Introduction to Image Processing

Prof. Alexandre Zaghetto
http://alexandre.zaghetto.com
zaghetto@unb.br

University of Brasília
Department of Computer Science
LISA: Laboratory of Imagens, Signals and Acoustics

Topic 05
Morphological Image Processing

1. Preliminaries

- The language of mathematical morphology is **set theory**.

- **Sets** in mathematical morphology represent **objects in an image**.

- **Binary images**, can be represented as sets whose components are in Z^2. Coordinates *(x, y)* of black (or white) pixels.

- **Grayscale images** can be represented as sets whose components are in Z^3 (x, y, gray level).

  - In this case, two components of each element of the set refer to the coordinates of a pixel, and the third corresponds to its discrete intensity value.

1. Preliminaries

- The concepts of set **reflection** and **translation** are used extensively in morphology.

- Reflection $\hat{B} = \{w | w = -b, \quad \text{for} \quad b \in B\}$

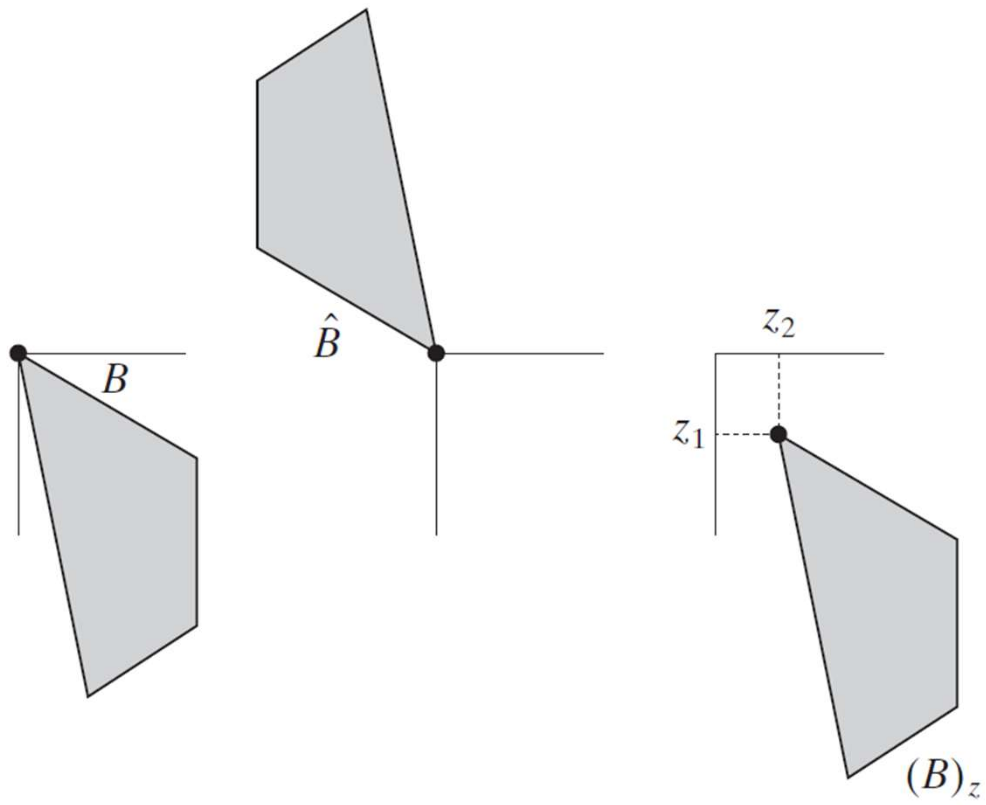
- If B is the set of pixels (2-D points) representing an object in an image, then $\hat{B}$ is simply the set of points in whose coordinates have been replaced by (-x, -y).

- The translation $(B)_z = \{c | c = b + z, \quad \text{for} \quad b \in B\}$

- If B is the set of pixels representing an object in an image, then $(B)_z$ is the set of points in whose coordinates have been replaced by $(x+z_1, y+z_2)$.

1. Preliminaries

- Reflection and translation



a b c

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

1. Preliminaries

• **Structuring elements** (SEs) are small sets or subimages used to probe an image for properties of interest.

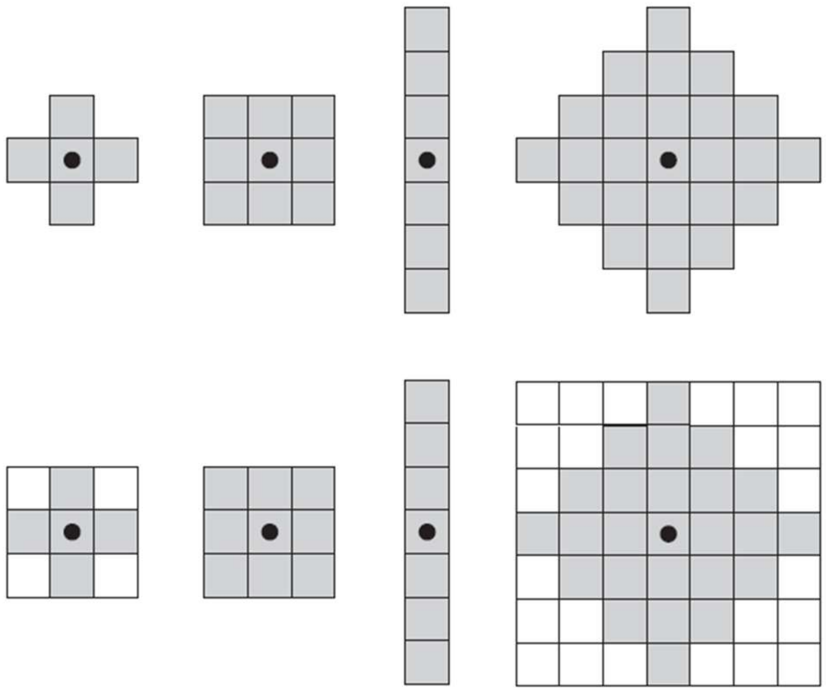


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

1. Preliminaries

• **Structuring elements** (SEs) are small sets or subimages used to probe an image for properties of interest.

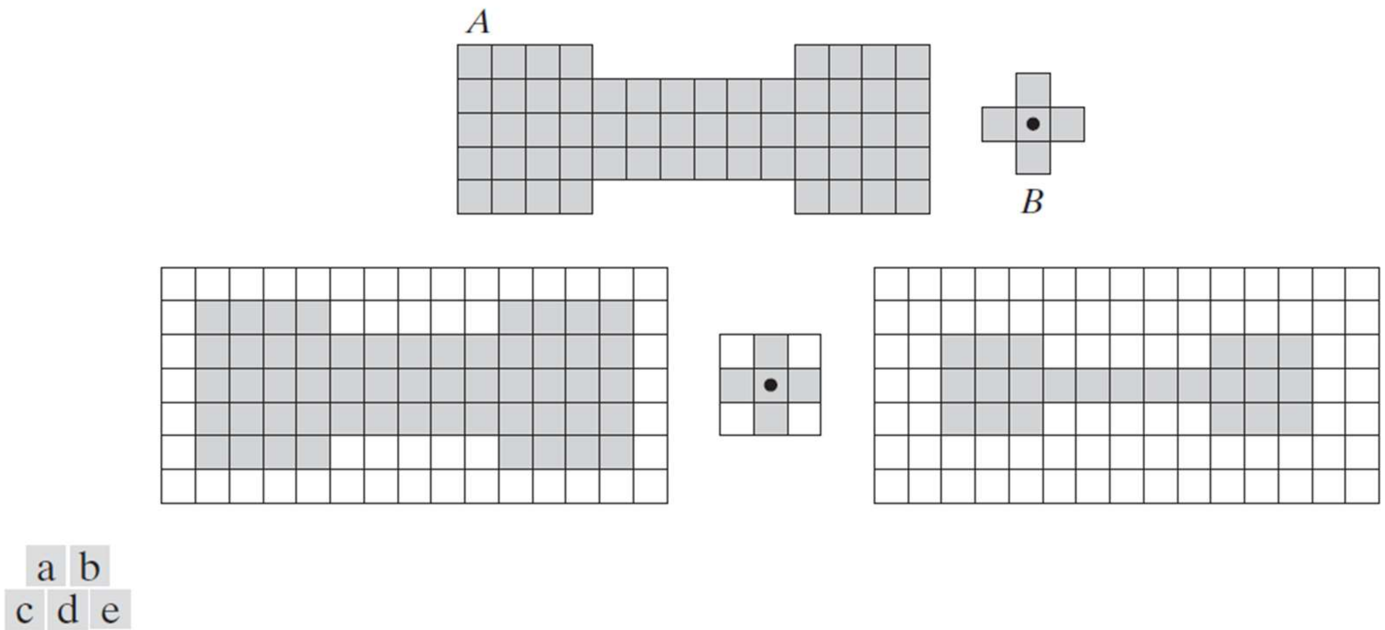
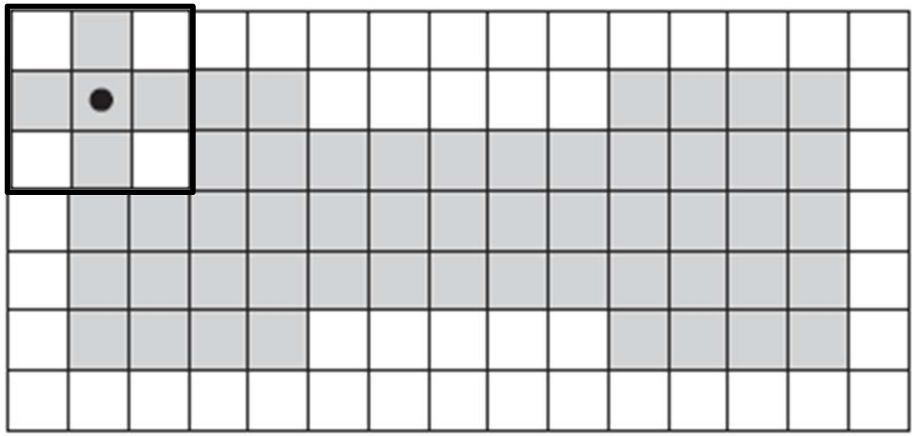


FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

1. Preliminaries

- Example

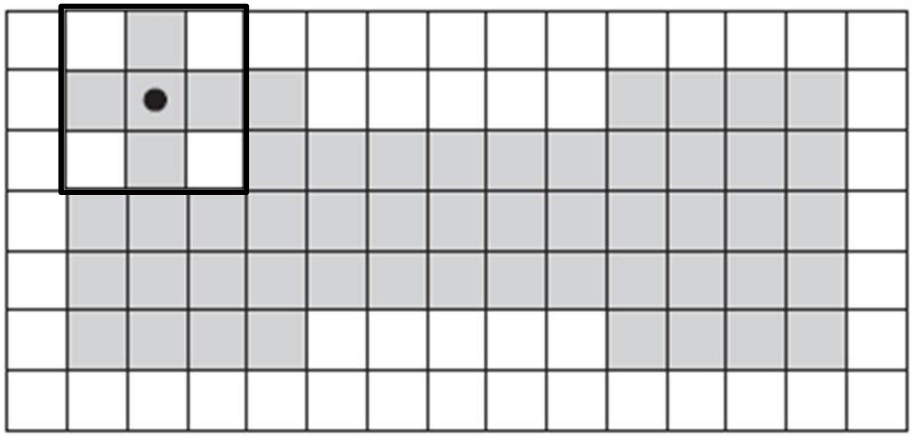
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

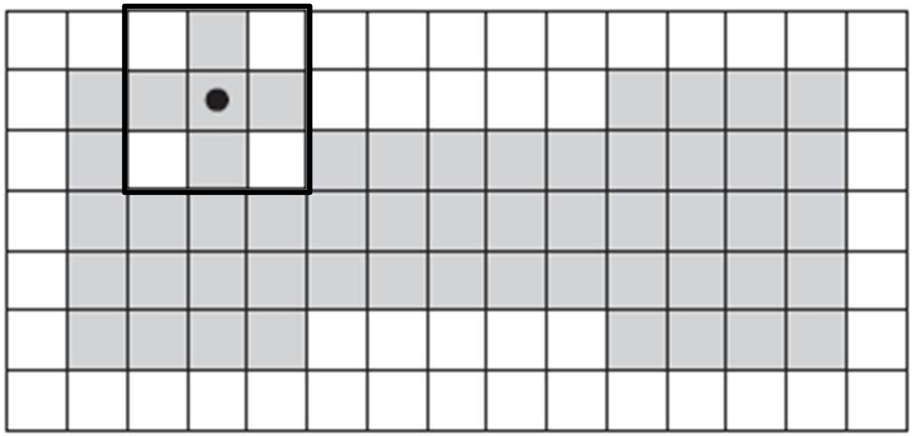
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

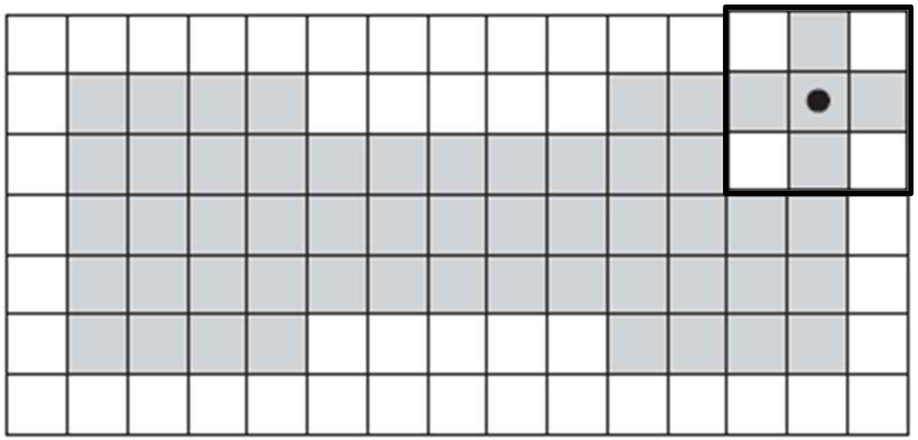
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

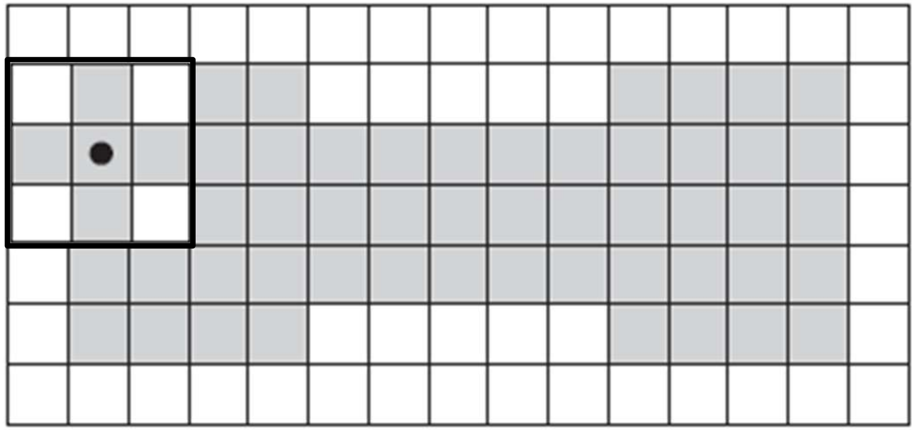
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

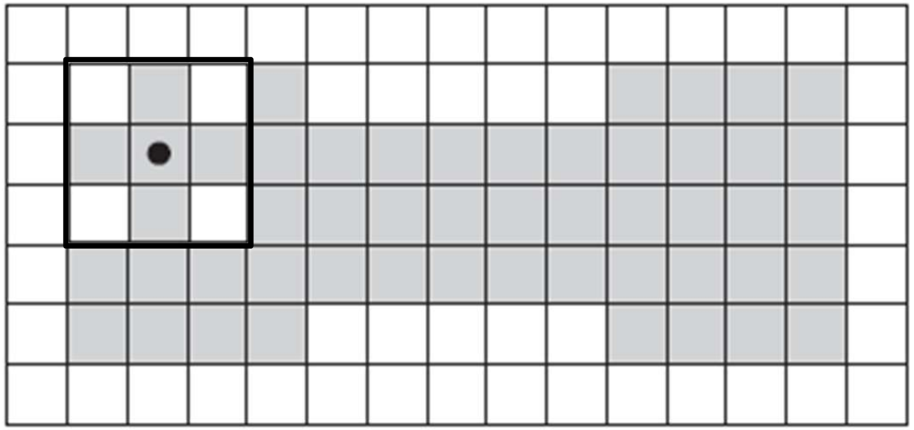
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

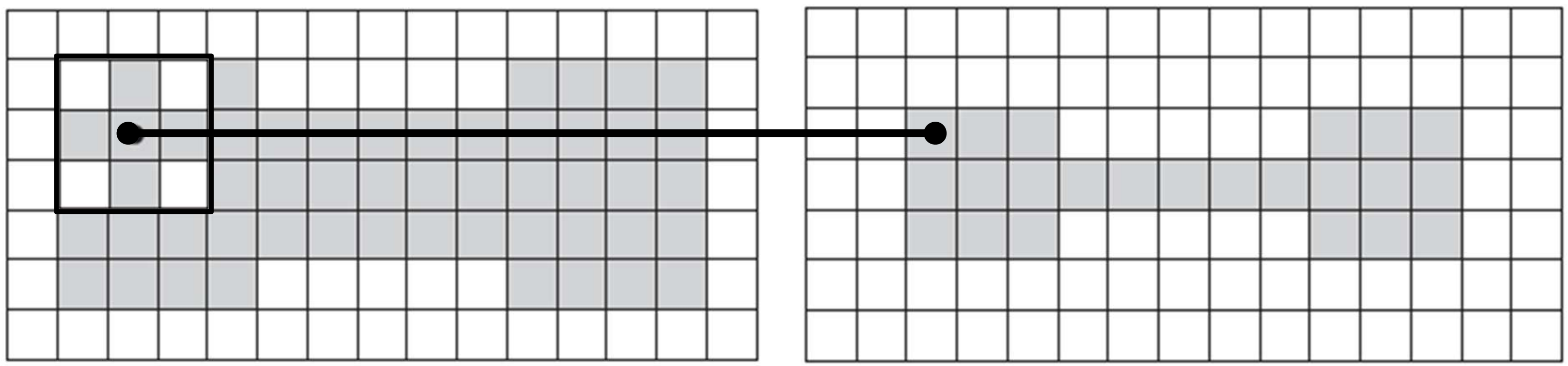
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

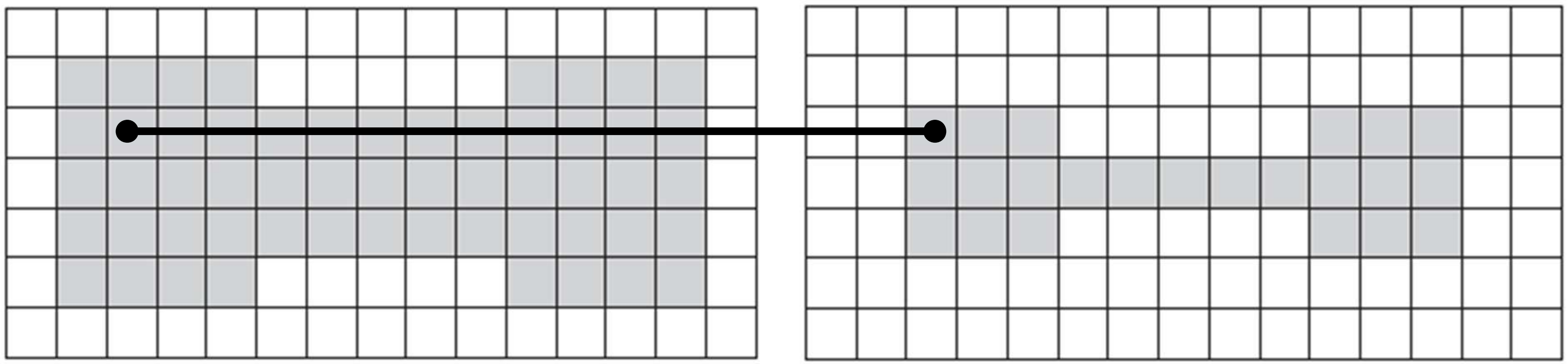
  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



1. Preliminaries

- Example

  - Create a new set by running B over A so that the origin of B visits every element of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of the new set (shown shaded); else mark it as not being a member of the new set (shown not shaded).



2. Erosion and Dilation

- **Erosion** and **dilation** are fundamental to morphological processing.

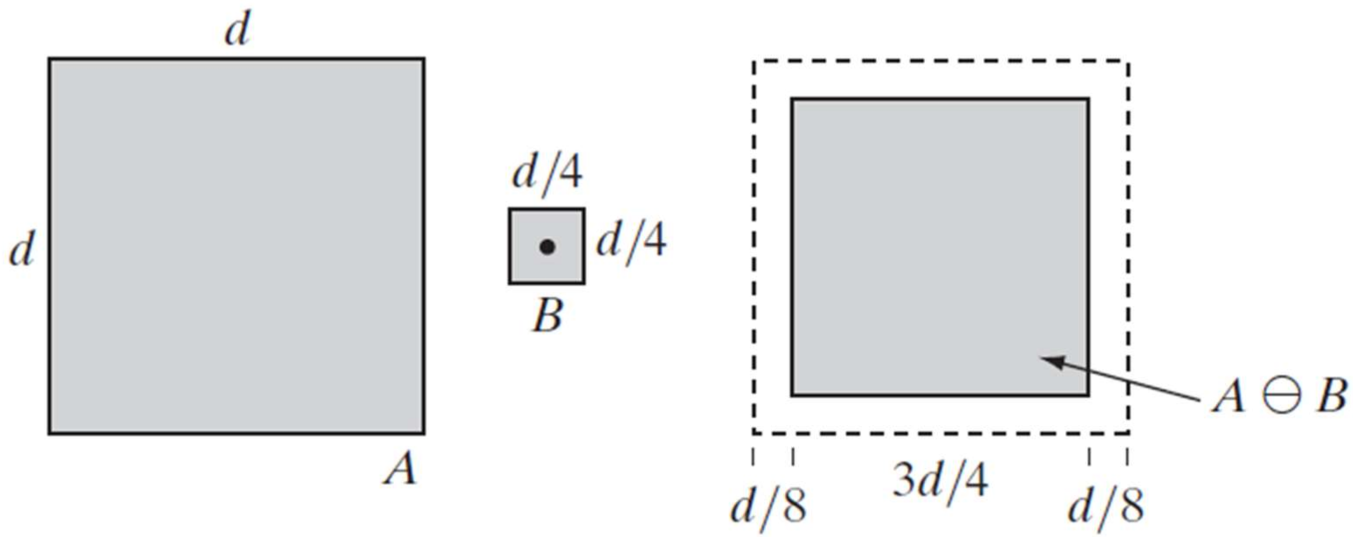
- In fact, many of the morphological algorithms discussed here are based on these two primitive operations.

2. Erosion and Dilation

- With A and B as sets in Z^2, the **erosion** of A by B is defined as

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

- 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.

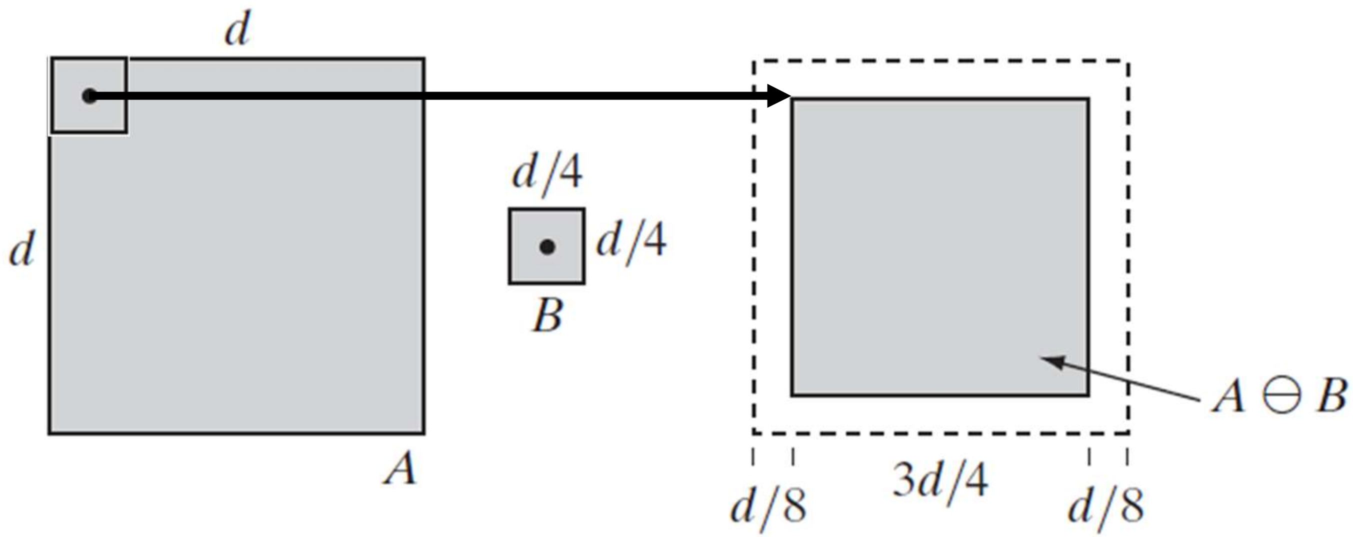


2. Erosion and Dilation

• With A and B as sets in Z^2, the **erosion** of A by B is defined as

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

• 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.

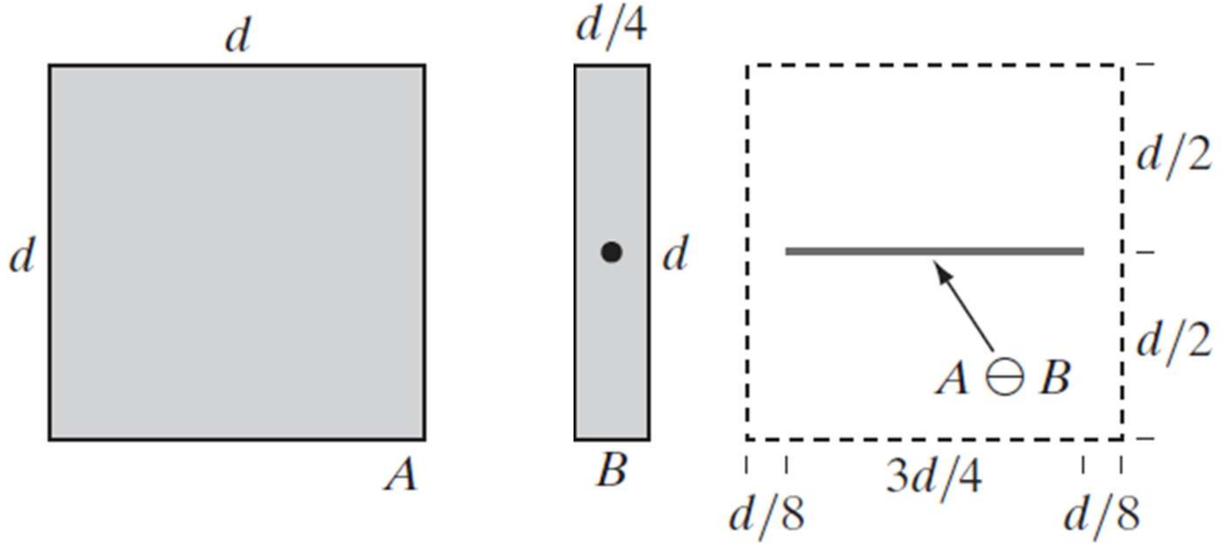


2. Erosion and Dilation

• With A and B as sets in Z^2, the **erosion** of A by B is defined as

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

• 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.

