3.3 Histogram Processing

The *histogram* of a digital image with intensity levels in the range $[0, L - 1]$ is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value and n_k is the number of pixels in the image with intensity r_k. It is common practice to normalize a histogram by dividing each of its components by the total number of pixels in the image, denoted by the product MN, where, as usual, M and N are the row and column dimensions of the image. Thus, a normalized histogram is given by $p(r_k) = r_k/MN$, for $k = 0, 1, 2, \dots, L - 1$. Loosely speaking, $p(r_k)$ is an estimate of the probability of occurrence of intensity level r_k in an image. The sum of all components of a normalized histogram is equal to 1.

Consult the book Web site for a review of basic probability theory.

Histograms are the basis for numerous spatial domain processing techniques. Histogram manipulation can be used for image enhancement, as shown in this section. In addition to providing useful image statistics, we shall see in subsequent chapters that the information inherent in histograms also is quite useful in other image processing applications, such as image compression and segmentation. Histograms are simple to calculate in software and also lend themselves to economic hardware implementations, thus making them a popular tool for real-time image processing.

As an introduction to histogram processing for intensity transformations, consider Fig. 3.16, which is the pollen image of Fig. 3.10 shown in four basic intensity characteristics: dark, light, low contrast, and high contrast. The right side of the figure shows the histograms corresponding to these images. The horizontal axis of each histogram plot corresponds to intensity values, r_k. The vertical axis corresponds to values of $h(r_k) = n_k$ or $p(r_k) = n_k/MN$ if the values are normalized. Thus, histograms may be viewed graphically simply as plots of $h(r_k) = n_k$ versus r_k or $p(r_k) = n_k/MN$ versus r_k.

We note in the dark image that the components of the histogram are concentrated on the low (dark) side of the intensity scale. Similarly, the components of the histogram of the light image are biased toward the high side of the scale. An image with low contrast has a narrow histogram located typically toward the middle of the intensity scale. For a monochrome image this implies a dull, washed-out gray look. Finally, we see that the components of the histogram in the high-contrast image cover a wide range of the intensity scale and, further, that the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others. Intuitively, it is reasonable to conclude that an image whose pixels tend to occupy the entire range of possible intensity levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones. The net effect will be an image that shows a great deal of gray-level detail and has high dynamic range. It will be shown shortly that it is possible to develop a transformation function that can automatically achieve this effect, based only on information available in the histogram of the input image.

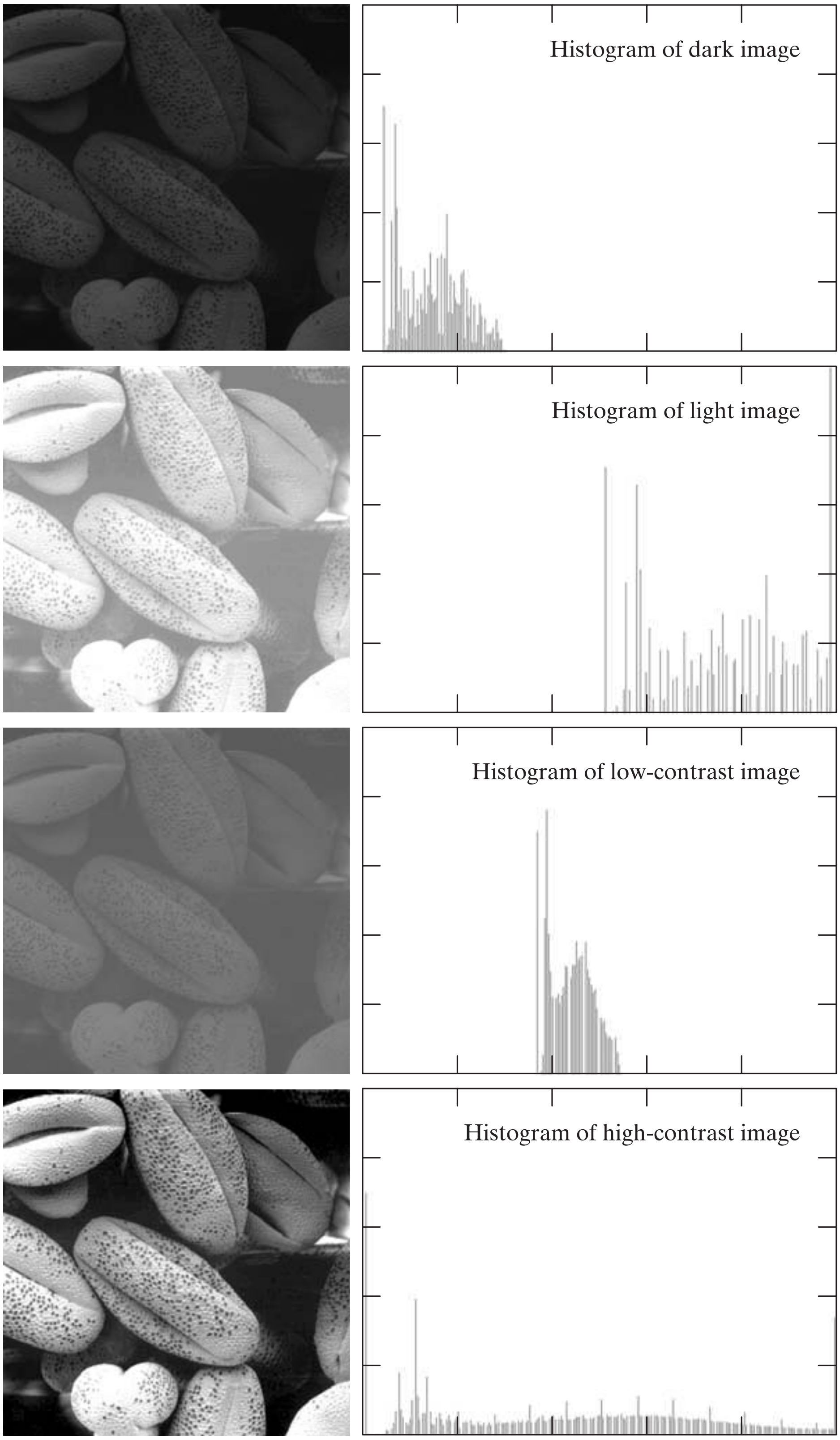


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

3.3.1 Histogram Equalization

Consider for a moment continuous intensity values and let the variable r denote the intensities of an image to be processed. As usual, we assume that r is in the range $[0, L - 1]$, with $r = 0$ representing black and $r = L - 1$ representing white. For r satisfying these conditions, we focus attention on transformations (intensity mappings) of the form

$$s = T(r) \quad 0 \le r \le L - 1 \tag{3.3-1}$$

that produce an output intensity level s for every pixel in the input image having intensity r. We assume that:

(a) $T(r)$ is a monotonically† increasing function in the interval $0 \le r \le L - 1$; and

(b) $0 \le T(r) \le L - 1$ for $0 \le r \le L - 1$.

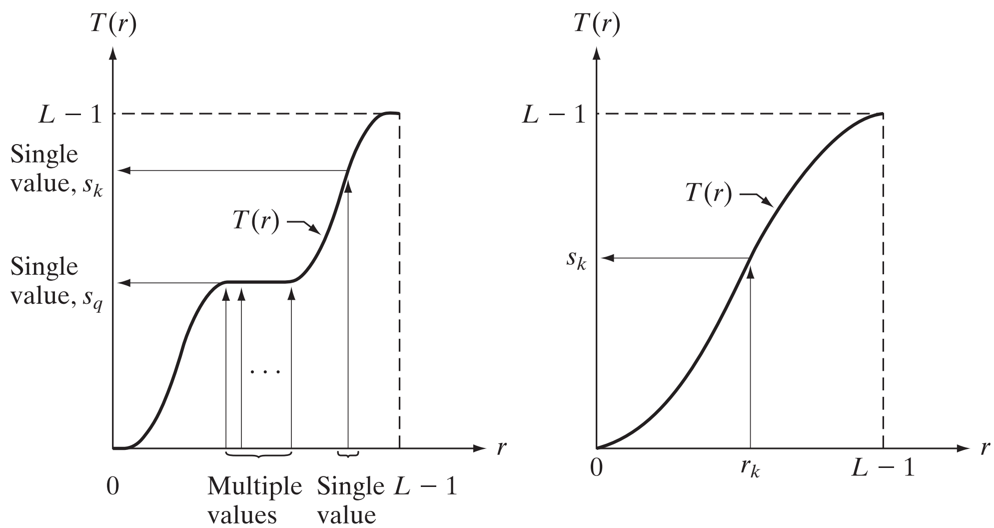
In some formulations to be discussed later, we use the inverse

$$r = T^{-1}(s) \quad 0 \le s \le L - 1 \tag{3.3-2}$$

in which case we change condition (a) to

(a′) $T(r)$ is a strictly monotonically increasing function in the interval $0 \le r \le L - 1$.

