

# Morphological processing

Instructor: Yiming Xiao

Email: [yiming.xiao@concordia.ca](mailto:yiming.xiao@concordia.ca)

Department of Computer Science  
and Software Engineering  
Concordia University

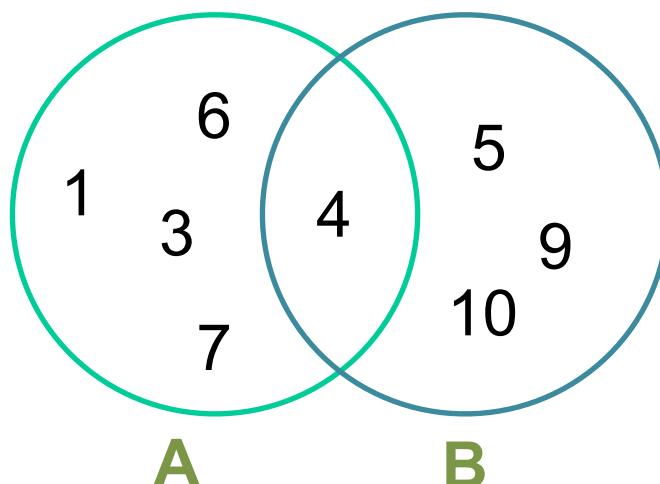
Materials adapted from Dr. T. D. Bui

# Morphology

The term ***morphology*** refers to the description of the properties of shape and structure of any objects.

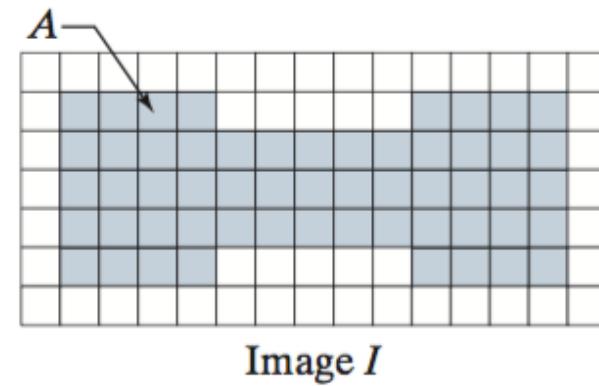
Operations of mathematical morphology were originally defined as operations on sets, but it soon became clear that they are also useful in the processing tasks of the set of points in the two-dimensional space, aka. images.

## Sets



$$\mathbf{A} = \{1, 3, 4, 6, 7\} \quad \mathbf{B} = \{4, 5, 9, 10\}$$

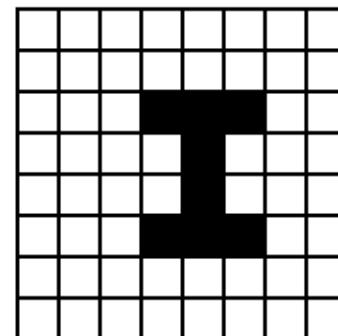
*Defined in the  $Z^2$  space*



**foreground pixels in an image**

# Morphological Processing

- A binary image can be considered as a set by considering “black” pixels (with image value “1”) as elements in the set and “white” pixels (with value “0”) as outside the set.

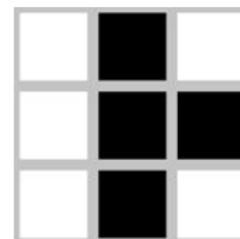


- Morphological filters are essentially set operations

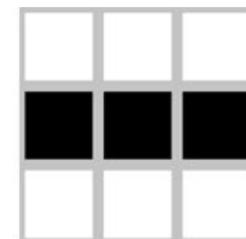
# Set Operations

***Basic operations: union, intersection, complement, difference***

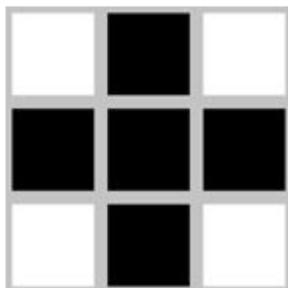
Input Images



A

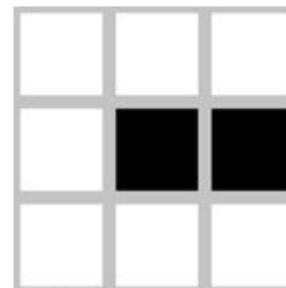


B



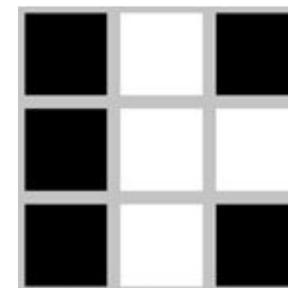
$$C = A \cup B$$

union (OR)



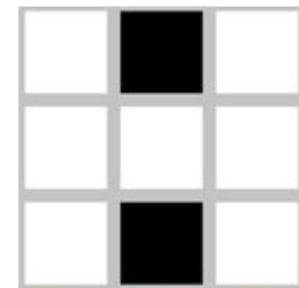
$$C = A \cap B$$

intersection  
(AND)



$$C = A^c$$

complement



$$C = A \setminus B$$

difference

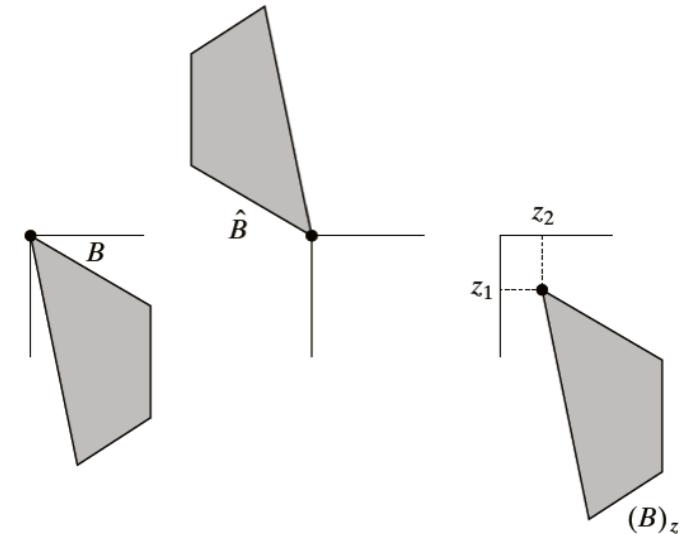
# Set Operations

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

*rotating by 180 degrees about the origin*

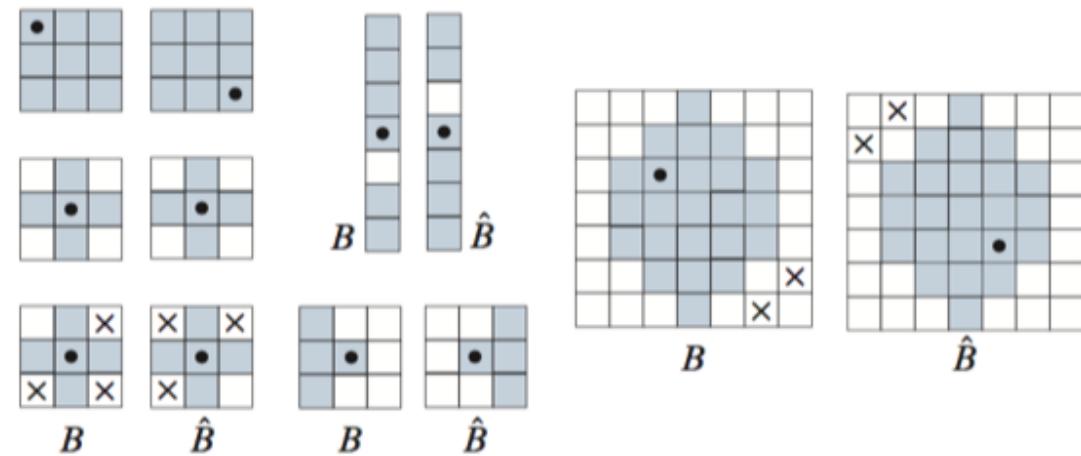
**Translation**  $(B)_z = \{c | c = b + z, \text{for } b \in B\}$

*moving the structuring elements by  $(z_1, z_2)$*



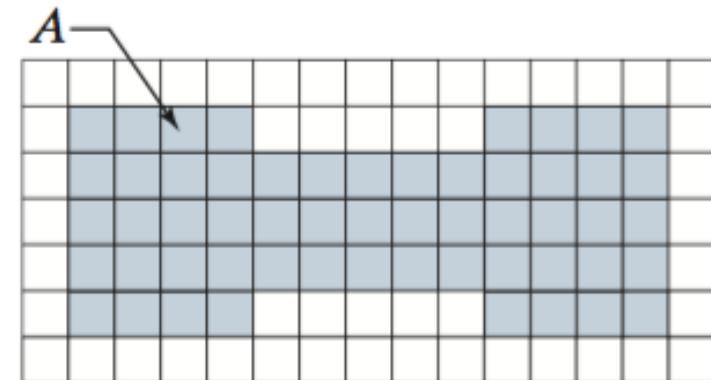
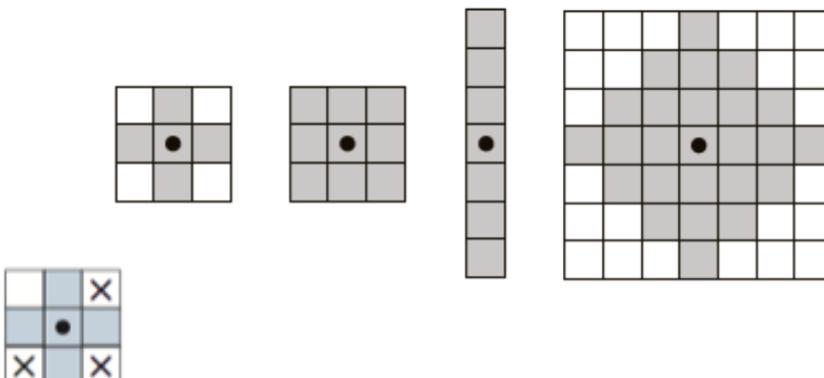
**FIGURE 9.2**

Structuring elements and their reflections about the origin (the x's are don't care elements, and the dots denote the origin). Reflection is rotation by 180° of an SE about its origin.



# Structural elements

- The input data for the mathematical morphology are the two images: the image/object and a **structuring element (SE)**.
- Typically, a structuring element is much smaller than the object to be processed. The shape of the structured element can be arbitrary, as long as it can be represented as a binary image of a given size.



