Morphological Image Processing

Reference:

Digital Image Processing

Rafael C. Gonzalez, Richard E. Woods

Introduction

- What is morphology or morphological image processing?
  - describes a range of image processing techniques that deal with the shape (or morphology) of features in an image
  - Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on bi-level images

Example

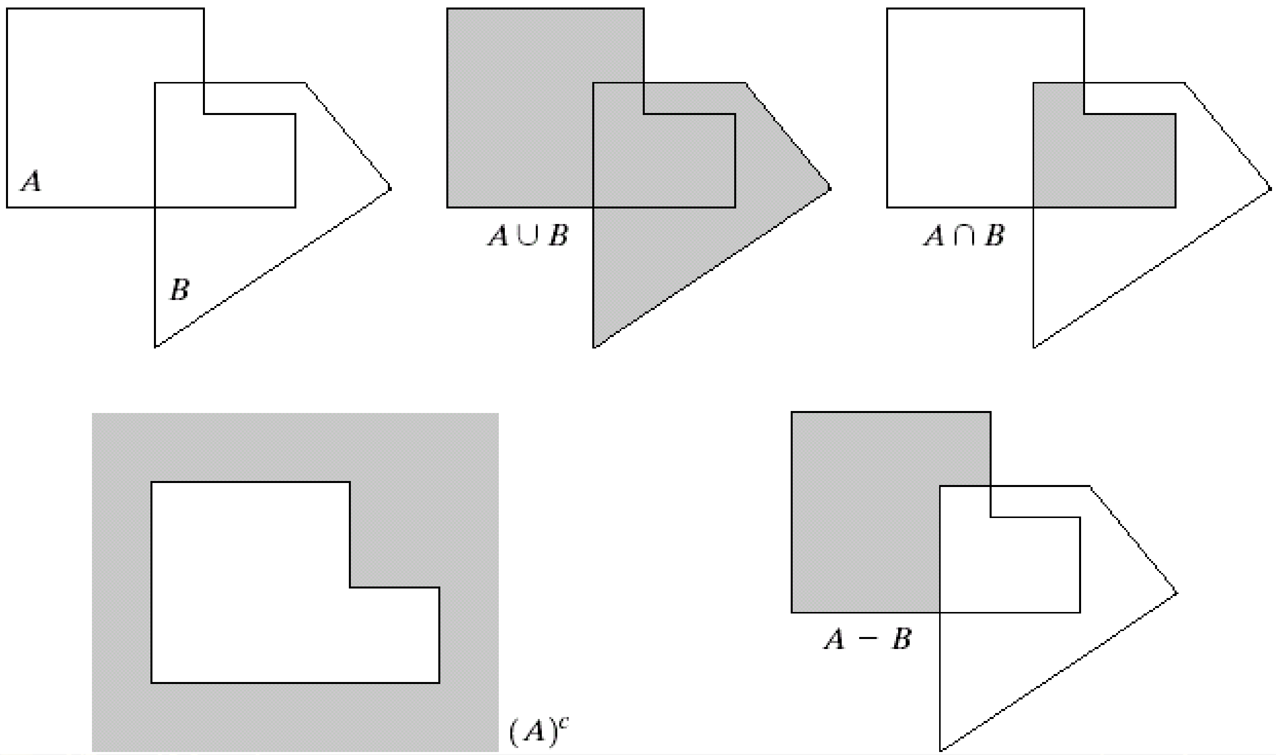
Introduction

- Morphological image processing is used to:
  - extract image components for representation and description of region shape, such as boundaries, skeletons, and the convex hull.
  - edge detection, noise removal, image enhancement and image segmentation.
- Morphological techniques probe an image with a small shape or template known as a **structuring element.**
  - The structuring element is positioned at all possible locations in the image.
  - It is compared with the corresponding neighborhood of pixels.
  - Morphological operations differ in how they carry out this comparison.

Basic Set Theory (1)

- If a=(a1,a2) is an element of A: $a \in A$
- If a is not an element of A : $a \notin A$
- The set with no elements is called the *null* or *empty set* and is denoted by the symbol $\varnothing$.
- If every element of a set A is also an element of another set B, then a is said to be a subset of B: $A \subseteq B$
- The **union** of two sets A and B is the set of all elements belonging to either A,B or both: $C=A \cup B$
- The **intersection** of two sets A and B is the set of all elements belonging to both A and B: $D=A \cap B$
- Two sets A and B are **disjoint or mutually exclusive** if they have no common elements. $A \cap B = \varnothing$

Basic Set Theory (1)

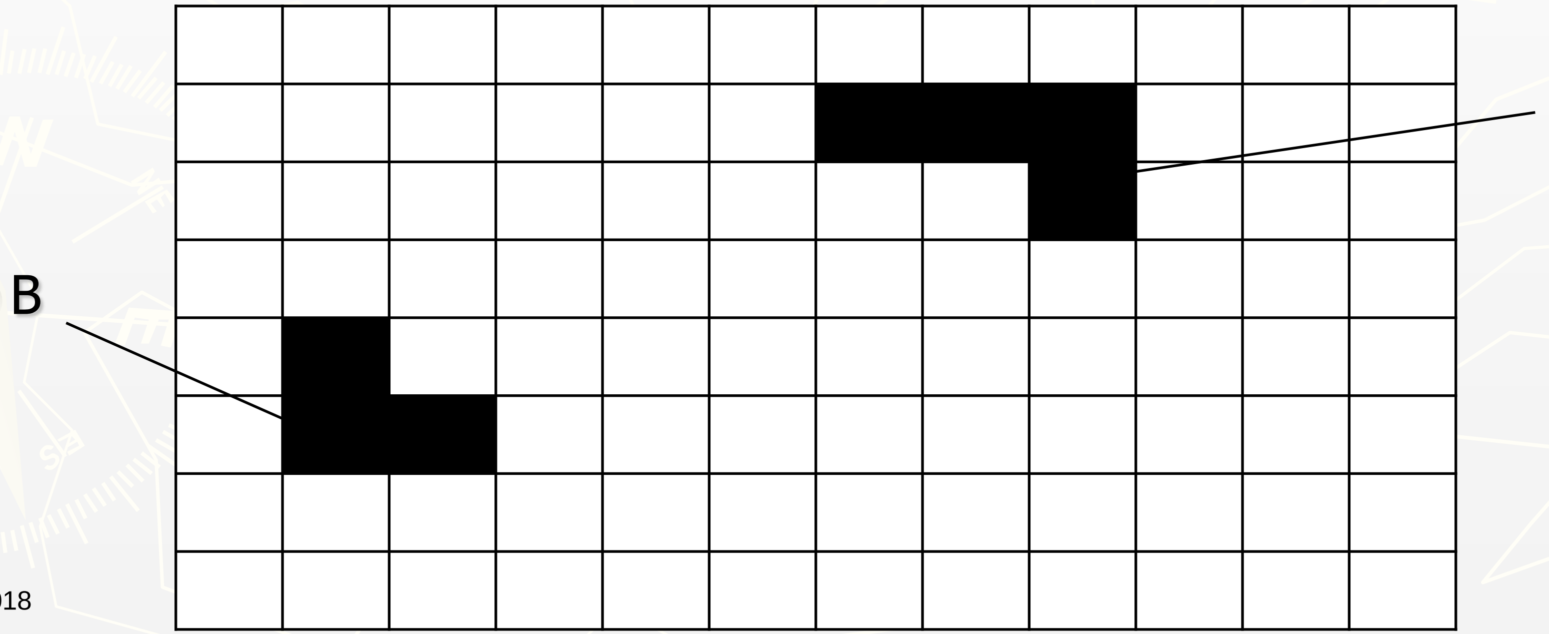


a b c
d e

FIGURE 9.1
(a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

Preliminaries (2)

- We define an image as:
  - an (amplitude) function of two discrete variables $a\,[m,n]$ or,
  - a set (or collection) of discrete coordinates. The set corresponds to the pixels belonging to objects in the image.



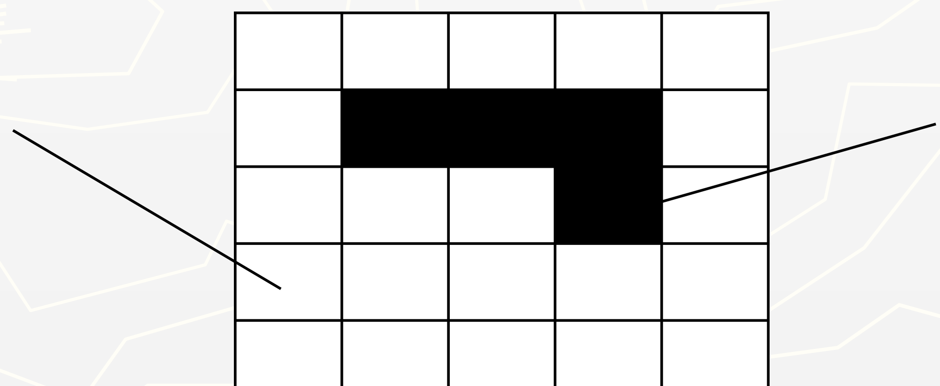
Preliminaries (3)

- The object ***A*** consists of those pixels α that share some common property:

$$A = \{\alpha \mid property(\alpha) == TRUE\}$$

- The background of ***A*** is given by ***A***^{c} (the *complement* of ***A***) which is defined as those elements that are not in ***A.***

$$A^c = \{\alpha \mid \alpha \notin A\}$$



Preliminaries (4)

- The **intersection** of any two binary images A and B , denoted by A $\cap$ B, is the binary image which is 1 at all pixels *p* which are 1 in both A and B:

$$A \cap B = \{p \mid p \in A \text{ and } p \in B\}$$

- The **union** of A and B, denoted by A$\cup$B, is the binary image which is 1 at all pixels p which are 1 in A or 1 in B or 1 in both:

$$A \cup B = \{p \mid p \in A \text{ or } p \in B\}$$

Preliminaries (5)

- **Reflection**

The reflection of a set B, denoted by $\widehat{B}$, is defined as,

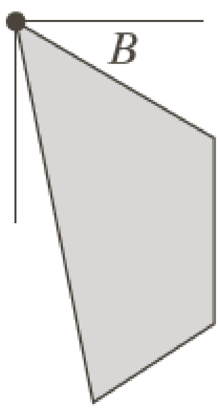
$$\widehat{B} = \{w \mid w = -b, \text{ for } b \in B\}$$

- **Translation**

The translation of a set B by point $z = (z_1, z_2)$, denoted $(B)_Z$, is defined as

$$(B)_Z = \{c \mid c = b + z, \text{for } b \in B\}$$

Example: Reflection and Translation



a b c

FIGURE 9.1
(a) A set, (b) its reflection, and (c) its translation by z.

Logic Operations

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

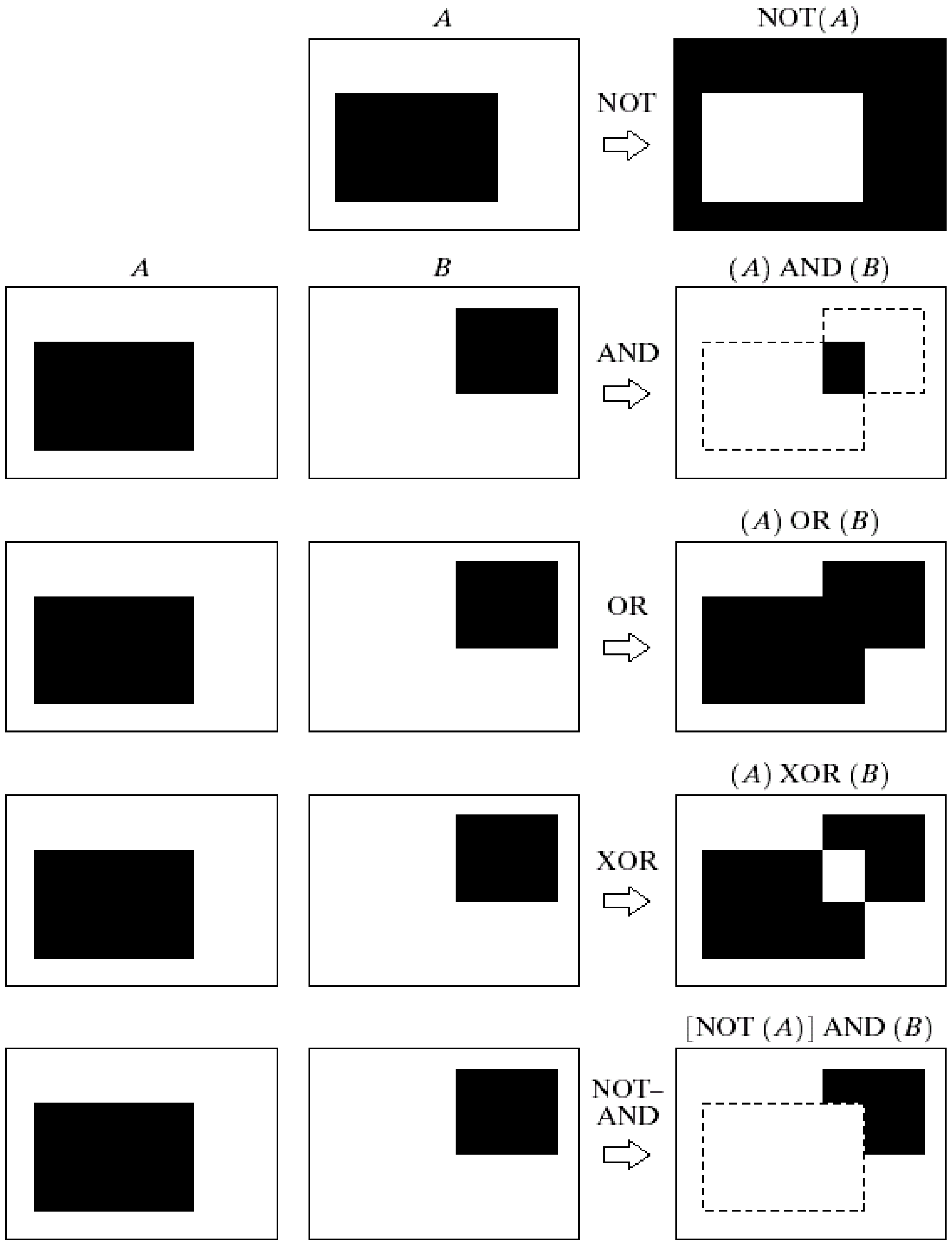


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Preliminaries (6)

- **Structuring elements (SE)**
  - Small sets or sub-images used to probe an image under study for properties of interest.
  - Consists of a pattern specified as the coordinates of a number of discrete points relative to some origin.
  - A convenient way of representing the element is as a small image on a rectangular grid.
  - The origin does not have to be in the center of the structuring element, but often it is.
  - Structuring elements that fit into a 3×3 grid with its origin at the center are the most commonly seen type.

Examples: Structuring Elements (1)

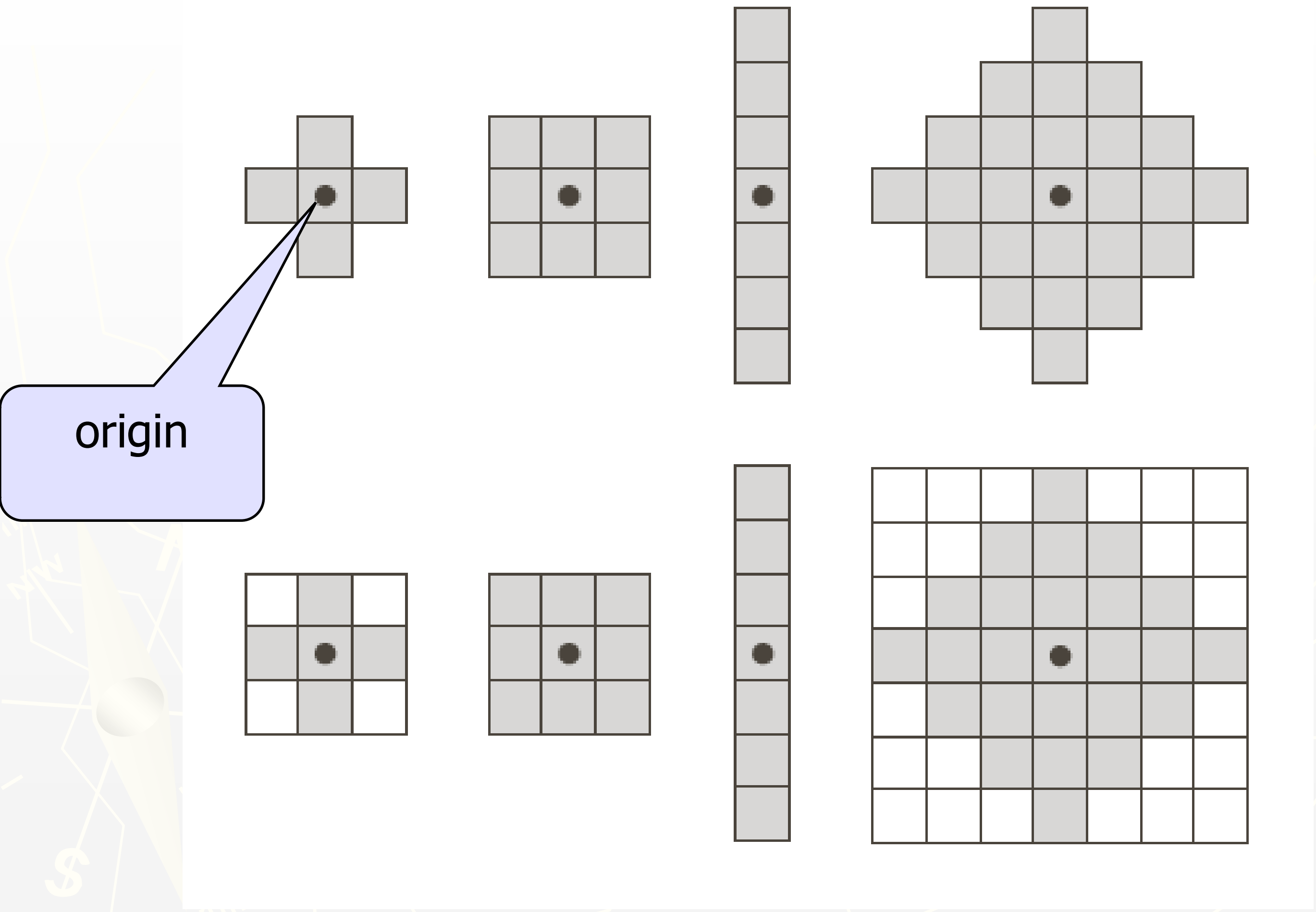
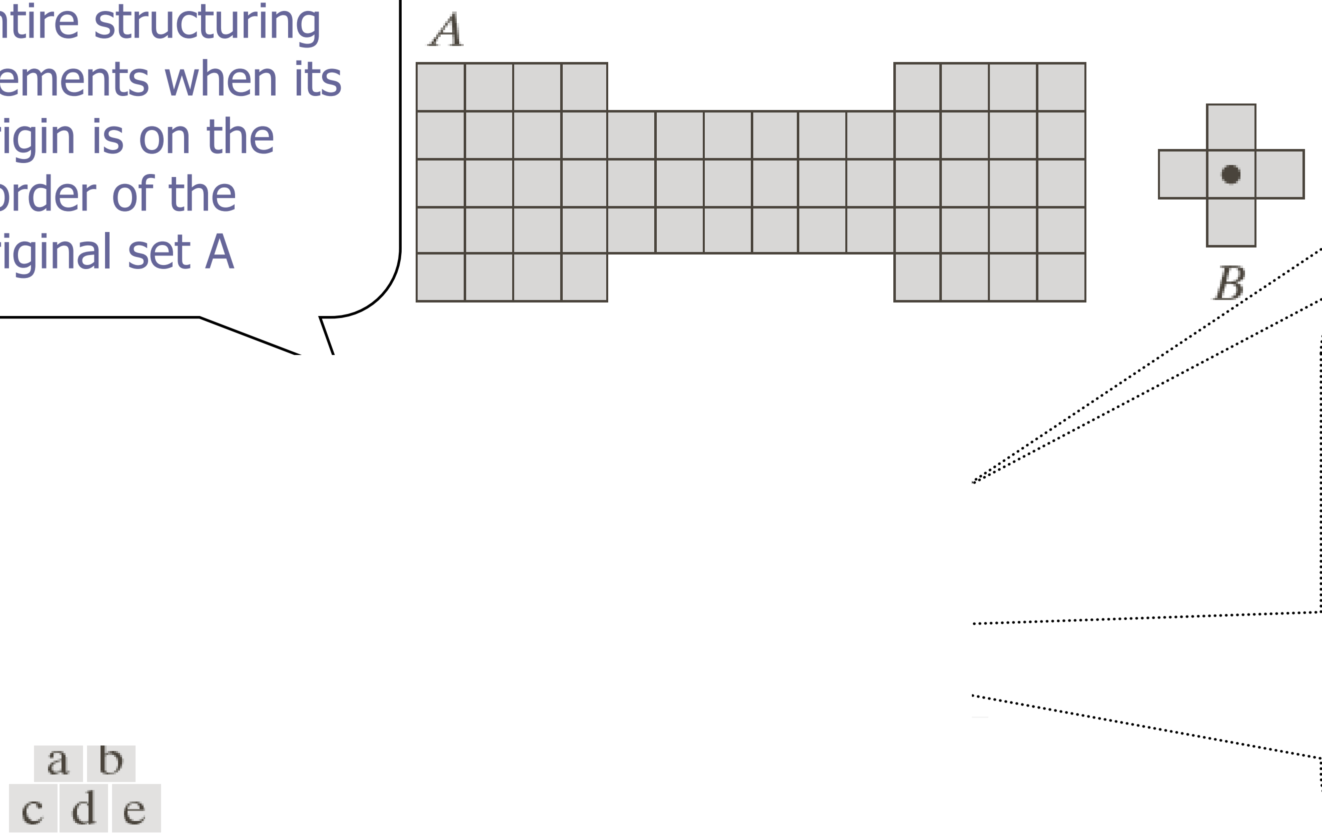


