, because a shift up and right in the curves simply means larger values for  $N$  and  $k$ , which implies better picture quality.

The key point of interest in the context of the present discussion is that isopreference curves tend to become more vertical as the detail in the image increases. This result suggests that for images with a large amount of detail only a few intensity levels may be needed. For example, the isopreference curve in Fig. 2.23 corresponding to the crowd is nearly vertical. This indicates that, for a fixed value of  $N$ , the perceived quality for this type of image is





**FIGURE 2.23**  
Typical isopreference curves for the three types of images in Fig. 2.22.

nearly independent of the number of intensity levels used (for the range of intensity levels shown in Fig. 2.23). It is of interest also to note that perceived quality in the other two image categories remained the same in some intervals in which the number of samples was increased, but the number of intensity levels actually decreased. The most likely reason for this result is that a decrease in  $k$  tends to increase the apparent contrast, a visual effect that humans often perceive as improved quality in an image.

#### 2.4.4 Image Interpolation

Interpolation is a basic tool used extensively in tasks such as zooming, shrinking, rotating, and geometric corrections. Our principal objective in this section is to introduce interpolation and apply it to image resizing (shrinking and zooming), which are basically image *resampling* methods. Uses of interpolation in applications such as rotation and geometric corrections are discussed in Section 2.6.5. We also return to this topic in Chapter 4, where we discuss image resampling in more detail.

Fundamentally, *interpolation* is the process of using known data to estimate values at unknown locations. We begin the discussion of this topic with a simple example. Suppose that an image of size  $500 \times 500$  pixels has to be enlarged 1.5 times to  $750 \times 750$  pixels. A simple way to visualize zooming is to create an imaginary  $750 \times 750$  grid with the same pixel spacing as the original, and then shrink it so that it fits exactly over the original image. Obviously, the pixel spacing in the shrunken  $750 \times 750$  grid will be less than the pixel spacing in the original image. To perform intensity-level assignment for any point in the overlay, we look for its closest pixel in the original image and assign the intensity of that pixel to the new pixel in the  $750 \times 750$  grid. When we are finished assigning intensities to all the points in the overlay grid, we expand it to the original specified size to obtain the zoomed image.

The method just discussed is called *nearest neighbor interpolation* because it assigns to each new location the intensity of its nearest neighbor in the original image (pixel neighborhoods are discussed formally in Section 2.5). This approach is simple but, as we show later in this section, it has the tendency to produce undesirable artifacts, such as severe distortion of straight edges. For this reason, it is used infrequently in practice. A more suitable approach is *bilinear interpolation*, in which we use the four nearest neighbors to estimate the intensity at a given location. Let  $(x, y)$  denote the coordinates of the location to which we want to assign an intensity value (think of it as a point of the grid described previously), and let  $v(x, y)$  denote that intensity value. For bilinear interpolation, the assigned value is obtained using the equation

$$v(x, y) = ax + by + cxy + d \quad (2.4-6)$$

Contrary to what the name suggests, note that bilinear interpolation is *not* linear because of the  $xy$  term.

where the four coefficients are determined from the four equations in four unknowns that can be written using the four nearest neighbors of point  $(x, y)$ . As you will see shortly, bilinear interpolation gives much better results than nearest neighbor interpolation, with a modest increase in computational burden.

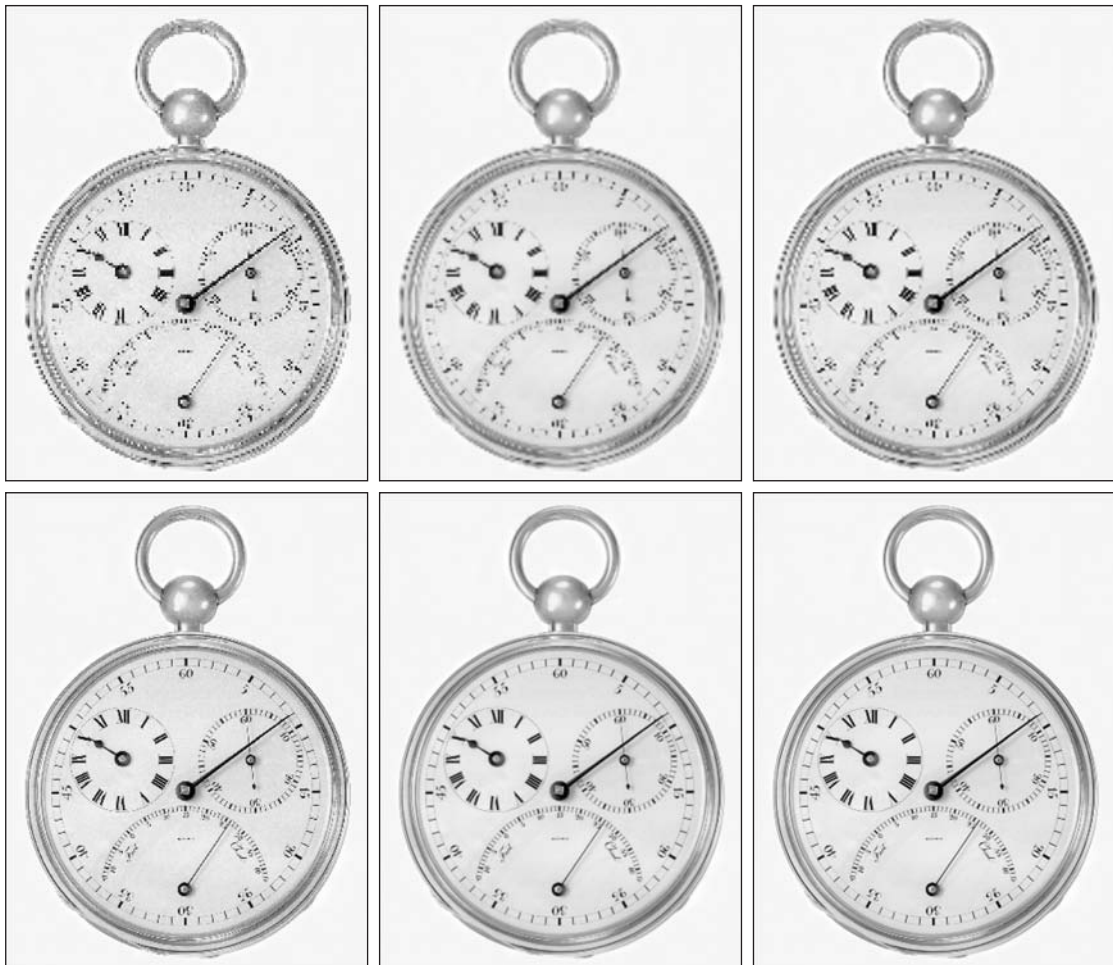
The next level of complexity is *bicubic interpolation*, which involves the sixteen nearest neighbors of a point. The intensity value assigned to point  $(x, y)$  is obtained using the equation

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (2.4-7)$$

where the sixteen coefficients are determined from the sixteen equations in sixteen unknowns that can be written using the sixteen nearest neighbors of point  $(x, y)$ . Observe that Eq. (2.4-7) reduces in form to Eq. (2.4-6) if the limits of both summations in the former equation are 0 to 1. Generally, bicubic interpolation does a better job of preserving fine detail than its bilinear counterpart. Bicubic interpolation is the standard used in commercial image editing programs, such as Adobe Photoshop and Corel Photopaint.

**EXAMPLE 2.4:** Comparison of interpolation approaches for image shrinking and zooming.

■ Figure 2.24(a) is the same image as Fig. 2.20(d), which was obtained by reducing the resolution of the 1250 dpi image in Fig. 2.20(a) to 72 dpi (the size shrank from the original size of  $3692 \times 2812$  to  $213 \times 162$  pixels) and then zooming the reduced image back to its original size. To generate Fig. 2.20(d) we used nearest neighbor interpolation both to shrink and zoom the image. As we commented before, the result in Fig. 2.24(a) is rather poor. Figures 2.24(b) and (c) are the results of repeating the same procedure but using, respectively, bilinear and bicubic interpolation for both shrinking and zooming. The result obtained by using bilinear interpolation is a significant improvement over nearest neighbor interpolation. The bicubic result is slightly sharper than the bilinear image. Figure 2.24(d) is the same as Fig. 2.20(c), which was obtained using nearest neighbor interpolation for both shrinking and zooming. We commented in discussing that figure that reducing the resolution to 150 dpi began showing degradation in the image. Figures 2.24(e) and (f) show the results of using



a	b	c
d	e	f

**FIGURE 2.24** (a) Image reduced to 72 dpi and zoomed back to its original size ( $3692 \times 2812$  pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

bilinear and bicubic interpolation, respectively, to shrink and zoom the image. In spite of a reduction in resolution from 1250 to 150, these last two images compare reasonably favorably with the original, showing once again the power of these two interpolation methods. As before, bicubic interpolation yielded slightly sharper results. ■

It is possible to use more neighbors in interpolation, and there are more complex techniques, such as using splines and wavelets, that in some instances can yield better results than the methods just discussed. While preserving fine detail is an exceptionally important consideration in image generation for 3-D graphics (Watt [1993], Shirley [2002]) and in medical image processing (Lehmann et al. [1999]), the extra computational burden seldom is justifiable for general-purpose digital image processing, where bilinear or bicubic interpolation typically are the methods of choice.

## 2.5 Some Basic Relationships between Pixels

In this section, we consider several important relationships between pixels in a digital image. As mentioned before, an image is denoted by  $f(x, y)$ . When referring in this section to a particular pixel, we use lowercase letters, such as  $p$  and  $q$ .

### 2.5.1 Neighbors of a Pixel

A pixel  $p$  at coordinates  $(x, y)$  has four *horizontal* and *vertical* neighbors whose coordinates are given by

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

This set of pixels, called the *4-neighbors* of  $p$ , is denoted by  $N_4(p)$ . Each pixel is a unit distance from  $(x, y)$ , and some of the neighbor locations of  $p$  lie outside the digital image if  $(x, y)$  is on the border of the image. We deal with this issue in Chapter 3.

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 by  $N_D(p)$ . These points, together with the 4-neighbors, are called the *8-neighbors* of  $p$ , denoted by  $N_8(p)$ . As before, some of the neighbor locations in  $N_D(p)$  and  $N_8(p)$  fall outside the image if  $(x, y)$  is on the border of the image.

### 2.5.2 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 gray-scale image, the idea is the same, but set  $V$  typically contains more elements. For example, in the adjacency of pixels with a range of possible intensity values 0 to 255, set  $V$  could be any subset of these 256 values. We consider three types of adjacency:

- (a) *4-adjacency*. Two pixels  $p$  and  $q$  with values from  $V$  are 4-adjacent if  $q$  is in the set  $N_4(p)$ .
- (b) *8-adjacency*. Two pixels  $p$  and  $q$  with values from  $V$  are 8-adjacent if  $q$  is in the set  $N_8(p)$ .
- (c) *m-adjacency* (mixed adjacency). Two pixels  $p$  and  $q$  with values from  $V$  are  $m$ -adjacent if
  - (i)  $q$  is in  $N_4(p)$ , or
  - (ii)  $q$  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 discuss sets in more detail in Section 2.6.4.

Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used. For example, consider the pixel arrangement shown in Fig. 2.25(a) for  $V = \{1\}$ . The three pixels at the top of Fig. 2.25(b) show multiple (ambiguous) 8-adjacency, as indicated by the dashed lines. This ambiguity is removed by using  $m$ -adjacency, as shown in Fig. 2.25(c).

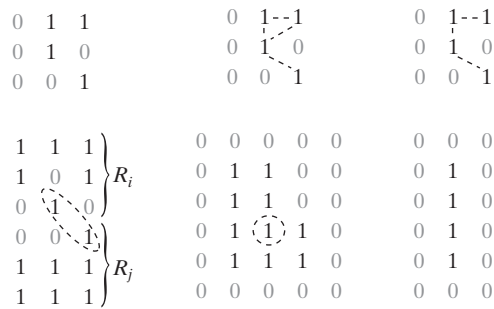
A (*digital*) *path* (or *curve*) from pixel  $p$  with coordinates  $(x, y)$  to pixel  $q$  with coordinates  $(s, t)$  is a sequence of distinct pixels with coordinates

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

where  $(x_0, y_0) = (x, y)$ ,  $(x_n, y_n) = (s, t)$ , and pixels  $(x_i, y_i)$  and  $(x_{i-1}, y_{i-1})$  are adjacent for  $1 \leq i \leq n$ . 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 shown in Fig. 2.25(b) between the top right and bottom right points are 8-paths, and the path in Fig. 2.25(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 connected component, then set  $S$  is called a *connected set*.

Let  $R$  be 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 1s) in Fig. 2.25(d) are adjacent only if 8-adjacency is used (according to the definition in the previous paragraph, a 4-path between the two regions does not exist, so their union is not a connected set).



a b c  
d e f

**FIGURE 2.25** (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c)  $m$ -adjacency. (d) Two regions (of 1s) that are adjacent if 8-adjacency is used. (e) The circled point is part of 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.