The requirement in condition (a) that $T(r)$ be monotonically increasing guarantees that output intensity values will never be less than corresponding input values, thus preventing artifacts created by reversals of intensity. Condition (b) guarantees that the range of output intensities is the same as the input. Finally, condition (a′) guarantees that the mappings from s back to r will be one-to-one, thus preventing ambiguities. Figure 3.17(a) shows a function



a b

FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

†Recall that a function $T(r)$ is *monotonically increasing* if $T(r_2) \ge T(r_1)$ for $r_2 > r_1$. $T(r)$ is a *strictly monotonically increasing* function if $T(r_2) > T(r_1)$ for $r_2 > r_1$. Similar definitions apply to monotonically decreasing functions.

that satisfies conditions (a) and (b). Here, we see that it is possible for multiple values to map to a single value and still satisfy these two conditions. That is, a monotonic transformation function performs a one-to-one or many-to-one mapping. This is perfectly fine when mapping from r to s. However, Fig. 3.17(a) presents a problem if we wanted to recover the values of r uniquely from the mapped values (inverse mapping can be visualized by reversing the direction of the arrows). This would be possible for the inverse mapping of s_k in Fig. 3.17(a), but the inverse mapping of s_q is a *range* of values, which, of course, prevents us in general from recovering the original value of r that resulted in s_q. As Fig. 3.17(b) shows, requiring that $T(r)$ be strictly monotonic guarantees that the inverse mappings will be *single valued* (i.e., the mapping is one-to-one in both directions). This is a theoretical requirement that allows us to derive some important histogram processing techniques later in this chapter. Because in practice we deal with integer intensity values, we are forced to round all results to their nearest integer values. Therefore, when strict monotonicity is not satisfied, we address the problem of a nonunique inverse transformation by looking for the closest integer matches. Example 3.8 gives an illustration of this.

The intensity levels in an image may be viewed as random variables in the interval $[0, L - 1]$. A fundamental descriptor of a random variable is its probability density function (PDF). Let $p_r(r)$ and $p_s(s)$ denote the PDFs of r and s, respectively, where the subscripts on p are used to indicate that p_r and p_s are different functions in general. A fundamental result from basic probability theory is that if $p_r(r)$ and $T(r)$ are known, and $T(r)$ is continuous and differentiable over the range of values of interest, then the PDF of the transformed (mapped) variable s can be obtained using the simple formula

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \tag{3.3-3}$$

Thus, we see that the PDF of the output intensity variable, s, is determined by the PDF of the input intensities and the transformation function used [recall that r and s are related by $T(r)$].

A transformation function of particular importance in image processing has the form

$$s = T(r) = (L - 1)\int_0^r p_r(w)\, dw \tag{3.3-4}$$

where w is a dummy variable of integration. The right side of this equation is recognized as the cumulative distribution function (CDF) of random variable r. Because PDFs always are positive, and recalling that the integral of a function is the area under the function, it follows that the transformation function of Eq. (3.3-4) satisfies condition (a) because the area under the function cannot decrease as r increases. When the upper limit in this equation is $r = (L - 1)$, the integral evaluates to 1 (the area under a PDF curve always is 1), so the maximum value of s is $(L - 1)$ and condition (b) is satisfied also.

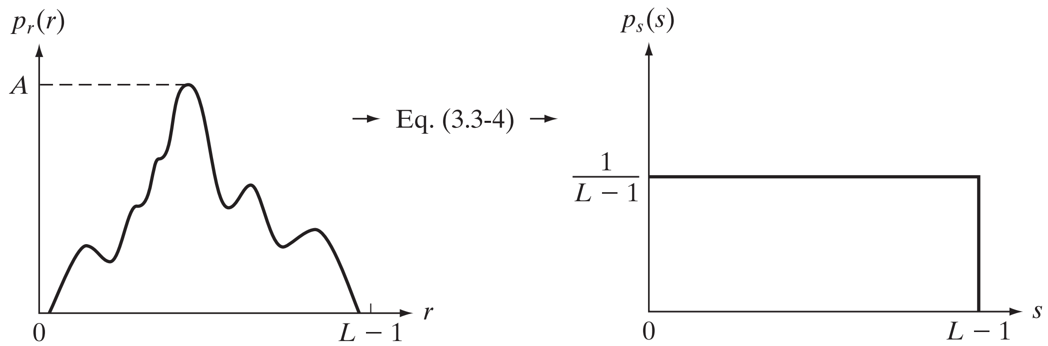
To find the $p_s(s)$ corresponding to the transformation just discussed, we use Eq. (3.3-3). We know from Leibniz's rule in basic calculus that the derivative of a definite integral with respect to its upper limit is the integrand evaluated at the limit. That is,

$$\begin{aligned}\frac{ds}{dr} &= \frac{dT(r)}{dr} \\ &= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] \\ &= (L-1)p_r(r)\end{aligned} \qquad (3.3\text{-}5)$$

Substituting this result for dr/ds in Eq. (3.3-3), and keeping in mind that all probability values are positive, yields

$$\begin{aligned}p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\ &= p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| \\ &= \frac{1}{L-1} \qquad 0 \le s \le L-1\end{aligned} \qquad (3.3\text{-}6)$$

We recognize the form of $p_s(s)$ in the last line of this equation as a *uniform* probability density function. Simply stated, we have demonstrated that performing the intensity transformation in Eq. (3.3-4) yields a random variable, s, characterized by a uniform PDF. It is important to note from this equation that $T(r)$ depends on $p_r(r)$ but, as Eq. (3.3-6) shows, the resulting $p_s(s)$ *always* is uniform, *independently* of the form of $p_r(r)$. Figure 3.18 illustrates these concepts.



a b

FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

EXAMPLE 3.4: Illustration of Eqs. (3.3-4) and (3.3-6).

■ To fix ideas, consider the following simple example. Suppose that the (continuous) intensity values in an image have the PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

From Eq. (3.3-4),

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

Suppose next that we form a new image with intensities, s, obtained using this transformation; that is, the s values are formed by squaring the corresponding intensity values of the input image and dividing them by $(L-1)$. For example, consider an image in which $L = 10$, and suppose that a pixel in an arbitrary location (x, y) in the input image has intensity $r = 3$. Then the pixel in that location in the new image is $s = T(r) = r^2/9 = 1$. We can verify that the PDF of the intensities in the new image is uniform simply by substituting $p_r(r)$ into Eq. (3.3-6) and using the fact that $s = r^2/(L-1)$; that is,

$$\begin{aligned} p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| &= \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right| \\ &= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right| \\ &= \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1} \end{aligned}$$

where the last step follows from the fact that r is nonnegative and we assume that $L > 1$. As expected, the result is a uniform PDF. ■

For discrete values, we deal with probabilities (histogram values) and summations instead of probability density functions and integrals.† As mentioned earlier, the probability of occurrence of intensity level r_k in a digital image is approximated by

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0, 1, 2, \ldots, L-1 \tag{3.3-7}$$

where MN is the total number of pixels in the image, n_k is the number of pixels that have intensity r_k, and L is the number of possible intensity levels in the image (e.g., 256 for an 8-bit image). As noted in the beginning of this section, a plot of $p_r(r_k)$ versus r_k is commonly referred to as a *histogram*.

---

†The conditions of monotonicity stated earlier apply also in the discrete case. We simply restrict the values of the variables to be discrete.

The discrete form of the transformation in Eq. (3.3-4) is

$$\begin{aligned} s_k = T(r_k) &= (L-1)\sum_{j=0}^{k} p_r(r_j) \\ &= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L-1 \end{aligned} \tag{3.3-8}$$

Thus, a processed (output) image is obtained by mapping each pixel in the input image with intensity r_k into a corresponding pixel with level s_k in the output image, using Eq. (3.3-8). The transformation (mapping) $T(r_k)$ in this equation is called a *histogram equalization* or *histogram linearization* transformation. It is not difficult to show (Problem 3.10) that this transformation satisfies conditions (a) and (b) stated previously in this section.

EXAMPLE 3.5: A simple illustration of histogram equalization.

■ Before continuing, it will be helpful to work through a simple example. Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in Table 3.1, where the intensity levels are integers in the range $[0, L-1] = [0, 7]$.