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Examples: Structuring Elements (2)

Accommodate the entire structuring elements when its origin is on the border of the original set A

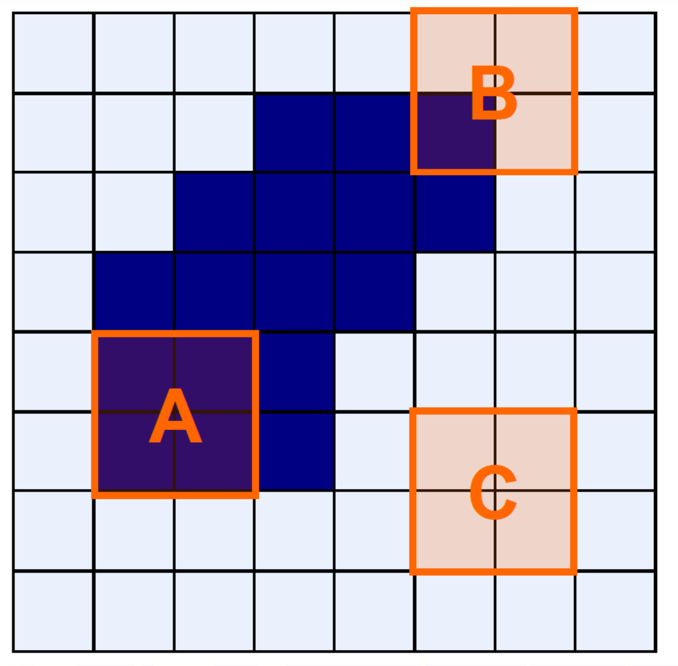


Origin of B visits every element of A

At each location of the origin of B, if B is completely contained in A, then the location is a member of the new set, otherwise it is not a member of the new set.

FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

Structuring Elements (Hits and Fits)



Fit: All one pixels in the structuring element cover one pixels in the image

Hit: Any one pixel in the structuring element covers any one pixel in the image

Basic Morphological Operations

- Erosion

- Dilation

- Combine to: Keep general shape but smooth w.r.t.
  - Opening $\longrightarrow$ Object
  - Closing $\longrightarrow$ Background

Erosion

Does the structuring element fit the set?

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

$$A \ominus B = \left\{ z \mid (B)_Z \subseteq A \right\}$$

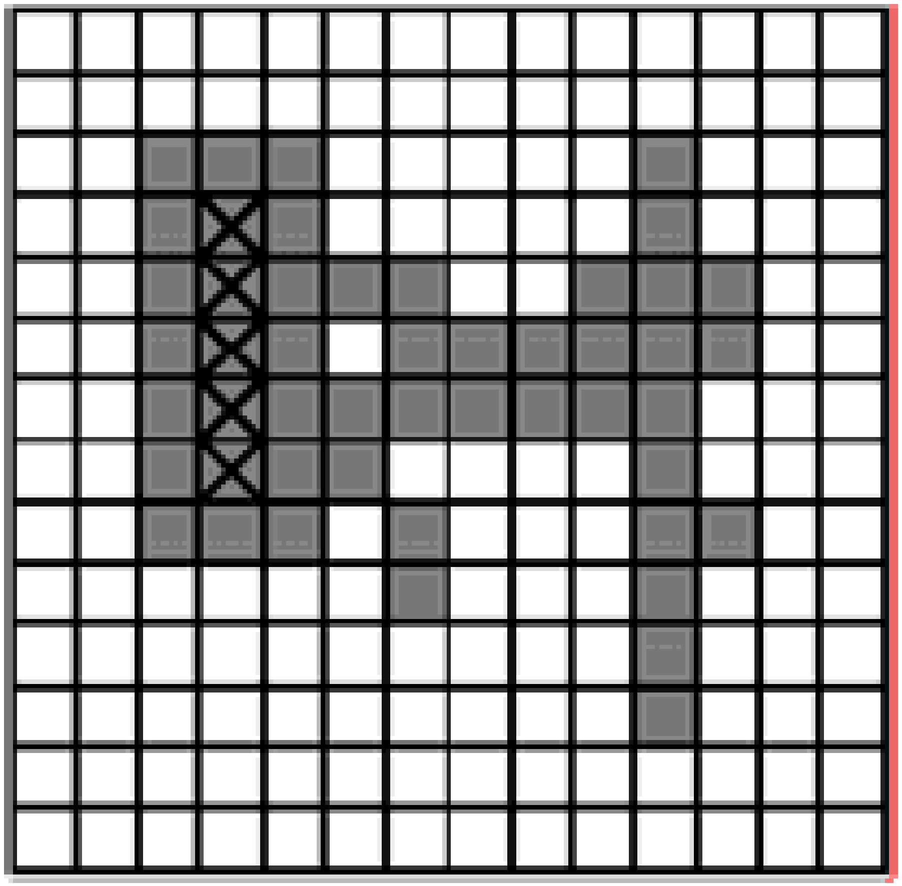
The set of all points z such that B, translated by z, is contained by A.

$$A \ominus B = \left\{ z \mid (B)_Z \cap A^c = \varnothing \right\}$$

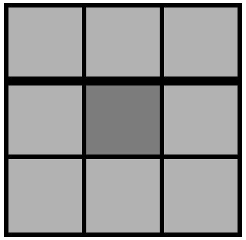
Erosion: How it works

- For each foreground pixel, superimpose the structuring element on the input image such that the origin of the structuring element coincides with the input pixel coordinates.
- If for *every* pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is.
- If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.

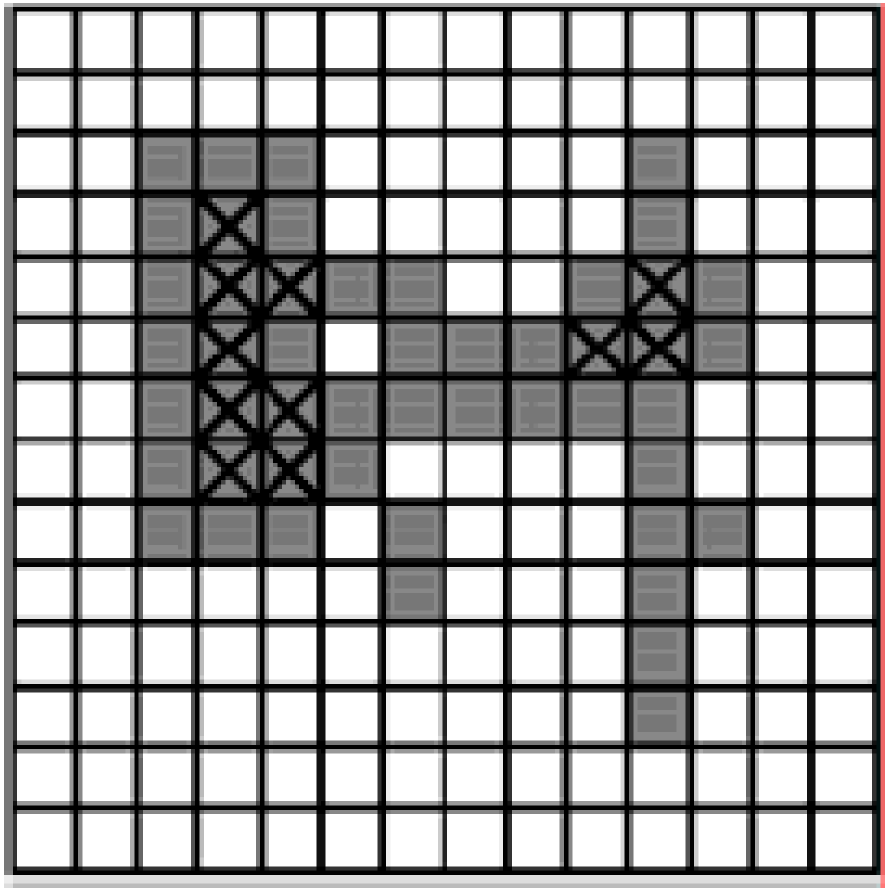
Erosion: How it works



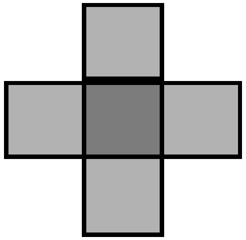
SE=



Erosion: How it works



SE=



Example of Erosion (1)

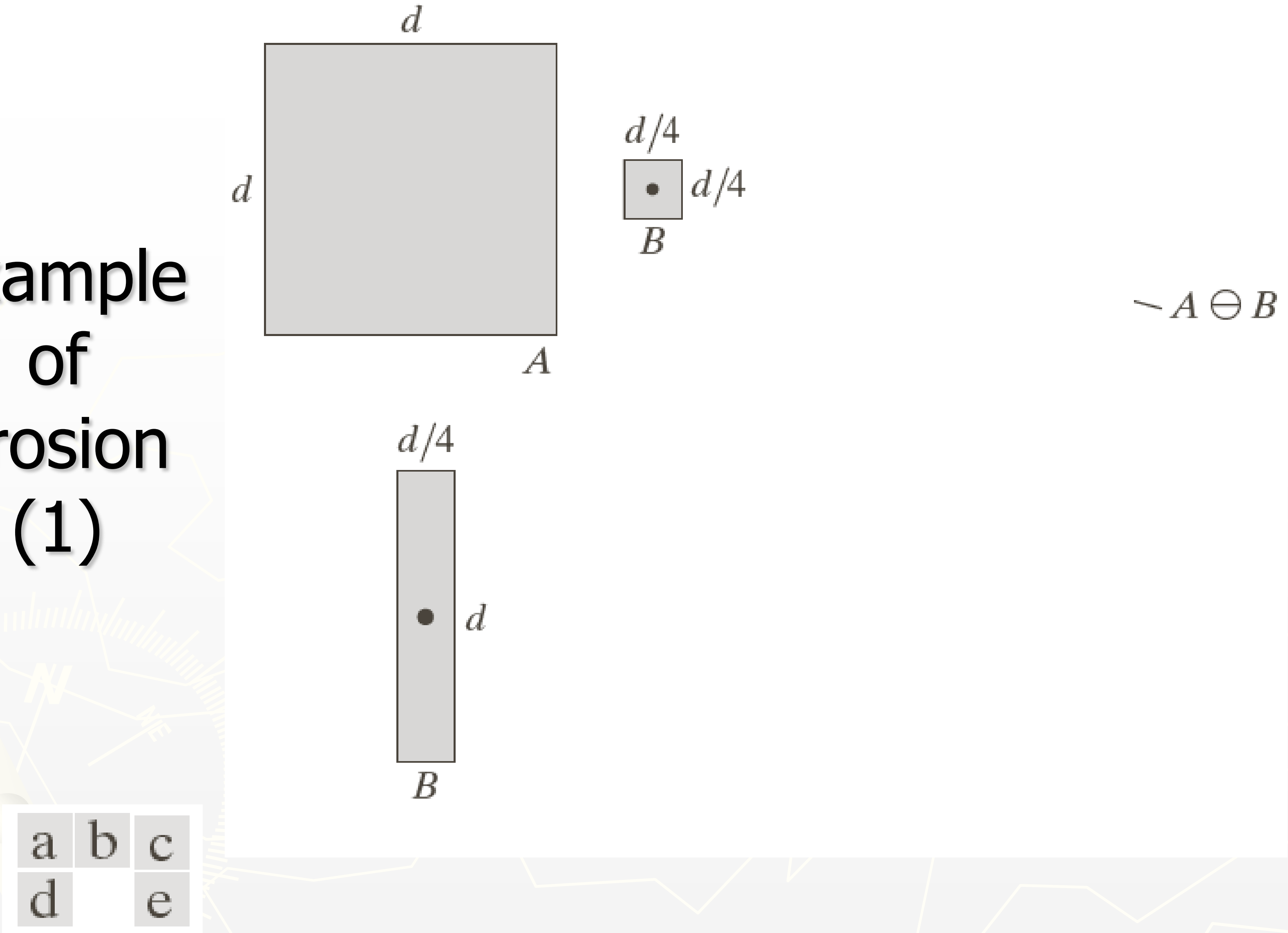
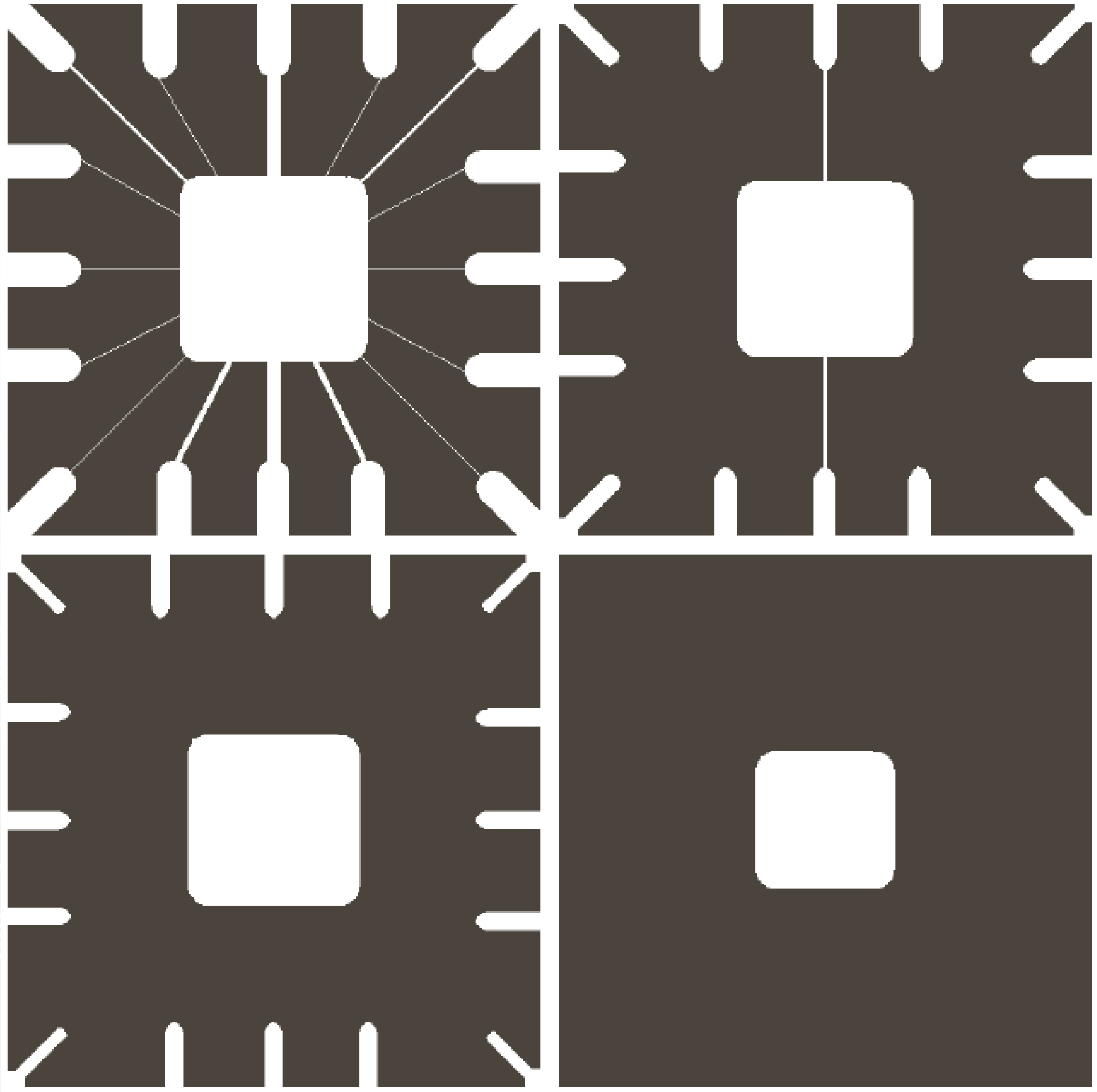


FIGURE 9.4 (a) Set A. (b) Square structuring element, B. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.

