CSE4227 Digital Image Processing

Chapter 9 – **Morphological Image Processing- (Part-II)**

Dr. Kazi A Kalpoma

Professor, Department of CSE

Ahsanullah University of Science & Technology (AUST)

Contact: kalpoma@aust.edu

Google Class code: bux3jc2

Today's Contents

- **Morphology: pixel shape based analysis**
  - **Hit-and-Miss Transformation**
  - **Thinning, and**
  - **Thickening**

Chapter 9 from R.C. Gonzalez and R.E. Woods, Digital Image Processing (3rd Edition), Prentice Hall, 2008 [**Section 9.4, 9.5.1, 9.5.5, 9.5.6**]

http://www.imagemagick.org/Usage/morphology/#hmt

Review of last lecture...........

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

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

Dilation and erosion are duals of each other:

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

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

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

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

Hit-and-Miss Transform ⊛

- Used to look for particular patterns of foreground and background pixels
- Very simple object recognition
- Example for a Hit-and-miss Structuring Element: Contains 0s, 1s and don’t care’s.

  - '1' meaning 'foreground
  - '0' meaning 'background'
  - 'Nan' or 'I Don't Care or 'Any Pixel

	1	
0	1	1
0	0	

- Similar to Pattern Matching:

Hit-and-miss Transform

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

Hit-or-Miss Transform

- A tool for shape detection or for the detection of a *disjoint region* in an image
- Idea
  - Sup (cir
  - We particular pattern of bits. We will call this pattern “shape B”
  - We then search image A for shape B
  - Whenever there is a ‘hit’, we indicate where the center of shape B was on image A.

Watch out: We actually look for ‘*fits*’ but we will be calling them ‘*hits*’ when talking about hit-or-miss transform

Hit-or-Miss Transform

- Structuring Element

So far we have considered the SEs where 0s are treated as Don't Cares i.e. we focus on the 1s only

	1	
1	1	1
	1	

0	1	0
1	1	1
0	1	0

Hit-or-Miss Transform

- Extended Structuring Element

Now we will distinguish between the 0s and the Don't cares

1	1	1
×	0	×
×	0	×

E.g. For a 'fit' the 0s of SE should match with 0s of the underlying image

Hit and miss structuring elements

0	0	0
don't care	1	don't care
1	1	1

Structuring element

- The structuring element has pixels with 3 values
  - 0: corresponding pixel must be 0
  - 1: corresponding pixel must be 1
  - don't care

Input pixels

0	0	0
1	1	0
1	1	1

= true

0	0	1
1	1	0
1	1	1

= false

Hit-or-Miss Transform

- Extended Structuring Element: Example

0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0
0	1	0	0	1	1	1	1
0	1	1	1	1	1	0	1
0	0	1	1	1	1	0	1

$\ominus$

1	1	1
×	0	×
×	0	×

Erosion Recap: Slide the SE on the image and look for the ‘fits’

Hit-or-Miss Transform

- Extended Structuring Element: Example

0	0	0	0	0	0	0	0
0	0	**1**	**1**	**1**	0	0	0
0	0	0	**0**	1	1	1	0
0	1	0	**0**	1	1	1	1
0	1	1	1	1	1	0	1
0	0	1	1	1	1	0	1

'Fit' encountered

$\ominus$

1	1	1
×	0	×
×	0	×

Hit-or-Miss Transform

- Extended Structuring Element: Example

0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0
0	1	0	0	1	**1**	**1**	**1**
0	1	1	1	1	1	**0**	1
0	0	1	1	1	1	**0**	1

$\ominus$

1	1	1
×	0	×
×	0	×

'Fit' encountered

Hit-or-Miss Transform

What we actually did???

