Unit-4

Binary Image Processing:

Mathematical Morphology, Structuring Elements, Morphological Image processing, Basic, Set theory, Logical operations, Standard binary morphological operations,

Dilation and erosion based operations, Properties of morphological operations, Morphological algorithms - Hit and Miss, Boundary Detection, Convex Hull, Thinning, Thickening, Distance Transform.

Introduction

- Morphology in context of image processing means description of *shape and structure* of the object in an image.
- The primary goal of morphological operation is to remove imperfection that mainly affects the shape and texture of images.
- It is obvious that morphological operations can be very useful in image segmentation as the process directly deals with 'shape extraction' in an image.
- Morphological operations (*Mathematical Morphology*) work on the basis of set theory
- Sets in Mathematical Morphology represents *objects* in an Image

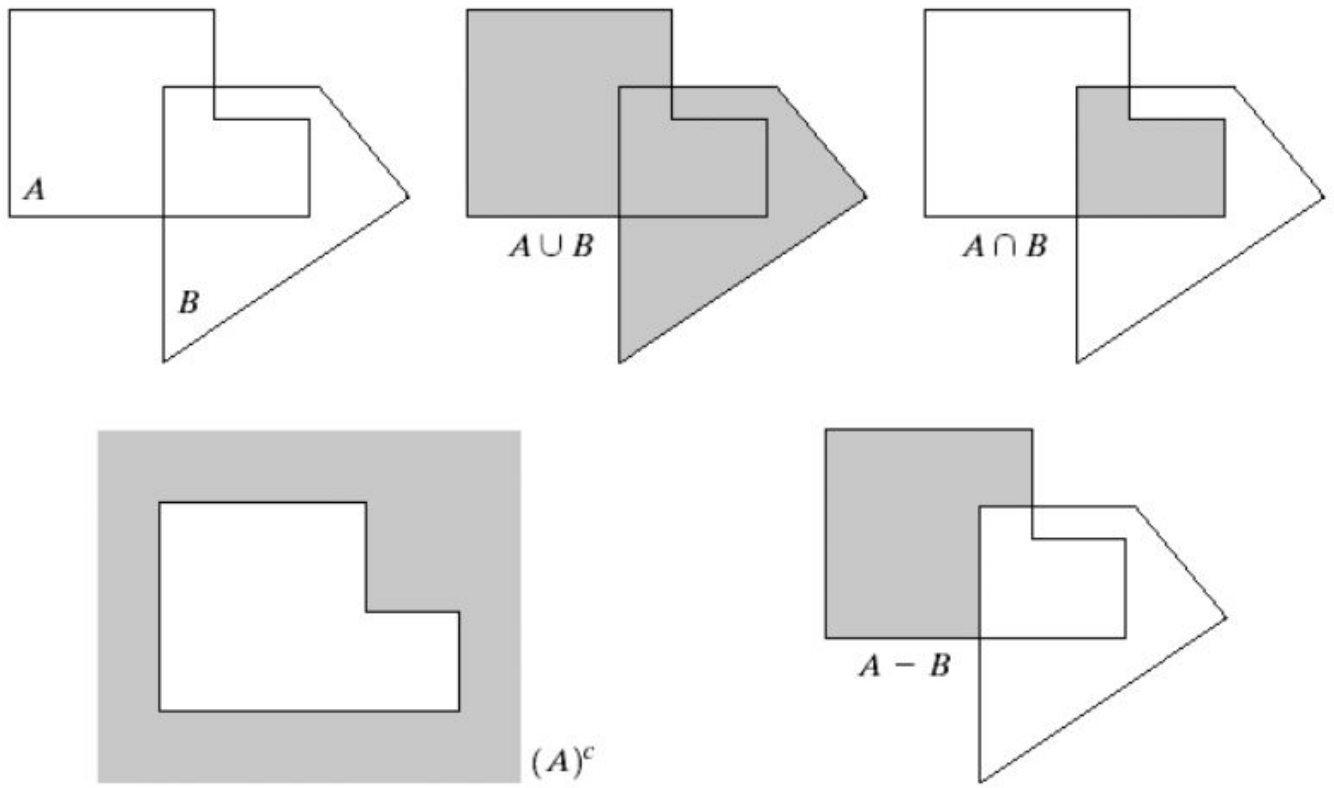
Mathematic Morphology

- It is a tool used to extract image components that are useful in the representation and description of region shape, such as-
  - boundaries extraction
  - Skeletons
  - Convex hull
  - Morphological filtering
  - Thinning
  - Pruning

