Histogram Matching (Specification)

As previously discussed, histogram equalization can automatically determine 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 since the results from this technique are predictable and the method is simple to implement.

However, 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.

Let r and z denote the intensity levels of input and output images, and $p_r(r)$ and $p_z(z)$ denote their corresponding continuous probability density functions. 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

$$s = T(r) = (L-1)\int_0^r p_r(\omega)d\omega, \qquad (3.3\text{-}10)$$

which is the continuous version of histogram equalization given in (3.3-4). Then, we define a random variable z with the property

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

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

$$z = G^{-1}\left[T(r)\right] = G^{-1}(s). \qquad (3.3\text{-}12)$$

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 (3.3-10) to obtain the values of s.
2. Use the specified PDF in (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.
4. Obtain the output image by first equalizing the input image using (3.3-10). The pixel values in this image are the s values.

   For each pixel with values 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 processed, the PDF of the output image will be equal to the specified PDF.

Example 3.7: Histogram specification

Suppose that an image has the intensity probability density function

$$p_r(r) = \frac{2r}{(L-1)^2}$$

for $0 \le r \le (L-1)$ and $p_r(r) = 0$ for other values of r.

We want to find the transformation function that will produce an image whose intensity PDF is

$$p_z(z) = \frac{3z^2}{(L-1)^3}$$

for $0 \le z \le (L-1)$ and $p_z(z) = 0$ for other values of z.

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

$$s = T(r) = (L-1)\int_0^r p_r(\omega)d\omega = \frac{2}{(L-1)}\int_0^r \omega d\omega = \frac{r^2}{(L-1)}$$

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

$$G(z) = (L-1)\int_0^z p_z(\omega)d\omega = \frac{3}{(L-1)^2}\int_0^z \omega^2 d\omega = \frac{z^3}{(L-1)^2}$$

Then, we require that $G(z) = s = z^3/(L-1)^2$, we can generate the z directly from the intensities, r, of the input image:

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

As Example 3.7 shows, histogram specification is straightforward in principle. In practice, however, a common difficulty is to find meaningful expressions for $T(r)$ and G^{-1}. This problem will be simplified significantly when dealing with discrete quantities.

The discrete formulation of

$$s = T(r) = (L-1)\int_0^r p_r(\omega)d\omega \qquad (3.3\text{-}10)$$

is the histogram equalization transformation in

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

Given a specific value of s_k, the discrete formulation of

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

involves computing the transformation function

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

for a value of q, so that

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

where $p_z(z_i)$ is the ith value of the specified histogram.

We can find the desired value z_q by obtaining the inverse transformation:

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

though we do not need to compute G^{-1} in practice.

Recalling that the s_k s are the values of histogram-equalized image, we can 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 (3.3-13). Then, round the resulting values, s_k, to the integer range $[1, L-1]$.

2. Compute all values of the transformation function G using (3.3-14) for $q = 0,1,2,..,L-1$, where $p_z(z_i)$ are the values of the specified histogram. Round the values of G to integers and store the values of G in a table.

3. For every value of s_k, $k = 0,1,2,..,L-1$, use the stored values of G 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. If there are more than one value of z_q satisfying the given s_k, choose the smallest value.