- Extended Structuring Element: Example

0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0
0	1	0	0	1	1	1	1
0	1	1	1	1	1	0	1
0	0	1	1	1	1	0	1

$\ominus$

1	1	1
×	0	×
×	0	×

=

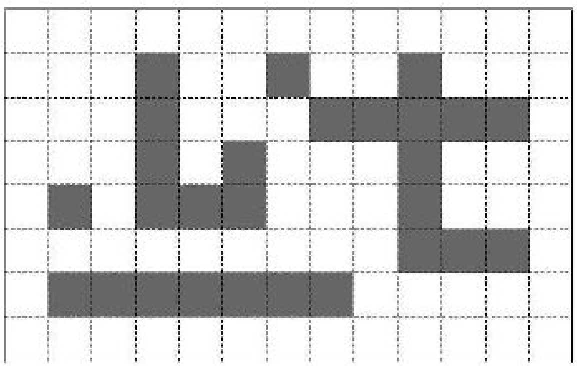
Output

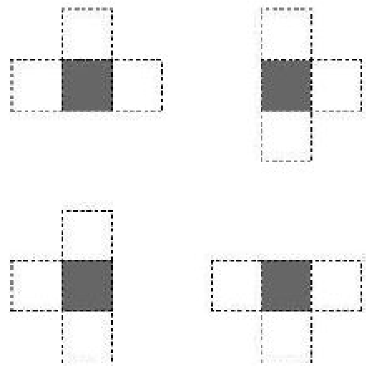
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	**1**	0
0	0	0	0	0	0	0	0

We have searched the pattern in the structuring element in the image

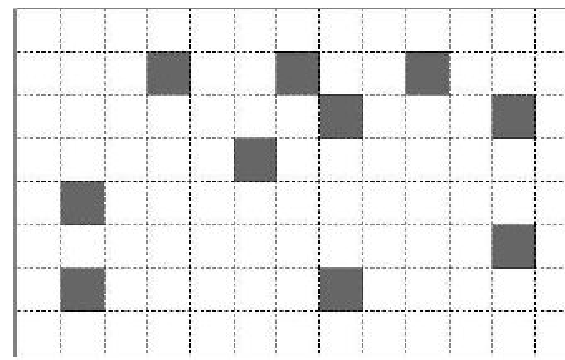
Hit-or-miss transform example

- Locating 4-connected endpoints





SEs for 4-connected endpoints



Resulting Hit-or-miss transform

Corner Detection with Hit-and-miss Transform

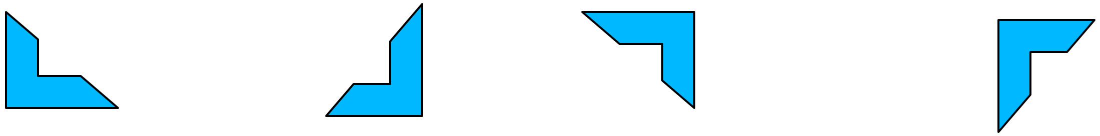
- Structuring Elements representing four corners

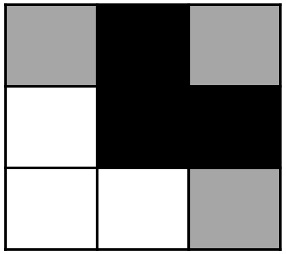
	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

