

Digital Image Processing

Morphological Image Processing



Introduction

- ▶ **Morphology:** a branch of biology that deals with the form and structure of animals and plants
- ▶ *Mathematical morphology* is a tool for extracting image components useful in the representation and description of region shape, such as boundaries, skeletons and convex hulls. The language of mathematical morphology is set theory, and as such it can apply directly to binary (two-level) images

Preliminaries (1)

Language of mathematical Morphological is Set Theory.

► **Reflection**

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

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

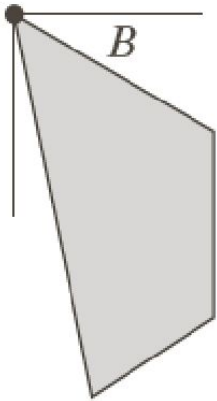
B is the set of pixels (2-D points) representing an object in image. And b is co-ordinate of the pixel

► **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 .

Examples: Structuring Elements (1)

Structure elements (SE): Small sets or sub-images used to probe an image under study for properties of interest

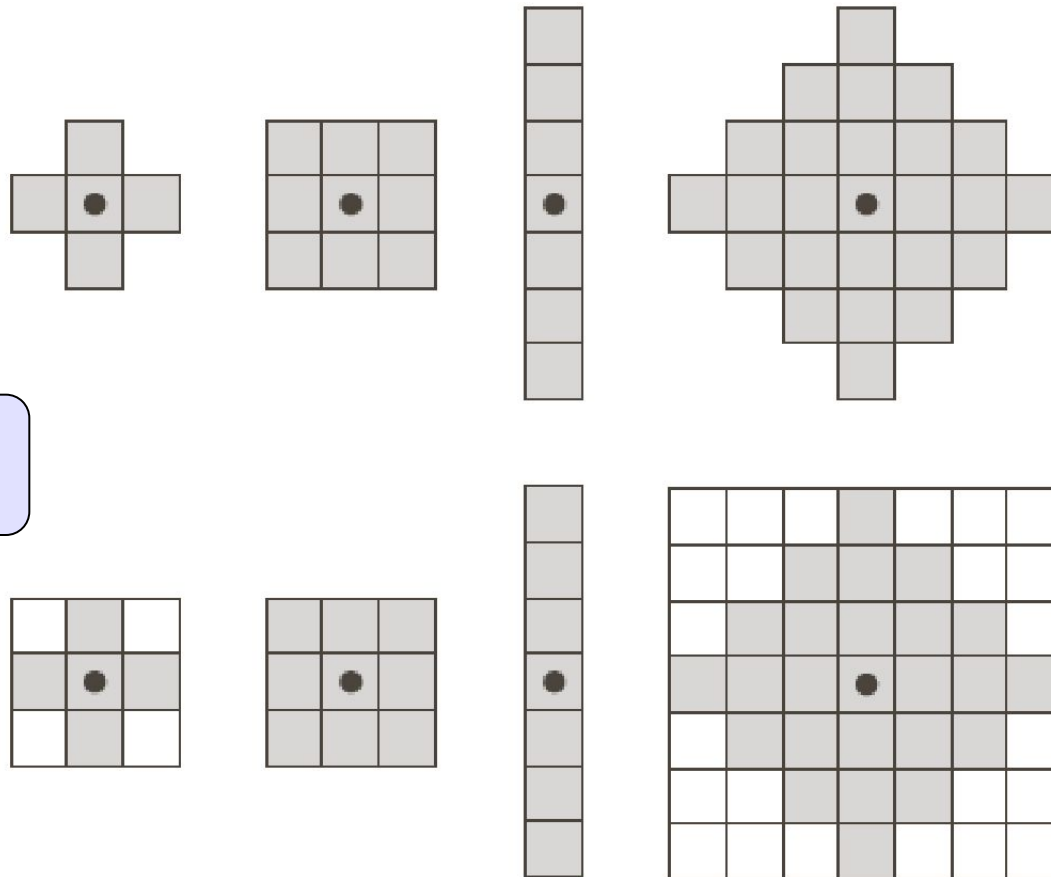
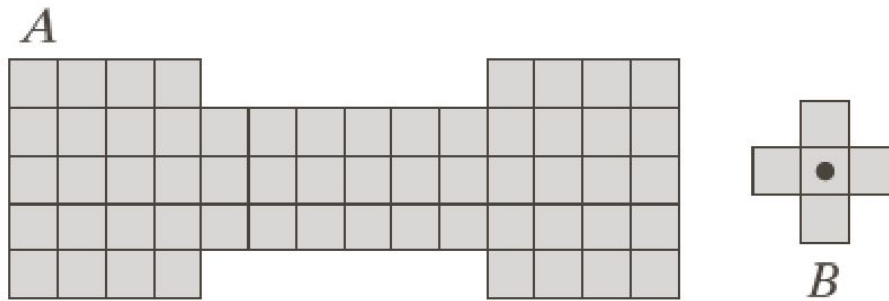