Suppose that an image contains  $K$  disjoint regions,  $R_k$ ,  $k = 1, 2, \dots, K$ , none of which touches the image border.<sup>†</sup> Let  $R_u$  denote the union of all the  $K$  regions, and let  $(R_u)^c$  denote its complement (recall that the *complement* of a set  $S$  is the set of points that are not in  $S$ ). We call all the points in  $R_u$  the *foreground*, and all the points in  $(R_u)^c$  the *background* of the image.

The *boundary* (also called the *border* or *contour*) of a region  $R$  is the set of points that are adjacent to points in the complement of  $R$ . Said another way, the border of a region is the set of pixels in the region that have at least one background neighbor. Here again, we must specify the connectivity being used to define adjacency. For example, the point circled in Fig. 2.25(e) is not a member of the border of the 1-valued region if 4-connectivity is used between the region and its background. As a rule, adjacency between points in a region and its background is defined in terms of 8-connectivity to handle situations like this.

The preceding definition sometimes is referred to as the *inner border* of the region to distinguish it from its *outer border*, which is the corresponding border in the background. This distinction is important in the development of border-following algorithms. Such algorithms usually are formulated to follow the outer boundary in order to guarantee that the result will form a closed path. For instance, the inner border of the 1-valued region in Fig. 2.25(f) is the region itself. This border does not satisfy the definition of a closed path given earlier. On the other hand, the outer border of the region does form a closed path around the region.

If  $R$  happens to be an entire image (which we recall is a rectangular set of pixels), then its boundary is defined as the set of pixels in the first and last rows and columns of the image. This extra definition is required because an image has no neighbors beyond its border. Normally, when we refer to a region, we are referring to a subset of an image, and any pixels in the boundary of the region that happen to coincide with the border of the image are included implicitly as part of the region boundary.

The concept of an *edge* is found frequently in discussions dealing with regions and boundaries. There is a key difference between these concepts, however. The boundary of a finite region forms a closed path and is thus a “global” concept. As discussed in detail in Chapter 10, edges are formed from pixels with derivative values that exceed a preset threshold. Thus, the idea of an edge is a “local” concept that is based on a measure of intensity-level discontinuity at a point. It is possible to link edge points into edge segments, and sometimes these segments are linked in such a way that they correspond to boundaries, but this is not always the case. The one exception in which edges and boundaries correspond is in binary images. Depending on the type of connectivity and edge operators used (we discuss these in Chapter 10), the edge extracted from a binary region will be the same as the region boundary.

---

<sup>†</sup>We make this assumption to avoid having to deal with special cases. This is done without loss of generality because if one or more regions touch the border of an image, we can simply pad the image with a 1-pixel-wide border of background values.

This is intuitive. Conceptually, until we arrive at Chapter 10, it is helpful to think of edges as intensity discontinuities and boundaries as closed paths.

### 2.5.3 Distance Measures

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

- (a)  $D(p, q) \geq 0$  ( $D(p, q) = 0$  iff  $p = q$ ),
- (b)  $D(p, q) = D(q, p)$ , and
- (c)  $D(p, z) \leq D(p, q) + D(q, z)$ .

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

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{\frac{1}{2}} \quad (2.5-1)$$

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

$$D_4(p, q) = |x - s| + |y - t| \quad (2.5-2)$$

In this case, the pixels having a  $D_4$  distance from  $(x, y)$  less than or equal to some value  $r$  form a diamond centered at  $(x, y)$ . For example, the pixels with  $D_4$  distance  $\leq 2$  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

$$D_8(p, q) = \max(|x - s|, |y - t|) \quad (2.5-3)$$

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

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

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

Note that the  $D_4$  and  $D_8$  distances between  $p$  and  $q$  are independent of any paths that might exist between the points because these distances involve only the coordinates of the points. If we elect to consider  $m$ -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 value 1 and that  $p_1$  and  $p_3$  can have a value of 0 or 1:

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

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) 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  $pp_1p_2p_3p_4$ .

## 2.6 An Introduction to the Mathematical Tools Used in Digital Image Processing



Before proceeding, you may find it helpful to download and study the review material available in the Tutorials section of the book Web site. The review covers introductory material on matrices and vectors, linear systems, set theory, and probability.

This section has two principal objectives: (1) to introduce you to the various mathematical tools we use throughout the book; and (2) to help you begin developing a “feel” for how these tools are used by applying them to a variety of basic image-processing tasks, some of which will be used numerous times in subsequent discussions. We expand the scope of the tools and their application as necessary in the following chapters.

### 2.6.1 Array versus Matrix Operations

An *array* operation involving one or more images is carried out on a *pixel-by-pixel* basis. We mentioned earlier in this chapter that images can be viewed equivalently as matrices. In fact, there are many situations in which operations between images are carried out using matrix theory (see Section 2.6.6). It is for this reason that a clear distinction must be made between array and matrix operations. For example, consider the following  $2 \times 2$  images:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

The *array product* of these two images is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$



On the other hand, the *matrix product* is given by

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

We assume array operations throughout the book, unless stated otherwise. For example, when we refer to raising an image to a power, we mean that each individual pixel is raised to that power; when we refer to dividing an image by another, we mean that the division is between corresponding pixel pairs, and so on.

### 2.6.2 Linear versus Nonlinear Operations

One of the most important classifications of an image-processing method is whether it is linear or nonlinear. Consider a general operator,  $H$ , that produces an output image,  $g(x, y)$ , for a given input image,  $f(x, y)$ :

$$H[f(x, y)] = g(x, y) \quad (2.6-1)$$

$H$  is said to be a *linear operator* if

$$\begin{aligned} H[a_i f_i(x, y) + a_j f_j(x, y)] &= a_i H[f_i(x, y)] + a_j H[f_j(x, y)] \\ &= a_i g_i(x, y) + a_j g_j(x, y) \end{aligned} \quad (2.6-2)$$

where  $a_i$ ,  $a_j$ ,  $f_i(x, y)$ , and  $f_j(x, y)$  are arbitrary constants and images (of the same size), respectively. Equation (2.6-2) indicates that the output of a linear operation due to the sum of two inputs is the same as performing the operation on the inputs individually and then summing the results. In addition, the output of a linear operation to a constant times an input is the same as the output of the operation due to the original input multiplied by that constant. The first property is called the property of *additivity* and the second is called the property of *homogeneity*.

As a simple example, suppose that  $H$  is the sum operator,  $\Sigma$ ; that is, the function of this operator is simply to sum its inputs. To test for linearity, we start with the left side of Eq. (2.6-2) and attempt to prove that it is equal to the right side:

$$\begin{aligned} \Sigma[a_i f_i(x, y) + a_j f_j(x, y)] &= \Sigma a_i f_i(x, y) + \Sigma a_j f_j(x, y) \\ &= a_i \Sigma f_i(x, y) + a_j \Sigma f_j(x, y) \\ &= a_i g_i(x, y) + a_j g_j(x, y) \end{aligned}$$

These are array summations, not the sums of all the elements of the images. As such, the sum of a single image is the image itself.

where the first step follows from the fact that summation is distributive. So, an expansion of the left side is equal to the right side of Eq. (2.6-2), and we conclude that the sum operator is linear.

On the other hand, consider the max operation, whose function is to find the maximum value of the pixels in an image. For our purposes here, the simplest way to prove that this operator is nonlinear, is to find an example that fails the test in Eq. (2.6-2). Consider the following two images

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

and suppose that we let  $a_1 = 1$  and  $a_2 = -1$ . To test for linearity, we again start with the left side of Eq. (2.6-2):

$$\begin{aligned} \max \left\{ (1) \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1) \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} &= \max \left\{ \begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix} \right\} \\ &= -2 \end{aligned}$$

Working next with the right side, we obtain

$$\begin{aligned} (1) \max \left\{ \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \right\} + (-1) \max \left\{ \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} &= 3 + (-1)7 \\ &= -4 \end{aligned}$$

The left and right sides of Eq. (2.6-2) are not equal in this case, so we have proved that in general the max operator is nonlinear.

As you will see in the next three chapters, especially in Chapters 4 and 5, linear operations are exceptionally important because they are based on a large body of theoretical and practical results that are applicable to image processing. Nonlinear systems are not nearly as well understood, so their scope of application is more limited. However, you will encounter in the following chapters several nonlinear image processing operations whose performance far exceeds what is achievable by their linear counterparts.

### 2.6.3 Arithmetic Operations

Arithmetic operations between images are array operations which, as discussed in Section 2.6.1, means that arithmetic operations are carried out between corresponding pixel pairs. The four arithmetic operations are denoted as

$$\begin{aligned} s(x, y) &= f(x, y) + g(x, y) \\ d(x, y) &= f(x, y) - g(x, y) \\ p(x, y) &= f(x, y) \times g(x, y) \\ v(x, y) &= f(x, y) \div g(x, y) \end{aligned} \tag{2.6-3}$$

It is understood that the operations are performed between corresponding pixel pairs in  $f$  and  $g$  for  $x = 0, 1, 2, \dots, M - 1$  and  $y = 0, 1, 2, \dots, N - 1$

## 2.6 ■ An Introduction to the Mathematical Tools Used in Digital Image Processing 75

where, as usual,  $M$  and  $N$  are the row and column sizes of the images. Clearly,  $s$ ,  $d$ ,  $p$ , and  $v$  are images of size  $M \times N$  also. Note that image arithmetic in the manner just defined involves images of the same size. The following examples are indicative of the important role played by arithmetic operations in digital image processing.

■ Let  $g(x, y)$  denote a corrupted image formed by the addition of noise,  $\eta(x, y)$ , to a noiseless image  $f(x, y)$ ; that is,

$$g(x, y) = f(x, y) + \eta(x, y) \quad (2.6-4)$$

**EXAMPLE 2.5:**  
Addition  
(averaging) of  
noisy images for  
noise reduction.

where the assumption is that at every pair of coordinates  $(x, y)$  the noise is uncorrelated<sup>†</sup> and has zero average value. The objective of the following procedure is to reduce the noise content by adding a set of noisy images,  $\{g_i(x, y)\}$ . This is a technique used frequently for image enhancement.

If the noise satisfies the constraints just stated, it can be shown (Problem 2.20) that if an image  $\bar{g}(x, y)$  is formed by averaging  $K$  different noisy images,

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y) \quad (2.6-5)$$

then it follows that

$$E\{\bar{g}(x, y)\} = f(x, y) \quad (2.6-6)$$

and

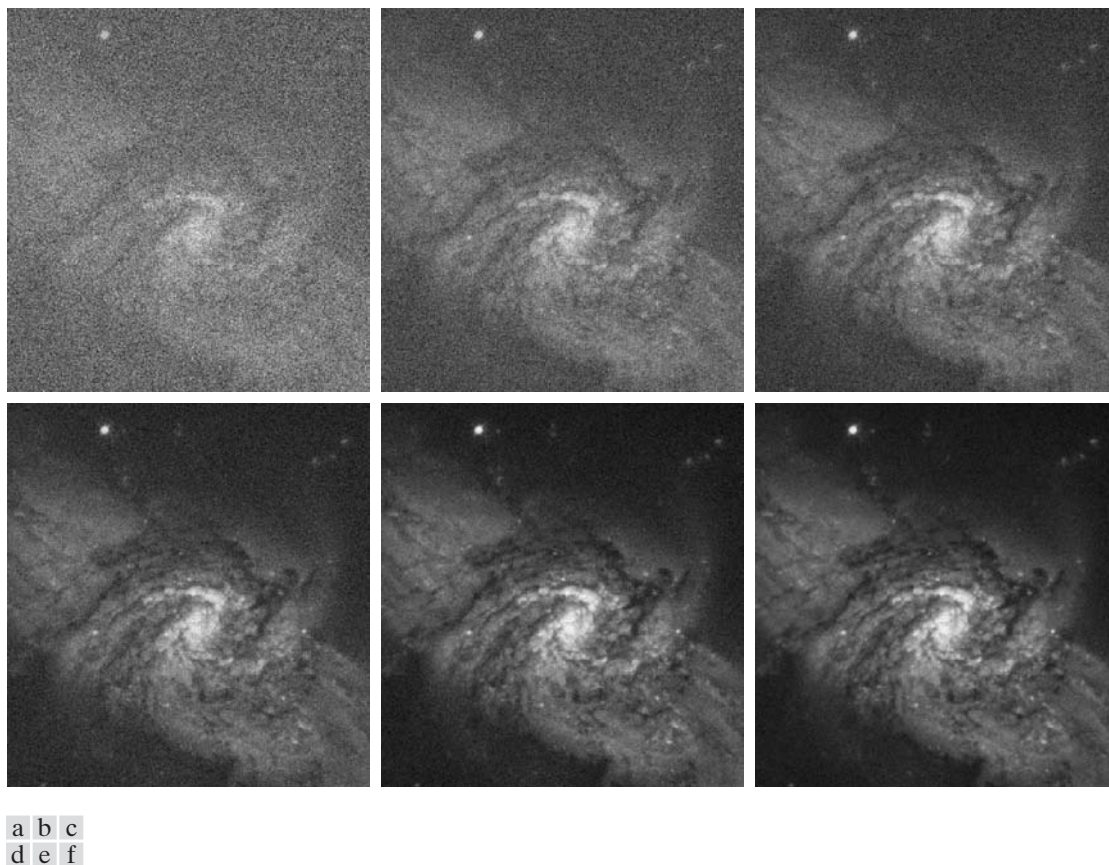
$$\sigma_{\bar{g}(x,y)}^2 = \frac{1}{K} \sigma_{\eta(x,y)}^2 \quad (2.6-7)$$

where  $E\{\bar{g}(x, y)\}$  is the expected value of  $\bar{g}$ , and  $\sigma_{\bar{g}(x,y)}^2$  and  $\sigma_{\eta(x,y)}^2$  are the variances of  $\bar{g}$  and  $\eta$ , respectively, all at coordinates  $(x, y)$ . The standard deviation (square root of the variance) at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)} \quad (2.6-8)$$

As  $K$  increases, Eqs. (2.6-7) and (2.6-8) indicate that the variability (as measured by the variance or the standard deviation) of the pixel values at each location  $(x, y)$  decreases. Because  $E\{\bar{g}(x, y)\} = f(x, y)$ , this means that  $\bar{g}(x, y)$  approaches  $f(x, y)$  as the number of noisy images used in the averaging process increases. In practice, the images  $g_i(x, y)$  must be *registered* (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

<sup>†</sup>Recall that the variance of a random variable  $z$  with mean  $m$  is defined as  $E[(z - m)^2]$ , where  $E\{\cdot\}$  is the expected value of the argument. The covariance of two random variables  $z_i$  and  $z_j$  is defined as  $E[(z_i - m_i)(z_j - m_j)]$ . If the variables are *uncorrelated*, their covariance is 0.