0	0	
0	1	1
	1	





Hit-and-miss example: corner detection

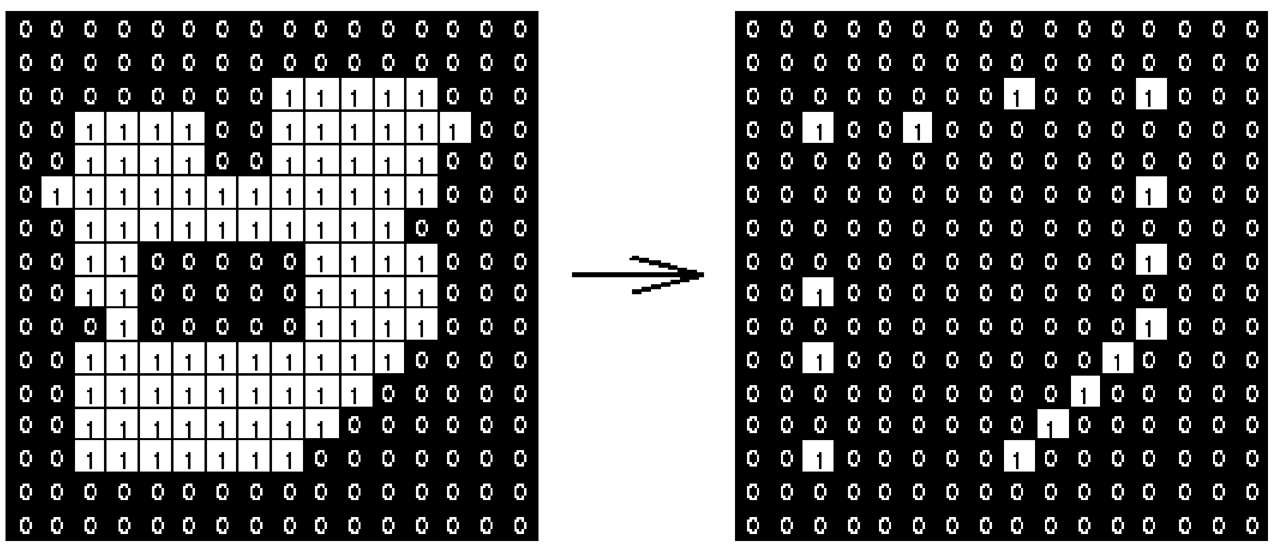
- Structuring Elements representing four corners.
- Apply each Structuring Element.
- Use OR operation to combine the four results.

	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

0	0	
0	1	1
	1	



Hit-and-Miss Transformation

- Alternative definition:

$$A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2)$$

- A background is necessary to detect disjoint sets.
- When we only aim to detect certain patterns within a set, a background is not required, and simple erosion is sufficient.

Some basic Morphological Algorithms

- When dealing with binary images, the principle application of morphology is extracting image components that are useful in the representation and description of shape

- Using the simple technique we have looked at so far we can begin to consider some more interesting morphological algorithms.

- We will look at:
  - Boundary extraction
  - Region filling

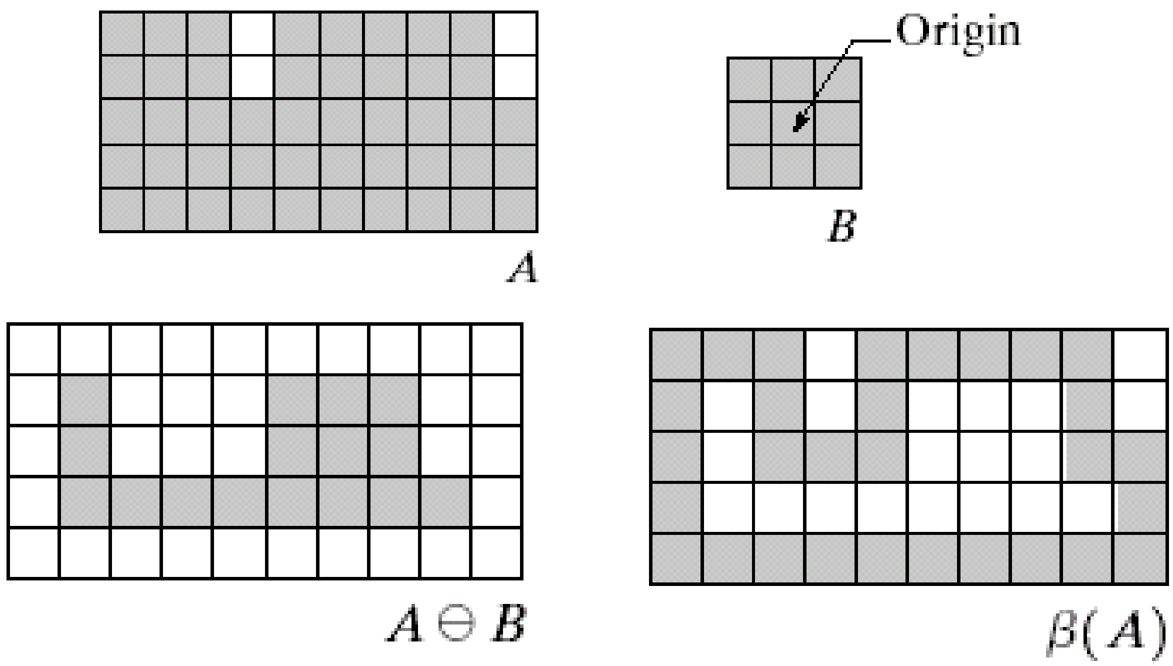
- There are lots of others as well though:
  - Extraction of connected components
  - Thinning/thickening
  - Skeleton

