Erosion

h A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, is defined as

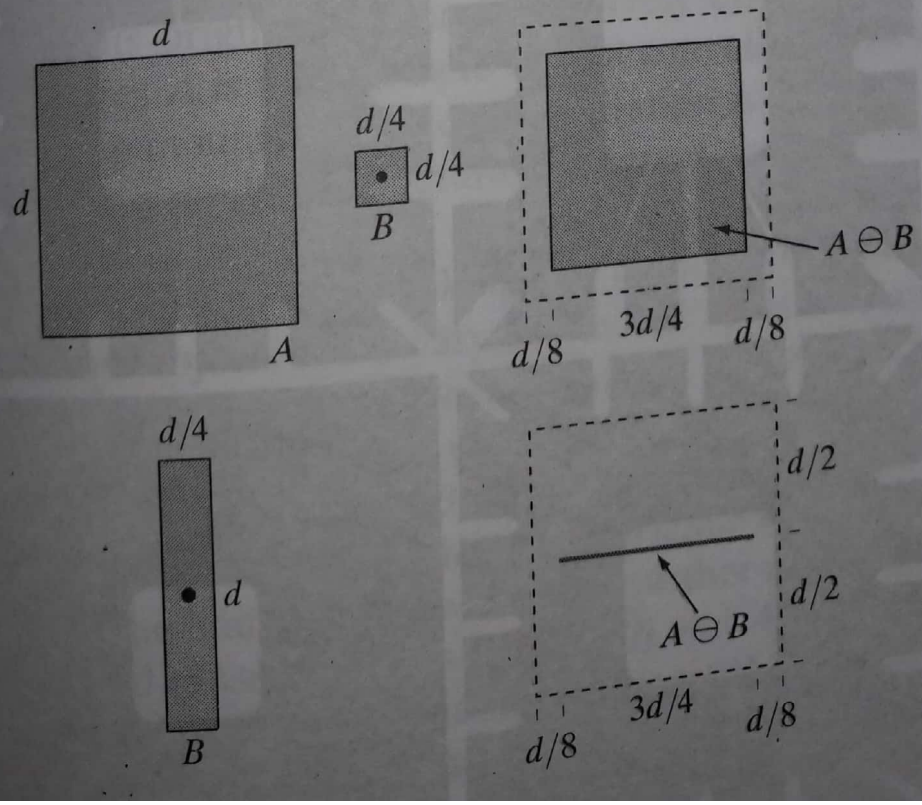
$$A \ominus B = \{z | (B)_z \subseteq A\} \tag{9.2-1}$$

words, this equation indicates that the erosion of A by B is the set of all nts z such that B, translated by z, is contained in A. In the following discus- n, set B is assumed to be a structuring element. Equation (9.2-1) is the thematical formulation of the example in Fig. 9.3(e), discussed at the end of last section. Because the statement that B has to be contained in A is uivalent to B not sharing any common elements with the background, we express erosion in the following equivalent form:

$$A \ominus B = \{z | (B)_z \cap A^c = \varnothing\} \tag{9.2-2}$$

here, as defined in Section 2.6.4, A^c is the complement of A and $\varnothing$ is the mpty set.

Figure 9.4 shows an example of erosion. The elements of A and B are own shaded and the background is white. The solid boundary in Fig. 9.4(c) the limit beyond which further displacements of the origin of B would use the structuring element to cease being completely contained in A. hus, the locus of points (locations of the origin of B) within (and includ- g) this boundary, constitutes the erosion of A by B. We show the erosion haded in Fig. 9.4(c). Keep in mind that that erosion is simply the *set* of



a b c
d

We see from this example that erosion shrinks or thins objects in a binary image. In fact, we can view erosion as a *morphological filtering* operation in which image details smaller than the structuring element are filtered (removed) from the image. In Fig. 9.5, erosion performed the function of a "filter." We return to the concept of a morphological filter in Sections 9.3 and 9.6.3. ■

9.2.2 Dilation

With A and B as sets in Z^2, the *dilation* of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{ z | (\hat{B})_z \cap A \neq \varnothing \right\} \tag{9.2-3}$$

This equation is based on reflecting B about its origin, and shifting this reflection by z (see Fig. 9.1). The dilation of A by B then is the set of all displacements, z, such that $\hat{B}$ and A overlap by at least one element. Based on this interpretation, Eq. (9.2-3) can be written equivalently as

$$A \oplus B = \left\{ z | [(\hat{B})_z \cap A] \subseteq A \right\} \tag{9.2-4}$$

As before, we assume that B is a structuring element and A is the set (image objects) to be dilated.

Equations (9.2-3) and (9.2-4) are not the only definitions of dilation currently in use (see Problems 9.11 and 9.12 for two different, yet equivalent, definitions). However, the preceding definitions have a distinct advantage over other formulations in that they are more intuitive when the structuring element B is viewed as a convolution mask. The basic process of flipping (rotating) B about its origin and then successively displacing it so that it slides over set (image) A is analogous to spatial convolution, as introduced in Section 3.4.2. Keep in mind, however, that dilation is based on set operations and therefore is a nonlinear operation, whereas convolution is a linear operation.

Unlike erosion, which is a shrinking or thinning operation, dilation "grows" or "thickens" objects in a binary image. The specific manner and extent of this thickening is controlled by the shape of the structuring element used. Figure 9.6(a) shows the same set used in Fig. 9.4, and Fig. 9.6(b) shows a structuring element (in this case $\hat{B} = B$ because the SE is symmetric about its origin). The dashed line in Fig. 9.6(c) shows the original set for reference, and the solid line shows the limit beyond which any further displacements of the origin of $\hat{B}$ by z would cause the intersection of $\hat{B}$ and A to be empty. Therefore, all points on and inside this boundary constitute the dilation of A by B. Figure 9.6(d) shows a structuring element designed to achieve more dilation vertically than horizontally, and Fig. 9.6(e) shows the dilation achieved with this element.

The *opening* of set A by structuring element B, denoted A fined as

$$A \circ B = (A \ominus B) \oplus B$$

Thus, the opening A by B is the erosion of A by B, followed by a the result by B.