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Examples: Structuring Elements (2)

Accommodate the entire structuring element 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.

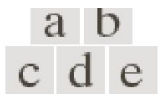


FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

Erosion: It shrinks or thins object in binary image

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

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

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

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

Erosion The basic effect of erosion operator on a binary image is to erode away the boundaries of foreground pixels (usually the white pixels).

- ▶ Thus areas of foreground pixels shrink in size, and "holes" within those areas become larger.
- ▶ Mathematically, erosion of set A by set B is a set of all points x such that B translated by x is still contained in A .
- ▶ To compute the erosion of a binary input image by this structuring element, we consider each of the foreground pixels in the input image in turn. For each 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 coordinates.
- ▶ With the structuring element chosen as above, the effect of this operation is to remove any foreground pixel that is not completely surrounded by other foreground pixels, assuming 8-connectedness. We can also see that this operation can be performed on binary images simply by applying a logical AND function.

Example of Erosion (1)

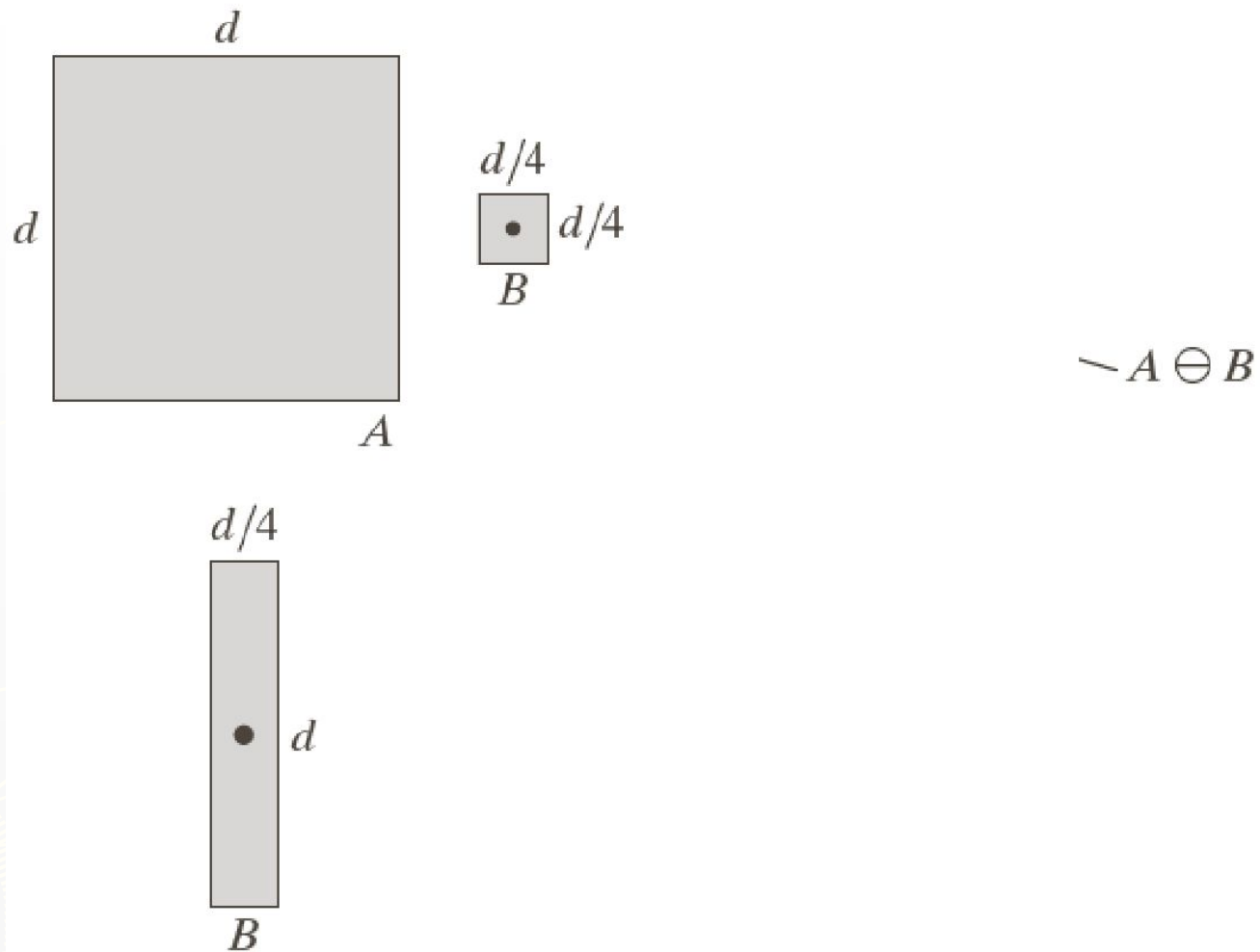
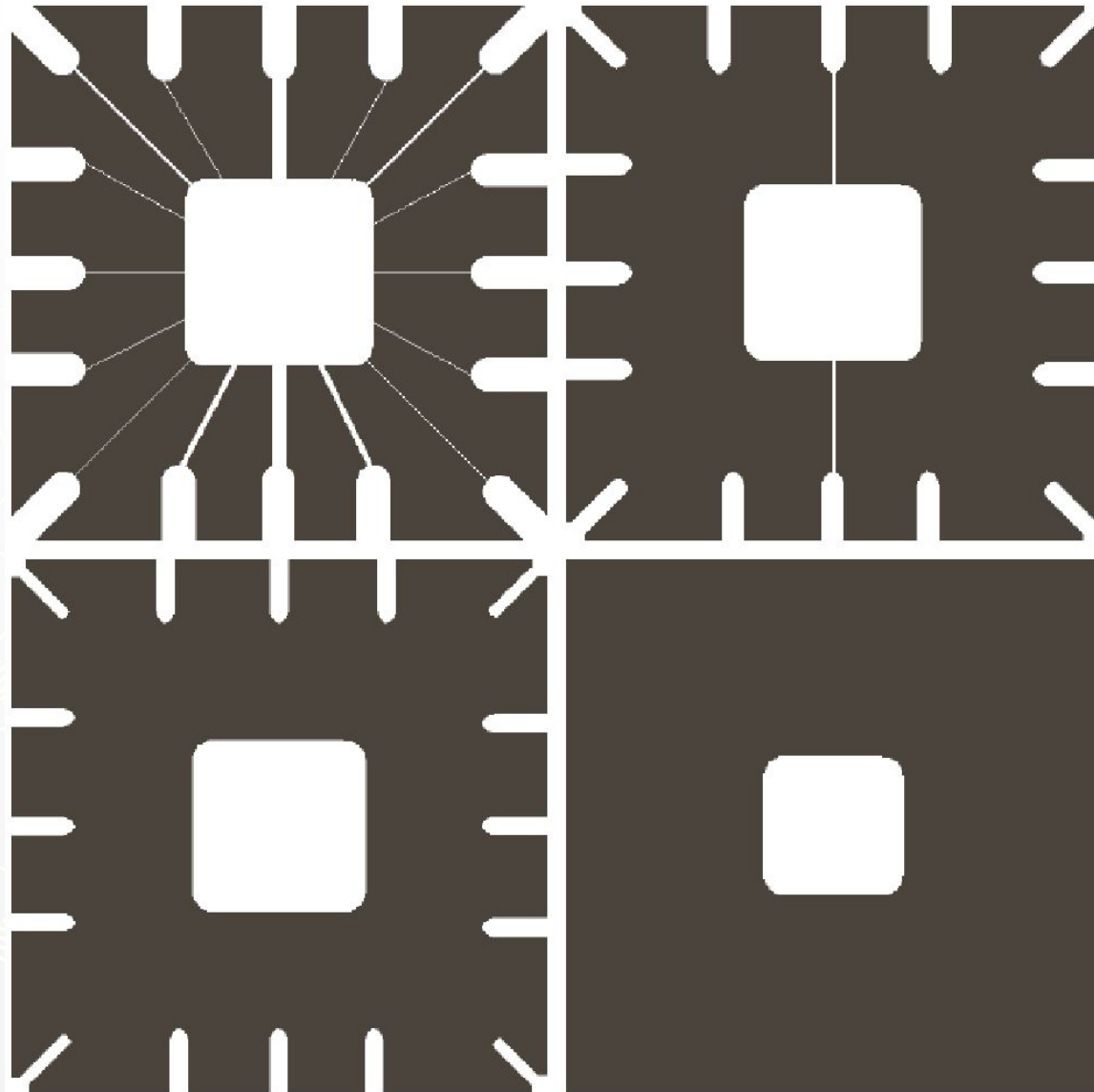