Boundary Extraction

Extracting the boundary (or outline) of an object is often extremely useful

The boundary can be given simply as

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



Boundary Extraction Example

A simple image and the result of performing boundary extraction using a square 3*3 structuring element

Original Image

Extracted Boundary

Home Work

- What is the smallest number of different structuring elements that you would need to use to locate all foreground points in an image which have at least one foreground neighbor, using the hit-and-miss transform? What do the structuring elements look like?

Thinning ($\otimes$) (Subtracting Pixels from a Shape)

1. Used to remove selected foreground pixels from binary images

1. After edge detection, lines are often thicker than one pixel.

1. Thinning can be used to thin those line to one pixel width.

Definition of Thinning

The thinning of a set A by a structuring element B:

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

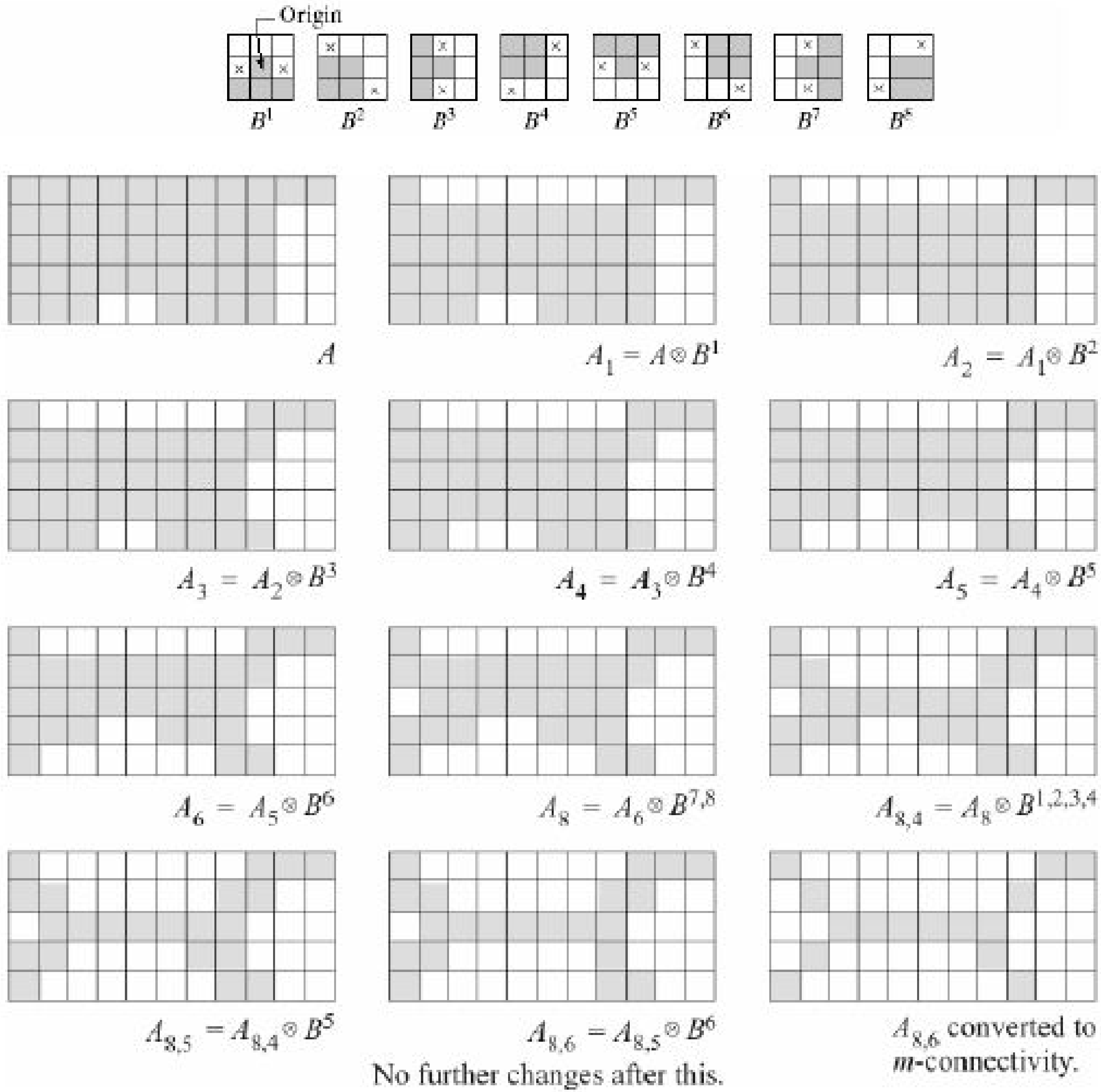
Symmetric thinning: sequence of structuring elements,