Example of Erosion (2)

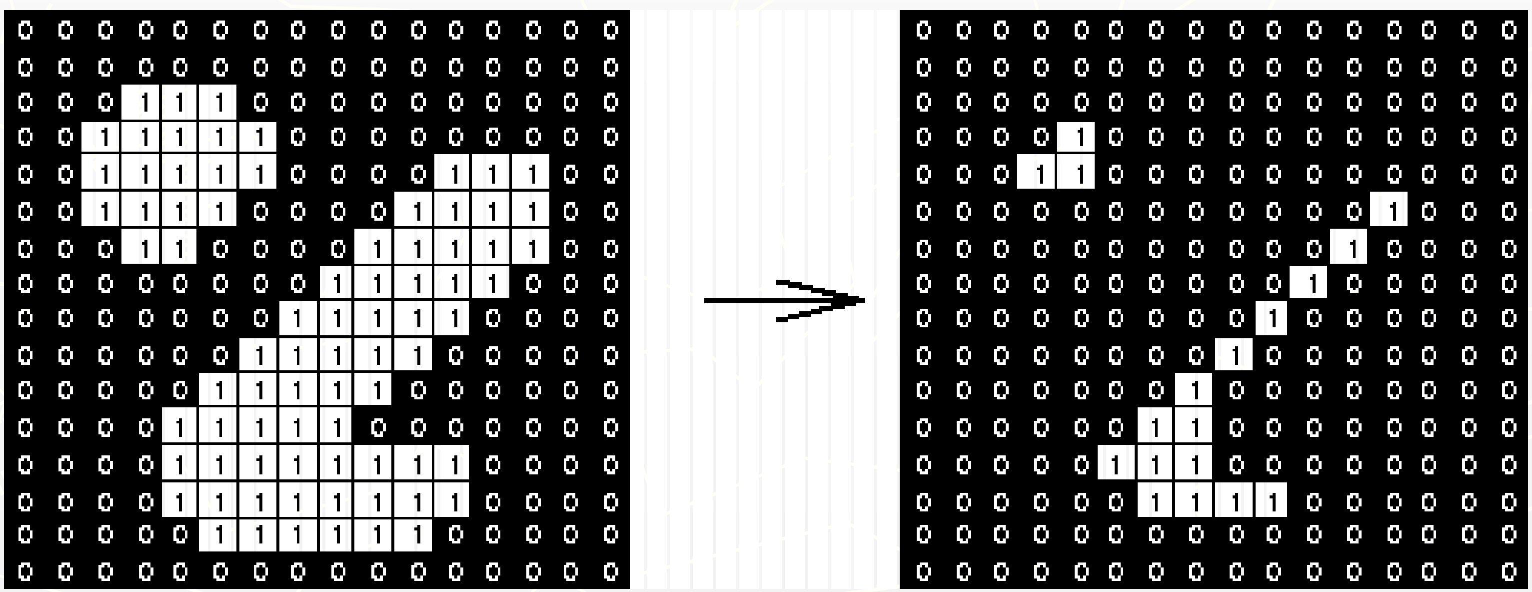


a b
c d

FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11×11, 15×15, and 45×45, respectively. The elements of the SEs were all 1s.

Erosion

- The basic effect of the operator on a binary image is to **erode away the boundaries of regions of foreground pixels** (*i.e.* white pixels, typically).
- Thus, areas of foreground pixels shrink in size, and holes within those areas become larger.

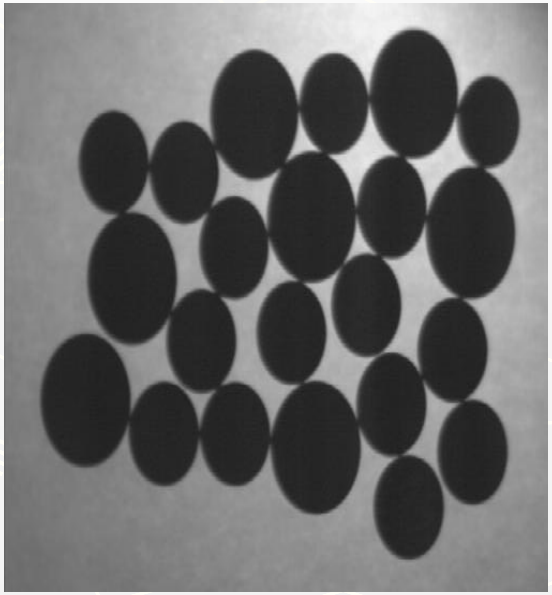


Erosion

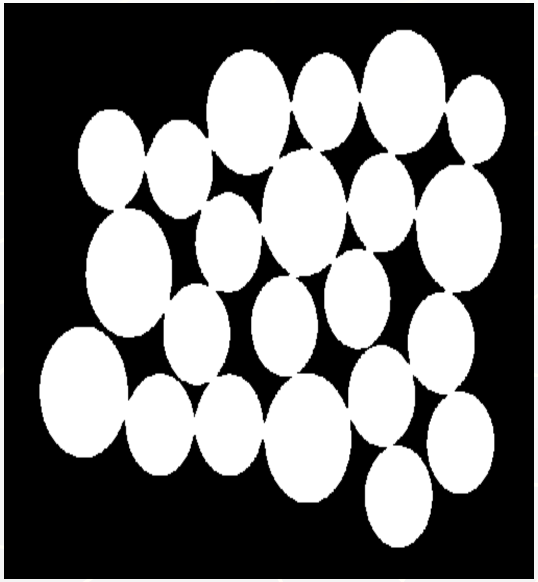
- Erosions can be made directional by using less symmetrical structuring elements.
- For example, a structuring element that is 10 pixels wide and 1 pixel high will erode in a horizontal direction only.
- Similarly, a 3×3 square structuring element with the origin in the middle of the top row rather than the center, will erode the bottom of a region more severely than the top.

Erosion

- There are many specialist uses for erosion.
- One of the more common is to separate touching objects in a binary image so that they can be counted using a **labeling algorithm**.



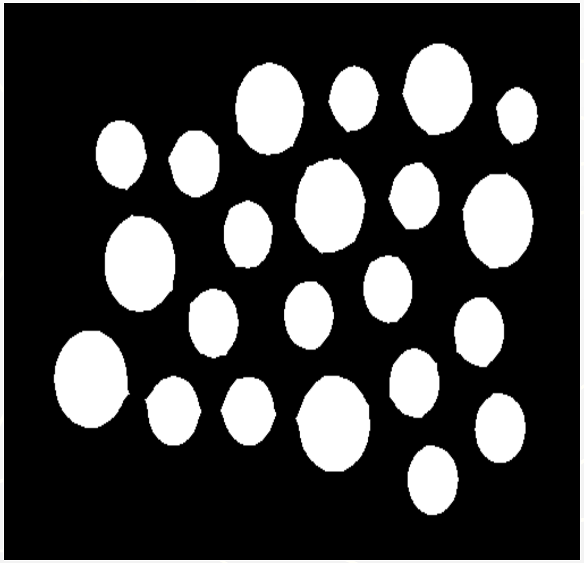
A no. of dark disks



Thresholding at 90

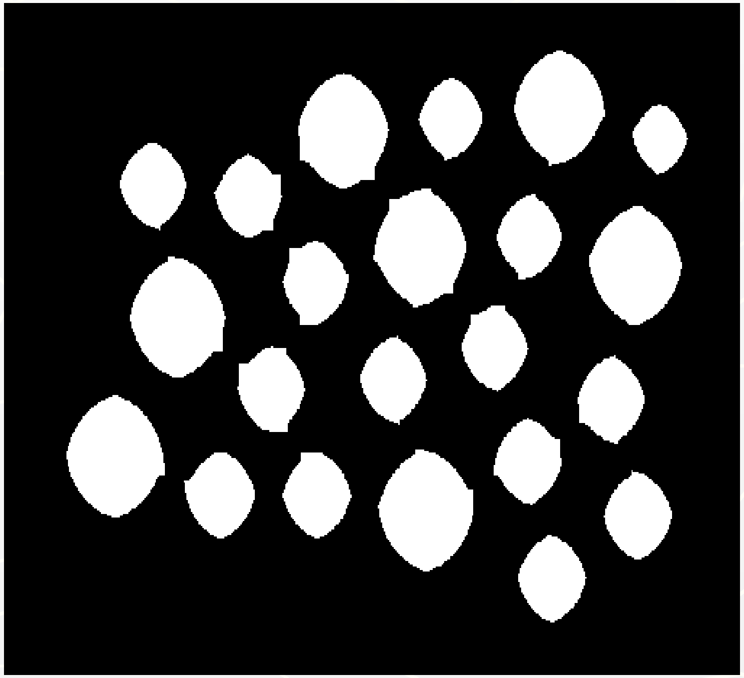
Erosion

- Result of eroding twice using a disk shaped structuring element 11 pixels in diameter.
- Now the coins can be easily counted.
- The relative sizes of the coins can be used to distinguish the various types.



Erosion

- 9×9 square structuring element is used instead of a disk .
- The square structuring element has led to distortion of the shapes, which in some situations could cause problems in identifying the regions after erosion.



Erosion

- Erosion can also be used to remove small spurious bright spots **('salt noise')** in images.
- We can also use erosion for **edge detection** by taking the erosion of an image and then **subtracting** it away from the original image, thus highlighting just those pixels at the edges of objects that were removed by the erosion.
- Erosion is also used as the basis for many other mathematical morphology operators.

Dilation

Does the structuring element hit the set?

With A and B as sets in Z^2, the dilation of A by B,

denoted $A \oplus B$, is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \phi\}$$

The set of all displacements z, the translated $\hat{B}$ and A

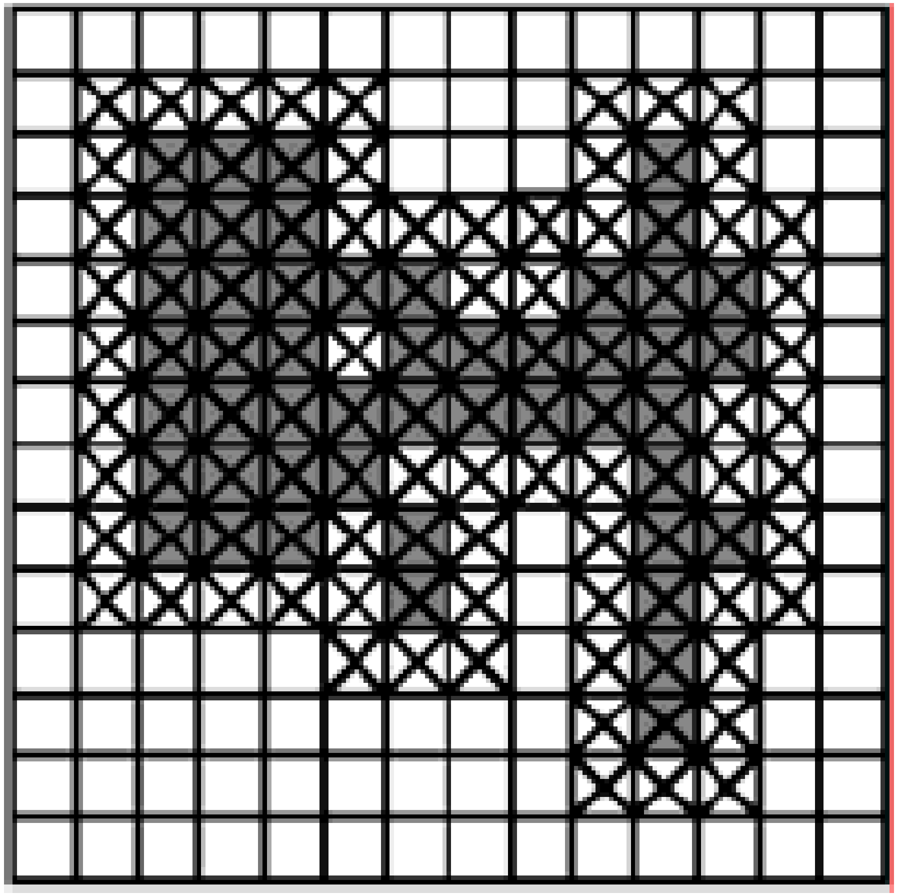
overlap by atleast one element

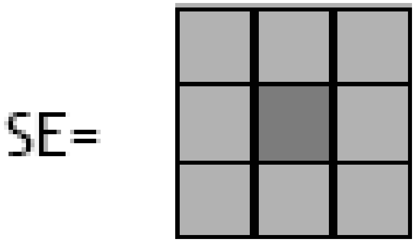
$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\}$$

Dilation: How it works

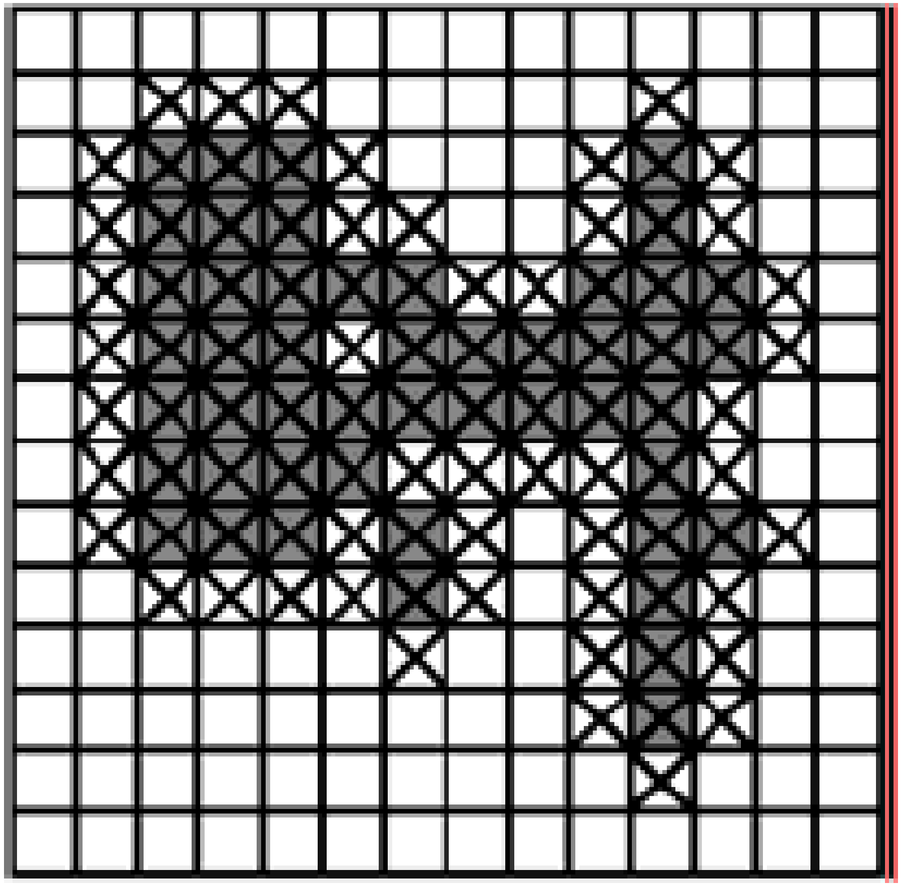
- For each background pixel (which we will call the *input pixel*) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel position.
- If *at least one* pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value.
- If all the corresponding pixels in the image are background, however, the input pixel is left at the background value.

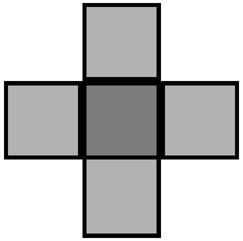
Dilation: How it works





Dilation: How it works





Examples of Dilation (1)