The histogram of our hypothetical image is sketched in Fig. 3.19(a). Values of the histogram equalization transformation function are obtained using Eq. (3.3-8). For instance,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

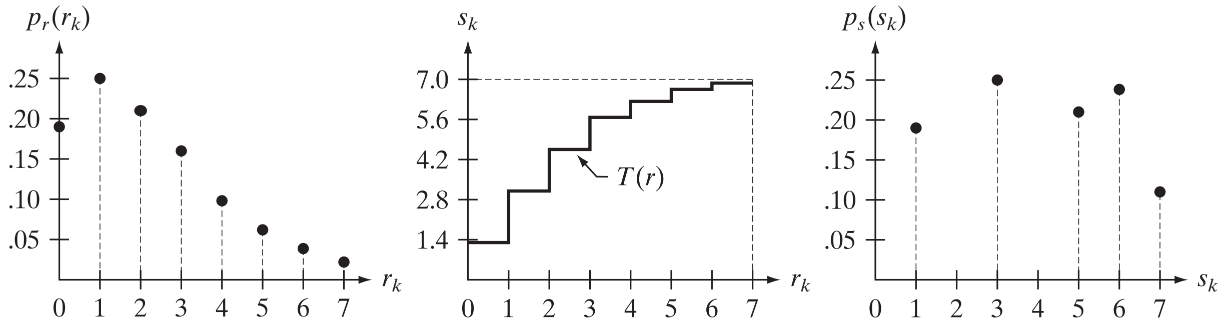
Similarly,

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

and $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, $s_7 = 7.00$. This transformation function has the staircase shape shown in Fig. 3.19(b).

TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

At this point, the s values still have fractions because they were generated by summing probability values, so we round them to the nearest integer:

$$
\begin{aligned}
s_0 &= 1.33 \rightarrow 1 & \qquad s_4 &= 6.23 \rightarrow 6 \\
s_1 &= 3.08 \rightarrow 3 & \qquad s_5 &= 6.65 \rightarrow 7 \\
s_2 &= 4.55 \rightarrow 5 & \qquad s_6 &= 6.86 \rightarrow 7 \\
s_3 &= 5.67 \rightarrow 6 & \qquad s_7 &= 7.00 \rightarrow 7
\end{aligned}
$$

These are the values of the equalized histogram. Observe that there are only five distinct intensity levels. Because $r_0 = 0$ was mapped to $s_0 = 1$, there are 790 pixels in the histogram equalized image with this value (see Table 3.1). Also, there are in this image 1023 pixels with a value of $s_1 = 3$ and 850 pixels with a value of $s_2 = 5$. However both r_3 and r_4 were mapped to the same value, 6, so there are $(656 + 329) = 985$ pixels in the equalized image with this value. Similarly, there are $(245 + 122 + 81) = 448$ pixels with a value of 7 in the histogram equalized image. Dividing these numbers by $MN = 4096$ yielded the equalized histogram in Fig. 3.19(c).

Because a histogram is an approximation to a PDF, and no new allowed intensity levels are created in the process, perfectly flat histograms are rare in practical applications of histogram equalization. Thus, unlike its continuous counterpart, it cannot be proved (in general) that discrete histogram equalization results in a uniform histogram. However, as you will see shortly, using Eq. (3.3-8) has the general tendency to spread the histogram of the input image so that the intensity levels of the equalized image span a wider range of the intensity scale. The net result is contrast enhancement. ■

We discussed earlier in this section the many advantages of having intensity values that cover the entire gray scale. In addition to producing intensities that have this tendency, the method just derived has the additional advantage that it is fully "automatic." In other words, given an image, the process of histogram equalization consists simply of implementing Eq. (3.3-8), which is based on information that can be extracted directly from the given image, without the need for further parameter specifications. We note also the simplicity of the computations required to implement the technique.

The *inverse transformation* from s back to r is denoted by

$$r_k = T^{-1}(s_k) \qquad k = 0, 1, 2, \ldots, L - 1 \tag{3.3-9}$$

It can be shown (Problem 3.10) that this inverse transformation satisfies conditions (a′) and (b) only if none of the levels, $r_k, k = 0, 1, 2, \ldots, L - 1$, are missing from the input image, which in turn means that none of the components of the image histogram are zero. Although the inverse transformation is not used in histogram equalization, it plays a central role in the histogram-matching scheme developed in the next section.

EXAMPLE 3.6: Histogram equalization.

■ The left column in Fig. 3.20 shows the four images from Fig. 3.16, and the center column shows the result of performing histogram equalization on each of these images. The first three results from top to bottom show significant improvement. As expected, histogram equalization did not have much effect on the fourth image because the intensities of this image already span the full intensity scale. Figure 3.21 shows the transformation functions used to generate the equalized images in Fig. 3.20. These functions were generated using Eq. (3.3-8). Observe that transformation (4) has a nearly linear shape, indicating that the inputs were mapped to nearly equal outputs.

The third column in Fig. 3.20 shows the histograms of the equalized images. It is of interest to note that, while all these histograms are different, the histogram-equalized images themselves are visually very similar. This is not unexpected because the basic difference between the images on the left column is one of contrast, not content. In other words, because the images have the same content, the increase in contrast resulting from histogram equalization was enough to render any intensity differences in the equalized images visually indistinguishable. Given the significant contrast differences between the original images, this example illustrates the power of histogram equalization as an adaptive contrast enhancement tool. ■

3.3.2 Histogram Matching (Specification)

As indicated in the preceding discussion, histogram equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram. When automatic enhancement is desired, this is a good approach because the results from this technique are predictable and the method is simple to implement. We show in this section that there are applications in which attempting to base enhancement on a uniform histogram is not the best approach. In particular, it is useful sometimes to be able to specify the shape of the histogram that we wish the processed image to have. The method used to generate a processed image that has a specified histogram is called *histogram matching* or *histogram specification*.