Similarly, the *closing* of set A by structuring element B, denoted defined as

$$A \bullet B = (A \oplus B) \ominus B$$

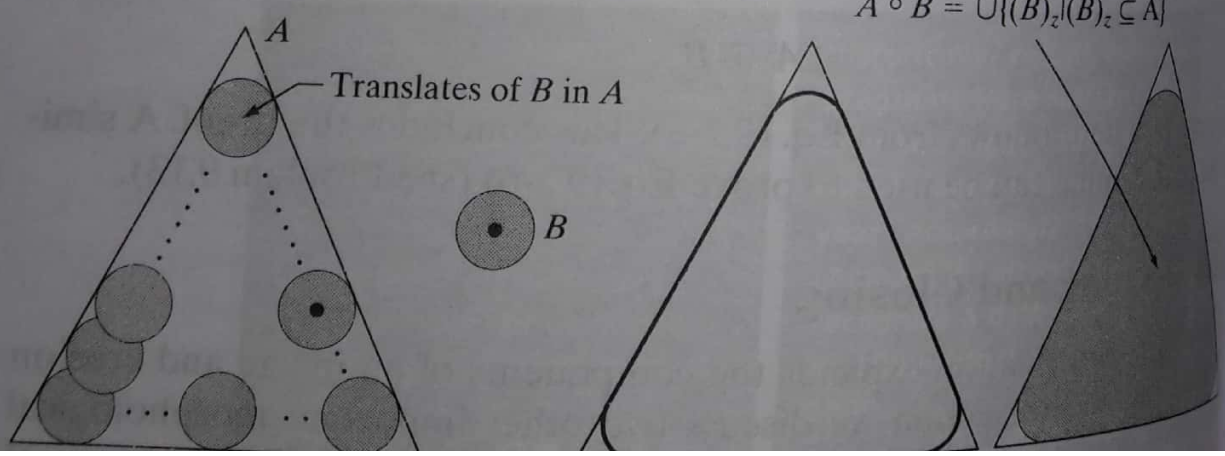
which says that the closing of A by B is simply the dilation of A by B, f by the erosion of the result by B.

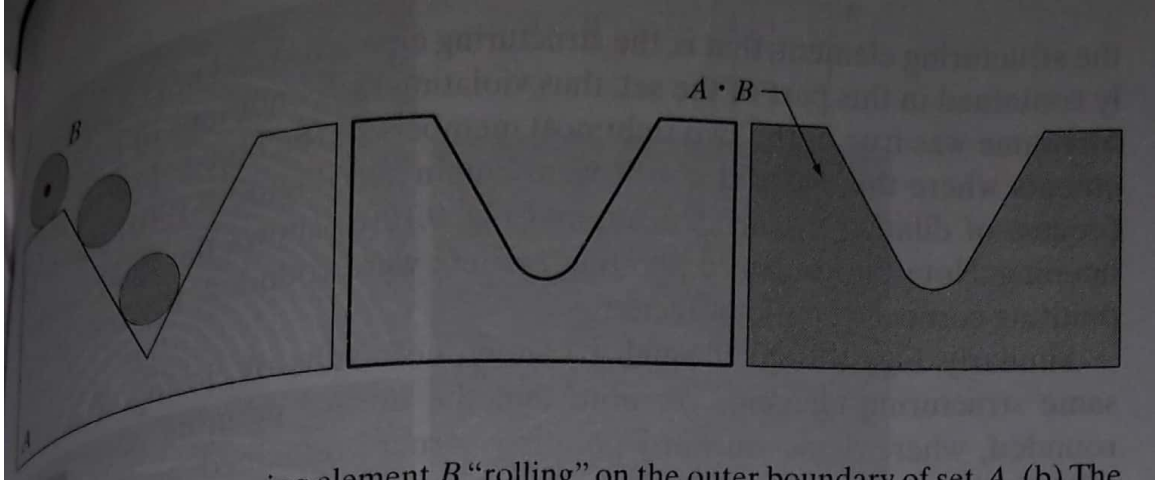
The opening operation has a simple geometric interpretation (Fi Suppose that we view the structuring element B as a (flat) "rolling ball *boundary* of $A \circ B$ is then established by the points in B that reac *farthest* into the boundary of A as B is rolled around the *inside* of this bo ary. This geometric *fitting* property of the opening operation leads to a theoretic formulation, which states that the opening of A by B is obtaine taking the union of all translates of B that fit into A. That is, opening can be pressed as a fitting process such that

$$A \circ B = \bigcup \{(B)_z | (B)_z \subseteq A\} \qquad (9.3$$

where $\cup\{\cdot\}$ denotes the union of all the sets inside the braces.

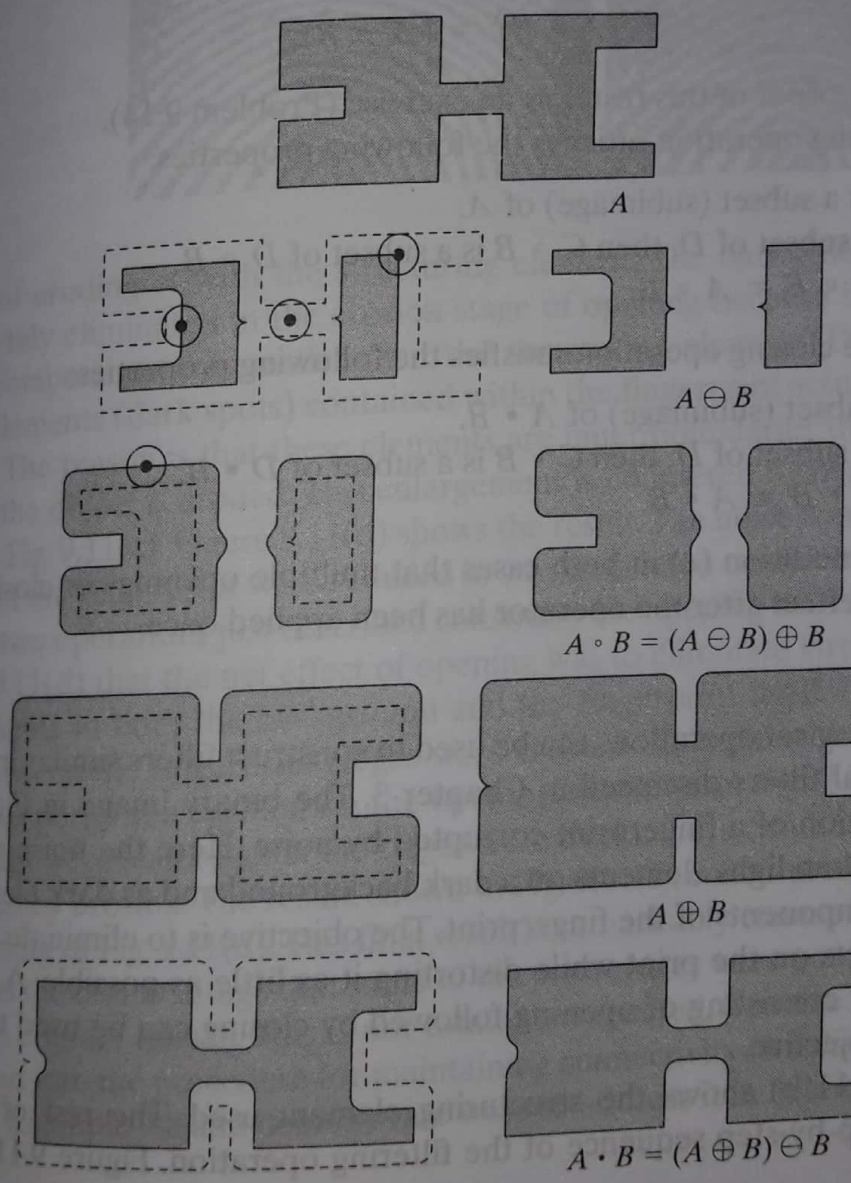
Closing has a similar geometric interpretation, except that now we roll B the outside of the boundary (Fig. 9.9). As discussed below, opening and closing are duals of each other, so having to roll the ball on the outside is not unex pected. Geometrically, a point w is an element of $A \bullet B$ if and only $(B)_z \cap A \neq \varnothing$ for any translate of $(B)_z$ that contains w. Figure 9.9 illustrates the basic geometrical properties of closing.