Basic Set Operations

Set operators	Denotations
A Subset B	$A \subseteq B$
Union of A and B	$C = A \cup B$
Intersection of A and B	$C = A \cap B$
Disjoint	$A \cap B = \varnothing$
Complement of A	$A^c = \{ w \mid w \notin A \}$
Difference of A and B	$A\text{-}B = \{ w \mid w \in A, w \notin B \}$
Reflection of A	$\hat{A} = \{ w \mid w = -a \text{ for } a \in A \}$
Translation of set A by point $z(z_1,z_2)$	$(A)_z = \{ c \mid c = a + z, \text{for } a \in A \}$

Basic Set Theory



a b c
d e

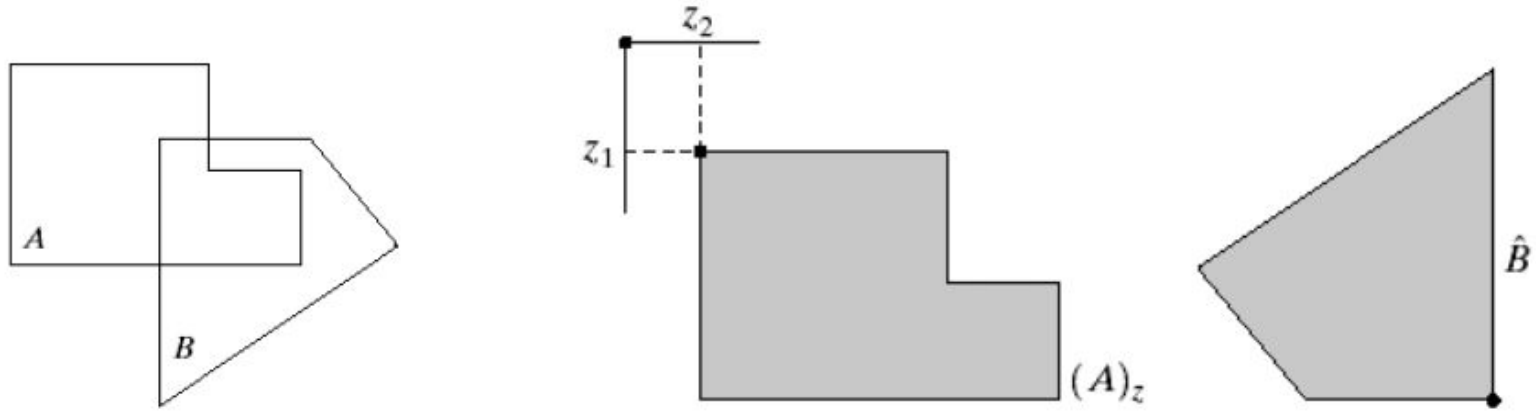
FIGURE 9.1
(a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