FIGURE 9.4 (a) Set A . (b) Square structuring element, B . (c) Erosion of A by B , shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. * The dotted border in (c) and (e) is the boundary of set A , shown only for reference.

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.

- ▶ The **dilation** is the other of the two basic operators in mathematical morphology, the first one being erosion. The basic effect of dilation on binary images is to enlarge the areas of foreground pixels (*i.e.* white pixels) at their borders. The areas of foreground pixels thus grow in size, while the background "holes" within them shrink.
- ▶ Mathematically, erosion of A by B is the set of all x displacement such that B and A overlap by at least one non-zero element.
- ▶ Again let us take a 3x3 matrix for the structuring element, with the center pixel used as the origin of the set. B , then the dilation can be performed using the logical OR function:

Dilation : It grows or thickens the object in binary image

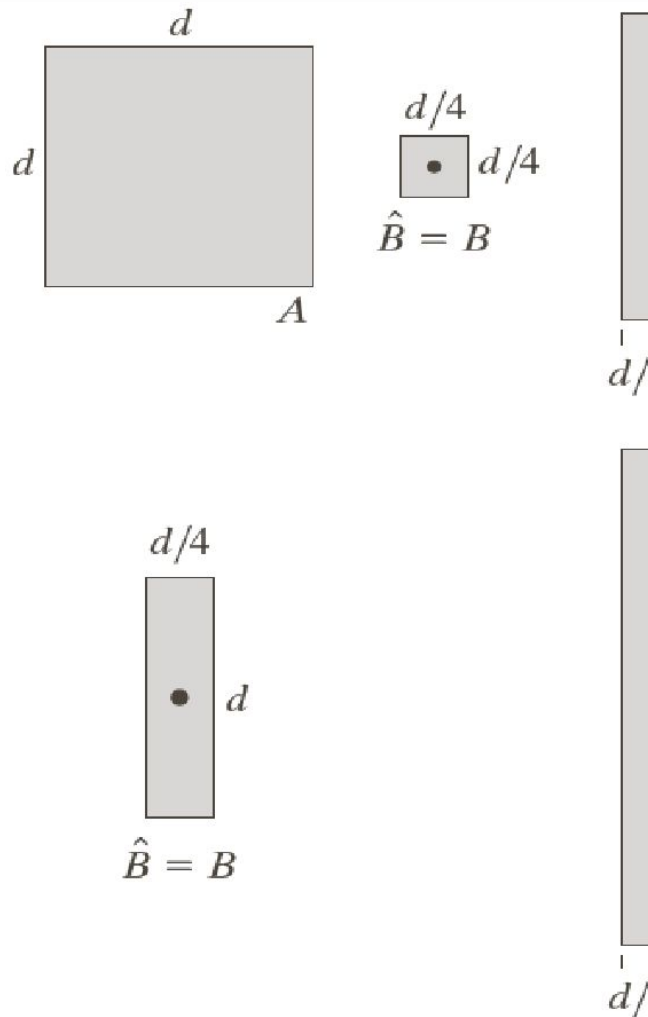
With A and B as sets in Z^2 , the dilation of A by B , denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \mid \left(\overset{\boxtimes}{B} \right)_z \cap A \neq \emptyset \right\}$$