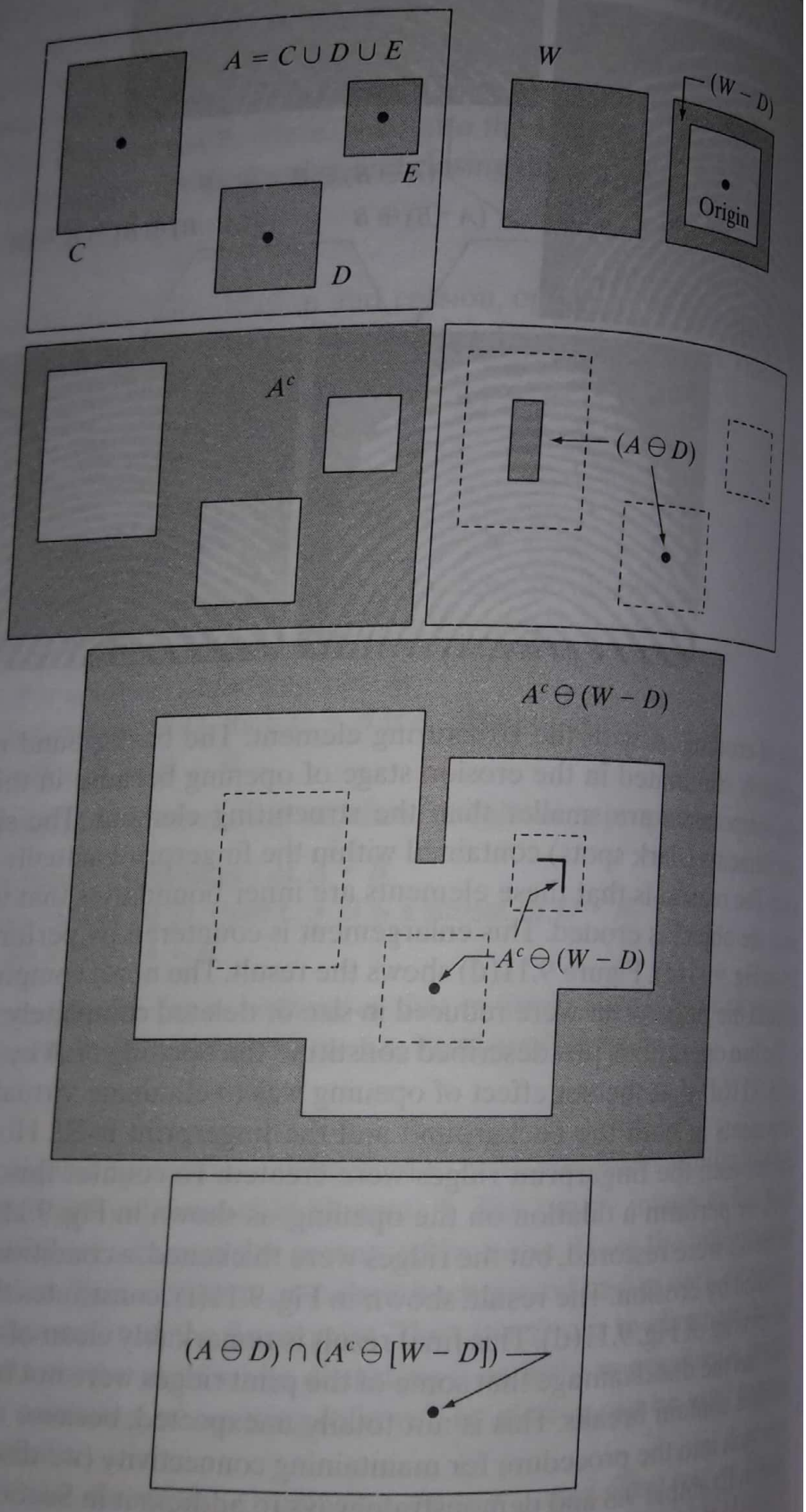
a b c
FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

Figure 9.10 further illustrates the opening and closing operations. Figure 9.10(a) shows a set A, and Fig. 9.10(b) shows various positions of a disk structuring element during the erosion process. When completed, this process resulted in the disjoint figure in Fig. 9.10(c). Note the elimination of the bridge between the two main sections. Its width was thin in relation to the diameter of



9.4 The Hit-or-Miss Transformation

The morphological hit-or-miss transform is a basic tool for shape de
We introduce this concept with the aid of Fig. 9.12, which shows a set
sisting of three shapes (subsets), denoted C, D, and E. The shading in Figs
through (c) indicates the original sets, whereas the shading in Figs. 9.12
(e) indicates the result of morphological operations. The objective is
the location of one of the shapes, say, D.



Let the origin of each [illegible] located at its center of gravity. Let D be enclosed by a small window, W. The *local background* of D with respect to W is defined as the set difference $(W - D)$, as shown in Fig. 9.12(b). Figure 9.12(c) shows the complement of A, which is needed later. Figure 9.12(d) shows the erosion of A by D (the dashed lines are included for reference). Recall that the erosion of A by D is the set of locations of the *origin* of D, such that D is completely contained in A. Interpreted another way, $A \ominus D$ may be viewed geometrically as the set of all locations of the origin of D at which D found a match (hit) in A. Keep in mind that in Fig. 9.12 A consists only of the three *disjoint* sets C, D, and E.

Figure 9.12(e) shows the erosion of the complement of A by the local background set $(W - D)$. The outer shaded region in Fig. 9.12(e) is part of the erosion. We note from Figs. 9.12(d) and (e) that the set of locations for which D *exactly* fits inside A is the *intersection* of the erosion of A by D and the erosion of A^c by $(W - D)$ as shown in Fig. 9.12(f). This intersection is precisely the location sought. In other words, if B denotes the set composed of D and its background, the match (or set of matches) of B in A, denoted $A \circledast B$, is

$$A \circledast B = (A \ominus D) \cap \left[A^c \ominus (W - D)\right] \tag{9.4-1}$$

We can generalize the notation somewhat by letting $B = (B_1, B_2)$, where B_1 is the set formed from elements of B associated with an object and B_2 is the set of elements of B associated with the corresponding background. From the preceding discussion, $B_1 = D$ and $B_2 = (W - D)$. With this notation, Eq. (9.4-1) becomes

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) \tag{9.4-2}$$

Thus, set $A \circledast B$ contains all the (origin) points at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c. By using the definition of set differences given in Eq. (2.6-19) and the dual relationship between erosion and dilation given in Eq. (9.2-5), we can write Eq. (9.4-2) as

$$A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2) \tag{9.4-3}$$

However, Eq. (9.4-2) is considerably more intuitive. We refer to any of the preceding three equations as the *morphological hit-or-miss transform*.