**Image I**

**Object**

# Dilation

Dilation of set **A** by a structure element **B** is represented by  $A \oplus B$

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

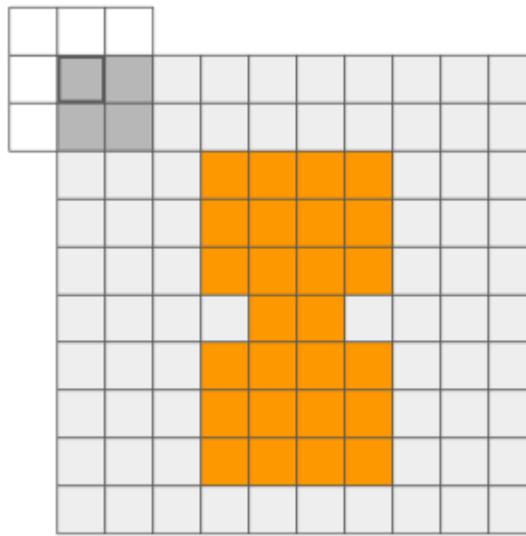
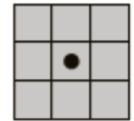
empty set

$C = A \oplus B$  is composed of all pixels that when  $\hat{B}$  shifts its origin to these pixels, at least one pixel in  $\hat{B}$  is in C

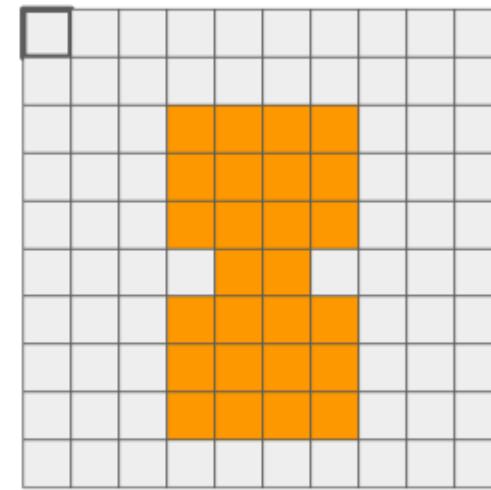
At the location of  $A \subset A \oplus B$ , the origin of  $\hat{B}$  takes the value 1

# Dilation

SE

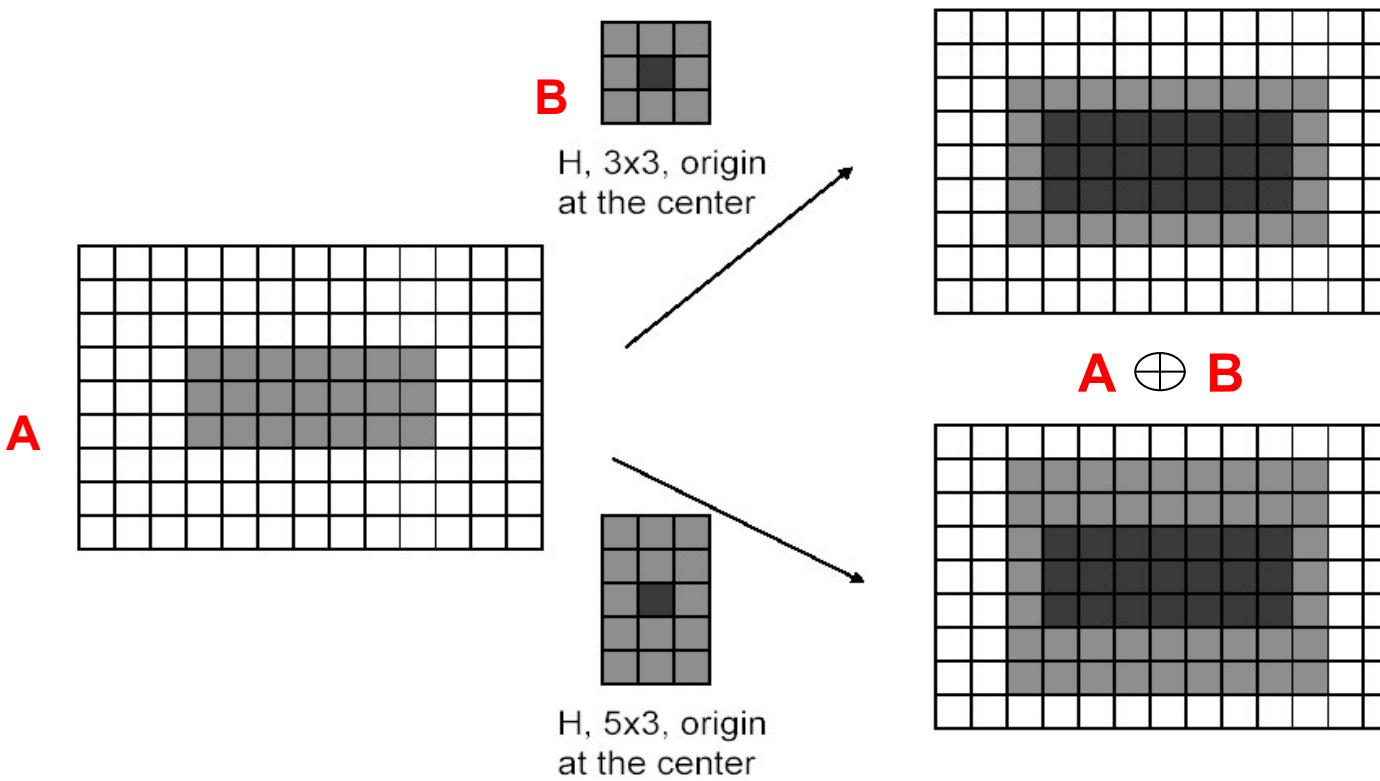


Object



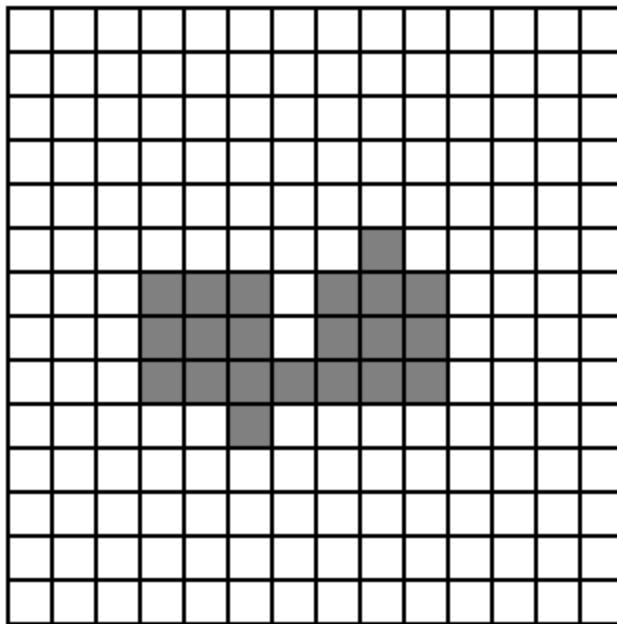
Result

# Dilation Example

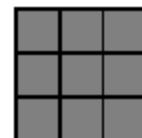


The dilation of A by B is the set of all displacements such that they overlap by at least one element.

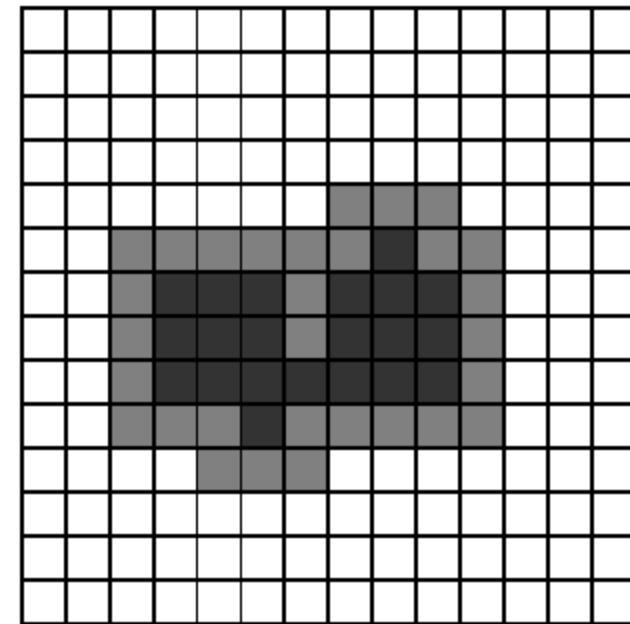
# Another Example of Dilation



A



B , 3x3, origin at the center



C

# Example of Dilation

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.



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

a      b      c

**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.

# Erosion

For **erosion**, the structural element passes through all pixels of the image. The pixel corresponds to the reference of the structure element is turned on if the entire structure element falls with foreground area (pixels).

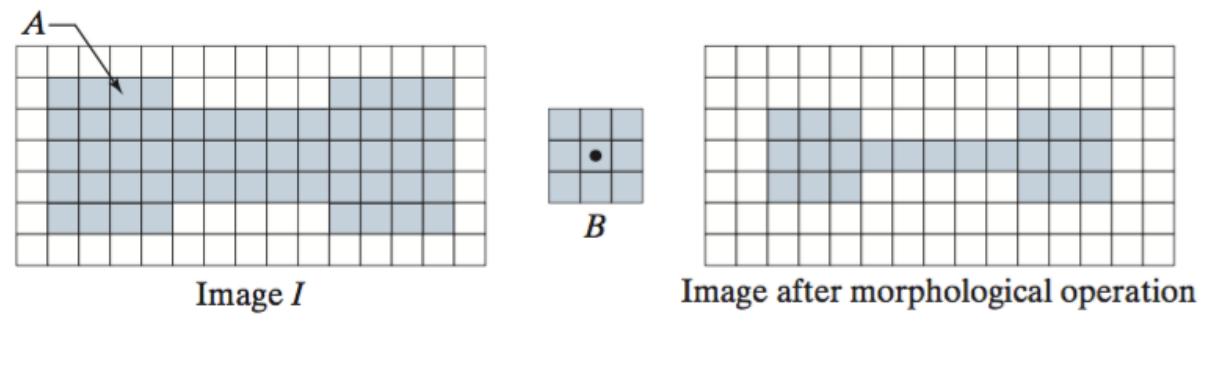
$$A \ominus B = \{z | (B)_z \subseteq A\}$$

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

a b c

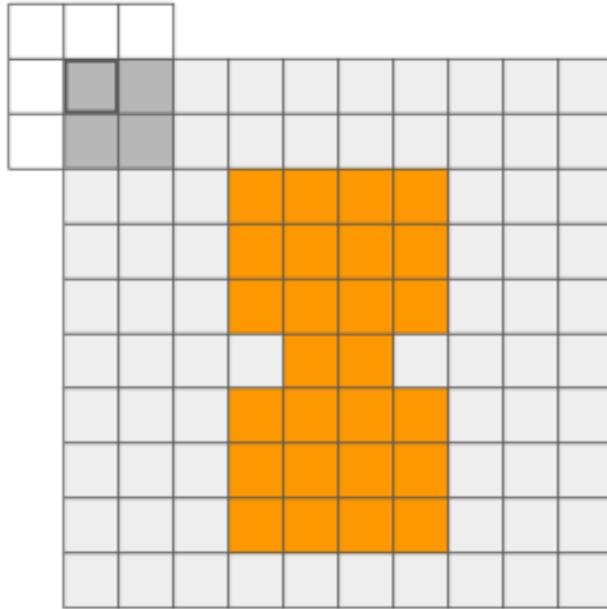
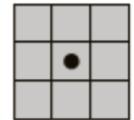
**FIGURE 9.3**

- (a) A binary image containing one object (set),  $A$ .
- (b) A structuring element,  $B$ .
- (c) Image resulting from a morphological operation (see text).