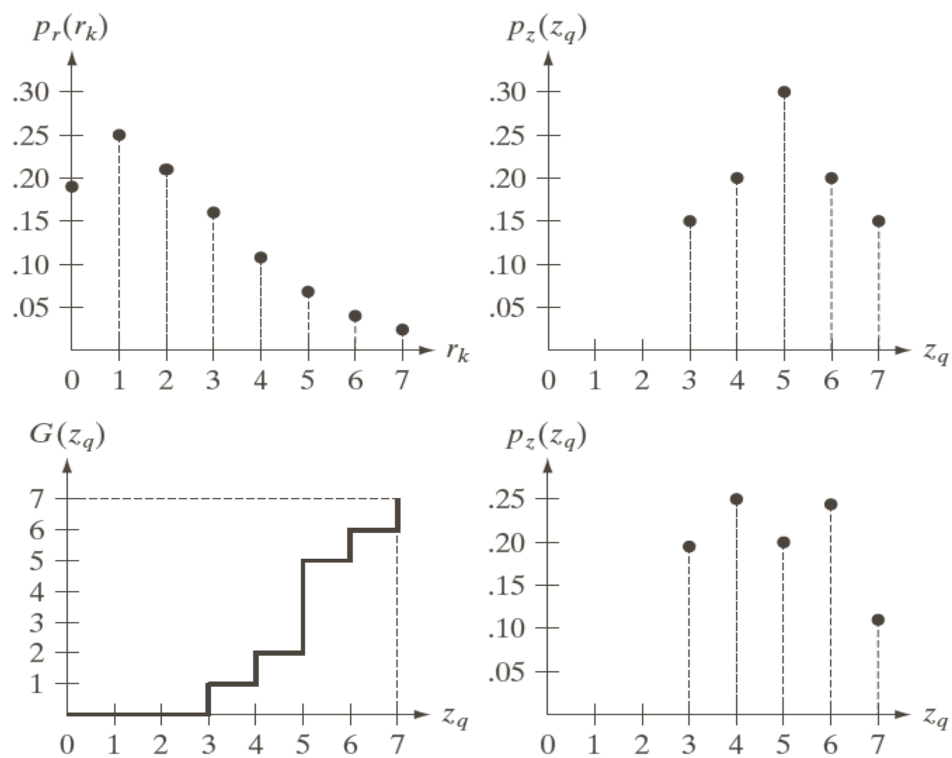
4. Form the histogram-specified image by histogram-equalizing the input image and then mapping every pixel value, s_k, of this image to the corresponding value z_q in the histogram-specified image using the mapping found in Step 3.

For G^{-1} to satisfy conditions (a') and (b), G has to be strictly monotonic, which means that none of the values $p_z(z_i)$ of the specified histogram can be zero.

When working with discrete quantities, the above condition may not be satisfied is not a serious implementation issue.

Example 3.8: A simple example of histogram specification

Consider the 64×64 hypothetical image from Example 3.5, whose histogram is repeated in Figure 3.22 (a).



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).

We transform this histogram in order to have the values specified in the second column of Table 3.2.

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

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).

The first step is to obtain the scaled histogram-equalized values, which was done in Example 3.5:

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

Then, by using (3.3-14), we compute all the values of the transformation function G:

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

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

and

$$G(z_2) = 0.00 \quad G(z_4) = 2.45 \quad G(z_6) = 5.95$$
$$G(z_3) = 1.05 \quad G(z_5) = 4.55 \quad G(z_7) = 7.00$$

To convert these fractional values to integers in our valid range $[1, L-1]$:

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

These results are summarized in Table 3.3, and the transformation function is sketched in Figure 3.22 (c).

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

Observe that G is not strictly monotonic, so condition (a') is violated. Therefore, we use the approach outlined in Step 3 of the algorithm to handle this situation.

We need to 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 $G(z_3) = 1$, so we have the correspondence $s_0 \to z_3$, which means that every pixel whose value is 1 in the histogram equalized image would map to a pixel valued 3 in the histogram-specified image. Table 3.4 shows the completed mappings with this manner:

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

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

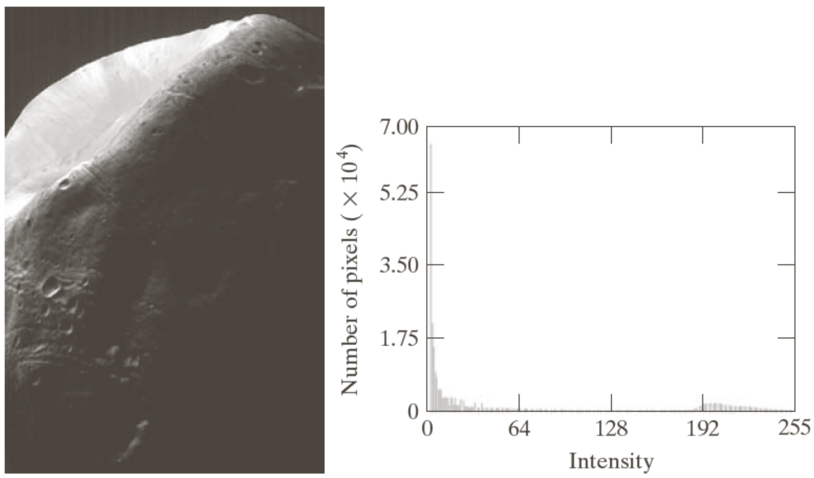
In the final step of the procedure, we use the mappings in Table 3.4 to map every pixel in the histogram equalized image into a corresponding pixel in the newly created histogram-specified image.

The values of the resulting histogram are listed in the third column of Table 3.2, and the histogram is sketched in Figure 3.22 (d).

Although the final result shown in Figure 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.

Example 3.9: Comparison between histogram equalization and histogram matching

Figure 3.23 (a) shows an image of the Mars moon, Phobos, and Figure 3.23 (b) shows the histogram of Figure 3.23 (a).



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.)

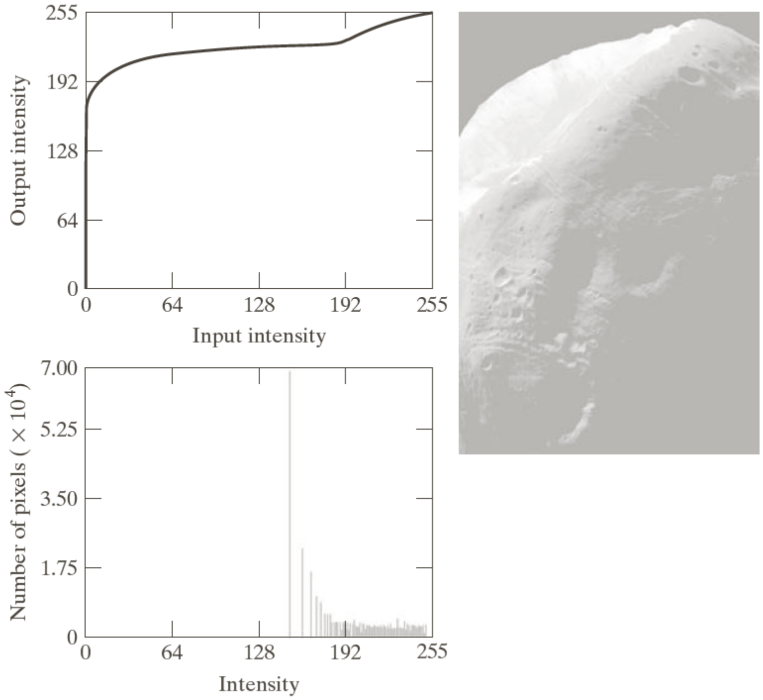
The image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in the dark end of the gray scale.

Would histogram equalization be a good approach to enhance this image, so that details in the dark areas become more visible?

Figure 3.24 (a) shows the histogram equalization transformation obtained from the histogram in Figure 3.23 (b), which rises sharply from intensity level 0 to 190.

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.

Figure 3.24 (b) shows the histogram-equalized image. Figure 3.24 (c) shows the histogram of Figure 3.24 (b).