The reason for using a structuring element B_1 associated with objects and an element B_2 associated with the background is based on an assumed definition that two or more objects are distinct only if they form disjoint (disconnected) sets. This is guaranteed by requiring that each object have at least a one-pixel-thick background around it. In some applications, we may be interested in detecting certain patterns (combinations) of 1s and 0s within a set, in which case a background is not required. In such instances, the hit-or-miss transform reduces to simple erosion. As indicated previously, erosion is still a set of matches, but without the additional requirement of a background match for detecting individual objects. This simplified pattern detection scheme is used in some of the algorithms developed in the following section.

the principal applications of morphology is in extracting image c
that are useful in the representation and description of shape. In
we consider morphological algorithms for extracting boundaries,
components, the convex hull, and the skeleton of a region. We al
several methods (for region filling, thinning, thickening, and pru
are used frequently in conjunction with these algorithms as pre
processing steps. We make extensive use in this section of "min
designed to clarify the mechanics of each morphological process
troduce it. These images are shown graphically with 1s shaded
white.

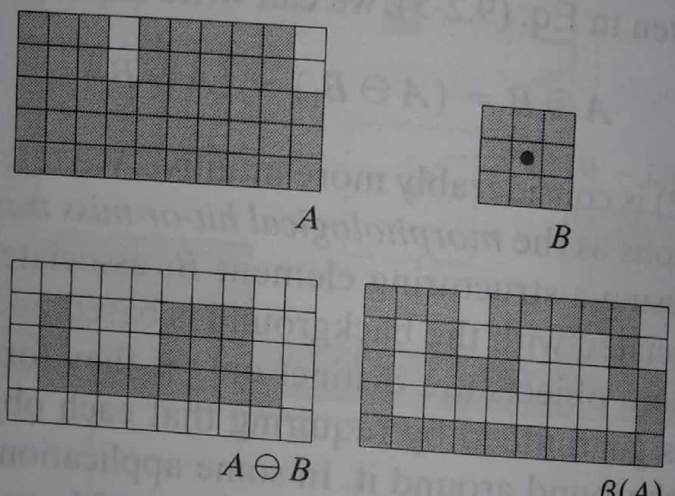
9.5.1 Boundary Extraction

The boundary of a set A, denoted by $\beta(A)$, can be obtained by first
A by B and then performing the set difference between A and its
That is,

$$\beta(A) = A - (A \ominus B)$$

where B is a suitable structuring element.

Figure 9.13 illustrates the mechanics of boundary extraction. It s
simple binary object, a structuring element B, and the result of usi
(9.5-1). Although the structuring element in Fig. 9.13(b) is among th
frequently used, it is by no means unique. For example, using a 5×5
turing element of 1s would result in a boundary between 2 and 3
thick.



a b
c d

FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Bound
given by the set difference between A and its erosion.

9.5.2 Hole Filling

A *hole* may be defined as a background region surrounded by a connected border of foreground pixels. In this section, we develop an algorithm based on set dilation, complementation, and intersection for filling holes in an image. Let A denote a set whose elements are 8-connected boundaries, each boundary enclosing a background region (i.e., a hole). Given a point in each hole, the objective is to fill all the holes with 1s.

We begin by forming an array, X_0, of 0s (the same size as the array containing A), except at the locations in X_0 corresponding to the given point in each hole, which we set to 1. Then, the following procedure fills all the holes with 1s:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1, 2, 3, \ldots \tag{9.5-2}$$

where B is the symmetric structuring element in Fig. 9.15(c). The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The set X_k then contains all the filled holes. The set union of X_k and A contains all the filled holes and their boundaries.

The dilation in Eq. (9.5-2) would fill the entire area if left unchecked. However, the intersection at each step with A^c limits the result to inside the region of interest. This is our first example of how a morphological process can be *conditioned* to meet a desired property. In the current application, it is appropriately called *conditional dilation*. The rest of Fig. 9.15 illustrates further the mechanics of Eq. (9.5-2). Although this example only has one hole, the concept clearly applies to any finite number of holes, assuming that a point inside each hole region is given.

nvexity. One simple approach to reduce this effect is to limit growth so hat it does not extend past the vertical and horizontal dimensions of the riginal set of points. Imposing this limitation on the example in Fig. 9.19 re-ulted in the image shown in Fig. 9.20. Boundaries of greater complexity can e used to limit growth even further in images with more detail. For exam-le, we could use the maximum dimensions of the original set of points along he vertical, horizontal, and diagonal directions. The price paid for refine-ments such as this is additional complexity and increased computational re-quirements of the algorithm.

9.5.5 Thinning

The thinning of a set A by a structuring element B, denoted $A \otimes B$, can be de-fined in terms of the hit-or-miss transform:

$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c \tag{9.5-6}$$

As in the previous section, we are interested only in pattern matching with the structuring elements, so no background operation is required in the hit-or-miss transform. A more useful expression for thinning A symmetrically is based on a *sequence* of structuring elements:

$$\{B\} = \{B^1, B^2, B^3, \ldots, B^n\} \tag{9.5-7}$$

where B^i is a rotated version of B^{i-1}. Using this concept, we now define thin-ning by a sequence of structuring elements as

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n) \tag{9.5-8}$$

The process is to thin A by *one pass* with B^1, then thin the result with one pass of B^2, and so on, until A is thinned with one pass of B^n. The entire process is repeated until no further changes occur. Each individual thinning pass is per-formed using Eq. (9.5-6).

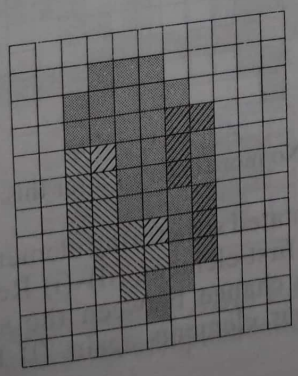


Figure 9.21(a) shows a set of structuring elements commo
thinning, and Fig. 9.21(b) shows a set A to be thinned by usin
dure just discussed. Figure 9.21(c) shows the result of thinnin
pass of A with B^1; and Figs. 9.21(d) through (k) show the resul
with the other structuring elements. Convergence was achieve
second pass of B^6. Figure 9.21(l) shows the thinned result. F
9.21(m) shows the thinned set converted to m-connectivity (s
2.5.2) to eliminate multiple paths.