The set of all displacements z , the translated $\overset{\boxtimes}{B}$ and A overlap by at least one element.

$$A \oplus B = \left\{ z \mid \left[\left(\overset{\boxtimes}{B} \right)_z \cap A \right] \subseteq A \right\}$$

Examples of Dilation (1)



a	b	c
d		e

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)

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



0	1	0
1	1	1
0	1	0

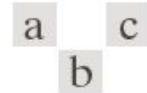


FIGURE 9.7

(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.

Duality

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

$$(A \ominus B)^c = A^c \oplus B$$

and

$$(A \oplus B)^c = A^c \ominus B$$

Opening and Closing

- ▶ Opening generally smooths the contour of an object,
- ▶ Smooths convex part of the object
- ▶ Closing tends to smooth sections of contours but it fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour,
- ▶ Smooths concave part of the object

Opening and Closing

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

$$A \boxminus B = (A \ominus B) \oplus B \Rightarrow A \circ B = \bigcup \{ (B)_z \mid (B)_z \subseteq A \}$$

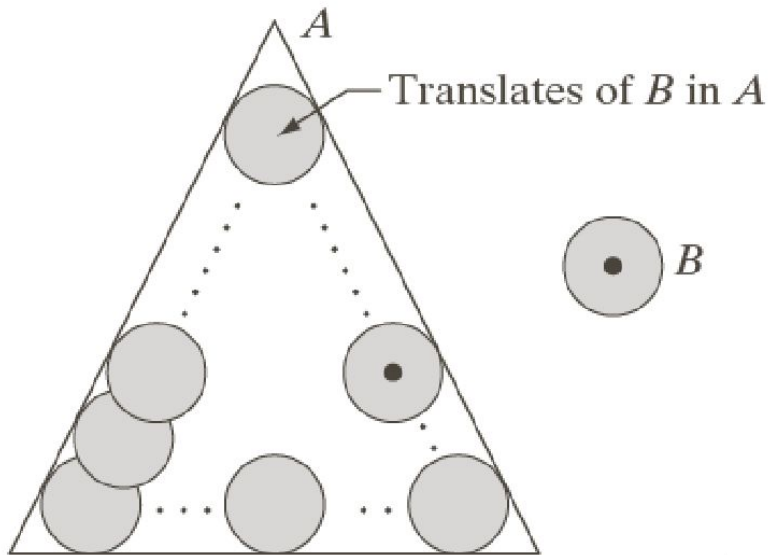
Union of all translates of B that fits into A (B as ball rolls inside the boundary of A . **Smooths convex** part of the object

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

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

Intersection of all translates of B such that B and A overlap by at least one non-zero element. (B as ball rolls outside the boundary of A .. **Smooths concave** part of the object

Example: Opening

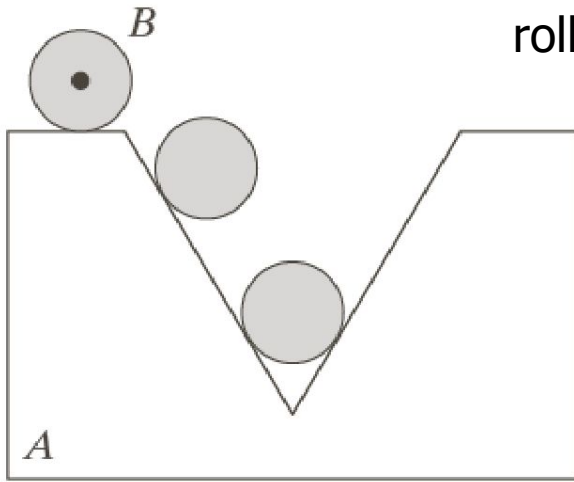


a b c d

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.

Example: Closing

Intersection of all translates of B that fits into A (B as ball rolls outside the boundary of A ..