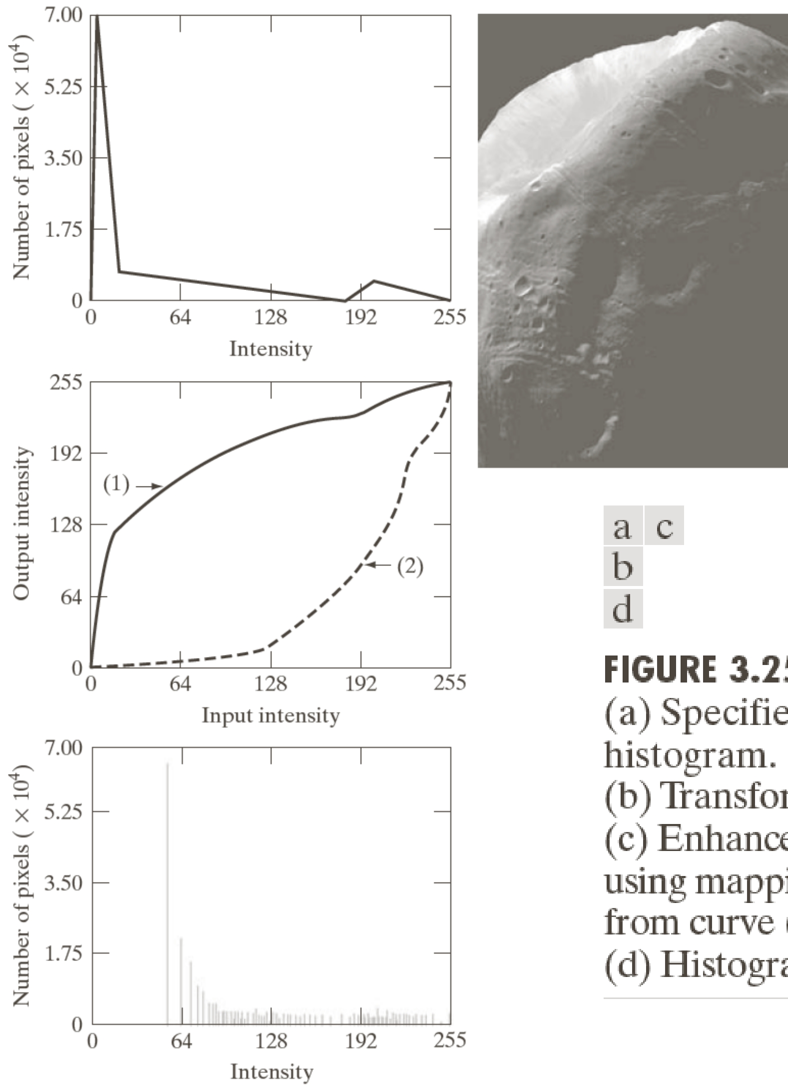


a b
c

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

The problem with the transformation function in Figure 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.



a c
b
d

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

The transformation function $G(z)$ obtained from the histogram shown in Figure 3.25 (a) using

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

is labelled transformation (1) in Figure 3.25 (b). The inverse transformation function $G^{-1}(s)$ obtained from

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

is labelled transformation (2) in Figure 3.25 (b).

The enhanced image shown in Figure 3.25 (c) was obtained by applying transformation (2) to the pixels of the histogram equalized image in Figure 3.24 (b).

Figure 3.25 (d) shows the histogram of Figure 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.

Compare Figure 3.25 (c) and Figure 3.24 (c), the improvement is obvious.

In general, there are no rules for specifying histograms, and one must resort to analysis on a case-by-case basis for any given enhancement task.

Local Histogram Processing

The histogram processing methods discussed previously are global, in the sense that pixels are modified by a transformation function based on the intensity distribution of an entire image.

In general, the global approach is suitable for overall enhancement.

Sometimes, we may want to enhance details over small areas in an image, in which pixels may have negligible influence on the computation of a global transformation.

The solution is to devise transformation functions based on the intensity distribution in a neighbourhood of every pixel in the image.

The histogram processing techniques previously described can be easily adapted to local enhancement.