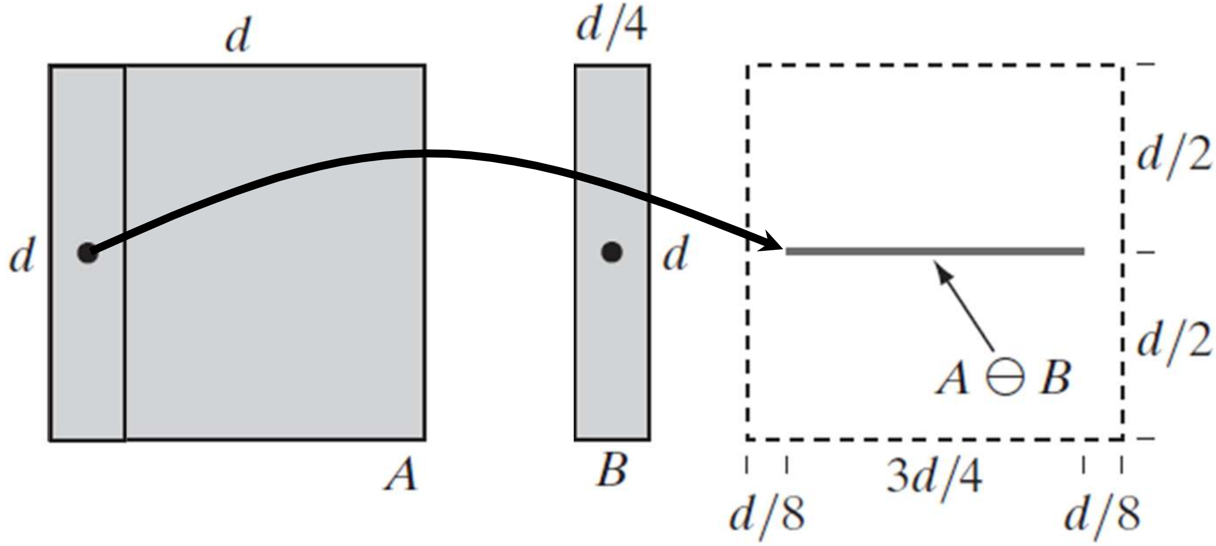


2. Erosion and Dilation

• With A and B as sets in Z^2, the **erosion** of A by B is defined as

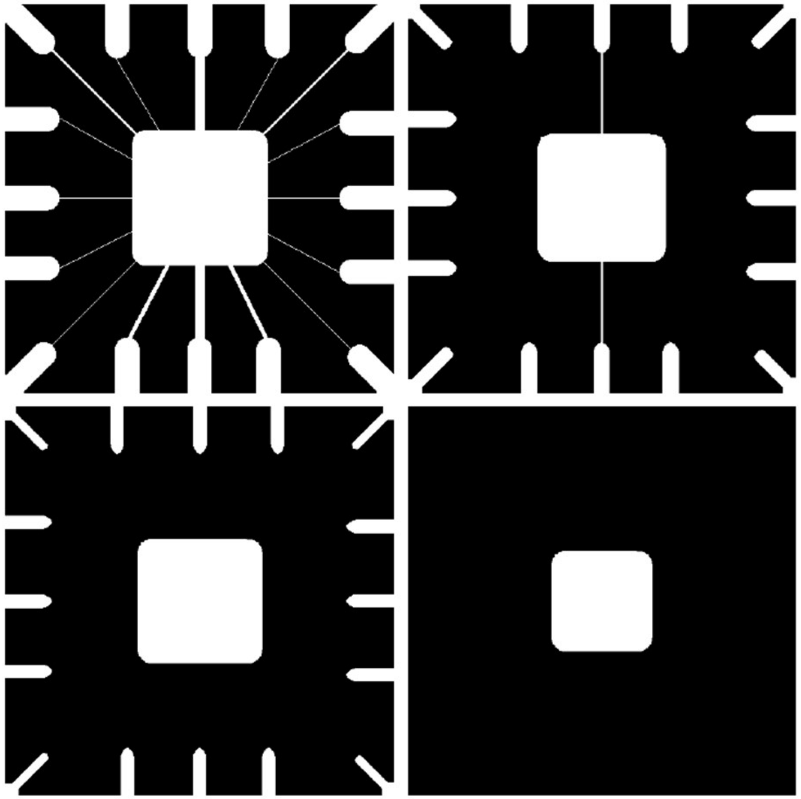
$$A \ominus B = \{z | (B)_z \subseteq A\}$$

• 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.



2. Erosion and Dilation

- Suppose that we wish to remove the lines connecting the center region to the border pads.

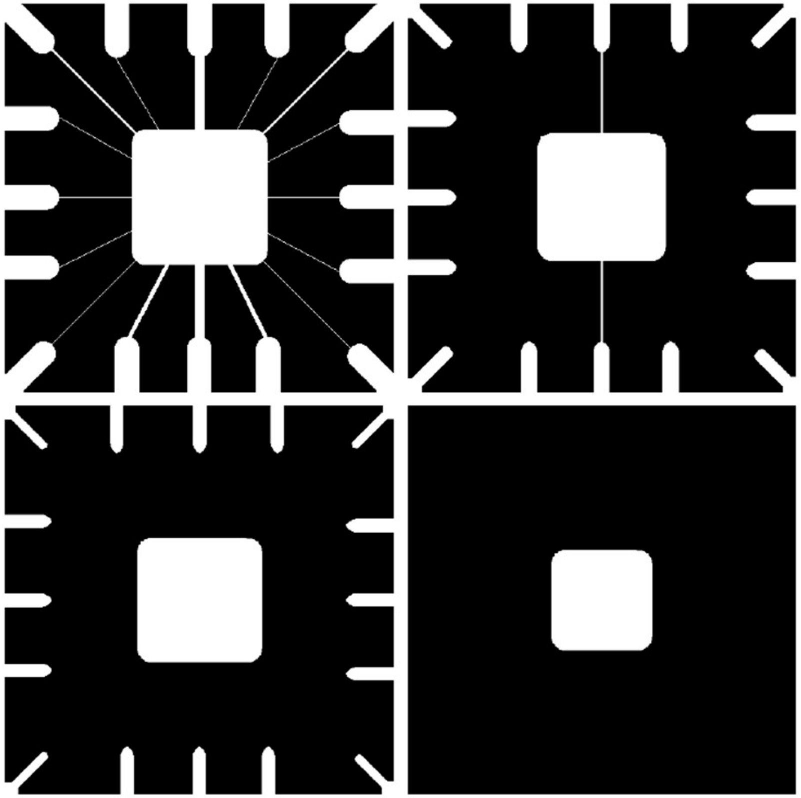


a b
c d

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.

2. Erosion and Dilation

- MATLAB: s22Erosion.m



2. Erosion and Dilation

- We see from previous examples that **erosion shrinks** or **thins** objects in a binary image.

- In fact, we can view **erosion** as a morphological **filtering operation**.

    - Details smaller than the structuring element are filtered (removed) from the image.

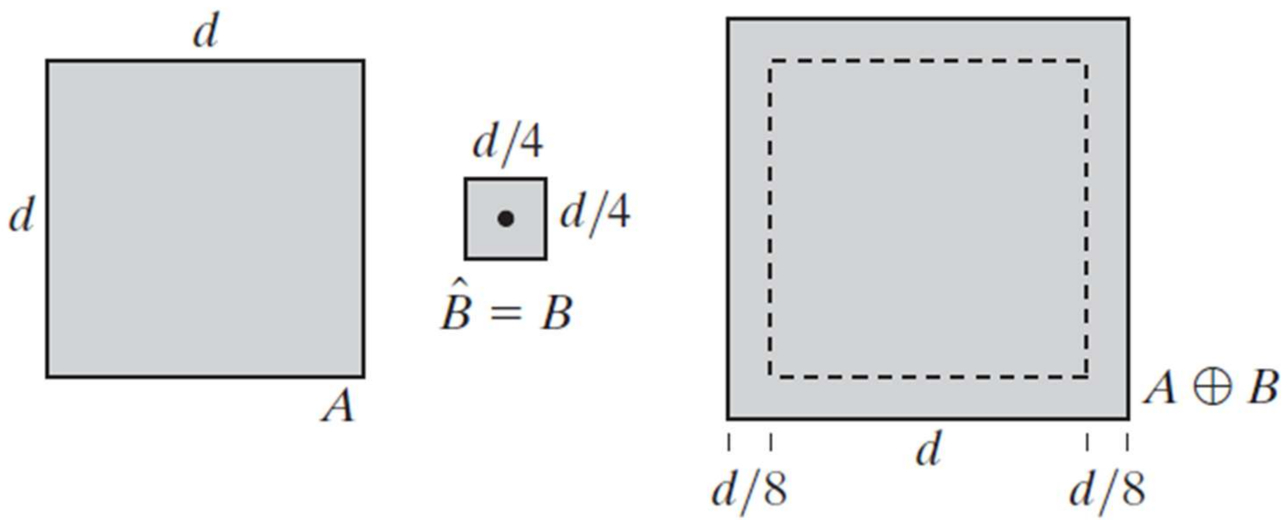
- In the last example, erosion performed the function of a "line filter."

2. Erosion and Dilation

- With A and B as sets in Z^2, the **dilation** of A by B is defined as

$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \varnothing \right\}$$

- The dilation of A by B then is the set of all displacements z, such that A and $\hat{B}$ overlap by at least one element.

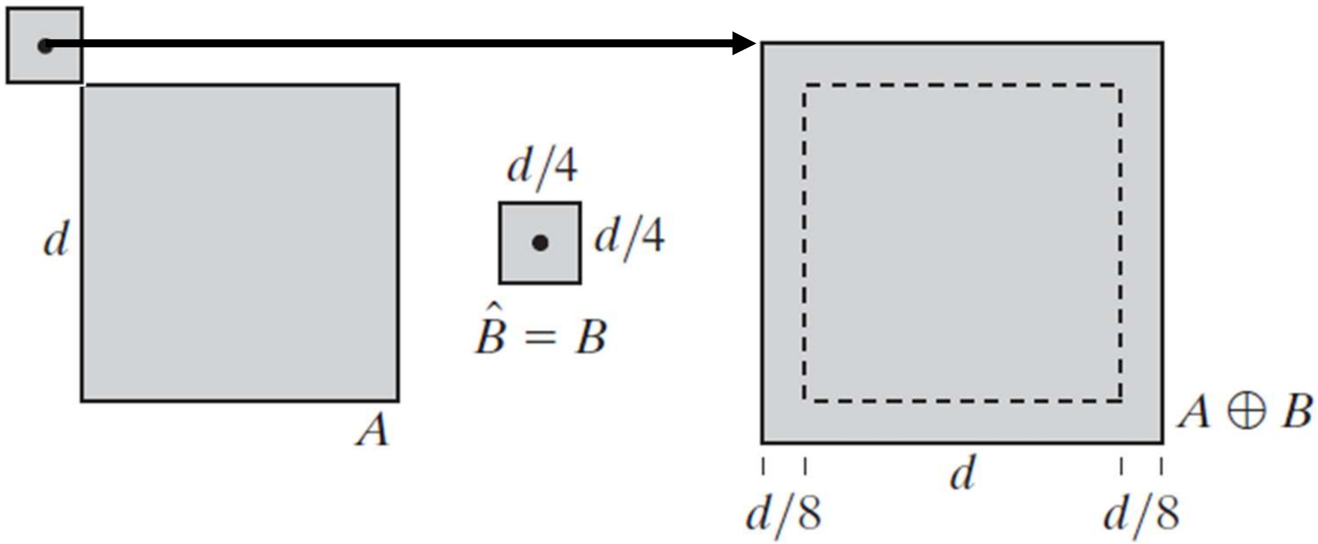


2. Erosion and Dilation

- With A and B as sets in Z^2, the **dilation** of A by B is defined as

$$A \oplus B = \left\{ z \,\middle|\, (\hat{B})_z \cap A \neq \varnothing \right\}$$

- The dilation of A by B then is the set of all displacements z, such that A and $\hat{B}$ overlap by at least one element.

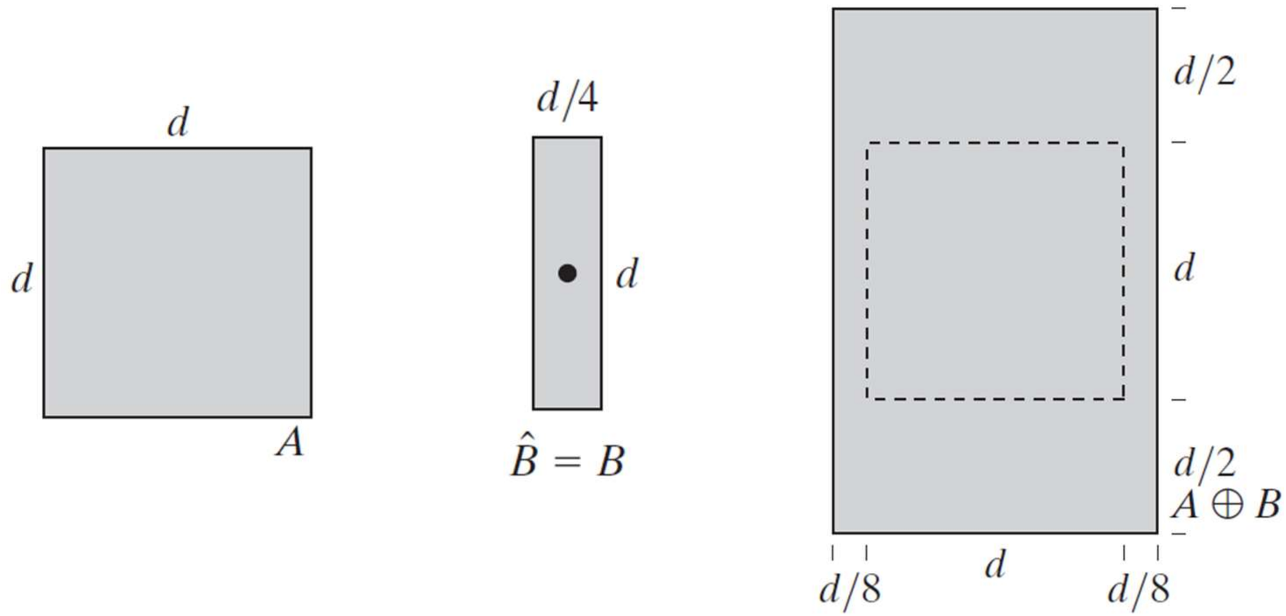


2. Erosion and Dilation

- With A and B as sets in Z^2, the **dilation** of A by B is defined as

$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \varnothing \right\}$$

- The dilation of A by B then is the set of all displacements z, such that A and $\hat{B}$ overlap by at least one element.

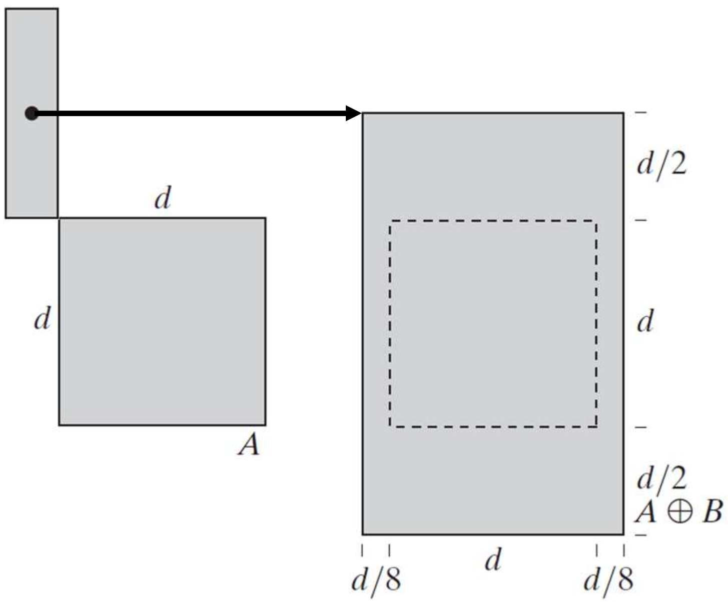


2. Erosion and Dilation

- With A and B as sets in Z^2, the **dilation** of A by B is defined as

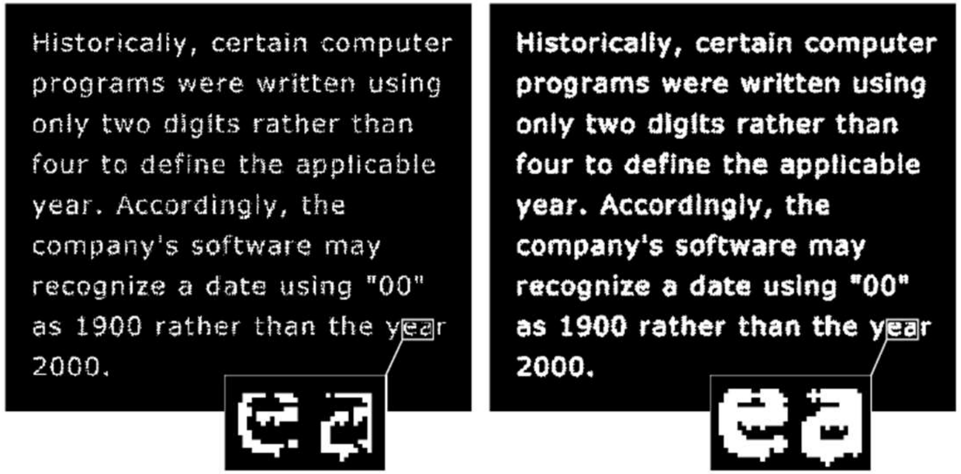
$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \varnothing \right\}$$

- The dilation of A by B then is the set of all displacements z, such that A and $\hat{B}$ overlap by at least one element.



2. Erosion and Dilation

- One of the simplest applications of dilation is for bridging gaps.

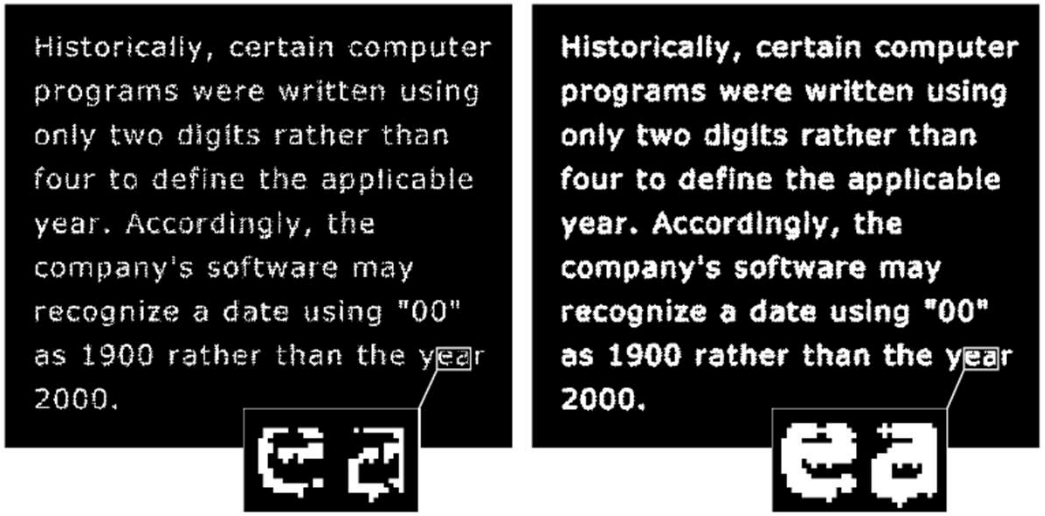


0	1	0
1	1	1
0	1	0

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.

2. Erosion and Dilation

- MATLAB: s29Dilation.m



0	1	0
1	1	1
0	1	0

2. Erosion and Dilation

- Duality

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

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

  - The duality property is useful particularly when the structuring element is symmetric with respect to its origin

  - We can obtain the erosion of an image by simply by dilating its background with the same structuring element and complementing the result.

  - The proof of this result is left as an exercise.

2. Erosion and Dilation

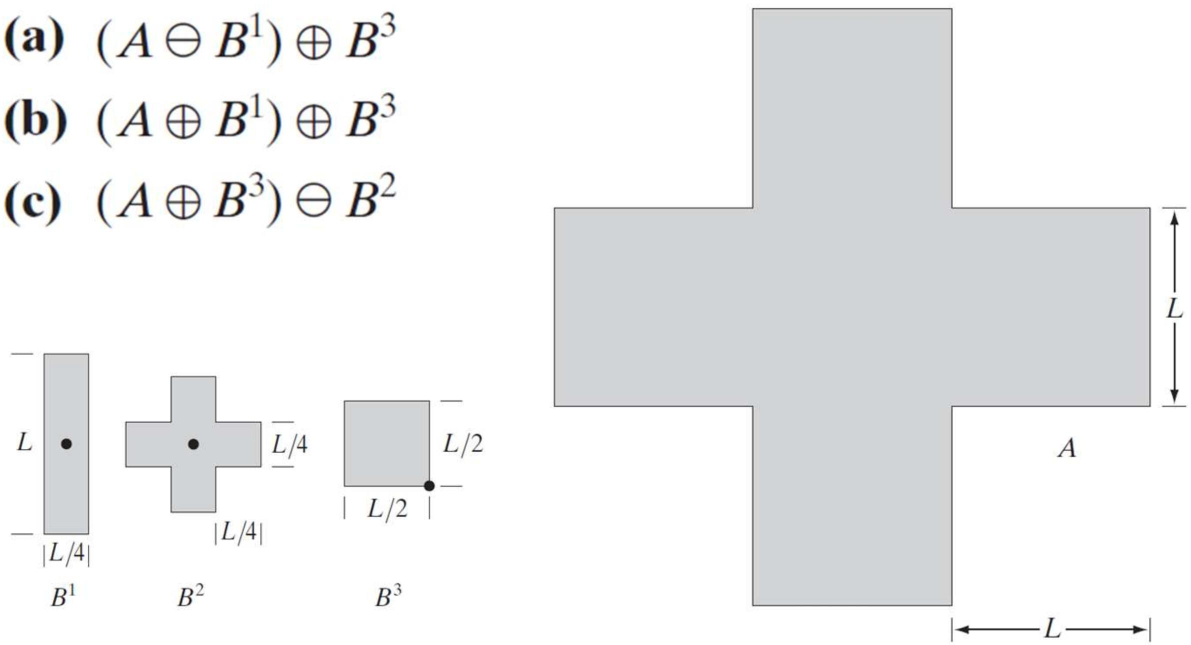
- Exercise

Let A denote the set shown shaded in the following figure. Refer to the structuring elements shown (the black dots denote the origin). Sketch the result of the following morphological operations:

(a) $(A \ominus B^1) \oplus B^3$

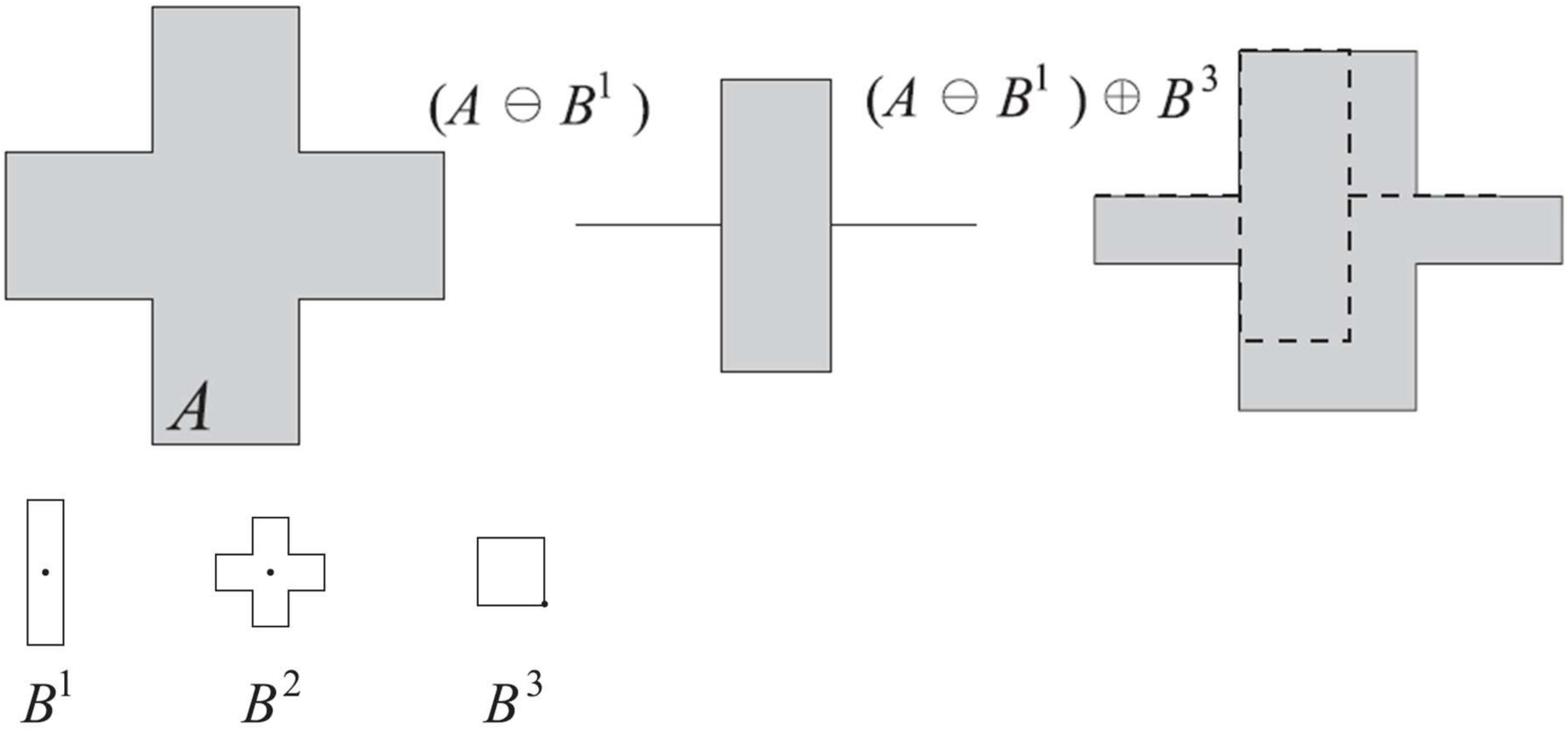
(b) $(A \oplus B^1) \oplus B^3$

(c) $(A \oplus B^3) \ominus B^2$



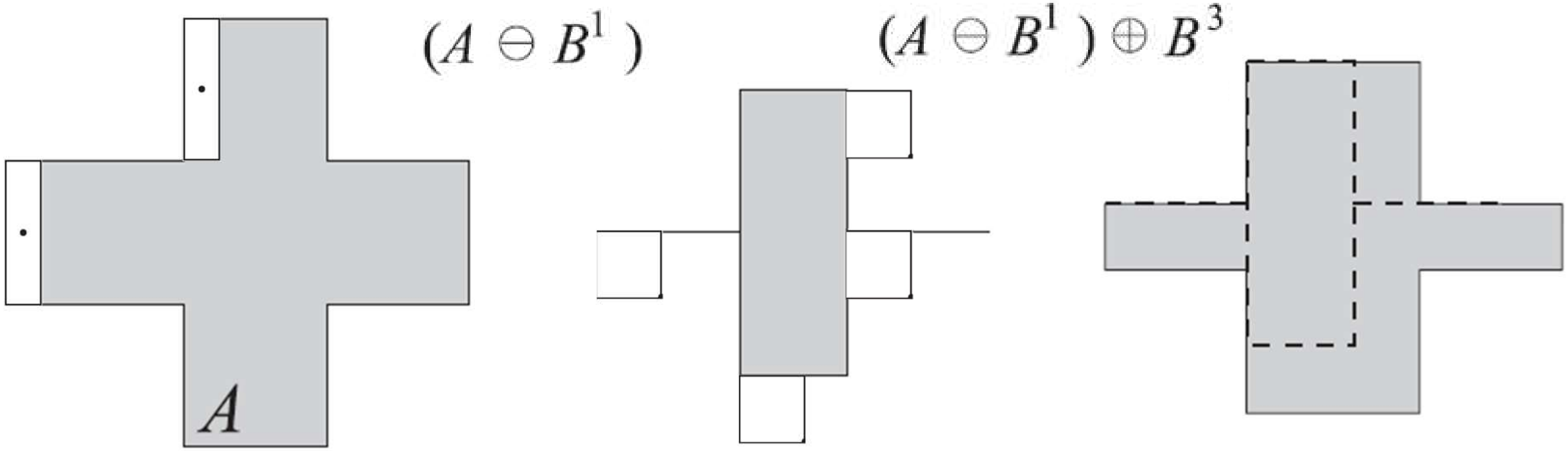
2. Erosion and Dilation

- Exercise



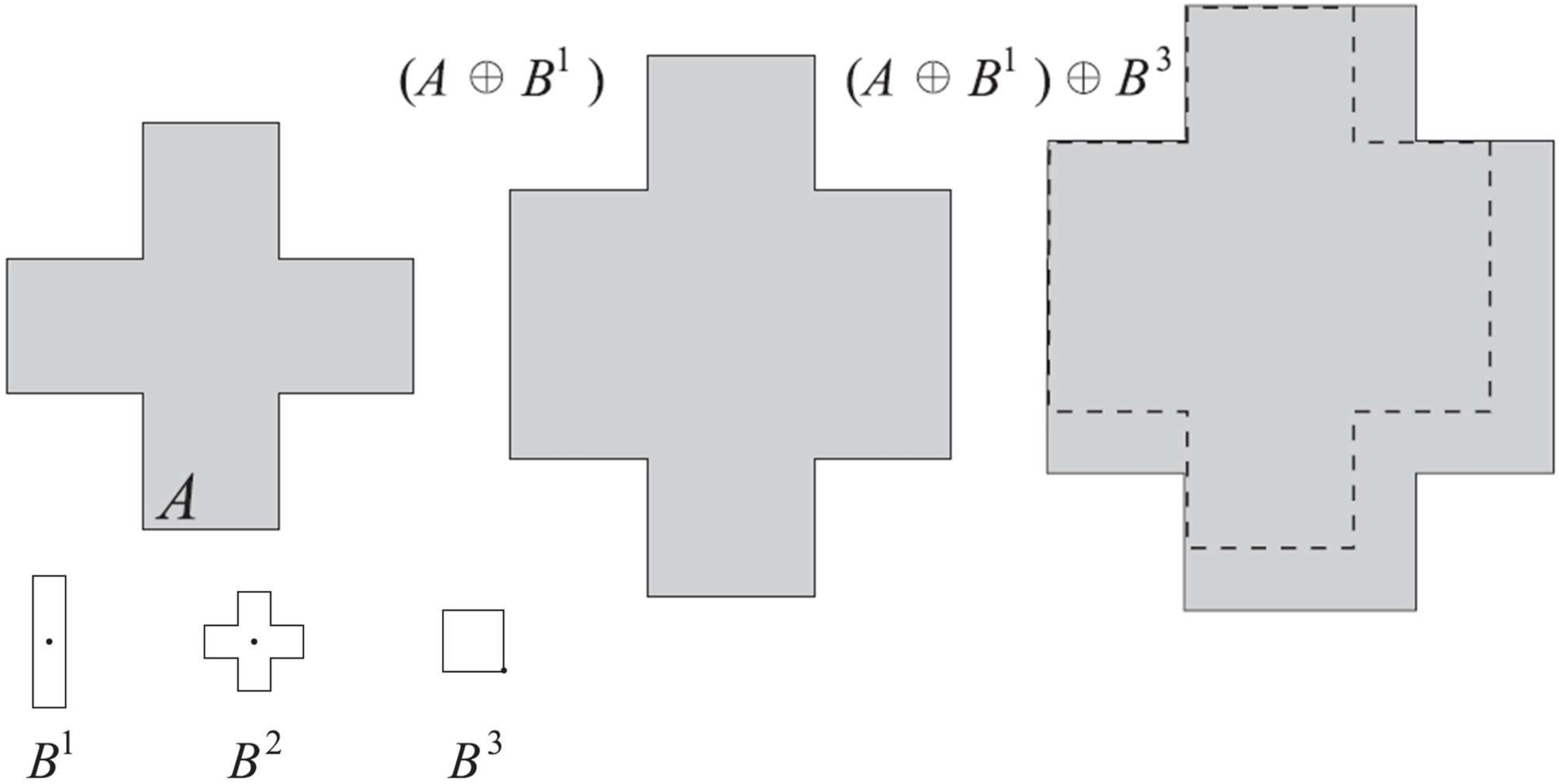
2. Erosion and Dilation

- Exercise



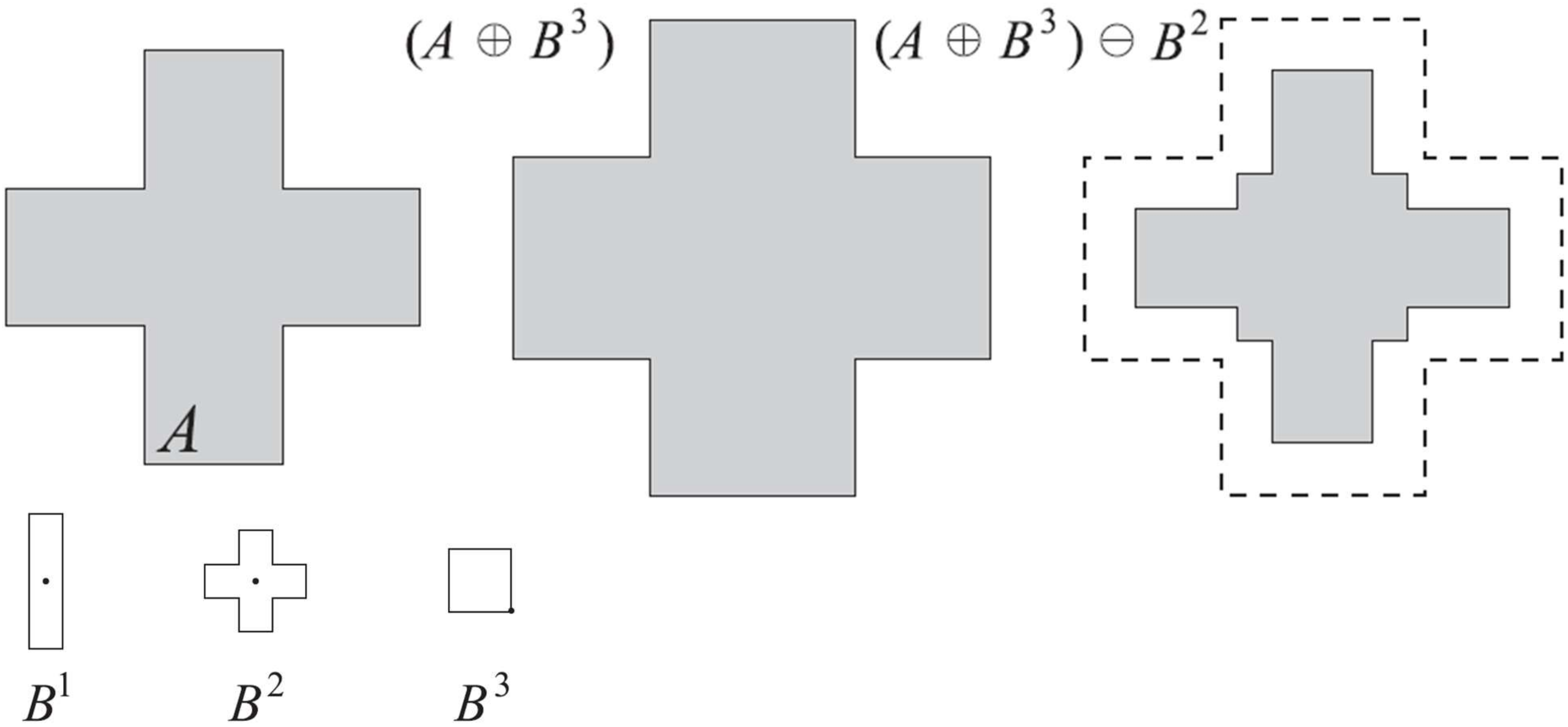
2. Erosion and Dilation

- Exercise



2. Erosion and Dilation

- Exercise



3. Opening and Closing

• As you have seen, dilation expands the components of an image and erosion shrinks them.

• Opening generally:

- smoothes the contour of an object
- breaks narrow isthmuses; and
- eliminates thin protrusions.

• Closing generally:

- fuses narrow breaks and long thin gulfs;
- eliminates small holes, and
- Fills gaps in the contour.

3. Opening and Closing

- The opening of set A by structuring element B is defined as

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

- Thus, the opening of 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 is defined as

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

- The closing of A by B is simply the dilation of A by B followed by the erosion of the result by B.

3. Opening and Closing

- Geometric interpretation
  - Opening

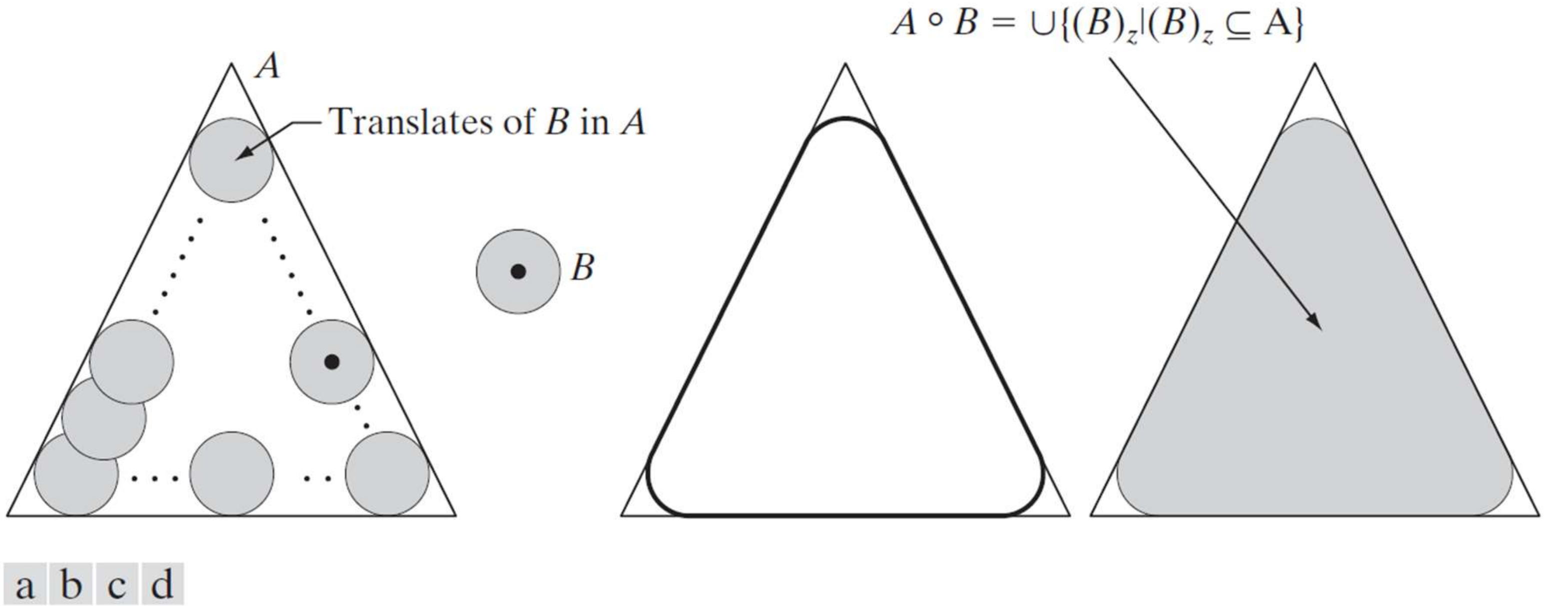


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

3. Opening and Closing

- Geometric interpretation

  - Closing

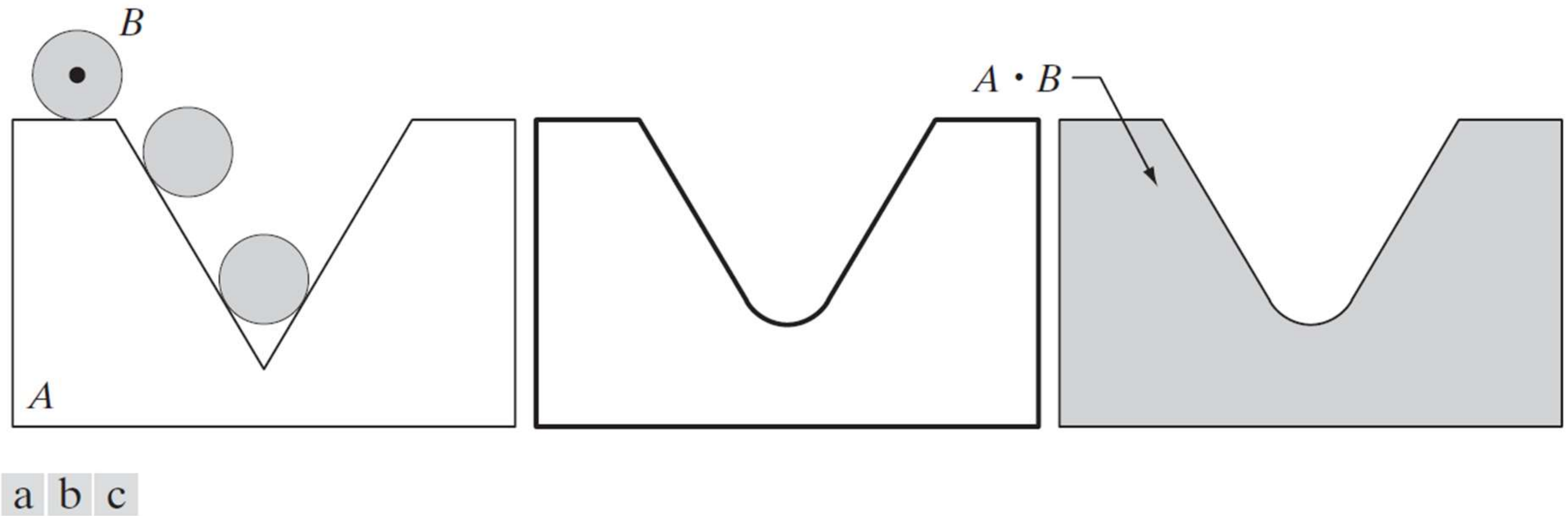
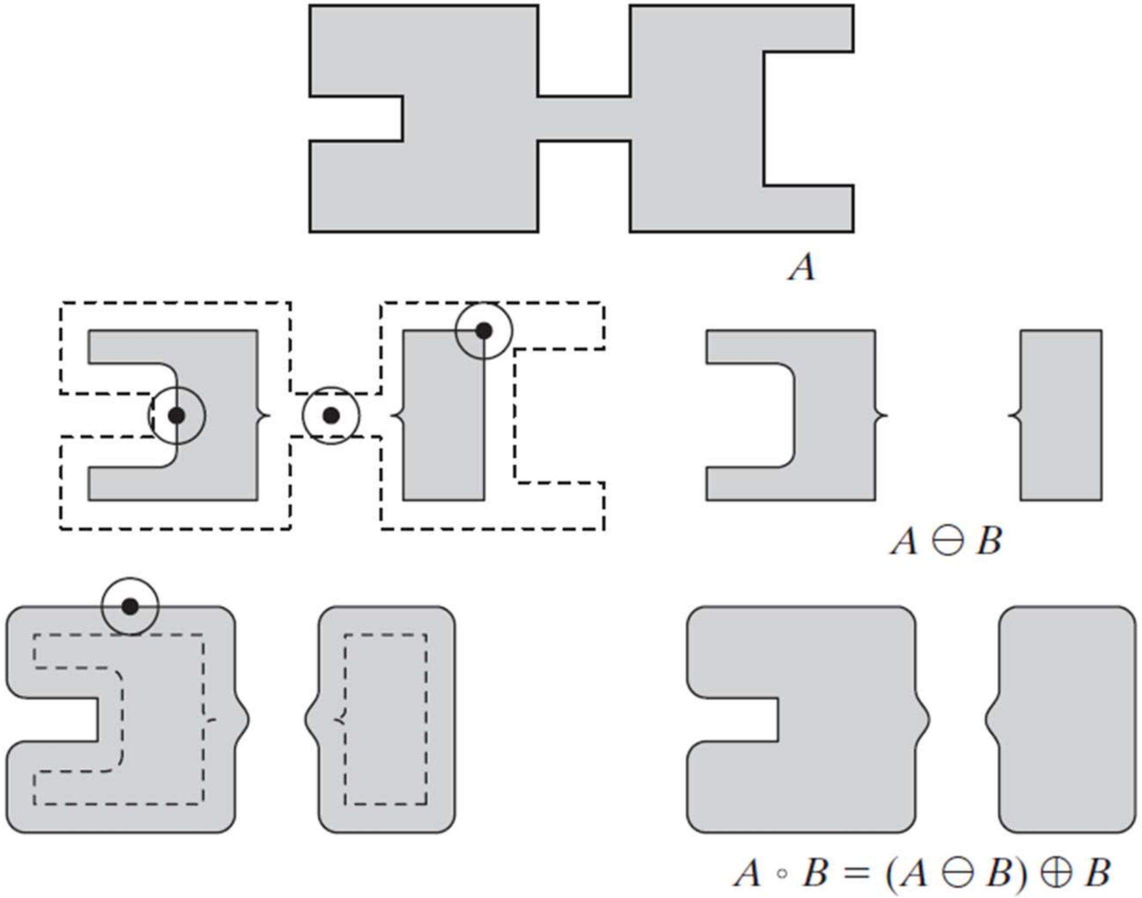


FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

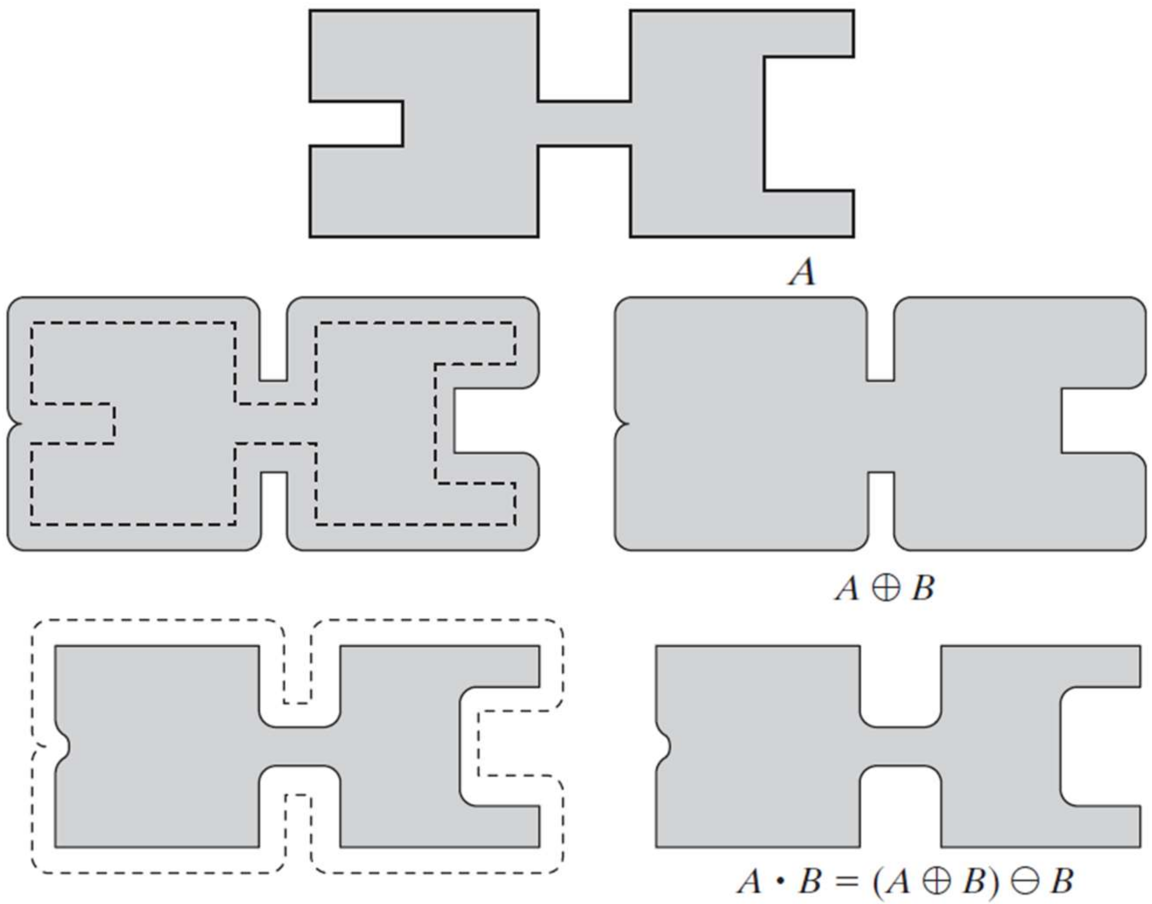
3. Opening and Closing

- Example



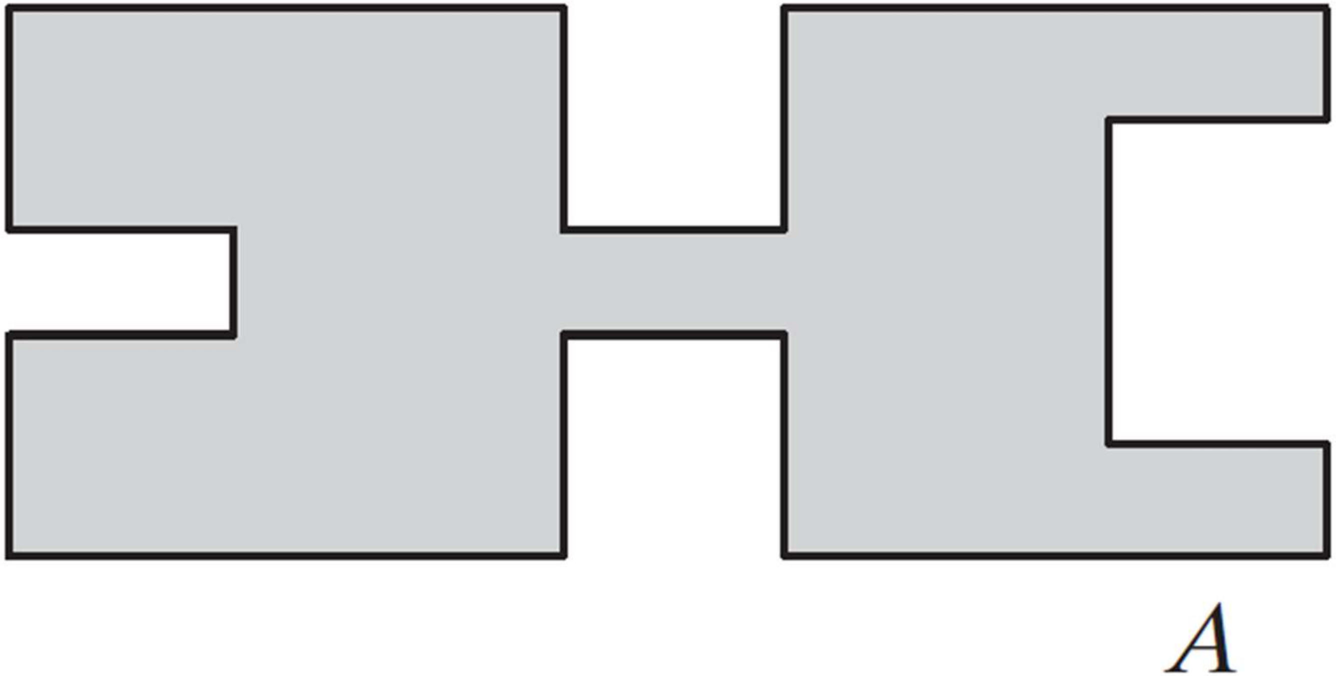
3. Opening and Closing

- Example



3. Opening and Closing

- MATLAB: s37OpenClose.m



3. Opening and Closing

- As in the case with dilation and erosion, **opening** and **closing** are **duals** of each other with respect to set complementation and reflection.

$$(A \bullet B)^c = (A^c \circ \hat{B}) \qquad (A \circ B)^c = (A^c \bullet \hat{B})$$

- The opening operation satisfies the following properties:

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

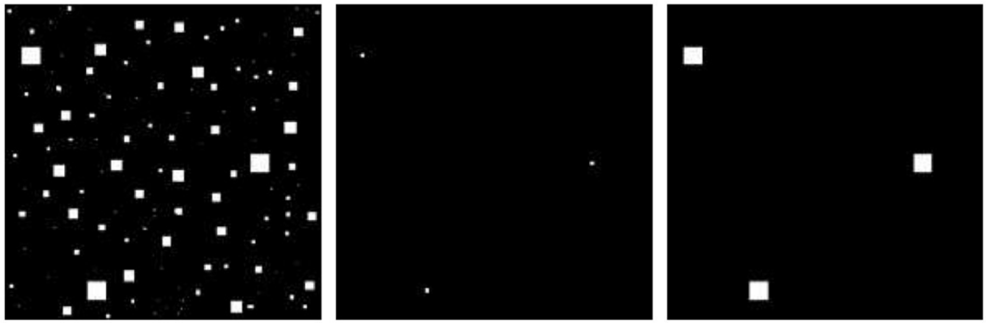
- Similarly, the closing operation satisfies the following properties:

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

3. Opening and Closing

- Exercise

Consider the three binary images shown in the following figure. The image on the left is composed of squares of sizes 1, 3, 5, 7, 9, and 15 pixels on the side. The image in the middle was generated by eroding the image on the left with a square structuring element of 1s, of size 13 × 13 pixels, with the objective of eliminating all the squares, except the largest ones. Finally, the image on the right is the result of dilating the image in the center with the same structuring element, with the objective of restoring the largest squares. You know that erosion followed by dilation is the opening of an image, and you know also that opening generally does not restore objects to their original form. Explain why full reconstruction of the large squares was possible in this case.



3. Opening and Closing

- Example

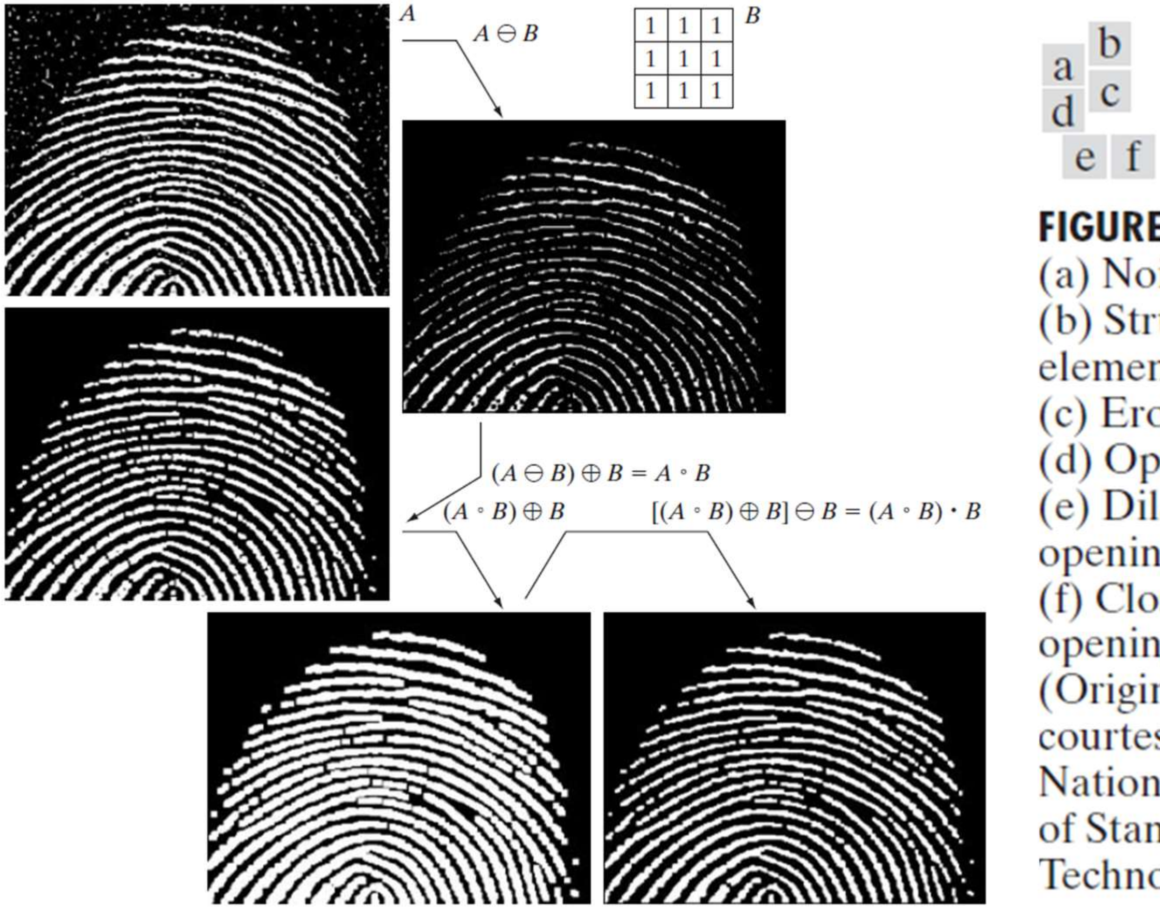


FIGURE 9.11 (a) Noisy image. (b) Structuring element. (c) Eroded image. (d) Opening of A. (e) Dilation of the opening. (f) Closing of the opening. (Original image courtesy of the National Institute of Standards and Technology.)