**FIGURE 2.26** (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

The images shown in this example are from a galaxy pair called NGC 3314, taken by NASA's Hubble Space Telescope. NGC 3314 lies about 140 million light-years from Earth, in the direction of the southern-hemisphere constellation Hydra. The bright stars forming a pinwheel shape near the center of the front galaxy were formed from interstellar gas and dust.

An important application of image averaging is in the field of astronomy, where imaging under very low light levels frequently causes sensor noise to render single images virtually useless for analysis. Figure 2.26(a) shows an 8-bit image in which corruption was simulated by adding to it Gaussian noise with zero mean and a standard deviation of 64 intensity levels. This image, typical of noisy images taken under low light conditions, is useless for all practical purposes. Figures 2.26(b) through (f) show the results of averaging 5, 10, 20, 50, and 100 images, respectively. We see that the result in Fig. 2.26(e), obtained with  $K = 50$ , is reasonably clean. The image Fig. 2.26(f), resulting from averaging 100 noisy images, is only a slight improvement over the image in Fig. 2.26(e).

Addition is a discrete version of continuous integration. In astronomical observations, a process equivalent to the method just described is to use the integrating capabilities of CCD (see Section 2.3.3) or similar sensors for noise reduction by observing the same scene over long periods of time. Cooling also is used to reduce sensor noise. The net effect, however, is analogous to averaging a set of noisy digital images. ■

■ A frequent application of image subtraction is in the enhancement of *differences* between images. For example, the image in Fig. 2.27(b) was obtained by setting to zero the least-significant bit of every pixel in Fig. 2.27(a). Visually, these images are indistinguishable. However, as Fig. 2.27(c) shows, subtracting one image from the other clearly shows their differences. Black (0) values in this difference image indicate locations where there is no difference between the images in Figs. 2.27(a) and (b).

As another illustration, we discuss briefly an area of medical imaging called *mask mode radiography*, a commercially successful and highly beneficial use of image subtraction. Consider image differences of the form

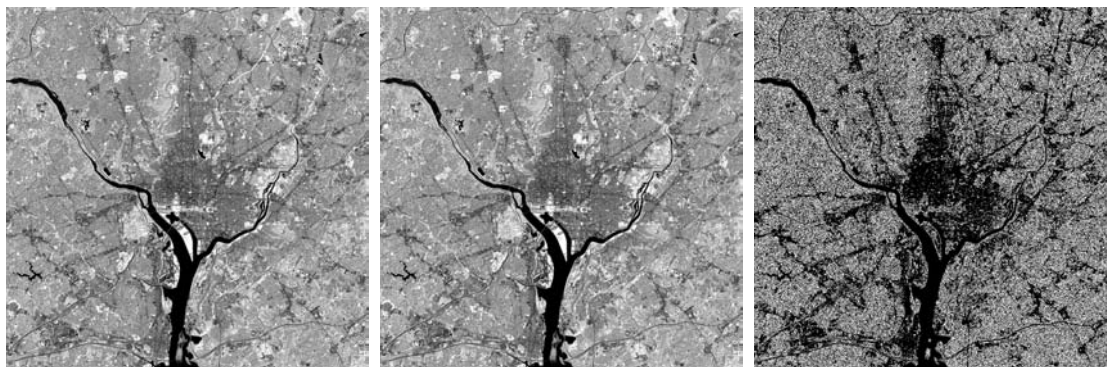
$$g(x, y) = f(x, y) - h(x, y) \quad (2.6-9)$$

In this case  $h(x, y)$ , the *mask*, is an X-ray image of a region of a patient's body captured by an intensified TV camera (instead of traditional X-ray film) located opposite an X-ray source. The procedure consists of injecting an X-ray contrast medium into the patient's bloodstream, taking a series of images called *live images* [samples of which are denoted as  $f(x, y)$ ] of the same anatomical region as  $h(x, y)$ , and subtracting the mask from the series of incoming live images after injection of the contrast medium. The net effect of subtracting the mask from each sample live image is that the areas that are different between  $f(x, y)$  and  $h(x, y)$  appear in the output image,  $g(x, y)$ , as enhanced detail. Because images can be captured at TV rates, this procedure in essence gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.

Figure 2.28(a) shows a mask X-ray image of the top of a patient's head prior to injection of an iodine medium into the bloodstream, and Fig. 2.28(b) is a sample of a live image taken after the medium was injected. Figure 2.28(c) is

**EXAMPLE 2.6:**  
Image subtraction for enhancing differences.

Change detection via image subtraction is used also in image segmentation, which is the topic of Chapter 10.



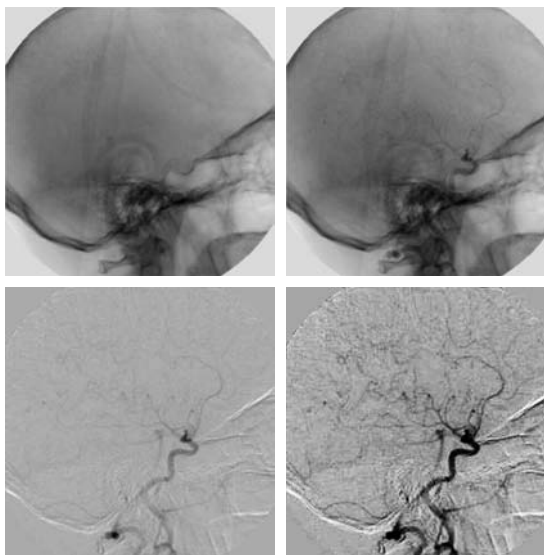
a b c

**FIGURE 2.27** (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range  $[0, 255]$  for clarity.

a	b
c	d

**FIGURE 2.28**

Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)



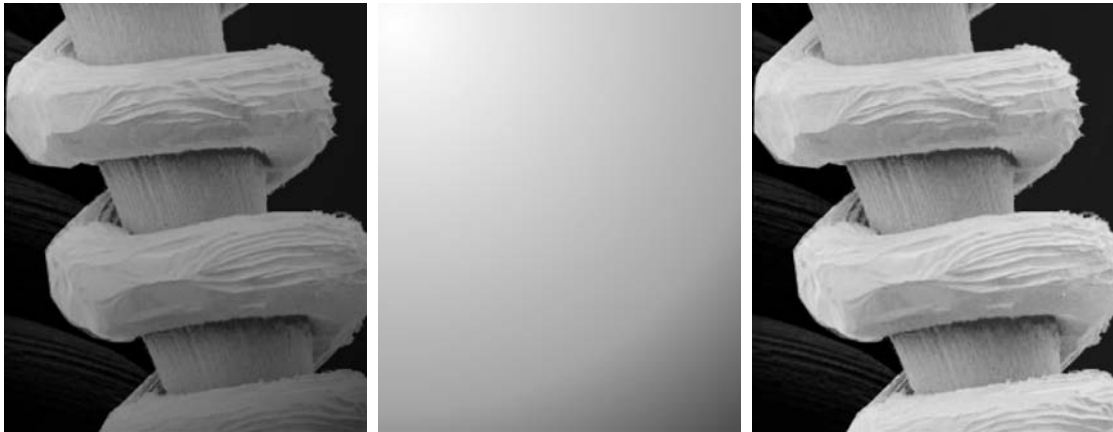
the difference between (a) and (b). Some fine blood vessel structures are visible in this image. The difference is clear in Fig. 2.28(d), which was obtained by enhancing the contrast in (c) (we discuss contrast enhancement in the next chapter). Figure 2.28(d) is a clear “map” of how the medium is propagating through the blood vessels in the subject’s brain. ■

**EXAMPLE 2.7:** Using image multiplication and division for shading correction.

■ An important application of image multiplication (and division) is *shading correction*. Suppose that an imaging sensor produces images that can be modeled as the product of a “perfect image,” denoted by  $f(x, y)$ , times a shading function,  $h(x, y)$ ; that is,  $g(x, y) = f(x, y)h(x, y)$ . If  $h(x, y)$  is known, we can obtain  $f(x, y)$  by multiplying the sensed image by the inverse of  $h(x, y)$  (i.e., dividing  $g$  by  $h$ ). If  $h(x, y)$  is not known, but access to the imaging system is possible, we can obtain an approximation to the shading function by imaging a target of constant intensity. When the sensor is not available, we often can estimate the shading pattern directly from the image, as we discuss in Section 9.6. Figure 2.29 shows an example of shading correction.

Another common use of image multiplication is in *masking*, also called *region of interest (ROI)*, operations. The process, illustrated in Fig. 2.30, consists simply of multiplying a given image by a mask image that has 1s in the ROI and 0s elsewhere. There can be more than one ROI in the mask image, and the shape of the ROI can be arbitrary, although rectangular shapes are used frequently for ease of implementation. ■

A few comments about implementing image arithmetic operations are in order before we leave this section. In practice, most images are displayed using 8 bits (even 24-bit color images consist of three separate 8-bit channels). Thus, we expect image values to be in the range from 0 to 255. When images

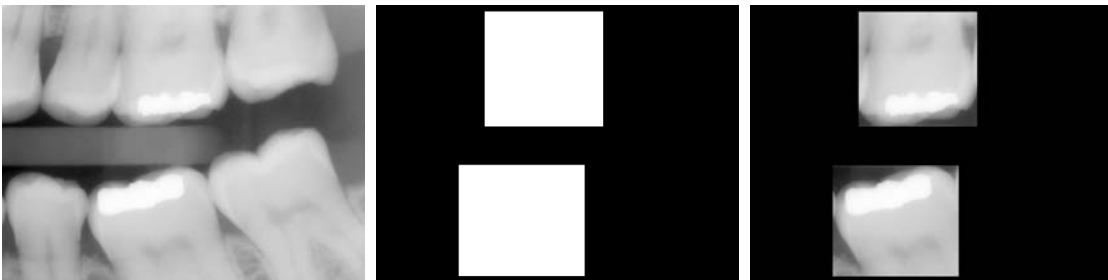


a b c

**FIGURE 2.29** Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

are saved in a standard format, such as TIFF or JPEG, conversion to this range is automatic. However, the approach used for the conversion depends on the system used. For example, the values in the difference of two 8-bit images can range from a minimum of  $-255$  to a maximum of  $255$ , and the values of a sum image can range from  $0$  to  $510$ . Many software packages simply set all negative values to  $0$  and set to  $255$  all values that exceed this limit when converting images to 8 bits. Given an image  $f$ , an approach that guarantees that the full range of an arithmetic operation between images is “captured” into a fixed number of bits is as follows. First, we perform the operation

$$f_m = f - \min(f) \quad (2.6-10)$$



a b c

**FIGURE 2.30** (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

which creates an image whose minimum value is 0. Then, we perform the operation

$$f_s = K \lceil f_m / \max(f_m) \rceil \quad (2.6-11)$$

which creates a scaled image,  $f_s$ , whose values are in the range  $[0, K]$ . When working with 8-bit images, setting  $K = 255$  gives us a scaled image whose intensities span the full 8-bit scale from 0 to 255. Similar comments apply to 16-bit images or higher. This approach can be used for all arithmetic operations. When performing division, we have the extra requirement that a small number should be added to the pixels of the divisor image to avoid division by 0.

### 2.6.4 Set and Logical Operations

In this section, we introduce briefly some important set and logical operations. We also introduce the concept of a fuzzy set.

#### Basic set operations

Let  $A$  be a set composed of *ordered pairs* of real numbers. If  $a = (a_1, a_2)$  is an *element* of  $A$ , then we write

$$a \in A \quad (2.6-12)$$

Similarly, if  $a$  is not an element of  $A$ , we write

$$a \notin A \quad (2.6-13)$$

The set with no elements is called the *null* or *empty set* and is denoted by the symbol  $\emptyset$ .