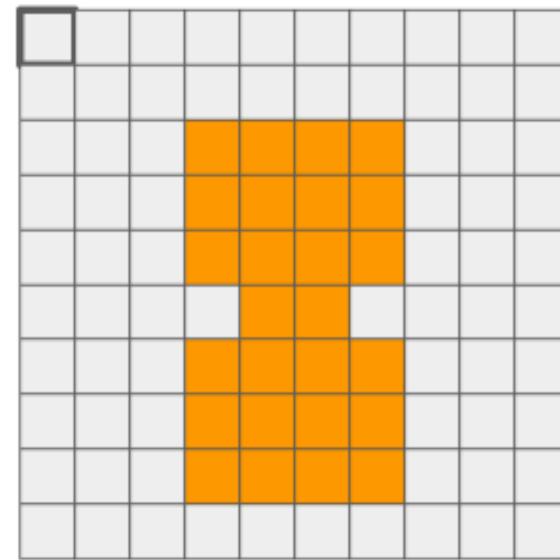


# Erosion

SE

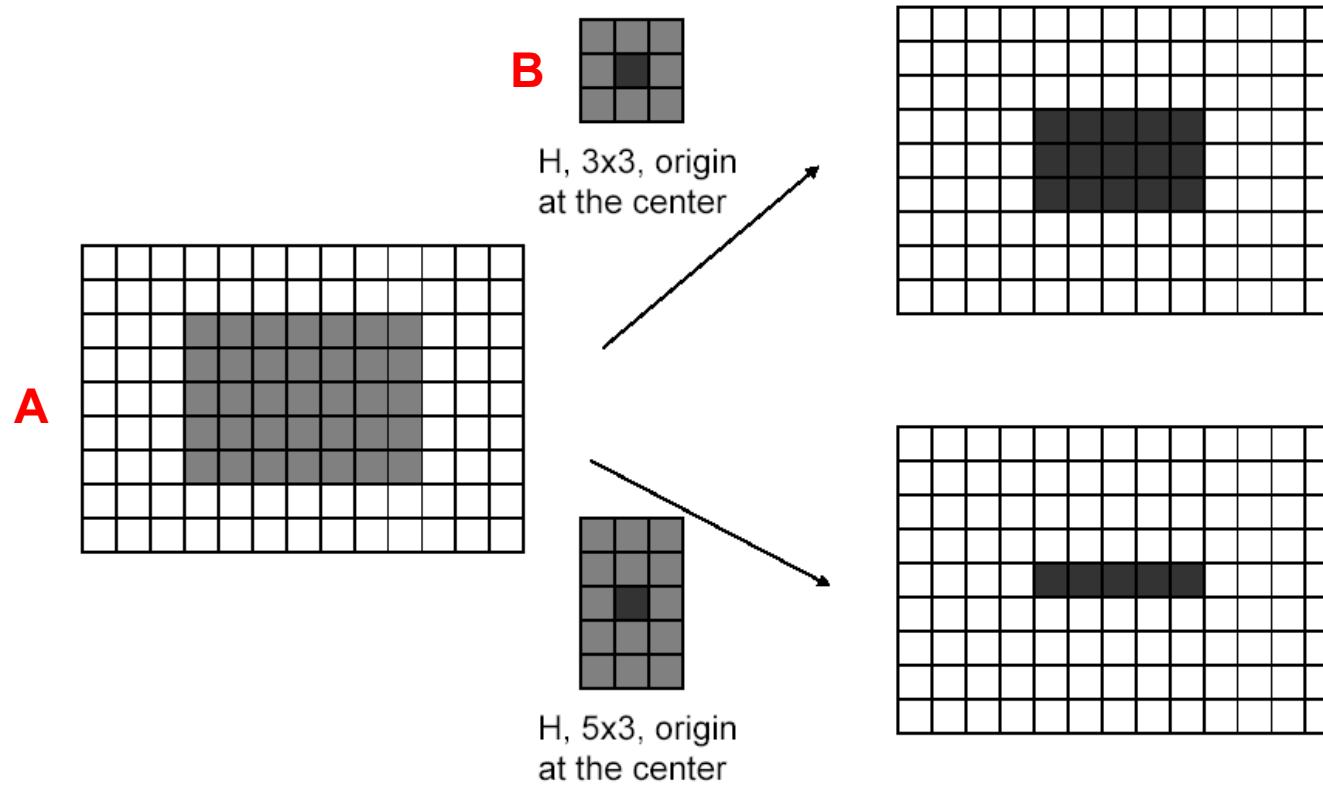


Object



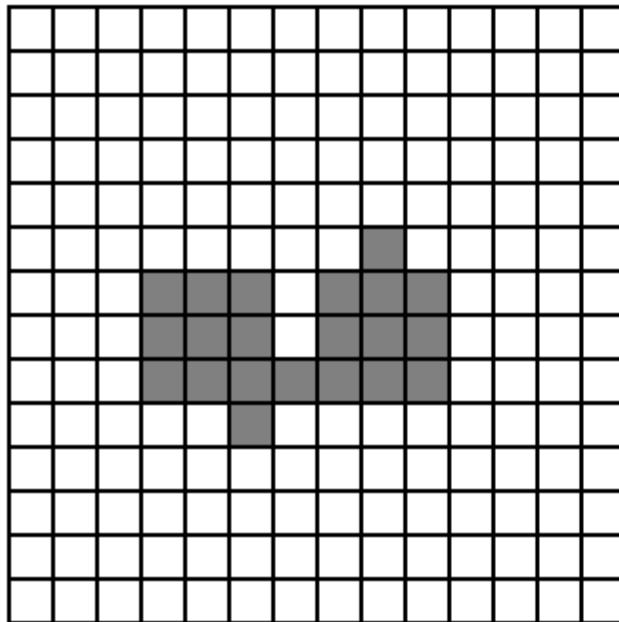
Result

# Erosion (Opposite of Dilation)

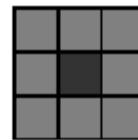


Passing B through A: at each location of origin of B, if B is completely contained in A, mark that location as a member of the new set (erosion set).

