UNIT 3
Basic Morphological Algorithms

Course: Digital Image Processing(EC662)
Course Instructor: Prof. Shashidhar R
Dept.: Electronics and Communication Engg.

CONTENTS

- Dilation and Erosion
- Opening and Closing
- The Hit or Miss transformation
- Boundary extraction
- Region filling
- Extraction of Connected components
- Convex Hull, Thinning, Thickening and Pruning

- The word morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants.
- We use the same word here in the context of mathematical morphology as a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull.
- Morphological techniques used for pre or post processing, such as morphological filtering, thinning and pruning.

- Morphological operations are defined in terms of sets.
- In image processing, we use morphology with two types of sets of pixels:
  - Objects
  - Structuring elements (SE’s).
- objects are defined as sets of foreground pixels.
- Structuring elements can be specified in terms of both foreground and background pixels.
- In addition, structuring elements sometimes contain so-called “don’t care” elements, denoted by X, signifying that the value of that particular element in the SE does not matter.

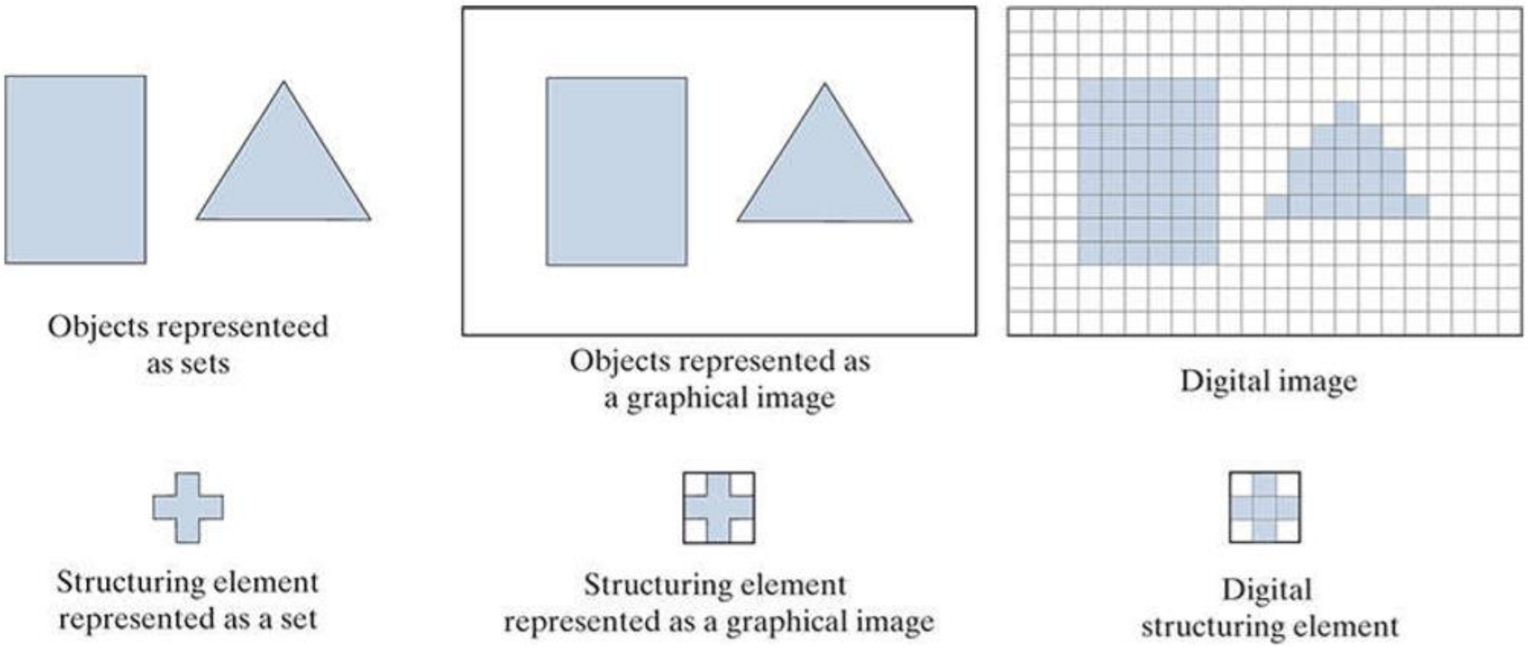


Fig.9.1 Top row.
Left: Objects represented as graphical sets. Center: Objects embedded in a background to form a graphical image. Right: Object and background are digitized to form a digital image (note the grid).

Second row: Example of a structuring element represented as a set, a graphical image, and finally as a digital SE.

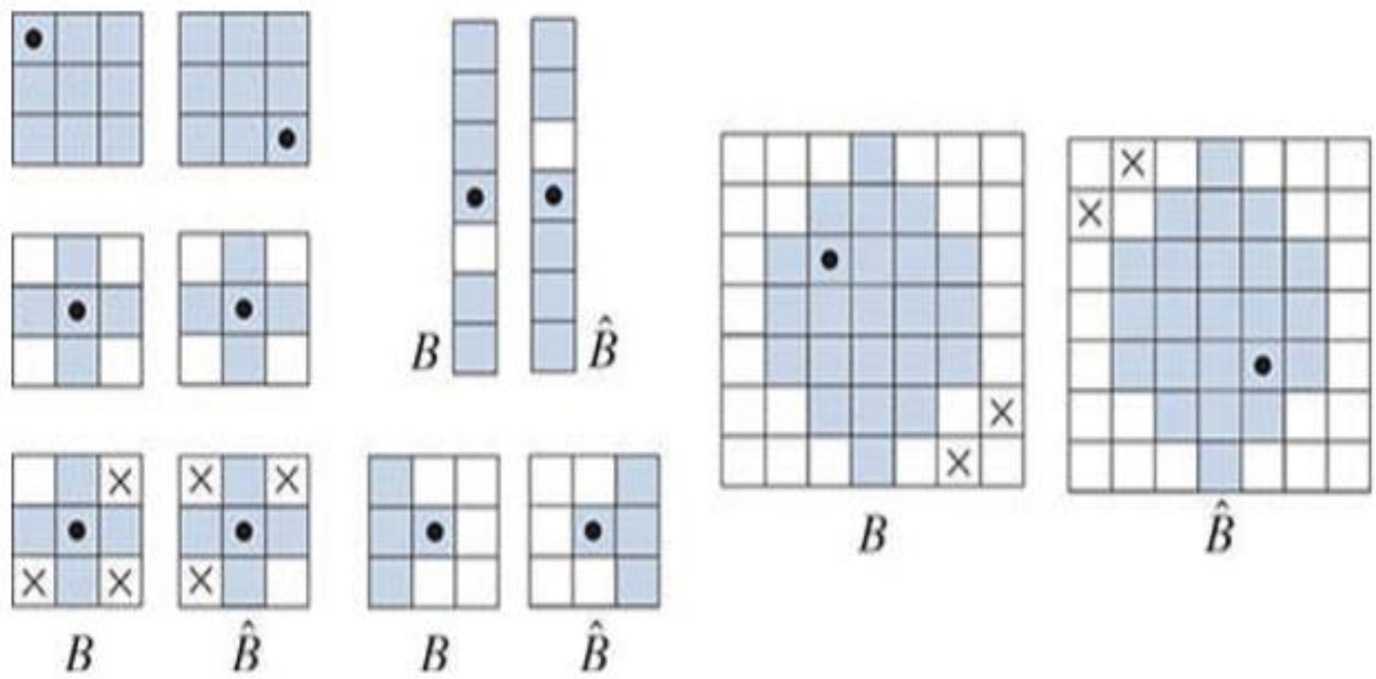


Figure 9.2. Structuring elements and their reflections about the origin (the are don’t care elements, and the dots denote the origin). Reflection is rotation by of an SE about its origin.

Erosion and Dilation

- Morphology by studying two operations: erosion and dilation.
- These operations are fundamental to morphological processing.
- Many of the morphological algorithms discussed in this chapter are based on these two primitive operations.

- **Erosion**: Morphological expressions are written in terms of structuring elements and a set, A, of foreground pixels, or in terms of structuring elements and an image, I, that contains A.
- We consider the former approach first. With A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, is defined as

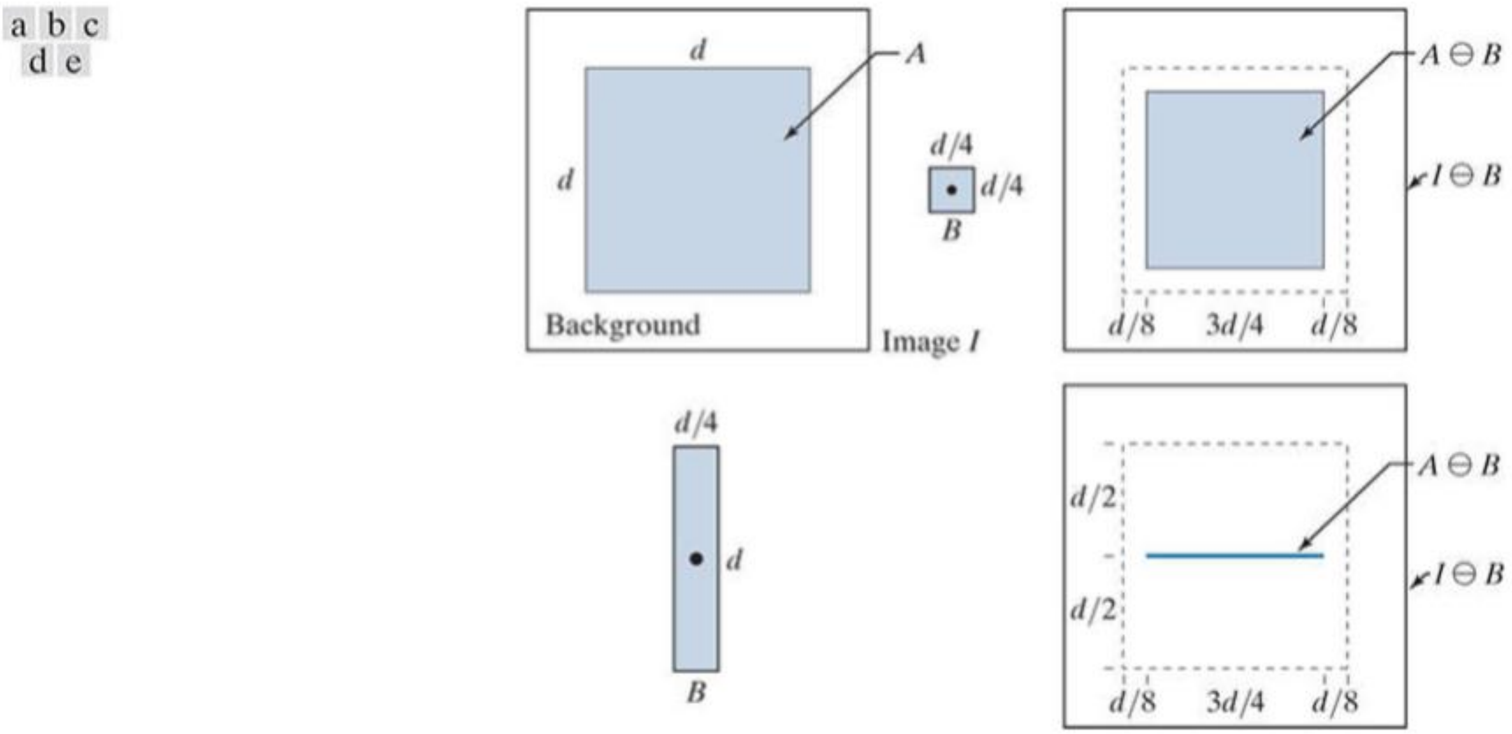
$$A \ominus B = \{z \mid (B)_z \subseteq A\} \tag{9-3}$$

where *A* is a set of foreground pixels, *B* is a structuring element, and the *z*'s are foreground values (1's). In words, this equation indicates that the erosion of *A* by *B* is the set of all points z such that *B*, translated by z, is contained in *A*. (Remember, displacement is defined with respect to the *origin* of *B*.) **Equation (9-3)** is the formulation that resulted in the *foreground* pixels of the image in **Fig. 9.3(c)**.

As noted, we work with sets of foreground pixels embedded in a set of background pixels to form a complete image, *I*. Thus, inputs and outputs of our morphological procedures are images, not individual sets. We *could* make this fact explicit by writing **Eq. (9-3)** as

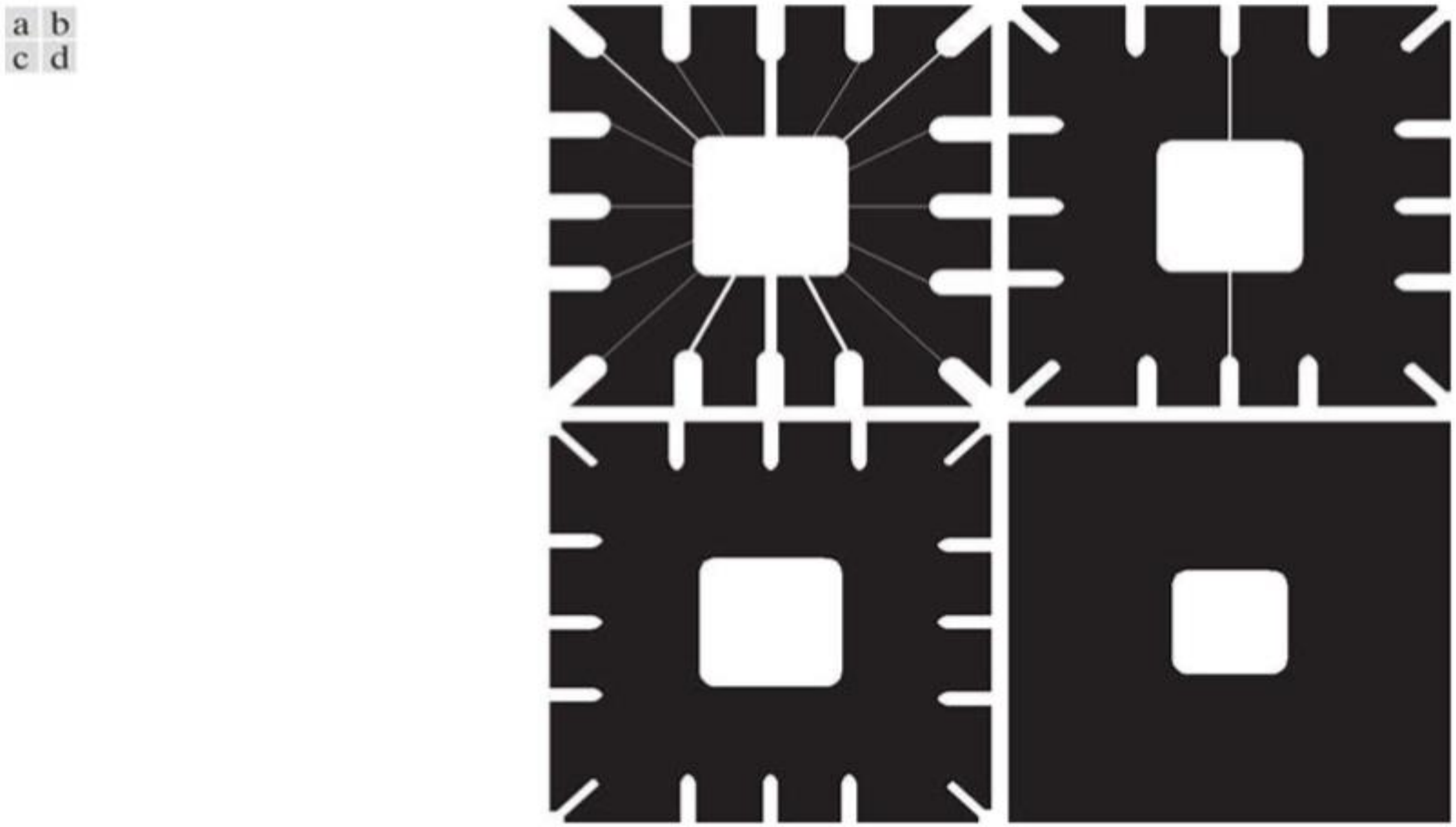
$$I \ominus B = \{z \mid (B)_z \subseteq A \text{ and } A \subseteq I\} \cup \{A^c \mid A^c \subseteq I\} \tag{9-4}$$