A set is specified by the contents of two braces:  $\{ \cdot \}$ . For example, when we write an expression of the form  $C = \{w | w = -d, d \in D\}$ , we mean that set  $C$  is the set of elements,  $w$ , such that  $w$  is formed by multiplying each of the elements of set  $D$  by  $-1$ . One way in which sets are used in image processing is to let the elements of sets be the *coordinates* of pixels (ordered pairs of integers) representing regions (objects) in an image.

If every element of a set  $A$  is also an element of a set  $B$ , then  $A$  is said to be a *subset* of  $B$ , denoted as

$$A \subseteq B \quad (2.6-14)$$

The *union* of two sets  $A$  and  $B$ , denoted by

$$C = A \cup B \quad (2.6-15)$$

is the set of elements belonging to either  $A$ ,  $B$ , or both. Similarly, the *intersection* of two sets  $A$  and  $B$ , denoted by

$$D = A \cap B \quad (2.6-16)$$

is the set of elements belonging to both  $A$  and  $B$ . Two sets  $A$  and  $B$  are said to be *disjoint* or *mutually exclusive* if they have no common elements, in which case,

$$A \cap B = \emptyset \quad (2.6-17)$$



## 2.6 ■ An Introduction to the Mathematical Tools Used in Digital Image Processing 81

The *set universe*,  $U$ , is the set of all elements in a given application. By definition, all set elements in a given application are members of the universe defined for that application. For example, if you are working with the set of real numbers, then the set universe is the real line, which contains all the real numbers. In image processing, we typically define the universe to be the rectangle containing all the pixels in an image.

The *complement* of a set  $A$  is the set of elements that are not in  $A$ :

$$A^c = \{w | w \notin A\} \quad (2.6-18)$$

The *difference* of two sets  $A$  and  $B$ , denoted  $A - B$ , is defined as

$$A - B = \{w | w \in A, w \notin B\} = A \cap B^c \quad (2.6-19)$$

We see that this is the set of elements that belong to  $A$ , but not to  $B$ . We could, for example, define  $A^c$  in terms of  $U$  and the set difference operation:  $A^c = U - A$ .

Figure 2.31 illustrates the preceding concepts, where the universe is the set of coordinates contained within the rectangle shown, and sets  $A$  and  $B$  are the sets of coordinates contained within the boundaries shown. The result of the set operation indicated in each figure is shown in gray.<sup>†</sup>

In the preceding discussion, set membership is based on position (coordinates). An implicit assumption when working with images is that the intensity of all pixels in the sets is the same, as we have not defined set operations involving intensity values (e.g., we have not specified what the intensities in the intersection of two sets is). The only way that the operations illustrated in Fig. 2.31 can make sense is if the images containing the sets are binary, in which case we can talk about set membership based on coordinates, the assumption being that all member of the sets have the same intensity. We discuss this in more detail in the following subsection.

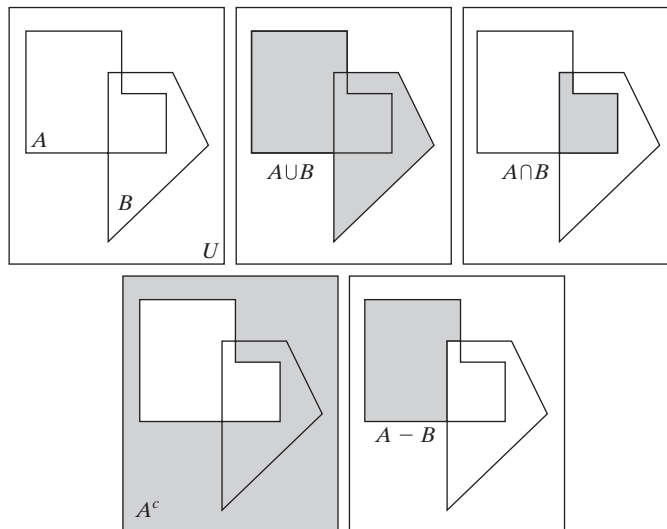
When dealing with gray-scale images, the preceding concepts are not applicable, because we have to specify the intensities of all the pixels resulting from a set operation. In fact, as you will see in Sections 3.8 and 9.6, the union and intersection operations for gray-scale values usually are defined as the max and min of corresponding pixel pairs, respectively, while the complement is defined as the pairwise differences between a constant and the intensity of every pixel in an image. The fact that we deal with corresponding pixel pairs tells us that gray-scale set operations are array operations, as defined in Section 2.6.1. The following example is a brief illustration of set operations involving gray-scale images. We discuss these concepts further in the two sections mentioned above.

<sup>†</sup>The operations in Eqs. (2.6-12)–(2.6-19) are the basis for the algebra of sets, which starts with properties such as the *commutative laws*:  $A \cup B = B \cup A$  and  $A \cap B = B \cap A$ , and from these develops a broad theory based on set operations. A treatment of the algebra of sets is beyond the scope of the present discussion, but you should be aware of its existence.

a b c  
d e

**FIGURE 2.31**

(a) Two sets of coordinates,  $A$  and  $B$ , in 2-D space. (b) The union of  $A$  and  $B$ . (c) The intersection of  $A$  and  $B$ . (d) The complement of  $A$ . (e) The difference between  $A$  and  $B$ . In (b)–(e) the shaded areas represent the members of the set operation indicated.

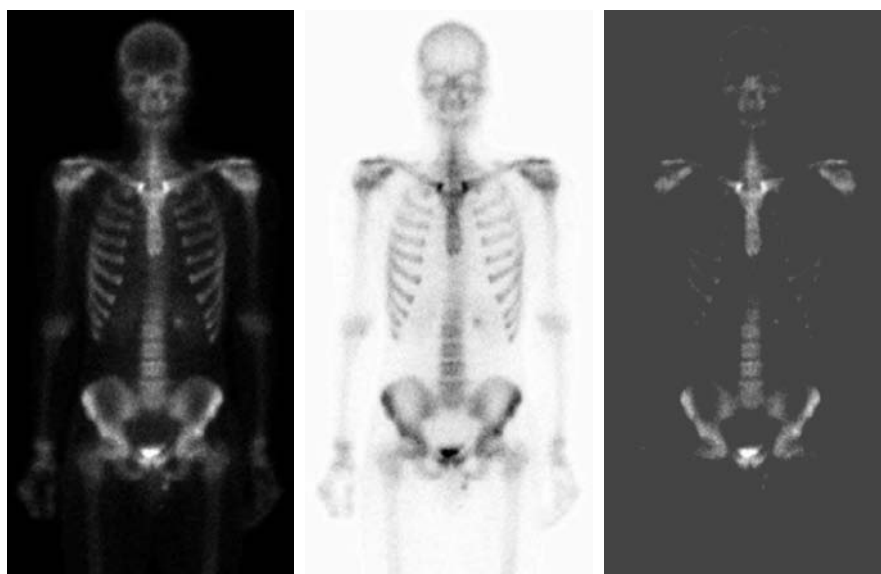


**EXAMPLE 2.8:** Set operations involving image intensities.

■ Let the elements of a gray-scale image be represented by a set  $A$  whose elements are triplets of the form  $(x, y, z)$ , where  $x$  and  $y$  are spatial coordinates and  $z$  denotes intensity, as mentioned in Section 2.4.2. We can define the *complement* of  $A$  as the set  $A^c = \{(x, y, K - z) | (x, y, z) \in A\}$ , which simply denotes the set of pixels of  $A$  whose intensities have been subtracted from a constant  $K$ . This constant is equal to  $2^k - 1$ , where  $k$  is the number of intensity bits used to represent  $z$ . Let  $A$  denote the 8-bit gray-scale image in Fig. 2.32(a), and suppose that we want to form the negative of  $A$  using set

a b c

**FIGURE 2.32** Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)



## 2.6 ■ An Introduction to the Mathematical Tools Used in Digital Image Processing 83

operations. We simply form the set  $A_n = A^c = \{(x, y, 255 - z) | (x, y, z) \in A\}$ . Note that the coordinates are carried over, so  $A_n$  is an image of the same size as  $A$ . Figure 2.32(b) shows the result.

The union of two gray-scale sets  $A$  and  $B$  may be defined as the set

$$A \cup B = \left\{ \max_z(a, b) | a \in A, b \in B \right\}$$

That is, the union of two gray-scale sets (images) is an array formed from the maximum intensity between pairs of spatially corresponding elements. Again, note that coordinates carry over, so the union of  $A$  and  $B$  is an image of the same size as these two images. As an illustration, suppose that  $A$  again represents the image in Fig. 2.32(a), and let  $B$  denote a rectangular array of the same size as  $A$ , but in which all values of  $z$  are equal to 3 times the mean intensity,  $m$ , of the elements of  $A$ . Figure 2.32(c) shows the result of performing the set union, in which all values exceeding  $3m$  appear as values from  $A$  and all other pixels have value  $3m$ , which is a mid-gray value. ■

### Logical operations

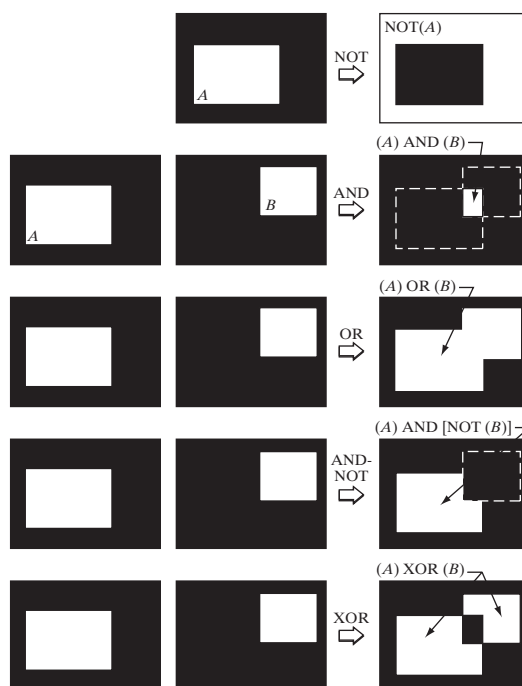
When dealing with binary images, we can think of *foreground* (1-valued) and *background* (0-valued) sets of pixels. Then, if we define regions (objects) as being composed of foreground pixels, the set operations illustrated in Fig. 2.31 become operations between the coordinates of objects in a binary image. When dealing with binary images, it is common practice to refer to union, intersection, and complement as the OR, AND, and NOT *logical* operations, where “logical” arises from logic theory in which 1 and 0 denote true and false, respectively.

Consider two regions (sets)  $A$  and  $B$  composed of foreground pixels. The OR of these two sets is the set of elements (coordinates) belonging either to  $A$  or  $B$  or to both. The AND operation is the set of elements that are common to  $A$  and  $B$ . The NOT operation of a set  $A$  is the set of elements not in  $A$ . Because we are dealing with images, if  $A$  is a given set of foreground pixels, NOT( $A$ ) is the set of all pixels in the image that are not in  $A$ , these pixels being background pixels and possibly other foreground pixels. We can think of this operation as turning all elements in  $A$  to 0 (black) and all the elements not in  $A$  to 1 (white). Figure 2.33 illustrates these operations. Note in the fourth row that the result of the operation shown is the set of foreground pixels that belong to  $A$  but not to  $B$ , which is the definition of set difference in Eq. (2.6-19). The last row in the figure is the XOR (exclusive OR) operation, which is the set of foreground pixels belonging to  $A$  or  $B$ , but not both. Observe that the preceding operations are between regions, which clearly can be irregular and of different sizes. This is as opposed to the gray-scale operations discussed earlier, which are array operations and thus require sets whose spatial dimensions are the same. That is, gray-scale set operations involve complete images, as opposed to regions of images.

We need to be concerned in theory only with the capability to implement the AND, OR, and NOT logic operators because these three operators are *functionally*

**FIGURE 2.33**

Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.



*complete*. In other words, any other logic operator can be implemented by using only these three basic functions, as in the fourth row of Fig. 2.33, where we implemented the set difference operation using AND and NOT. Logic operations are used extensively in image morphology, the topic of Chapter 9.

### Fuzzy sets

The preceding set and logical results are *crisp* concepts, in the sense that elements either are or are not members of a set. This presents a serious limitation in some applications. Consider a simple example. Suppose that we wish to categorize all people in the world as being young or not young. Using crisp sets, let  $U$  denote the set of all people and let  $A$  be a subset of  $U$ , which we call the *set of young people*. In order to form set  $A$ , we need a *membership function* that assigns a value of 1 or 0 to every element (person) in  $U$ . If the value assigned to an element of  $U$  is 1, then that element is a member of  $A$ ; otherwise it is not. Because we are dealing with a bi-valued logic, the membership function simply defines a threshold at or below which a person is considered young, and above which a person is considered not young. Suppose that we define as young any person of age 20 or younger. We see an immediate difficulty. A person whose age is 20 years and 1 sec would not be a member of the set of young people. This limitation arises regardless of the age threshold we use to classify a person as being young. What we need is more flexibility in what we mean by “young,” that is, we need a *gradual* transition from young to not young. The theory of *fuzzy sets* implements this concept by utilizing membership functions

that are gradual between the limit values of 1 (definitely young) to 0 (definitely not young). Using fuzzy sets, we can make a statement such as a person being 50% young (in the middle of the transition between young and not young). In other words, age is an imprecise concept, and fuzzy logic provides the tools to deal with such concepts. We explore fuzzy sets in detail in Section 3.8.