# Another Erosion Example

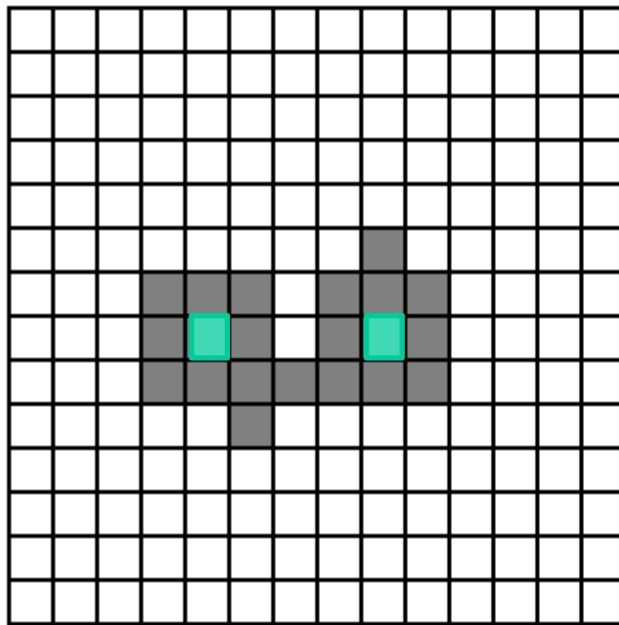


$F$



$H$ , 3x3, origin at the center

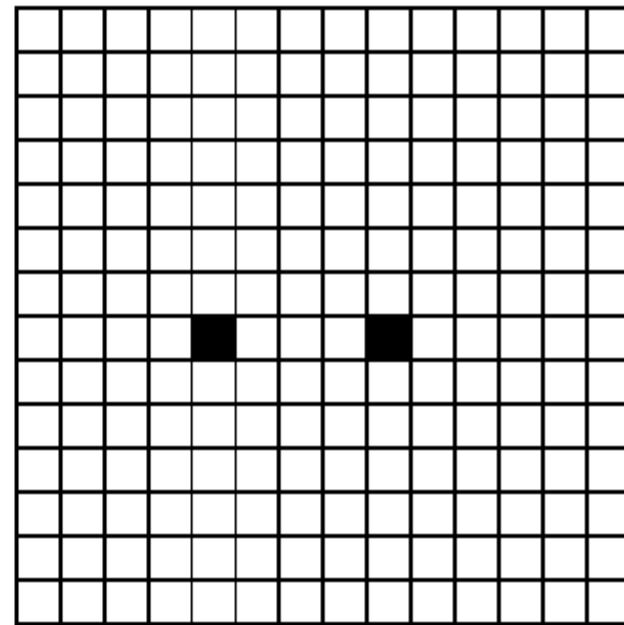
# Another Erosion Example



F

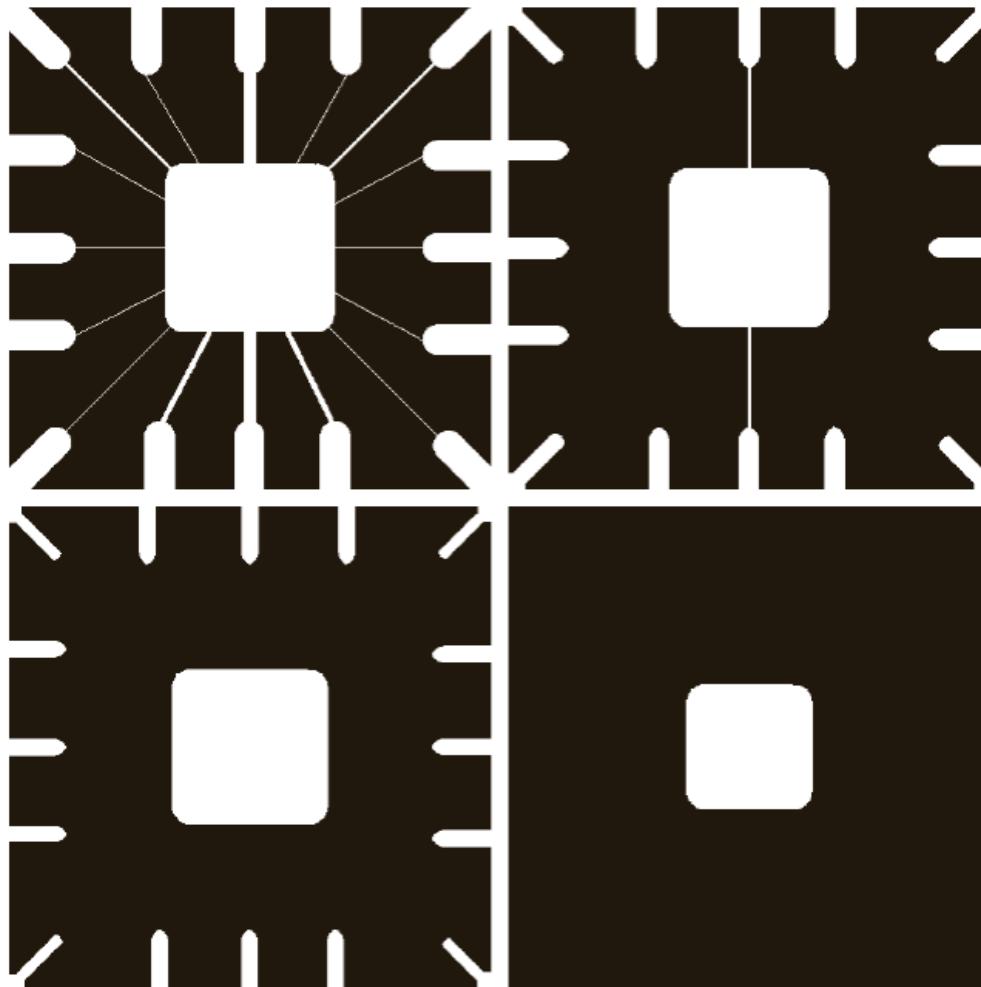


H, 3x3, origin at the center



G

# Erosion Example



a	b
c	d

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

# Closing and Opening

- Closing

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

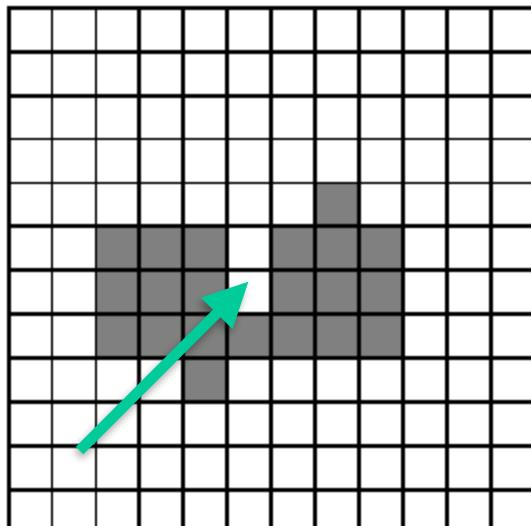
- Smooth the contour of an image
- Fill small gaps and holes

- Opening

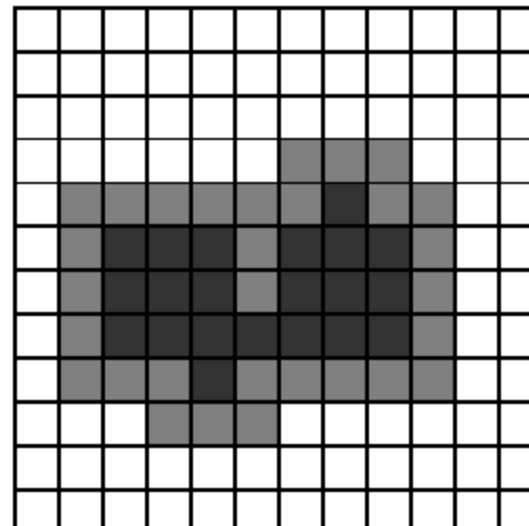
$$A \circ B = (A \ominus B) \oplus B$$

- Smooth the contour of an image
- Eliminate false touching, thin ridges and branches

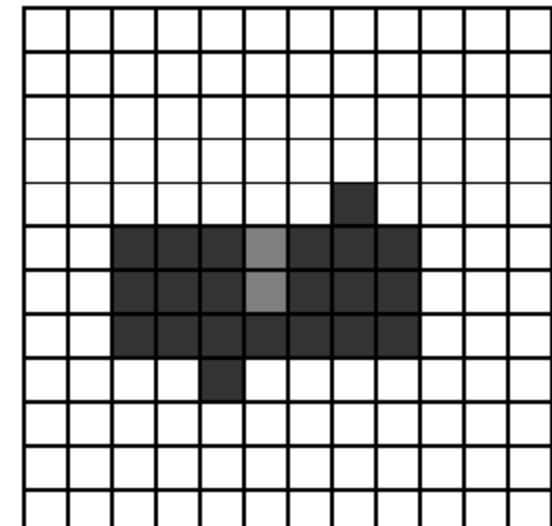
# Closing Example



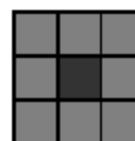
$F$



$F \oplus H$

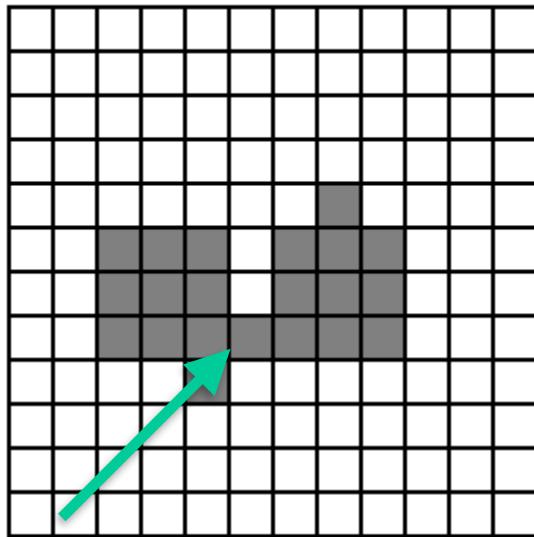


$(F \oplus H) \Theta H$

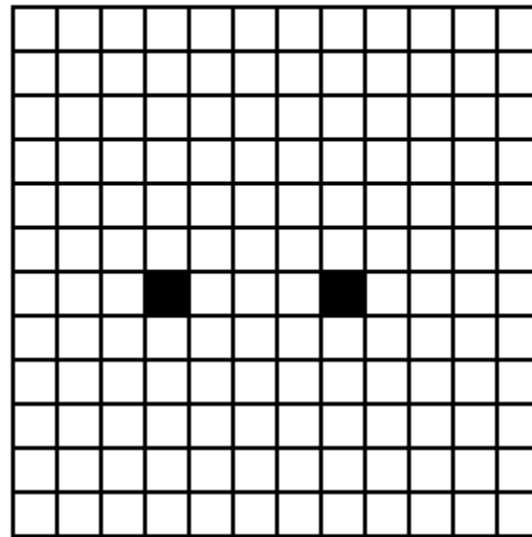
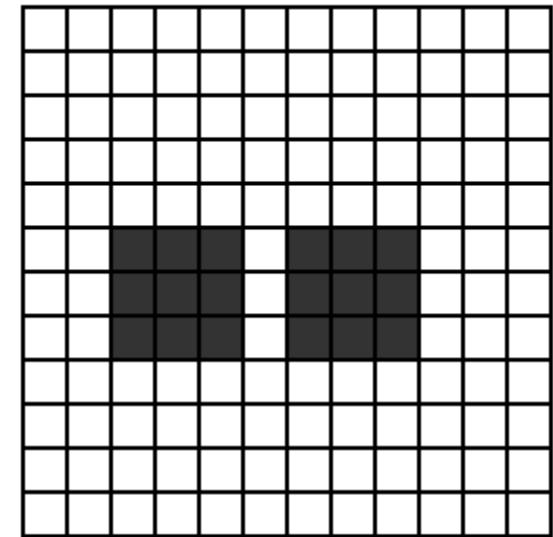
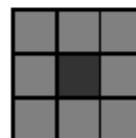