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

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

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

Thinning Illustration:



Thinning Illustration:

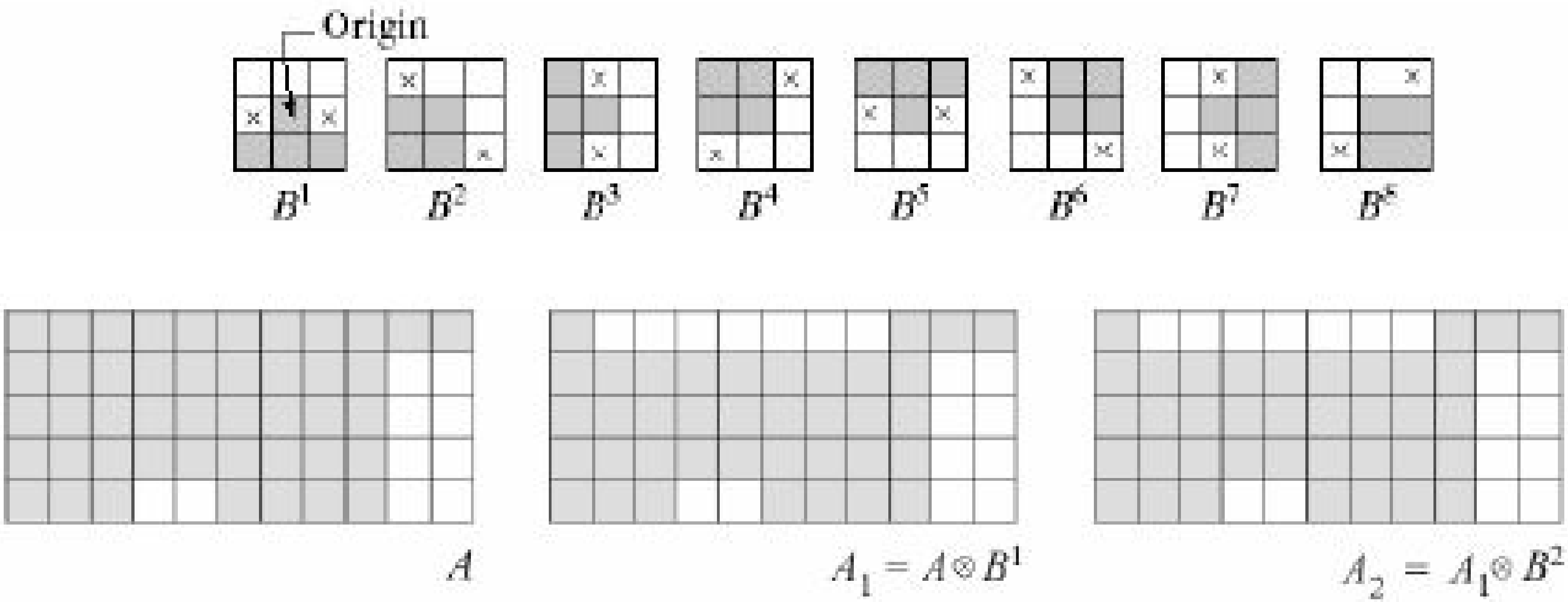


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there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