Reflection and Translation

- **Reflection**- the reflection of a set B denoted by $\hat{B}$ is defined as-

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

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

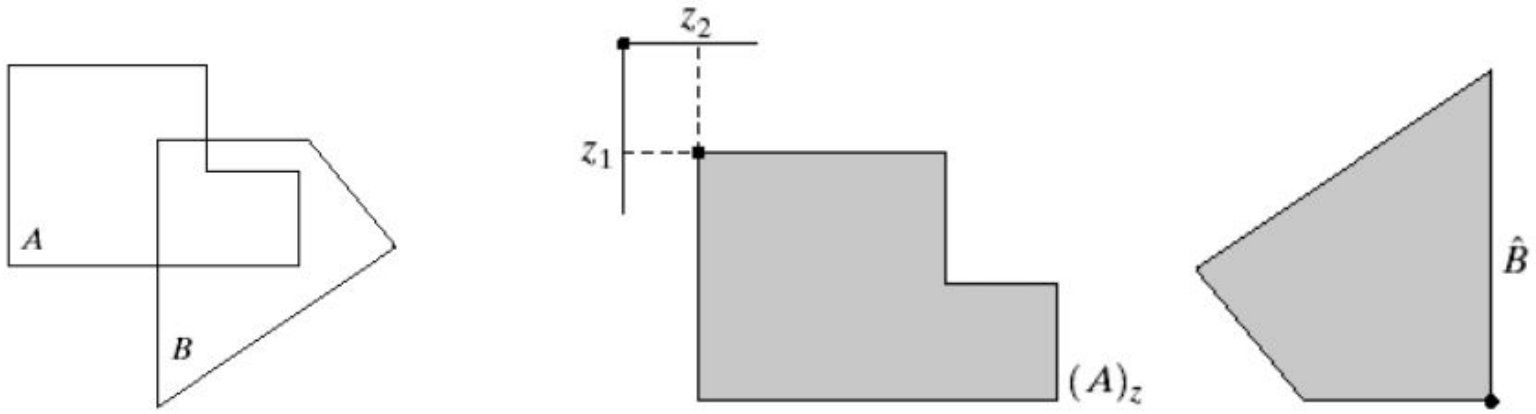


Reflection and Translation

- **Translation** – the translation of set A by points $z=(z_1, z_2)$, denoted by $(A)_z$, is defined as-

$$(A)_z = \{c \mid c = a + z, \ for\ a \in A\}$$

- If Ais the set of pixels(2-D points) representing an object in an image, the $(A)_z$ is the set of points in A whose (x, y) coordinates have been replaced by $(x+z_1, y+z_2)$



Logical Operations

A simple and effective “difference” measure is the two-input exclusive-OR operator XOR. The XOR takes logical value '1' only if its two inputs are different.