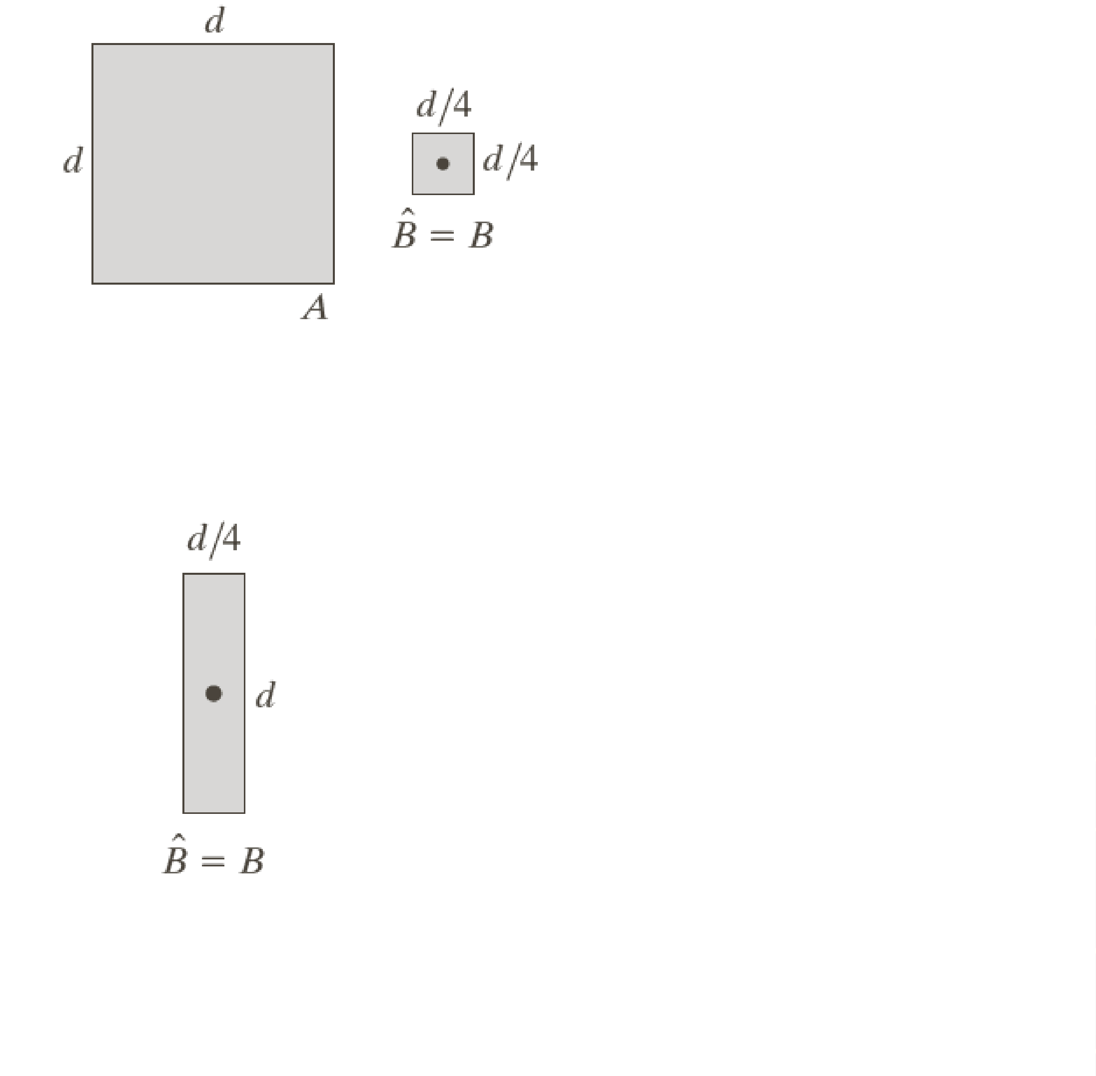
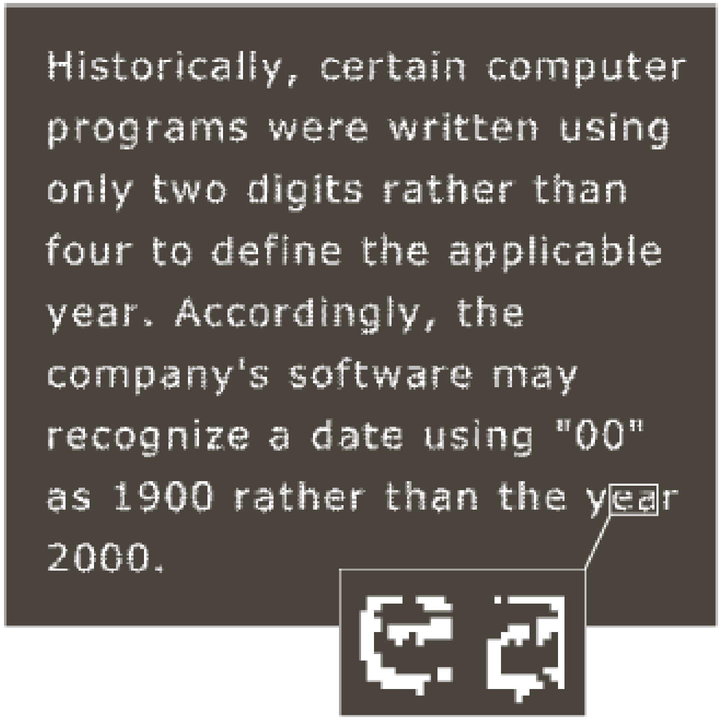


FIGURE 9.6
(a) Set A.
(b) Square structuring element (the dot denotes the origin).
(c) Dilation of A by B, shown shaded.
(d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference

Examples of Dilation (2)



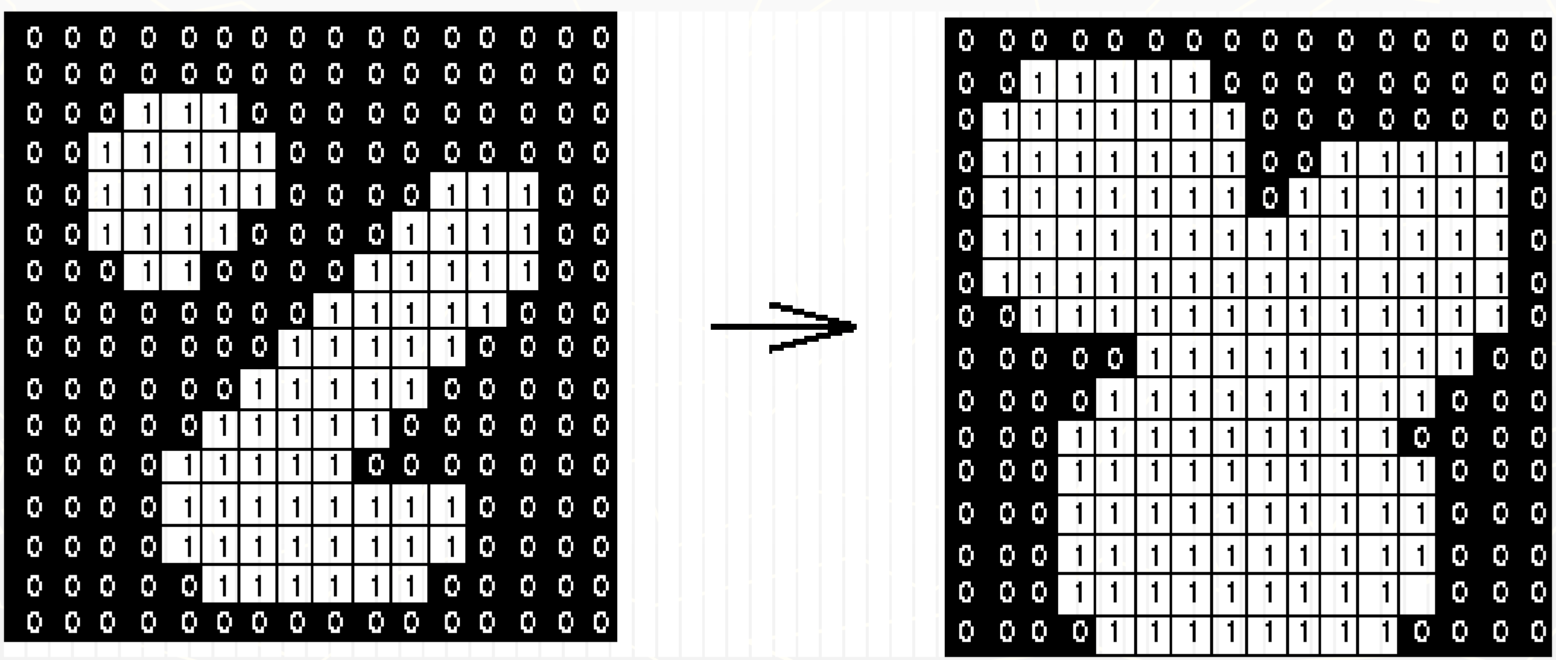
0	1	0
1	1	1
0	1	0

a c
b

FIGURE 9.7
(a) Sample text of poor resolution with broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

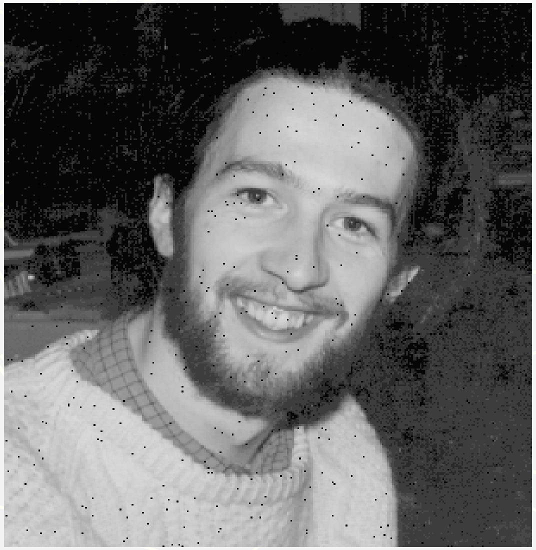
Dilation

- The basic effect of the operator on a binary image is to **gradually enlarge the boundaries of regions of foreground pixels** (*i.e.* white pixels, typically). Thus areas of foreground pixels grow in size while holes within those regions become smaller.



Dilation

- There are many specialist uses for dilation. For instance it can be used to fill in small spurious holes (`**pepper noise'**) in images.
- Result of dilating image with a 3×3 square structuring element. Although the noise has been effectively removed, the image has been degraded significantly.



Duality

- Erosion and dilation are duals of each other with respect to set complementation and reflection

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

and

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

Duality

- Erosion and dilation are duals of each other with respect to set complementation and reflection

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

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

$$= \{z \mid (B)_z \cap A^c \neq \phi\}$$

$$= A^c \oplus \hat{B}$$

Duality

- Erosion and dilation are duals of each other with respect to set complementation and reflection

$$(A \oplus B)^c = \{z \mid (\hat{B})_z \cap A \neq \phi\}^c$$

$$= \{z \mid (\hat{B})_z \cap A^c = \phi\}$$

$$= A^c \ominus \hat{B}$$

Opening and Closing

- Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions.

- Closing tends to smooth sections of contours but it generates fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

Opening and Closing

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

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

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

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

Opening

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

$$A \circ B = \bigcup \left\{ \left(B \right)_Z \mid \left(B \right)_Z \subseteq A \right\}$$

Opening

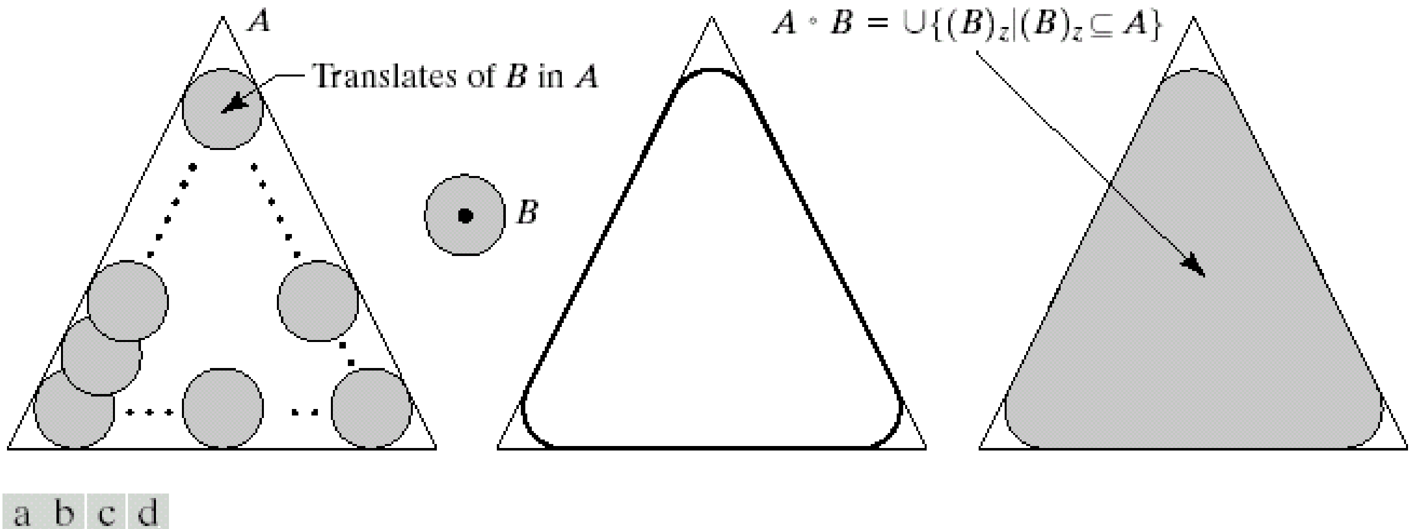


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

Opening

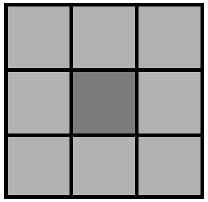
- The basic effect of an **opening** is somewhat like **erosion.**
  - It tends to remove some of the foreground (bright) pixels from the edges of regions of foreground pixels. However it is less destructive than erosion in general.
- The exact operation is determined by a structuring element.
- The effect of the operator is to preserve *foreground* regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels.

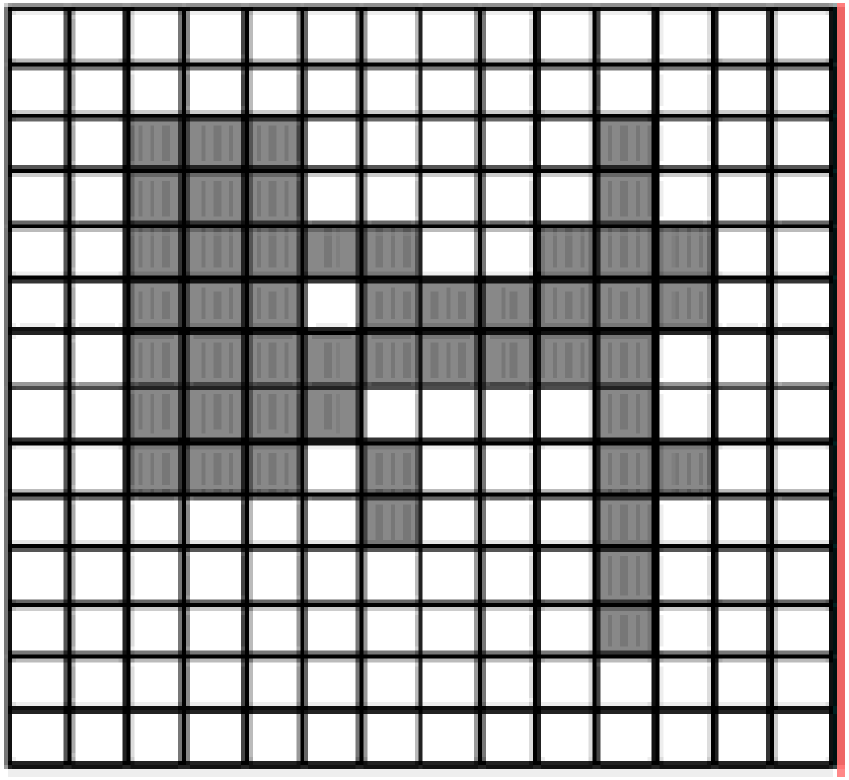
Opening

- Erosion has the big disadvantage that it affects *all* regions of foreground pixels indiscriminately. Opening gets around this by performing both an erosion and a dilation on the image.
- Imagine taking the structuring element and sliding it around *inside* each foreground region, without changing its orientation.
- All pixels which can be covered by the structuring element with the structuring element being entirely within the foreground region will be preserved.
- All foreground pixels which cannot be reached by the structuring element without parts of it moving out of the foreground region will be eroded away.
- The new boundaries of foreground regions will all be such that the structuring element fits inside them, and so further openings with the same element have no effect. The property is known as *idempotence*.