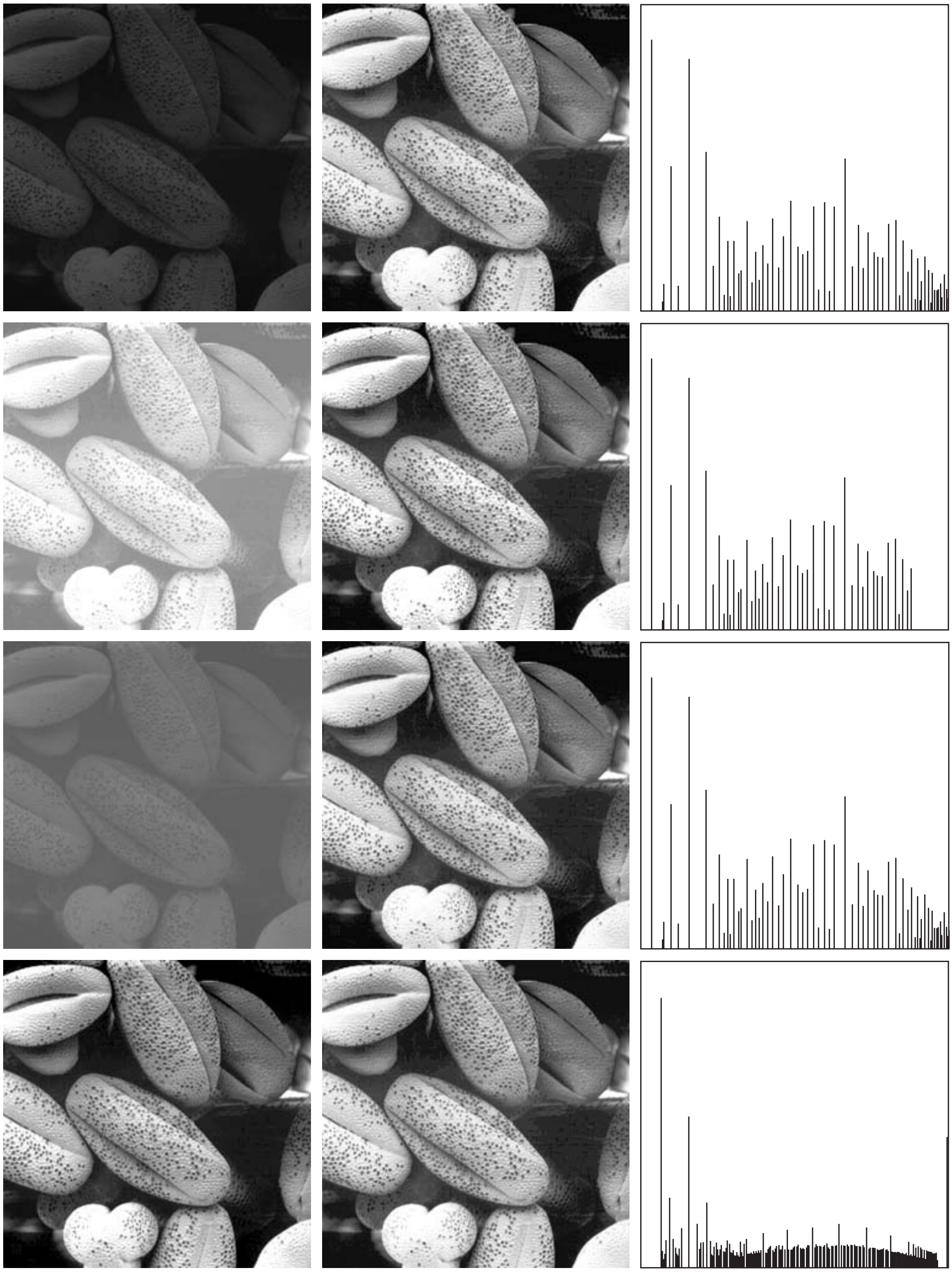


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

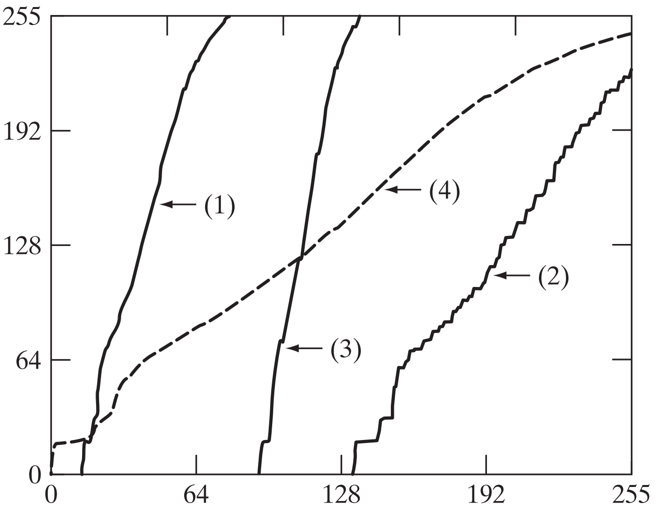


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

Let us return for a moment to continuous intensities r and z (considered continuous random variables), and let $p_r(r)$ and $p_z(z)$ denote their corresponding continuous probability density functions. In this notation, r and z denote the intensity levels of the input and output (processed) images, respectively. We can estimate $p_r(r)$ from the given input image, while $p_z(z)$ is the *specified* probability density function that we wish the output image to have.

Let s be a random variable with the property

$$s = T(r) = (L - 1)\int_0^r p_r(w)\,dw \tag{3.3-10}$$

where, as before, w is a dummy variable of integration. We recognize this expression as the continuous version of histogram equalization given in Eq. (3.3-4).

Suppose next that we define a random variable z with the property

$$G(z) = (L - 1)\int_0^z p_z(t)\,dt = s \tag{3.3-11}$$

where t is a dummy variable of integration. It then follows from these two equations that $G(z) = T(r)$ and, therefore, that z must satisfy the condition

$$z = G^{-1}[T(r)] = G^{-1}(s) \tag{3.3-12}$$

The transformation $T(r)$ can be obtained from Eq. (3.3-10) once $p_r(r)$ has been estimated from the input image. Similarly, the transformation function $G(z)$ can be obtained using Eq. (3.3-11) because $p_z(z)$ is given.

Equations (3.3-10) through (3.3-12) show that an image whose intensity levels have a specified probability density function can be obtained from a given image by using the following procedure:

1. Obtain $p_r(r)$ from the input image and use Eq. (3.3-10) to obtain the values of s.
2. Use the specified PDF in Eq. (3.3-11) to obtain the transformation function $G(z)$.

3. Obtain the inverse transformation $z = G^{-1}(s)$; because z is obtained from s, this process is a *mapping* from s to z, the latter being the desired values.
4. Obtain the output image by first equalizing the input image using Eq. (3.3-10); the pixel values in this image are the s values. For each pixel with value s in the equalized image, perform the inverse mapping $z = G^{-1}(s)$ to obtain the corresponding pixel in the output image. When all pixels have been thus processed, the PDF of the output image will be equal to the specified PDF.

EXAMPLE 3.7: Histogram specification.

■ Assuming continuous intensity values, suppose that an image has the intensity PDF $p_r(r) = 2r/(L-1)^2$ for $0 \le r \le (L-1)$ and $p_r(r) = 0$ for other values of r. Find the transformation function that will produce an image whose intensity PDF is $p_z(z) = 3z^2/(L-1)^3$ for $0 \le z \le (L-1)$ and $p_z(z) = 0$ for other values of z.

First, we find the histogram equalization transformation for the interval $[0, L-1]$:

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{(L-1)}\int_0^r w\,dw = \frac{r^2}{(L-1)}$$

By definition, this transformation is 0 for values outside the range $[0, L-1]$. Squaring the values of the input intensities and dividing them by $(L-1)^2$ will produce an image whose intensities, s, have a uniform PDF because this is a histogram-equalization transformation, as discussed earlier.

We are interested in an image with a specified histogram, so we find next

$$G(z) = (L-1)\int_0^z p_z(w)\,dw = \frac{3}{(L-1)^2}\int_0^z w^2\,dw = \frac{z^3}{(L-1)^2}$$

over the interval $[0, L-1]$; this function is 0 elsewhere by definition. Finally, we require that $G(z) = s$, but $G(z) = z^3/(L-1)^2$; so $z^3/(L-1)^2 = s$, and we have

$$z = \left[(L-1)^2 s\right]^{1/3}$$

So, if we multiply every histogram equalized pixel by $(L-1)^2$ and raise the product to the power $1/3$, the result will be an image whose intensities, z, have the PDF $p_z(z) = 3z^2/(L-1)^3$ in the interval $[0, L-1]$, as desired.

Because $s = r^2/(L-1)$ we can generate the z's directly from the intensities, r, of the input image:

$$z = \left[(L-1)^2 s\right]^{1/3} = \left[(L-1)^2\frac{r^2}{(L-1)}\right]^{1/3} = \left[(L-1)r^2\right]^{1/3}$$

Thus, squaring the value of each pixel in the original image, multiplying the result by $(L-1)$, and raising the product to the power $1/3$ will yield an image whose intensity levels, z, have the specified PDF. We see that the intermediate step of equalizing the input image can be skipped; all we need is to obtain the transformation function $T(r)$ that maps r to s. Then, the two steps can be combined into a single transformation from r to z. ■

As the preceding example shows, histogram specification is straightforward in principle. In practice, a common difficulty is finding meaningful analytical expressions for $T(r)$ and G^{-1}. Fortunately, the problem is simplified significantly when dealing with discrete quantities. The price paid is the same as for histogram equalization, where only an approximation to the desired histogram is achievable. In spite of this, however, some very useful results can be obtained, even with crude approximations.

The discrete formulation of Eq. (3.3-10) is the histogram equalization transformation in Eq. (3.3-8), which we repeat here for convenience:

$$\begin{aligned} s_k = T(r_k) &= (L-1)\sum_{j=0}^{k} p_r(r_j) \\ &= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L-1 \end{aligned} \tag{3.3-13}$$

where, as before, MN is the total number of pixels in the image, n_j is the number of pixels that have intensity value r_j, and L is the total number of possible intensity levels in the image. Similarly, given a specific value of s_k, the discrete formulation of Eq. (3.3-11) involves computing the transformation function

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \tag{3.3-14}$$

for a value of q, so that

$$G(z_q) = s_k \tag{3.3-15}$$

where $p_z(z_i)$, is the ith value of the specified histogram. As before, we find the desired value z_q by obtaining the inverse transformation:

$$z_q = G^{-1}(s_k) \tag{3.3-16}$$

In other words, this operation gives a value of z for each value of s; thus, it performs a *mapping* from s to z.

In practice, we do not need to compute the inverse of G. Because we deal with intensity levels that are integers (e.g., 0 to 255 for an 8-bit image), it is a simple matter to compute all the possible values of G using Eq. (3.3-14) for $q = 0, 1, 2, \ldots, L-1$. These values are scaled and rounded to their nearest integer values spanning the range $[0, L-1]$. The values are stored in a table. Then, given a particular value of s_k, we look for the closest match in the values stored in the table. If, for example, the 64th entry in the table is the closest to s_k, then $q = 63$ (recall that we start counting at 0) and z_{63} is the best solution to Eq. (3.3-15). Thus, the given value s_k would be associated with z_{63} (i.e., that specific value of s_k would *map* to z_{63}). Because the zs are intensities used as the basis for specifying the histogram $p_z(z)$, it follows that $z_0 = 0$, $z_1 = 1, \ldots, z_{L-1} = L - 1$, so z_{63} would have the intensity value 63. By repeating this procedure, we would find the mapping of each value of s_k to the value of z_q that is the closest solution to Eq. (3.3-15). These mappings are the solution to the histogram-specification problem.