where *I* is a rectangular array of foreground and background pixels. The contents of the first braces say the same thing as **Eq. (9-3)**, with the added clarification that *A* is a subset of (i.e., is contained in) *I*. The union with the operation inside the second set of braces "adds" the pixels that are not in subset *A* (i.e., A^c, which is the set of background pixels) to the result from the first braces, requiring also that the background pixels be part of the rectangle defined by *I*. In words, all this equation says is that erosion of *I* by *B* is the set of all points, *z*, such that *B*, translated by *z*, is contained in *A*. The equation also makes explicit that *A* is contained in *I*, that the result is embedded in a set of background pixels, and that the entire process is of the same size as *I*.



(a) Image I, consisting of a set (object) A, and background. (b) Square SE, B (the dot is the origin). (c) Erosion of A by B (shown shaded in the resulting image). (d) Elongated SE. (e) Erosion of A by B. (The erosion is a line.) The dotted border in (c) and (e) is the boundary of A, shown for reference

EXAMPLE 9.1: USING EROSION TO REMOVE IMAGE COMPONENTS.



Using erosion to remove image components. (a) A binary image of a wire-bond mask in which foreground pixels are shown in white. (b)–(d) Image eroded using square structuring elements of sizes 11x11,15x15 and 45x45 elements, respectively, all valued 1.

- Figure 9.5(a) is a binary image depicting a simple wire-bond mask. As mentioned previously, we generally show the foreground pixels in binary images in white and the background in black. Suppose that we want to remove the lines connecting the center region to the border pads in Fig.9.5(a) . Eroding the image (i.e., eroding the foreground pixels of the image) with a square structuring element of size whose components are all 1's removed most of the lines, as Fig.9.5(b) shows. The reason that the two vertical lines in the center were thinned but not removed completely is that their width is greater than 11 pixels. Changing the SE size to elements and eroding the original image again did remove all the connecting lines, as Fig. 9.5(c) shows. An alternate approach would have been to erode the image in Fig.9.5(b) again, using the same , or smaller, SE. Increasing the size of the structuring element even more would eliminate larger components. For example, the connecting lines and the border pads can be removed with a structuring element of size elements applied to the original image, as Fig. 9.5(d)

DILATION

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

$$A \oplus B = \left\{ z \,\middle|\, (\hat{B})_z \cap A \neq \emptyset \right\} \tag{9-6}$$

This equation is based on reflecting B about its origin and translating the reflection by z, as in erosion. The dilation of A by B then is the set of all displacements, z, such that the foreground elements of $\hat{B}$ overlap at least one element of A. (Remember, z is the displacement of the origin of $\hat{B}$.) Based on this interpretation, **Eq. (9-6)** can be written equivalently as

$$A \oplus B = \left\{ z \,\middle|\, [(\hat{B})_z \cap A] \subseteq A \right\} \tag{9-7}$$

Equations (9-6) and **(9-7)** are not the only definitions of dilation currently in use (see **Problems 9.14** and **9.15** for two different, yet equivalent, definitions). As with erosion, the preceding definitions have the advantage of being more intuitive when structuring element B is viewed as a convolution kernel. As noted earlier, the basic process of flipping (rotating) B about its origin and then successively displacing it so that it slides over set A is analogous to spatial convolution. However, keep in mind that dilation is based on set operations and therefore is a nonlinear operation, whereas convolution is a sum of products, which is a linear operation.

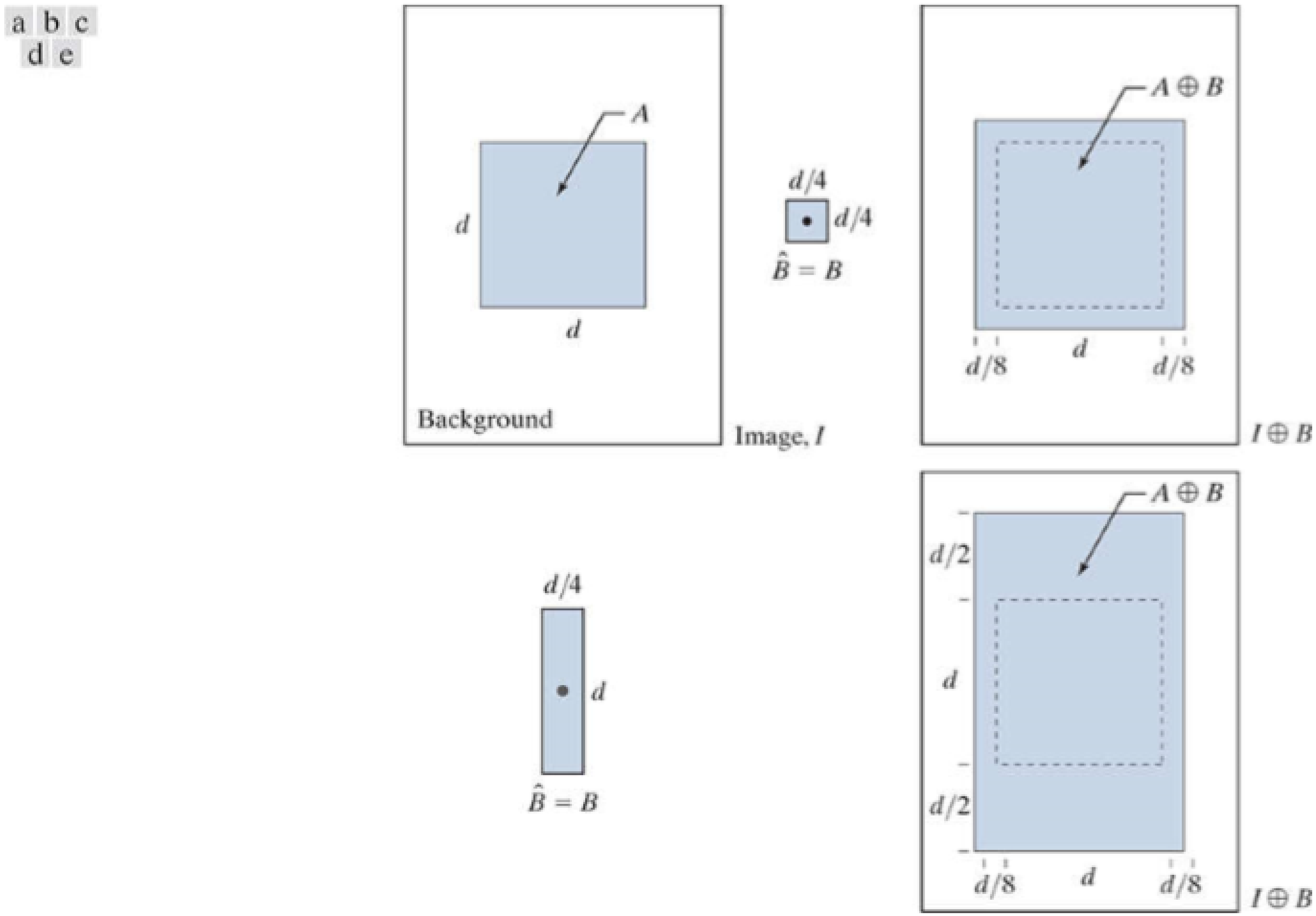


Figure 9.6 (a) Image I, composed of set(object) A and background.(b)Square SE(the dot is the origin). (c) Dilation of A by B(shown shaded).(d) Elongated SE. (e) Dilation of A by this element. The dotted line in (c) and (e) is the boundary of A.

EXAMPLE 9.1: USING DILATION TO REPAIR BROKEN CHARACTERS IN AN IMAGE

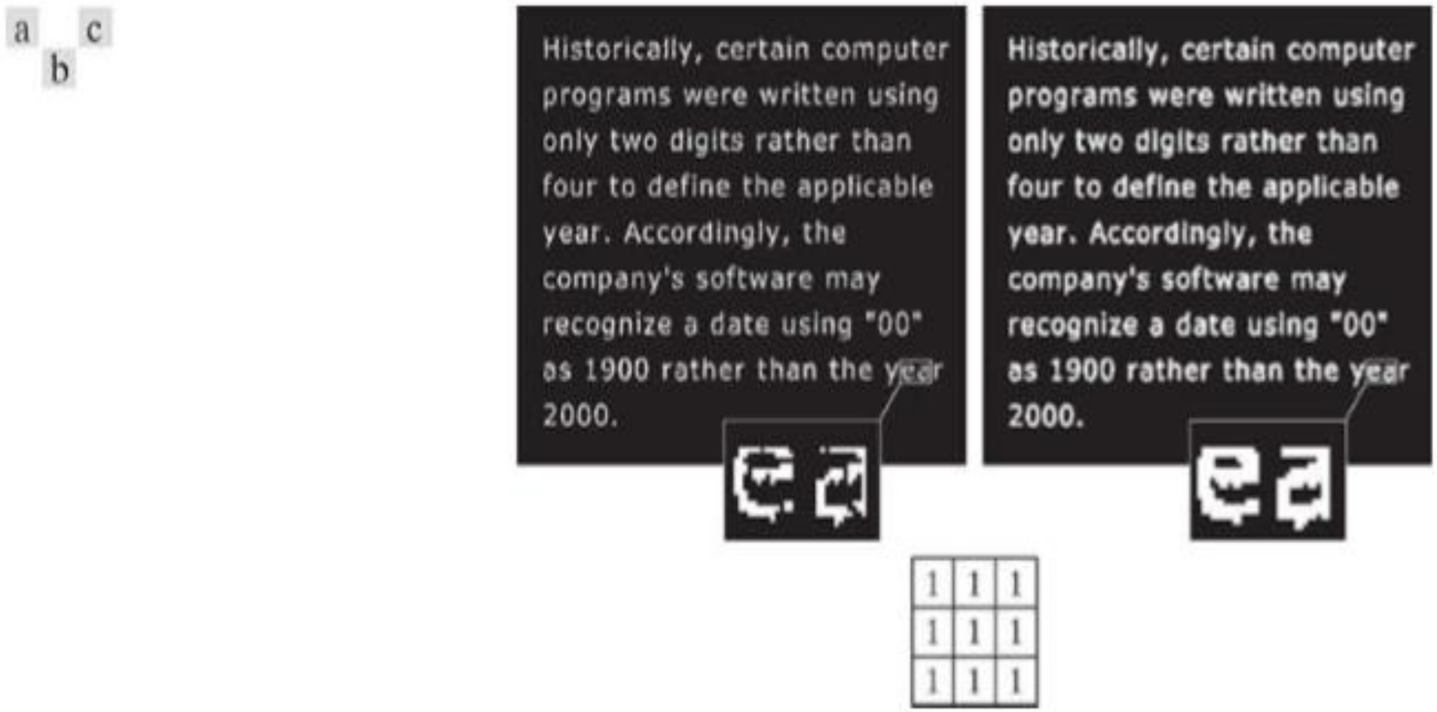


Figure 9.7 (a)Low-resolution text showing broken characters (see magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined

DUALITY

Erosion and dilation are *duals* of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B} \tag{9-8}$$

and

$$(A \oplus B)^c = A^c \ominus \hat{B} \tag{9-9}$$

Equation (9-8) indicates that erosion of *A* by *B* is the complement of the dilation of A^c by $\hat{B}$, and vice versa. The duality property is useful when the structuring element values are symmetric with respect to its origin (as often is the case), so that $\hat{B} = B$. Then, we can obtain the erosion of *A* simply by dilating its background (i.e., dilating A^c) with the same structuring element and complementing the result. Similar comments apply to Eq. (9-9) .

We proceed to prove formally the validity of Eq. (9-8) in order to illustrate a typical approach for establishing the validity of morphological expressions. Starting with the definition of erosion, it follows that

$$(A \ominus B)^c = \{z \mid (B)_z \subseteq A\}^c$$

If set $(B)_z$ is contained in *A*, then it follows that $(B)_z \cap A^c = \emptyset$, in which case the preceding expression becomes

$$(A \ominus B)^c = \{z \mid (B)_z \cap A^c = \emptyset\}^c$$

But the *complement* of the set of *z*'s that satisfy $(B)_z \cap A^c = \emptyset$ is the set of *z*'s such that $(B)_z \cap A^c \neq \emptyset$. Therefore,

$$\begin{aligned}(A \ominus B)^c &= \{z \mid (B)_z \cap A^c \neq \emptyset\} \\ &= A^c \oplus \hat{B}\end{aligned}$$

Opening and Closing

- In the previous section, dilation expands the components of a set and erosion shrinks it.
- In this section, we discuss two other important morphological operations: opening and closing.
- Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions.
- Closing also tends to smooth sections of contours, but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour

The *opening* of set A by structuring element B, denoted by $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{9-10}$$

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

Similarly, the *closing* of set A by structuring element B, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{9-11}$$

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

- Equation (9-10) has a simple geometrical interpretation: The opening of A by B is the union of all the translations of B so that B fits entirely in A.