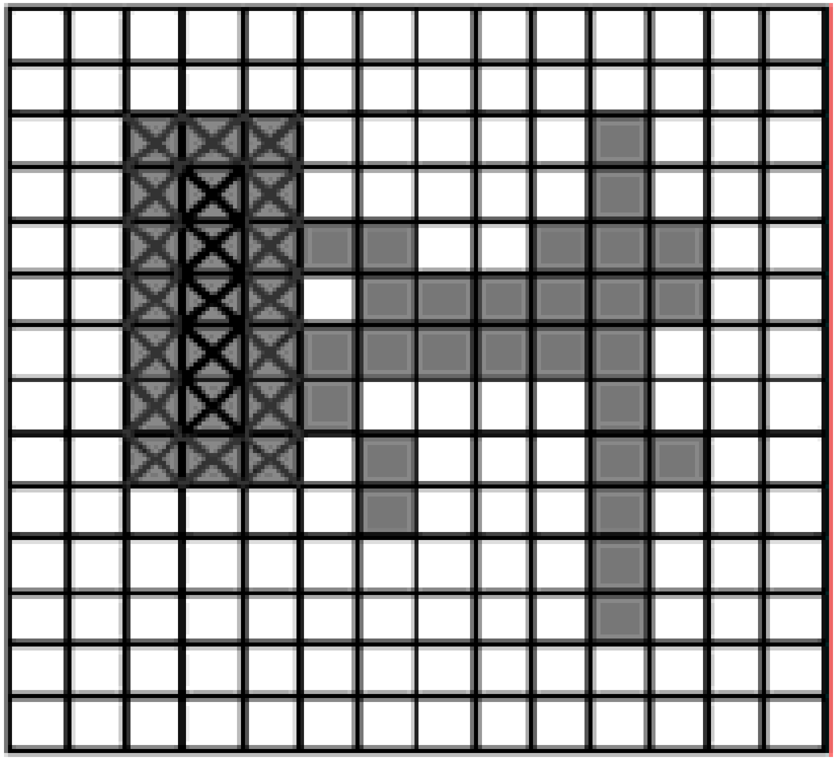
Opening

B=





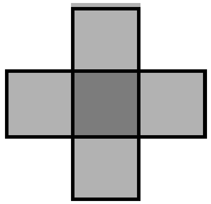
A

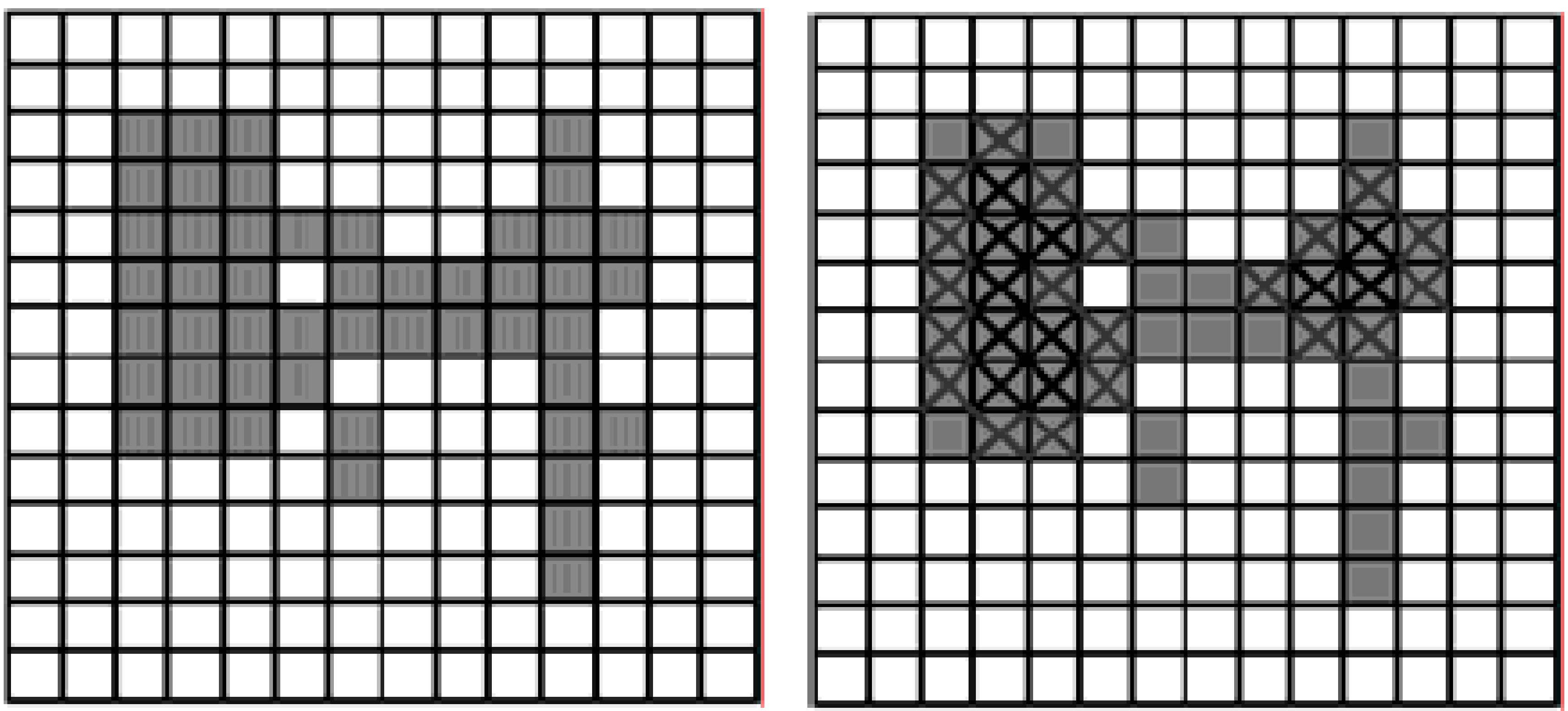


A⊖B   A∘B

Opening

B=



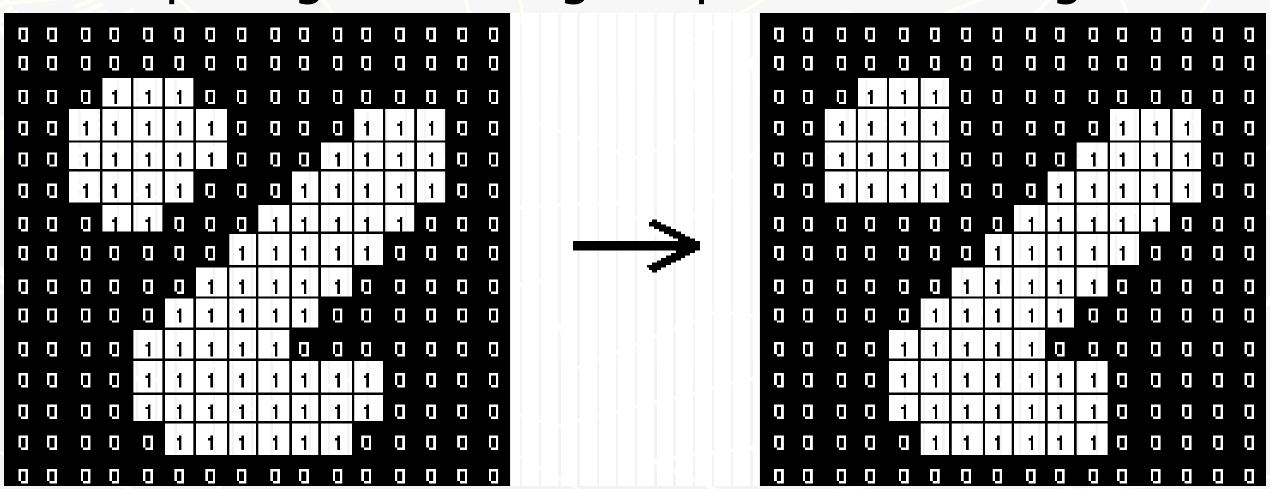


A

$A \ominus B$ $A \circ B$

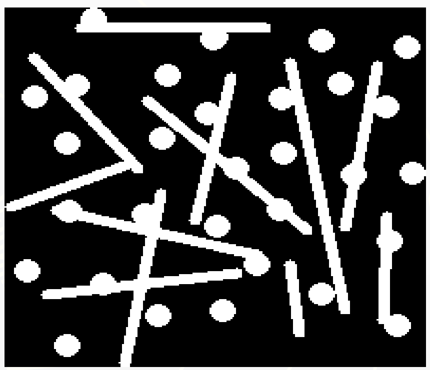
Opening

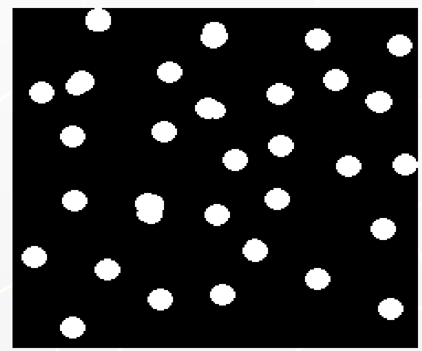
- As with erosion and dilation, it is very common to use the 3×3 structuring element.
- The effect in the figure is rather subtle since the structuring element is quite compact and so it fits into the foreground boundaries quite well even before the opening operation.
- To increase the effect, multiple erosions are often performed with this element followed by the same number of dilations. This effectively performs an opening with a larger square structuring element.



Opening

- A binary image containing a mixture of circles and lines. Suppose that we want to separate out the circles from the lines, so that they can be counted.
- Opening with a disk shaped structuring element 11 pixels in diameter:

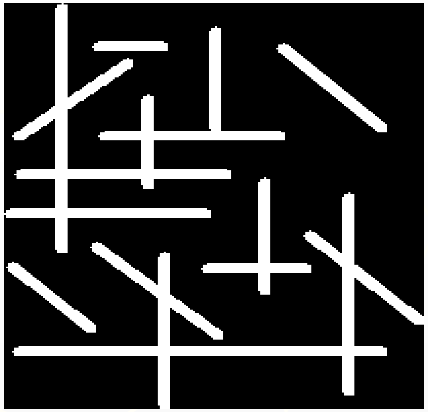




- Some of the circles are slightly distorted, but in general, the lines have been almost completely removed while the circles remain almost completely unaffected.

Opening

- We wish to separately extract the horizontal and vertical lines.



Opening with a 3×9 vertically oriented structuring element

Opening with a 9×3 horizontally oriented structuring element

Closing

- Closing is similar in some ways to **dilation:**
  - it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions).
  - but it is less destructive of the original boundary shape.
- The effect of the operator is to preserve *background* regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

Closing

- The effect of closing can be quite easily visualized.
  - Imagine taking the structuring element and sliding it around *outside* each foreground region, without changing its orientation.
  - For any background boundary point, if the structuring element can be made to touch that point, without any part of the element being inside a foreground region, then that point remains background.
  - If this is not possible, then the pixel is set to foreground.
  - After the closing has been carried out the background region will be such that the structuring element can be made to cover any point in the background without any part of it also covering a foreground point, and so further closings will have no effect. This property is known as *idempotence*

Closing

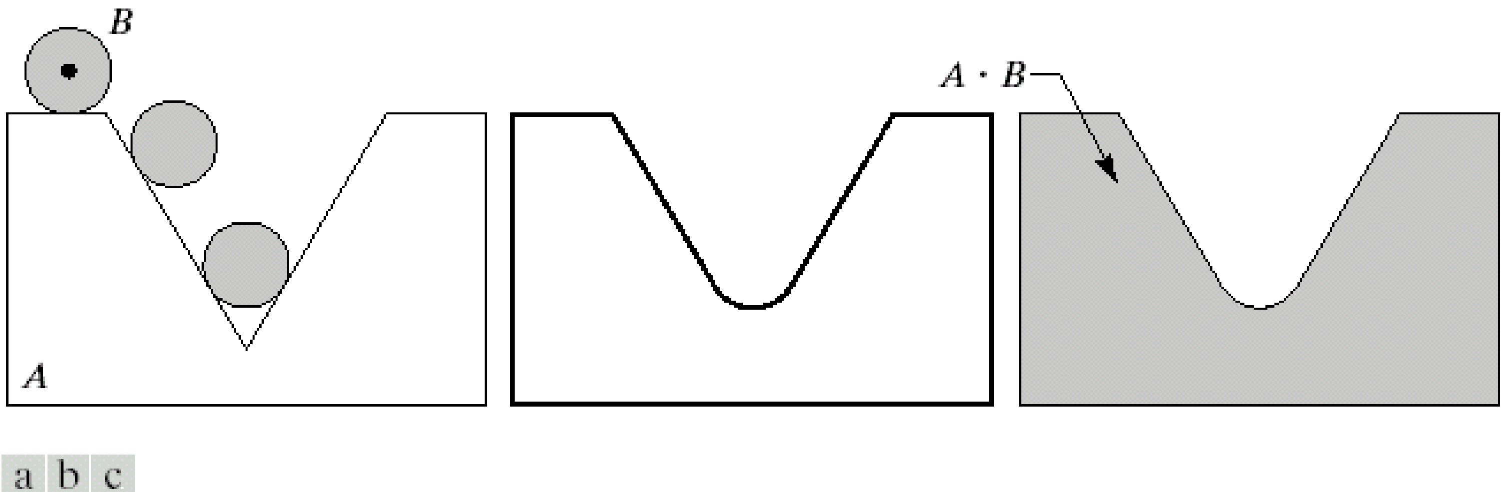
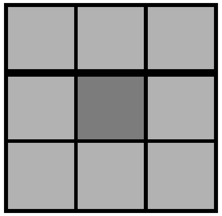
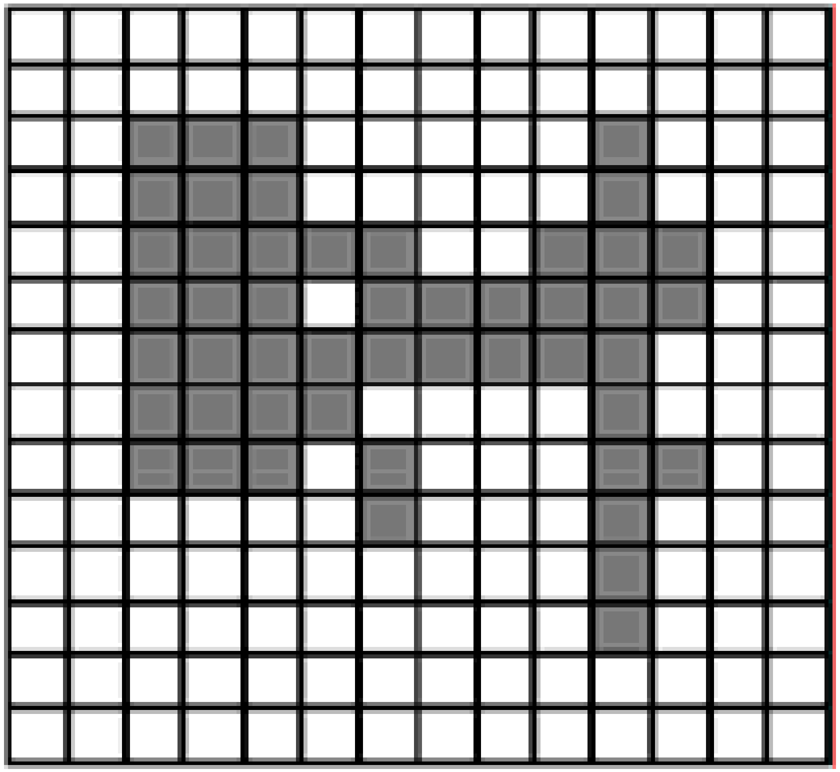


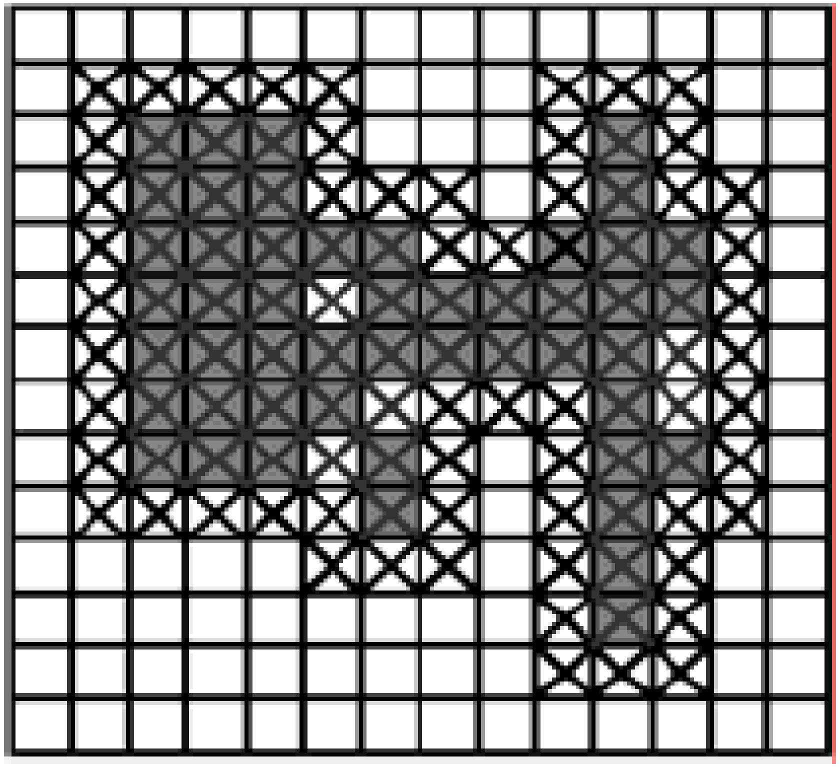
FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