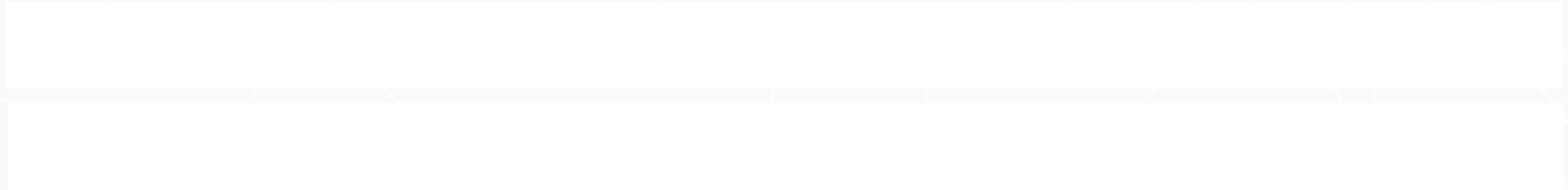
a b c

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

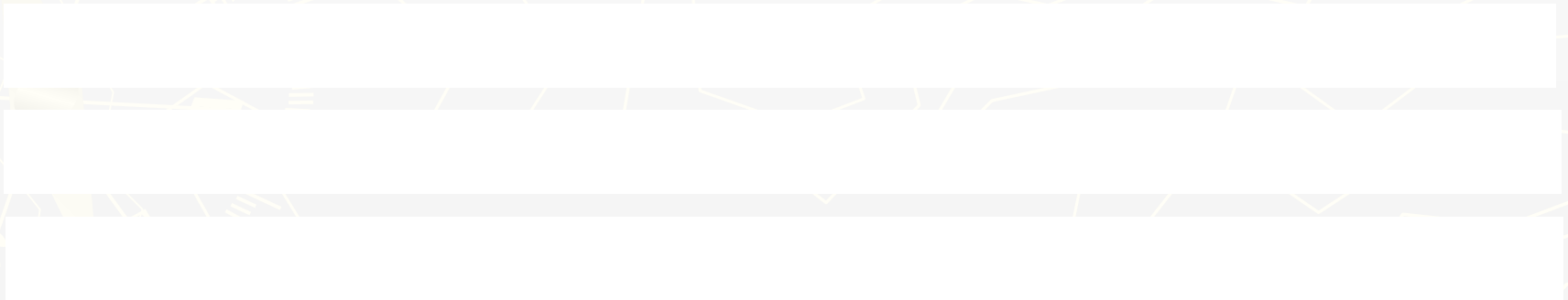
The Properties of Opening and Closing

► Properties of Opening

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



► Properties of Closing



M



Metasploit

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

msf5

The Hit-or-Miss Transformation

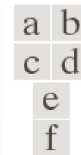
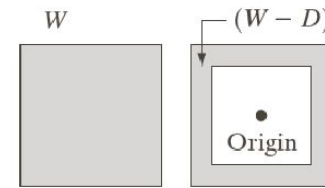
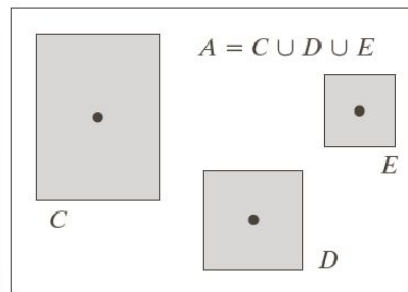


FIGURE 9.12

(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 .

$$B = (B_1, B_2)$$

B_1 : object

B_2 : background

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

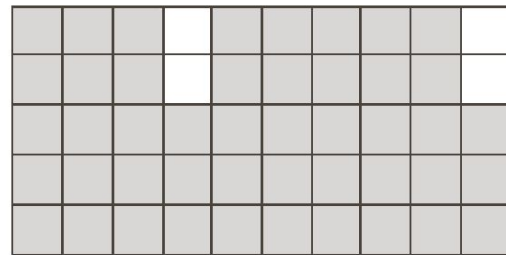
Some Basic Morphological Algorithms (1)

► **Boundary Extraction**

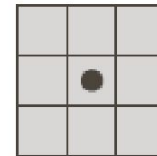
The boundary of a set A , can be obtained by first eroding A by B and then performing the set difference between A and its erosion.

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

Example 1



A



B

$A \ominus B$

$\beta(A)$

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.