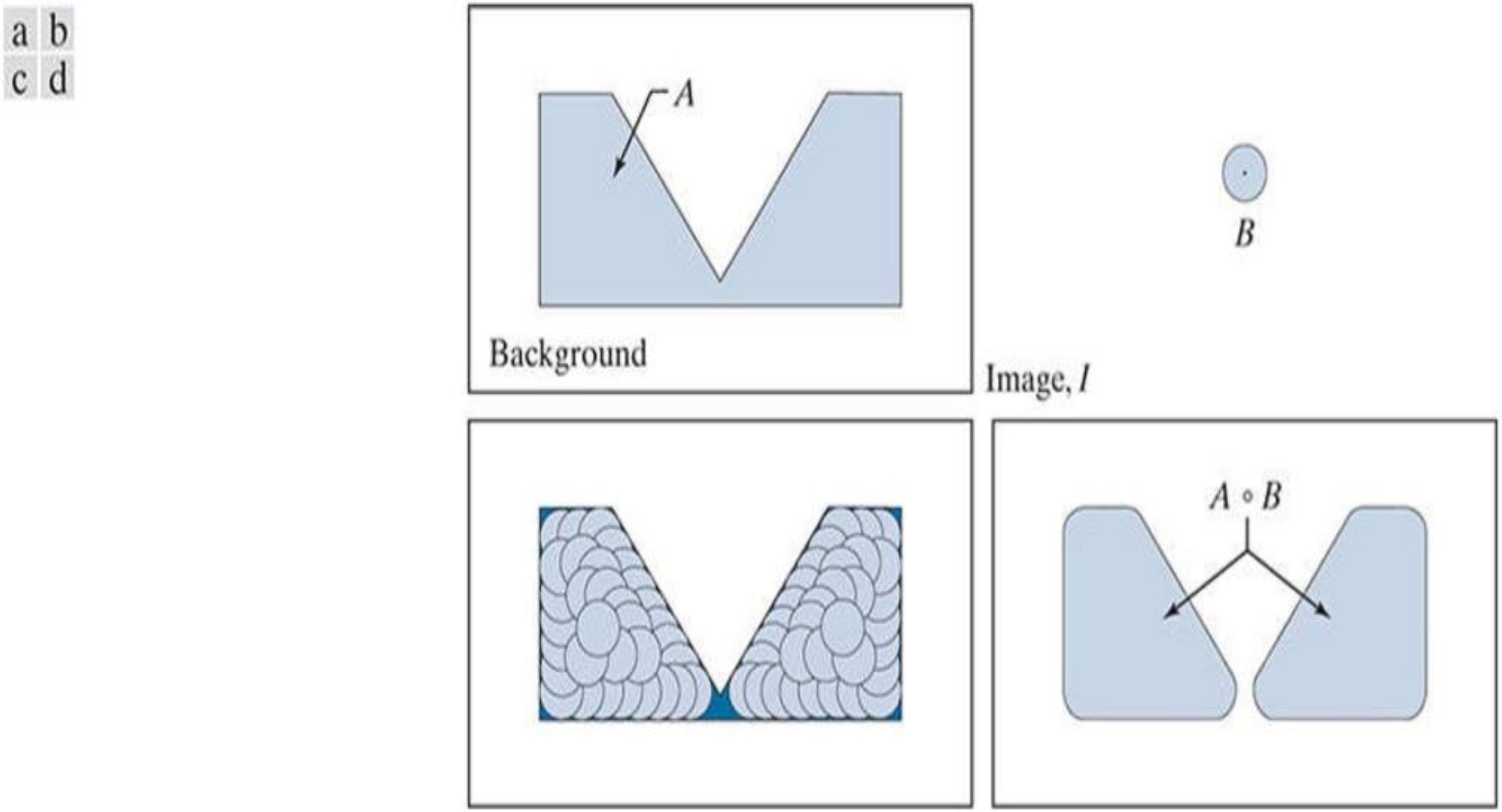


FIGURE 9.8

(a) Image I, composed of set (object) A and background. (b) Structuring element, B. (c) Translations of B while being contained in A. (A is shown dark for clarity.) (d) Opening of A by B.

Figure9.8(a) shows an image containing a set (object) A and

Fig. 9.8(b) is a solid, circular structuring element, B.

Figure 9.8(c) shows some of the translations of B such that it is contained within A, and the set shown shaded in Fig.9.8(d) is the union of all such possible translations.

Observe that, in this case, the opening is a set composed of two disjoint subsets, resulting from the fact that B could not fit in the narrow segment in the center of A.

As you will see shortly, the ability to eliminate regions narrower than the structuring element is one of the key features of morphological opening

The interpretation that the opening of A by B is the union of all the translations of B such that B fits entirely within A can be written in equation form as

- The interpretation that the opening of A by B is the union of all the translations of B such that B fits entirely within A can be written in equation form as

$$A \circ B = \cup\{(B)_z \mid (B)_z \subseteq A\} \tag{9-12}$$

where $\cup$ denotes the union of the sets inside the braces.

Closing has a similar geometric interpretation, except that now we translate B outside A. The closing is then the complement of the union of all translations of B that do not overlap A.

Figure 9.9 illustrates this concept. Note that the boundary of the closing is determined by the furthest points B could reach without going inside any part of A. Based on this interpretation, we can write the closing of A by B as

$$A \bullet B = \left[\cup\{(B)_z \mid (B)_z \cap A = \emptyset\}\right]^c \tag{9-13}$$

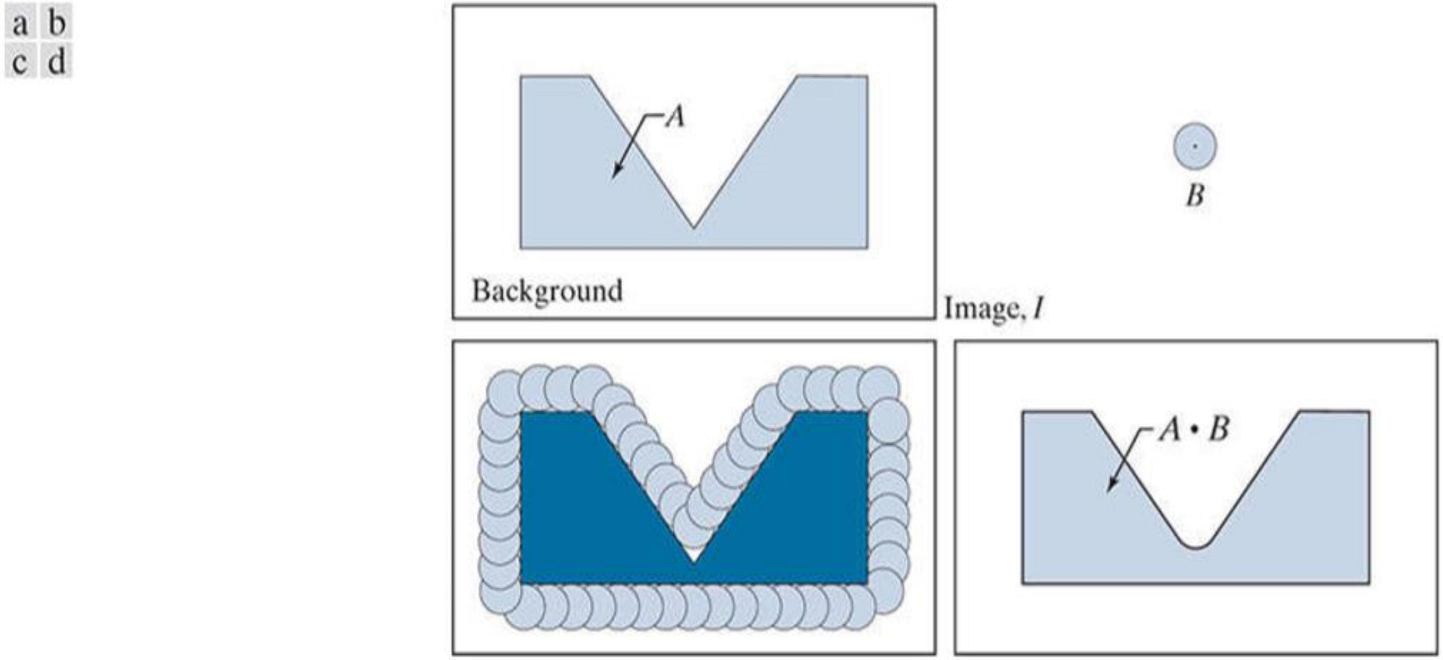


FIGURE 9.9

(a) Image I, composed of set (object) A, and background. (b) Structuring element B. (c) Translations of B such that B does not overlap any part of A. (A is shown dark for clarity.) (d) Closing of A by B.

- (A+B)-B

EXAMPLE 9.3: MORPHOLOGICAL OPENING AND CLOSING

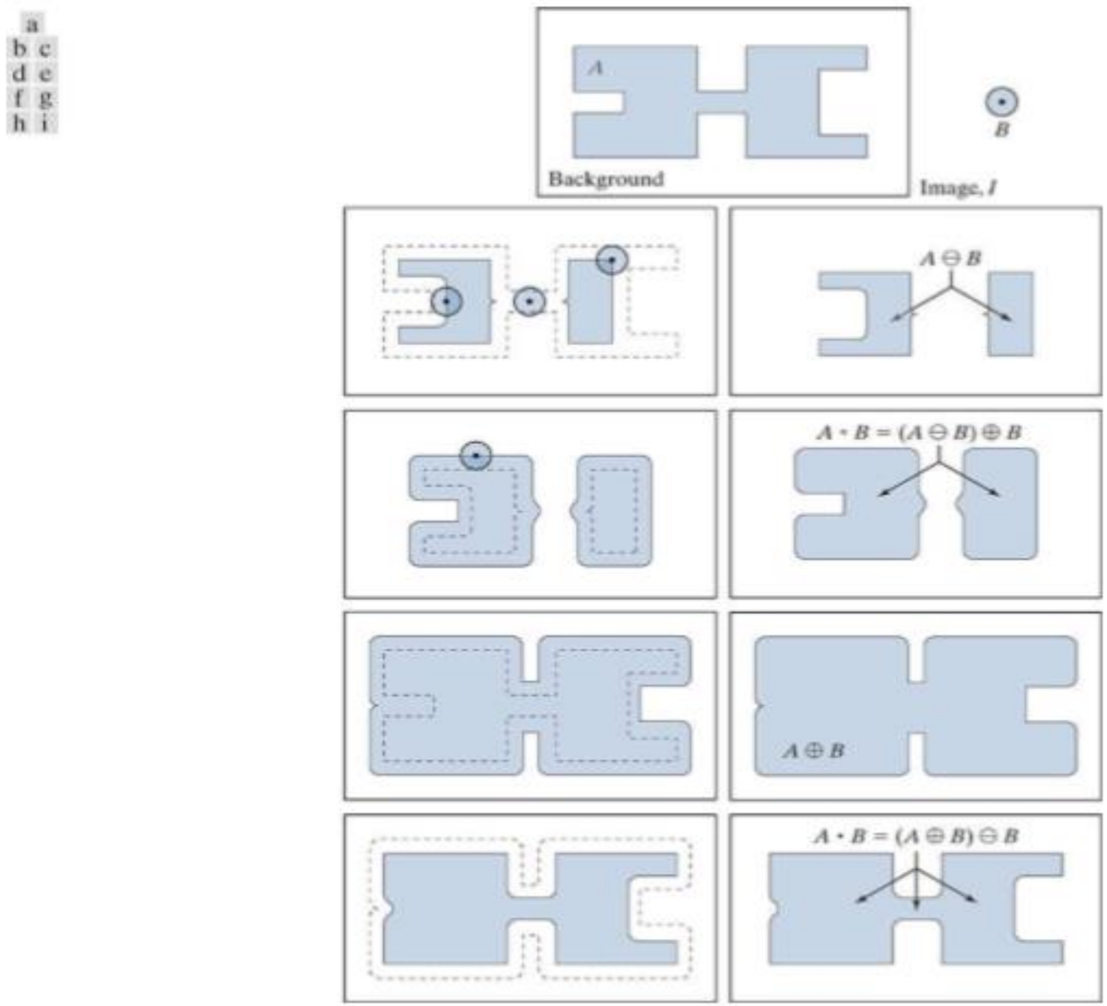


FIGURE 9.10
Morphological opening and closing. (a) Image I, composed of a set (object) A and background; a solid, circular structuring element is shown also. (The dot is the origin.) (b) Structuring element in various positions. (c)-(i) The morphological operations used to obtain the opening and closing.

- Figure 9.10 shows in more detail the process and properties of opening and closing. Unlike Figs.9.8 and 9.9, whose main objectives are overall geometrical interpretations, this figure shows the individual processes and also pays more attention to the relationship between the scale of the final results and the size of the structuring elements.
- Figure 9.10(a) shows an image containing a single object (set) A, and a disk structuring element.
- Figure9.10(b) shows various positions of the structuring element during erosion.
- This process resulted in the disjoint set in Fig.9.10(c)
- Note how the bridge between the two main sections was eliminated. Its width was thin in relation to the diameter of the structuring element, which could not be completely contained in this part of the set, thus violating the definition of erosion.
- The same was true of the two rightmost members of the object. Protruding elements where the disk did not fit were eliminated.
- Figure 9.10(d) shows the process of dilating the eroded set,

- Fig.9.10(e) shows the final result of opening. Morphological opening removes regions that cannot contain the structuring element, smoothes object contours, breaks thin connections, and removes thin protrusions.
- Figures9.10(f) through (i) show the results of closing A with the same structuring element. As with opening, closing also smoothes the contours of objects. However, unlike opening, closing tends to join narrow breaks, fills long thin gulfs, and fills objects smaller than the structuring element. In this example, the principal result of closing was that it filled the small gulf on the left of set A.

As with erosion and dilation, opening and closing are duals of each other with respect to set complementation and reflection:

$$(A \circ B)^c = \left(A^c \bullet \hat{B}\right) \tag{9-14}$$

and

$$(A \bullet B)^c = \left(A^c \circ \hat{B}\right) \tag{9-15}$$

- Morphological opening has the following properties:

a. $A \circ B$ is a subset of A.

b. If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.

c. $(A \circ B) \circ B = A \circ B$.

- Similarly, closing satisfies the following properties

a. A is a subset of $A \bullet B$.

b. If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.

c. $(A \bullet B) \bullet B = A \bullet B$.

Note from condition (c) in both cases that multiple openings or closings of a set have no effect after the operation has been applied once.

EXAMPLE 9.4 USING OPENING AND CLOSING FOR MORPHOLOGICAL FILTERING.