Closing

B=




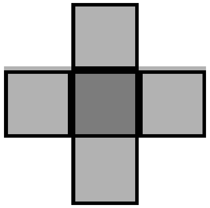
A

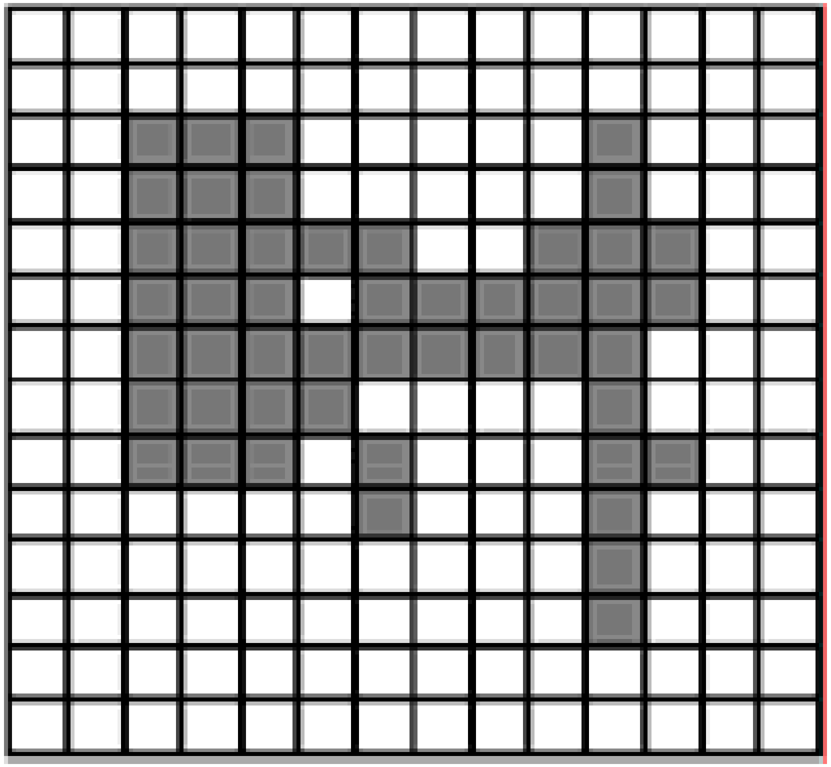


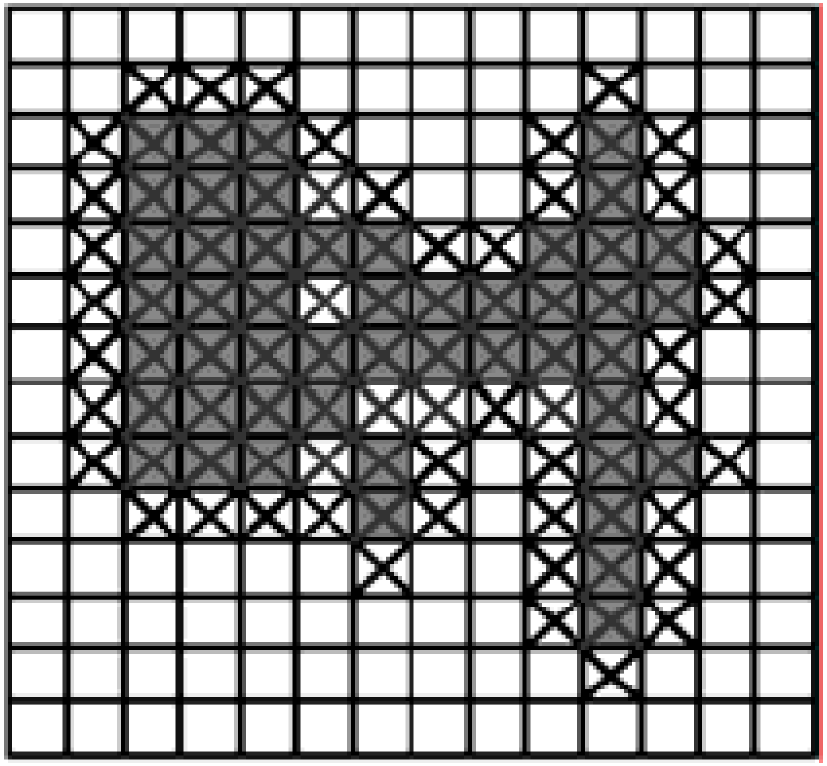
$A \oplus B$ $A \bullet B$

Closing

B=





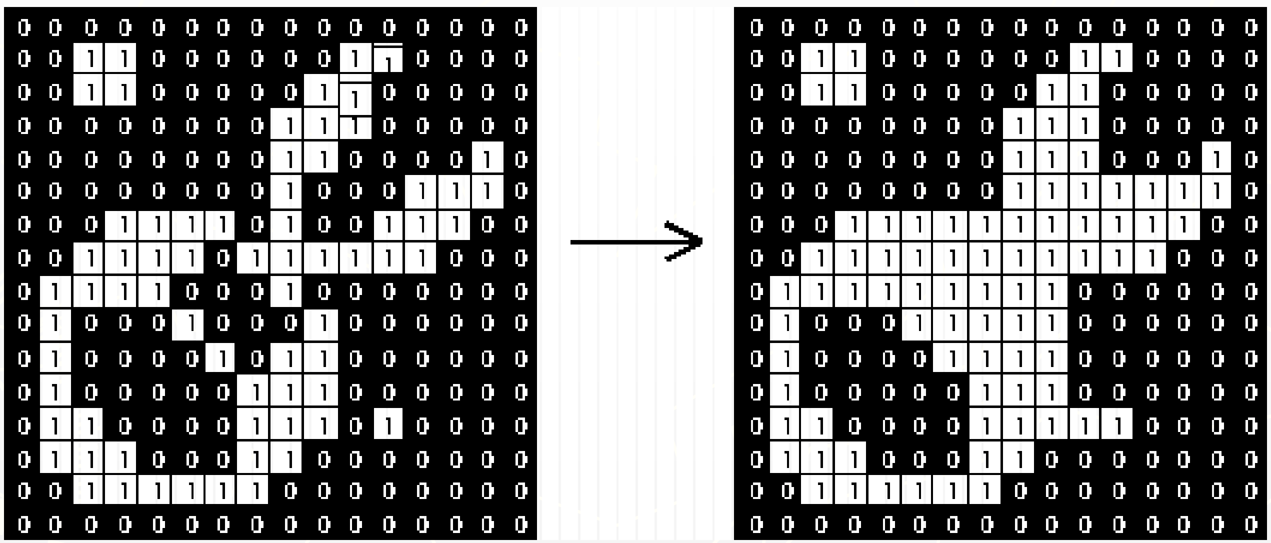


A

$A \oplus B$ $\quad A \bullet B$

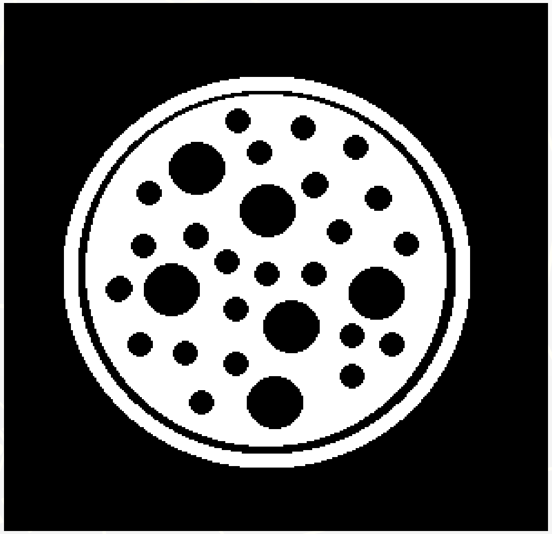
Closing

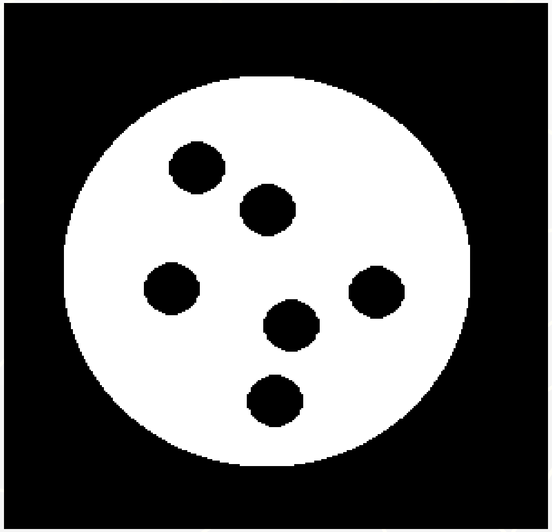
- Effect of closing using a 3x3 square structuring element.



Closing

- An image containing large holes and small holes.
  - If it is desired to remove the small holes while retaining the large holes, then we can simply perform **a closing with a disk-shaped structuring element with a diameter larger than the smaller holes, but smaller than the large holes.**





Closing with 22 pixel dia. disk

The Properties of Opening and Closing

- Properties of Opening

  (a) $A \circ B$ is a subset (subimage) of A

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

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

- Properties of Closing

  (a) A is a subset (subimage) of $A \bullet B$

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

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

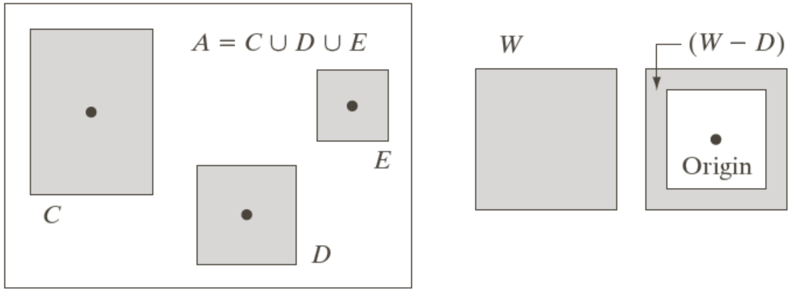
The Hit-or-Miss Transformation

- Used to look for particular patterns of foreground and background pixels.
- Very simple object recognition.
- All other morphological operations can be derived from it.
- Input:
  - Binary Image
  - Structuring Element, containing 0s and 1s.

The Hit-or-Miss Transformation

- Similar to Pattern Matching:
- If foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color.

The Hit-or-Miss Transformation



a b
c d
e
f

FIGURE 9.12 (a) Set A. (b) A window, W, and the local background of D with respect to W, $(W - D)$. (c) Complement of A. (d) Erosion of A by D. (e) Erosion of A^c by $(W - D)$. (f) Intersection of (d) and (e), showing the location of the origin of D, as desired. The dots indicate the origins of C, D, and E.

if B denotes the set composed of D and its background, the match (or set of matches) of B in A, denoted $A \circledast B$,

$$A \circledast B = (A \ominus D) \cap \left[A^c \ominus (W - D) \right]$$

$$B = (B_1, B_2)$$

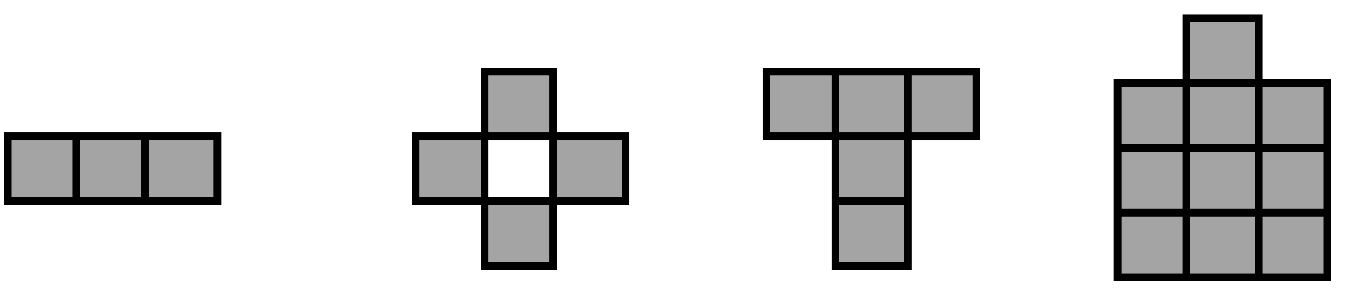
B_1 : object

B_2 : background

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

The Hit-or-Miss Transformation

- Find location of one shape among a set of shapes: ”template matching”



- Composite SE: object part (B1) and background part (B2)
- Does B1 ***fits the object while, simultaneously,*** B2 misses the object, i.e., ***fits the background?***

The Hit-or-Miss Transformation

- Corner detection:
  - Structuring Elements representing four corners

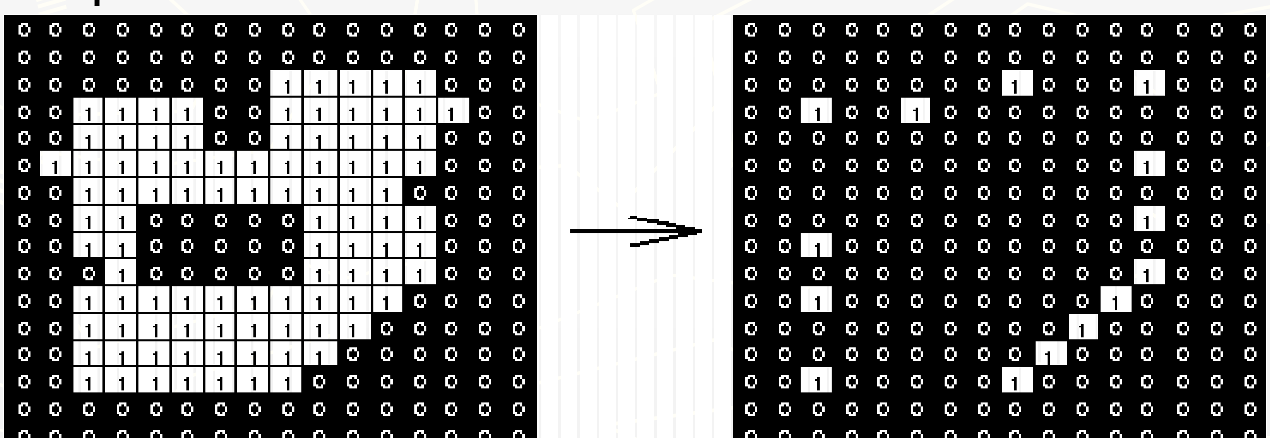
	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

0	0	
0	1	1
	1	

  - Apply each Structuring Element
  - Use OR operation to combine the four results



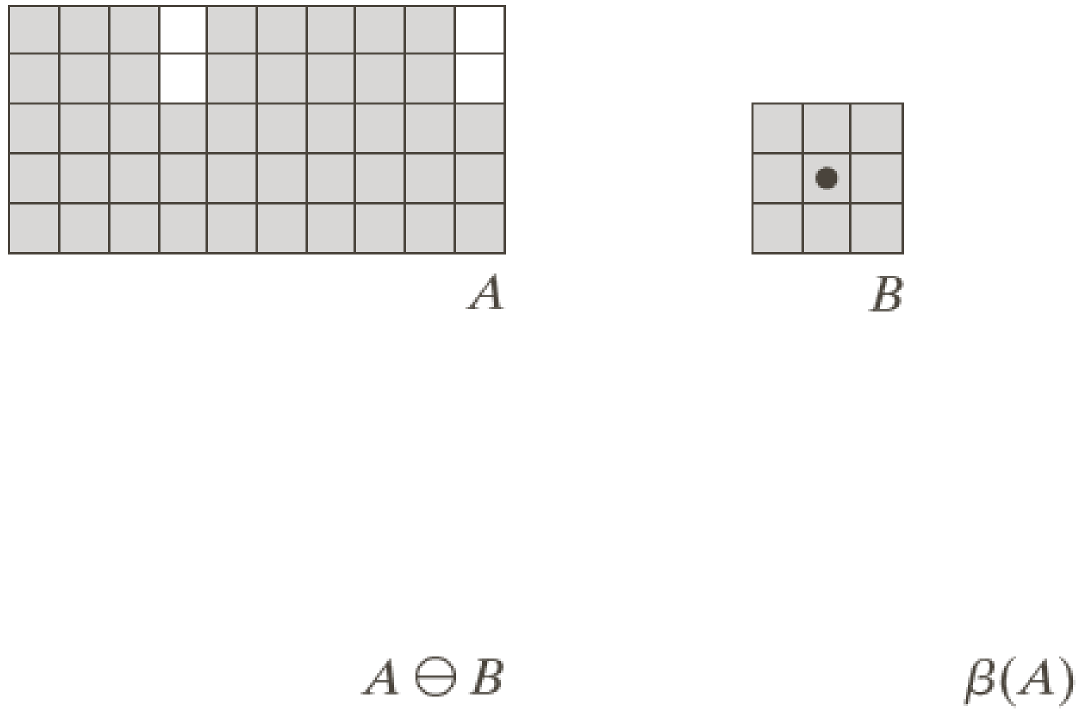
Some Basic Morphological Algorithms

- **Boundary Extraction**

The boundary of a set *A*, can be obtained by first eroding *A* by *B* and then performing the set difference between *A* and its erosion.

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

Boundary Extraction - Example



a	b
c	d

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

Boundary Extraction - Example

a b

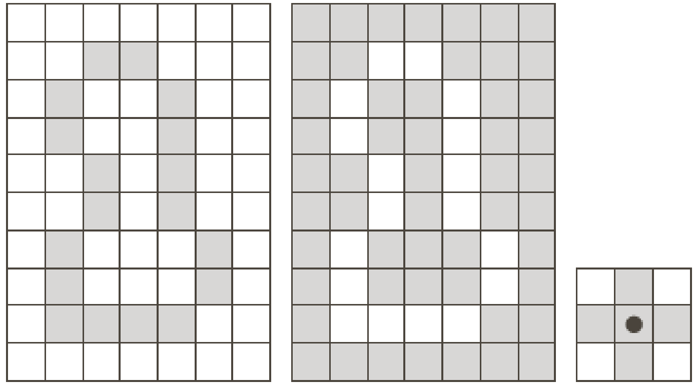
FIGURE 9.14 (a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

Some Basic Morphological Algorithms

- **Region Filling**
- Region filling applies **logical NOT**, **logical AND** and dilation iteratively. The process can be described by the following formula:

$$X_k = \mathrm{dilate}(X_{k-1}, \mathrm{J}) \cap \mathrm{A}_{\mathrm{not}}$$

where X_k is the region which after convergence fills the boundary, J is the structuring element and A_{not} is the negative of the boundary.



A A^c B

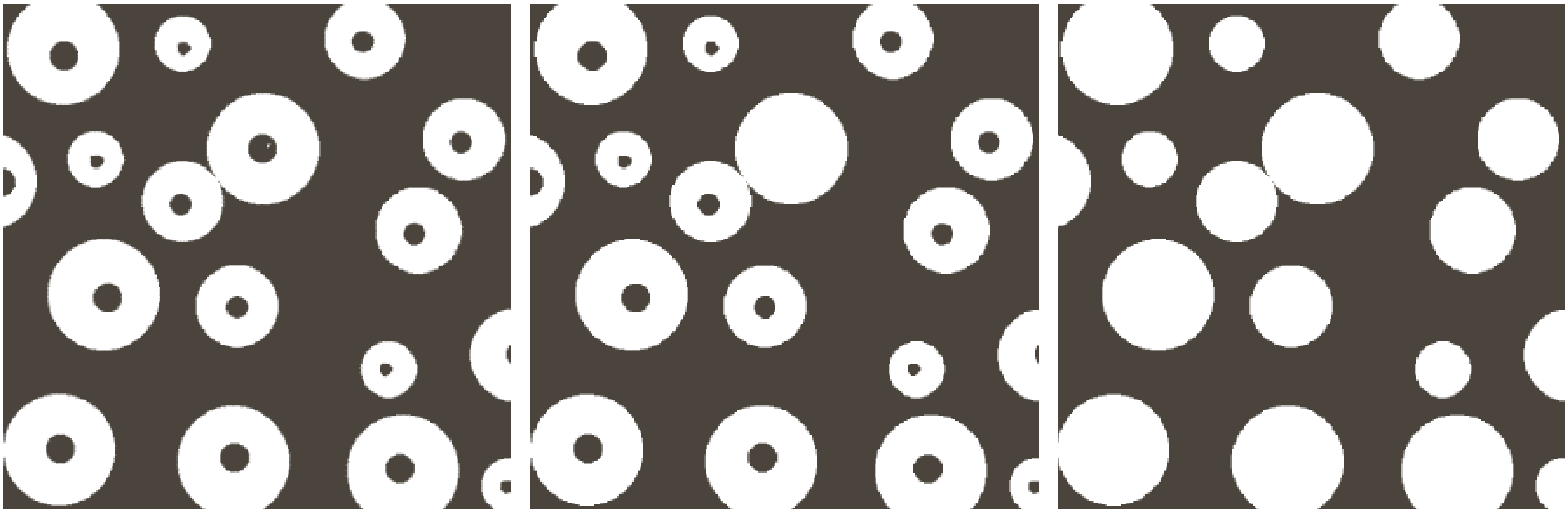
X_0 X_1 X_2

X_6 X_8 $X_8 \cup A$

a	b	c
d	e	f
g	h	i

FIGURE 9.15 Hole filling. (a) Set A (shown shaded). (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

Region Filling - Example



a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

Some Basic Morphological Algorithms

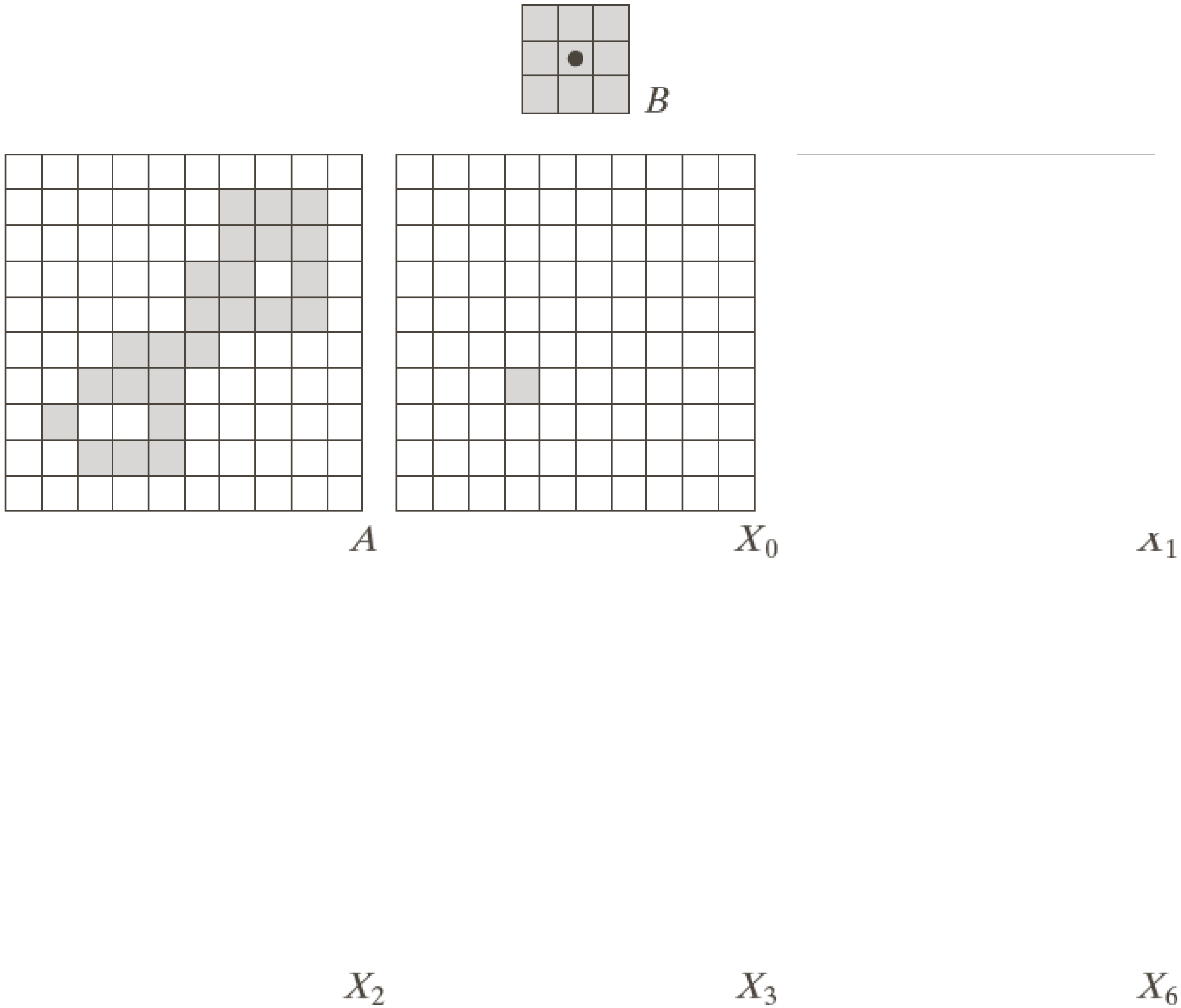
- **Extraction of Connected Components**
- Let *A* be a set containing one or more connected components.
- Let *Y* represent a connected component in set *A*.
- Let point *p* of *Y* be known. Following yields all elements of *Y*:

$$X_k = (X_{k-1} \oplus B) \cap A$$

$$B: \text{structuring element}$$

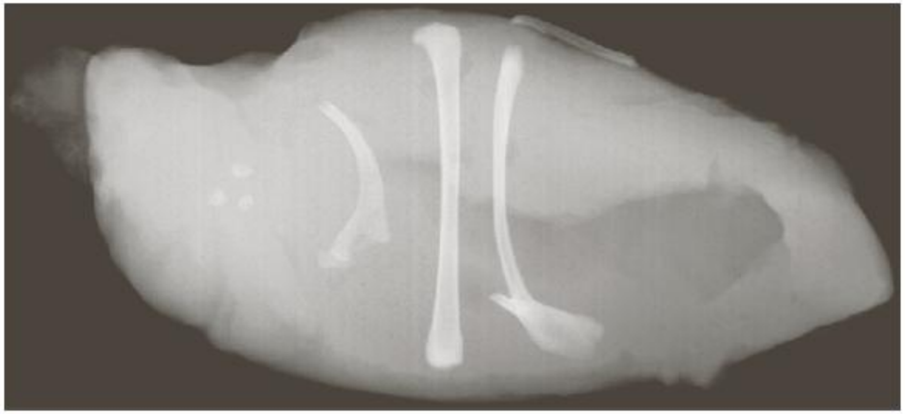
$$\text{until } X_k = X_{k\text{-}1}$$

- where, $X_0 = p$ and *B* is a suitable structuring element.