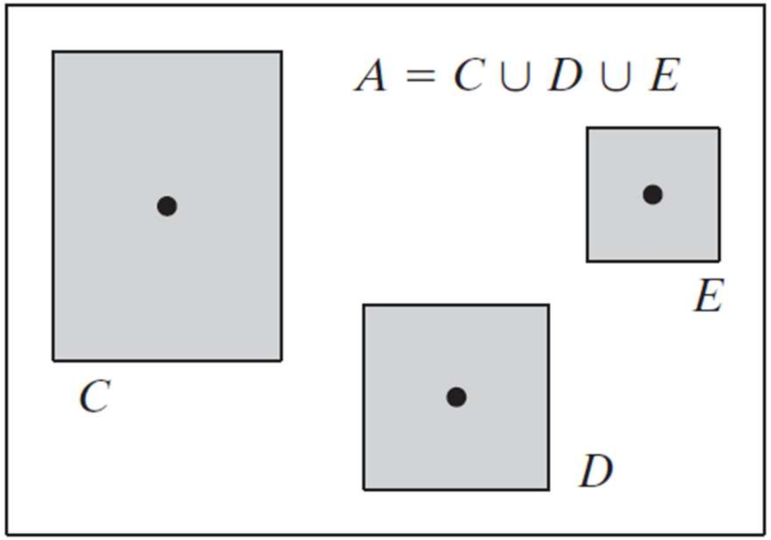
3. Opening and Closing

- MATLAB: s41Fingerprint.m

4. The Hit-or-Miss Transformation

• The morphological **hit-or-miss** transformation is a basic tool for **shape detection**.

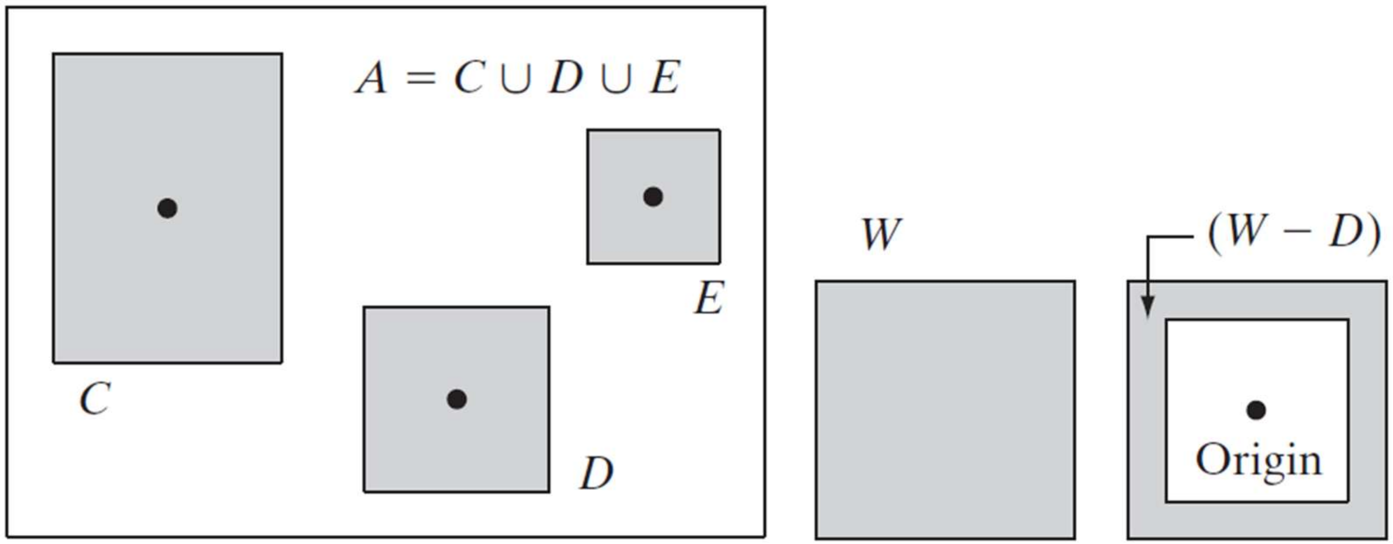
• The objective is to find the location of one of the shapes, say, D.



4. The Hit-or-Miss Transformation

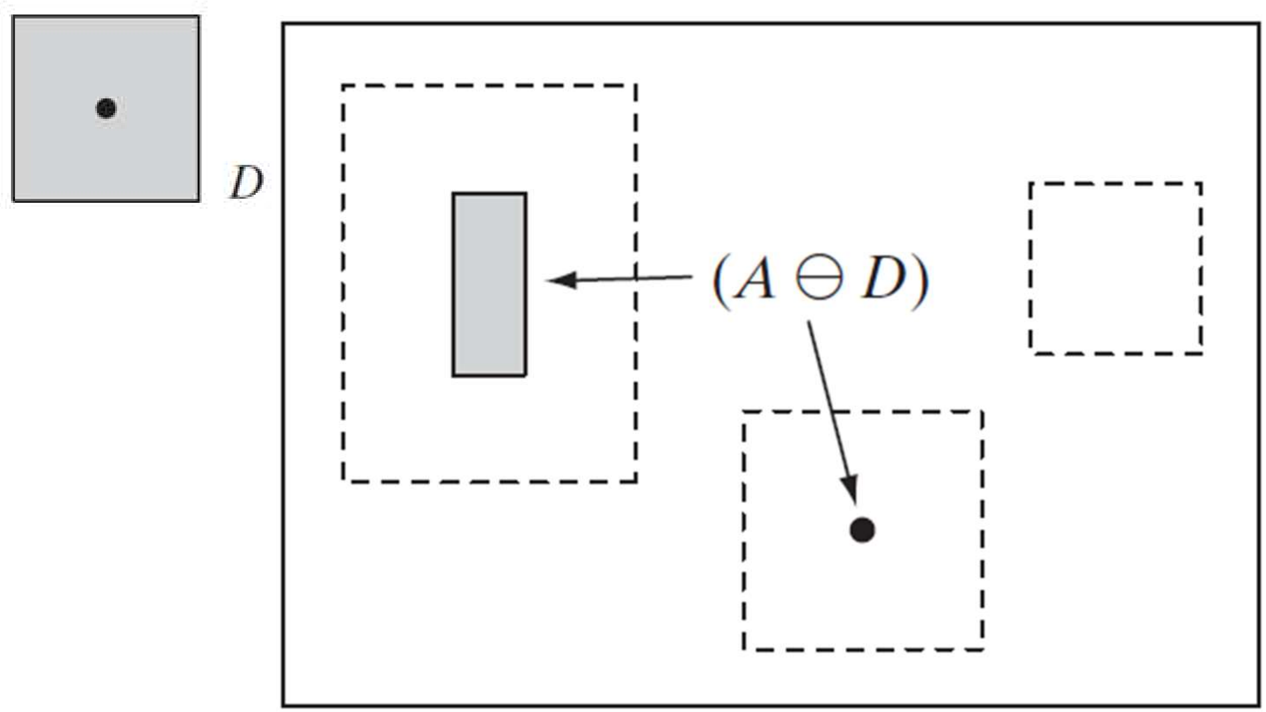
- Let D be enclosed by a small window, W.

- The local background of D with respect to W is defined as the set difference (W-D).



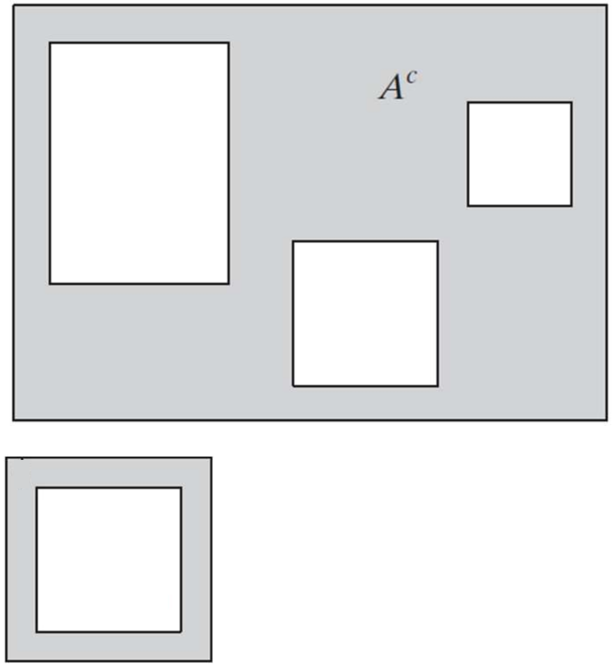
4. The Hit-or-Miss Transformation

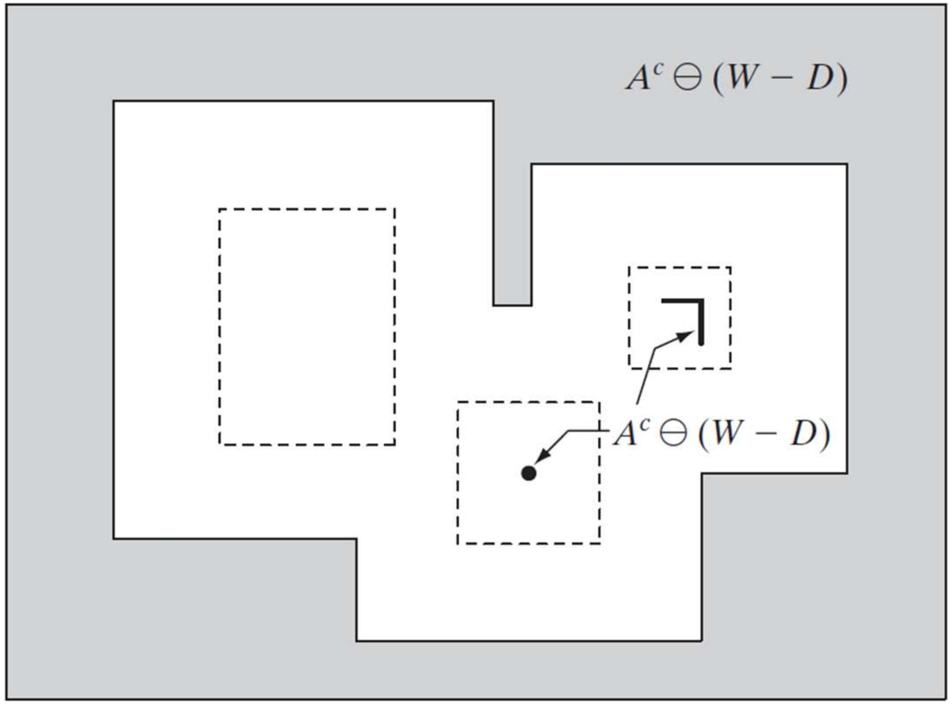
- Erosion of A by D (the dashed lines are included for reference).



4. The Hit-or-Miss Transformation

- Erosion of the complement of A by the local background set (W-D).

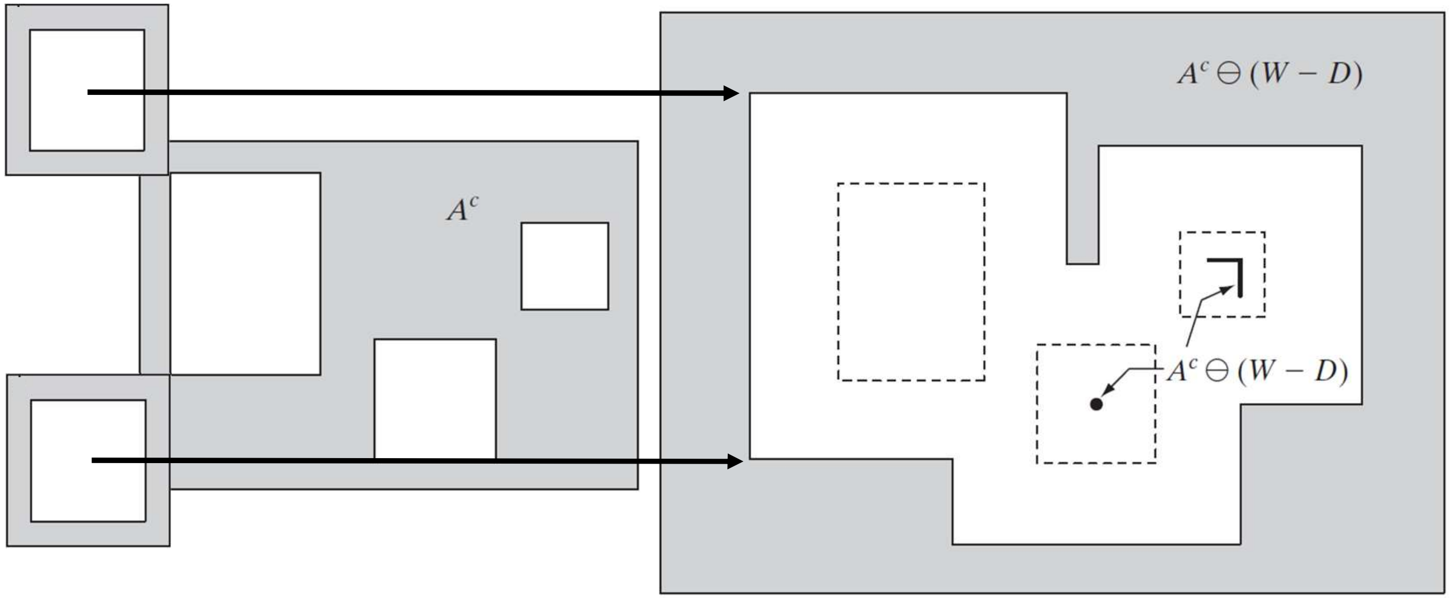




4. The Hit-or-Miss Transformation

- Erosion of the complement of A by the local background set (W-D).

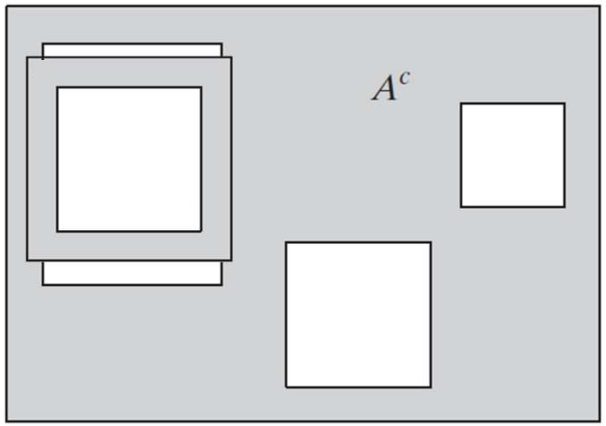
Grey regions match.

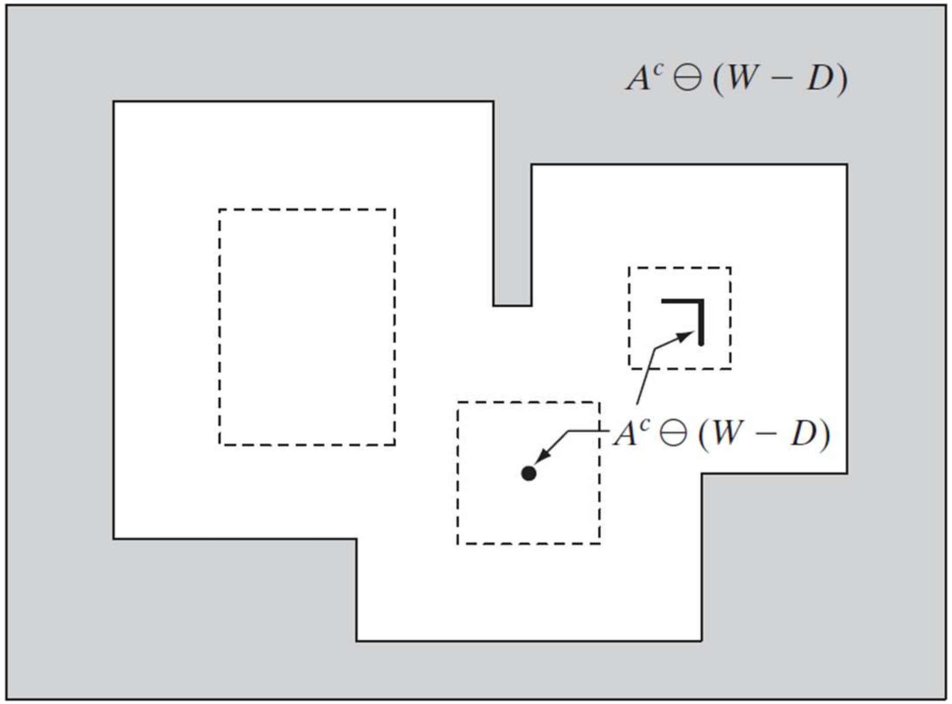


4. The Hit-or-Miss Transformation

- Erosion of the complement of A by the local background set (W-D).

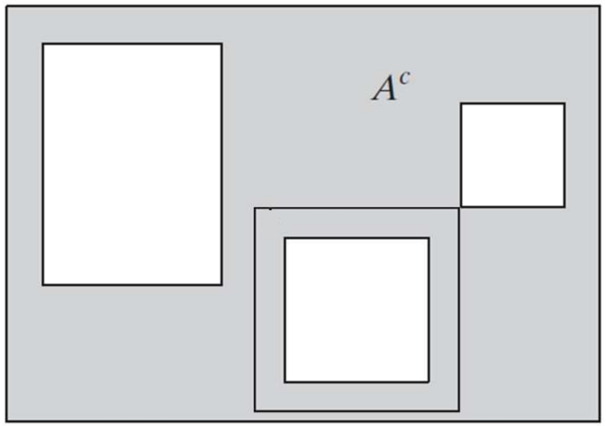
Grey regions do not match.



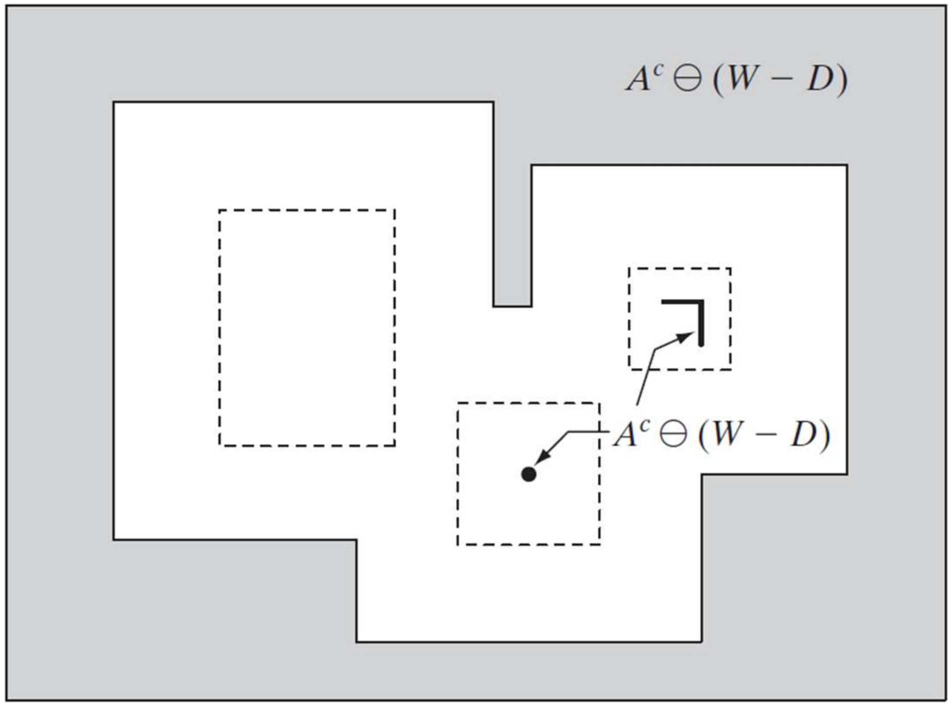


4. The Hit-or-Miss Transformation

• Erosion of the complement of A by the local background set (W-D).

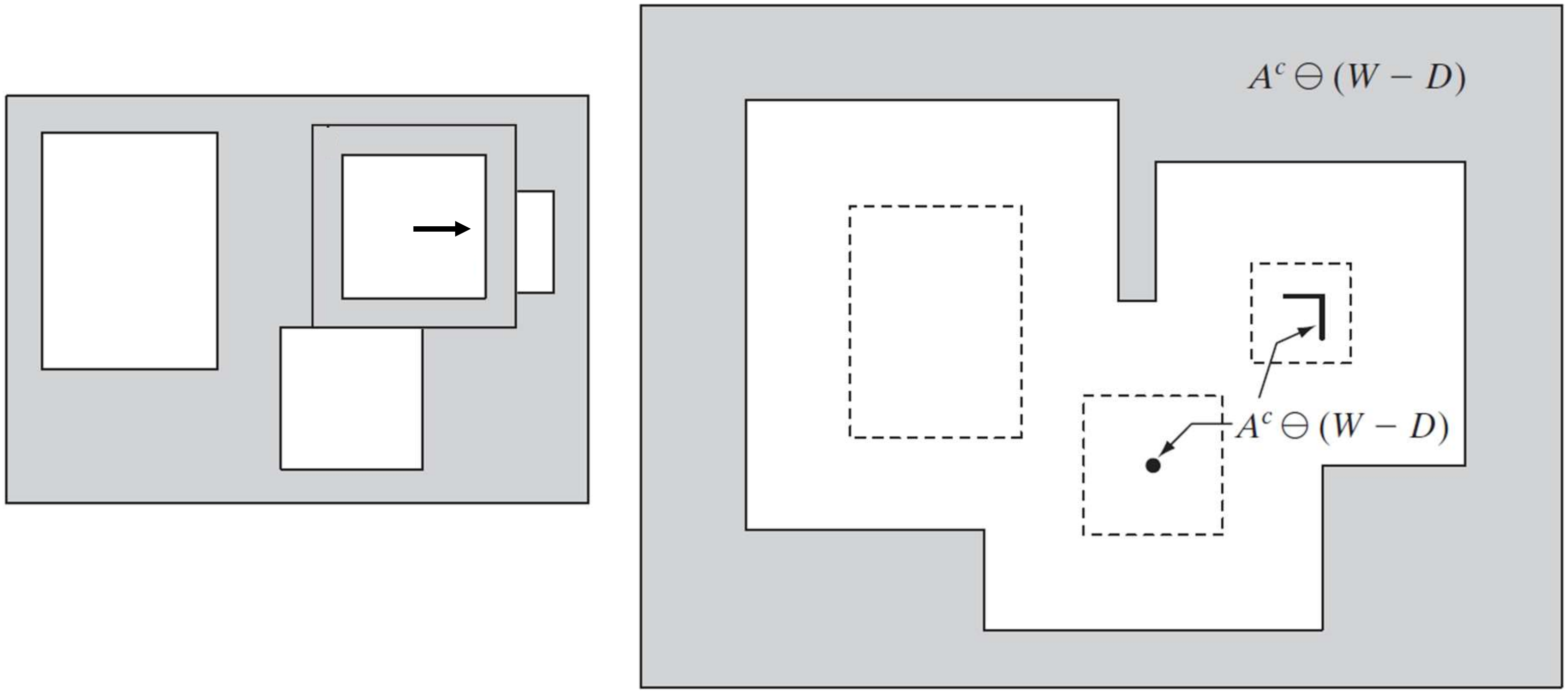


Grey regions match only in one single point.



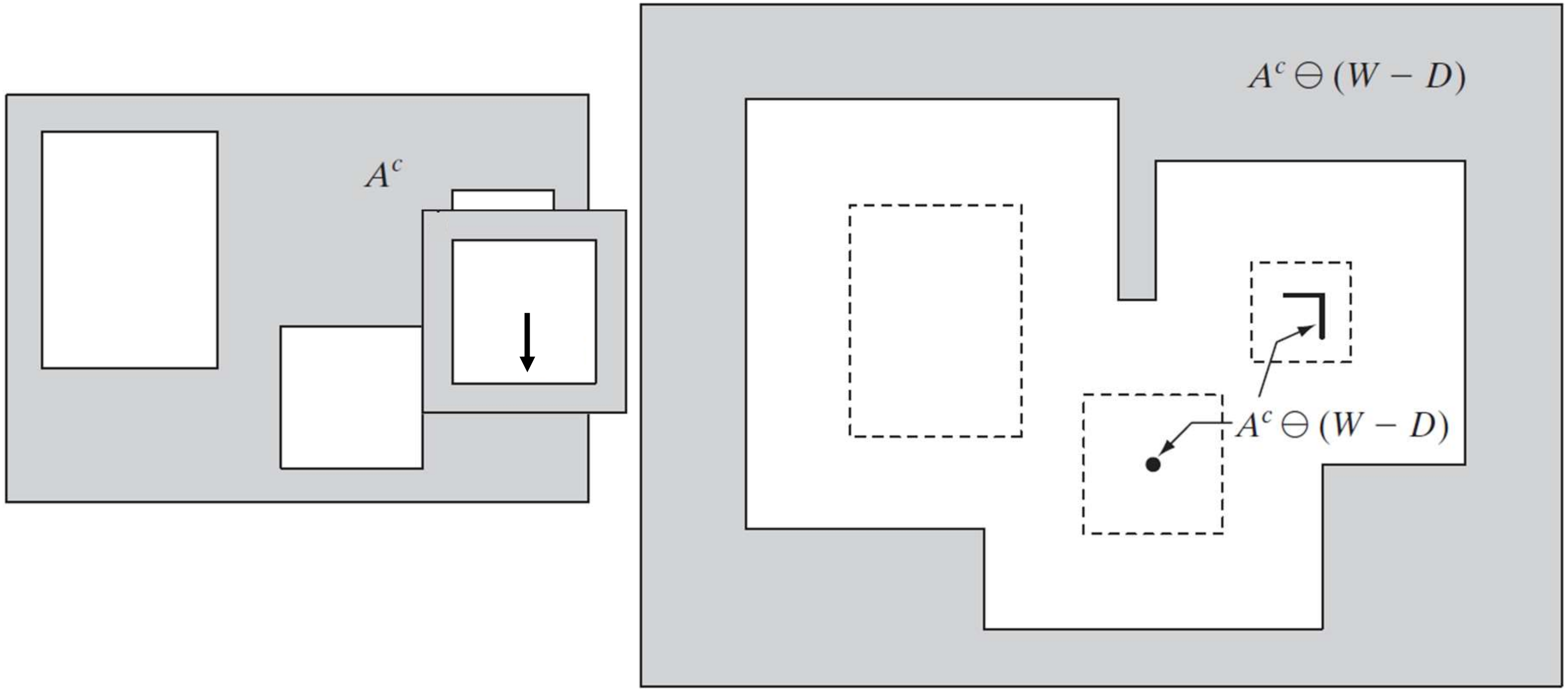
4. The Hit-or-Miss Transformation

• Erosion of the complement of A by the local background set (W-D).



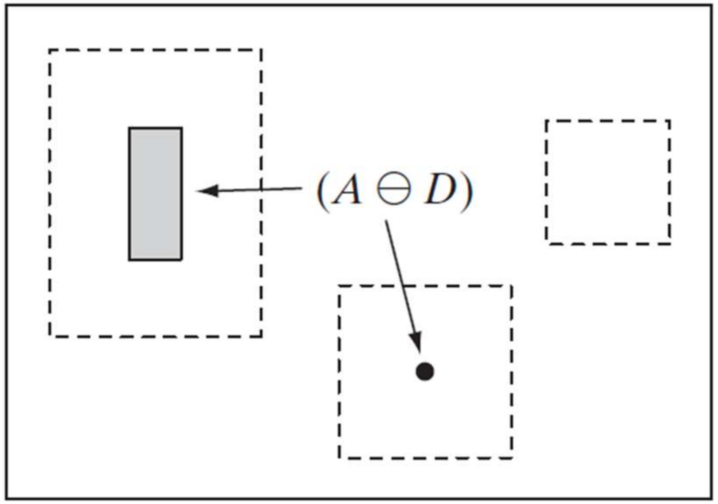
4. The Hit-or-Miss Transformation

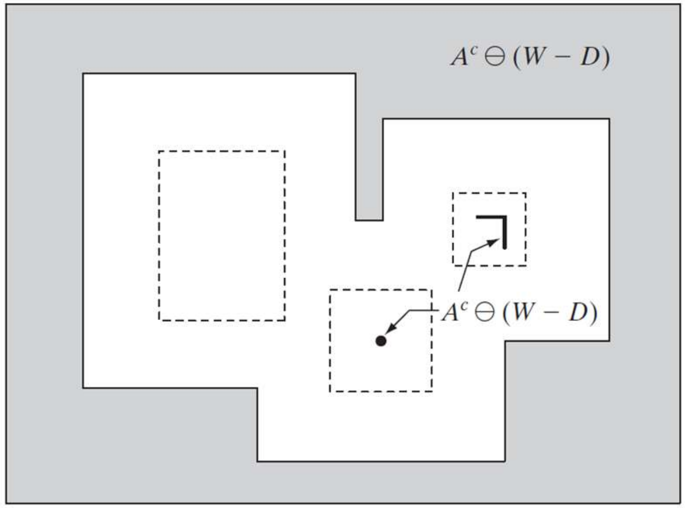
- Erosion of the complement of A by the local background set (W-D).