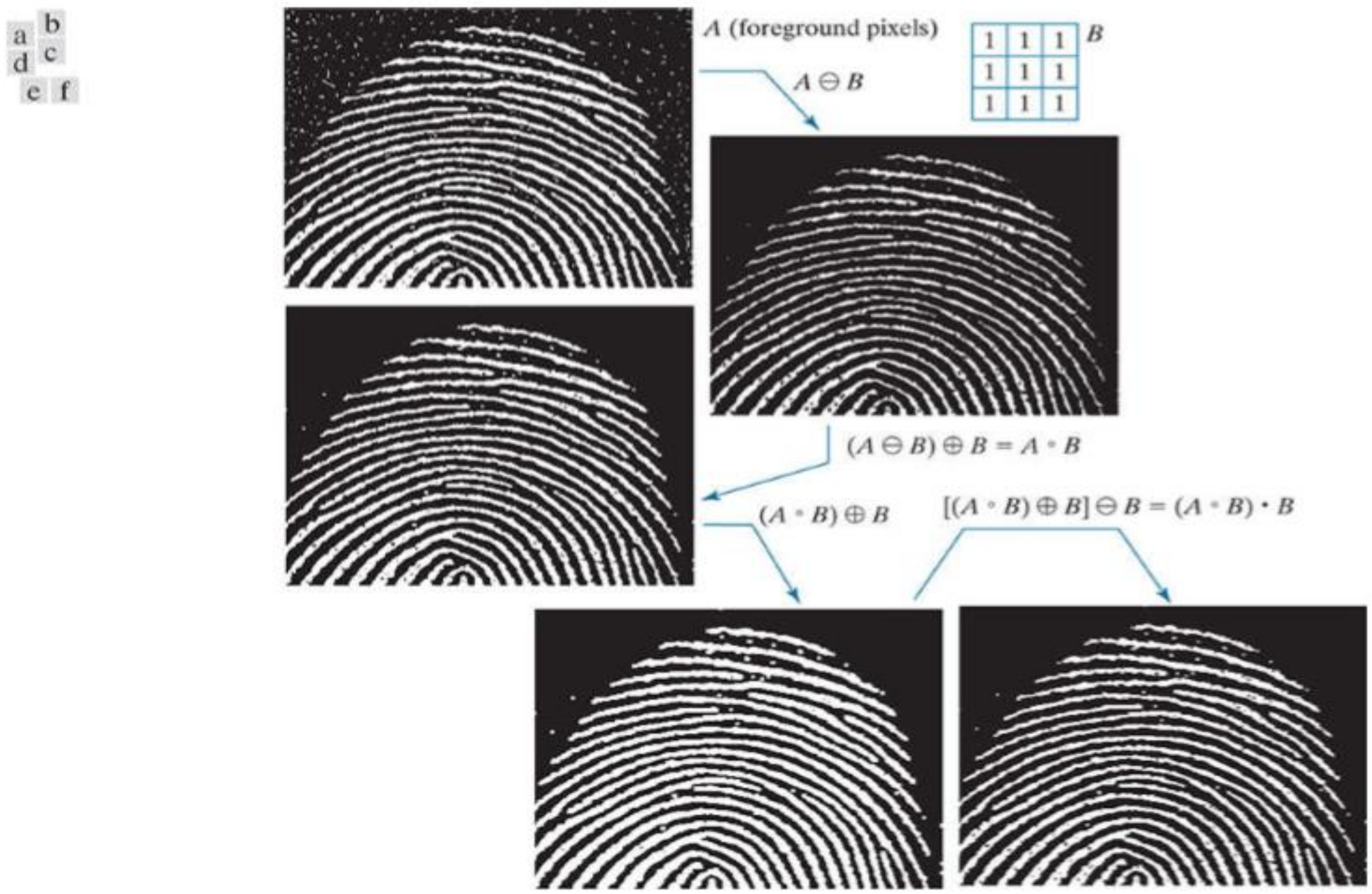


FIGURE9.11 (a) Noisy image. (b) Structuring element. (c) Eroded image. (d) Dilation of the erosion (opening of A). (e) Dilation of the opening. (f) Closing of the opening.

- Morphological operations can be used to construct filters similar in concept to the spatial filters.
- The binary image in Fig.9.11(a) shows a section of a fingerprint corrupted by noise.
- In terms of our previous notation, A is the set of all foreground (white) pixels, which includes objects of interest (the fingerprint ridges) as well as white specks of random noise.
- The background is black, as before. The noise manifests itself as white specks on a dark background and dark specks on the white components of the fingerprint.
- The objective is to eliminate the noise and its effects on the print, while distorting it as little as possible.

- A morphological filter consisting of an opening followed by a closing can be used to accomplish this objective.
- Figure 9.11(b) shows the structuring element we used.
- Figure 9.11(c) is the result of eroding A by B. The white speckled noise in the background was eliminated almost completely in the erosion stage of opening because in this case most noise components are smaller than the structuring element.
- The size of the noise elements (dark spots) contained within the fingerprint actually increased in size. The reason is that these elements are inner boundaries that increase in size as objects are eroded.
- This enlargement is countered by performing dilation on Fig.9.11(c) . Figure 9.11(d) shows the result.

- The two operations just described constitute the opening of A by B. We note in Fig.9.11(d) that the net effect of opening was to reduce all noise components in both the background and the fingerprint itself.
- However, new gaps between the fingerprint ridges were created. To counter this undesirable effect, we perform a dilation on the opening, as shown in Fig. 9.11(e)
- Most of the breaks were restored, but the ridges were thickened, a condition that can be remedied by erosion. The result, shown in Fig. 9.11(f) , is the closing of the opening of Fig.9.11(d)
- This final result is remarkably clean of noise specks, but it still shows some specks of noise that appear as single pixels.

The Hit-Or-Miss Transform

- The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image.
- The morphological hit-or-miss transform (HMT) is a basic tool for **shape detection**.
- Let I be a binary image composed of foreground (A) and background pixels, respectively.
- Unlike the morphological methods discussed thus far, the HMT utilizes two structuring elements:
- B_1 = for detecting shapes in the foreground
- B_2 = for detecting shapes in the background

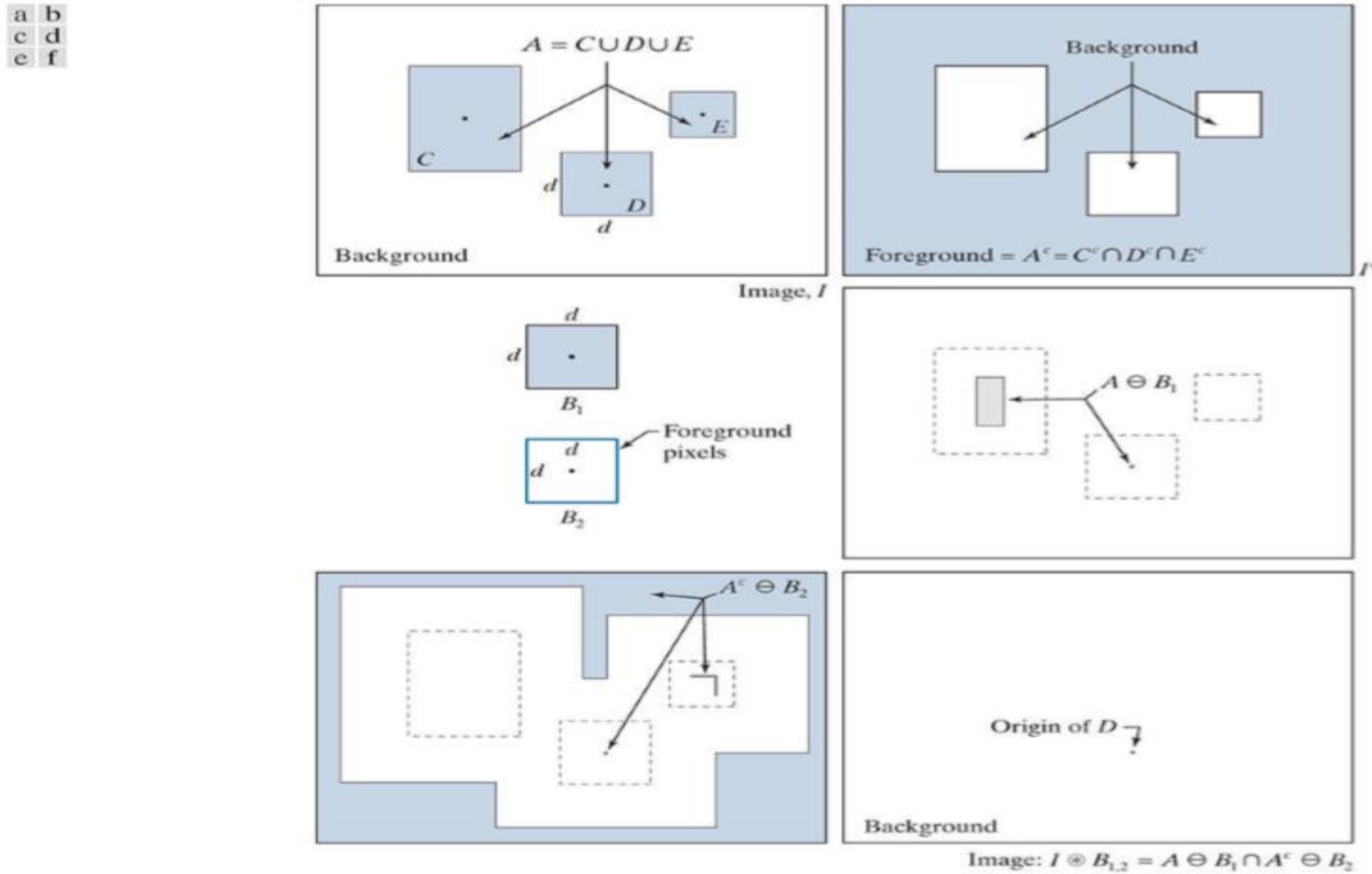


FIGURE 9.12 (a) Image consisting of a foreground (1's) equal to the union, A, of set of objects, and a background of 0's. (b) Image with its foreground defined as A^c (c) Structuring elements designed to detect object D. (d) Erosion of A by B1 (e) Erosion of A^c by B2 (f) Intersection of (d) and (e), showing the location of the origin of D, as desired. The dots indicate the origin of their respective components. Each dot is a single pixel

- The HMT of image I is defined as

$$I \circledast B_{1,2} = \{z \mid (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c\} \qquad (9\text{-}16)$$
$$= (A \ominus B_1) \cap (A^c \ominus B_2)$$

- In words, this equation says that the morphological HMT is the set of translations, z, of structuring elements B1 and B2 such that, simultaneously, B1 found a match in the foreground (i.e., B1 is contained in A) and B2 found a match in the background (i.e.,B2 is contained in A to the power c A^c i).
- The word "simultaneous" implies that z is the same translation of both structuring elements.
- The word "miss" in the HMT arises from the fact that B2 finding a match in A to the power c A^c is the same as B2 not finding (missing) a match in A.

- Fig. 9.12 which show a set A consisting of three shapes(subsets), denoted C,D and E

- Figure 9.12(a) shows that I is composed of foreground (A) and background pixels.
- Figure9.12(b) is the complement of I. The foreground of is defined as the set of pixels in and the background is the union of the complement of the three objects.
- Figure 9.12(c) shows the two structuring elements needed to detect D. Element is equal to D itself.
- Fig.9.12(d) shows, the erosion of A by contains a single point: the origin of D, as desired, but it also contains parts of object C.

- we define a structuring element, B, identical to D, but having in addition a border of background elements with a width of one pixel. We can use a structuring element formed in such a way to restate the HMT as

$$I \circledast B = \{z \mid (B)_z \subseteq I\} \tag{9-17}$$

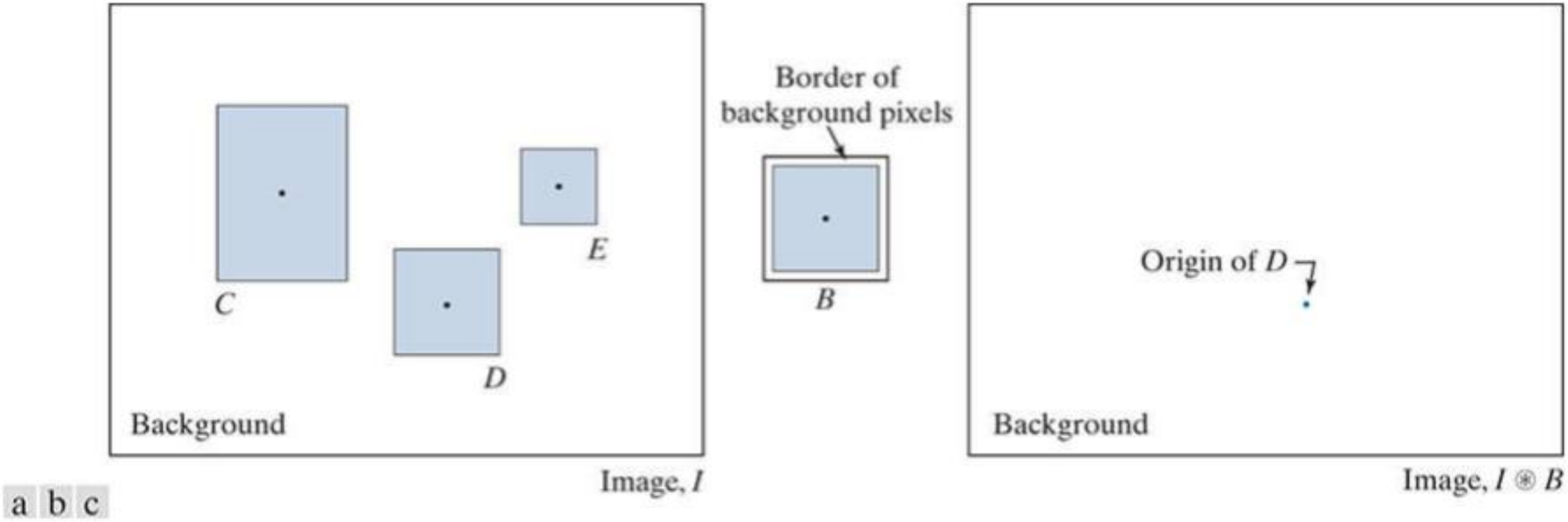


FIGURE 9.13 Same solution as in Fig. 9.12 , but using Eq.(9-17) with a single structuring element.