9.5.6 Thickening

Thickening is the morphological dual of thinning and is defined by the e

$$A \odot B = A \cup (A \circledast B)$$

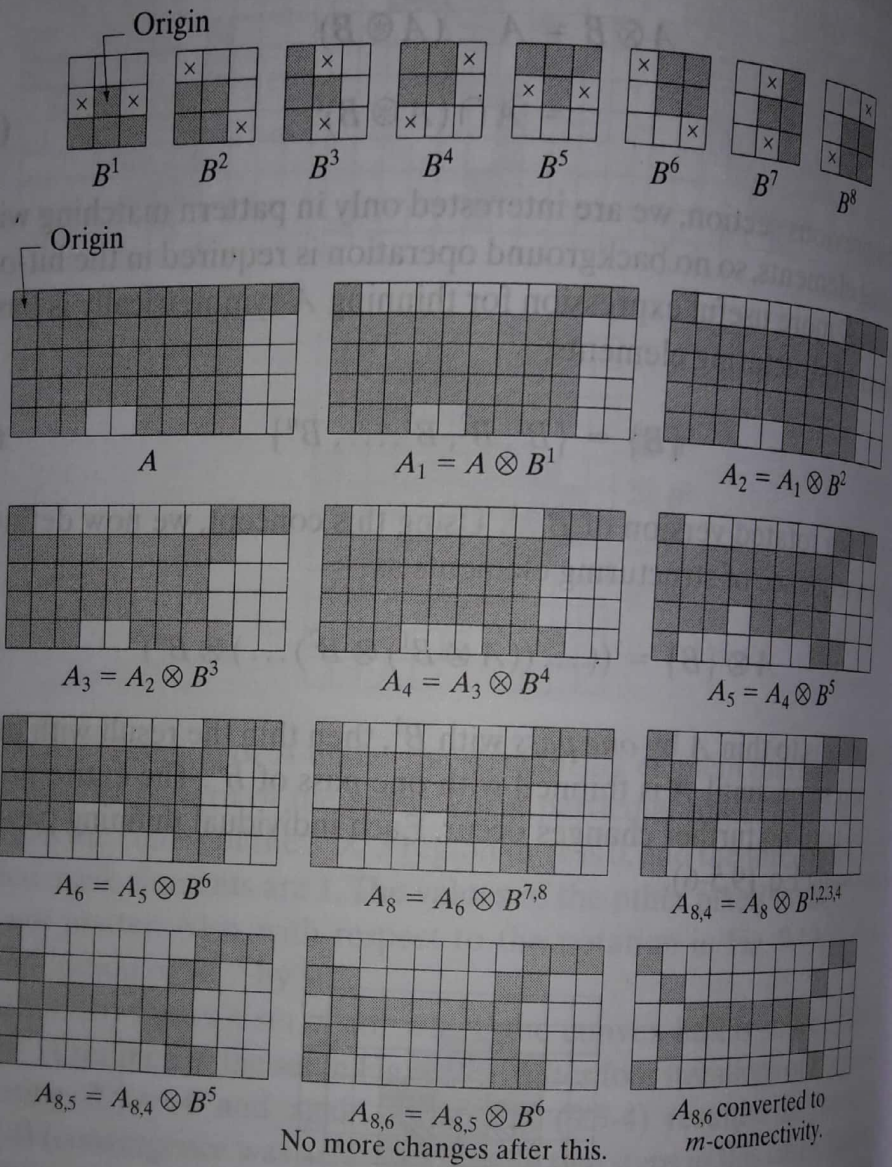


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to m-connectivity.

B is a structuring element suitable for thickening. As in thinning, thickening can be defined as a sequential operation:

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n) \qquad (9.5\text{-}10)$$

structuring elements used for thickening have the same form as those in Fig. 9.21(a), but with all 1s and 0s interchanged. However, a separate rithm for thickening is seldom used in practice. Instead, the usual procedure is to thin the background of the set in question and then complement the t. In other words, to thicken a set A, we form $C = A^c$, thin C, and then C^c. Figure 9.22 illustrates this procedure.

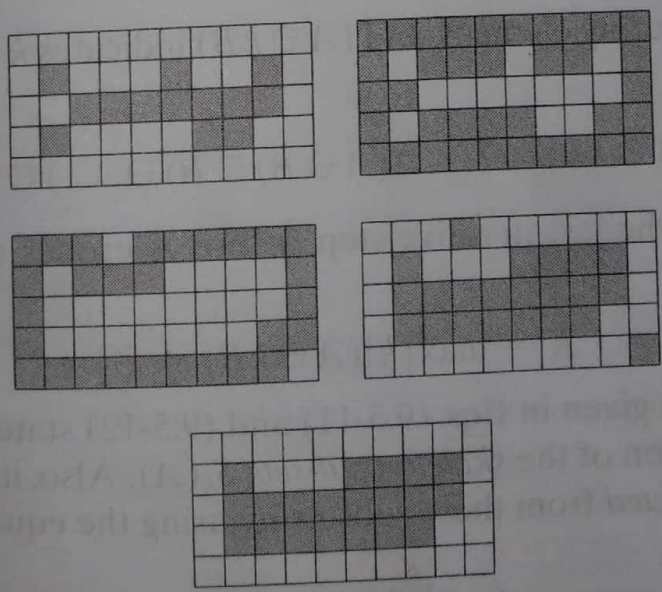
epending on the nature of A, this procedure can result in disconnected ts, as Fig. 9.22(d) shows. Hence thickening by this method usually is fold by postprocessing to remove disconnected points. Note from Fig. 9.22(c) the thinned background forms a boundary for the thickening process. useful feature is not present in the direct implementation of thickening g Eq. (9.5-10), and it is one of the principal reasons for using background ning to accomplish thickening.

7 Skeletons

Fig. 9.23 shows, the notion of a skeleton, $S(A)$, of a set A is intuitively sim- We deduce from this figure that