### 2.6.5 Spatial Operations

Spatial operations are performed directly on the pixels of a given image. We classify spatial operations into three broad categories: (1) single-pixel operations, (2) neighborhood operations, and (3) geometric spatial transformations.

#### Single-pixel operations

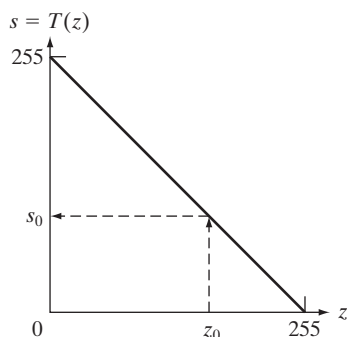
The simplest operation we perform on a digital image is to alter the values of its individual pixels based on their intensity. This type of process may be expressed as a transformation function,  $T$ , of the form:

$$s = T(z) \quad (2.6-20)$$

where  $z$  is the intensity of a pixel in the original image and  $s$  is the (mapped) intensity of the corresponding pixel in the processed image. For example, Fig. 2.34 shows the transformation used to obtain the negative of an 8-bit image, such as the image in Fig. 2.32(b), which we obtained using set operations. We discuss in Chapter 3 a number of techniques for specifying intensity transformation functions.

#### Neighborhood operations

Let  $S_{xy}$  denote the set of coordinates of a neighborhood centered on an arbitrary point  $(x, y)$  in an image,  $f$ . Neighborhood processing generates a corresponding pixel at the same coordinates in an output (processed) image,  $g$ , such that the value of that pixel is determined by a specified operation involving the pixels in the input image with coordinates in  $S_{xy}$ . For example, suppose that the specified operation is to compute the average value of the pixels in a rectangular neighborhood of size  $m \times n$  centered on  $(x, y)$ . The locations of pixels

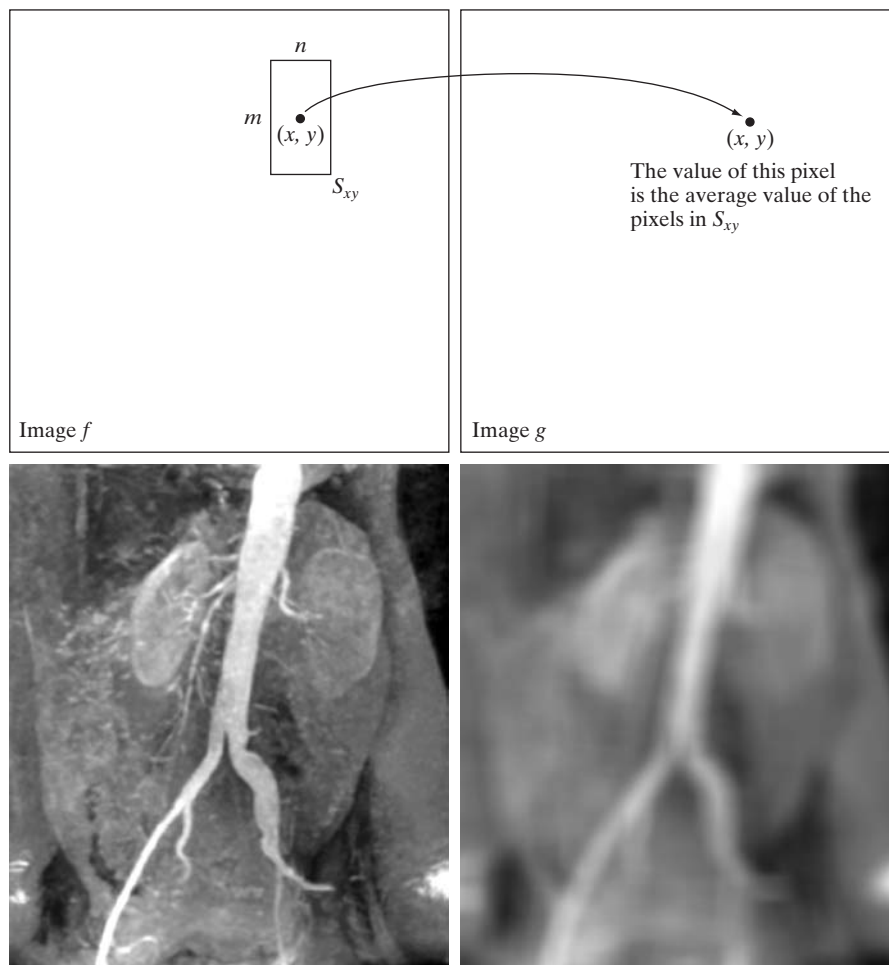


**FIGURE 2.34** Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value  $z_0$  into its corresponding output value  $s_0$ .

a b  
c d

**FIGURE 2.35**

Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with  $m = n = 41$ . The images are of size  $790 \times 686$  pixels.



in this region constitute the set  $S_{xy}$ . Figures 2.35(a) and (b) illustrate the process. We can express this operation in equation form as

$$g(x, y) = \frac{1}{mn} \sum_{(r,c) \in S_{xy}} f(r, c) \quad (2.6-21)$$

where  $r$  and  $c$  are the row and column coordinates of the pixels whose coordinates are members of the set  $S_{xy}$ . Image  $g$  is created by varying the coordinates  $(x, y)$  so that the center of the neighborhood moves from pixel to pixel in image  $f$ , and repeating the neighborhood operation at each new location. For instance, the image in Fig. 2.35(d) was created in this manner using a neighborhood of size  $41 \times 41$ . The net effect is to perform local blurring in the original image. This type of process is used, for example, to eliminate small details and thus render “blobs” corresponding to the largest regions of an image. We

discuss neighborhood processing in Chapters 3 and 5, and in several other places in the book.

### Geometric spatial transformations and image registration

Geometric transformations modify the spatial relationship between pixels in an image. These transformations often are called *rubber-sheet* transformations because they may be viewed as analogous to “printing” an image on a sheet of rubber and then stretching the sheet according to a predefined set of rules. In terms of digital image processing, a geometric transformation consists of two basic operations: (1) a spatial transformation of coordinates and (2) intensity interpolation that assigns intensity values to the spatially transformed pixels.

The transformation of coordinates may be expressed as

$$(x, y) = T\{(v, w)\} \quad (2.6-22)$$

where  $(v, w)$  are pixel coordinates in the original image and  $(x, y)$  are the corresponding pixel coordinates in the transformed image. For example, the transformation  $(x, y) = T\{(v, w)\} = (v/2, w/2)$  shrinks the original image to half its size in both spatial directions. One of the most commonly used spatial coordinate transformations is the *affine transform* (Wolberg [1990]), which has the general form

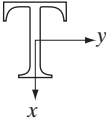
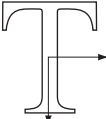
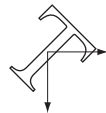
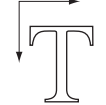
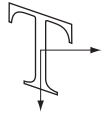
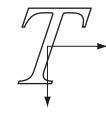
$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} \mathbf{T} = \begin{bmatrix} v & w & 1 \end{bmatrix} \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} \quad (2.6-23)$$

This transformation can scale, rotate, translate, or shear a set of coordinate points, depending on the value chosen for the elements of matrix  $\mathbf{T}$ . Table 2.2 illustrates the matrix values used to implement these transformations. The real power of the matrix representation in Eq. (2.6-23) is that it provides the framework for concatenating together a sequence of operations. For example, if we want to resize an image, rotate it, and move the result to some location, we simply form a  $3 \times 3$  matrix equal to the product of the scaling, rotation, and translation matrices from Table 2.2.

The preceding transformations relocate pixels on an image to new locations. To complete the process, we have to assign intensity values to those locations. This task is accomplished using intensity interpolation. We already discussed this topic in Section 2.4.4. We began that section with an example of zooming an image and discussed the issue of intensity assignment to new pixel locations. Zooming is simply scaling, as detailed in the second row of Table 2.2, and an analysis similar to the one we developed for zooming is applicable to the problem of assigning intensity values to the relocated pixels resulting from the other transformations in Table 2.2. As in Section 2.4.4, we consider nearest neighbor, bilinear, and bicubic interpolation techniques when working with these transformations.

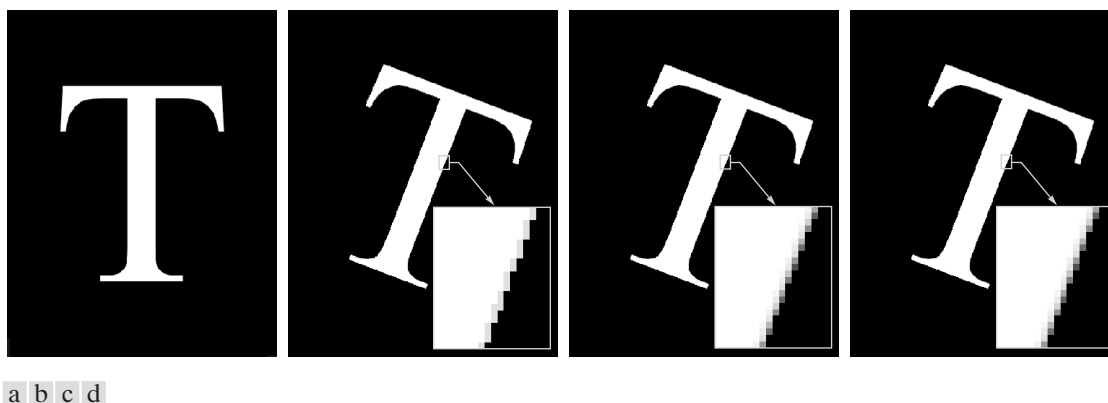
In practice, we can use Eq. (2.6-23) in two basic ways. The first, called a *forward mapping*, consists of scanning the pixels of the input image and, at

**TABLE 2.2**  
Affine transformations based on Eq. (2.6-23).

Transformation Name	Affine Matrix, $T$	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v \\ y &= w \end{aligned}$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= c_x v \\ y &= c_y w \end{aligned}$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v \cos \theta - w \sin \theta \\ y &= v \sin \theta + w \cos \theta \end{aligned}$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$\begin{aligned} x &= v + t_x \\ y &= w + t_y \end{aligned}$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v + s_v w \\ y &= w \end{aligned}$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v \\ y &= s_h v + w \end{aligned}$	

each location,  $(v, w)$ , computing the spatial location,  $(x, y)$ , of the corresponding pixel in the output image using Eq. (2.6-23) directly. A problem with the forward mapping approach is that two or more pixels in the input image can be transformed to the same location in the output image, raising the question of how to combine multiple output values into a single output pixel. In addition, it is possible that some output locations may not be assigned a pixel at all. The second approach, called *inverse mapping*, scans the output pixel locations and, at each location,  $(x, y)$ , computes the corresponding location in the input image using  $(v, w) = T^{-1}(x, y)$ . It then interpolates (using one of the techniques discussed in Section 2.4.4) among the nearest input pixels to determine the intensity of the output pixel value. Inverse mappings are more efficient to implement than forward mappings and are used in numerous commercial implementations of spatial transformations (for example, MATLAB uses this approach).





**FIGURE 2.36** (a) A 300 dpi image of the letter T. (b) Image rotated  $21^\circ$  using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated  $21^\circ$  using bilinear interpolation. (d) Image rotated  $21^\circ$  using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

■ The objective of this example is to illustrate image rotation using an affine transform. Figure 2.36(a) shows a 300 dpi image and Figs. 2.36(b)–(d) are the results of rotating the original image by  $21^\circ$ , using nearest neighbor, bilinear, and bicubic interpolation, respectively. Rotation is one of the most demanding geometric transformations in terms of preserving straight-line features. As we see in the figure, nearest neighbor interpolation produced the most jagged edges and, as in Section 2.4.4, bilinear interpolation yielded significantly improved results. As before, using bicubic interpolation produced slightly sharper results. In fact, if you compare the enlarged detail in Figs. 2.36(c) and (d), you will notice in the middle of the subimages that the number of vertical gray “blocks” that provide the intensity transition from light to dark in Fig. 2.36(c) is larger than the corresponding number of blocks in (d), indicating that the latter is a sharper edge. Similar results would be obtained with the other spatial transformations in Table 2.2 that require interpolation (the identity transformation does not, and neither does the translation transformation if the increments are an integer number of pixels). This example was implemented using the inverse mapping approach discussed in the preceding paragraph. ■

**EXAMPLE 2.9:** Image rotation and intensity interpolation.

Image registration is an important application of digital image processing used to align two or more images of the same scene. In the preceding discussion, the form of the transformation function required to achieve a desired geometric transformation was known. In image registration, we have available the input and output images, but the specific transformation that produced the output image from the input generally is unknown. The problem, then, is to estimate the transformation function and then use it to register the two images. To clarify terminology, the input image is the image that we wish to transform, and what we call the *reference* image is the image against which we want to register the input.