$H$ , 3x3, origin at the center

# Opening Example

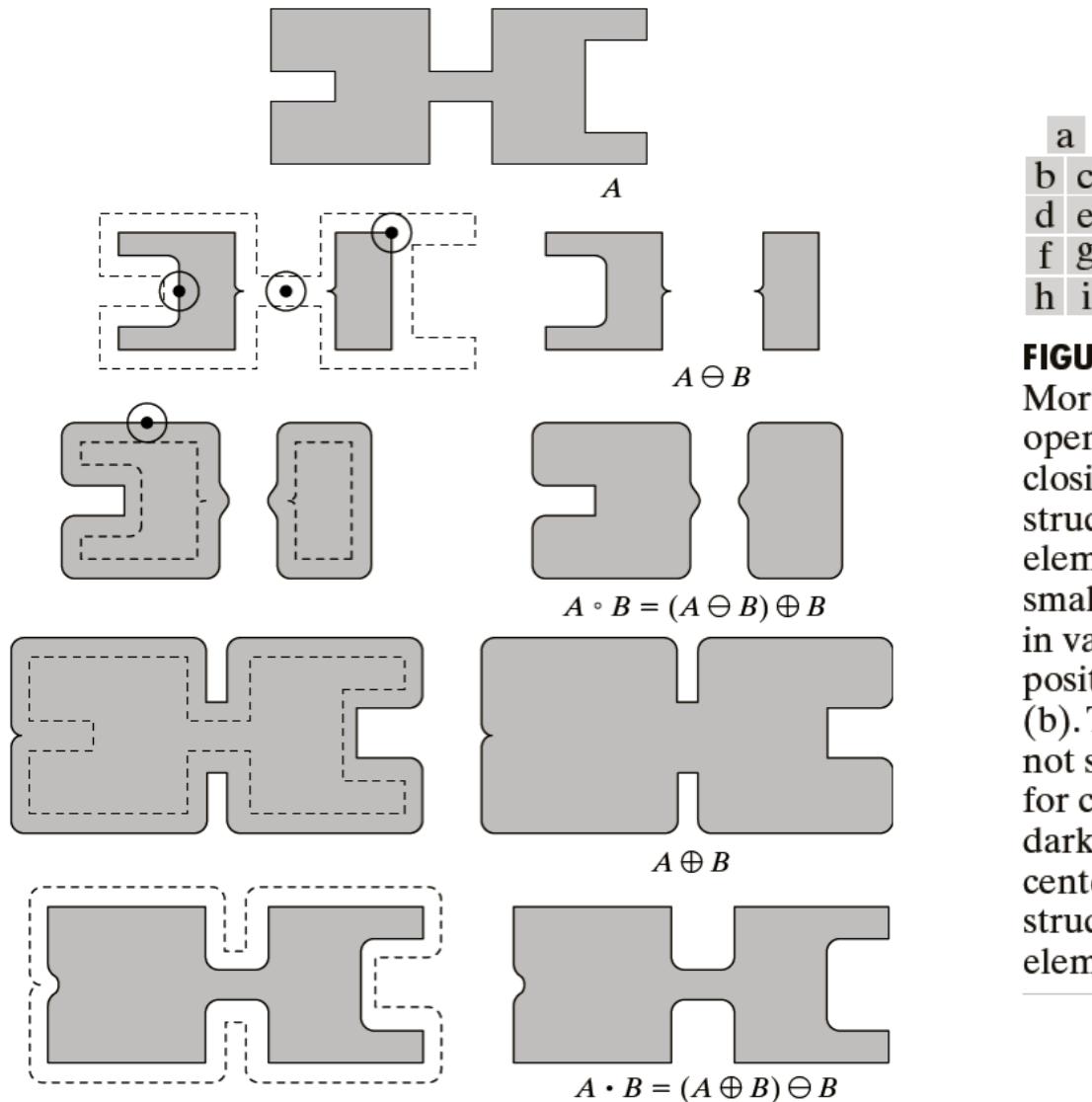


F

 $F \Theta H$  $(F \Theta H) \oplus H$ 

H, 3x3, origin at the center

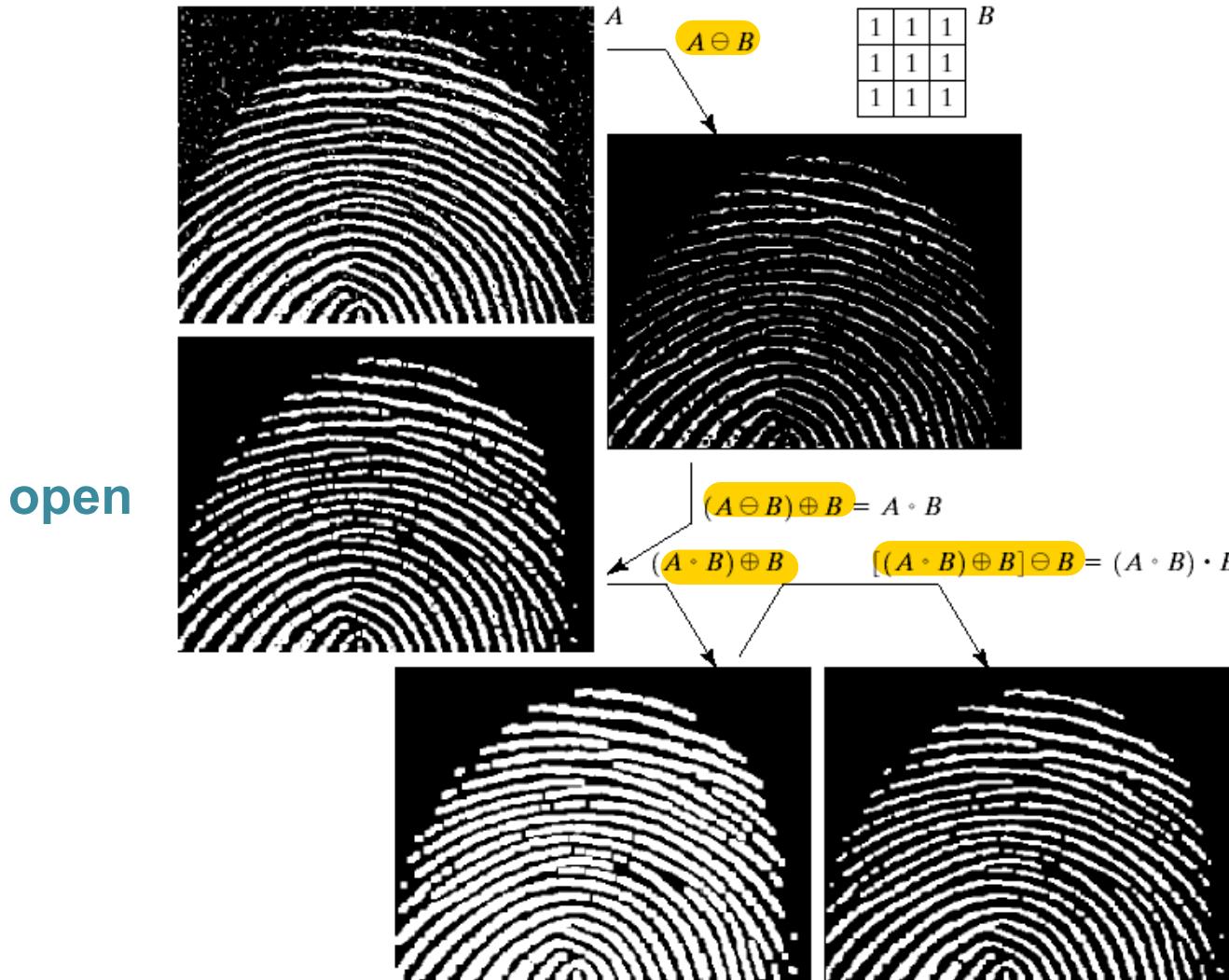
# Opening & Closing Example



a
b
c
d
e
f
g
h
i

**FIGURE 9.10**  
Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.

# Morphological Filtering Example: Opening followed by Closing



a	b
d	c
e	f

**FIGURE 9.11**

- (a) Noisy image.
- (c) Eroded image.
- (d) Opening of A.
- (d) Dilation of the opening.
- (e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

closing

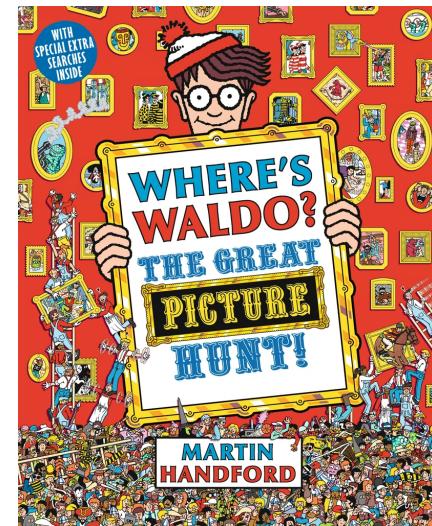


Take a break!

# Hit-or-Miss Transform

- **Hit-or-Miss transform (HMT)** is a basic tool for shape detection
- HMT uses two structuring elements:  $B_1$  for detecting shapes in the foreground, and  $B_2$  for detecting shapes in the background
- The word “miss” in the HMT arises from the fact that  $B_2$  finding a match in  $A^c$  is the same as  $B_2$  missing a match in  $A$ .

$$\begin{aligned} I \circledast B_{1,2} &= \left\{ z \mid (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c \right\} \\ &= (A \ominus B_1) \cap (A^c \ominus B_2) \end{aligned}$$



# Hit-or-Miss Transform

a  
b  
c  
d  
e  
f

FIGURE 9.12

(a) Image consisting of a foreground (1's) equal to the union,  $A$ , of set of objects, and a background of 0's.

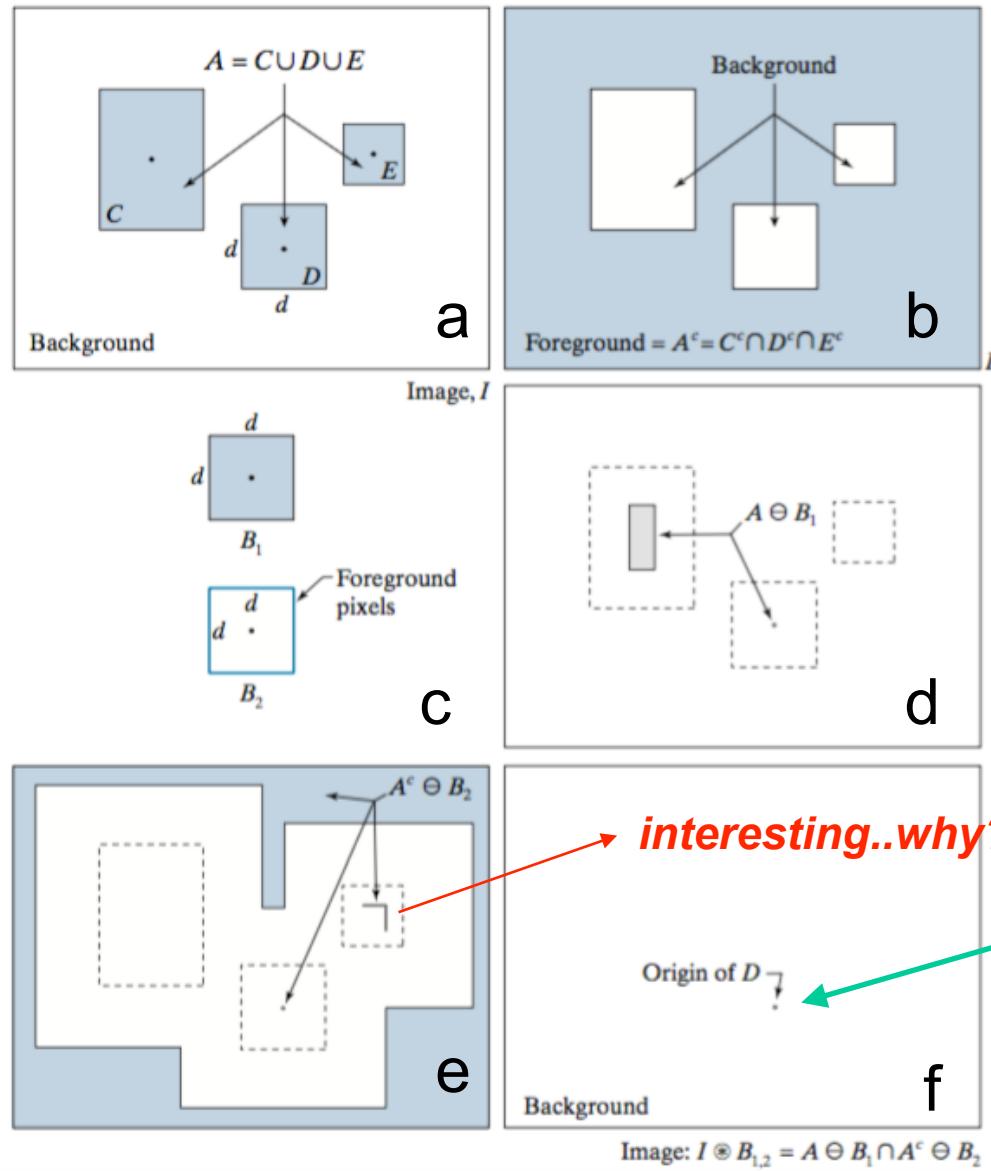
(b) Image with its foreground defined as  $A^c$ .