Recalling that the s_ks are the values of the histogram-equalized image, we may summarize the histogram-specification procedure as follows:

1. Compute the histogram $p_r(r)$ of the given image, and use it to find the histogram equalization transformation in Eq. (3.3-13). Round the resulting values, s_k, to the integer range $[0, L - 1]$.
2. Compute all values of the transformation function G using the Eq. (3.3-14) for $q = 0, 1, 2, \ldots, L - 1$, where $p_z(z_i)$ are the values of the specified histogram. Round the values of G to integers in the range $[0, L - 1]$. Store the values of G in a table.
3. For every value of s_k, $k = 0, 1, 2, \ldots, L - 1$, use the stored values of G from step 2 to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k and store these mappings from s to z. When more than one value of z_q satisfies the given s_k (i.e., the mapping is not unique), choose the smallest value by convention.
4. Form the histogram-specified image by first histogram-equalizing the input image and then mapping every equalized pixel value, s_k, of this image to the corresponding value z_q in the histogram-specified image using the mappings found in step 3. As in the continuous case, the intermediate step of equalizing the input image is conceptual. It can be skipped by combining the two transformation functions, T and G^{-1}, as Example 3.8 shows.

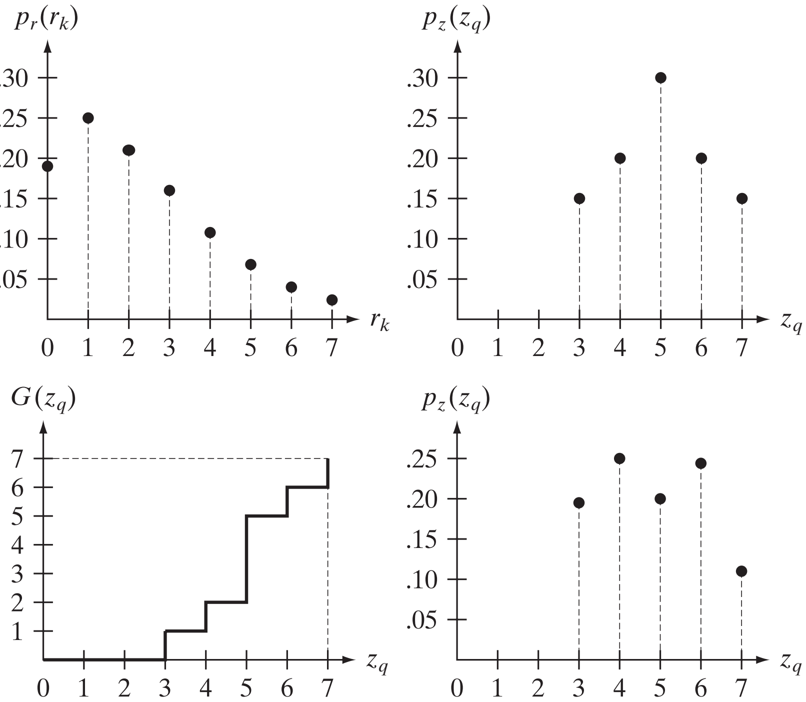
As mentioned earlier, for G^{-1} to satisfy conditions (a′) and (b), G has to be strictly monotonic, which, according to Eq. (3.3-14), means that none of the values $p_z(z_i)$ of the specified histogram can be zero (Problem 3.10). When working with discrete quantities, the fact that this condition may not be satisfied is not a serious implementation issue, as step 3 above indicates. The following example illustrates this numerically.

EXAMPLE 3.8: A simple example of histogram specification.

■ Consider again the 64×64 hypothetical image from Example 3.5, whose histogram is repeated in Fig. 3.22(a). It is desired to transform this histogram so that it will have the values specified in the second column of Table 3.2. Figure 3.22(b) shows a sketch of this histogram.

The first step in the procedure is to obtain the scaled histogram-equalized values, which we did in Example 3.5:

$$s_0 = 1 \quad s_2 = 5 \quad s_4 = 7 \quad s_6 = 7$$
$$s_1 = 3 \quad s_3 = 6 \quad s_5 = 7 \quad s_7 = 7$$



a b
c d

FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

In the next step, we compute all the values of the transformation function, G, using Eq. (3.3-14):

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0.00$$

Similarly,

$$G(z_1) = 7\sum_{j=0}^{1} p_z(z_j) = 7\big[p(z_0) + p(z_1)\big] = 0.00$$

and

$$G(z_2) = 0.00 \quad G(z_4) = 2.45 \quad G(z_6) = 5.95$$

$$G(z_3) = 1.05 \quad G(z_5) = 4.55 \quad G(z_7) = 7.00$$

TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

z_q	**Specified** $p_z(z_q)$	**Actual** $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

As in Example 3.5, these fractional values are converted to integers in our valid range, [0, 7]. The results are:

$$G(z_0) = 0.00 \rightarrow 0 \qquad G(z_4) = 2.45 \rightarrow 2$$
$$G(z_1) = 0.00 \rightarrow 0 \qquad G(z_5) = 4.55 \rightarrow 5$$
$$G(z_2) = 0.00 \rightarrow 0 \qquad G(z_6) = 5.95 \rightarrow 6$$
$$G(z_3) = 1.05 \rightarrow 1 \qquad G(z_7) = 7.00 \rightarrow 7$$

These results are summarized in Table 3.3, and the transformation function is sketched in Fig. 3.22(c). Observe that G is not strictly monotonic, so condition (a′) is violated. Therefore, we make use of the approach outlined in step 3 of the algorithm to handle this situation.

In the third step of the procedure, we find the smallest value of z_q so that the value $G(z_q)$ is the closest to s_k. We do this for every value of s_k to create the required mappings from s to z. For example, $s_0 = 1$, and we see that $G(z_3) = 1$, which is a perfect match in this case, so we have the correspondence $s_0 \rightarrow z_3$. That is, every pixel whose value is 1 in the histogram equalized image would map to a pixel valued 3 (in the corresponding location) in the histogram-specified image. Continuing in this manner, we arrive at the mappings in Table 3.4.

In the final step of the procedure, we use the mappings in Table 3.4 to map every pixel in the histogram equalized image into a corresponding pixel in the newly created histogram-specified image. The values of the resulting histogram are listed in the third column of Table 3.2, and the histogram is sketched in Fig. 3.22(d). The values of $p_z(z_q)$ were obtained using the same procedure as in Example 3.5. For instance, we see in Table 3.4 that $s = 1$ maps to $z = 3$, and there are 790 pixels in the histogram-equalized image with a value of 1. Therefore, $p_z(z_3) = 790/4096 = 0.19$.

Although the final result shown in Fig. 3.22(d) does not match the specified histogram exactly, the general trend of moving the intensities toward the high end of the intensity scale definitely was achieved. As mentioned earlier, obtaining the histogram-equalized image as an intermediate step is useful for explaining the procedure, but this is not necessary. Instead, we could list the mappings from the rs to the ss and from the ss to the zs in a three-column

TABLE 3.3
All possible values of the transformation function G scaled, rounded, and ordered with respect to z.

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

TABLE 3.4
Mappings of all the values of s_k into corresponding values of z_q.

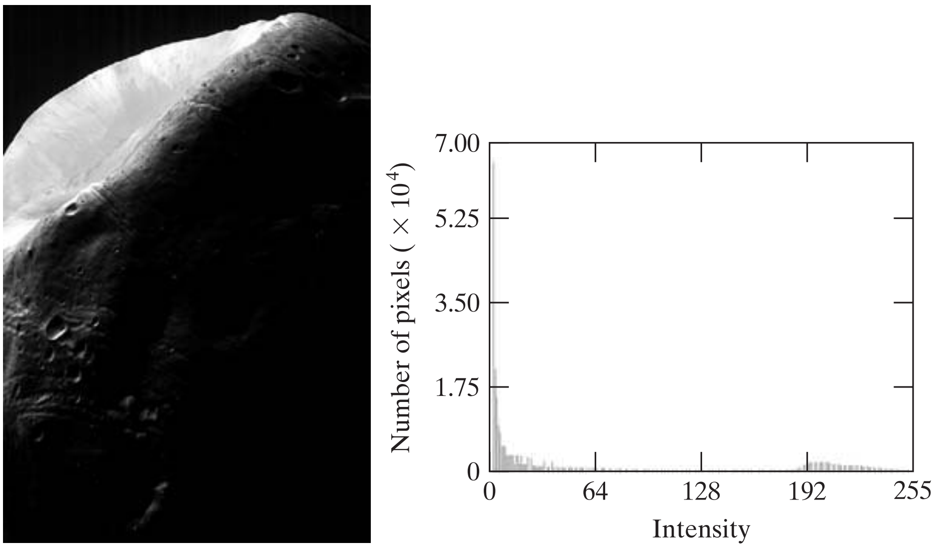
s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

table. Then, we would use those mappings to map the original pixels directly into the pixels of the histogram-specified image. ■

EXAMPLE 3.9: Comparison between histogram equalization and histogram matching.