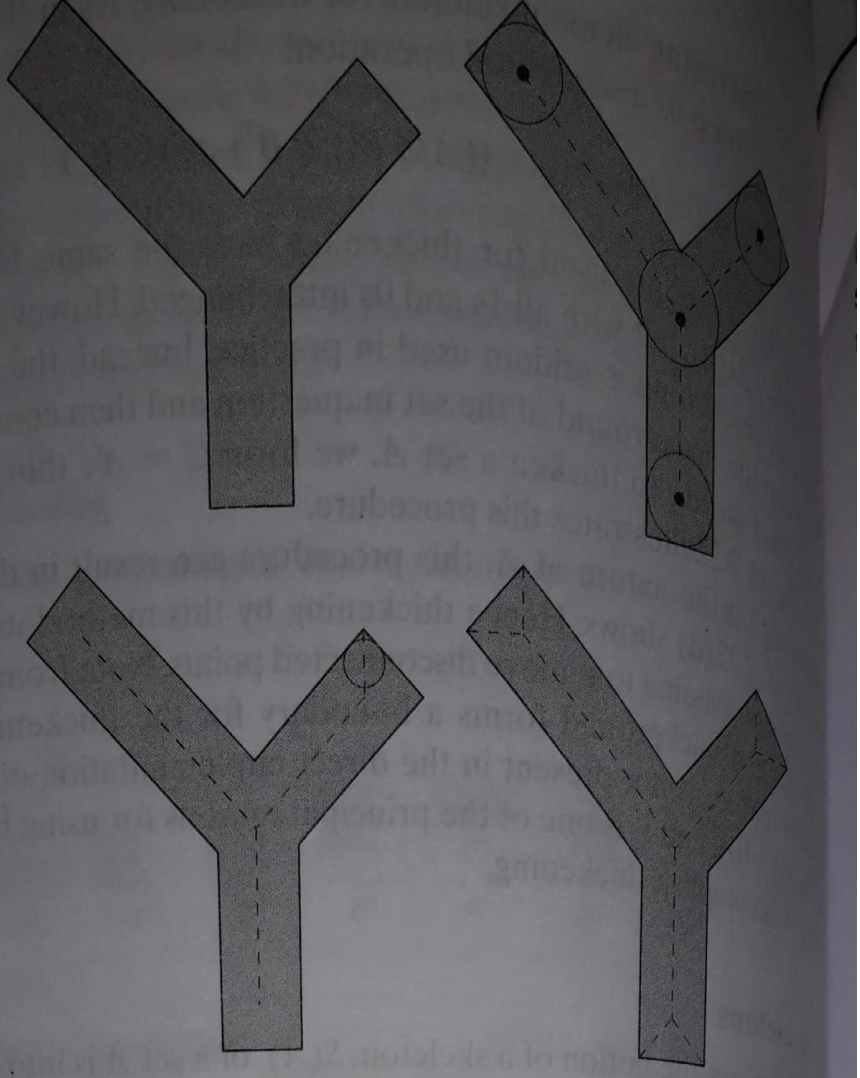
If z is a point of $S(A)$ and $(D)_z$ is the largest disk centered at z and contained in A, one cannot find a larger disk (not necessarily centered at z) containing $(D)_z$ and included in A. The disk $(D)_z$ is called a *maximum disk*.

The disk $(D)_z$ touches the boundary of A at two or more different places.



b
d

GURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement A. (d) Thickened set obtained by complementing (c). (e) Final result, with no sconnected points.



The skeleton of A can be expressed in terms of erosions and openings. That is, it can be shown (Serra [1982]) that

$$S(A) = \bigcup_{k=0}^{K} S_k(A) \tag{9.5-}$$

with

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B \tag{9.5-}$$

where B is a structuring element, and $(A \ominus kB)$ indicates k successive erosions of A:

$$(A \ominus kB) = ((\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B) \tag{9.5-}$$

k times, and K is the last iterative step before A erodes to an empty set. In other words,

$$K = \max\{k | (A \ominus kB) \neq \emptyset\} \tag{9.5-}$$

The formulation given in Eqs. (9.5-11) and (9.5-12) states that $S(A)$ can be obtained as the union of the *skeleton subsets* $S_k(A)$. Also, it can be shown that A can be *reconstructed* from these subsets by using the equation

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB) \tag{9.5-}$$

where $(S_k(A) \oplus kB)$ denotes k successive dilations of $S_k(A)$; that is,

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B) \tag{9.5-}$$

uncorrelated with [illegible] between pixel values and the values of noise components). Although assumptions are at least partially invalid in some applications (quantum-limited imaging, such as in X-ray and nuclear-medicine imaging, is a good example), the complexities of dealing with spatially dependent and correlated noise are beyond the scope of our discussion.

5.2.2 Some Important Noise Probability Density Functions

Based on the assumptions in the previous section, the *spatial* noise descriptor with which we shall be concerned is the statistical behavior of the intensity values in the noise component of the model in Fig. 5.1. These may be considered random variables, characterized by a probability density function (PDF). The following are among the most common PDFs found in image processing applications.

Gaussian noise

Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (also called *normal*) noise models are used frequently in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best.

The PDF of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2} \tag{5.2}$$

where z represents intensity, $\bar{z}$ is the mean[†] (average) value of z, and σ is its standard deviation. The standard deviation squared, σ^2, is called the *variance* of z. A plot of this function is shown in Fig. 5.2(a). When z is described by Eq. (5.2), approximately 70% of its values will be in the range $[(\bar{z} - \sigma), (\bar{z} + \sigma)]$, and about 95% will be in the range $[(\bar{z} - 2\sigma), (\bar{z} + 2\sigma)]$.

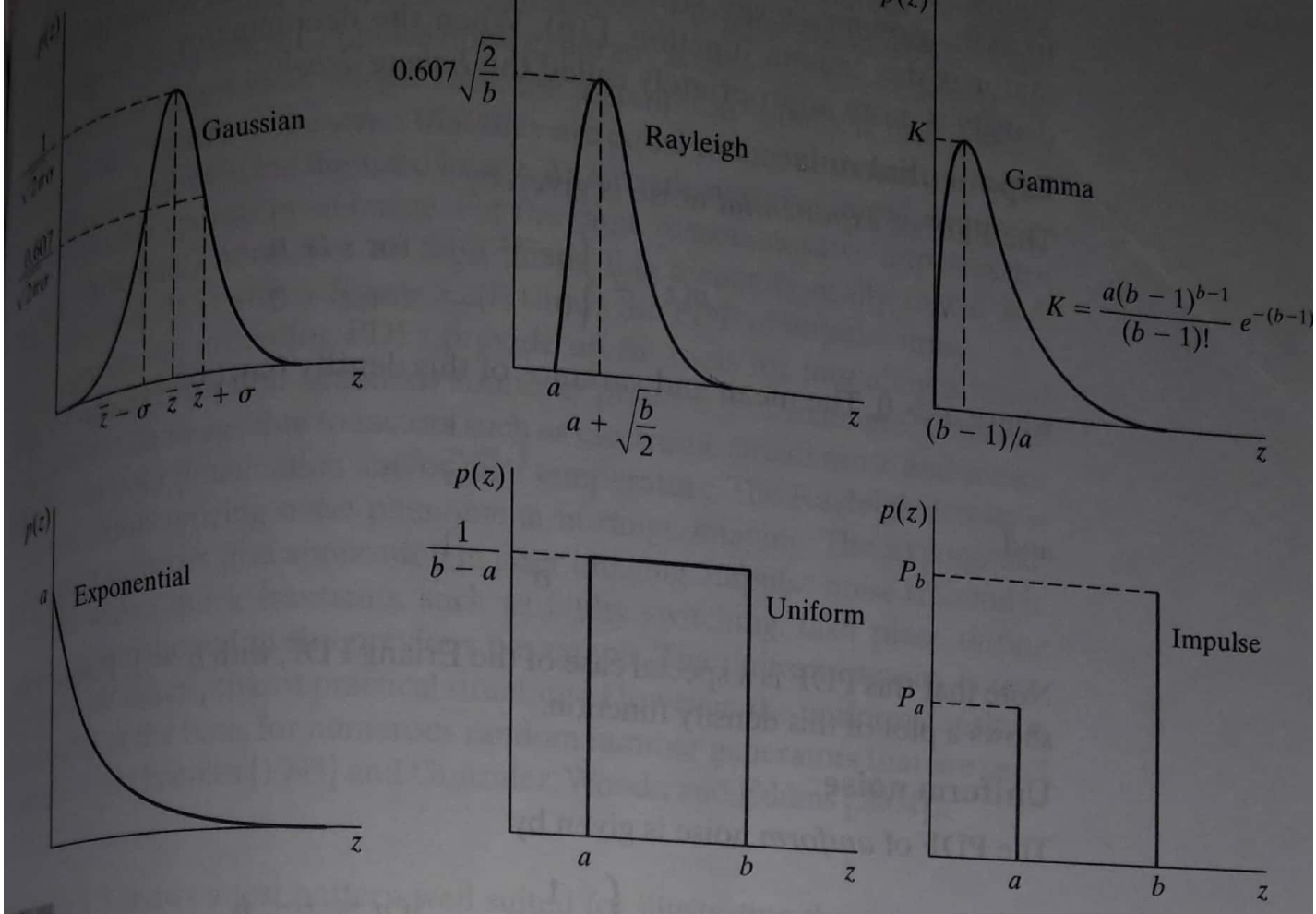
Rayleigh noise

The PDF of Rayleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b}(z - a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \tag{5.2}$$