Example 2



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 (2)

► **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.

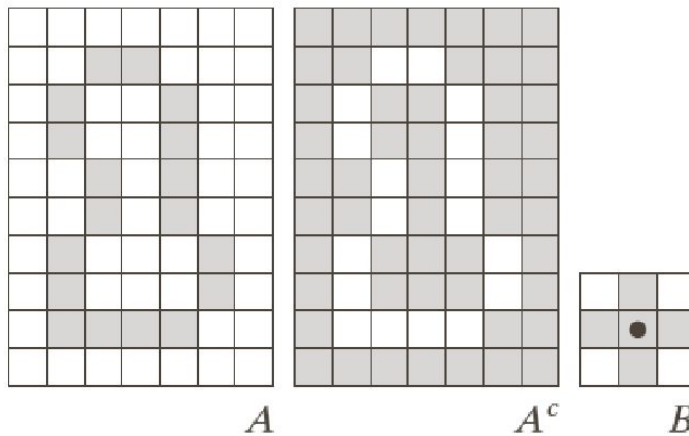
Some Basic Morphological Algorithms (2)

► Hole Filling

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

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

Stop the iteration if $X_k = X_{k-1}$



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)].

X_0

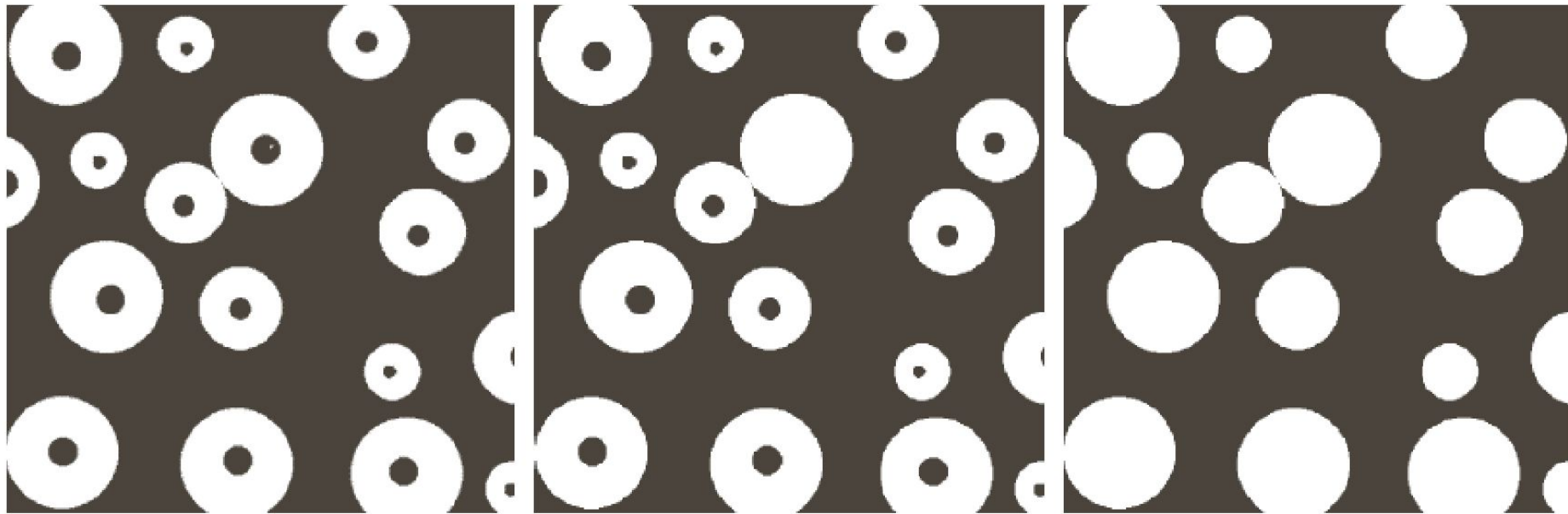
X_1

X_2

X_6

X_8

$X_8 \cup A$



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 (3)

► **Extraction of Connected Components**

Central to many automated image analysis applications.

Let A be a set containing one or more connected components, and form an array X_0 (of the same size as the array containing A) whose elements are 0s, except at each location known to correspond to a point in each connected component in A , which is set to 1.