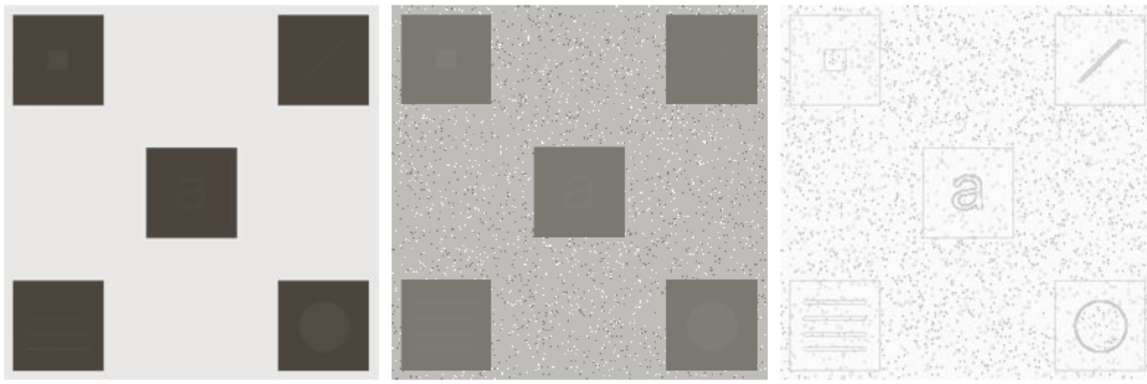
The procedure is to define a neighbourhood and move its center from pixel to pixel. At each location, the histogram of the points in the neighbourhood is computed and either a histogram equalization or histogram specification transformation function is obtained.

The center of the neighbourhood region is then moved to an adjacent pixel location and the procedure is repeated.

Since only one row or column of the neighbourhood changes during a pixel-to-pixel translation of the neighbourhood, updating the histogram obtained in the previous location with the new data introduced at each motion step is possible.

Example 3.10: Local histogram equalization

Figure 3.26 (a) shows an 8-bit, 512×512 image, which is slightly noisy.



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.

Figure 3.26 (b) shows the result of global histogram equalization. As often is the case with histogram equalization of smooth, noisy regions, Figure 3.26 (b) shows significant enhancement of the noise.

Figure 3.26 (c) was obtained using local histogram equalization with a neighbourhood of size 3×3.

Figure 3.26 (c) shows the significant detail contained within the dark squares.

Since the intensity values of the objects within small squares were too close to the intensity of the large squares, and their size were too small, to influence global histogram equalization significantly to show this detail.

Using Histogram Statistics for Image Enhancement

3.4 Fundamental of Spatial Filtering

Spatial filtering is one of the principal tools used in image processing for a broad spectrum of applications.

The Mechanics of Spatial Filtering

As explained briefly in Figure 3.1, a spatial filter consists of a neighbourhood and a predefined operation that is performed on the image pixels encompassed by the neighbourhood.

Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighbourhood, and whose value is the result of the filtering operation.

If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. Otherwise, the filter is nonlinear.

Figure 3.28 shows the mechanics of linear spatial filtering using a 3×3 neighbourhood.