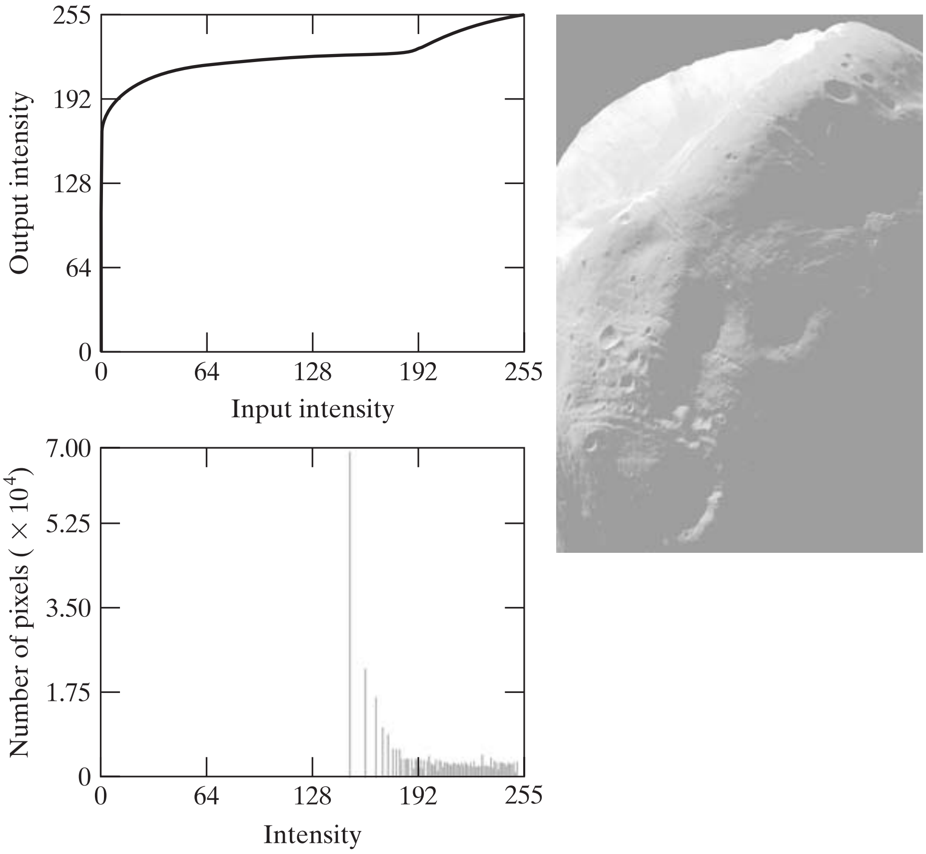
■ Figure 3.23(a) shows an image of the Mars moon, Phobos, taken by NASA's *Mars Global Surveyor*. Figure 3.23(b) shows the histogram of Fig. 3.23(a). The image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in the dark end of the gray scale. At first glance, one might conclude that histogram equalization would be a good approach to enhance this image, so that details in the dark areas become more visible. It is demonstrated in the following discussion that this is not so.

Figure 3.24(a) shows the histogram equalization transformation [Eq. (3.3-8) or (3.3-13)] obtained from the histogram in Fig. 3.23(b). The most relevant characteristic of this transformation function is how fast it rises from intensity level 0 to a level near 190. This is caused by the large concentration of pixels in the input histogram having levels near 0. When this transformation is applied to the levels of the input image to obtain a histogram-equalized result, the net effect is to map a very narrow interval of dark pixels into the upper end of the gray scale of the output image. Because numerous pixels in the input image have levels precisely in this interval, we would expect the result to be an image with a light, washed-out appearance. As Fig. 3.24(b) shows, this is indeed the



a b

FIGURE 3.23
(a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)

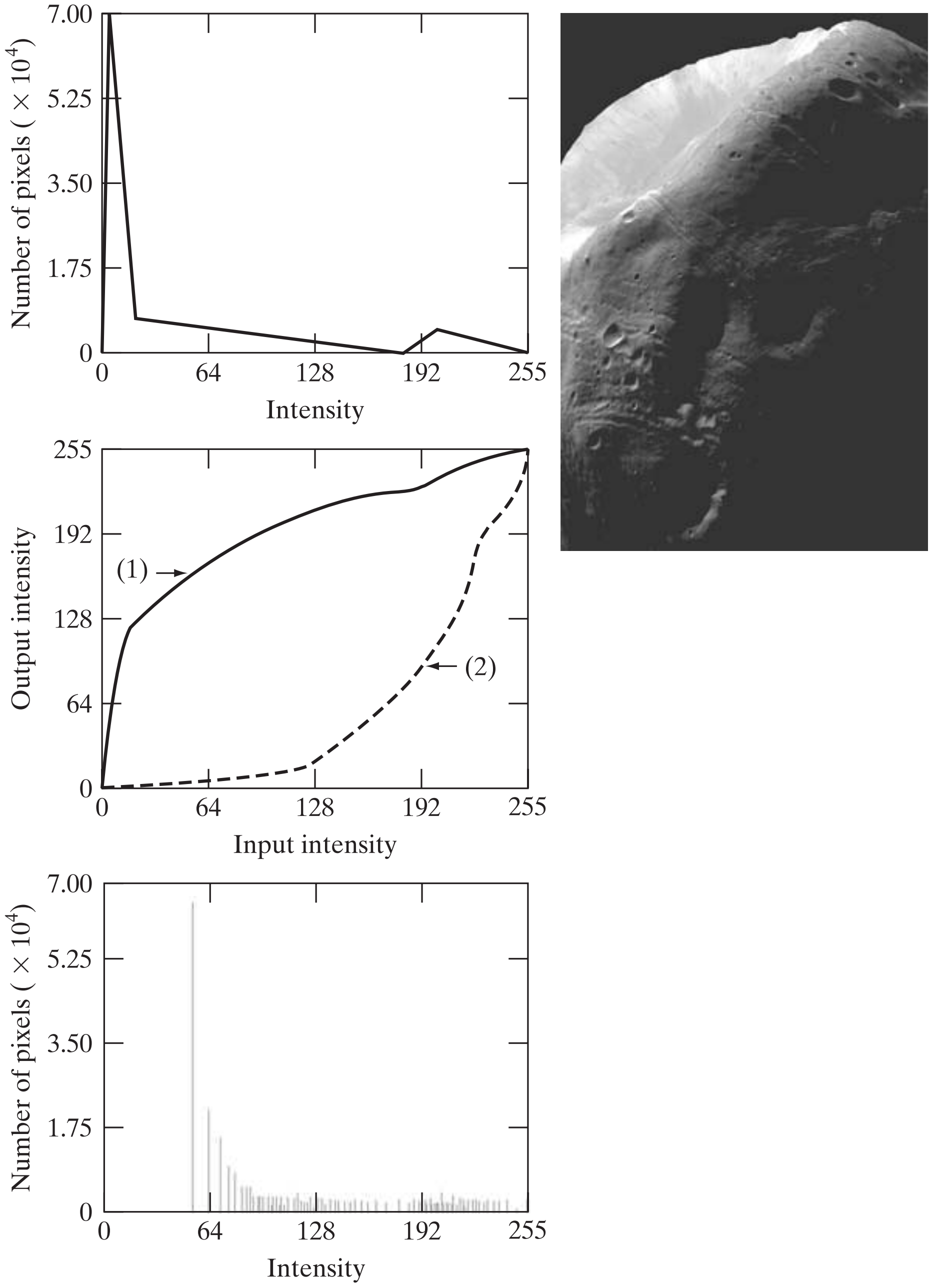


a b
c

FIGURE 3.24 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

case. The histogram of this image is shown in Fig. 3.24(c). Note how all the intensity levels are biased toward the upper one-half of the gray scale.