Some Basic Morphological Algorithms (3)

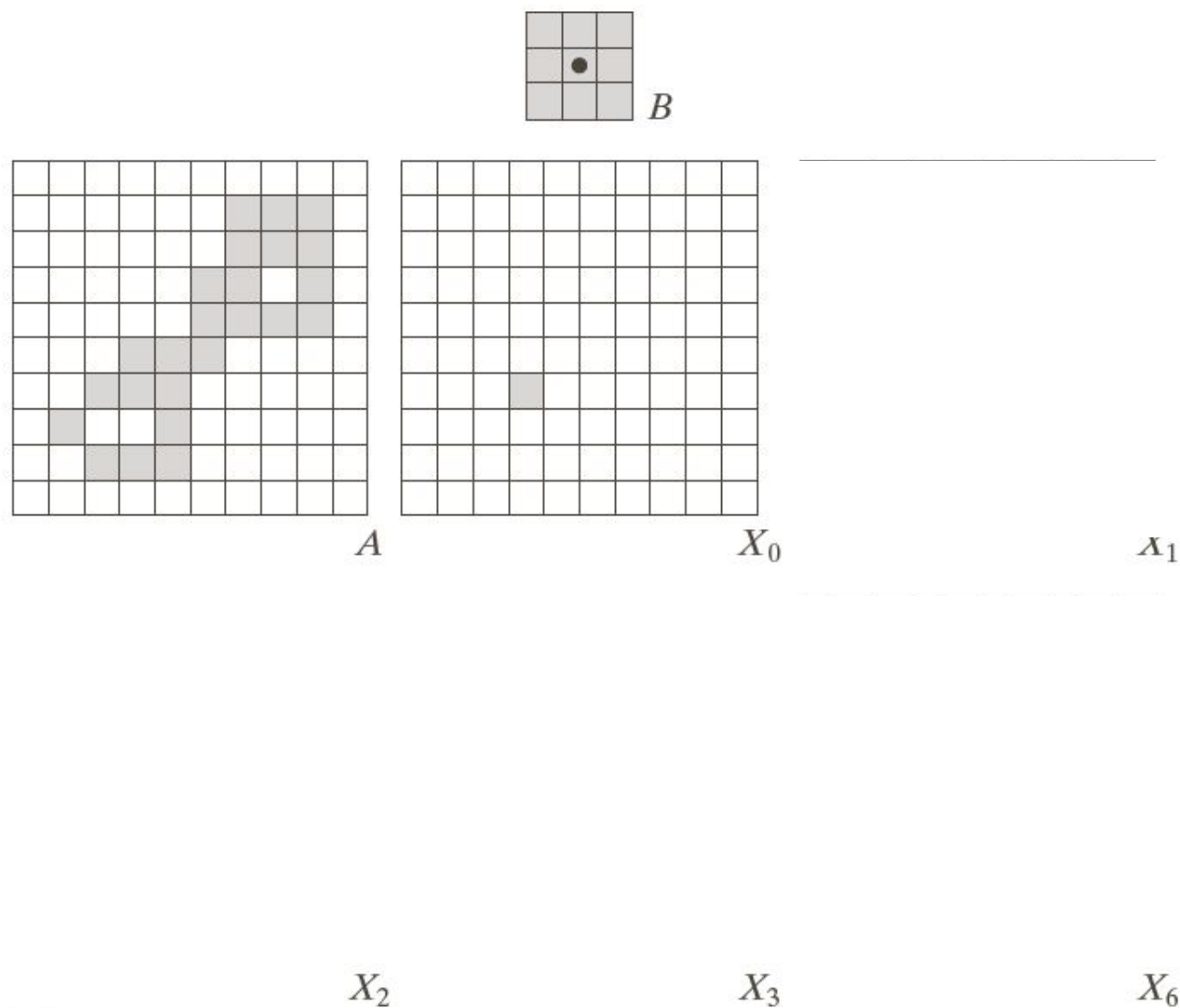
► **Extraction of Connected Components**

Central to many automated image analysis applications.

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

B : structuring element

until $X_k = X_{k-1}$



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).

► Thank You