The mean and variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b/4} \tag{5.2}$$



a b c
d e f

FIGURE 5.2 Some important probability density functions.

nd

$$\sigma^2 = \frac{b(4-\pi)}{4} \tag{5.2-4}$$

igure 5.2(b) shows a plot of the Rayleigh density. Note the displacement from he origin and the fact that the basic shape of this density is skewed to the right. he Rayleigh density can be quite useful for approximating skewed histograms.

rlang (gamma) noise

he PDF of Erlang noise is given by

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \tag{5.2-5}$$

ere the parameters are such that $a > 0$, b is a positive integer, and "!" indi- es factorial. The mean and variance of this density are given by

$$\bar{z} = \frac{b}{a} \tag{5.2-6}$$

d

$$\sigma^2 = \frac{b}{a^2} \tag{5.2-7}$$

Figure 5.2(c) shows a plot of this density. Although Eq. (5.2-5) often is refer to as the *gamma density*, strictly speaking this is correct only when the den inator is the gamma function, $\Gamma(b)$. When the denominator is as shown density is more appropriately called the *Erlang density*.

Exponential noise

The PDF of *exponential* noise is given by

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \tag{5.2}$$

where $a > 0$. The mean and variance of this density function are

$$\bar{z} = \frac{1}{a} \tag{5.2}$$

and

$$\sigma^2 = \frac{1}{a^2} \tag{5.2}$$

Note that this PDF is a special case of the Erlang PDF, with $b = 1$. Figure 5.2(shows a plot of this density function.

Uniform noise

The PDF of *uniform* noise is given by

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \tag{5.2-11}$$

The mean of this density function is given by

$$\bar{z} = \frac{a+b}{2} \tag{5.2-12}$$

and its variance by

$$\sigma^2 = \frac{(b-a)^2}{12} \tag{5.2-13}$$

Figure 5.2(e) shows a plot of the uniform density.

Impulse (salt-and-pepper) noise

The PDF of (*bipolar*) *impulse* noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \tag{5.2-14}$$

If $b > a$, intensity b will appear as a light dot in the image. Conversely, level a wil appear like a dark dot. If either P_a or P_b is zero, the impulse noise is called *unipolar*. If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly dis tributed over the image. For this reason, bipolar impulse noise also is called *salt-and-pepper* noise. *Data-drop-out* and *spike* noise also are terms used to refer to this type of noise. We use the terms *impulse* or *salt-and-pepper* noise interchangeably

Median filter

The best-known order-statistic filter is the *median filter*, which, as its name implies, replaces the value of a pixel by the median of the intensity levels in the neighborhood of that pixel:

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s, t)\} \tag{5.3}$$

The value of the pixel at (x, y) is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. In fact, as Example 5.3 below shows, the median filter yields excellent results for images corrupted by this type of noise. Computation of the median and implementation of this filter are discussed in Section 3.5.2.

Max and min filters

Although the median filter is by far the order-statistic filter most used in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but you will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called *max filter*, given by

$$\hat{f}(x, y) = \max_{(s,t)\in S_{xy}}\{g(s, t)\} \tag{5.3}$$

This filter is useful for finding the brightest points in an image. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area S_{xy}.

The 0th percentile filter is the *min filter*:

$$\hat{f}(x, y) = \min_{(s,t)\in S_{xy}} \{g(s, t)\} \tag{5.3-9}$$

This filter is useful for finding the darkest points in an image. Also, it reduces salt noise as a result of the min operation.

Midpoint filter

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

$$\hat{f}(x, y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s, t)\} + \min_{(s,t)\in S_{xy}} \{g(s, t)\}\right] \tag{5.3-10}$$

Note that this filter combines order statistics and averaging. It works best for randomly distributed noise, like Gaussian or uniform noise.

Alpha-trimmed mean filter

Suppose that we delete the $d/2$ lowest and the $d/2$ highest intensity values of $g(s, t)$ in the neighborhood S_{xy}. Let $g_r(s, t)$ represent the remaining $mn - d$ pixels. A filter formed by averaging these remaining pixels is called an *alpha-trimmed mean* filter:

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t)\in S_{xy}} g_r(s, t) \tag{5.3-11}$$

where the value of d can range from 0 to $mn - 1$. When $d = 0$, the alpha-trimmed filter reduces to the arithmetic mean filter discussed in the previous section. If we choose $d = mn - 1$, the filter becomes a median filter. For other values of d, the alpha-trimmed filter is useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise.

Figure 5.10(a) shows the circuit board image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. Figure 5.10(b) shows the result of median filtering with a filter of size 3×3. The improvement over Fig. 5.10(a) is significant, but several noise points still are visible. A second pass [on the image in Fig. 5.10(b)] with the median filter removed most of these points, leaving only few, barely visible noise points. These were removed with a third pass of the filter. These results are good examples of the power of median filtering in handling impulse-like additive noise. Keep in mind that repeated passes of a median filter will blur the image, so it is desirable to keep the number of passes as low as possible.

Figure 5.11(a) shows the result of applying the max filter to the pepper noise image of Fig. 5.8(a). The filter did a reasonable job of removing the pepper noise, but we note that it also removed (set to a light intensity level) some dark pixels from the borders of the dark objects. Figure 5.11(b) shows the result of applying the min filter to the image in Fig. 5.8(b). In this case, the min filter did a better job than the max filter on noise removal, but it removed some white points around the border of light objects. These made the light objects smaller and

5.3.1 Mean Filters

In this section we discuss briefly the noise-reduction capabilities of the spa filters introduced in Section 3.5 and develop several other filters whose p formance is in many cases superior to the filters discussed in that section.

Arithmetic mean filter

This is the simplest of the mean filters. Let S_{xy} represent the set of coordinate a rectangular subimage window (neighborhood) of size $m \times n$, centered at p (x, y). The arithmetic mean filter computes the average value of the corru image $g(x, y)$ in the area defined by S_{xy}. The value of the restored image point (x, y) is simply the arithmetic mean computed using the pixels in the reg defined by S_{xy}. In other words,

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$$

e that m and n tegers.