4. The Hit-or-Miss Transformation

- The intersection is precisely the location sought.





$$A \circledast B = (A \ominus D) \cap \left[A^c \ominus (W - D)\right] \qquad B = (B_1, B_2)$$

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

$$B_1 = D \qquad B_2 = (W - D)$$

4. The Hit-or-Miss Transformation

• Using a structuring element B_1 associated with objects and an element B_2 associated with the background is based on the assumption that **objects must be disjoint sets**.

> ➢ Each object has at least a **one-pixel-thick background** around it.

• Sometimes, we may want to detect certain patterns of 1s and 0s within a set, in which case a **background is not required**.

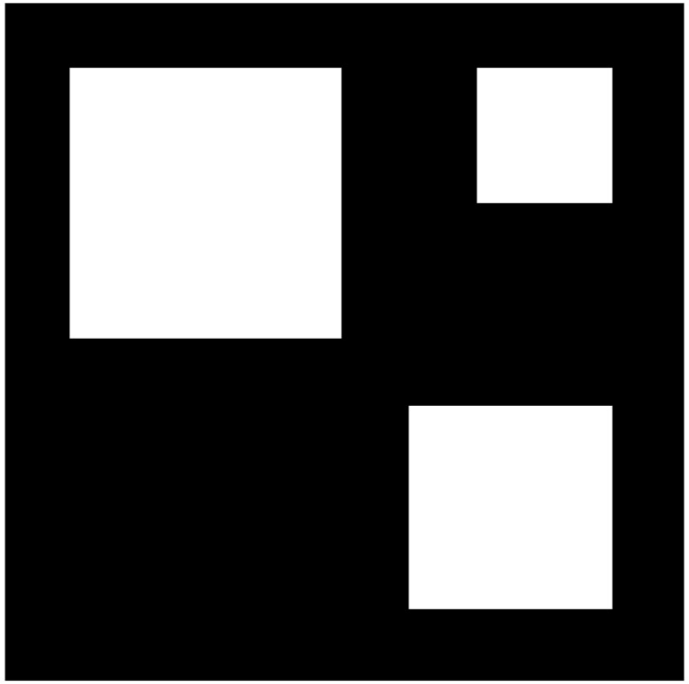
• In such instances, the **hit-or-miss transformation** is reduced to a **simple erosion**.

> ➢ It is still a set of matches, but without the requirement of a background match for detecting individual objects.

• This simplified pattern detection scheme is used in some of the following algorithms.

4. The Hit-or-Miss Transformation

- MATLAB: s52HitMiss.m



5. Boundary Extraction

- The boundary of a set A is defined as

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

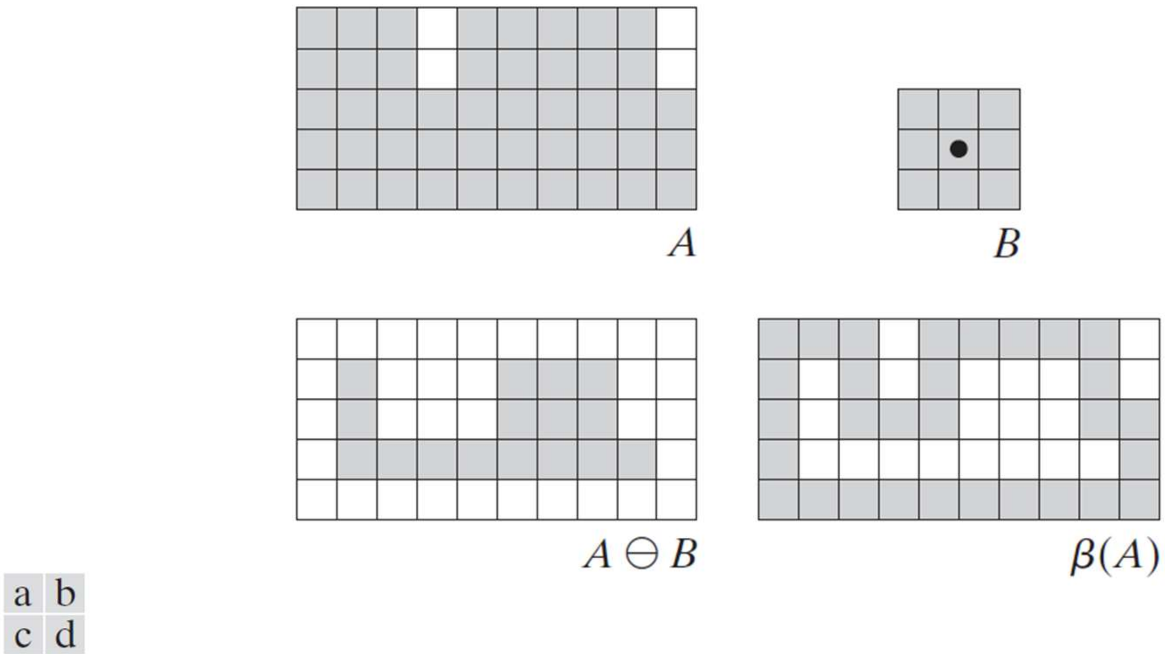
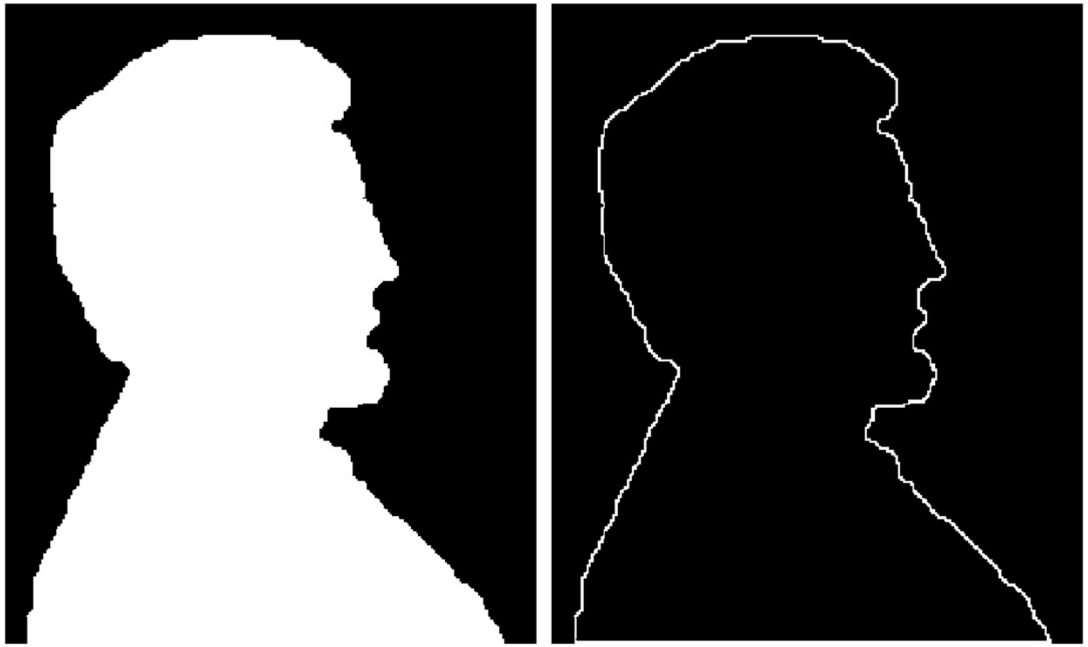


FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

5. Boundary Extraction

- MATLAB: s55Boundary.m



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).

6. Hole Filling

• A **hole** may be defined as a background region surrounded by a connected border of foreground pixels.

• Let A denote a set whose elements are 8-connected boundaries, each boundary enclosing a background region (i.e., a hole).

• Given a point in each hole, the objective is to fill all the holes with 1s.

• First form an array, X_0, of 0s (the same size as the array containing A), except at the locations in X_0 corresponding to the given point in each hole, which we set to 1.

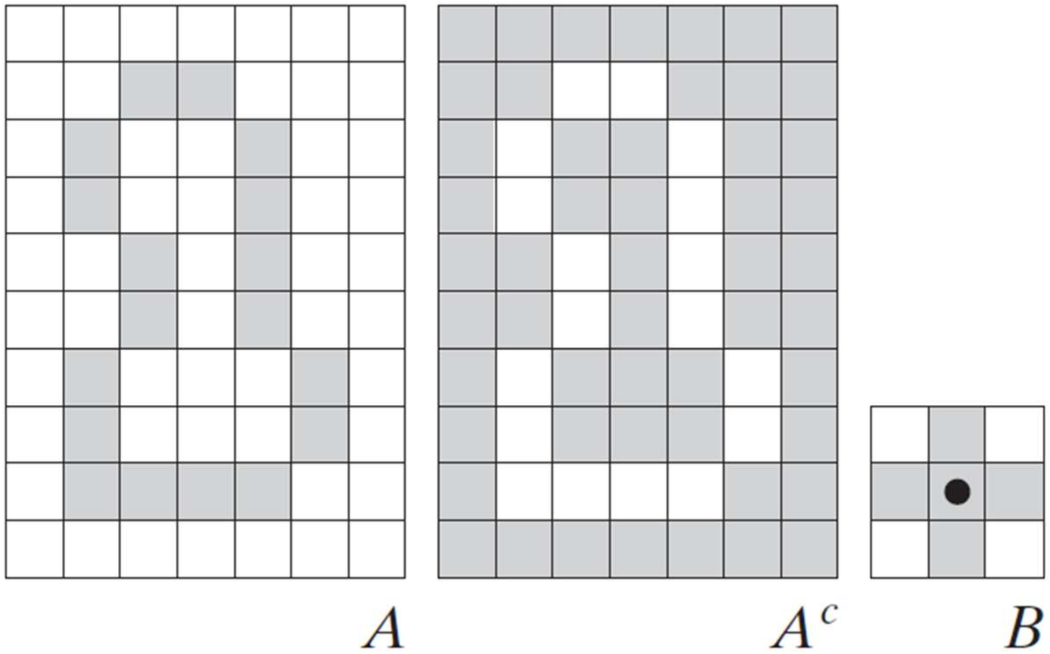
• Then, we apply the following procedure,

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

6. Hole Filling

- Then, we apply the following procedure,

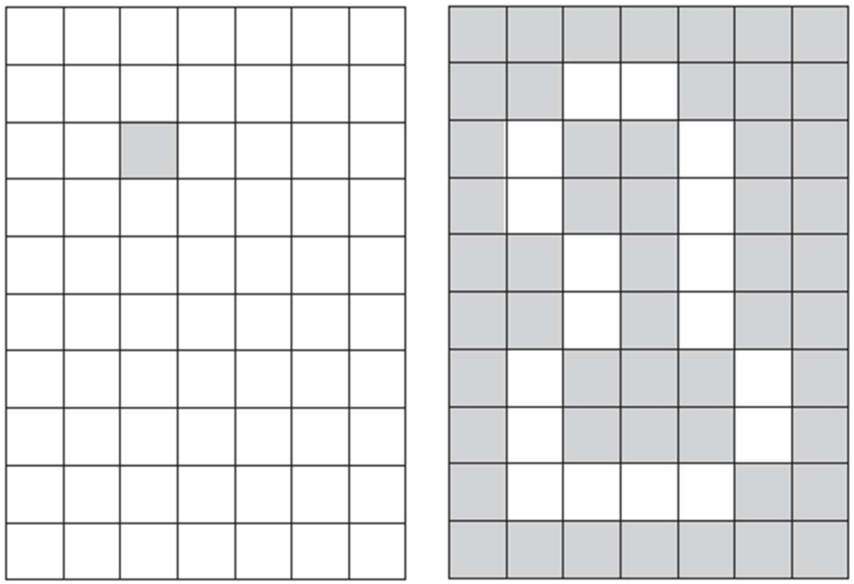
$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1, 2, 3, \ldots$$



6. Hole Filling

- Then, we apply the following procedure,

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

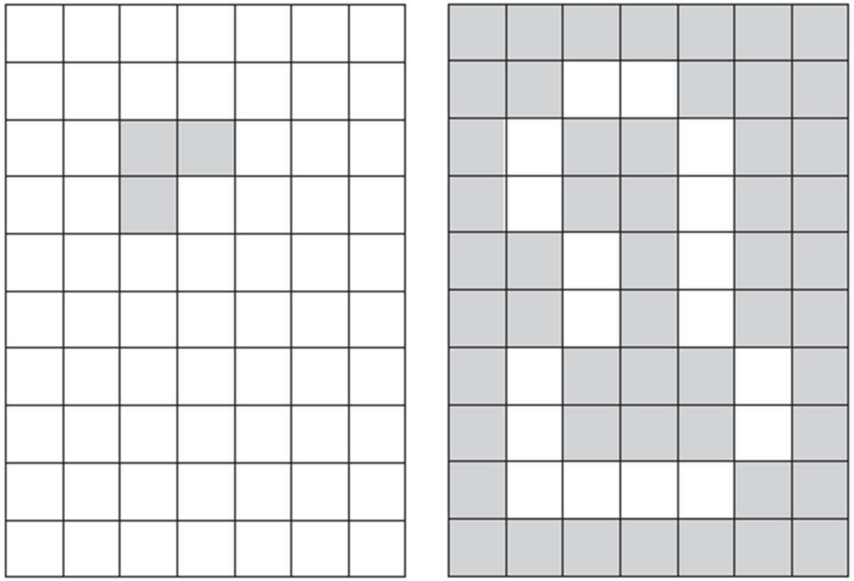


X_0

6. Hole Filling

- Then, we apply the following procedure,

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

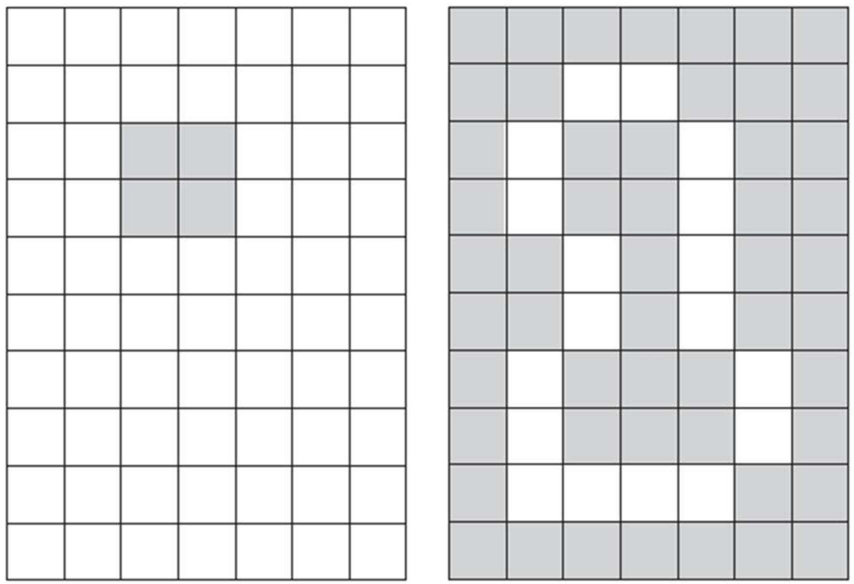


X_1

6. Hole Filling

- Then, we apply the following procedure,

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

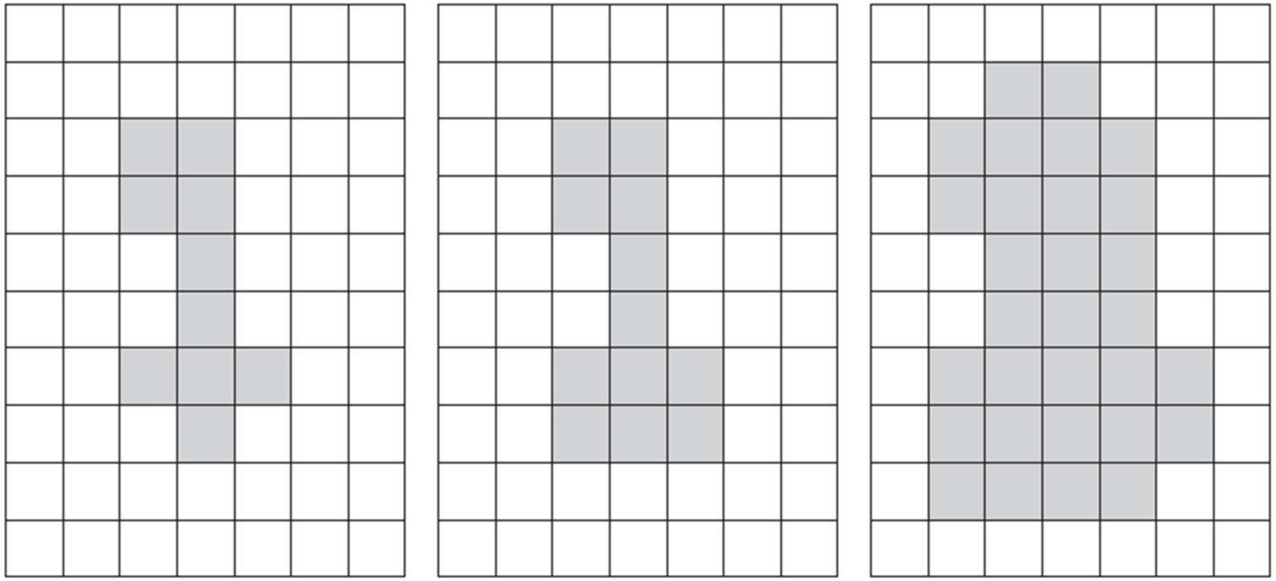


X_2

6. Hole Filling

- Then, we apply the following procedure,

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

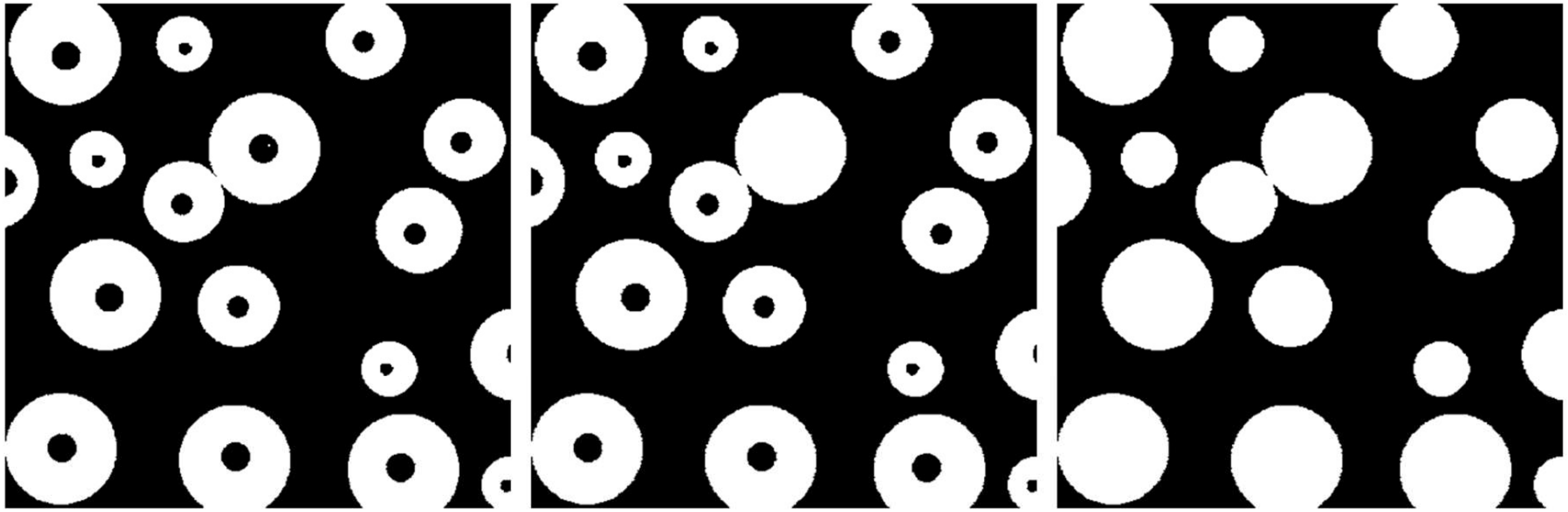


X_6 $\quad$ X_8 $\quad$ $X_8 \cup A$

6. Hole Filling

- Example/ Exercise: Fig0916(a)(region-filling-reflections).tif

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



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.

7. Extraction of Connected Components

- Extraction of connected components from a binary image is central to many automated image analysis applications.

- Let A be a set containing one or more connected components.

- Form an array, X_0, whose elements are 0s, except at each location known to correspond to a point in each connected component in A, which we set to 1.

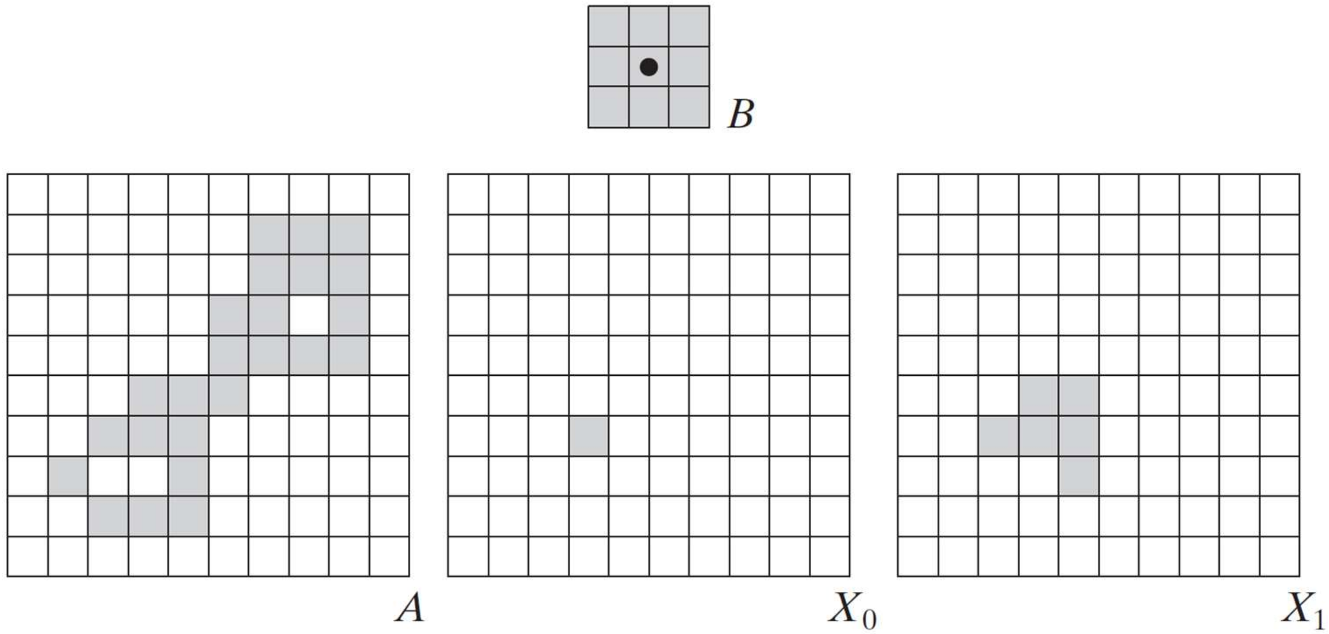
- Then, we apply the following procedure,

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

7. Extraction of Connected Components

- Then, we apply the following procedure,

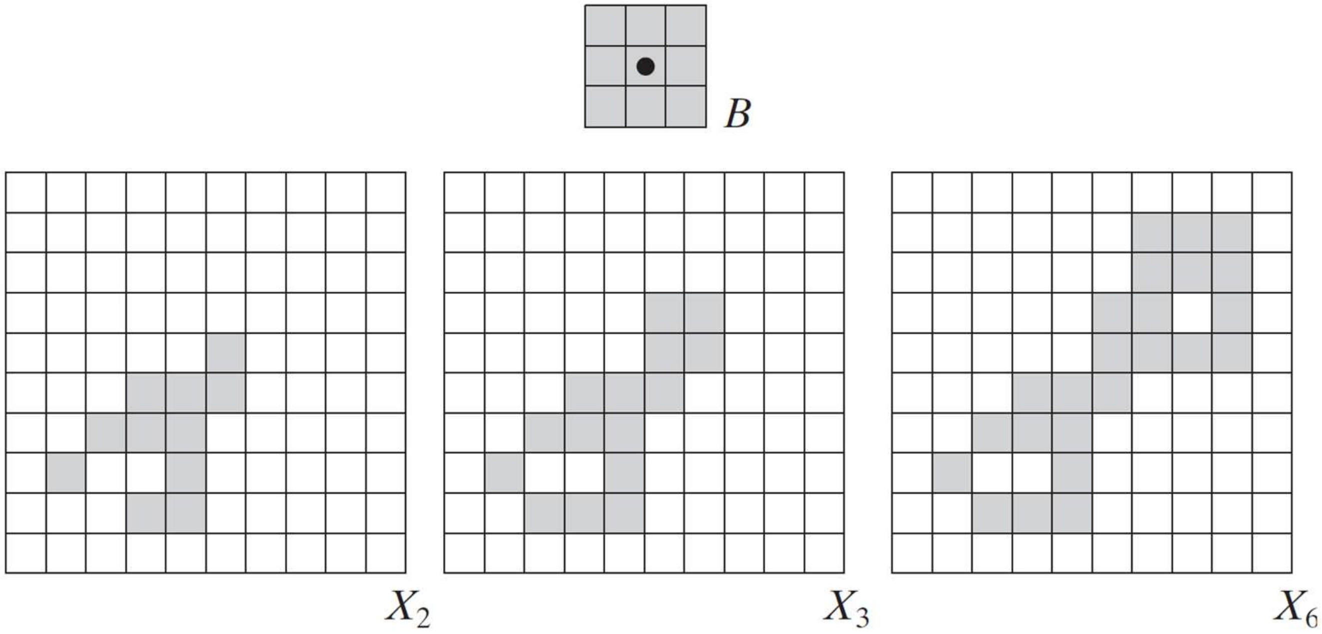
$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots$$



7. Extraction of Connected Components

- Then, we apply the following procedure,

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

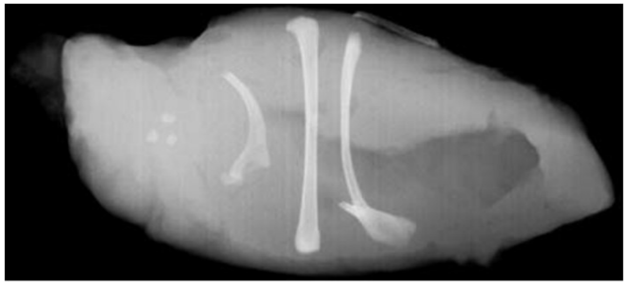


7. Extraction of Connected Components

- Example/Exercise: X-ray image of a chicken breast that contains bone fragments.

- It is of considerable interest to be able to detect such objects in processed food before packaging and/or shipping.

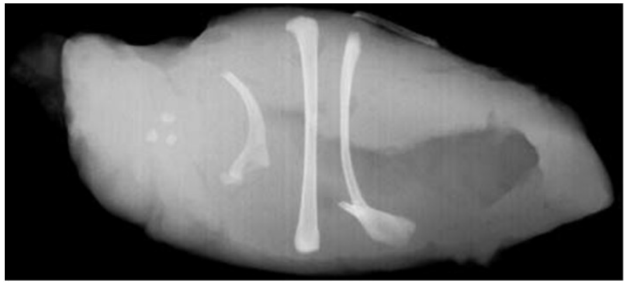
- Fig0918(a)(Chickenfilet with bones).tif



7. Extraction of Connected Components

• In this particular case, the density of the bones is such that their nominal intensity values are different from the background.

• This makes extraction of the bones from the background a simple matter by using a single threshold.



7. Extraction of Connected Components

- We use erosion to make sure that only objects of "significant" size remain.

- In this example, we define as significant any object that remains after erosion with a 5x5 structuring element of 1s

7. Extraction of Connected Components

- We use erosion to make sure that only objects of "significant" size remain.

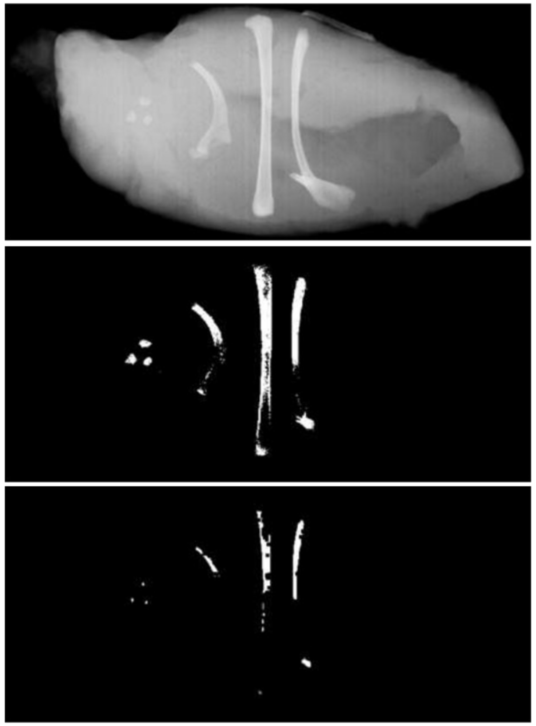
- In this example, we define as significant any object that remains after erosion with a 5x5 structuring element of 1s.