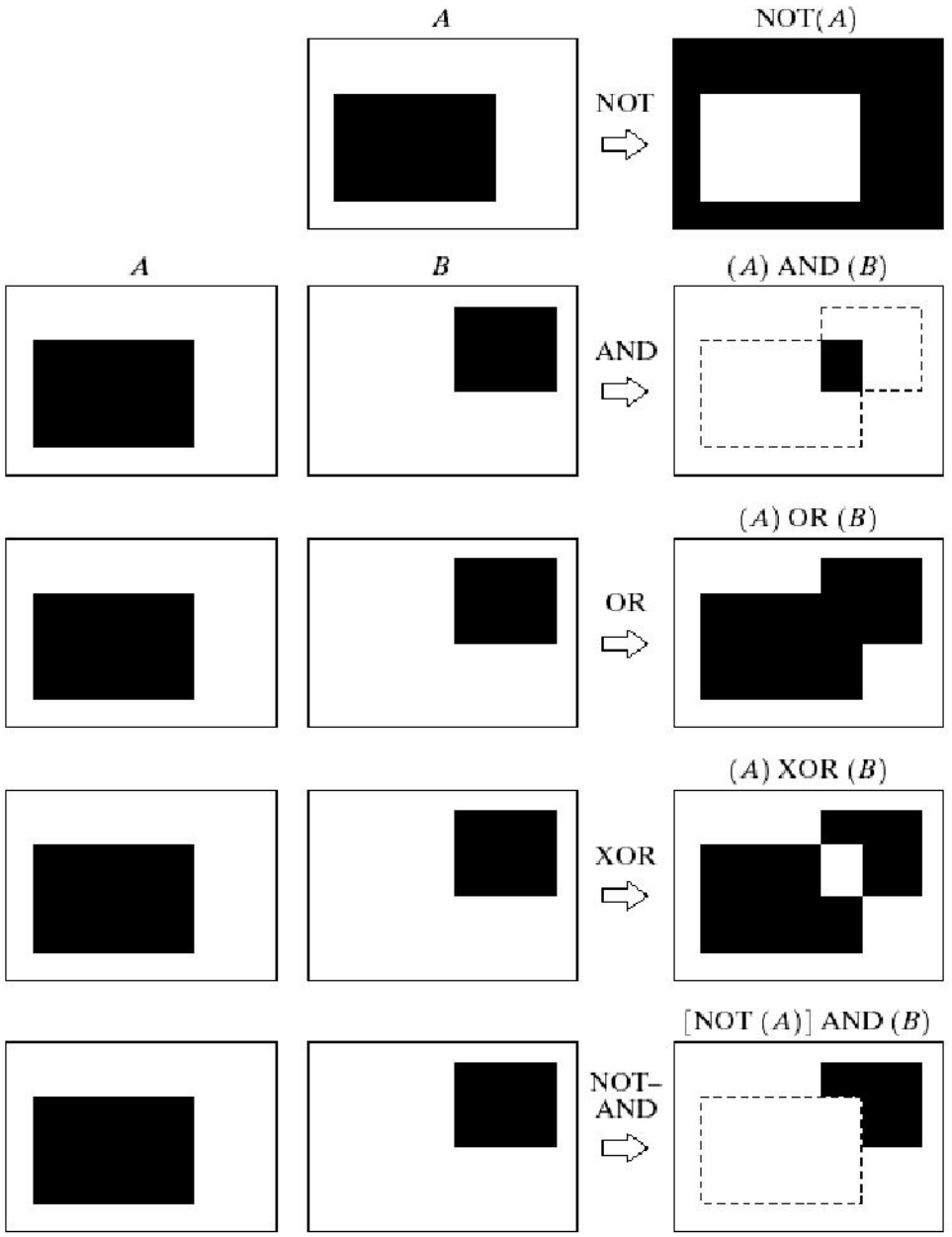


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

STRUCTURING ELEMENTS (SE)

Structuring elements are small sets in matrix form or a sub-image used to interact with the image to be probed.