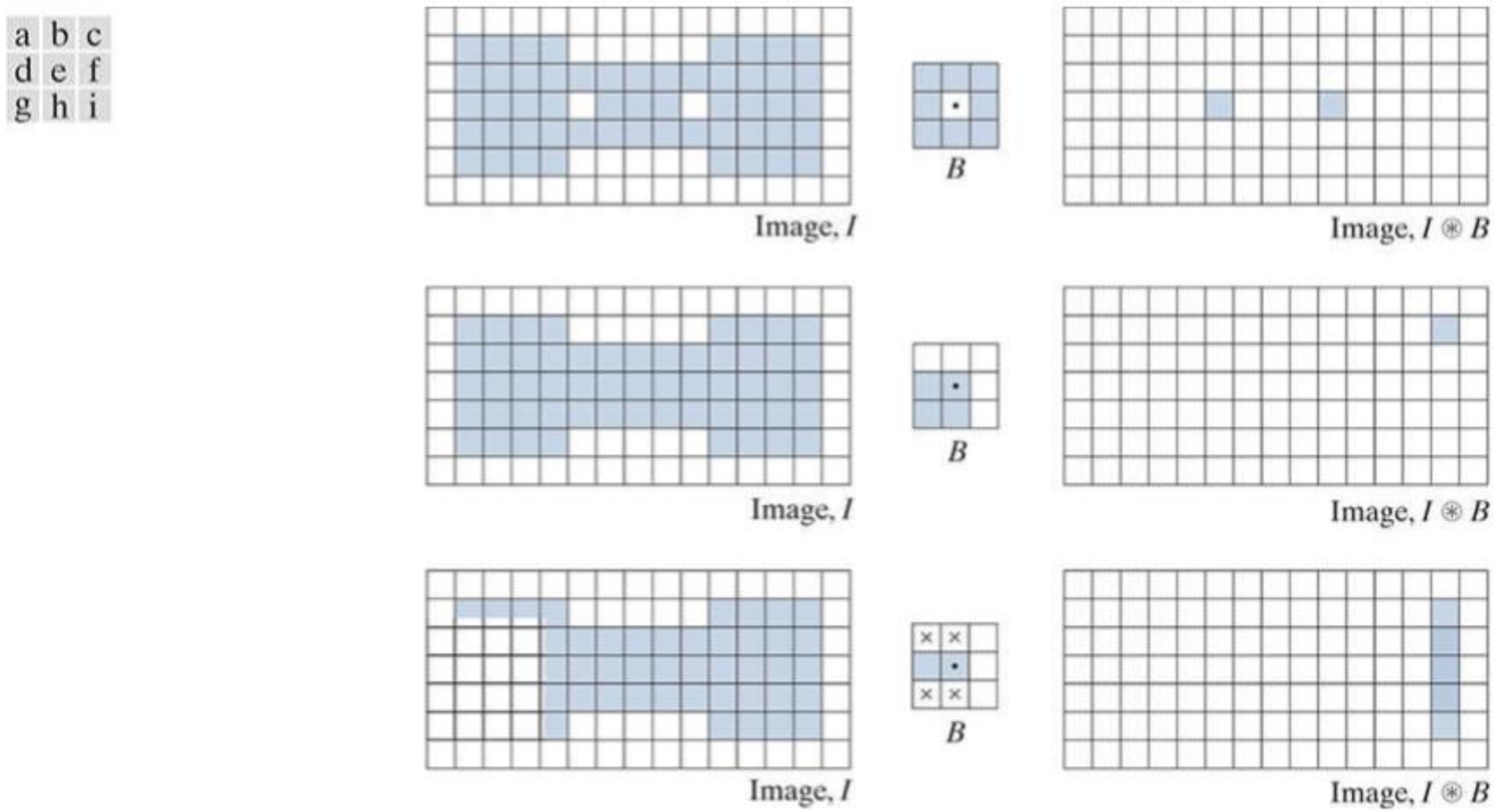
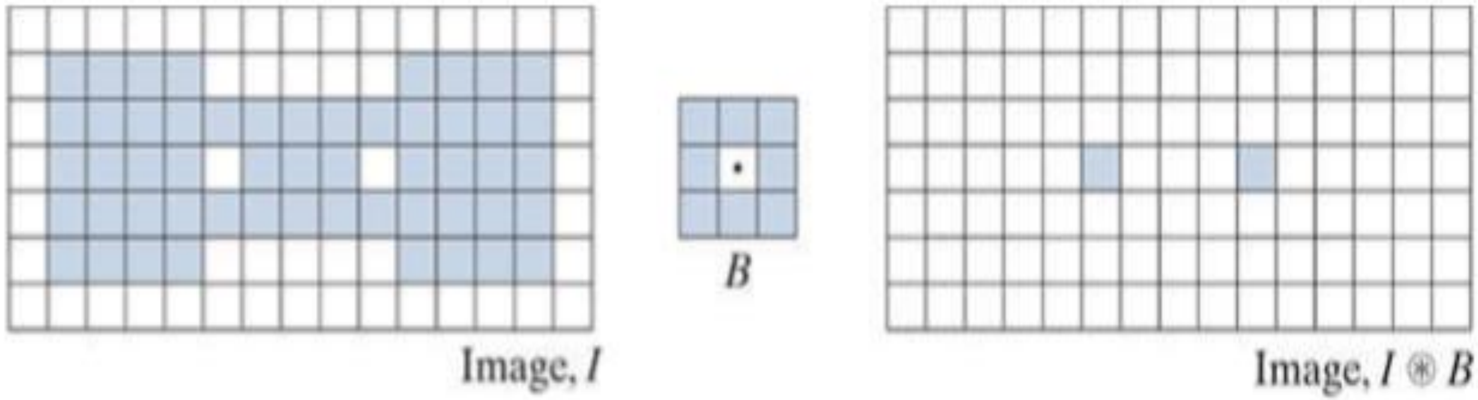
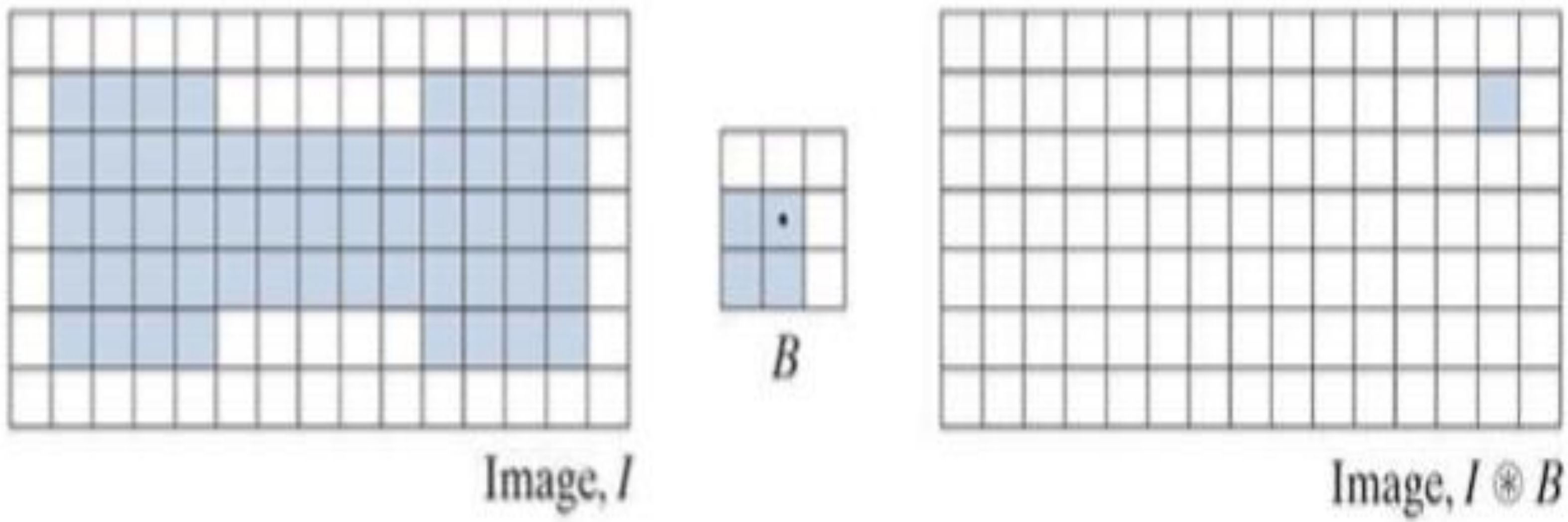


FIGURE 9.14
Three examples of using a single structuring element and Eq. (9-17) to detect specific features. First row: detection of single-pixel holes. Second row: detection of an upper-right corner. Third row: detection of multiple features.

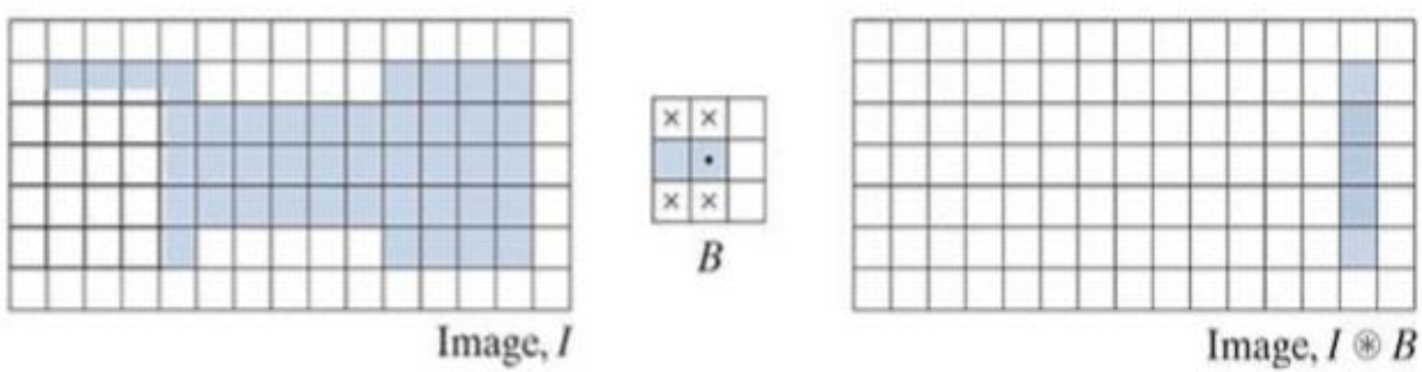


- The first row shows the result of using a small SE composed of both foreground (shaded) an background elements.
- This SE is designed to detect one-pixel holes (i.e., one background pixel surrounded by a connected border of foreground pixels) contained in image I.



- The SE in the second row is capable of detecting the foreground corner pixel of the top, right corner of the object in I.
- Using this SE in Eq.(9-17) yielded the image on the right. As you can see, the correct pixel was identified.

$$I \circledast B = \{z \mid (B)_z \subseteq I\} \qquad (9\text{-}17)$$



- The last row of Fig.9.14 is more interesting, as it shows a structuring element composed of foreground, background, and “don’t care” elements which, as mentioned earlier, we denote by X’s.
- You can think of the value of a don’t care element as always matching its corresponding pixel in an image.

- In this example, when the SE is centered on the top, right corner pixel, the don't care elements in the top of the SE can be considered to be background, and the don't care elements on the bottom row as foreground, producing a correct match.
- When the SE is centered on the bottom, right corner pixel, the role of the don't care elements is reversed, again resulting in a correct match.
- The other border pixels between the two corners were similarly detected by considering all don't care elements as foreground.
- Thus, using don't care elements increases the flexibility of structuring elements to perform multiple roles.

Some Basic Morphological Algorithms

- When dealing with binary images, one of the principal applications of morphology is in extracting image components that are useful in the representation and description of shape.
- In particular, we consider morphological algorithms for extracting boundaries, connected components, the convex hull, and the skeleton of a region.
- We also develop several methods (for region filling, thinning, thickening, and pruning) that are used frequently for pre- or post-processing.
- We make extensive use in this section of "mini-images," designed to clarify the mechanics of each morphological method as we introduce it.
- These binary images are shown graphically with foreground (1's) shaded and background (0's) in white, as before.

Boundary Extraction

- The boundary of a set A of foreground pixels, denoted by ß(A) can be obtained by first eroding A by a suitable structuring element B, and then performing the set difference between A and its erosion. That is

$$\beta(A) = A - (A \ominus B) \qquad (9\text{-}18)$$

- Figure 9.15 illustrates the mechanics of boundary extraction. It shows a simple binary object, a structuring element B, and the result of using Eq. (9-18)
- The structuring element in Fig.9.15(b) is among the most frequently used, but it is not unique.

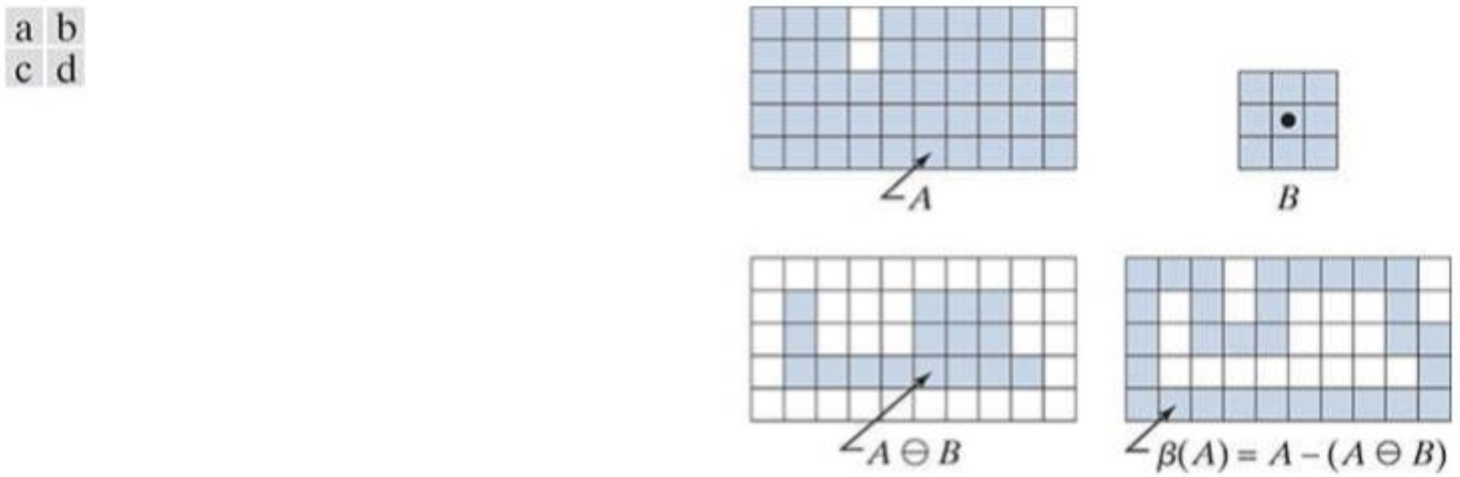


FIGURE 9.15 (a) Set, A, of foreground pixels. (b) Structuring element. (c) A eroded by B. (d) Boundary of A.