This operation can be implemented using a spatial filter of size m which all coefficients have value $1/mn$. A mean filter smooths local var in an image, and noise is reduced as a result of blurring.

metric mean filter

image restored using a *geometric mean* filter is given by the expression

$$\hat{f}(x, y) = \left[\prod_{(s,t)\in S_{xy}} g(s, t) \right]^{\frac{1}{mn}} \tag{5.3-4}$$

each restored pixel is given by the product of the pixels in the subimage ow, raised to the power $1/mn$. As shown in Example 5.2, a geometric mean achieves smoothing comparable to the arithmetic mean filter, but it tends se less image detail in the process.

monic mean filter

harmonic mean filtering operation is given by the expression

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s, t)}} \tag{5.3-5}$$

harmonic mean filter works well for salt noise, but fails for pepper noise. es well also with other types of noise like Gaussian noise.

traharmonic mean filter

contraharmonic mean filter yields a restored image based on the expression

$$\hat{f}(x, y) = \frac{\sum_{(s,t)\in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s, t)^{Q}} \tag{5.3-6}$$

re Q is called the *order* of the filter. This filter is well suited for reducing or ually eliminating the effects of salt-and-pepper noise. For positive values of Q, filter eliminates pepper noise. For negative values of Q it eliminates salt noise. annot do both simultaneously. Note that the contraharmonic filter reduces to arithmetic mean filter if $Q = 0$, and to the harmonic mean filter if $Q = -1$.

Figure 5.7(a) shows an 8-bit X-ray image of a circuit board, and Fig. 5.7(b) ws the same image, but corrupted with additive Gaussian noise of zero an and variance of 400. For this type of image this is a significant level of se. Figures 5.7(c) and (d) show, respectively, the result of filtering the noisy

5.6.1 Estimation by Image Observation

Suppose that we are given a degraded image without any knowledge about the degradation function H. Based on the assumption that the image was degraded by a linear, position-invariant process, one way to estimate H is to gather information from the image itself. For example, if the image is blurred, we can look at a small rectangular section of the image containing sample structures like part of an object and the background. In order to reduce the effect of noise, we would look for an area in which the signal content is strong (e.g., an area of high contrast). The next step would be to process the subimage to arrive at a result that is as unblurred as possible. For example, we can do this by sharpening the subimage with a sharpening filter and even by processing small areas by hand.

Let the observed subimage be denoted by $g_s(x, y)$, and let the processed subimage (which in reality is our estimate of the original image in that area) be denoted by $\hat{f}_s(x, y)$. Then, assuming that the effect of noise is negligible because of our choice of a strong-signal area, it follows from Eq. (5.5-17) that

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)} \tag{5.6-1}$$

From the characteristics of this function, we then deduce the complete degradation function $H(u, v)$ based on our assumption of position invariance. For example, suppose that a radial plot of $H_s(u, v)$ has the approximate shape of a Gaussian curve. We can use that information to construct a function $H(u, v)$ on a larger scale, but having the same basic shape. We then use $H(u, v)$ in one of the restoration approaches to be discussed in the following sections. Clearly, this is a laborious process used only in very specific circumstances such as, for example, restoring an old photograph of historical value.

5.6.2 Estimation by Experimentation

If equipment similar to the equipment used to acquire the degraded image is available, it is possible in principle to obtain an accurate estimate of the degradation. Images similar to the degraded image can be acquired with various system settings until they are degraded as closely as possible to the image we wish to restore. Then the idea is to obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system settings. As noted in Section 5.5, a linear, space-invariant system is characterized completely by its impulse response.

An impulse is simulated by a bright dot of light, as bright as possible to reduce the effect of noise to negligible values. Then, recalling that the Fourier transform of an impulse is a constant, it follows from Eq. (5.5-17) that

$$H(u, v) = \frac{G(u, v)}{A} \tag{5.6-2}$$

where, as before, $G(u, v)$ is the Fourier transform of the observed image and A is a constant describing the strength of the impulse. Figure 5.24 shows an example.

5.6.3 Estimation by Modeling

Degradation modeling has been used for many years because of the insight it affords into the image restoration problem. In some cases, the model can even take into account environmental conditions that cause degradations. For example, a degradation model proposed by Hufnagel and Stanley [1964] is based on the physical characteristics of atmospheric turbulence. This model has a familiar form:

$$H(u, v) = e^{-k(u^2+v^2)^{5/6}} \tag{5.6-3}$$

where k is a constant that depends on the nature of the turbulence. With the exception of the 5/6 power on the exponent, this equation has the same form as the Gaussian lowpass filter discussed in Section 4.8.3. In fact, the Gaussian LPF is used sometimes to model mild, uniform blurring. Figure 5.25 shows examples

...ated by simulating blurring an image using Eq. (5.6-3) with values ...0.0025 (severe turbulence), $k = 0.001$ (mild turbulence), and $k = 0.00025$ ...turbulence). All images are of size 480×480 pixels.

Another major approach in modeling is to derive a mathematical model ...ting from basic principles. We illustrate this procedure by treating in some ...the case in which an image has been blurred by uniform linear motion ...ween the image and the sensor during image acquisition. Suppose that an ...e $f(x, y)$ undergoes planar motion and that $x_0(t)$ and $y_0(t)$ are the time- ...ing components of motion in the x- and y-directions, respectively. The ...l exposure at any point of the recording medium (say, film or digital mem- ...) is obtained by integrating the instantaneous exposure over the time inter- ...during which the imaging system shutter is open.

Assuming that shutter opening and closing takes place instantaneously, and ...hat the optical imaging process is perfect, isolates the effect of image motion. ...hen, if T is the duration of the exposure, it follows that

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)]\, dt \tag{5.6-4}$$

...here $g(x, y)$ is the blurred image.

From Eq. (4.5-7), the Fourier transform of Eq. (5.6-4) is

$$\begin{aligned} G(u, v) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x, y)e^{-j2\pi(ux+vy)}\, dx\, dy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f[x - x_0(t), y - y_0(t)]\, dt\right]e^{-j2\pi(ux+vy)}\, dx\, dy \end{aligned} \tag{5.6-5}$$

Reversing the order of integration allows Eq. (5.6-5) to be expressed in the form

$$G(u, v) = \int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)]e^{-j2\pi(ux+vy)}\, dx\, dy\right] dt \tag{5.6-6}$$

The term inside the outer brackets is the Fourier transform of the displaced ...unction $f[x - x_0(t), y - y_0(t)]$. Using Eq. (4.6-4) then yields the expression

$$\begin{aligned} G(u, v) &= \int_0^T F(u, v)e^{-j2\pi[ux_0(t)+vy_0(t)]}\, dt \\ &= F(u, v)\int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\, dt \end{aligned} \tag{5.6-7}$$

...where the last step follows from the fact that $F(u, v)$ is independent of t.

By defining

$$H(u, v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]}\, dt \tag{5.6-8}$$