Thinning Illustration:

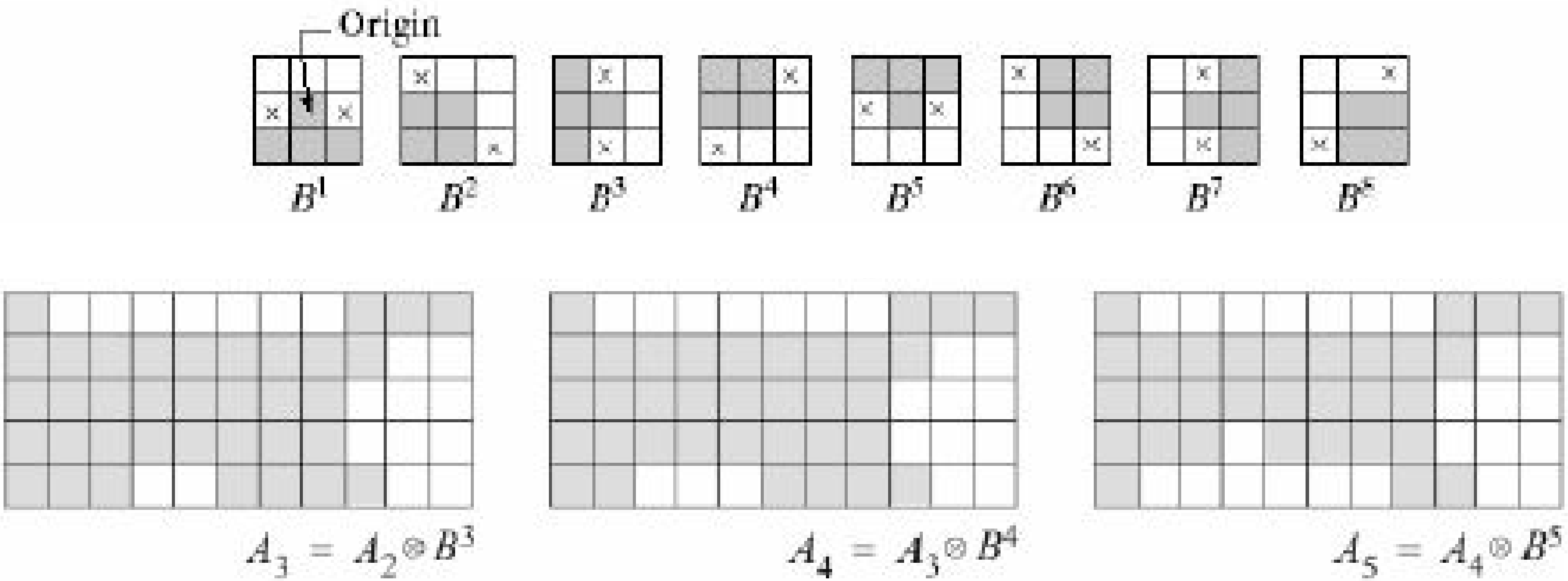


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there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

Thinning Illustration:

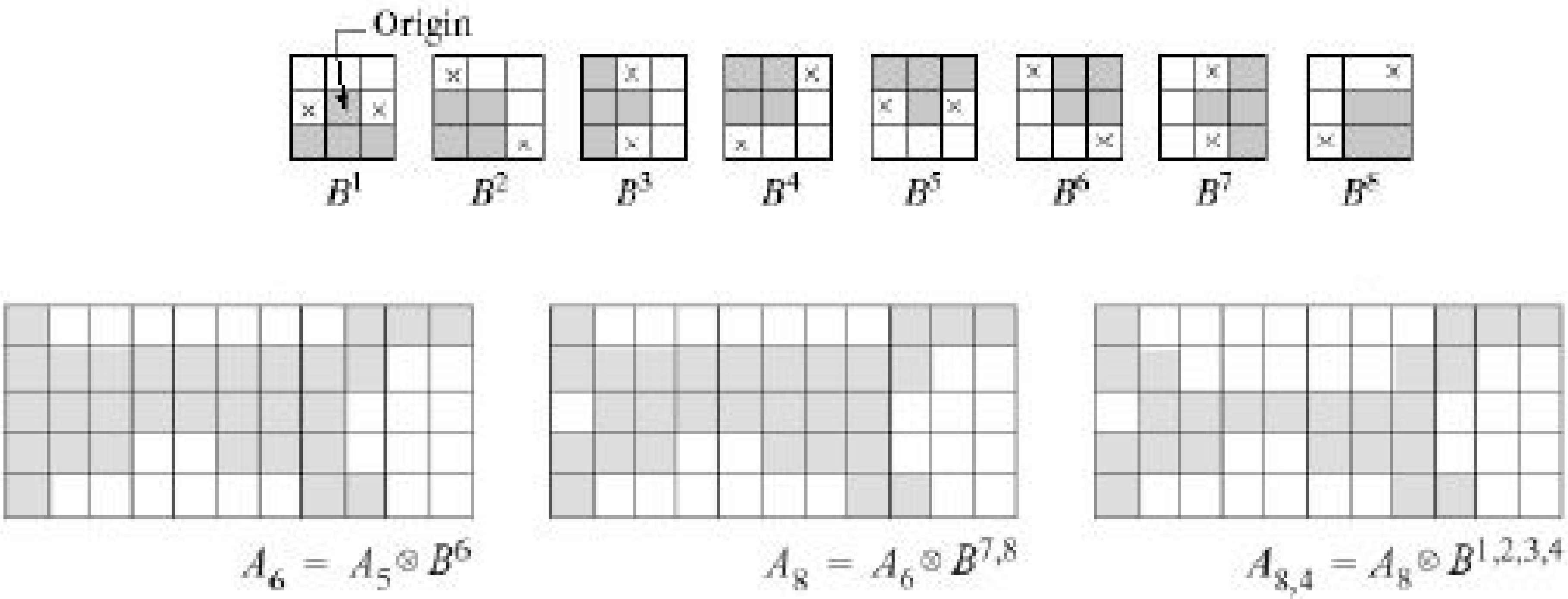


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there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

Thinning Illustration:

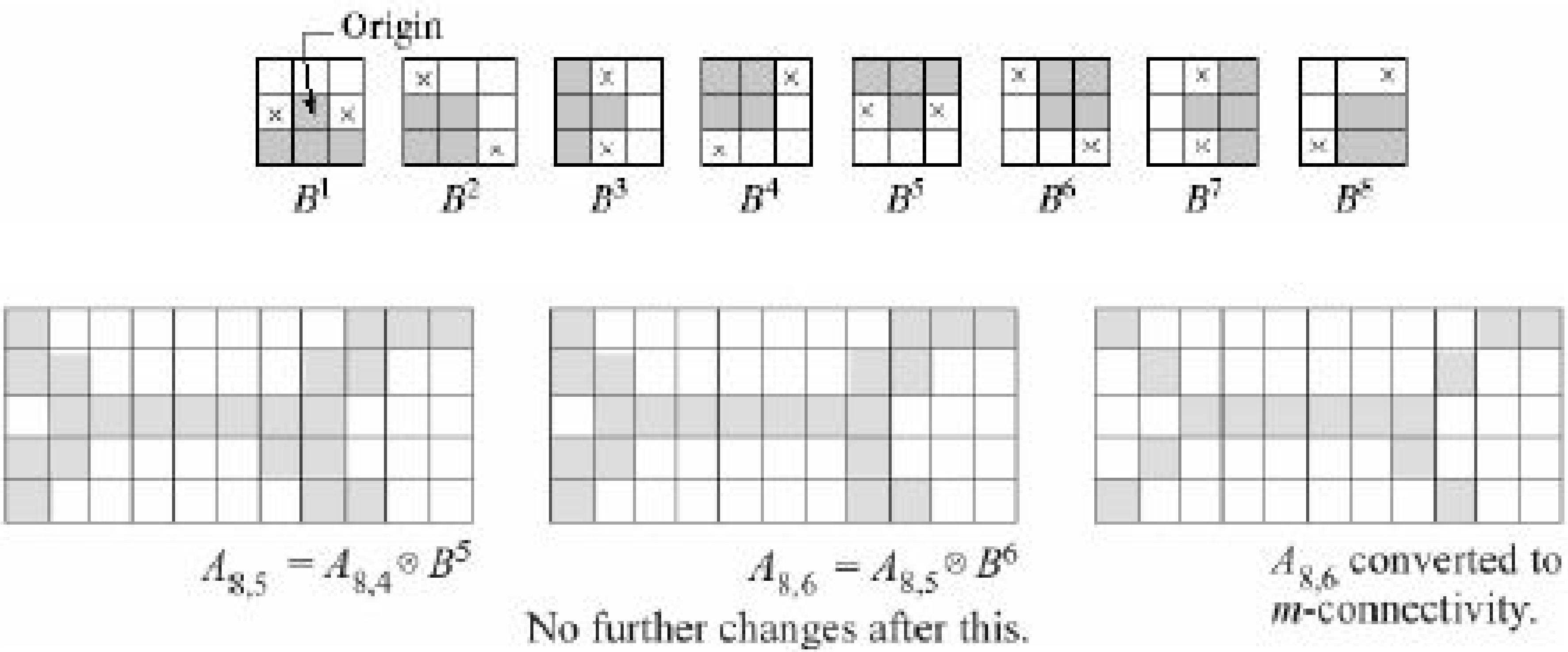
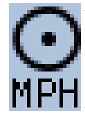