For example, it may be of interest to align (register) two or more images taken at approximately the same time, but using different imaging systems, such as an MRI (magnetic resonance imaging) scanner and a PET (positron emission tomography) scanner. Or, perhaps the images were taken at different times using the same instrument, such as satellite images of a given location taken several days, months, or even years apart. In either case, combining the images or performing quantitative analysis and comparisons between them requires compensating for geometric distortions caused by differences in viewing angle, distance, and orientation; sensor resolution; shift in object positions; and other factors.

One of the principal approaches for solving the problem just discussed is to use *tie points* (also called *control points*), which are corresponding points whose locations are known precisely in the input and reference images. There are numerous ways to select tie points, ranging from interactively selecting them to applying algorithms that attempt to detect these points automatically. In some applications, imaging systems have physical artifacts (such as small metallic objects) embedded in the imaging sensors. These produce a set of *known points* (called *reseau marks*) directly on all images captured by the system, which can be used as guides for establishing tie points.

The problem of estimating the transformation function is one of modeling. For example, suppose that we have a set of four tie points each in an input and a reference image. A simple model based on a bilinear approximation is given by

$$x = c_1v + c_2w + c_3vw + c_4 \quad (2.6-24)$$

and

$$y = c_5v + c_6w + c_7vw + c_8 \quad (2.6-25)$$

where, during the estimation phase,  $(v, w)$  and  $(x, y)$  are the coordinates of tie points in the input and reference images, respectively. If we have four pairs of corresponding tie points in both images, we can write eight equations using Eqs. (2.6-24) and (2.6-25) and use them to solve for the eight unknown coefficients,  $c_1, c_2, \dots, c_8$ . These coefficients constitute the model that transforms the pixels of one image into the locations of the pixels of the other to achieve registration.

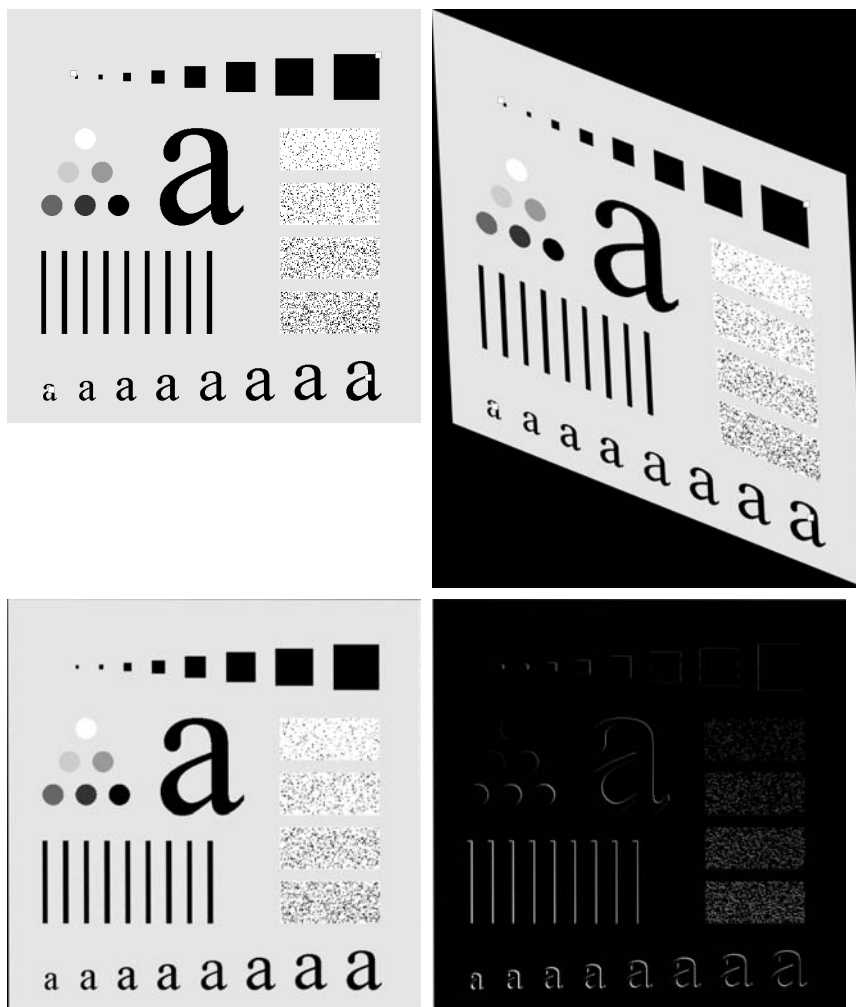
Once we have the coefficients, Eqs. (2.6-24) and (2.6-25) become our vehicle for transforming all the pixels in the input image to generate the desired new image, which, if the tie points were selected correctly, should be registered with the reference image. In situations where four tie points are insufficient to obtain satisfactory registration, an approach used frequently is to select a larger number of tie points and then treat the quadrilaterals formed by groups of four tie points as subimages. The subimages are processed as above, with all the pixels within a quadrilateral being transformed using the coefficients determined from those tie points. Then we move to another set of four tie points and repeat the procedure until all quadrilateral regions have been processed. Of course, it is possible to use regions that are more complex than quadrilaterals and employ more complex models, such as polynomials fitted by least

## 2.6 ■ An Introduction to the Mathematical Tools Used in Digital Image Processing 91

squares algorithms. In general, the number of control points and sophistication of the model required to solve a problem is dependent on the severity of the geometric distortion. Finally, keep in mind that the transformation defined by Eqs. (2.6-24) and (2.6-25), or any other model for that matter, simply maps the spatial coordinates of the pixels in the input image. We still need to perform intensity interpolation using any of the methods discussed previously to assign intensity values to those pixels.

■ Figure 2.37(a) shows a reference image and Fig. 2.37(b) shows the same image, but distorted geometrically by vertical and horizontal shear. Our objective is to use the reference image to obtain tie points and then use the tie points to register the images. The tie points we selected (manually) are shown as small white squares near the corners of the images (we needed only four tie

**EXAMPLE 2.10:**  
Image registration.



a b  
c d

**FIGURE 2.37**

Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the border). (d) Difference between (a) and (c), showing more registration errors.

points because the distortion is linear shear in both directions). Figure 2.37(c) shows the result of using these tie points in the procedure discussed in the preceding paragraphs to achieve registration. We note that registration was not perfect, as is evident by the black edges in Fig. 2.37(c). The difference image in Fig. 2.37(d) shows more clearly the slight lack of registration between the reference and corrected images. The reason for the discrepancies is error in the manual selection of the tie points. It is difficult to achieve perfect matches for tie points when distortion is so severe. ■



Consult the Tutorials section in the book Web site for a brief tutorial on vectors and matrices.

## 2.6.6 Vector and Matrix Operations

Multispectral image processing is a typical area in which vector and matrix operations are used routinely. For example, you will learn in Chapter 6 that color images are formed in RGB color space by using red, green, and blue component images, as Fig. 2.38 illustrates. Here we see that *each* pixel of an RGB image has three components, which can be organized in the form of a *column vector*

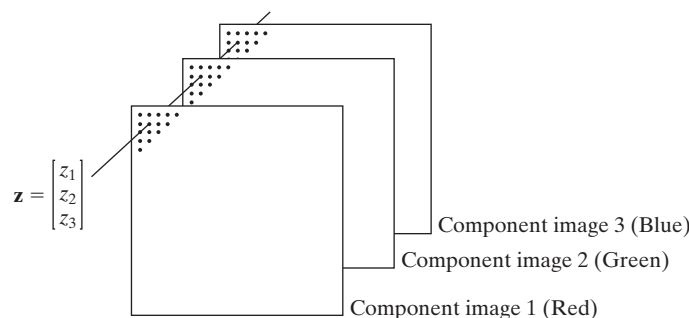
$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (2.6-26)$$

where  $z_1$  is the intensity of the pixel in the red image, and the other two elements are the corresponding pixel intensities in the green and blue images, respectively. Thus an RGB color image of size  $M \times N$  can be represented by three component images of this size, or by a total of  $MN$  3-D vectors. A general multispectral case involving  $n$  component images (e.g., see Fig. 1.10) will result in  $n$ -dimensional vectors. We use this type of vector representation in parts of Chapters 6, 10, 11, and 12.

Once pixels have been represented as vectors we have at our disposal the tools of vector-matrix theory. For example, the *Euclidean distance*,  $D$ , between a pixel vector  $\mathbf{z}$  and an arbitrary point  $\mathbf{a}$  in  $n$ -dimensional space is defined as the vector product

$$\begin{aligned} D(\mathbf{z}, \mathbf{a}) &= [(\mathbf{z} - \mathbf{a})^T(\mathbf{z} - \mathbf{a})]^{\frac{1}{2}} \\ &= [(z_1 - a_1)^2 + (z_2 - a_2)^2 + \cdots + (z_n - a_n)^2]^{\frac{1}{2}} \end{aligned} \quad (2.6-27)$$

**FIGURE 2.38**  
Formation of a vector from corresponding pixel values in three RGB component images.



We see that this is a generalization of the 2-D Euclidean distance defined in Eq. (2.5-1). Equation (2.6-27) sometimes is referred to as a *vector norm*, denoted by  $\|\mathbf{z} - \mathbf{a}\|$ . We will use distance computations numerous times in later chapters.

Another important advantage of pixel vectors is in linear transformations, represented as

$$\mathbf{w} = \mathbf{A}(\mathbf{z} - \mathbf{a}) \quad (2.6-28)$$

where  $\mathbf{A}$  is a matrix of size  $m \times n$  and  $\mathbf{z}$  and  $\mathbf{a}$  are column vectors of size  $n \times 1$ . As you will learn later, transformations of this type have a number of useful applications in image processing.

As noted in Eq. (2.4-2), entire images can be treated as matrices (or, equivalently, as vectors), a fact that has important implication in the solution of numerous image processing problems. For example, we can express an image of size  $M \times N$  as a vector of dimension  $MN \times 1$  by letting the first row of the image be the first  $N$  elements of the vector, the second row the next  $N$  elements, and so on. With images formed in this manner, we can express a broad range of linear processes applied to an image by using the notation

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n} \quad (2.6-29)$$

where  $\mathbf{f}$  is an  $MN \times 1$  vector representing an input image,  $\mathbf{n}$  is an  $MN \times 1$  vector representing an  $M \times N$  noise pattern,  $\mathbf{g}$  is an  $MN \times 1$  vector representing a processed image, and  $\mathbf{H}$  is an  $MN \times MN$  matrix representing a linear process applied to the input image (see Section 2.6.2 regarding linear processes). It is possible, for example, to develop an entire body of generalized techniques for image restoration starting with Eq. (2.6-29), as we discuss in Section 5.9. We touch on the topic of using matrices again in the following section, and show other uses of matrices for image processing in Chapters 5, 8, 11, and 12.

### 2.6.7 Image Transforms

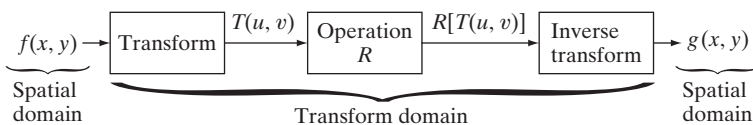
All the image processing approaches discussed thus far operate directly on the pixels of the input image; that is, they work directly in the *spatial domain*. In some cases, image processing tasks are best formulated by transforming the input images, carrying the specified task in a *transform domain*, and applying the inverse transform to return to the spatial domain. You will encounter a number of different transforms as you proceed through the book. A particularly important class of 2-D linear transforms, denoted  $T(u, v)$ , can be expressed in the general form

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)r(x, y, u, v) \quad (2.6-30)$$

where  $f(x, y)$  is the input image,  $r(x, y, u, v)$  is called the *forward transformation kernel*, and Eq. (2.6-30) is evaluated for  $u = 0, 1, 2, \dots, M - 1$  and  $v = 0, 1, 2, \dots, N - 1$ . As before,  $x$  and  $y$  are spatial variables, while  $M$  and  $N$

**FIGURE 2.39**

General approach for operating in the linear transform domain.



are the row and column dimensions of  $f$ . Variables  $u$  and  $v$  are called the *transform variables*.  $T(u, v)$  is called the *forward transform* of  $f(x, y)$ . Given  $T(u, v)$ , we can recover  $f(x, y)$  using the *inverse transform* of  $T(u, v)$ ,

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v)s(x, y, u, v) \quad (2.6-31)$$

for  $x = 0, 1, 2, \dots, M - 1$  and  $y = 0, 1, 2, \dots, N - 1$ , where  $s(x, y, u, v)$  is called the *inverse transformation kernel*. Together, Eqs. (2.6-30) and (2.6-31) are called a *transform pair*.

Figure 2.39 shows the basic steps for performing image processing in the linear transform domain. First, the input image is transformed, the transform is then modified by a predefined operation, and, finally, the output image is obtained by computing the inverse of the modified transform. Thus, we see that the process goes from the spatial domain to the transform domain and then back to the spatial domain.