7. Extraction of Connected Components

• The next step is to analyze the size of the objects that remain. We label these objects by extracting the connected components in the image.

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

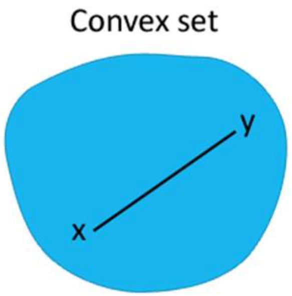
8. Convex Hull

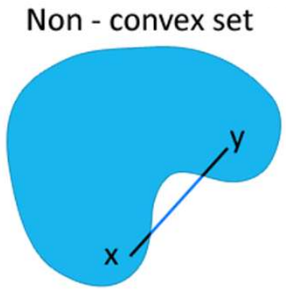
- A set A is said to be convex if the straight line segment joining any two points in A lies entirely within A

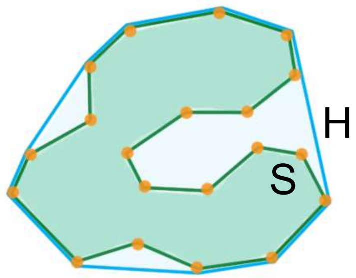
- The *convex hull* H of an arbitrary set S is the smallest convex set containing S*.*

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

- The convex hull and convex deficiency are useful for object description



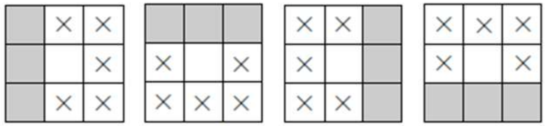




https://www.easycalculation.com/maths-dictionary/convex_set.html

8. Convex Hull

- Let B^i, $i = 1..4$ represent the following four structuring elements



B^1 B^2 B^3 B^4

- The procedure consists of implementing the equation

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \quad \text{and} \quad k = 1, 2, 3, \ldots$$

$$X_0^i = A$$

- When the procedure converges, we let

$$D^i = X_k^i$$

- Then the convex hull of A is

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

8. Convex Hull

• In other words, the method consists of iteratively applying the hit-or-miss transform to A with B^i.

• The union of the four resulting *Ds* constitutes the convex hull.

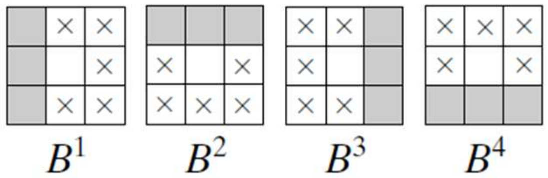
• We are using the simplified implementation of the hit-or-miss transform in which no background match is required, as discussed before.

• The x entries indicate “don’t care” conditions.

• A structuring element is said to have found a match in A if the 3x3 region of A under the structuring element mask at that location matches the pattern of the mask.

8. Convex Hull

- In other words, for a particular mask, a pattern match occurs when the center of the 3 x 3 region in A is 0, and the three pixels under the shaded mask elements are 1.

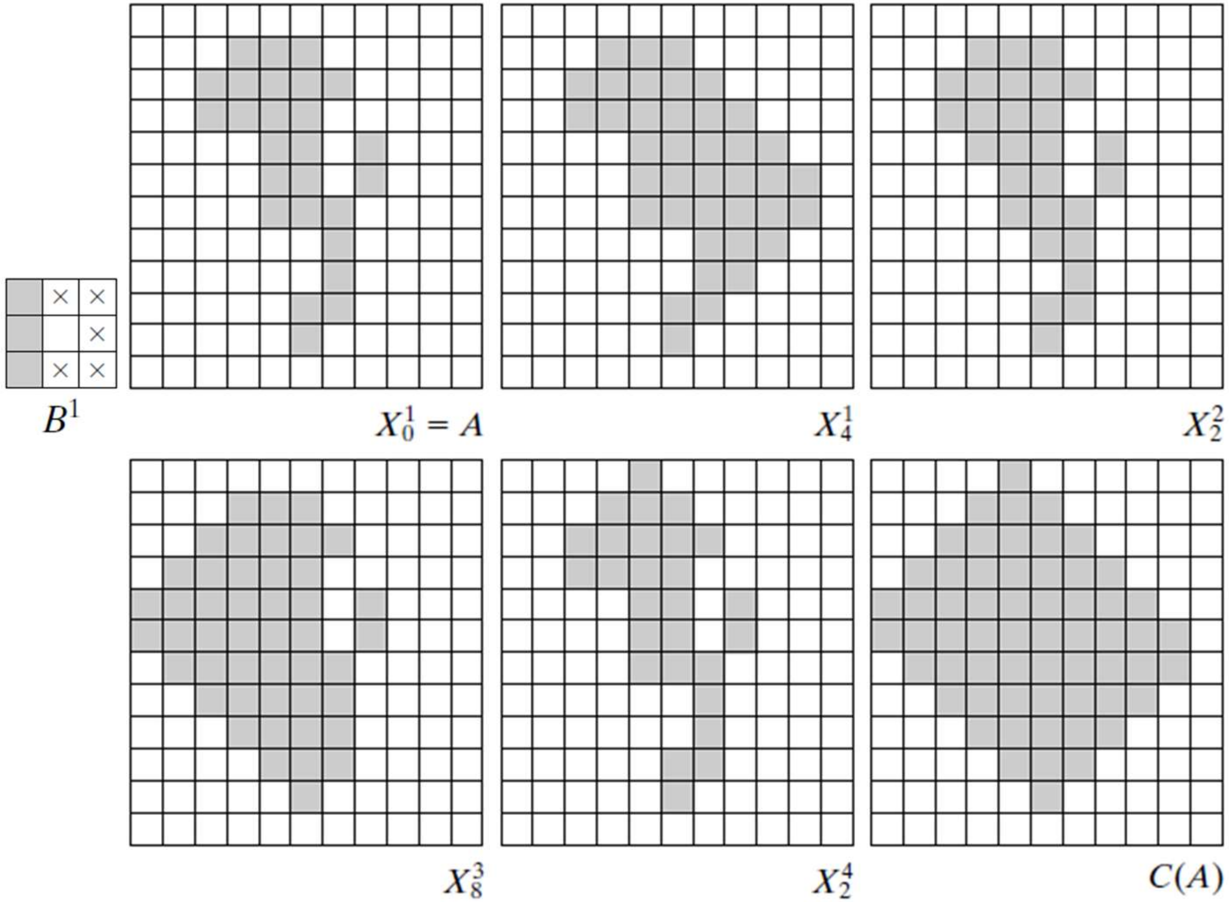


- The values of the other pixels in the region do not matter.

- Observe that B^i is a clockwise rotation of B^{i-1} by 90º.

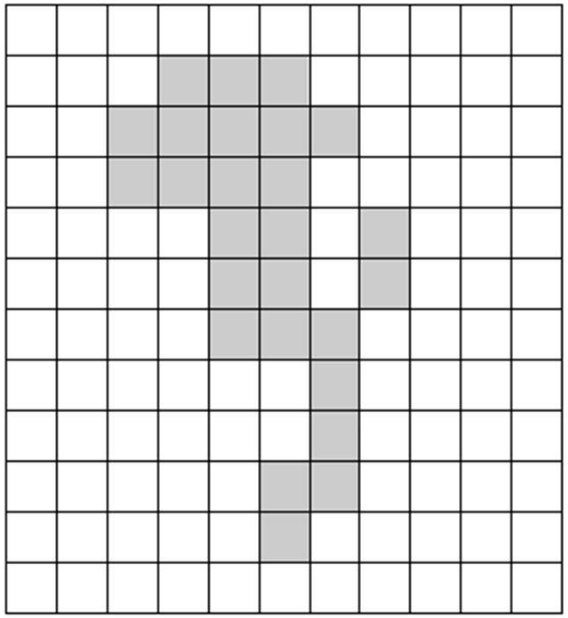
8. Convex Hull

- Example



8. Convex Hull

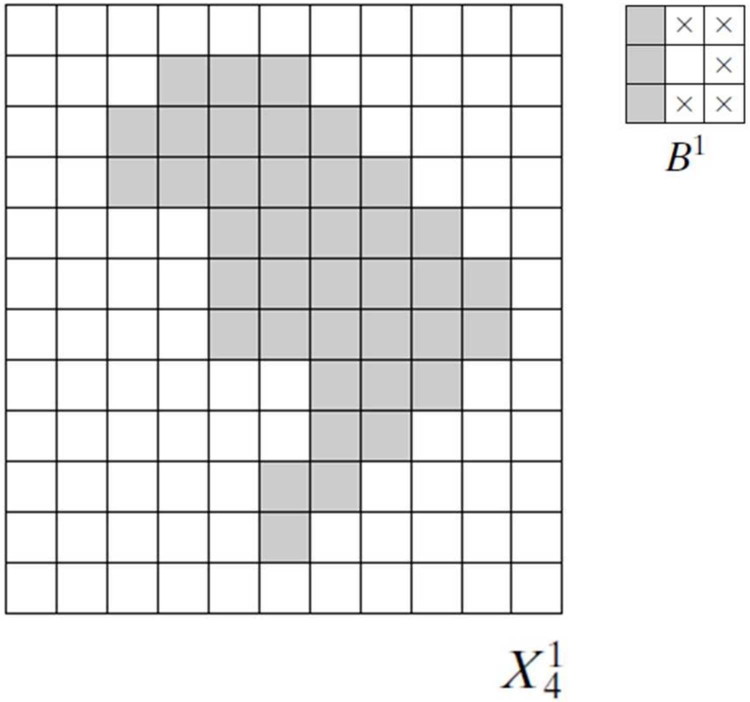
- Example



A

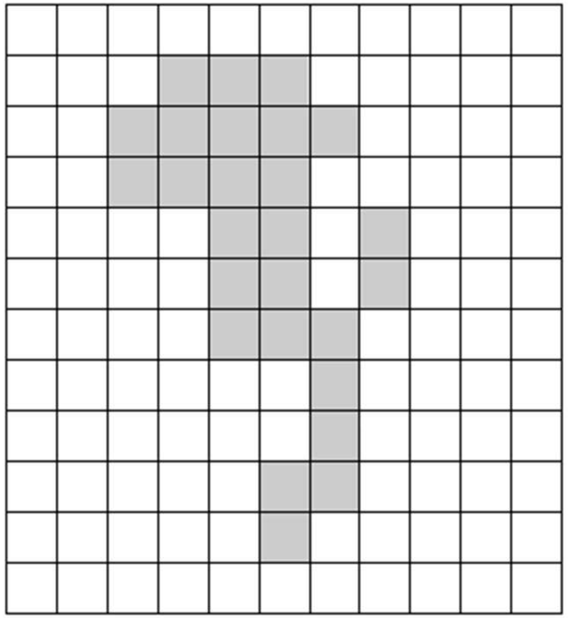
8. Convex Hull

- Example



8. Convex Hull

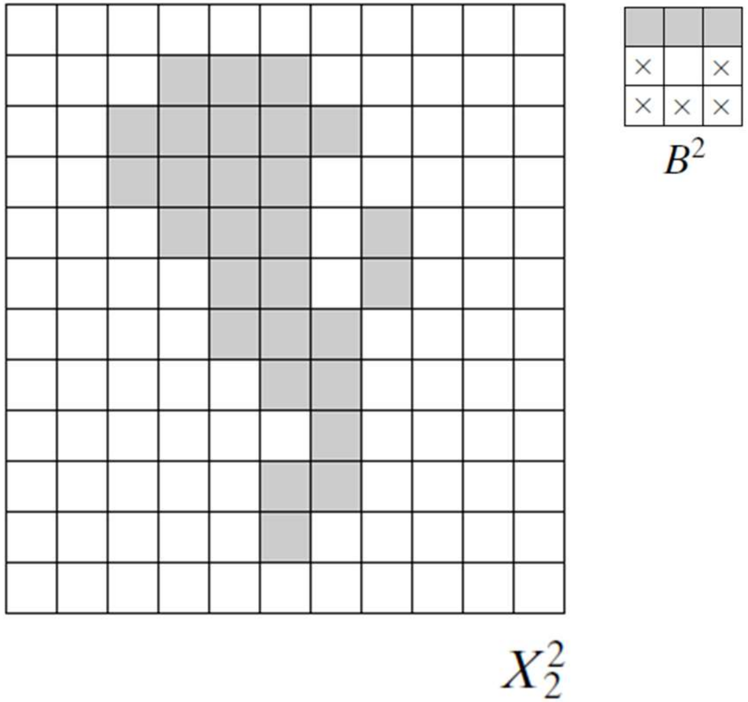
- Example



A

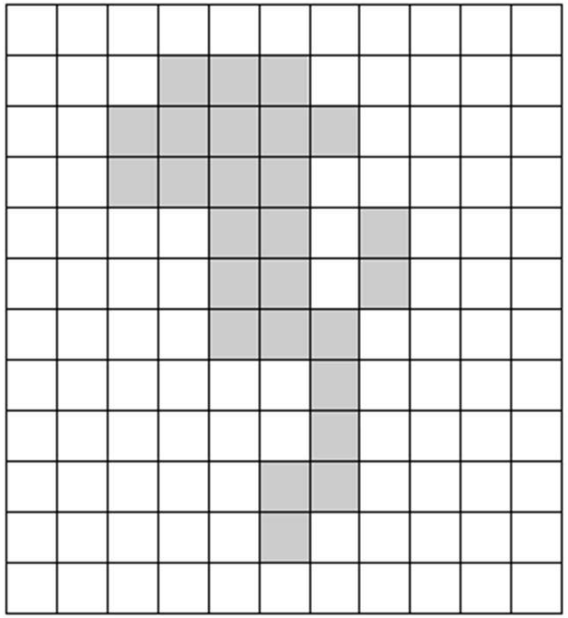
8. Convex Hull

- Example



8. Convex Hull

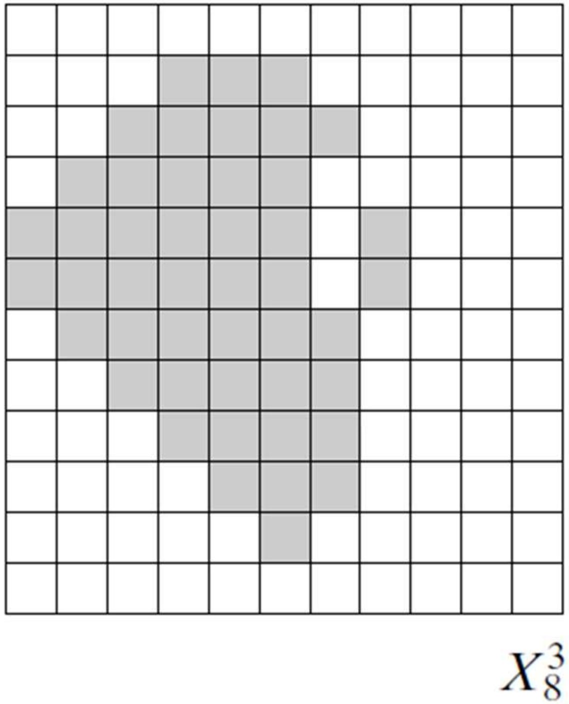
- Example

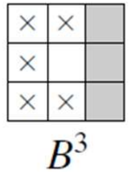


A

8. Convex Hull

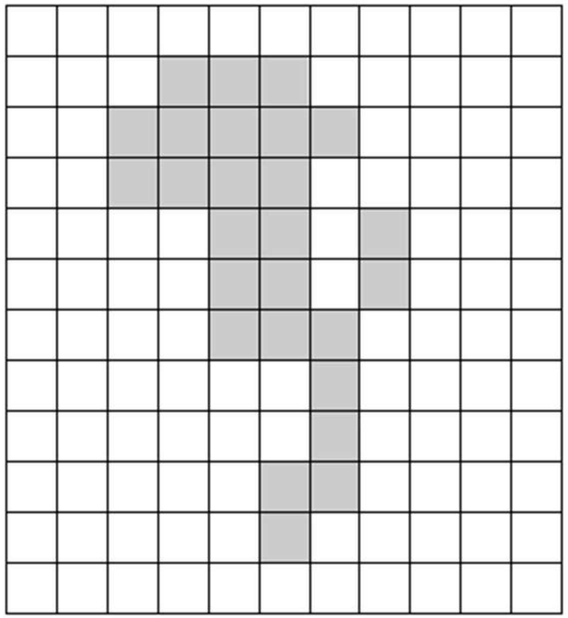
- Example





8. Convex Hull

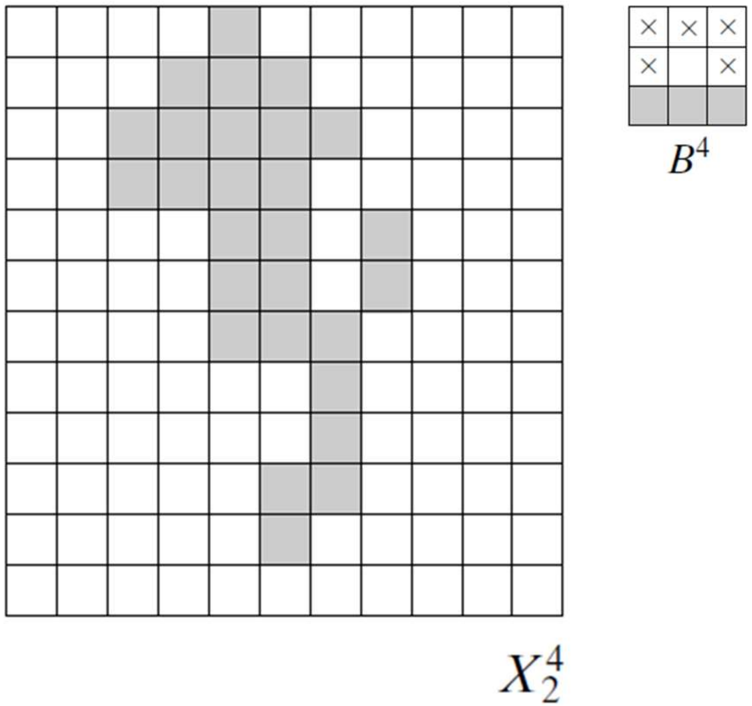
- Example



A

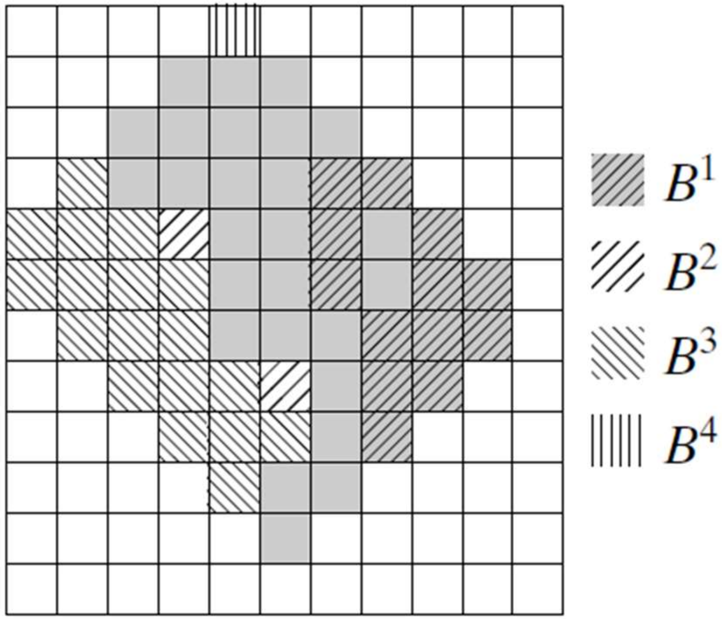
8. Convex Hull

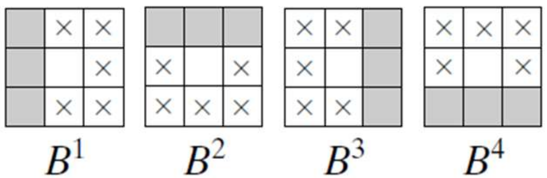
- Example



8. Convex Hull

- Example





8. Convex Hull

- Example

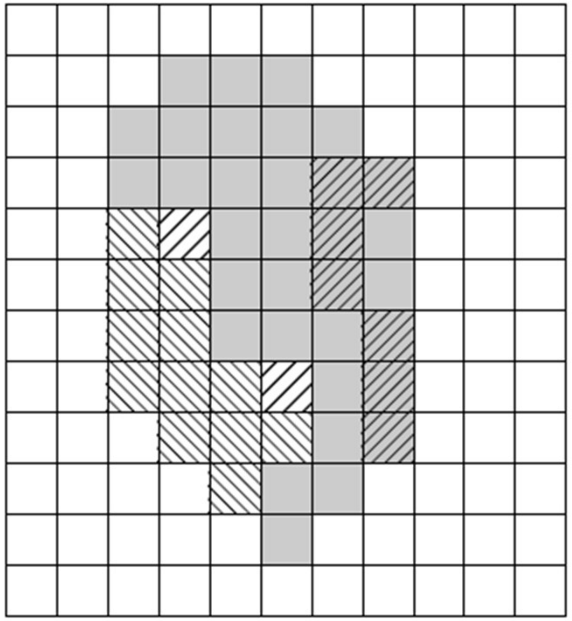


FIGURE 9.20 Result of limiting growth of the convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

9. Thinning

- The **thinning** of a set A by a structuring B can be defined in terms of the hit-or-miss transform

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

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

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

- The entire process below is repeated until no further changes occur.

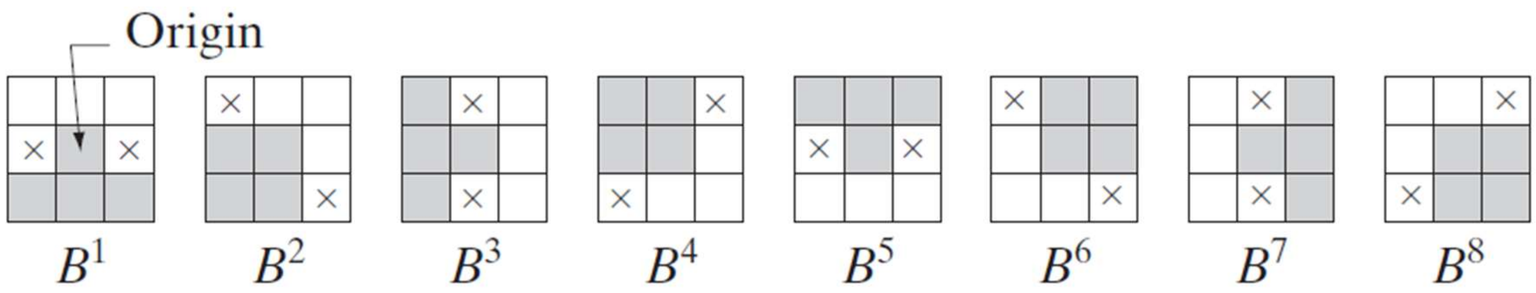
$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$$

9. Thinning

- The thinning of a set A by a structuring B can be defined in terms of the hit-or-miss transform:

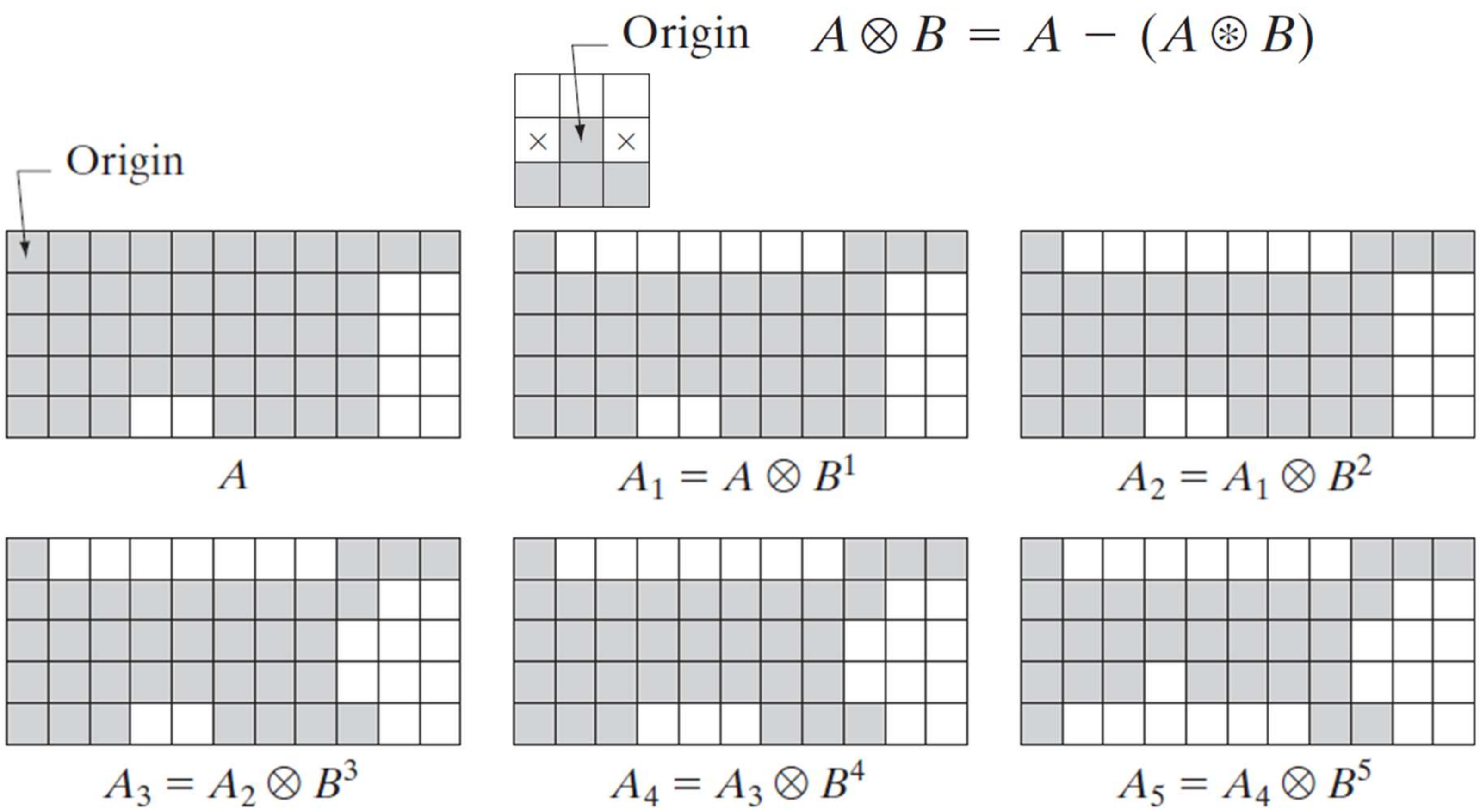
$$\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$$

- B^i is a clockwise rotation of B^{i-1}.