For example, using a structuring element of 1's would result in a boundary between 2 and 3 pixels thick.
It is understood that the image in Fig. 9.15(a) was padded with a border of background elements, and that the results were cropped back to the original size after the morphological operations were completed.

EXAMPLE 9.5: BOUNDARY EXTRACTION.

a b

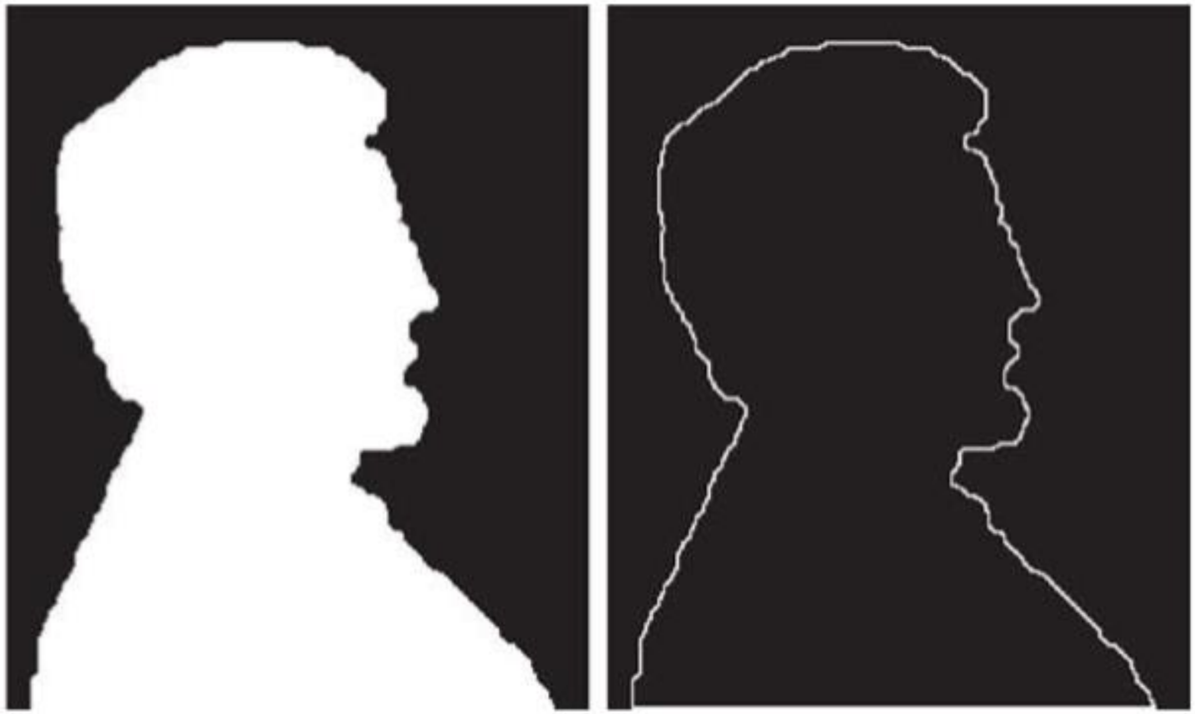


FIGURE9.16 (a) A binary image. (b) Result of using Eq. (9-18) with the structuring element in Fig.9.15(b)

$$\beta(A) = A - (A \ominus B) \tag{9-18}$$

- Figure 9.16 further illustrates the use of Eq. (9-18) using a structuring element of 1’s.
- As before when working with images, we show foreground pixels (1’s) in white and background pixels (0’s) in black. The elements of the SE, which are 1’s, also are treated as white. Because of the size of the structuring element used, the boundary in Fig. 9.16(b) is one pixel thick.

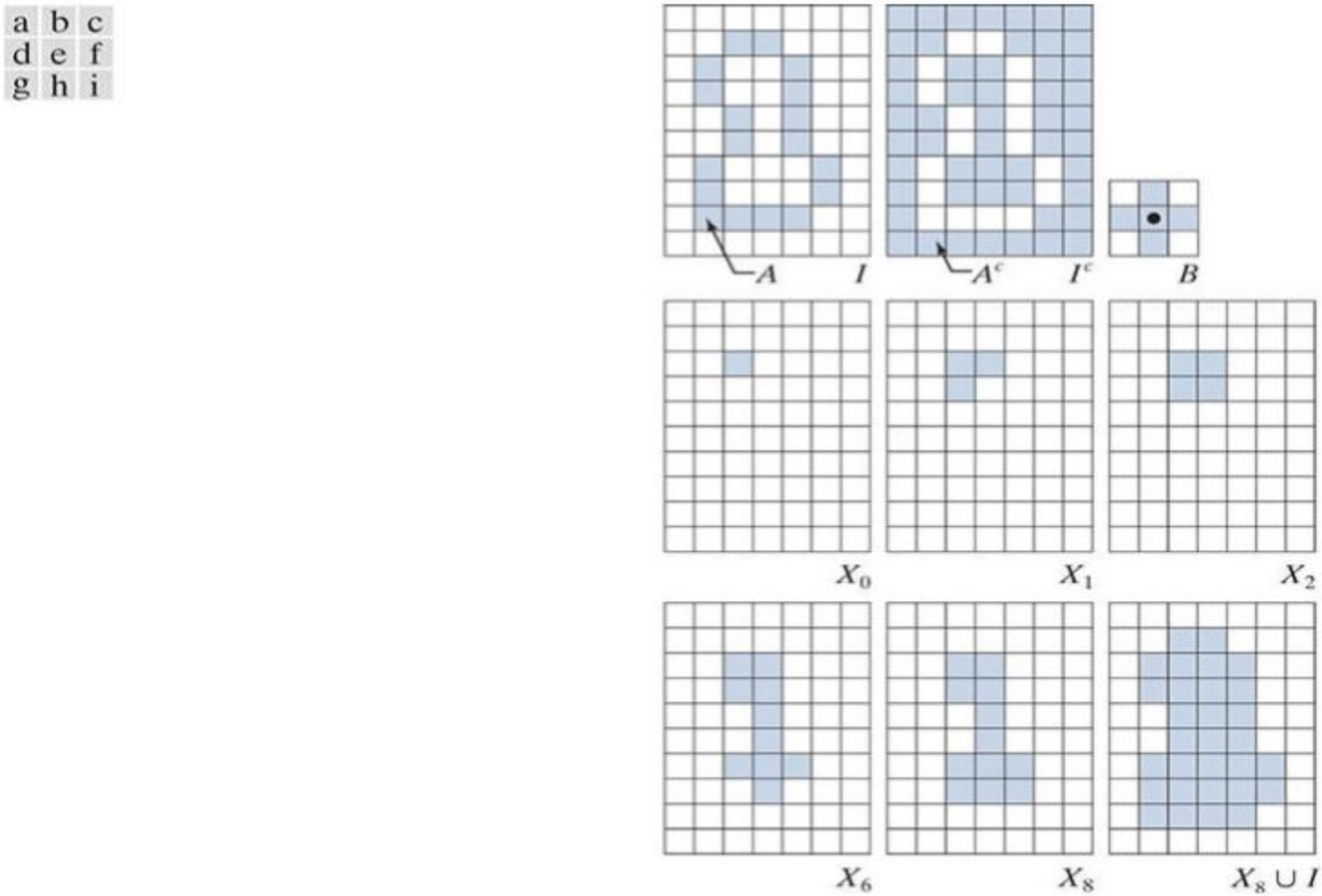
Hole Filling

- 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, with 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 foreground elements (1's).

- We begin by forming an array, Xo of 0's (the same size as I, the image containing A), except at locations in Xo that correspond to pixels that are known to be holes, which we set to 1. Then, the following procedure fills all the holes with 1's.

$$X_k = (X_{k-1} \oplus B) \cap I^c \quad k = 1, 2, 3, \ldots \tag{9-19}$$

- where B is the symmetric structuring element in Fig.9.17(c) . The algorithm terminates at iteration step k if $X_k = X_{k-1}$ Then, X_k contains all the filled holes.
- The set union of X_k and I contains all the filled holes and their boundaries



- FIGURE 9.17 Hole filling. (a) Set A (shown shaded) contained in image I. (b) Complement of I. (c) Structuring element B. Only the foreground elements are used in computations (d) Initial point inside hole, set to 1. (e)–(h) Various steps of Eq. (9-19) . (i) Final result [union of (a) and (h)].

- The dilation in Eq.(9-19) would fill the entire area if left unchecked, but the intersection at each step with 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, the process is appropriately called conditional dilation. The rest of Fig.9.17 illustrates further the mechanics of Eq. (9-19) .

EXAMPLE 9.6: MORPHOLOGICAL HOLE FILLING.

a b

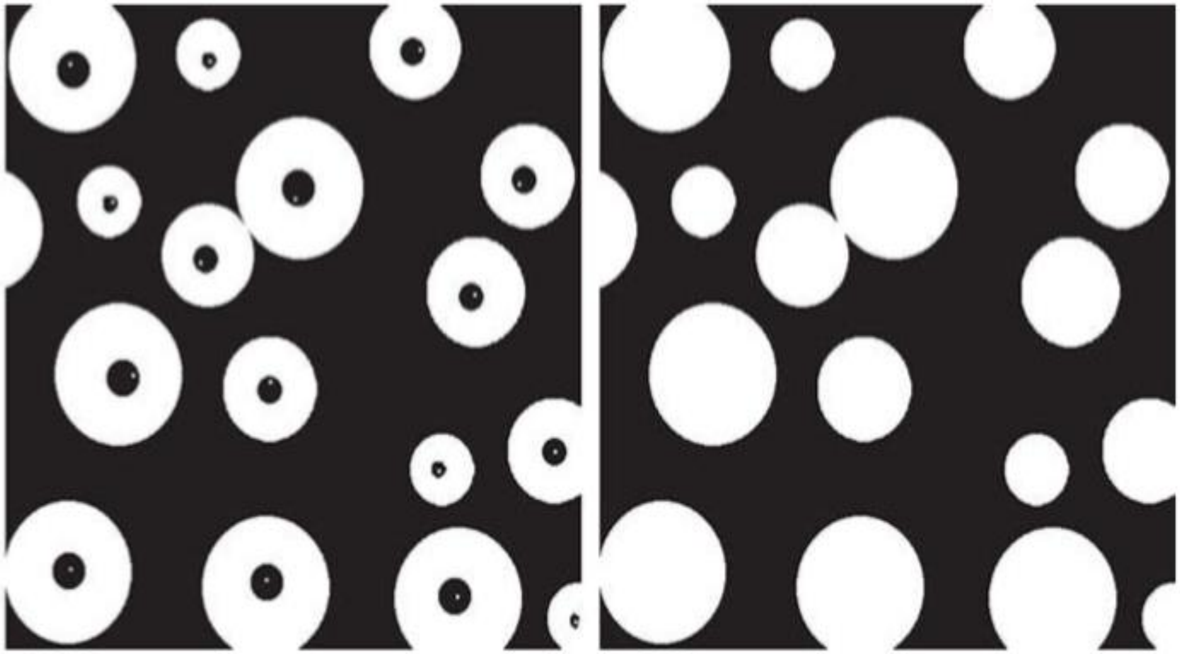


FIGURE9.18 (a) Binary image. The white dots inside the regions (shown enlarged for clarity) are the starting points for the hole-filling algorithm. (b) Result of filling all holes

- Figure 9.18(a) shows an image of white circles with black holes. An image such as this might result from thresholding into two levels a scene containing polished spheres (e.g., ball bearings). The dark circular areas inside the spheres would result from reflections. The objective is to eliminate the reflections by filling the holes in the image.
- Figure 9.18(b) shows the result of filling all the spheres. Because it must be known whether black points are background points or sphere inner points (i.e., holes), fully automating this procedure requires that additional “intelligence” be built into the algorithm

Extraction of Connected Components

- Being able to extract connected components from a binary image is central to many automated image analysis applications.
- Let A be a set of foreground pixels consisting of one or more connected components, and form an image X_0 (of the same size as I, the image containing A) whose elements are 0's (background values), except at each location known to correspond to a point in each connected component in A, which we set to 1 (foreground value).
- The objective is to start with and find all the connected components in I. The following iterative procedure accomplishes this:

$$X_k = (X_{k-1} \oplus B) \cap I \quad k = 1, 2, 3, \ldots \tag{9-20}$$

a
b c d
e f g

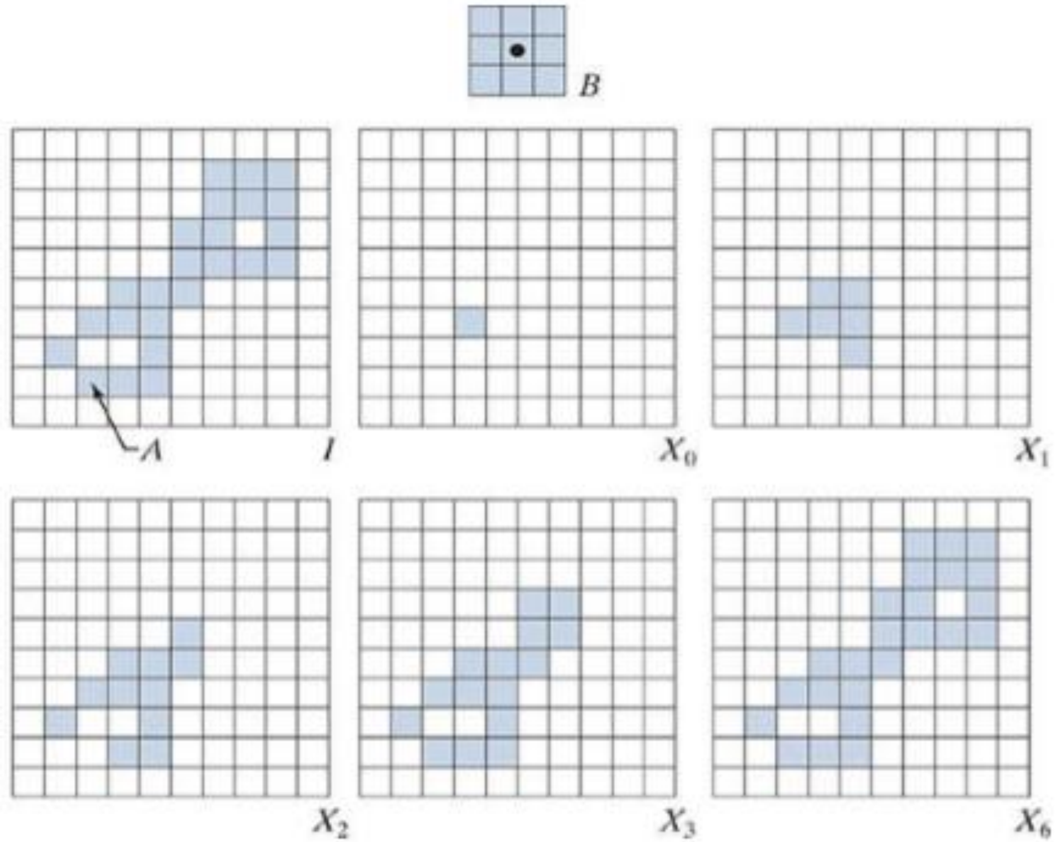
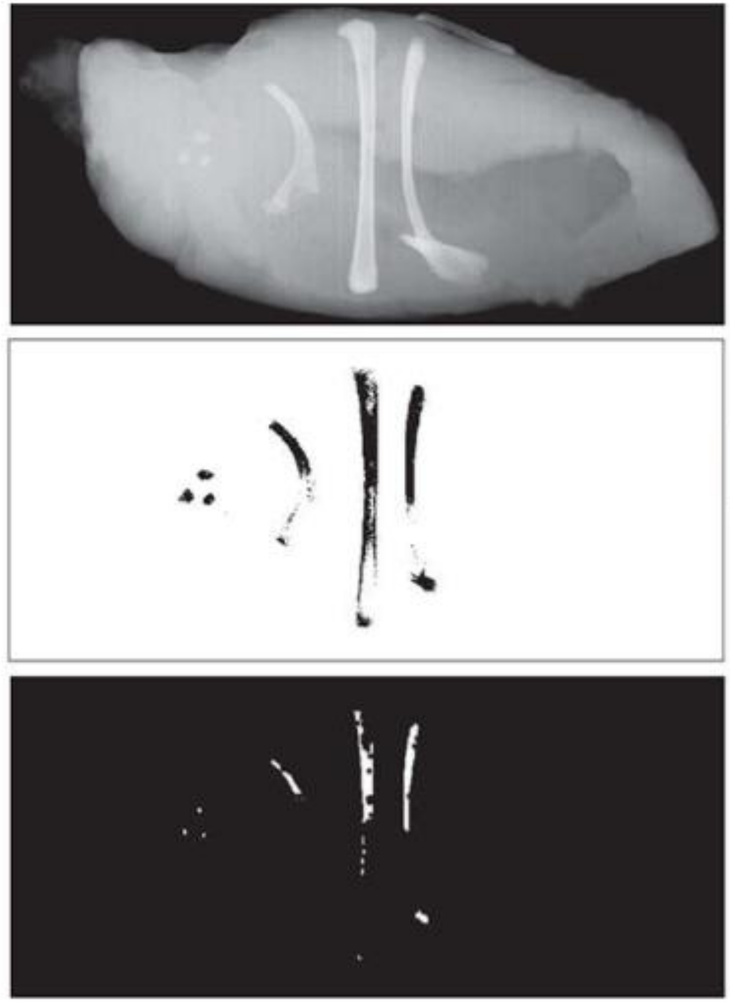


FIGURE 9.19 (a) Structuring element. (b) Image containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9-20)

- where B is the SE in Fig.9.19(a) . The procedure terminates when $X_k = X_{k-1}$, with X_k containing all the connected components of foreground pixels in the image.
- Both Eqs.(9-19) and (9-20) use conditional dilation to limit the growth of set dilation, but Eq. (9-20) uses I instead of Ic.
- This is because here we are looking for foreground points, while the objective of (9-19) is to find background points. Figure 9.19 illustrates the mechanics of Eq. (9-20), with convergence being achieved for k=6 Note that the shape of the structuring element used is based on 8-connectivity between pixels.
- As in the hole-filling algorithm, Eq. (9-20) is applicable to any finite number of connected components contained in I.

EXAMPLE 9.7: USING CONNECTED COMPONENTS TO DETECT FOREIGN OBJECTS IN PACKAGED FOOD.

a
b
c d



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

FIGURE9.20 (a) X-ray image of a chicken filet with bone fragments. (b) Thresholded image (shown as the negative for clarity). (c) Image eroded with a SE of 1’s. (d) Number of pixels in the connected components of (c).

- Connected components are used frequently for automated inspection. Figure9.20(a) shows an X-ray image of a chicken breast that contains bone fragments.
- It is important to be able to detect such foreign objects in processed foods before shipping. In this application, the density of the bones is such that their nominal intensity values are significantly different from the background.
- This makes extraction of the bones from the background a simple matter by using a single threshold (thresholding was introduced in Section 3.1 and we will discuss in more detail in Section 10.3). The result is the binary image in Fig.9.20(b)

- The most significant feature in this figure is the fact that the points that remain after thresholding are clustered into objects (bones), rather than being scattered.
- We can make sure that only objects of “significant” size are contained in the binary image by eroding its foreground.
- In this example, we define as significant any object that remains after erosion with a 5x5 SE of 1’s. Figure 9.20(c) shows the result of erosion. The next step is to analyze the size of the objects that remain. We label (identify) these objects by extracting the connected components in the image.
- The table in Fig. 9.20(d) lists the results of the extraction. There are 15 connected components, with four of them being dominant in size. This is enough evidence to conclude that significant, undesirable objects are contained in the original image

Convex Hull

- A set, S, of points in the Euclidean plane is said to be convex if and only if a straight line segment joining any two points in S lies entirely within S.
- The convex hull, H, of S is the smallest convex set containing S.
- The convex deficiency of S is defined as the set difference H-S.
- Unlike the Euclidean plane, the digital image plane (see Fig. 2.19) only allows points at discrete coordinates.
- Thus, the sets with which we work are digital sets. The same concepts of convexity are applicable to digital sets, but the definition of a convex digital set is slightly different.

- A digital set, A, is said to be convex if and only if its Euclidean convex hull only contains digital points belonging to A.
- A simple way to visualize if a digital set of foreground points is convex is to join its boundary points by straight (continuous) Euclidean line segments.
- If only foreground points are contained within the set formed by the line segments, then the set is convex; otherwise it is not.
- The definitions of convex hull and convex deficiency given above for S, extend directly to digital sets.
- The following morphological algorithm can be used to obtain an approximation of the convex hull of a set A of foreground pixels, embedded in a binary image, I.

- Let B^i, $i = 1, 2, 3, 4$, denote the four structuring elements in Fig. 9.21(a) . The procedure consists of implementing the morphological equation

$$X_k^i = (X_{k-1}^i \circledast B^i) \cup X_{k-1}^i \quad i = 1, 2, 3, 4 \text{ a n } k = 1, 2, 3, \dots \tag{9-21}$$

- With $X_0^i = I$. When the procedure converges using the ith structuring element ((i.e., when $X_k^i = X_{k-1}^i$), we let $D^i = X_k^i$. we let $D^i = X_k^i$. Then, the convex hull of A is the union of the four results,

$$C(A) = \bigcup_{i=1}^{4} D^i$$

- Thus, the method consists of iteratively applying the hit-or-miss transform to I with B^1 until convergence, then letting

$D^1 = X_k^1$, where k is the step at which convergence occurred.

- The procedure is repeated B^2 with (applied to I) until no further changes occur, and so on. The union of the four resulting D^i constitutes the convex hull of A. The algorithm is initialized with k=0 and $X_0^i = I$ every time that i (i.e., the structuring element) changes

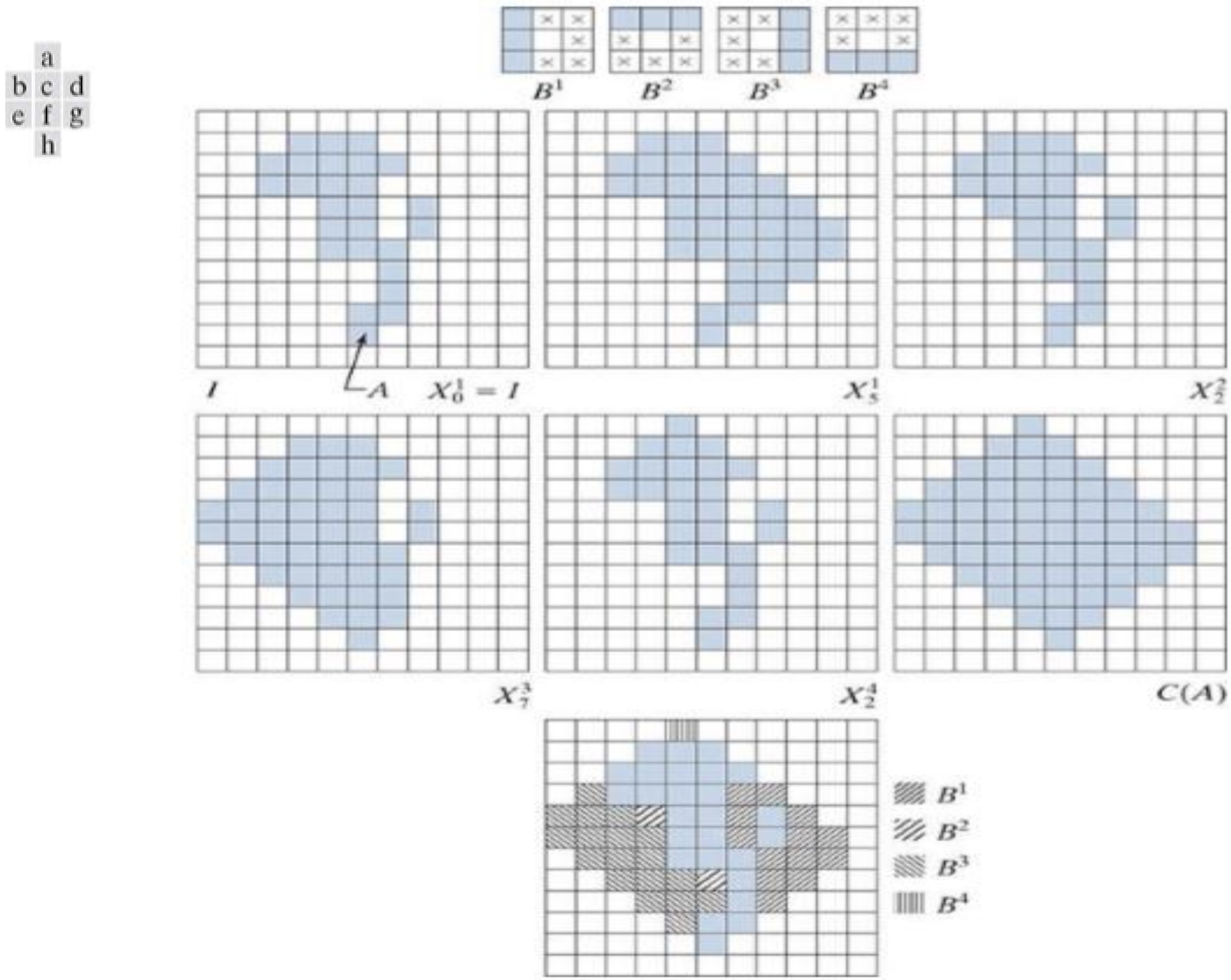


FIGURE 9.21 (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.

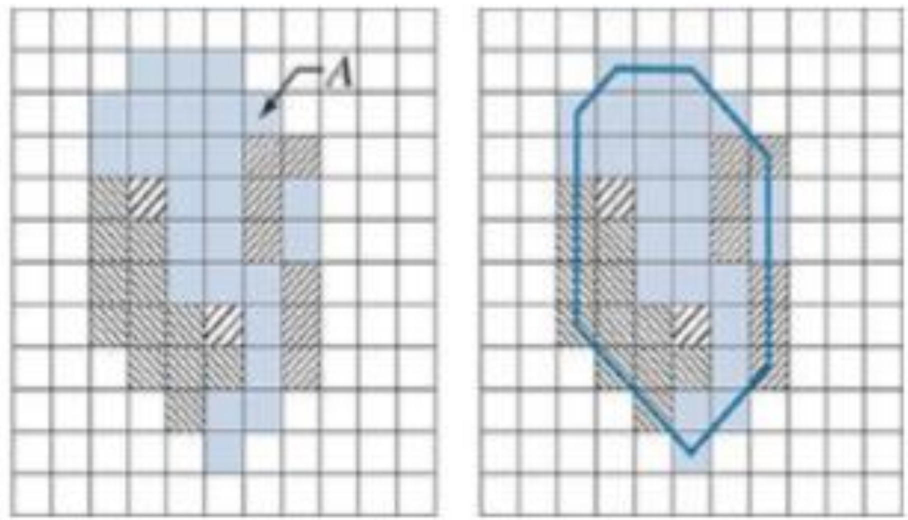


FIGURE 9.22 (a) Result of limiting growth of the convex hull algorithm. (b) Straight lines connecting the boundary points show that the new set is convex also

Thinning and Thickening

- Thinning is an image-processing operation in which binary valued image regions are reduced to lines.
- The purpose of thinning is to reduce the image components to their essential information for further analysis and recognition.
- Thickening is changing a pixel from 1 to 0 if any neighbors of the pixel are 1.
- Thickening followed by thinning can be used for filling undesirable holes.
- Thinning followed by thickening is used for determining isolated components and clusters

Thinning

- Thinning of a set A of foreground pixels by a structuring element B, denoted $A \otimes B$, can be defined in terms of the hit-or-miss transform

$$\begin{aligned} A \otimes B &= A - (A \circledast B) \\ &= A \cap (A \circledast B)^c \end{aligned} \qquad (9\text{-}23)$$

- where the second line follows from the definition of set difference given in Eq.(2-40) . A more useful expression for thinning A symmetrically is based on a sequence of structuring elements:

$$\{B\} = \{B^1, B^2, B^3, \ldots, B^n\} \qquad (9\text{-}24)$$

- Using this concept, we now define thinning by a sequence of structuring elements as

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n) \qquad (9\text{-}25)$$

- 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 after one complete pass through all structuring elements. Each individual thinning pass is performed using Eq. (9-23)
- Figure 9.23(a) shows a set of structuring elements used routinely for thinning (note that B^i is equal to B^{i-1} rotated clockwise by and Fig.9.23(b) shows a set A to be thinned, using the procedure just discussed. Figure 9.23(c) shows the result of thinning A with one pass B^1 of to obtain A_1.
- Figure9.23(c) is the result of thinning A_1 with B^2 and Figs.9.21(e) through (k) show the results of passes with the remaining structuring elements (there were no changes from A_7 to A_8 or from A_9 to A_{11} Convergence was achieved after the second pass of B^6Figure 9.23(l) shows the thinned result.

- Finally, Fig.9.23(m) shows the shows the thinned set converted to m-connectivity (see Section 2.5 and Problem9.30) to eliminate multiple paths.