**EXAMPLE 2.11:**

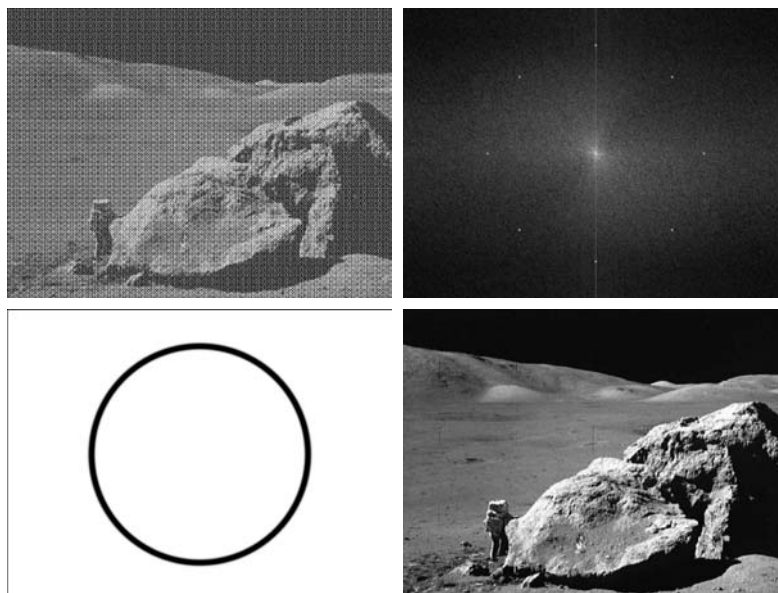
Image processing in the transform domain.

■ Figure 2.40 shows an example of the steps in Fig. 2.39. In this case the transform used was the Fourier transform, which we mention briefly later in this section and discuss in detail in Chapter 4. Figure 2.40(a) is an image corrupted

a b  
c d

**FIGURE 2.40**

(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)



## 2.6 ■ An Introduction to the Mathematical Tools Used in Digital Image Processing 95

by sinusoidal interference, and Fig. 2.40(b) is the magnitude of its Fourier transform, which is the output of the first stage in Fig. 2.39. As you will learn in Chapter 4, sinusoidal interference in the spatial domain appears as bright bursts of intensity in the transform domain. In this case, the bursts are in a circular pattern that can be seen in Fig. 2.40(b). Figure 2.40(c) shows a mask image (called a *filter*) with white and black representing 1 and 0, respectively. For this example, the operation in the second box of Fig. 2.39 is to multiply the mask by the transform, thus eliminating the bursts responsible for the interference. Figure 2.40(d) shows the final result, obtained by computing the inverse of the modified transform. The interference is no longer visible, and important detail is quite clear. In fact, you can even see the *fiducial marks* (faint crosses) that are used for image alignment. ■

The forward transformation kernel is said to be *separable* if

$$r(x, y, u, v) = r_1(x, u)r_2(y, v) \quad (2.6-32)$$

In addition, the kernel is said to be *symmetric* if  $r_1(x, y)$  is functionally equal to  $r_2(x, y)$ , so that

$$r(x, y, u, v) = r_1(x, u)r_1(y, v) \quad (2.6-33)$$

Identical comments apply to the inverse kernel by replacing  $r$  with  $s$  in the preceding equations.

The 2-D Fourier transform discussed in Example 2.11 has the following forward and inverse kernels:

$$r(x, y, u, v) = e^{-j2\pi(ux/M+vy/N)} \quad (2.6-34)$$

and

$$s(x, y, u, v) = \frac{1}{MN} e^{j2\pi(ux/M+vy/N)} \quad (2.6-35)$$

respectively, where  $j = \sqrt{-1}$ , so these kernels are complex. Substituting these kernels into the general transform formulations in Eqs. (2.6-30) and (2.6-31) gives us the *discrete Fourier transform pair*:

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)} \quad (2.6-36)$$

and

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) e^{j2\pi(ux/M+vy/N)} \quad (2.6-37)$$

These equations are of fundamental importance in digital image processing, and we devote most of Chapter 4 to deriving them starting from basic principles and then using them in a broad range of applications.

It is not difficult to show that the Fourier kernels are separable and symmetric (Problem 2.25), and that separable and symmetric kernels allow 2-D transforms to be computed using 1-D transforms (Problem 2.26). When the

forward and inverse kernels of a transform pair satisfy these two conditions, and  $f(x, y)$  is a square image of size  $M \times M$ , Eqs. (2.6-30) and (2.6-31) can be expressed in matrix form:

$$\mathbf{T} = \mathbf{AFA} \quad (2.6-38)$$

where  $\mathbf{F}$  is an  $M \times M$  matrix containing the elements of  $f(x, y)$  [see Eq. (2.4-2)],  $\mathbf{A}$  is an  $M \times M$  matrix with elements  $a_{ij} = r_1(i, j)$ , and  $\mathbf{T}$  is the resulting  $M \times M$  transform, with values  $T(u, v)$  for  $u, v = 0, 1, 2, \dots, M - 1$ .

To obtain the inverse transform, we pre- and post-multiply Eq. (2.6-38) by an inverse transformation matrix  $\mathbf{B}$ :

$$\mathbf{BTB} = \mathbf{BAFAB} \quad (2.6-39)$$

If  $\mathbf{B} = \mathbf{A}^{-1}$ ,

$$\mathbf{F} = \mathbf{BTB} \quad (2.6-40)$$

indicating that  $\mathbf{F}$  [whose elements are equal to image  $f(x, y)$ ] can be recovered completely from its forward transform. If  $\mathbf{B}$  is not equal to  $\mathbf{A}^{-1}$ , then use of Eq. (2.6-40) yields an approximation:

$$\hat{\mathbf{F}} = \mathbf{BAFAB} \quad (2.6-41)$$

In addition to the Fourier transform, a number of important transforms, including the Walsh, Hadamard, discrete cosine, Haar, and slant transforms, can be expressed in the form of Eqs. (2.6-30) and (2.6-31) or, equivalently, in the form of Eqs. (2.6-38) and (2.6-40). We discuss several of these and some other types of image transforms in later chapters.

### 2.6.8 Probabilistic Methods

Probability finds its way into image processing work in a number of ways. The simplest is when we treat intensity values as random quantities. For example, let  $z_i, i = 0, 1, 2, \dots, L - 1$ , denote the values of all possible intensities in an  $M \times N$  digital image. The probability,  $p(z_k)$ , of intensity level  $z_k$  occurring in a given image is estimated as

$$p(z_k) = \frac{n_k}{MN} \quad (2.6-42)$$

where  $n_k$  is the number of times that intensity  $z_k$  occurs in the image and  $MN$  is the total number of pixels. Clearly,

$$\sum_{k=0}^{L-1} p(z_k) = 1 \quad (2.6-43)$$

Once we have  $p(z_k)$ , we can determine a number of important image characteristics. For example, the mean (average) intensity is given by

$$m = \sum_{k=0}^{L-1} z_k p(z_k) \quad (2.6-44)$$



Consult the Tutorials section in the book Web site for a brief overview of probability theory.



## 2.6 ■ An Introduction to the Mathematical Tools Used in Digital Image Processing 97

Similarly, the variance of the intensities is

$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 p(z_k) \quad (2.6-45)$$

The variance is a measure of the spread of the values of  $z$  about the mean, so it is a useful measure of image contrast. In general, the  $n$ th moment of random variable  $z$  about the mean is defined as

$$\mu_n(z) = \sum_{k=0}^{L-1} (z_k - m)^n p(z_k) \quad (2.6-46)$$

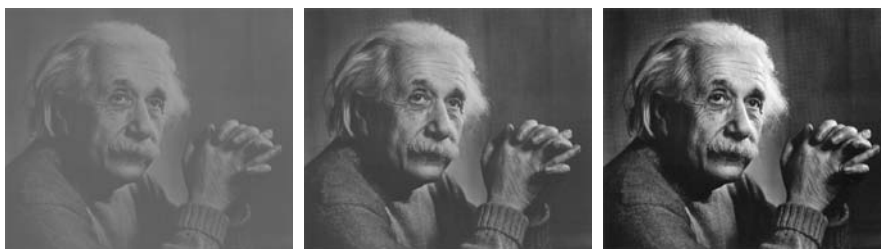
We see that  $\mu_0(z) = 1$ ,  $\mu_1(z) = 0$ , and  $\mu_2(z) = \sigma^2$ . Whereas the mean and variance have an immediately obvious relationship to visual properties of an image, higher-order moments are more subtle. For example, a positive third moment indicates that the intensities are biased to values higher than the mean, a negative third moment would indicate the opposite condition, and a zero third moment would tell us that the intensities are distributed approximately equally on both sides of the mean. These features are useful for computational purposes, but they do not tell us much about the appearance of an image in general.

The units of the variance are in intensity values squared. When comparing contrast values, we usually use the standard deviation,  $\sigma$  (square root of the variance), instead because its dimensions are directly in terms of intensity values.

■ Figure 2.41 shows three 8-bit images exhibiting low, medium, and high contrast, respectively. The standard deviations of the pixel intensities in the three images are 14.3, 31.6, and 49.2 intensity levels, respectively. The corresponding variance values are 204.3, 997.8, and 2424.9, respectively. Both sets of values tell the same story but, given that the range of possible intensity values in these images is  $[0, 255]$ , the standard deviation values relate to this range much more intuitively than the variance. ■

**EXAMPLE 2.12:** Comparison of standard deviation values as measures of image intensity contrast.

As you will see in progressing through the book, concepts from probability play a central role in the development of image processing algorithms. For example, in Chapter 3 we use the probability measure in Eq. (2.6-42) to derive intensity transformation algorithms. In Chapter 5, we use probability and matrix formulations to develop image restoration algorithms. In Chapter 10, probability is used for image segmentation, and in Chapter 11 we use it for texture description. In Chapter 12, we derive optimum object recognition techniques based on a probabilistic formulation.



a b c

**FIGURE 2.41** Images exhibiting (a) low contrast, (b) medium contrast, and (c) high contrast.

Thus far, we have addressed the issue of applying probability to a single random variable (intensity) over a single 2-D image. If we consider sequences of images, we may interpret the third variable as time. The tools needed to handle this added complexity are *stochastic* image processing techniques (the word *stochastic* is derived from a Greek word meaning roughly “to aim at a target,” implying randomness in the outcome of the process). We can go a step further and consider an *entire* image (as opposed to a point) to be a spatial random event. The tools needed to handle formulations based on this concept are techniques from *random fields*. We give one example in Section 5.8 of how to treat entire images as random events, but further discussion of stochastic processes and random fields is beyond the scope of this book. The references at the end of this chapter provide a starting point for reading about these topics.

## Summary

The material in this chapter is primarily background for subsequent discussions. Our treatment of the human visual system, although brief, provides a basic idea of the capabilities of the eye in perceiving pictorial information. The discussion on light and the electromagnetic spectrum is fundamental in understanding the origin of the many images we use in this book. Similarly, the image model developed in Section 2.3.4 is used in the Chapter 4 as the basis for an image enhancement technique called *homomorphic filtering*.

The sampling and interpolation ideas introduced in Section 2.4 are the foundation for many of the digitizing phenomena you are likely to encounter in practice. We will return to the issue of sampling and many of its ramifications in Chapter 4, after you have mastered the Fourier transform and the frequency domain.

The concepts introduced in Section 2.5 are the basic building blocks for processing techniques based on pixel neighborhoods. For example, as we show in the following chapter, and in Chapter 5, neighborhood processing methods are at the core of many image enhancement and restoration procedures. In Chapter 9, we use neighborhood operations for image morphology; in Chapter 10, we use them for image segmentation; and in Chapter 11 for image description. When applicable, neighborhood processing is favored in commercial applications of image processing because of their operational speed and simplicity of implementation in hardware and/or firmware.

The material in Section 2.6 will serve you well in your journey through the book. Although the level of the discussion was strictly introductory, you are now in a position to conceptualize what it means to process a digital image. As we mentioned in that section, the tools introduced there are expanded as necessary in the following chapters. Rather than dedicate an entire chapter or appendix to develop a comprehensive treatment of mathematical concepts in one place, you will find it considerably more meaningful to learn the necessary extensions of the mathematical tools from Section 2.6 in later chapters, in the context of how they are applied to solve problems in image processing.

## References and Further Reading

Additional reading for the material in Section 2.1 regarding the structure of the human eye may be found in Atchison and Smith [2000] and Oyster [1999]. For additional reading on visual perception, see Regan [2000] and Gordon [1997]. The book by Hubel [1988] and the classic book by Cornsweet [1970] also are of interest. Born and Wolf [1999] is a basic reference that discusses light in terms of electromagnetic theory. Electromagnetic energy propagation is covered in some detail by Felsen and Marcuvitz [1994].

The area of image sensing is quite broad and very fast moving. An excellent source of information on optical and other imaging sensors is the Society for Optical Engineering (SPIE). The following are representative publications by the SPIE in this area: Blouke et al. [2001], Hoover and Doty [1996], and Freeman [1987].