9. Thinning

- Example



9. Thinning

- Example

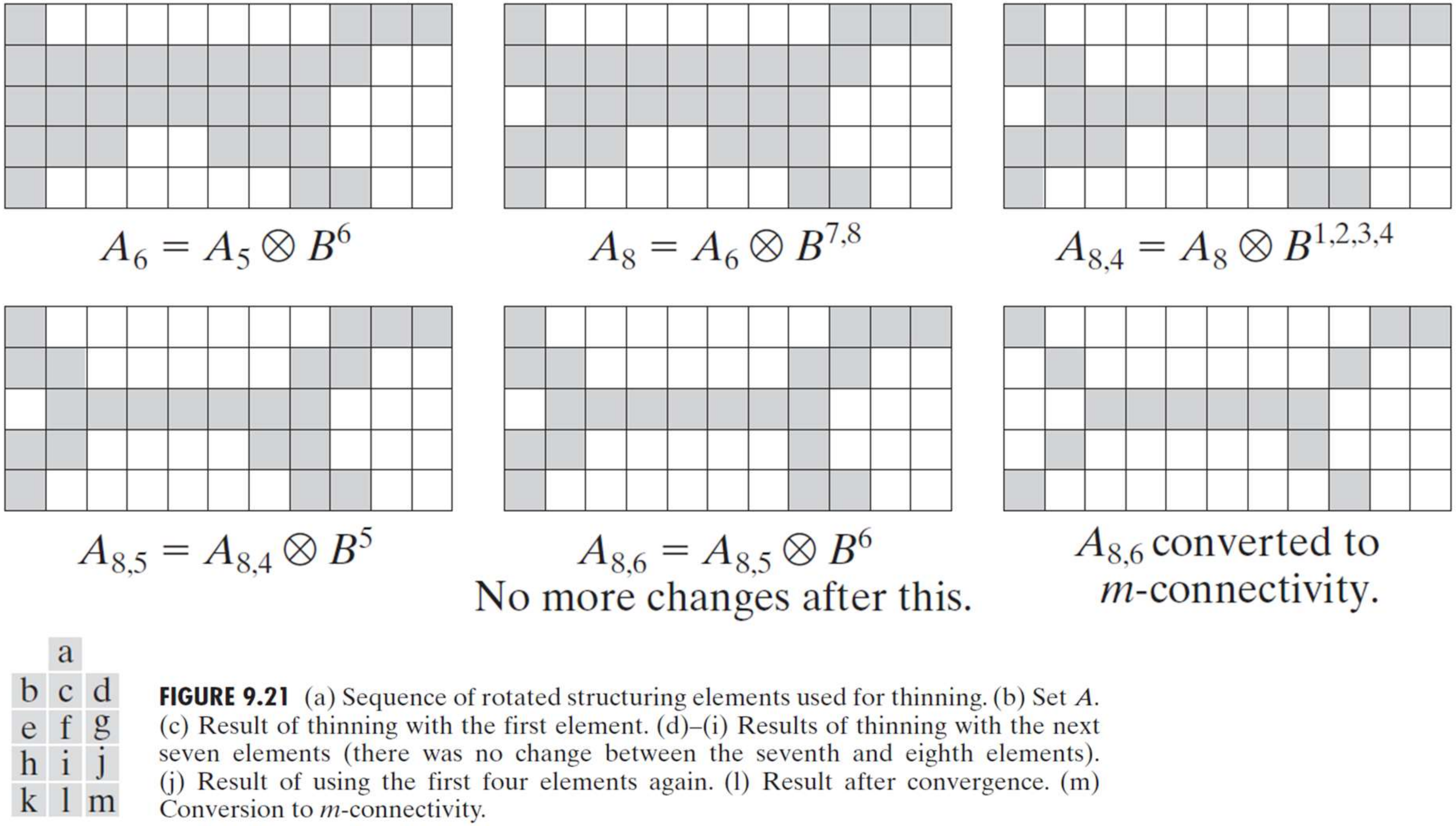
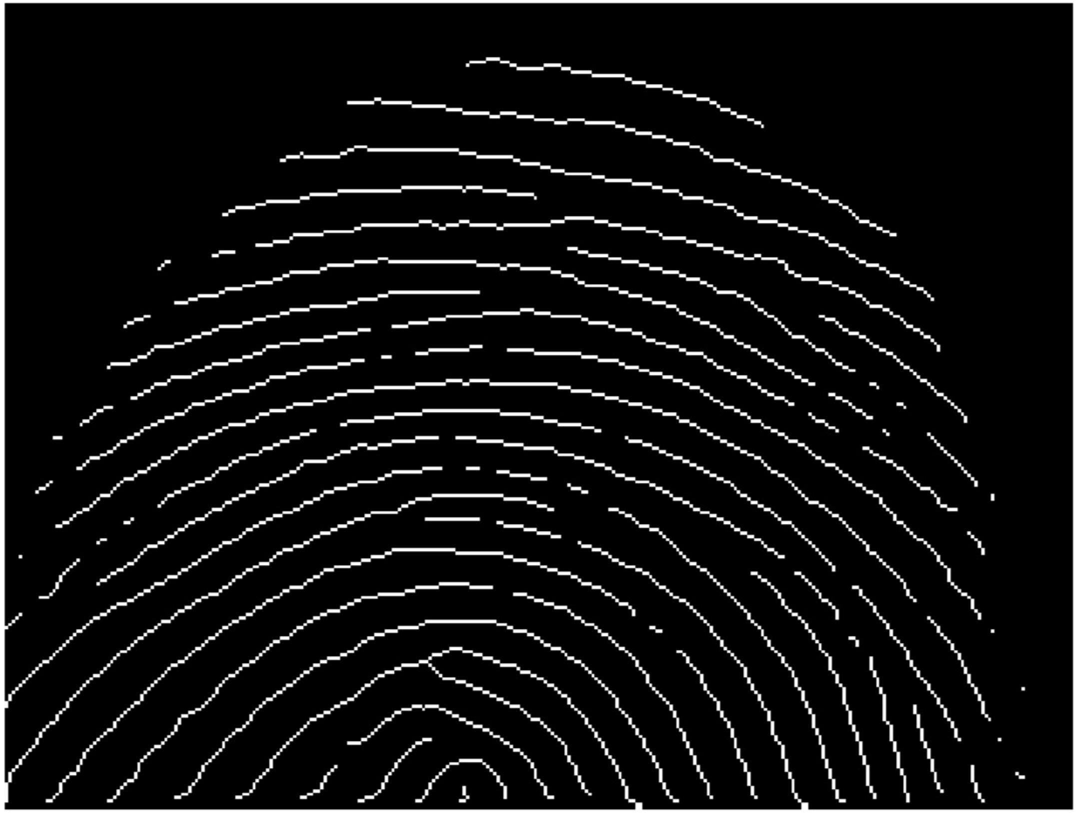


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again.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

9. Thinning

- MATLAB: s90Thinning.m



9. Thinning

- Example

9. Thinning

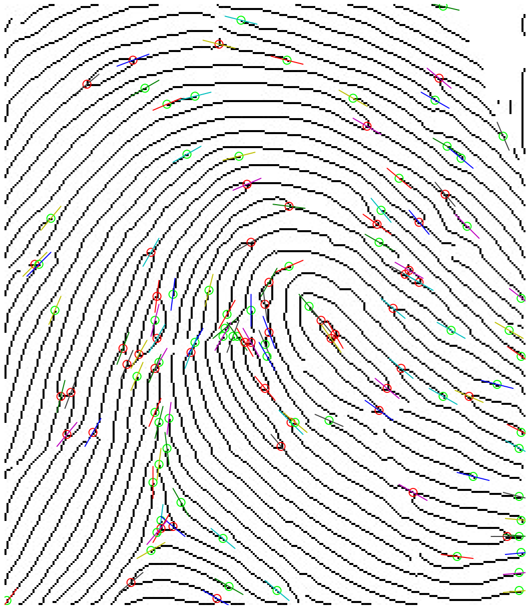
- Example

9. Thinning

- Example

9. Thinning

- Example



9. Thickening

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

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

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

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

- The structuring elements used for thickening have the same form as those shown before, but with all 1s and 0s interchanged.

9. Thickening

- 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 and then complement the result.

- In other words, to thicken a set A,

  1. We form $C = A^C$;

  2. Thin C; and then

  3. Form C^C.

- Thickening by this method usually is followed by postprocessing to remove disconnected points.

9. Thickening

- Example:

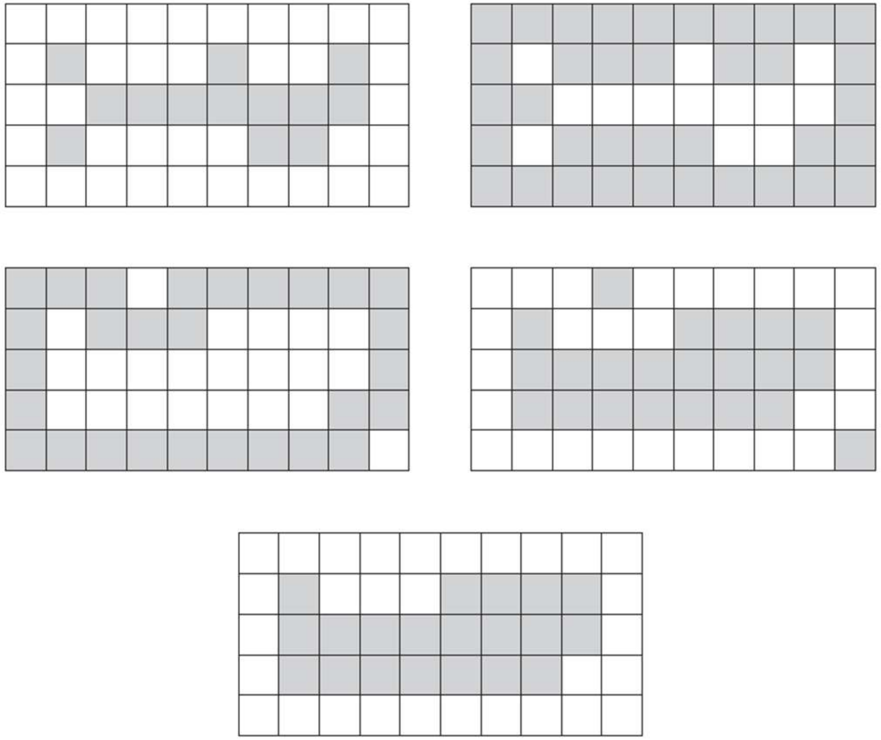


FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

10. Skeletons

• The skeleton of A can be expressed in terms of erosions and openings:

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

Successive erosions.

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

• S(A) can be obtained as the union of the skeleton **subsets** S_k(A).

• *k* is the last iterative step before A erodes to an empty set.

$$K = \max\{k \mid (A \ominus kB) \neq \varnothing\}$$

10. Skeletons

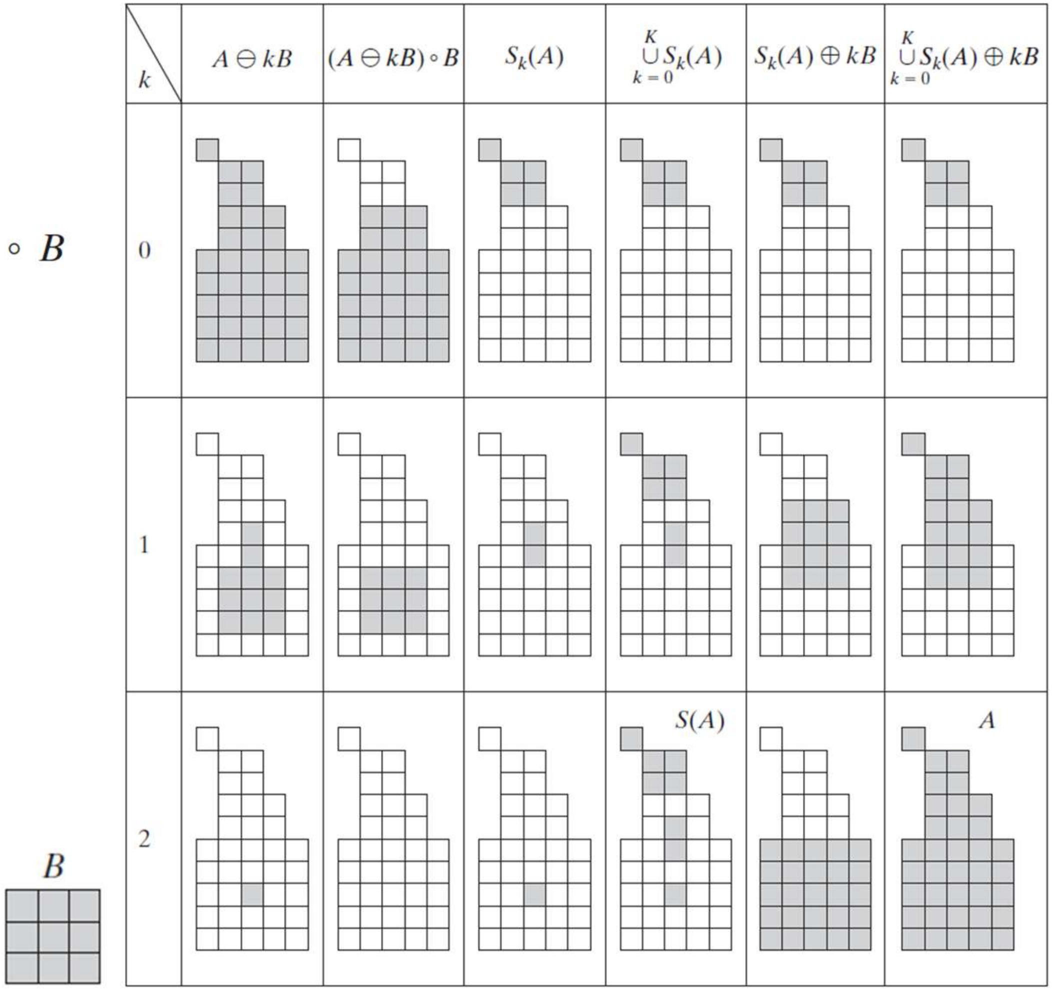
- Example:

$$K = \max\{k \mid (A \ominus kB) \neq \varnothing\}$$

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

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

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

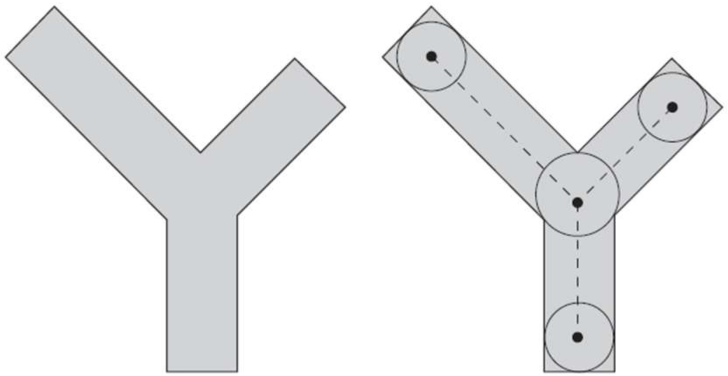


10. Skeletons

- Intuition

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

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

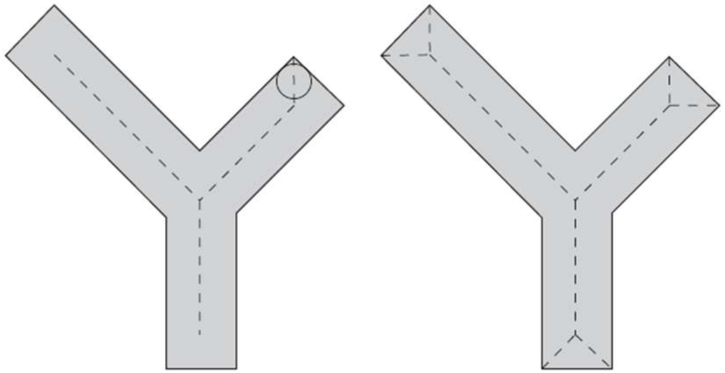


10. Skeletons

- Intuition

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

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



10. Skeletons

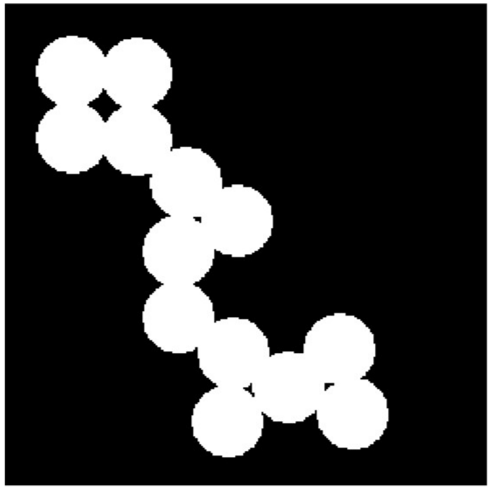
- Example:

```
clear all
close all
clc
BW1 = imread('circles.png');
imshow(BW1);
BW2 = bwmorph(BW1,'thin',Inf);
figure
imshow(BW2)
BW3 = bwmorph(BW1,'skel',Inf);
figure
imshow(BW3)
```

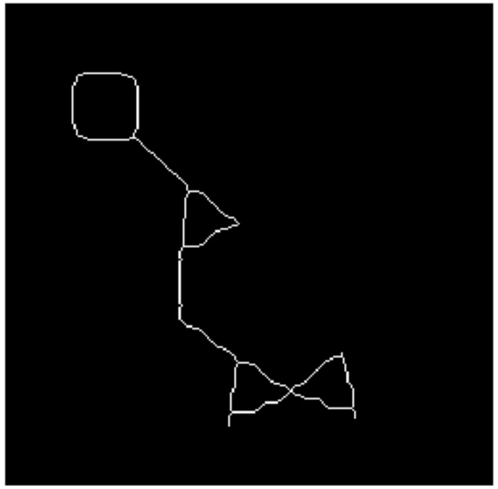
10. Skeletons

- Example:

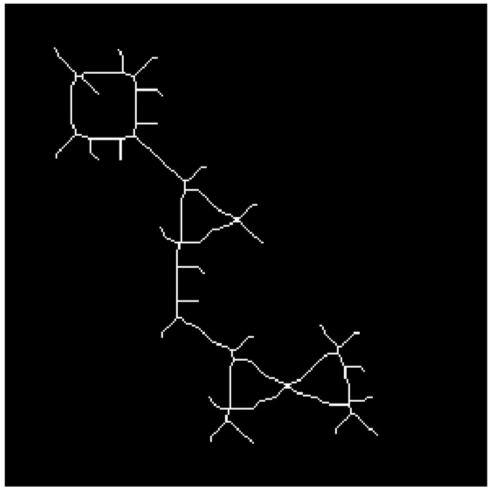
Original



Thinning



Skeletonization



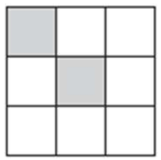
11. Pruning

- **Pruning** methods are an essential **complement to thinning and skeletonizing** algorithms because these procedures tend to leave parasitic components that need to be "**cleaned up**" by postprocessing.

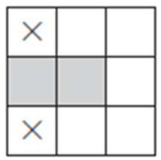
- Thinning of an input set with a sequence *B* of structuring elements designed to detect only end points achieves the desired result.

- First we calculate,

$$X_1 = A \otimes \{B\}$$



B^5, B^6, B^7, B^8 (rotated 90°)



B^1, B^2, B^3, B^4 (rotated 90°)

11. Pruning

• The next step is to "restore" the character to its original form. To do so first we form a set containing all end points in X_1.

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

• Next we dilate the end points (3 times, for instance), using set A as a delimiter.

$$X_3 = (X_2 \oplus H) \cap A$$

where *H* is a 3x3 structuring element of 1s and the intersection with A is applied after each step.

• Finally, the union of X_1 and X_3 yields the desired result,

$$X_4 = X_1 \cup X_3$$

11. Pruning

- Example:

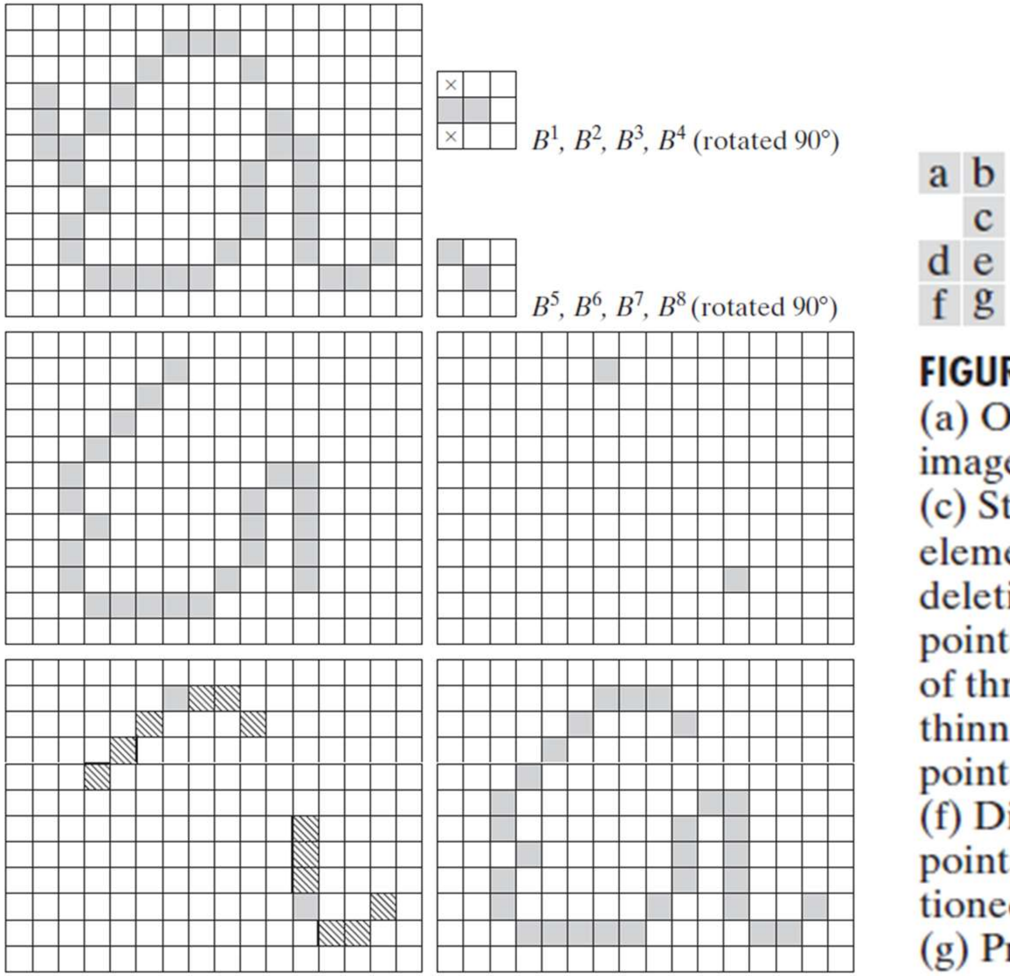


FIGURE 9.25 (a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.

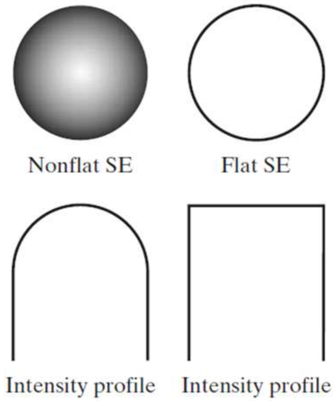
12. Gray-Scale Morphology

• In this section, we extend to gray-scale images the basic operations of dilation, erosion, opening, and closing.

• Structuring elements in gray-scale morphology belong to one of two categories: **nonflat** and **flat**.

• Gray-scale SEs are used infrequently in practice.

• All the examples are based on symmetrical, flat structuring elements of unit height whose origins are at the center.



12. Gray-Scale Morphology

- Erosion

  - The *erosion* of *f* by a flat structuring element *b* at any location (x, y) is defined as the **minimum value** of the image in the region coincident with *b* when the origin of *b* is at (x , y),

$$\left[f \ominus b\right](x, y) = \min_{(s,t)\in b}\left\{f(x+s, y+t)\right\}$$

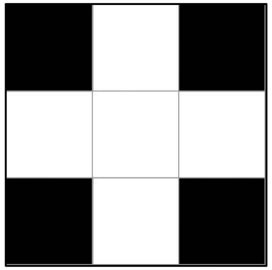
  - In general an eroded gray-scale image will be darker than the original.

  - The sizes of **bright** features will be **reduced**, and that the sizes of **dark** features will be **increased**

12. Gray-Scale Morphology

- Erosion (flat SE)

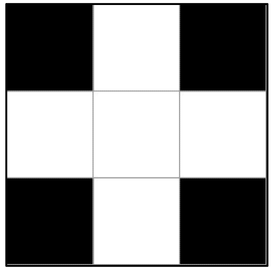
0	0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	50	40	30	0
0	0	0	60	130	0	0	50	0	0
0	0	0	90	120	175	0	0	0	0
0	0	0	0	130	0	175	0	0	0
0	0	90	0	0	120	220	230	0	0
0	0	0	50	0	0	100	0	0	0
0	0	60	5	70	0	0	0	0	0
0	0	0	120	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



12. Gray-Scale Morphology

- Erosion (flat SE)

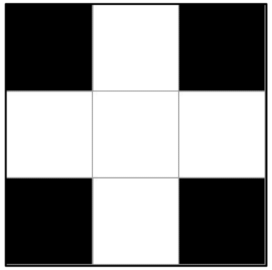
0	0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	50	40	30	0
0	0	0	60	130	0	0	50	0	0
0	0	0	90	120	175	0	0	0	0
0	0	0	0	130	0	175	0	0	0
0	0	90	0	0	120	220	230	0	0
0	0	0	50	0	0	100	0	0	0
0	0	60	5	70	0	0	0	0	0
0	0	0	120	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



12. Gray-Scale Morphology

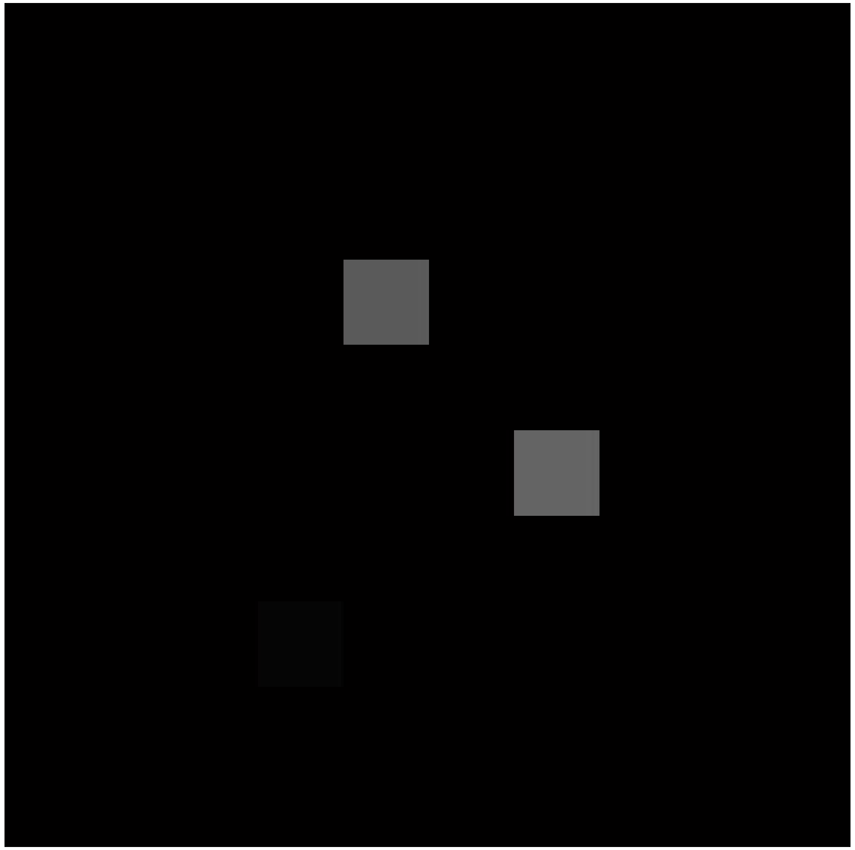
- Erosion (flat SE)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	30	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	90	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	100	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	5	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



12. Gray-Scale Morphology

- Erosion (flat SE)



12. Gray-Scale Morphology

- Dilation (flat SE)

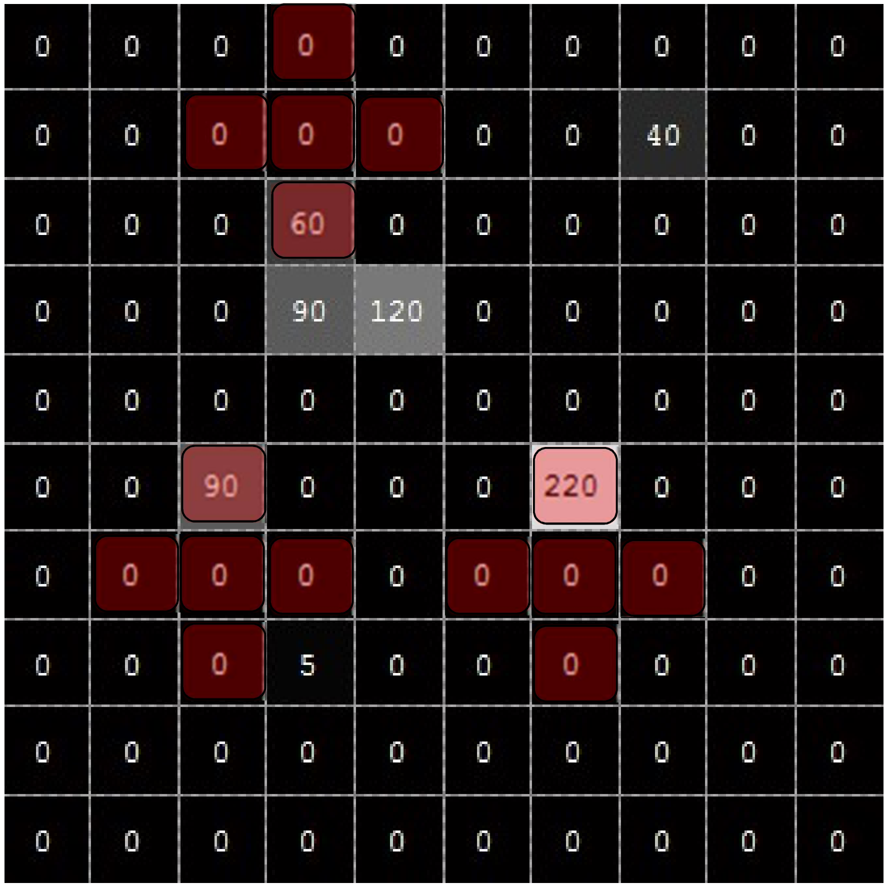
  - Similarly, the *dilation* of f by a flat structuring element *b* at any location (x, y) is defined as the maximum value of the image in the window outlined by $\hat{b}$ when the origin of $\hat{b}$ is at (x, y).