Because the problem with the transformation function in Fig. 3.24(a) was caused by a large concentration of pixels in the original image with levels near 0, a reasonable approach is to modify the histogram of that image so that it does not have this property. Figure 3.25(a) shows a *manually specified* function that preserves the general shape of the original histogram, but has a smoother transition of levels in the dark region of the gray scale. Sampling this function into 256 equally spaced discrete values produced the desired specified histogram. The transformation function $G(z)$ obtained from this histogram using Eq. (3.3-14) is labeled transformation (1) in Fig. 3.25(b). Similarly, the inverse transformation $G^{-1}(s)$ from Eq. (3.3-16) (obtained using the step-by-step procedure discussed earlier) is labeled transformation (2) in Fig. 3.25(b). The enhanced image in Fig. 3.25(c) was obtained by applying transformation (2) to the pixels of the histogram-equalized image in Fig. 3.24(b). The improvement of the histogram-specified image over the result obtained by histogram equalization is evident by comparing these two images. It is of interest to note that a rather modest change in the original histogram was all that was required to obtain a significant improvement in appearance. Figure 3.25(d) shows the histogram of Fig. 3.25(c). The most distinguishing feature of this histogram is how its low end has shifted right toward the lighter region of the gray scale (but not excessively so), as desired. ■



a c
b
d

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Although it probably is obvious by now, we emphasize before leaving this section that histogram specification is, for the most part, a trial-and-error process. One can use guidelines learned from the problem at hand, just as we did in the preceding example. At times, there may be cases in which it is possible to formulate what an "average" histogram should look like and use that as the specified histogram. In cases such as these, histogram specification becomes a straightforward process. In general, however, there are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

3.3.3 Local Histogram Processing

The histogram processing methods discussed in the previous two sections are *global*, in the sense that pixels are modified by a transformation function based on the intensity distribution of an entire image. Although this global approach is suitable for overall enhancement, there are cases in which it is necessary to enhance details over small areas in an image. The number of pixels in these areas may have negligible influence on the computation of a global transformation whose shape does not necessarily guarantee the desired local enhancement. The solution is to devise transformation functions based on the intensity distribution in a neighborhood of every pixel in the image.

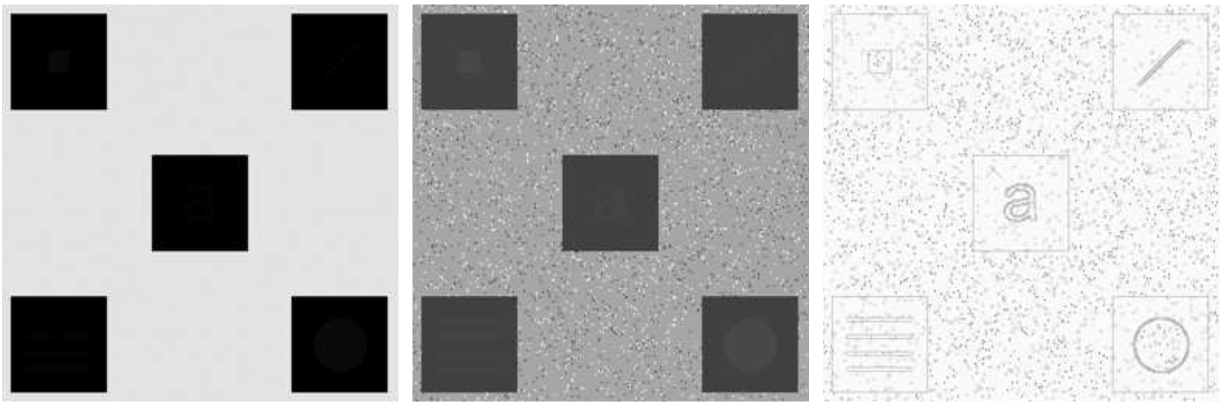
The histogram processing techniques previously described are easily adapted to local enhancement. The procedure is to define a neighborhood and move its center from pixel to pixel. At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is obtained. This function is then used to map the intensity of the pixel centered in the neighborhood. The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated. Because only one row or column of the neighborhood changes during a pixel-to-pixel translation of the neighborhood, updating the histogram obtained in the previous location with the new data introduced at each motion step is possible (Problem 3.12). This approach has obvious advantages over repeatedly computing the histogram of all pixels in the neighborhood region each time the region is moved one pixel location. Another approach used sometimes to reduce computation is to utilize nonoverlapping regions, but this method usually produces an undesirable "blocky" effect.

EXAMPLE 3.10: Local histogram equalization.

■ Figure 3.26(a) shows an 8-bit, 512×512 image that at first glance appears to contain five black squares on a gray background. The image is slightly noisy, but the noise is imperceptible. Figure 3.26(b) shows the result of global histogram equalization. As often is the case with histogram equalization of smooth, noisy regions, this image shows significant enhancement of the noise. Aside from the noise, however, Fig. 3.26(b) does not reveal any new significant details from the original, other than a very faint hint that the top left and bottom right squares contain an object. Figure 3.26(c) was obtained using local histogram equalization with a neighborhood of size 3×3. Here, we see significant detail contained within the dark squares. The intensity values of these objects were too close to the intensity of the large squares, and their sizes were too small, to influence global histogram equalization significantly enough to show this detail. ■

3.3.4 Using Histogram Statistics for Image Enhancement

Statistics obtained directly from an image histogram can be used for image enhancement. Let r denote a discrete random variable representing intensity values in the range $[0, L - 1]$, and let $p(r_i)$ denote the normalized histogram



a b c

FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

component corresponding to value r_i. As indicated previously, we may view $p(r_i)$ as an estimate of the probability that intensity r_i occurs in the image from which the histogram was obtained.

As we discussed in Section 2.6.8, the nth moment of r about its mean is defined as

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i) \tag{3.3-17}$$

We follow convention in using m for the mean value. Do not confuse it with the same symbol used to denote the number of rows in an $m \times n$ neighborhood, in which we also follow notational convention.

where m is the mean (average intensity) value of r (i.e., the average intensity of the pixels in the image):

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \tag{3.3-18}$$

The second moment is particularly important:

$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) \tag{3.3-19}$$

We recognize this expression as the intensity variance, normally denoted by σ^2 (recall that the standard deviation is the square root of the variance). Whereas the mean is a measure of average intensity, the variance (or standard deviation) is a measure of contrast in an image. Observe that all moments are computed easily using the preceding expressions once the histogram has been obtained from a given image.

When working with only the mean and variance, it is common practice to estimate them directly from the sample values, without computing the histogram. Appropriately, these estimates are called the *sample mean* and *sample variance*. They are given by the following familiar expressions from basic statistics:

$$m = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \tag{3.3-20}$$

and

$$\sigma^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x, y) - m\right]^2 \tag{3.3-21}$$

for $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$. In other words, as we know, the mean intensity of an image can be obtained simply by summing the values of all its pixels and dividing the sum by the total number of pixels in the image. A similar interpretation applies to Eq. (3.3-21). As we illustrate in the following example, the results obtained using these two equations are identical to the results obtained using Eqs. (3.3-18) and (3.3-19), provided that the histogram used in these equations is computed from the same image used in Eqs. (3.3-20) and (3.3-21).

The denominator in Eq. (3.3-21) is written sometimes as $MN - 1$ instead of MN. This is done to obtain a so-called *unbiased* estimate of the variance. However, we are more interested in Eqs. (3.3-21) and (3.3-19) agreeing when the histogram in the latter equation is computed from the same image used in Eq. (3.3-21). For this we require the MN term. The difference is negligible for any image of practical size.

EXAMPLE 3.11: Computing histogram statistics.

■ Before proceeding, it will be useful to work through a simple numerical example to fix ideas. Consider the following 2-bit image of size 5×5:

$$\begin{matrix} 0 & 0 & 1 & 1 & 2 \\ 1 & 2 & 3 & 0 & 1 \\ 3 & 3 & 2 & 2 & 0 \\ 2 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 2 & 2 \end{matrix}$$

The pixels are represented by 2 bits; therefore, $L = 4$ and the intensity levels are in the range $[0, 3]$. The total number of pixels is 25, so the histogram has the components

$$p(r_0) = \frac{6}{25} = 0.24; \; p(r_1) = \frac{7}{25} = 0.28;$$

$$p(r_2) = \frac{7}{25} = 0.28; \; p(r_3) = \frac{5}{25} = 0.20$$

where the numerator in $p(r_i)$ is the number of pixels in the image with intensity level r_i. We can compute the average value of the intensities in the image using Eq. (3.3-18):

$$\begin{aligned} m &= \sum_{i=0}^{3} r_i\, p(r_i) \\ &= (0)(0.24) + (1)(0.28) + (2)(0.28) + (3)(0.20) \\ &= 1.44 \end{aligned}$$

Letting $f(x, y)$ denote the preceding 5×5 array and using Eq. (3.3-20), we obtain

$$\begin{aligned} m &= \frac{1}{25}\sum_{x=0}^{4}\sum_{y=0}^{4} f(x, y) \\ &= 1.44 \end{aligned}$$

As expected, the results agree. Similarly, the result for the variance is the same (1.1264) using either Eq. (3.3-19) or (3.3-21). ■

We consider two uses of the mean and variance for enhancement purposes. The *global* mean and variance are computed over an entire image and are useful for gross adjustments in overall intensity and contrast. A more powerful use of these parameters is in local enhancement, where the *local* mean and variance are used as the basis for making changes that depend on image characteristics in a neighborhood about each pixel in an image.