The image model presented in Section 2.3.4 is from Oppenheim, Schafer, and Stockham [1968]. A reference for the illumination and reflectance values used in that section is the *IESNA Lighting Handbook* [2000]. For additional reading on image sampling and some of its effects, such as aliasing, see Bracewell [1995]. We discuss this topic in more detail in Chapter 4. The early experiments mentioned in Section 2.4.3 on perceived image quality as a function of sampling and quantization were reported by Huang [1965]. The issue of reducing the number of samples and intensity levels in an image while minimizing the ensuing degradation is still of current interest, as exemplified by Papamarkos and Atsalakis [2000]. For further reading on image shrinking and zooming, see Sid-Ahmed [1995], Unser et al. [1995], Umbaugh [2005], and Lehmann et al. [1999]. For further reading on the topics covered in Section 2.5, see Rosenfeld and Kak [1982], Marchand-Maillet and Sharaiha [2000], and Ritter and Wilson [2001].

Additional reading on linear systems in the context of image processing (Section 2.6.2) may be found in Castleman [1996]. The method of noise reduction by image averaging (Section 2.6.3) was first proposed by Kohler and Howell [1963]. See Peebles [1993] regarding the expected value of the mean and variance of a sum of random variables. Image subtraction (Section 2.6.3) is a generic image processing tool used widely for change detection. For image subtraction to make sense, it is necessary that the images being subtracted be registered or, alternatively, that any artifacts due to motion be identified. Two papers by Meijering et al. [1999, 2001] are illustrative of the types of techniques used to achieve these objectives.

A basic reference for the material in Section 2.6.4 is Cameron [2005]. For more advanced reading on this topic, see Toulakis [2003]. For an introduction to fuzzy sets, see Section 3.8 and the corresponding references in Chapter 3. For further details on single-point and neighborhood processing (Section 2.6.5), see Sections 3.2 through 3.4 and the references on these topics in Chapter 3. For geometric spatial transformations, see Wolberg [1990].

Noble and Daniel [1988] is a basic reference for matrix and vector operations (Section 2.6.6). See Chapter 4 for a detailed discussion on the Fourier transform (Section 2.6.7), and Chapters 7, 8, and 11 for examples of other types of transforms used in digital image processing. Peebles [1993] is a basic introduction to probability and random variables (Section 2.6.8) and Papoulis [1991] is a more advanced treatment of this topic. For foundation material on the use of stochastic and random fields for image processing, see Rosenfeld and Kak [1982], Jähne [2002], and Won and Gray [2004].

For details of software implementation of many of the techniques illustrated in this chapter, see Gonzalez, Woods, and Eddins [2004].

## Problems

- ★2.1 Using the background information provided in Section 2.1, and thinking purely in geometric terms, estimate the diameter of the smallest printed dot that the eye can discern if the page on which the dot is printed is 0.2 m away from the eyes. Assume for simplicity that the visual system ceases to detect the dot when the image of the dot on the fovea becomes smaller than the diameter of one receptor (cone) in that area of the retina. Assume further that the fovea can be



Detailed solutions to the problems marked with a star can be found in the book Web site. The site also contains suggested projects based on the material in this chapter.

modeled as a square array of dimensions  $1.5 \text{ mm} \times 1.5 \text{ mm}$ , and that the cones and spaces between the cones are distributed uniformly throughout this array.

- 2.2** When you enter a dark theater on a bright day, it takes an appreciable interval of time before you can see well enough to find an empty seat. Which of the visual processes explained in Section 2.1 is at play in this situation?
- ★2.3** Although it is not shown in Fig. 2.10, alternating current certainly is part of the electromagnetic spectrum. Commercial alternating current in the United States has a frequency of 60 Hz. What is the wavelength in kilometers of this component of the spectrum?
- 2.4** You are hired to design the front end of an imaging system for studying the boundary shapes of cells, bacteria, viruses, and protein. The front end consists, in this case, of the illumination source(s) and corresponding imaging camera(s). The diameters of circles required to enclose individual specimens in each of these categories are 50, 1, 0.1, and  $0.01 \mu\text{m}$ , respectively.
- (a)** Can you solve the imaging aspects of this problem with a single sensor and camera? If your answer is yes, specify the illumination wavelength band and the type of camera needed. By “type,” we mean the band of the electromagnetic spectrum to which the camera is most sensitive (e.g., infrared).
- (b)** If your answer in (a) is no, what type of illumination sources and corresponding imaging sensors would you recommend? Specify the light sources and cameras as requested in part (a). Use the *minimum* number of illumination sources and cameras needed to solve the problem.
- By “solving the problem,” we mean being able to detect circular details of diameter 50, 1, 0.1, and  $0.01 \mu\text{m}$ , respectively.
- 2.5** A CCD camera chip of dimensions  $7 \times 7 \text{ mm}$ , and having  $1024 \times 1024$  elements, is focused on a square, flat area, located 0.5 m away. How many line pairs per mm will this camera be able to resolve? The camera is equipped with a 35-mm lens. (*Hint:* Model the imaging process as in Fig. 2.3, with the focal length of the camera lens substituting for the focal length of the eye.)
- ★2.6** An automobile manufacturer is automating the placement of certain components on the bumpers of a limited-edition line of sports cars. The components are color coordinated, so the robots need to know the color of each car in order to select the appropriate bumper component. Models come in only four colors: blue, green, red, and white. You are hired to propose a solution based on imaging. How would you solve the problem of automatically determining the color of each car, keeping in mind that *cost* is the most important consideration in your choice of components?
- 2.7** Suppose that a flat area with center at  $(x_0, y_0)$  is illuminated by a light source with intensity distribution
- $$i(x, y) = Ke^{-[(x-x_0)^2+(y-y_0)^2]}$$
- Assume for simplicity that the reflectance of the area is constant and equal to 1.0, and let  $K = 255$ . If the resulting image is digitized with  $k$  bits of intensity resolution, and the eye can detect an abrupt change of eight shades of intensity between adjacent pixels, what value of  $k$  will cause visible false contouring?
- 2.8** Sketch the image in Problem 2.7 for  $k = 2$ .
- ★2.9** A common measure of transmission for digital data is the *baud rate*, defined as the number of bits transmitted per second. Generally, transmission is accomplished

in packets consisting of a start bit, a byte (8 bits) of information, and a stop bit. Using these facts, answer the following:

- (a) How many minutes would it take to transmit a  $1024 \times 1024$  image with 256 intensity levels using a 56K baud modem?
- (b) What would the time be at 3000K baud, a representative medium speed of a phone DSL (Digital Subscriber Line) connection?

**2.10** High-definition television (HDTV) generates images with 1125 horizontal TV lines interlaced (where every other line is painted on the tube face in each of two fields, each field being  $1/60$ th of a second in duration). The width-to-height aspect ratio of the images is 16:9. The fact that the number of horizontal lines is fixed determines the vertical resolution of the images. A company has designed an image capture system that generates digital images from HDTV images. The resolution of each TV (horizontal) line in their system is in proportion to vertical resolution, with the proportion being the width-to-height ratio of the images. Each pixel in the color image has 24 bits of intensity resolution, 8 bits each for a red, a green, and a blue image. These three “primary” images form a color image. How many bits would it take to store a 2-hour HDTV movie?

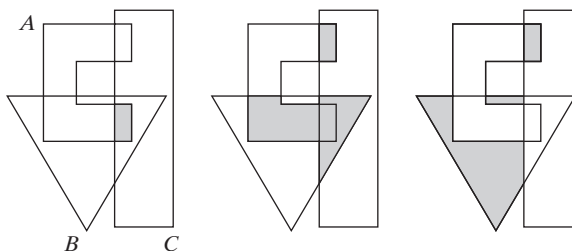
- ★**2.11** Consider the two image subsets,  $S_1$  and  $S_2$ , shown in the following figure. For  $V = \{1\}$ , determine whether these two subsets are (a) 4-adjacent, (b) 8-adjacent, or (c)  $m$ -adjacent.

	$S_1$					$S_2$				
0	0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1	1

- ★**2.12** Develop an algorithm for converting a one-pixel-thick 8-path to a 4-path.
- 2.13** Develop an algorithm for converting a one-pixel-thick  $m$ -path to a 4-path.
- 2.14** Refer to the discussion at the end of Section 2.5.2, where we defined the background as  $(R_u)^c$ , the complement of the union of all the regions in an image. In some applications, it is advantageous to define the background as the subset of pixels  $(R_u)^c$  that are not region hole pixels (informally, think of holes as sets of background pixels surrounded by region pixels). How would you modify the definition to exclude hole pixels from  $(R_u)^c$ ? An answer such as “the background is the subset of pixels of  $(R_u)^c$  that are not hole pixels” is not acceptable. (*Hint:* Use the concept of connectivity.)
- 2.15** Consider the image segment shown.
- ★(a) Let  $V = \{0, 1\}$  and compute the lengths of the shortest 4-, 8-, and  $m$ -path between  $p$  and  $q$ . If a particular path does not exist between these two points, explain why.
- (b) Repeat for  $V = \{1, 2\}$ .

	3	1	2	1( $q$ )
	2	2	0	2
	1	2	1	1
( $p$ )	1	0	1	2

- 2.16** ★(a) Give the condition(s) under which the  $D_4$  distance between two points  $p$  and  $q$  is equal to the shortest 4-path between these points.  
 (b) Is this path unique?
- 2.17** Repeat Problem 2.16 for the  $D_8$  distance.
- ★**2.18** In the next chapter, we will deal with operators whose function is to compute the sum of pixel values in a small subimage area,  $S$ . Show that these are linear operators.
- 2.19** The median,  $\zeta$ , of a set of numbers is such that half the values in the set are below  $\zeta$  and the other half are above it. For example, the median of the set of values  $\{2, 3, 8, 20, 21, 25, 31\}$  is 20. Show that an operator that computes the median of a subimage area,  $S$ , is nonlinear.
- ★**2.20** Prove the validity of Eqs. (2.6-6) and (2.6-7). [*Hint*: Start with Eq. (2.6-4) and use the fact that the expected value of a sum is the sum of the expected values.]
- 2.21** Consider two 8-bit images whose intensity levels span the full range from 0 to 255.  
 (a) Discuss the limiting effect of repeatedly subtracting image (2) from image (1). Assume that the result is represented also in eight bits.  
 (b) Would reversing the order of the images yield a different result?
- ★**2.22** Image subtraction is used often in industrial applications for detecting missing components in product assembly. The approach is to store a “golden” image that corresponds to a correct assembly; this image is then subtracted from incoming images of the same product. Ideally, the differences would be zero if the new products are assembled correctly. Difference images for products with missing components would be nonzero in the area where they differ from the golden image. What conditions do you think have to be met in practice for this method to work?
- 2.23** ★(a) With reference to Fig. 2.31, sketch the set  $(A \cap B) \cup (A \cup B)^c$ .  
 (b) Give expressions for the sets shown shaded in the following figure in terms of sets  $A$ ,  $B$ , and  $C$ . The shaded areas in each figure constitute one set, so give one expression for each of the three figures.



- 2.24** What would be the equations analogous to Eqs. (2.6-24) and (2.6-25) that would result from using triangular instead of quadrilateral regions?
- 2.25** Prove that the Fourier kernels in Eqs. (2.6-34) and (2.6-35) are separable and symmetric.
- ★**2.26** Show that 2-D transforms with separable, symmetric kernels can be computed by (1) computing 1-D transforms along the individual rows (columns) of the input, followed by (2) computing 1-D transforms along the columns (rows) of the result from step (1).

- 2.27** A plant produces a line of translucent miniature polymer squares. Stringent quality requirements dictate 100% visual inspection, and the plant manager finds the use of human inspectors increasingly expensive. Inspection is semiautomated. At each inspection station, a robotic mechanism places each polymer square over a light located under an optical system that produces a magnified image of the square. The image completely fills a viewing screen measuring  $80 \times 80$  mm. Defects appear as dark circular blobs, and the inspector's job is to look at the screen and reject any sample that has one or more such dark blobs with a diameter of 0.8 mm or larger, as measured on the scale of the screen. The manager believes that if she can find a way to automate the process completely, she will increase profits by 50%. She also believes that success in this project will aid her climb up the corporate ladder. After much investigation, the manager decides that the way to solve the problem is to view each inspection screen with a CCD TV camera and feed the output of the camera into an image processing system capable of detecting the blobs, measuring their diameter, and activating the accept/reject buttons previously operated by an inspector. She is able to find a system that can do the job, as long as the smallest defect occupies an area of at least  $2 \times 2$  pixels in the digital image. The manager hires you to help her specify the camera and lens system, but requires that you use off-the-shelf components. For the lenses, assume that this constraint means any integer multiple of 25 mm or 35 mm, up to 200 mm. For the cameras, it means resolutions of  $512 \times 512$ ,  $1024 \times 1024$ , or  $2048 \times 2048$  pixels. The *individual* imaging elements in these cameras are squares measuring  $8 \times 8$   $\mu\text{m}$ , and the spaces between imaging elements are 2  $\mu\text{m}$ . For this application, the cameras cost much more than the lenses, so the problem should be solved with the lowest-resolution camera possible, based on the choice of lenses. As a consultant, you are to provide a written recommendation, showing in reasonable detail the analysis that led to your conclusion. Use the same imaging geometry suggested in Problem 2.5.