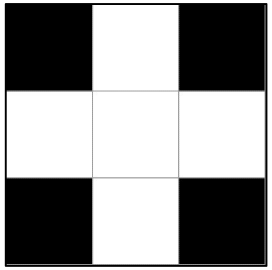
$$[f \oplus b](x, y) = \max_{(s,t) \in b} \{ f(x - s, y - t) \}$$

  - The **bright** features were **thickened** and the intensities of the **dark** features were **reduced**.

12. Gray-Scale Morphology

- Dilation (flat SE)

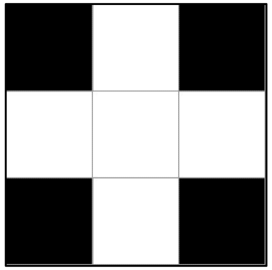




12. Gray-Scale Morphology

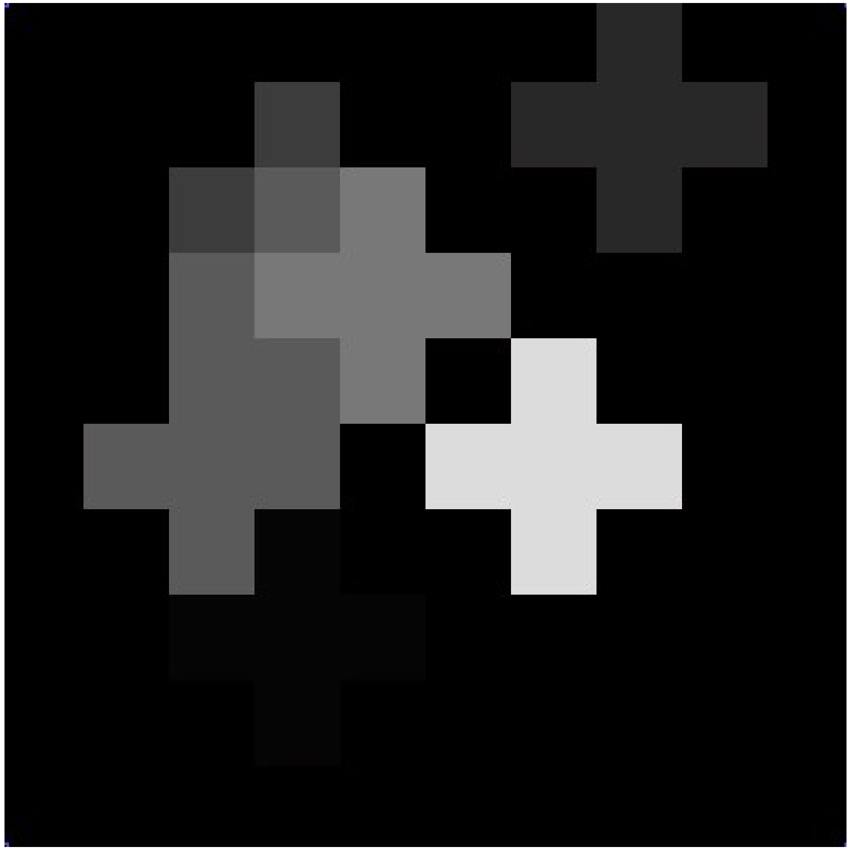
- Dilation (flat SE)

0	0	0	0	0	0	0	40	0	0
0	0	0	60	0	0	40	40	40	0
0	0	60	90	120	0	0	40	0	0
0	0	90	120	120	120	0	0	0	0
0	0	90	90	120	0	220	0	0	0
0	90	90	90	0	220	220	220	0	0
0	0	90	5	0	0	220	0	0	0
0	0	5	5	5	0	0	0	0	0
0	0	0	5	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



12. Gray-Scale Morphology

- Dilation (flat SE)



12. Gray-Scale Morphology

- Erosion and dilation example (flat SE)

a b c

FIGURE 9.35 (a) A gray-scale X-ray image of size 448 × 425 pixels. (b) Erosion using a flat disk SE with a radius of two pixels. (c) Dilation using the same SE. (Original image courtesy of Lixi, Inc.)

12. Gray-Scale Morphology

- Erosion and dilation (nonflat SE)

  - The *erosion* of image *f* by nonflat structuring element b_N, is defined as,

$$\left[f \ominus b_N\right](x, y) = \min_{(s,t)\in b_N}\left\{f(x+s, y+t) - b_N(s,t)\right\}$$

  - The *dilation* using a nonflat SE is defined as,

$$\left[f \oplus b_N\right](x, y) = \max_{(s,t)\in b_N}\left\{f(x-s, y-t) + b_N(s,t)\right\}$$

12. Gray-Scale Morphology

- Erosion and dilation (nonflat SE)

  - Problems:

    - ✓ Erosion and dilation using a nonflat SE are not bounded in general by the values of *f*, which can present **problems in interpreting results**.

    - ✓ Potential difficulties in selecting meaningful elements for b_N.

    - ✓ Computational burden.

12. Gray-Scale Morphology

- Erosion and dilation (nonflat SE)

  - Erosion and dilation are duals with respect to function complementation and reflection;

$$(f \ominus b)^c = (f^c \oplus \hat{b})$$

$$(f \oplus b)^c = (f^c \ominus \hat{b})$$

$$f^c = -f(x, y)$$

$$\hat{b} = b(-x, -y)$$

12. Gray-Scale Morphology

- Opening and Closing

  - The opening of image *f* by structuring element *b* is

$$f \circ b = (f \ominus b) \oplus b$$

  - Similarly, the closing of *f* by *b* is

$$f \bullet b = (f \oplus b) \ominus b$$

  - The opening and closing for gray-scale images are duals with respect to complementation and SE reflection:

$$(f \bullet b)^c = f^c \circ \hat{b} \qquad (f \circ b)^c = f^c \bullet \hat{b}$$

12. Gray-Scale Morphology

- Opening and Closing

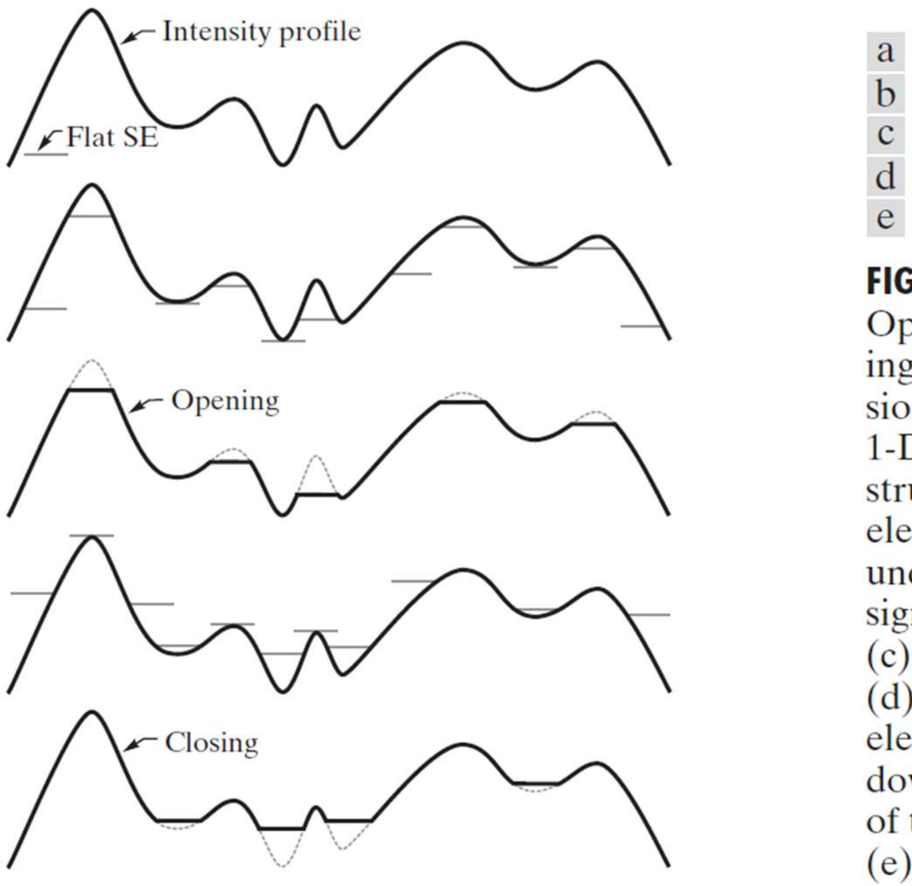
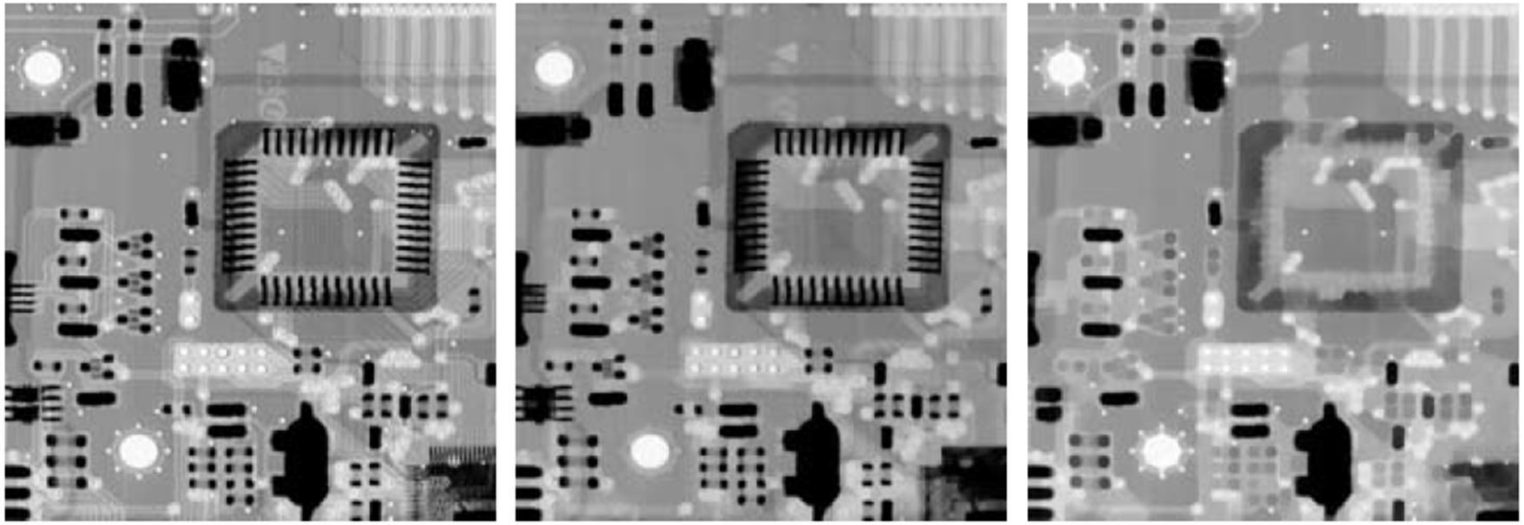


FIGURE 9.36 Opening and closing in one dimension. (a) Original 1-D signal. (b) Flat structuring element pushed up underneath the signal. (c) Opening. (d) Flat structuring element pushed down along the top of the signal. (e) Closing.

12. Gray-Scale Morphology

- Opening and Closing



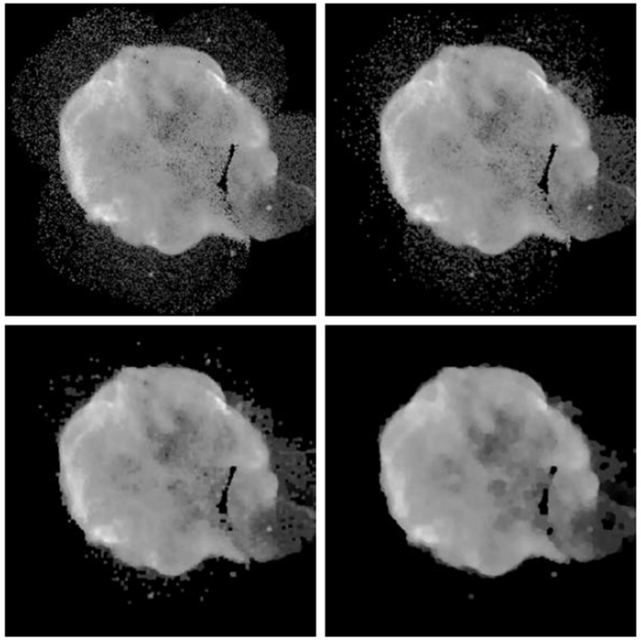
a b c

FIGURE 9.37 (a) A gray-scale X-ray image of size 448 × 425 pixels. (b) Opening using a disk SE with a radius of 3 pixels. (c) Closing using an SE of radius 5.

12. Gray-Scale Morphology

- Morphological smoothing

  - *Alternating sequential filtering*: an opening–closing sequence starts with the original image, but subsequent steps perform the opening and closing on the results.



a b
c d

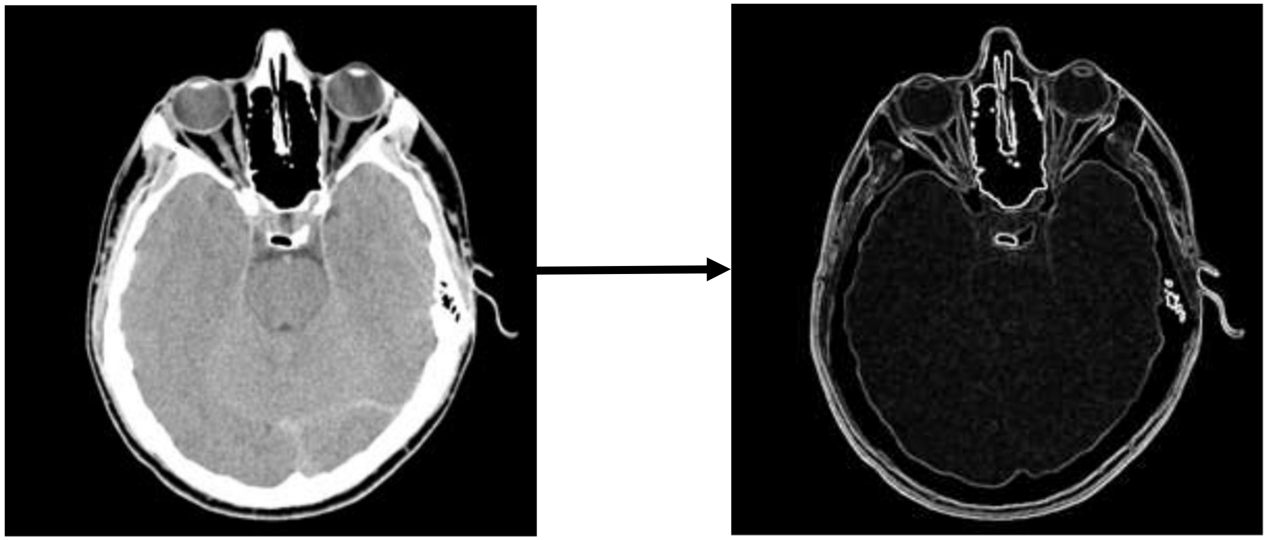
FIGURE 9.38 (a) 566 × 566 image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of performing opening and closing sequences on the original image with disk structuring elements of radii, 1, 3, and 5, respectively. (Original image courtesy of NASA.)

12. Gray-Scale Morphology

- Morphological gradient

  - Dilation and erosion can be used in combination with image subtraction to obtain the morphological gradient of an image,

$$g = (f \oplus b) - (f \ominus b)$$



12. Gray-Scale Morphology

- Top-hat and bottom-hat transformations

  - An important use of top-hat transformations is in correcting the effects of nonuniform illumination.

  - Proper illumination plays a central role in the process of extracting objects from the background.

  - The *top-hat transformation* of a grayscale image *f* is defined as *f* minus its *opening (*used for light objects on a dark background),

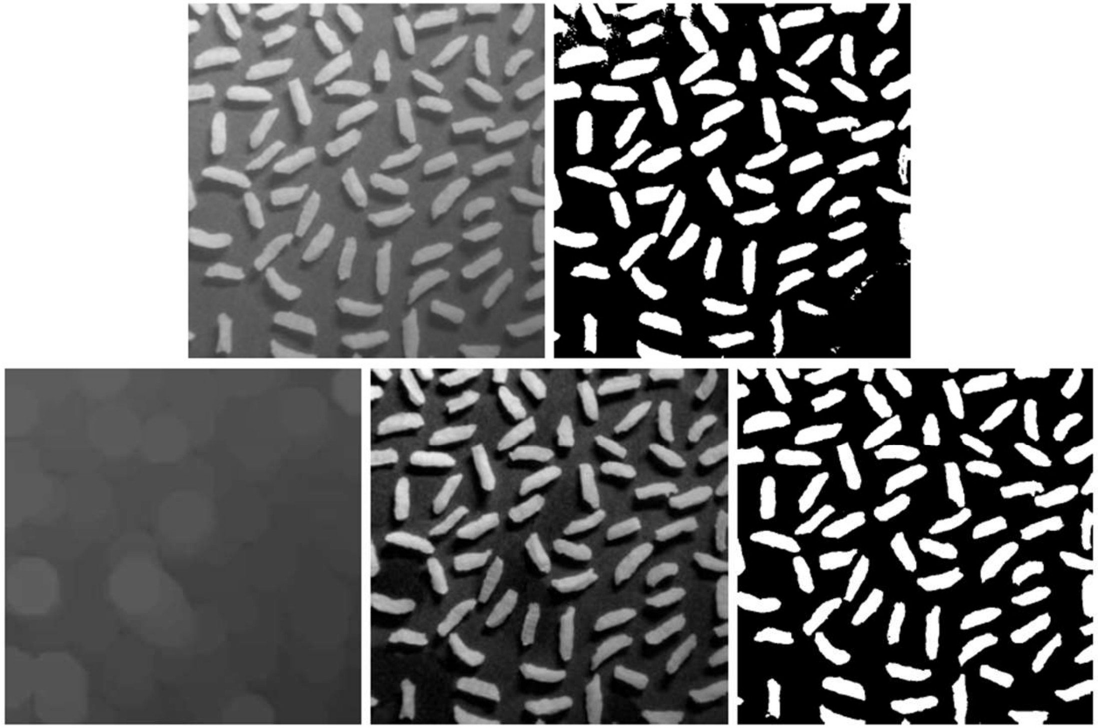
$$T_{\text{hat}}(f) = f - (f \circ b)$$

  - The *bottom-hat transformation* f of is defined as the closing of f minus f (used for the converse),

$$B_{\text{hat}}(f) = (f \bullet b) - f$$

12. Gray-Scale Morphology

- Top-hat and bottom-hat transformations

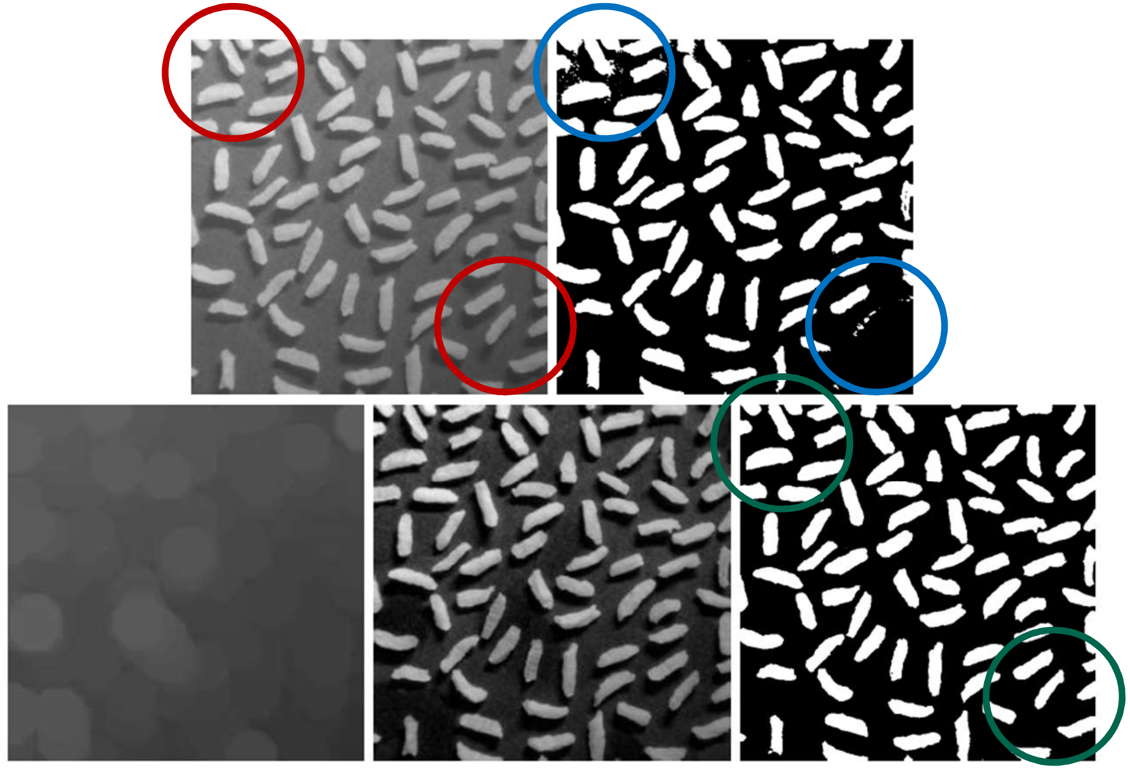


a b
c d e

FIGURE 9.40 Using the top-hat transformation for *shading correction*. (a) Original image of size 600 × 600 pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.

12. Gray-Scale Morphology

- Top-hat and bottom-hat transformations



a b
c d e

FIGURE 9.40 Using the top-hat transformation for *shading correction*. (a) Original image of size 600 × 600 pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.

12. Gray-Scale Morphology

- Granulometry

  - In terms of image processing, *granulometry* is a field that deals with determining the size distribution of particles in an image.

  - In practice, particles seldom are neatly separated, which makes particle counting by identifying individual particles a difficult task.

  - Morphology can be used to estimate particle size distribution indirectly, without having to identify and measure every particle in the image.

12. Gray-Scale Morphology

- Granulometry

  - With particles having regular shapes that are lighter than the background, the method consists of applying openings with SEs of increasing size.

  - For each opening, the sum of the pixel values in the opening is computed.

  - This sum, sometimes called the *surface area*, decreases as a function of increasing SE size because.

  - This procedure yields a 1-D array of such numbers.

  - To emphasize changes between successive openings, we compute the difference between adjacent elements of the 1-D array.

12. Gray-Scale Morphology

- Granulometry

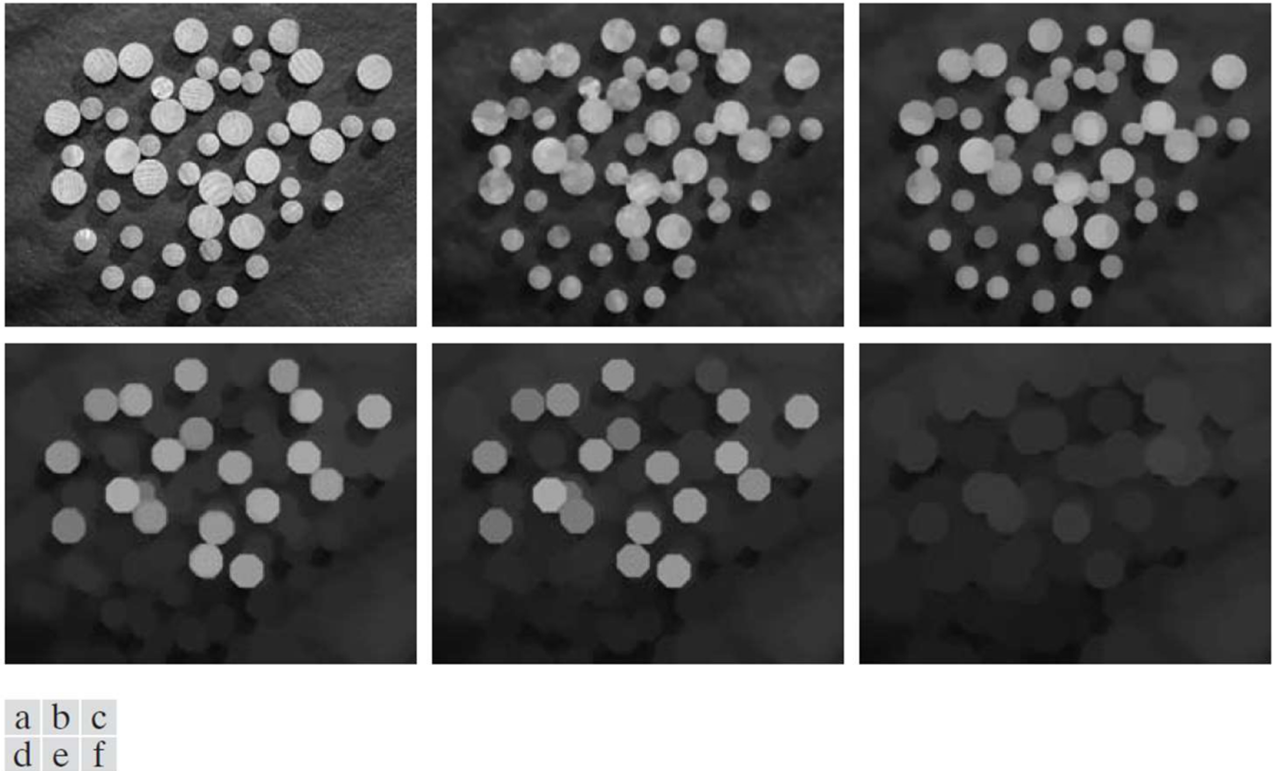
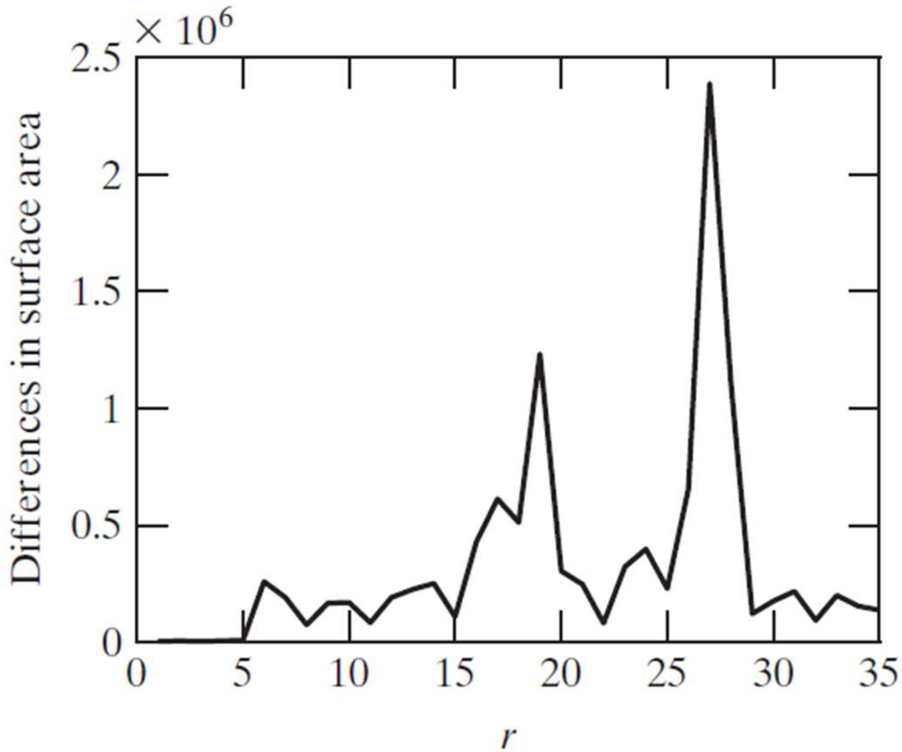


FIGURE 9.41 (a) 531 × 675 image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)

12. Gray-Scale Morphology

- Granulometry



12. Gray-Scale Morphology

- MATLAB: s129Granul.m