(c) Structuring elements designed to detect object  $D$ .

(d) Erosion of  $A$  by  $B_1$ .

(e) Erosion of  $A^c$  by  $B_2$ .

(f) Intersection of (d) and (e), showing the location of the origin of  $D$ , as desired. The dots indicate the origin of their respective components. Each dot is a single pixel.



Several shapes



→ *interesting..why?*

Location of the target shape

# Hit-or-Miss Transform

08 - Mathematical morphology

Part 2 - implementing a hit-or-miss transform

```
In [1]: import imageio
import numpy as np
from skimage import morphology
import matplotlib.pyplot as plt
```

Hit-or-miss transform

It is a type of shape-detection / pattern matching algorithm

- an erosion can be seen as a *hit* or a matching
- but only erosion cannot guarantee that the hit is a disjunct object
- so we use local background information to do that so

The hit-or-miss of an image  $A$  with a structuring element  $S$  is:

$$(A \ominus S) \cap (A^C \ominus (W - S))$$

where  $A^C$  is complement of  $A$  and  $W$  is a window that can be obtained by the dilation of  $S$

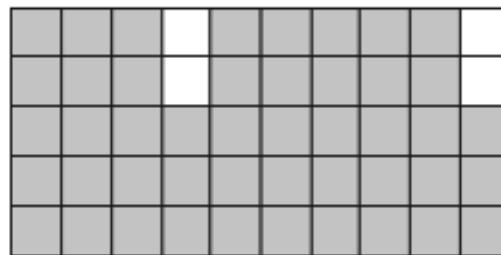
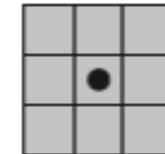
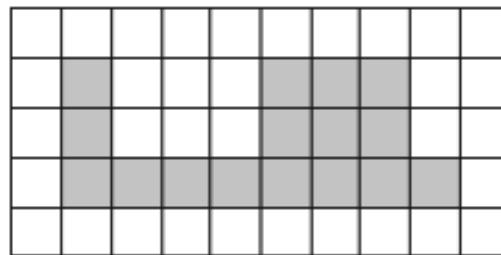
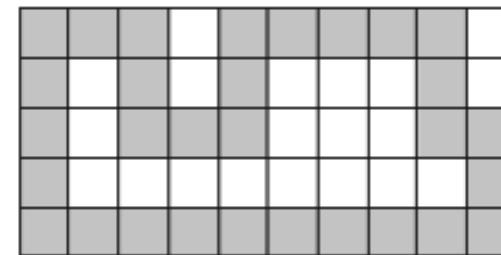
```
In [1]: S = np.zeros([7,7])
S[3, 1:6] = 1
S[1:6, 0:4] = 1
S
```

DIP 08 - Mathematical Morphology (4) - implementing the hit-or-miss in python

[https://www.youtube.com/watch?v=P8NwA\\_p07a8](https://www.youtube.com/watch?v=P8NwA_p07a8)

Question: How is hit-or-miss transform differ from the template matching using cross-correlation ?

# Boundary Extraction

 $A$  $B$ **erosion**  $A \ominus B$  $\beta(A)$ 

a	b
c	d

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

**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.

# Boundary Extraction



a b

**FIGURE 9.14**

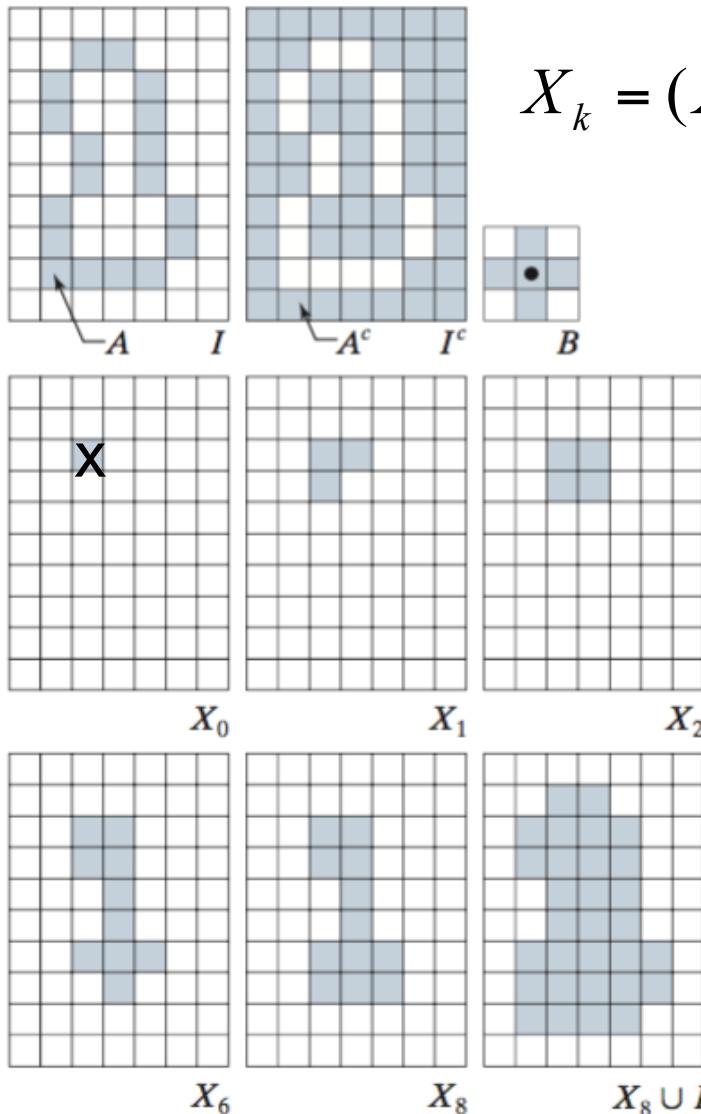
(a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

# Hole Filling

a	b	c
d	e	f
g	h	i

**FIGURE 9.17**

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

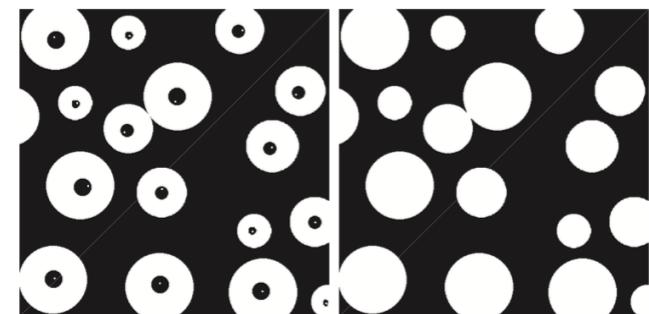


$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$

dilation

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

**Conditional dilation:** constrain the dilation within the hole by checking with the complement of  $A$  at each operation



# Extraction of Connected Components

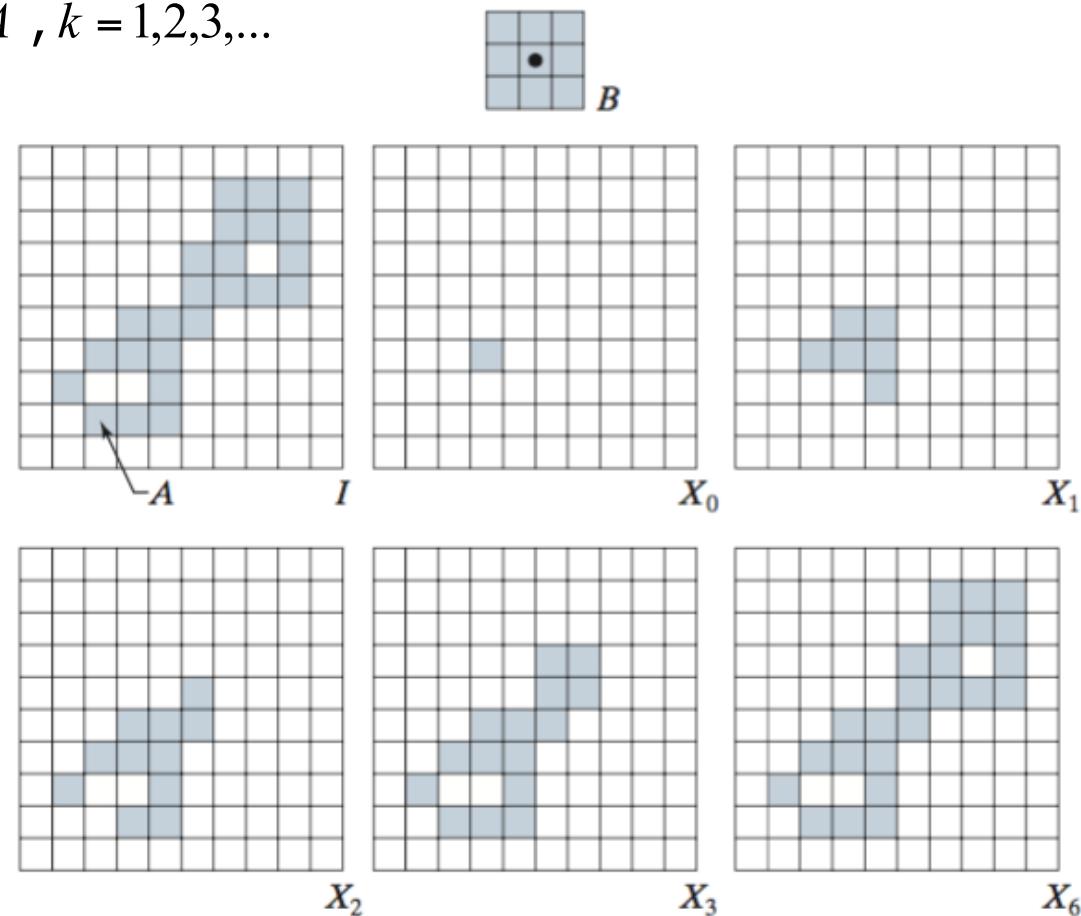
Finding pixels that are connected: similar to hole-filling, but checking with A itself at each step, instead of the complement of A