	a	
b	c	d
e	f	g

FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array 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. (9.5-3).



a
b
c d

FIGURE 9.18
(a) X-ray image of chicken filet with bone fragments.
(b) Thresholded image. (c) Image eroded with a 5×5 structuring element of 1s.
(d) Number of pixels in the connected components of (c).
(Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)

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

Some Basic Morphological Algorithms

- **Convex Hull**

  A set *A* is said to be ***convex*** if the straight line segment joining any two points in *A* lies entirely within *A*.

- The ***convex hull*** *H* or of an arbitrary set *S* is the smallest convex set containing *S*.

- The set difference *H* - *S* is called the convex deficiency of *S*.

Convex Hull

Let B^i, $i = 1,\ 2,\ 3,\ 4$, represent the four structuring elements.

The procedure consists of implementing the equation:

$$X_k^i = (X_{k-1} \circledast B^i) \cup A$$

$$i = 1, 2, 3, 4 \ \text{ and } \ k = 1, 2, 3, \ldots$$

with $X_0^i = A$.

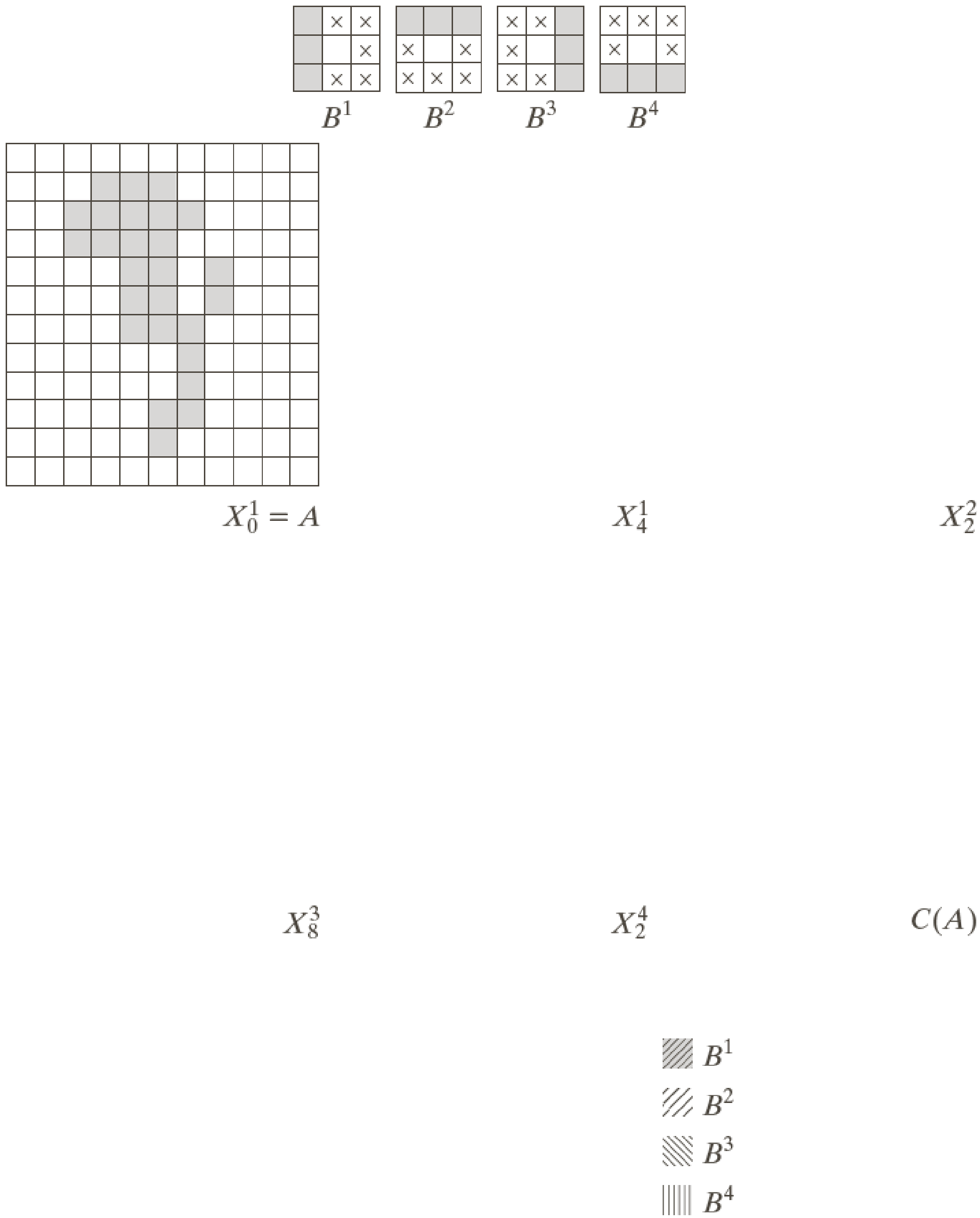
When the procedure converges, or $X_k^i = X_{k-1}^i$, let $D^i = X_k^i$,

the convex hull of A is

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

Convex Hull

- The hit-or-miss transformation is applied iteratively to A with B^1.
  - When there is no further change, we perform union with A and call the result D^1.
- Procedure is repeated with B^2 and so on.
- Union of four resulting Ds constitute the convex hull.
  - We use simplified hit0or-miss transform in which no background match is done.



a
b c d
e f g
h

FIGURE 9.19 (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.

Convex Hull

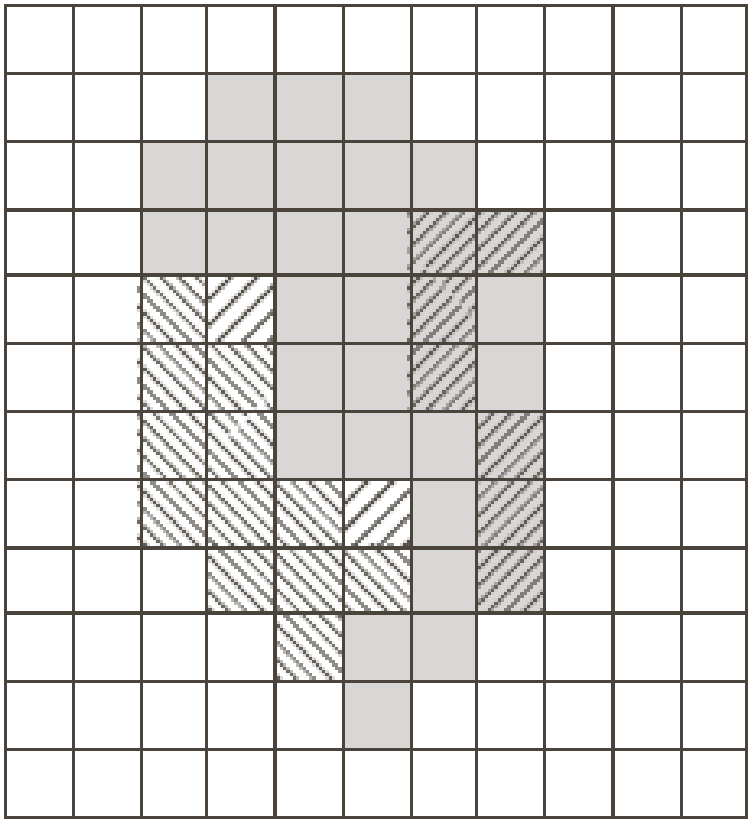


FIGURE 9.20 Result of limiting growth of the convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

Some Basic Morphological Algorithms

- **Thinning**

  The thinning of a set *A* by a structuring element *B*, defined by:

$$A \otimes B = A - (A \circledast B)$$

$$= A \cap (A \circledast B)^c$$

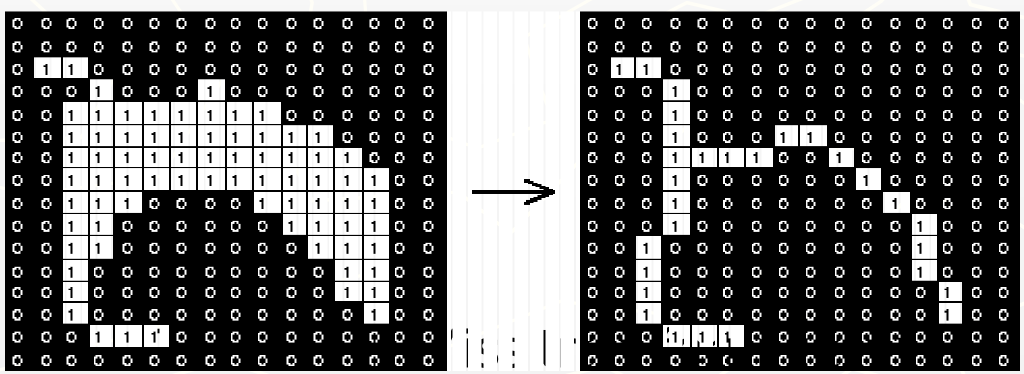
- Used to remove selected foreground pixels from binary images.
- After edge detection, lines are often thicker than one pixel.
- Thinning can be used to thin those lines to one pixel width.

Thinning

- If foreground and background fit the structuring element exactly, then the pixel at the origin of the SE is set to 0.
- Note that the value of the SE at the origin is 1 or *don't care*!

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	



Thinning

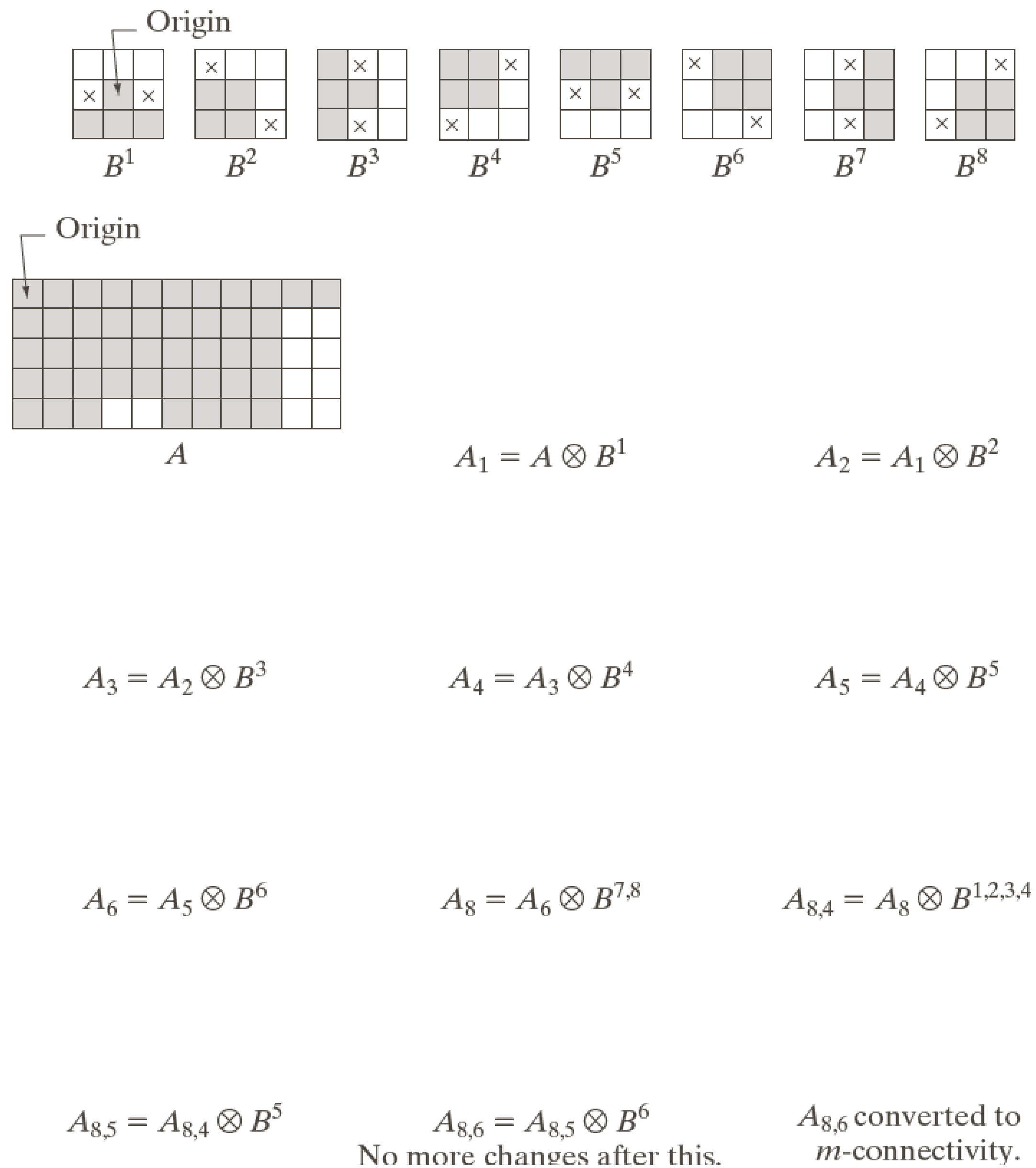
- A more useful expression for thinning *A* symmetrically is based on a sequence of structuring elements:

$$\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$$

where B^i is a rotated version of $B^{i\text{-}1}$

The thinning of A by a sequence of structuring element $\{B\}$

$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$$



a
b c d
e f g
h i j
k l m

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.

Some Basic Morphological Algorithms

- **Thickening:**

The thickening is defined by the expression

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

The thickening of A by a sequence of structuring elements $\{B\}$

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$$

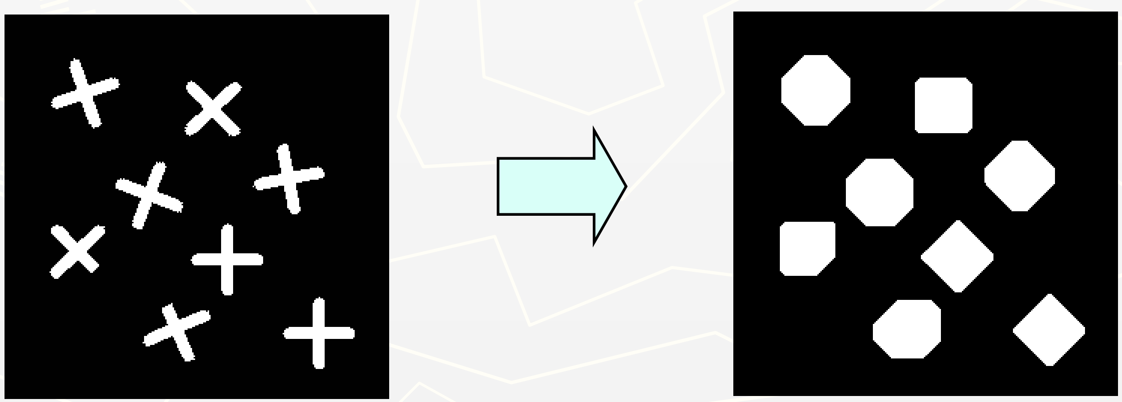
In practice, the usual procedure is to thin the background of the set and then complement the result.

Thickening

- Used to grow selected regions of foreground pixels
- E.g. applications like approximation of *convex hull*
- If foreground and background match exactly the SE, then set the pixel at its origin to 1!
- Note that the value of the SE at the origin is 0 or *don't care*!