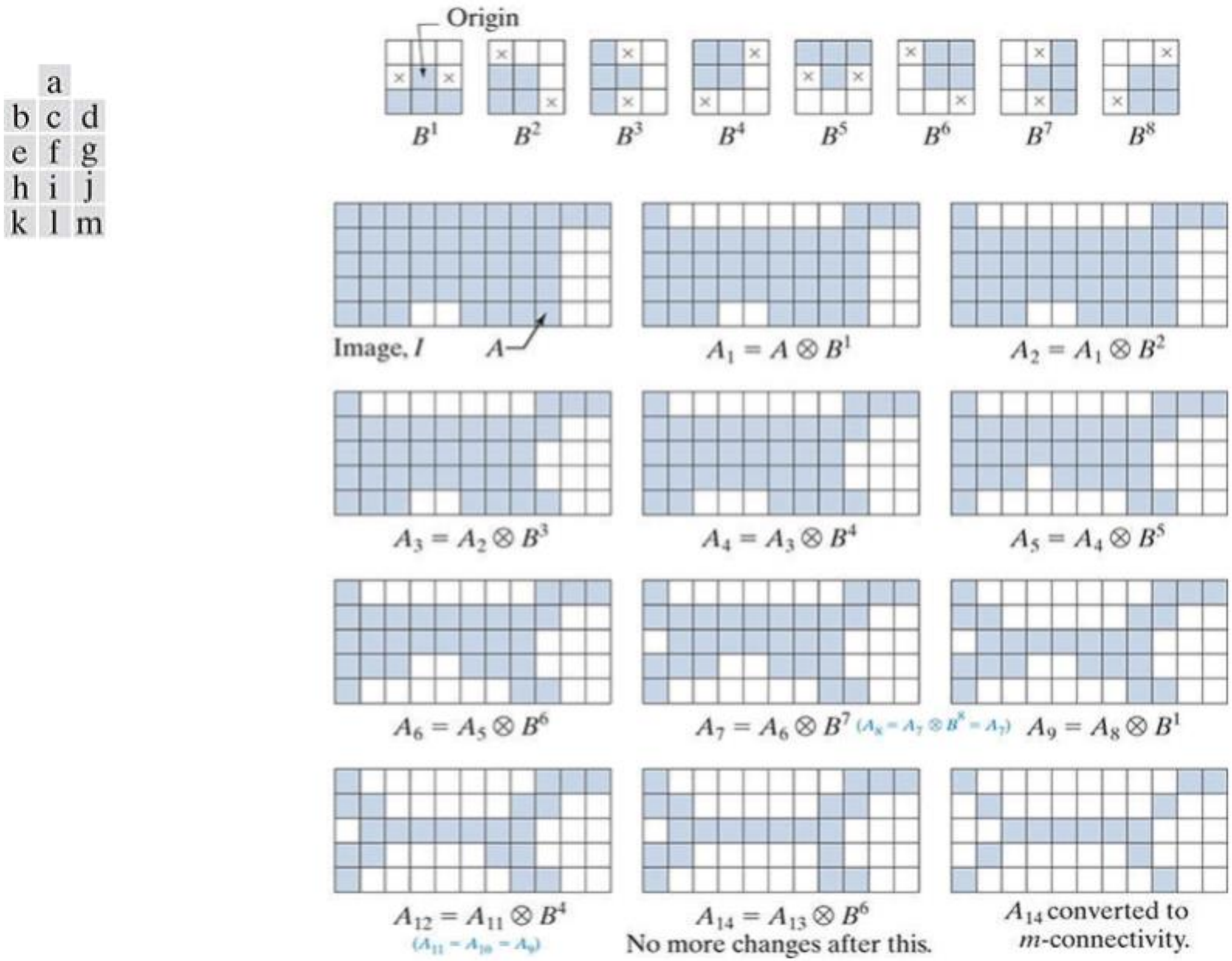


FIGURE 9.23 (a) Structuring elements. (b) Set A. (c) Result of thinning A with (shaded). (d) Result of thinning with (e)–(i) Results of thinning with the next six SEs. (There was no change between and (j)–(k) Result of using the first four elements again. (l) Result after convergence. (m) Result converted to m-connectivity. .

THICKENING

- Thickening is the morphological dual of thinning and is defined by the expression

$$A \odot B = A \cup (A \circledast B) \tag{9-26}$$

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

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n) \tag{9-27}$$

- The structuring elements used for thickening have the same form as those shown in Fig.9.23(a) , but with all 1's and 0's interchanged.
- However, a separate algorithm for thickening is seldom used in practice. Instead, the usual procedure is to thin the background of the set in question, then complement the result.

- In other words, to thicken a set A we form A^c thin A^c and then complement the thinned set to obtain the thickening of A. Figure9.24 illustrates this procedure. As before, we show only set A and image I, and not the padded version of I.

a b
c d
e

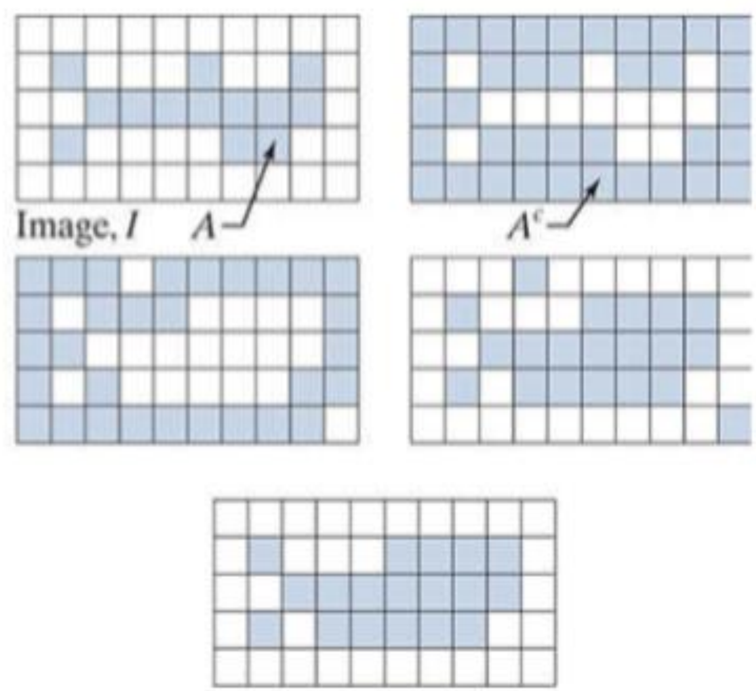


FIGURE 9.24 (a) Set A. (b) Complement of A. (c) Result of thinning the complement. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

- Depending on the structure of A, this procedure can result in disconnected points, as Fig.9.24(d) shows. Hence thickening by this method usually is followed by postprocessing to remove disconnected points.
- Note from Fig.9.24(c) that the thinned background forms a boundary for the thickening process.
- This useful feature is not present in the direct implementation of thickening using Eq. (9-27) , and it is one of the principal reasons for using background thinning to accomplish thickening.

Pruning

- Pruning methods are an essential complement to thinning and skeletonizing algorithms, because these procedures tend to leave spurs (“parasitic” components) that need to be “cleaned up” by post processing.
- We begin the discussion with a pruning problem, then develop a solution based on the material introduced in the preceding sections.
- Thus, we take this opportunity to illustrate how to solve a problem by combining several of the morphological techniques discussed up to this point.
- A common approach in the automated recognition of hand-printed characters is to analyze the shape of the skeleton of a character.

- These skeletons often contain spurs, caused during erosion by noise and non-uniformities in the character strokes.
- In this section we develop a morphological technique for handling this problem, starting with the assumption that the length of a parasitic component does not exceed a specified number of pixels.
- Figure 9.27(a) shows the skeleton of a hand-printed letter “a.” The spur on the leftmost part of the character exemplifies what we are interested in removing.

- The solution is based on suppressing a spur branch by successively eliminating its end point. Of course, this also shortens (or eliminates) other branches in the character but, in the absence of other structural information, the assumption in this example is that any branch with three or less pixels is to be eliminated. Thinning of a set A, with a sequence of structuring elements designed to detect only end points, achieves the desired result. That is, let

$$X_1 = A \otimes \{B\} \tag{9-34}$$

- where denotes the structuring element sequence in Fig. 9.27(b) [see Eq. (9-24) regarding structuring-element sequences]. The sequence of structuring elements consists of two different structures, each of which is rotated for a total of eight elements. The in Fig. 9.27(b) signifies a "don't care" condition, as defined earlier. (Note that each SE is a detector for an end point in a particular orientation.)

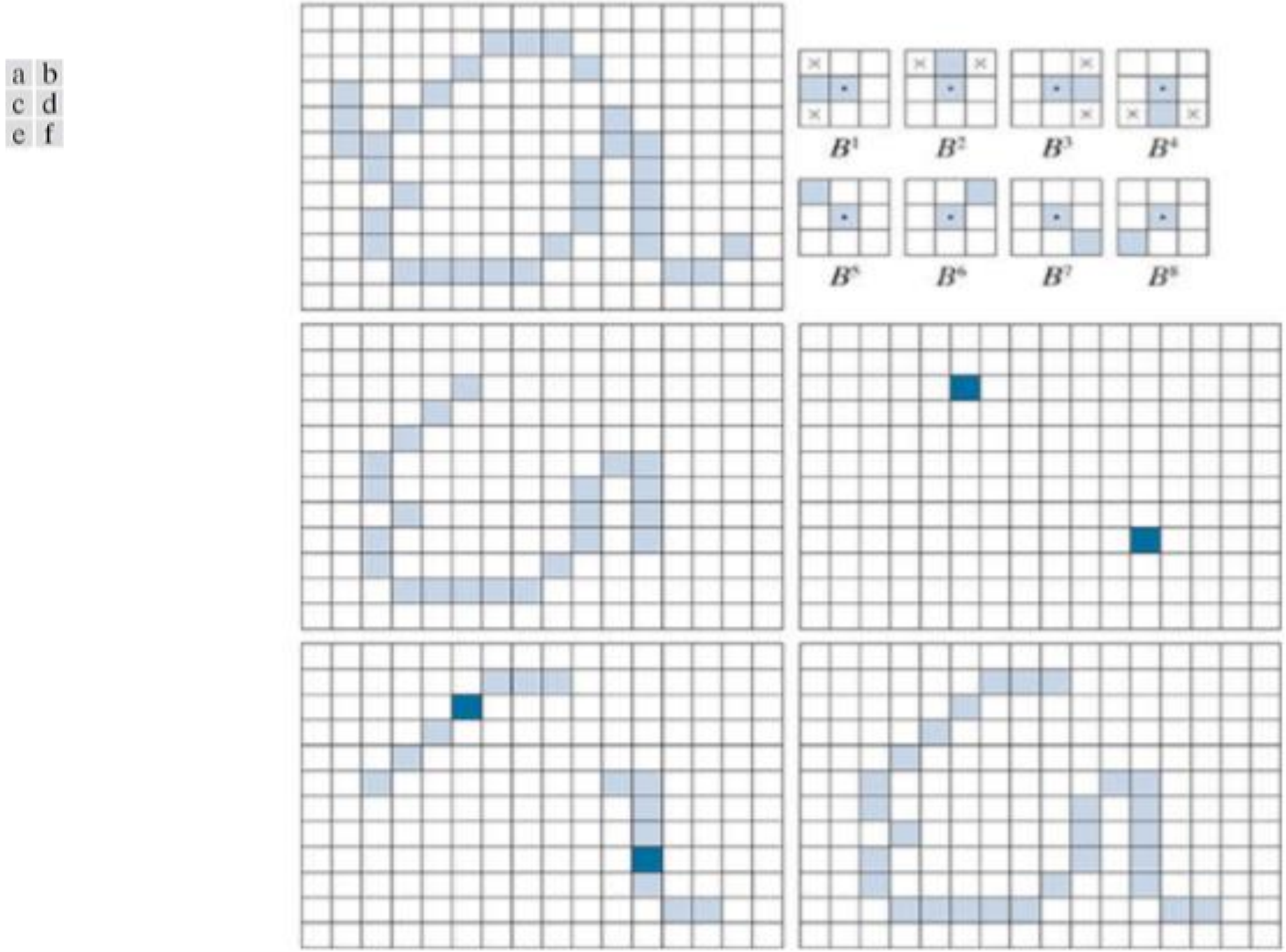


FIGURE 9.27 (a) Set A of foreground pixels (shaded). (b) SEs used for deleting end points. (c) Result of three cycles of thinning. (d) End points of (c). (e) Dilation of end points conditioned on (a). (f) Pruned image.

- Applying Eq.(9-34) to A three times yielded the set in Fig.9.27(c) . The next step is to "restore" the character to its original form, but with the parasitic branches removed. This requires that we first form a set X_2 Containing all end points in X_1 [Fig.9.27(e)]

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k) \tag{9-35}$$

- where the B^k are the end-point detectors in Fig.9.27(b) . The next step is dilation of the end points. Typically, the number of dilations is less than the number of end-point removals to reduce the probability of "growing" back some of the spurs. In this case, we know by inspection that no new spurs are created, so we dilate the end points three times using A as a delimiter.

- This is the same number of thinning passes:

$$X_3 = (X_2 \oplus H) \cap A \quad (9\text{-}36)$$

- where H is a 3x3 structuring element of 1's, and the intersection with A is applied after each step. As in the case of region filling, this type of conditional dilation prevents the creation of 1-valued elements outside the region of interest, as illustrated by the result in Fig. 9.27(e) . Finally, the union of X_1 and X_3

$$X_4 = X_1 \cup X_3 \quad (9\text{-}37)$$

- yields the desired result in Fig. 9.27(f)

- In more complex scenarios, using Eq. (9-36) sometimes picks up the “tips” of some branches. This can occur when the end points of these branches are near the skeleton.
- Although Eq. (9-36) may eliminate them, they can be picked up again during dilation because they are valid points in A. However, unless entire parasitic elements are picked up again (a rare case if these elements are short with respect to valid strokes), detecting and eliminating the reconstructed elements is easy because they are disconnected regions.
- A natural thought at this juncture is that there must be easier ways to solve this problem. For example, we could just keep track of all deleted points and simply reconnect the appropriate points to all end points left after application of Eq. (9-34)

- This argument is valid, but the advantage of the formulation just presented is that we used existing morphological co
- When a set of such tools is available, the advantage is that no new algorithms have to be written. We simply combine the necessary morphological functions into a sequence of operations.
- Sometimes you will encounter end point detectors based on a single structuring element, similar to the first SE in Fig.9.27(b) , but having “don’t care” conditions along the entire first column instead having a foreground element separating the corner This is incorrect. For example, the former element would identify the point located in the eighth row, fourth column of Fig.9.27(a) as an end point, thus eliminating it and breaking the connectivity of that part of the stroke.