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

a
b
c
d
e
f
g

**FIGURE 9.19**

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

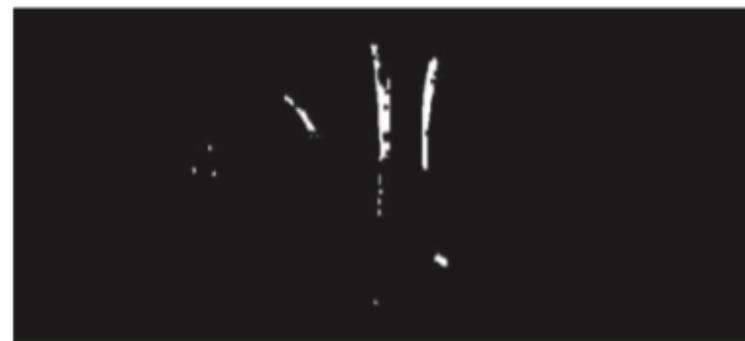


# Extraction of Connected Components

a  
b  
c d

**FIGURE 9.20**

- (a) X-ray image of a chicken fillet with bone fragments.
- (b) Thresholded image (shown as the negative for clarity).
- (c) Image eroded with a  $5 \times 5$  SE of 1's.
- (d) Number of pixels in the connected components of (c). (Image (a) courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, [www.ntbxray.com.](http://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

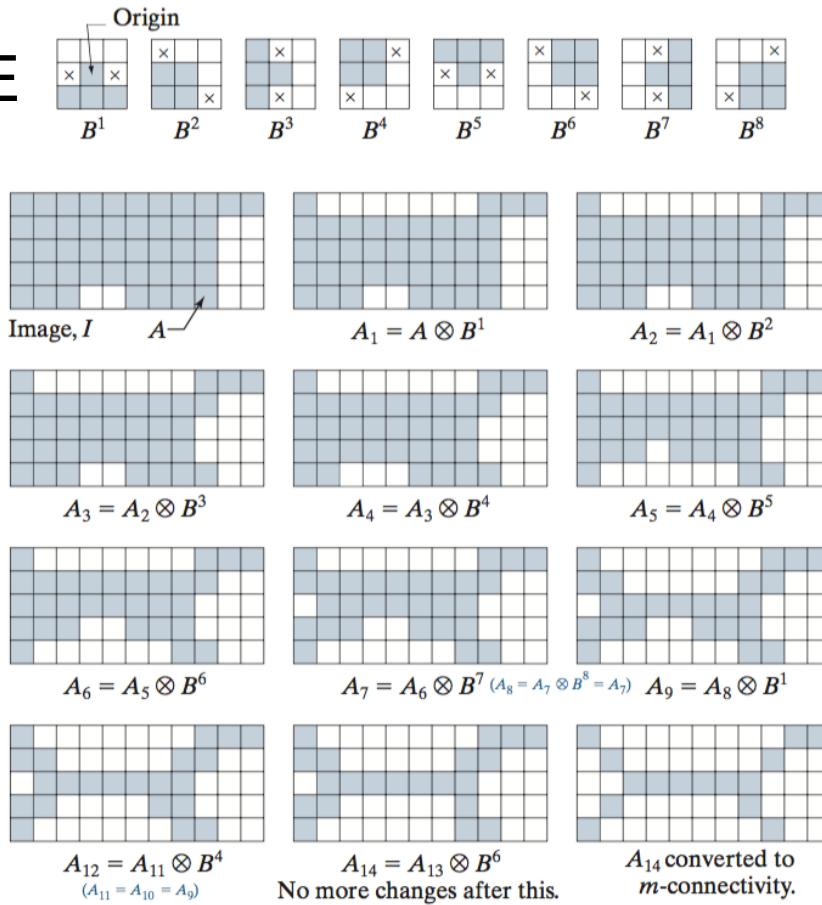
# Thinning

**SE**

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

FIGURE 9.23

- (a) Structuring elements.
- (b) Set  $A$ .
- (c) Result of thinning  $A$  with  $B^1$  (shaded).
- (d) Result of thinning  $A_1$  with  $B_2$ .
- (e)–(i) Results of thinning with the next six SEs. (There was no change between  $A_7$  and  $A_8$ .)
- (j)–(k) Result of using the first four elements again.
- (l) Result after convergence.
- (m) Result converted to  $m$ -connectivity.



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

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

$$A \otimes \{B\} = \left( \left( \dots \left( (A \otimes B^1) \otimes B^2 \right) \dots \right) \otimes B^n \right)$$

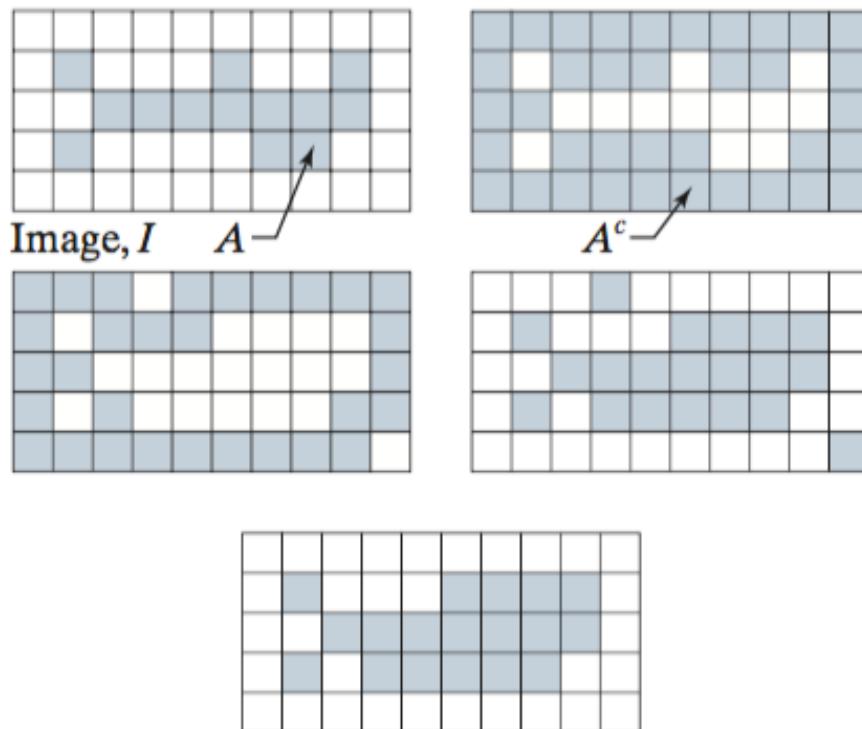
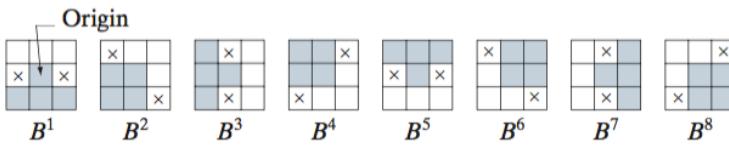
$$\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$$

*Shrink objects*

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.

# Thickening



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

$$A \odot \{B\} = \left( \left( \dots \left( (A \odot B^1) \odot B^2 \right) \dots \right) \odot B^n \right)$$

a	b
c	d
e	

**FIGURE 9.24**

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

same kernels as  
thinning but with  
1's and 0's  
switched

*Differences? thinning vs. erosion ; thickening vs. dilation*

# Skeletonization

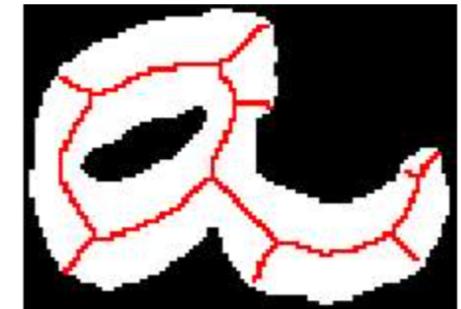
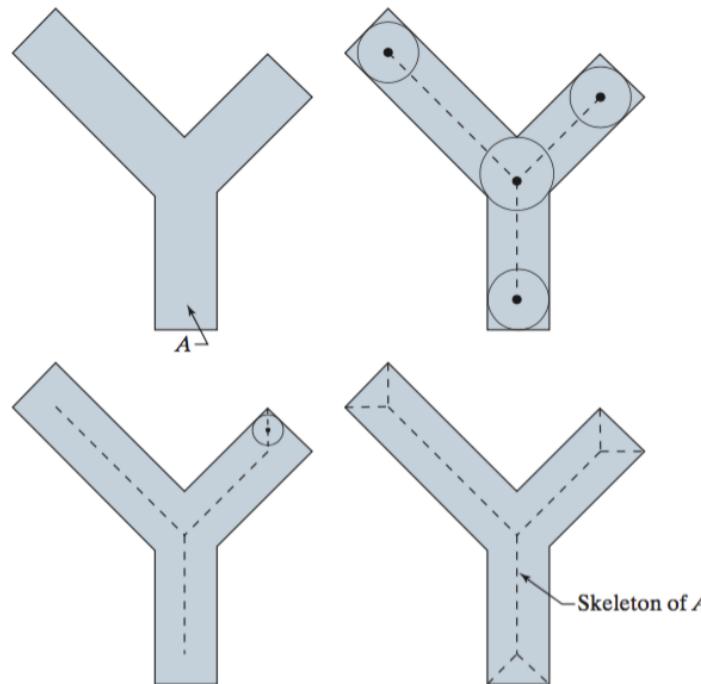
- Compact or minimal representation of objects in an image while retaining homotopy of the image (the object can be reconstructed from the skeleton)
- Is the best of all points that are equally distanced from two closest points of the object boundary
- Equivalently, the union of all maximal disk centres that are contained in the object

Analogy: starting a fire at the boundary of an object, letting it burn inward

a	b
c	d

FIGURE 9.25

- (a) Set  $A$ .  
 (b) Various positions of maximum disks whose centers partially define the skeleton of  $A$ .  
 (c) Another maximum disk, whose center defines a different segment of the skeleton of  $A$ .  
 (d) Complete skeleton (dashed).