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there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

Thickening () (Adding Pixels to a Shape)

- Used to grow selected regions of foreground pixels
- E.g. applications like approximation of *convex hull*

Definition Thickening

Thickening is the morphological dual of thinning and is defined by

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

where B is a structuring element

Similar to thinning...

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

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

A Separate Thickening Algorithm

Thin the background instead, then complement the result!

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

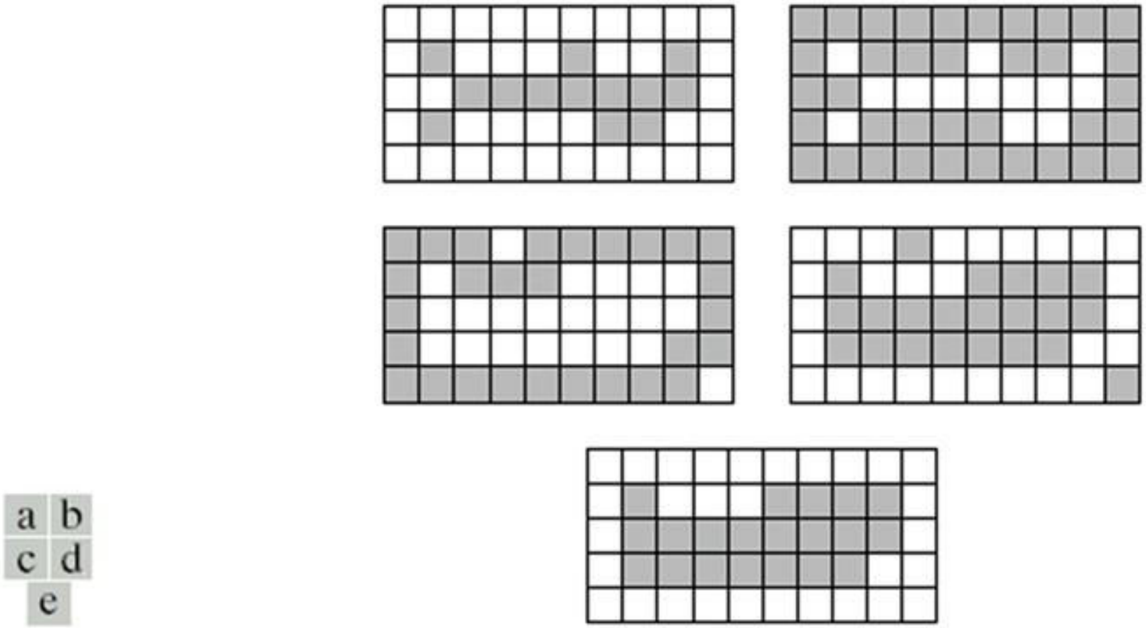


FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.