Let (x, y) denote the coordinates of any pixel in a given image, and let S_{xy} denote a neighborhood (subimage) of specified size, centered on (x, y). The mean value of the pixels in this neighborhood is given by the expression

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i \, p_{S_{xy}}(r_i) \tag{3.3-22}$$

where $p_{S_{xy}}$ is the histogram of the pixels in region S_{xy}. This histogram has L components, corresponding to the L possible intensity values in the input image. However, many of the components are 0, depending on the size of S_{xy}. For example, if the neighborhood is of size 3×3 and $L = 256$, only between 1 and 9 of the 256 components of the histogram of the neighborhood will be nonzero. These non-zero values will correspond to the number of different intensities in S_{xy} (the maximum number of possible different intensities in a 3×3 region is 9, and the minimum is 1).

The variance of the pixels in the neighborhood similarly is given by

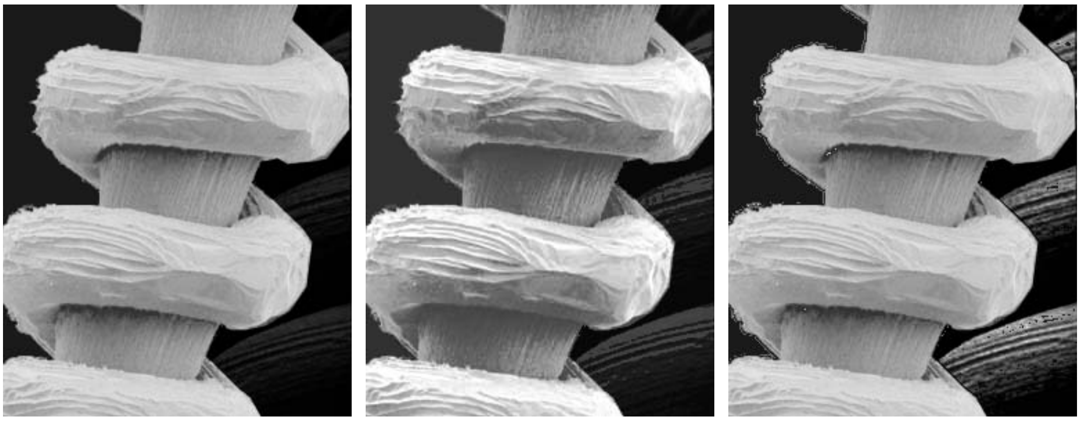
$$\sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 \, p_{S_{xy}}(r_i) \tag{3.3-23}$$

As before, the local mean is a measure of average intensity in neighborhood S_{xy}, and the local variance (or standard deviation) is a measure of intensity contrast in that neighborhood. Expressions analogous to (3.3-20) and (3.3-21) can be written for neighborhoods. We simply use the pixel values in the neighborhoods in the summations and the number of pixels in the neighborhood in the denominator.

As the following example illustrates, an important aspect of image processing using the local mean and variance is the flexibility they afford in developing simple, yet powerful enhancement techniques based on statistical measures that have a close, predictable correspondence with image appearance.

EXAMPLE 3.12: Local enhancement using histogram statistics.

■ Figure 3.27(a) shows an SEM (scanning electron microscope) image of a tungsten filament wrapped around a support. The filament in the center of the image and its support are quite clear and easy to study. There is another filament structure on the right, dark side of the image, but it is almost imperceptible, and its size and other characteristics certainly are not easily discernable. Local enhancement by contrast manipulation is an ideal approach to problems such as this, in which parts of an image may contain hidden features.



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

In this particular case, the problem is to enhance dark areas while leaving the light area as unchanged as possible because it does not require enhancement. We can use the concepts presented in this section to formulate an enhancement method that can tell the difference between dark and light and, at the same time, is capable of enhancing only the dark areas. A measure of whether an area is relatively light or dark at a point (x, y) is to compare the average local intensity, $m_{S_{xy}}$, to the average image intensity, called the *global mean* and denoted m_G. This quantity is obtained with Eq. (3.3-18) or (3.3-20) using the entire image. Thus, we have the first element of our enhancement scheme: We will consider the pixel at a point (x, y) as a candidate for processing if $m_{S_{xy}} \leq k_0 m_G$, where k_0 is a positive constant with value less than 1.0.

Because we are interested in enhancing areas that have low contrast, we also need a measure to determine whether the contrast of an area makes it a candidate for enhancement. We consider the pixel at a point (x, y) as a candidate for enhancement if $\sigma_{S_{xy}} \leq k_2 \sigma_G$, where σ_G is the *global standard deviation* obtained using Eqs. (3.3-19) or (3.3-21) and k_2 is a positive constant. The value of this constant will be greater than 1.0 if we are interested in enhancing light areas and less than 1.0 for dark areas.

Finally, we need to restrict the lowest values of contrast we are willing to accept; otherwise the procedure would attempt to enhance constant areas, whose standard deviation is zero. Thus, we also set a lower limit on the local standard deviation by requiring that $k_1 \sigma_G \leq \sigma_{S_{xy}}$, with $k_1 < k_2$. A pixel at (x, y) that meets all the conditions for local enhancement is processed simply by multiplying it by a specified constant, E, to increase (or decrease) the value of its intensity level relative to the rest of the image. Pixels that do not meet the enhancement conditions are not changed.

We summarize the preceding approach as follows. Let $f(x, y)$ represent the value of an image at any image coordinates (x, y), and let $g(x, y)$ represent the corresponding enhanced value at those coordinates. Then,

$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{S_{xy}} \leq k_0 m_G \text{ AND } k_1 \sigma_G \leq \sigma_{S_{xy}} \leq k_2 \sigma_G \\ f(x, y) & \text{otherwise} \end{cases} \tag{3.3-24}$$

for $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$, where, as indicated above, E, k_0, k_1, and k_2 are specified parameters, m_G is the global mean of the input image, and σ_G is its standard deviation. Parameters $m_{S_{xy}}$ and $\sigma_{S_{xy}}$ are the local mean and standard deviation, respectively. As usual, M and N are the row and column image dimensions.

Choosing the parameters in Eq. (3.3-24) generally requires a bit of experimentation to gain familiarity with a given image or class of images. In this case, the following values were selected: $E = 4.0$, $k_0 = 0.4$, $k_1 = 0.02$, and $k_2 = 0.4$. The relatively low value of 4.0 for E was chosen so that, when it was multiplied by the levels in the areas being enhanced (which are dark), the result would still tend toward the dark end of the scale, and thus preserve the general visual balance of the image. The value of k_0 was chosen as less than half the global mean because we can see by looking at the image that the areas that require enhancement definitely are dark enough to be below half the global mean. A similar analysis led to the choice of values for k_1 and k_2. Choosing these constants is not difficult in general, but their choice definitely must be guided by a logical analysis of the enhancement problem at hand. Finally, the size of the local area S_{xy} should be as small as possible in order to preserve detail and keep the computational burden as low as possible. We chose a region of size 3×3.

As a basis for comparison, we enhanced the image using global histogram equalization. Figure 3.27(b) shows the result. The dark area was improved but details still are difficult to discern, and the light areas were changed, something we did not want to do. Figure 3.27(c) shows the result of using the local statistics method explained above. In comparing this image with the original in Fig. 3.27(a) or the histogram equalized result in Fig. 3.27(b), we note the obvious detail that has been brought out on the right side of Fig. 3.27(c). Observe, for example, the clarity of the ridges in the dark filaments. It is noteworthy that the light-intensity areas on the left were left nearly intact, which was one of our initial objectives. ■

3.4 Fundamentals of Spatial Filtering

In this section, we introduce several basic concepts underlying the use of spatial filters for image processing. Spatial filtering is one of the principal tools used in this field for a broad spectrum of applications, so it is highly advisable that you develop a solid understanding of these concepts. As mentioned at the beginning of this chapter, the examples in this section deal mostly with the use of spatial filters for image enhancement. Other applications of spatial filtering are discussed in later chapters.