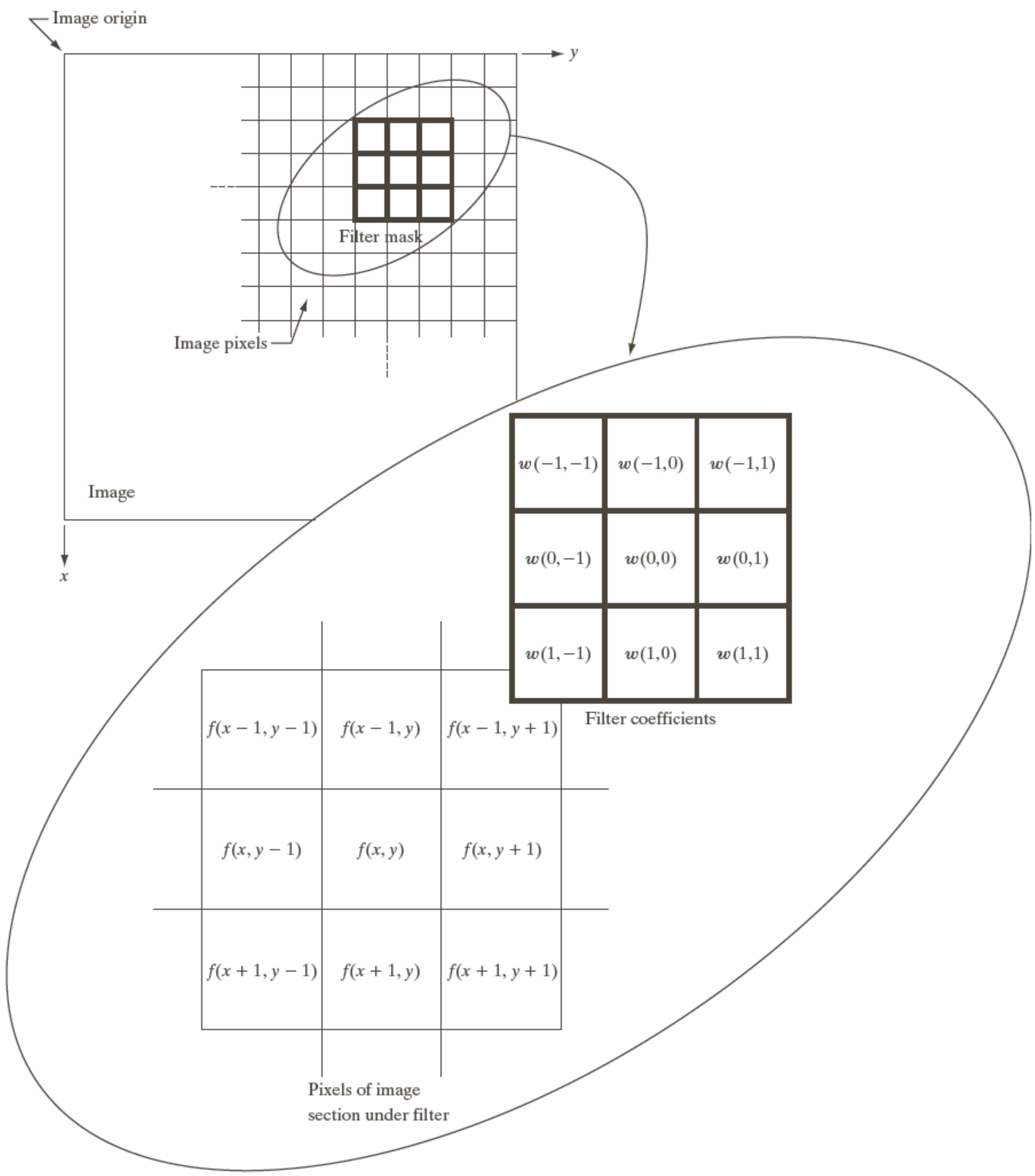


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

At any point (x, y) in Figure 3.28, the response, $g(x, y)$, of the filter is the sum of products of the filter coefficients and the image pixels encompassed by the filter:

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots$$
$$+ w(0,0)f(x,y) + \ldots + w(1,1)f(x+1,y+1)$$

In general, linear spatial filtering of an image of size $M \times N$ with a filter of size $m \times n$ is given by the expression:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

where x and y are varied so that each pixel in w visits every pixel in f.

Spatial Correlation and Convolution

There are two closely related concepts when performing linear spatial filtering, correlation and convolution.

Correlation is the process of moving a filter mask over the image and computing the sum of products at each location.

A 1-D example of correlation and convolution: Figure 3.29 (a) – (h).

The concept of convolution is a cornerstone of linear system theory. A fundamental property of convolution is that convolving a function with a unit impulse yields a copy of the function at the location of the impulse.

To perform convolution, all we do is to rotate one function by 180° and perform the same operation as in correlation, as the right column in Figure 3.29 shows.