1	1	
1	0	
1		0

	1	1
	0	1
0		1



Some Basic Morphological Algorithms

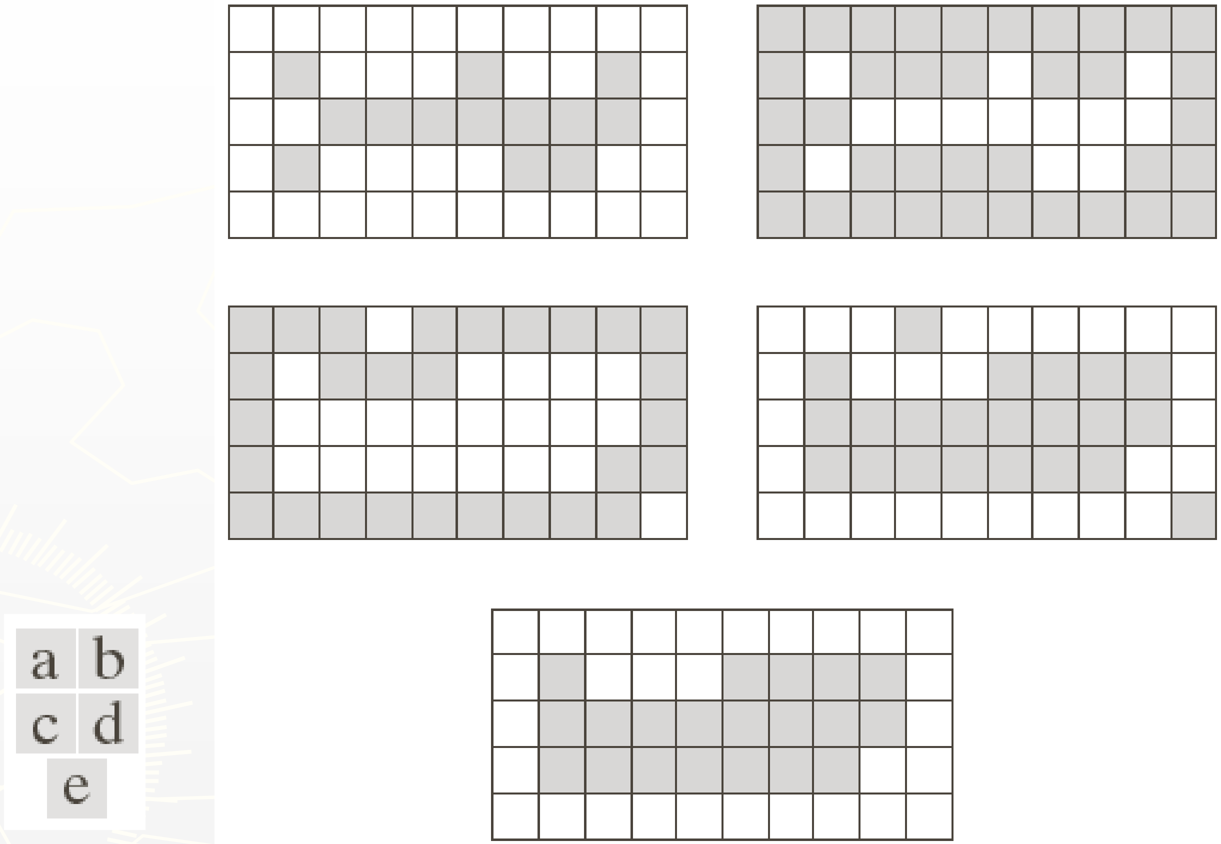
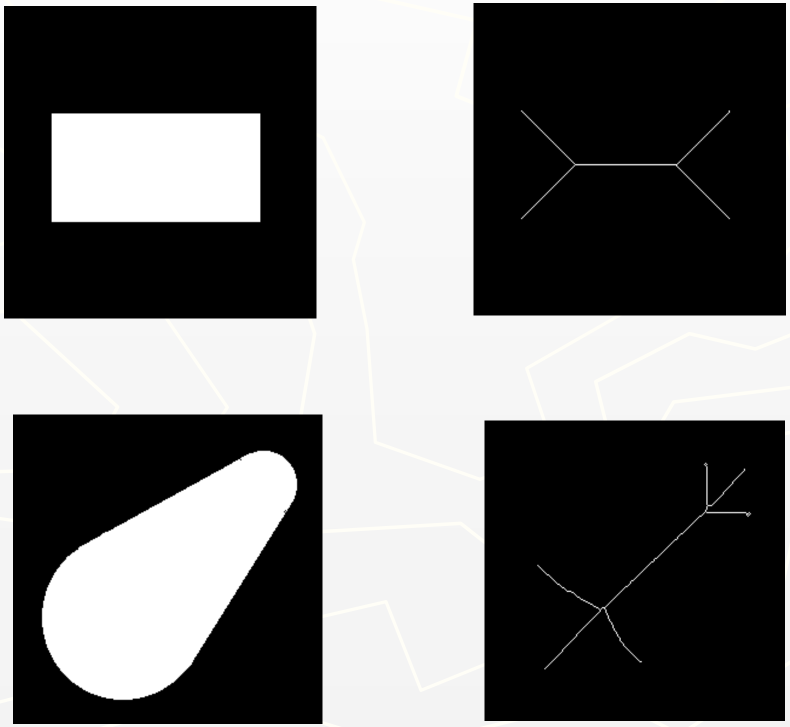


FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

Some Basic Morphological Algorithms

- Skeletons
  - To extract a region-based shape feature representing the general form of an object



Skeletonization

- Skeletons
  - To extract a region-based shape feature representing the general form of an object

A skeleton, $S(A)$ of a set A has the following properties

a. 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 containing $(D)_z$ and included in A. The disk $(D)_z$ is called a maximum disk.

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

Skeletons

a	b
c	d

FIGURE 9.23
(a) Set A.
(b) Various positions of maximum disks with centers on the skeleton of A.
(c) Another maximum disk on a different segment of the skeleton of A.
(d) Complete skeleton.

Skeletons

The skeleton of A can be expressed in terms of erosion and openings.

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

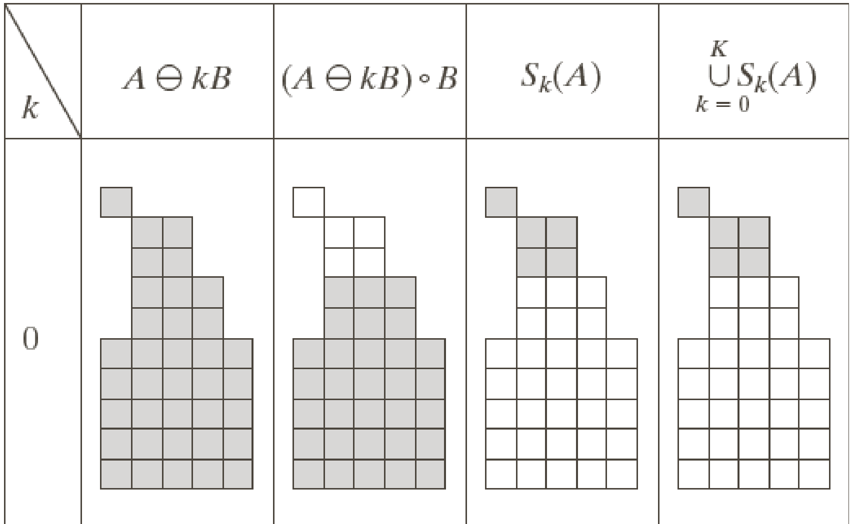
with $K = \max\{k \mid A \ominus kB \neq \phi\}$;

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

where B is a structuring element, and

$$(A \ominus kB) = ((..((A \ominus B) \ominus B) \ominus ...) \ominus B)$$

k successive erosions of A.



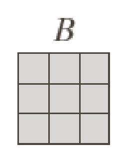


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

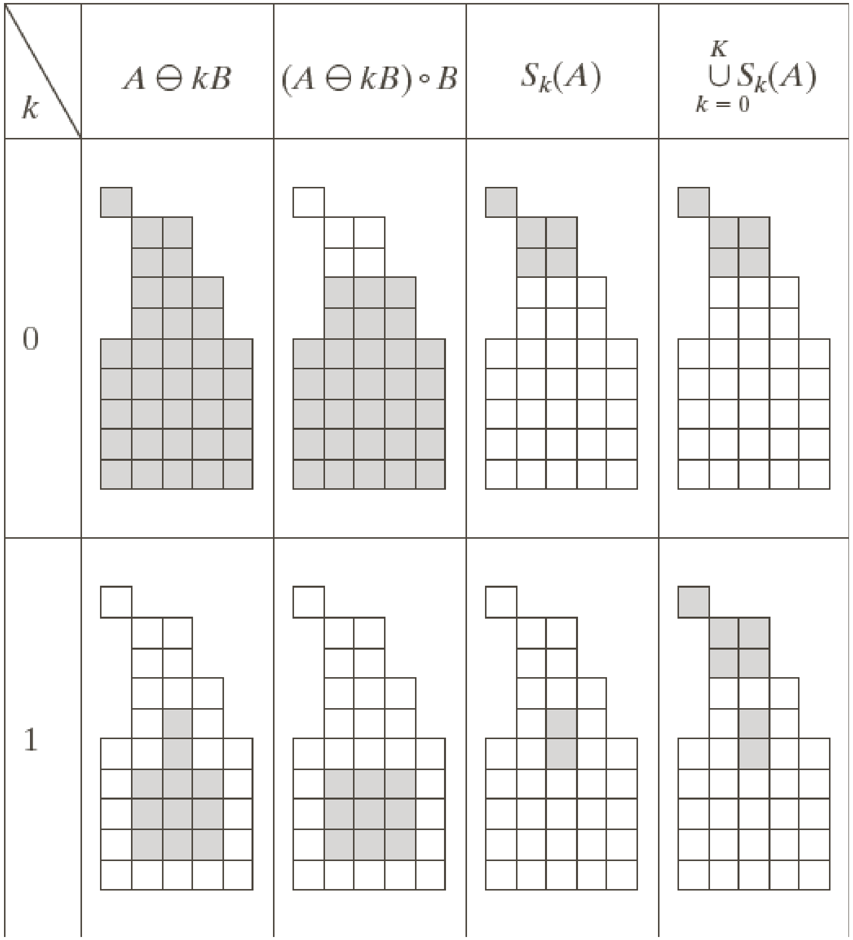
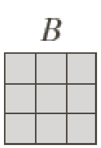


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.



Skeletons

A can be reconstructed from the subsets by using

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

where $S_k(A) \oplus kB$ denotes k successive dilations of A.

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B)\ldots \oplus B)$$

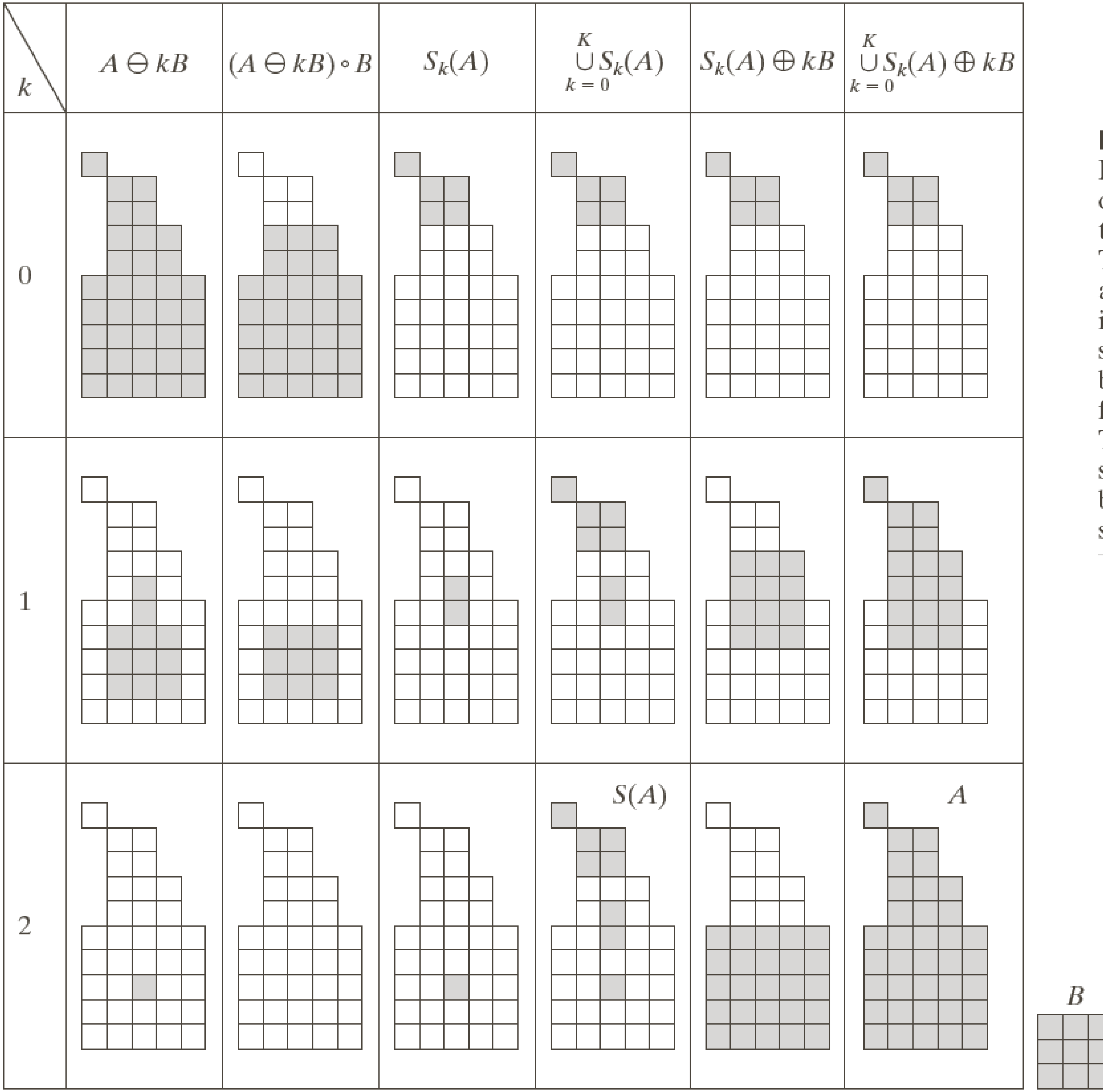


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

Summary (1)

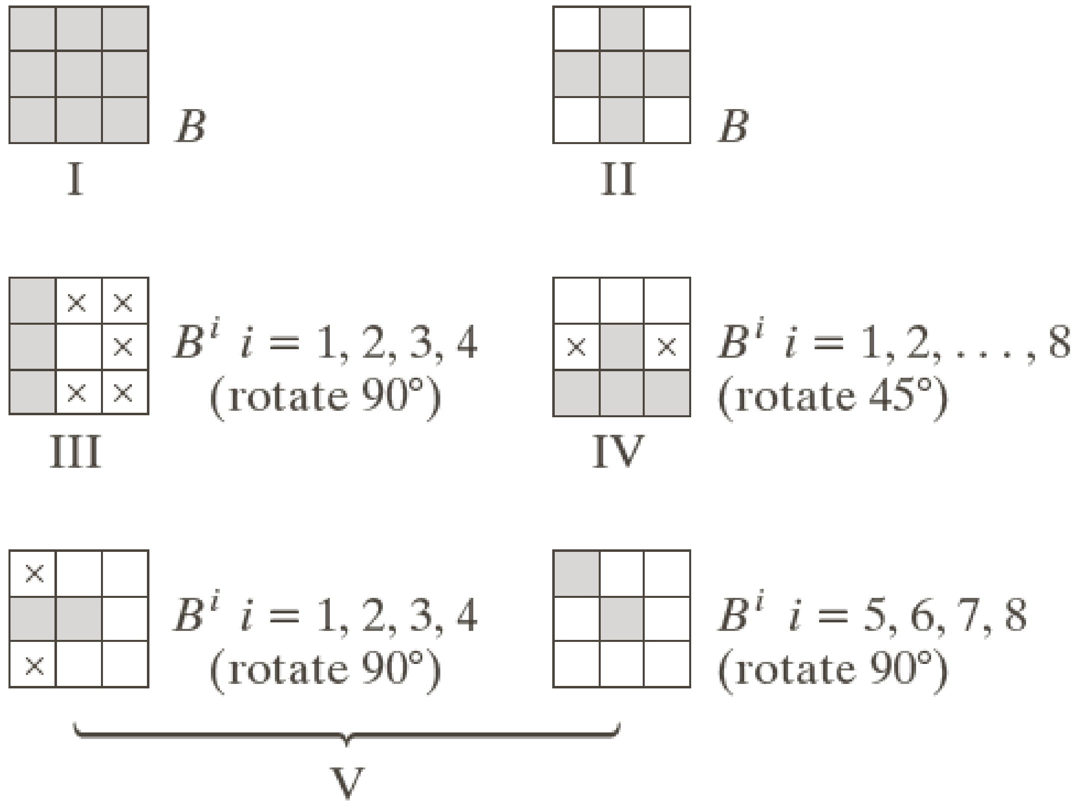


FIGURE 9.33 Five basic types of structuring elements used for binary morphology. The origin of each element is at its center and the ×'s indicate "don't care" values.

Summary (2)

TABLE 9.1 Summary of morphological operations and their properties.

Operation	Equation	Comments (The Roman numerals refer to the structuring elements in Fig. 9.33.)
Translation	$(B)_z = \{w \mid w = b + z, \text{ for } b \in B\}$	Translates the origin of B to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \{z \mid (\hat{B}_z) \cap A \neq \varnothing\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	"Contracts" the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)

(Continued)

Operation	Equation	Comments (The Roman numerals refer to the structuring elements in Fig. 9.33.)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)
Hit-or-miss transform	$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$ $= (A \ominus B_1) - (A \oplus \hat{B}_2)$	The set of points (coordinates) at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Hole filling	$X_k = (X_{k-1} \oplus B) \cap A^c$; $k = 1, 2, 3, \ldots$	Fills holes in A; X_0 = array of 0s with a 1 in each hole. (II)
Connected components	$X_k = (X_{k-1} \oplus B) \cap A$; $k = 1, 2, 3, \ldots$	Finds connected components in A; X_0 = array of 0s with a 1 in each connected component. (I)
Convex hull	$X_k^i = (X_{k-1}^i \circledast B^i) \cup A$; $i = 1, 2, 3, 4$; $k = 1, 2, 3, \ldots$; $X_0^i = A$; and $D^i = X_{\text{conv}}^i$	Finds the convex hull $C(A)$ of set A, where "conv" indicates convergence in the sense that $X_k^i = X_{k-1}^i$. (III)
Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	Thins set A. The first two equations give the basic definition of thinning. The last equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)
Thickening	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} =$ $((\ldots(A \odot B^1) \odot B^2 \ldots) \odot B^n)$	Thickens set A. (See preceding comments on sequences of structuring elements.) Uses IV with 0s and 1s reversed.
Skeletons	$S(A) = \bigcup_{k=0}^{K} S_k(A)$ $S_k(A) = \bigcup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B]\}$ Reconstruction of A: $A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$	Finds the skeleton $S(A)$ of set A. The last equation indicates that A can be reconstructed from its skeleton subsets $S_k(A)$. In all three equations, K is the value of the iterative step after which the set A erodes to the empty set. The notation $(A \ominus kB)$ denotes the kth iteration of successive erosions of A by B. (I)

(Continued)

END