# Skeletonization

Skeletonization algorithm

$\text{Input}(A, B); A : \text{the given set}, B : \text{structuring element for } S(A)$

for  $k = 0, 1, 2, \dots$

do  $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$ ;  $\ominus = \text{erosion}, \circ = \text{opening}$

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

Reconstruction algorithm

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

$k = k$  successive operations

A for loop is used

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

$k \setminus$	$A \ominus kB$	$(A \ominus kB) \circ B$	$S_k(A)$	$\bigcup_{k=0}^K S_k(A)$	$S_k(A) \oplus kB$	$\bigcup_{k=0}^K S_k(A) \oplus kB$
0						
1						
2						

FIGURE 9.26

Implementation of Eqs. (9-28) through (9-33). 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.

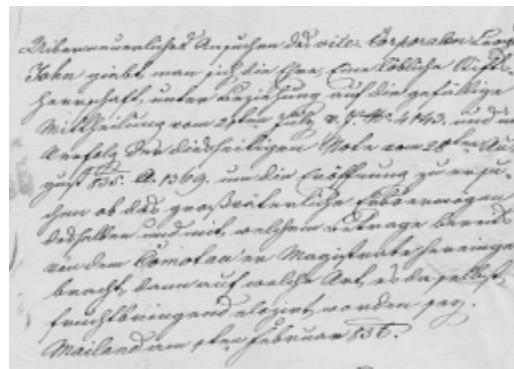


SE

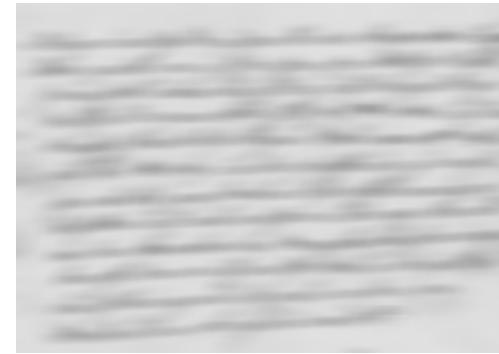


# Morphological Filtering Example: Handwritten Document Processing

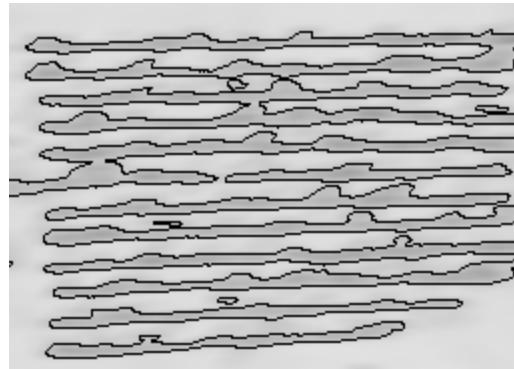
a



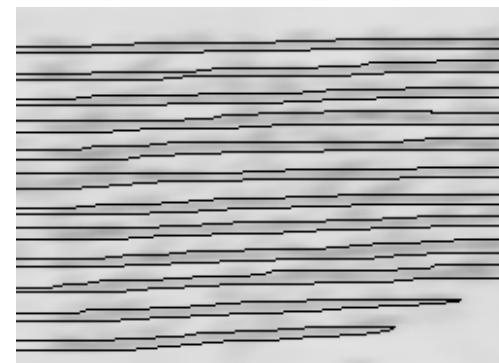
b



c

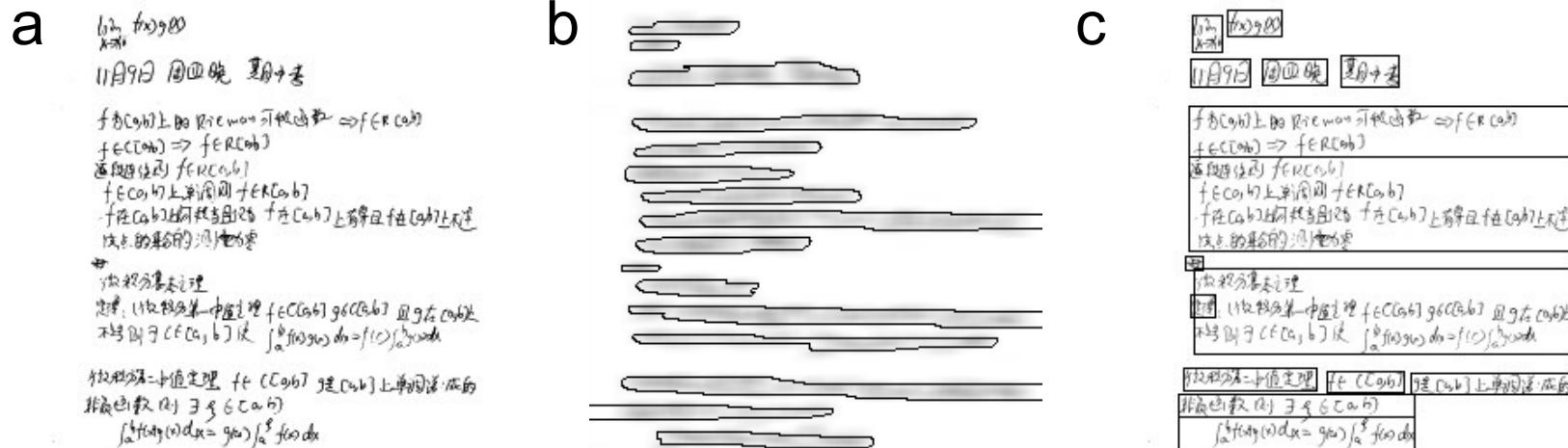


d



Process of text line segmentation. a: original image; b: blurred image; c: segmented image; d: final segmentation result after morphological processing.

# Morphological Filtering Example: Handwritten Document Processing



Process of text line segmentation. a: original image;  
 b: segmentation result based on morphological processing;  
 c: segmentation result based on connected component analysis.

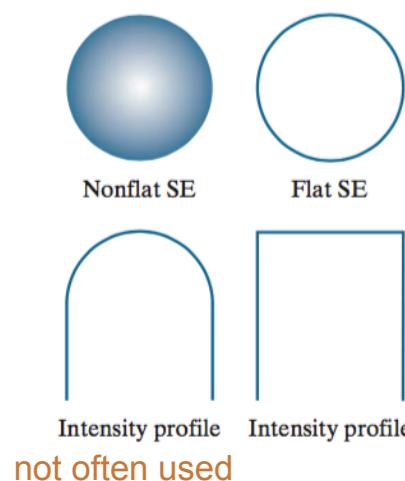
# Grayscale Morphology

Structuring elements in grayscale morphology perform the same basic functions as their binary counterparts

Structuring elements in grayscale morphology belong to one of two categories: **non-flat** and **flat**

a b  
c d

**FIGURE 9.36**  
Nonflat and flat structuring elements, and corresponding horizontal intensity profiles through their centers. All examples in this section are based on flat SEs.



## Grayscale Morphology

The *grayscale erosion* of  $f$  by a flat structuring element  $b$  at location  $(x, y)$  is defined as the *minimum* value of the image in the region coincident with  $b(x, y)$  when the origin of  $b$  is at  $(x, y)$ . In equation form, the **erosion** at  $(x, y)$  of an image  $f$  by a structuring element  $b$  is given as

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

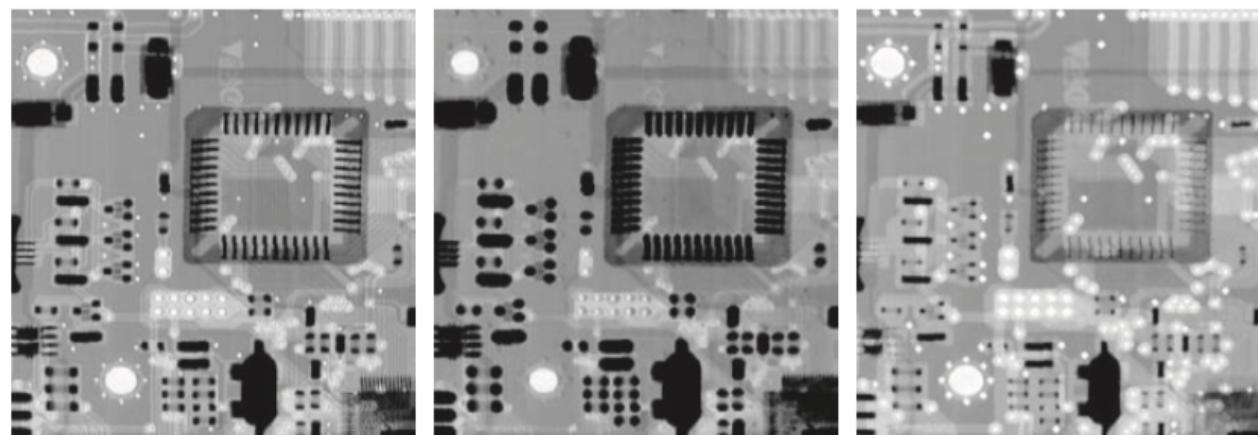
The **dilation** at  $(x, y)$  of an image  $f$  by a structuring element  $b$  is given as

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

a b c

**FIGURE 9.37**

(a) Gray-scale X-ray image of size  $448 \times 425$  pixels. (b) Erosion using a flat disk SE with a radius of 2 pixels. (c) Dilation using the same SE. (Original image courtesy of Lixi, Inc.)



# Summary

- Set operations, structuring elements, objects
- Dilation, erosion -> closing, opening
- Hit-or-miss transform, boundary extraction, hole filling, connected component extraction
- Thinning, thickening, skeletonization
- Simple grayscale morphology: flat SE erosion and dilation

## Reading materials

- Chapter 9
- Page 635-663, 674-677

## Questions

- How does "hit-or-miss" transform differ from cross-correlation in terms of locating an object in an image using image kernels?
- What are the differences between thinning and erosion?
- What are the differences between thickening and dilation?
- To smooth out a foreground object, what type of morphological operation can be used?
- Page 25 of the slide: explain why there is a "corner" shape left in the image after erosion?
- What are the ways to spot a disk shape of known size in an image?