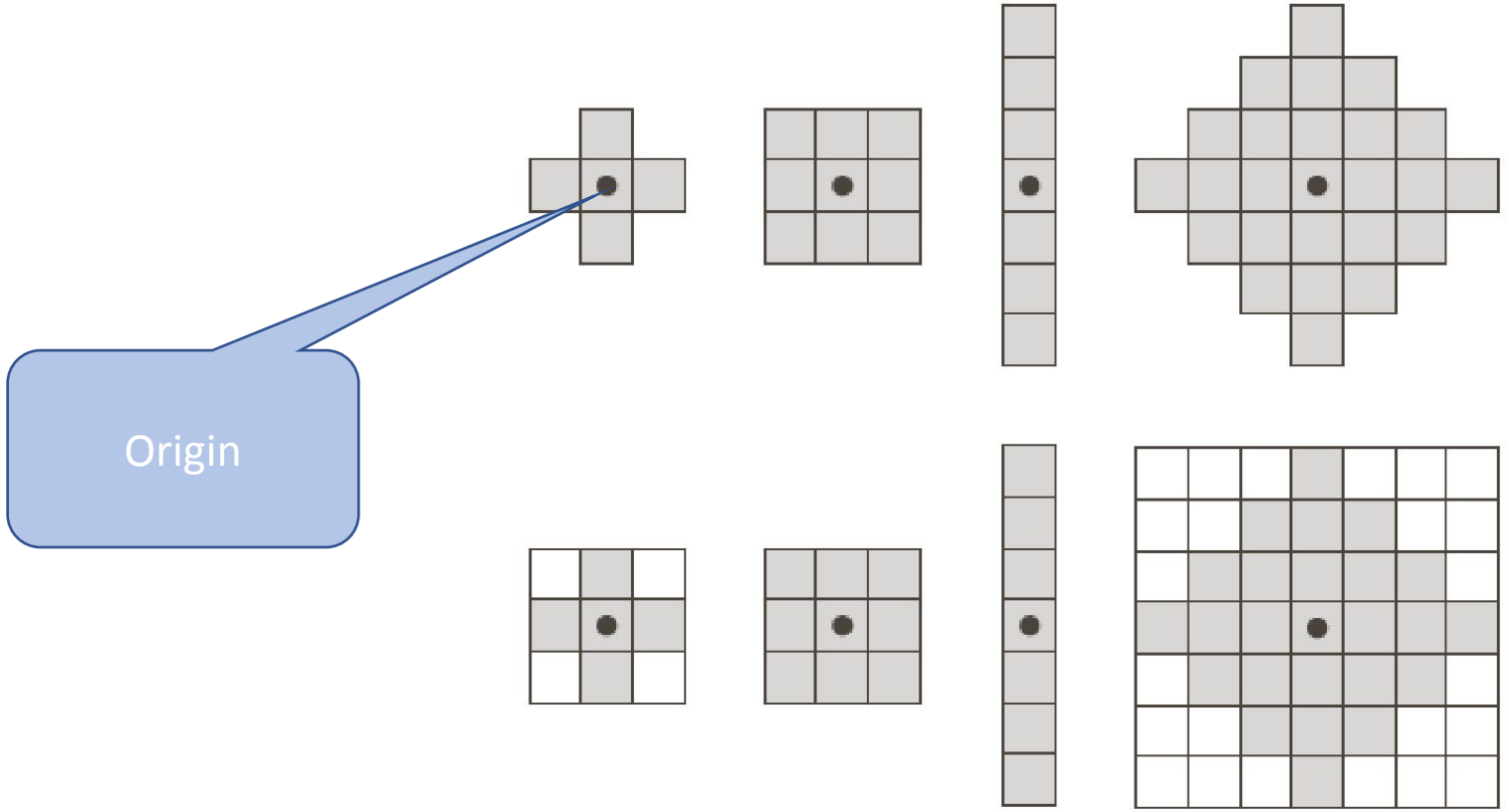


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

STRUCTURING ELEMENTS (SE)

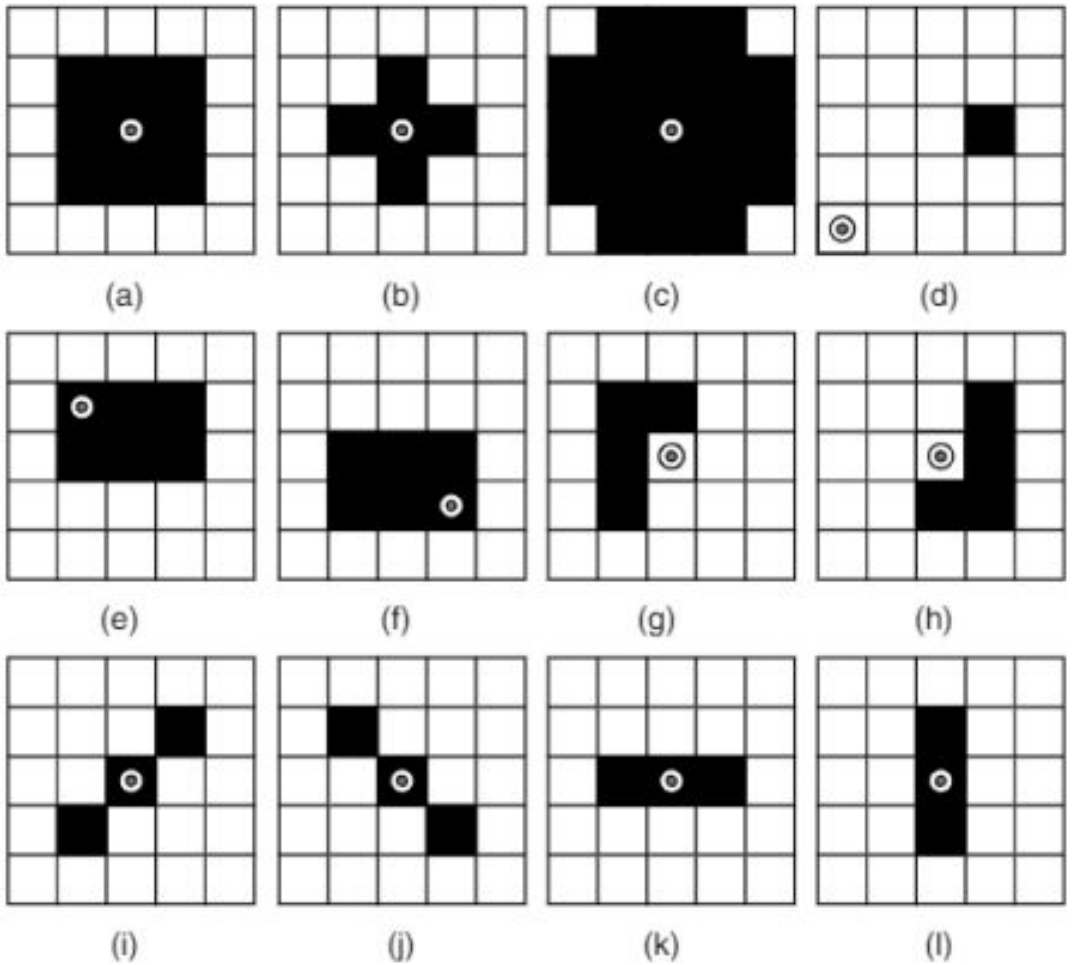
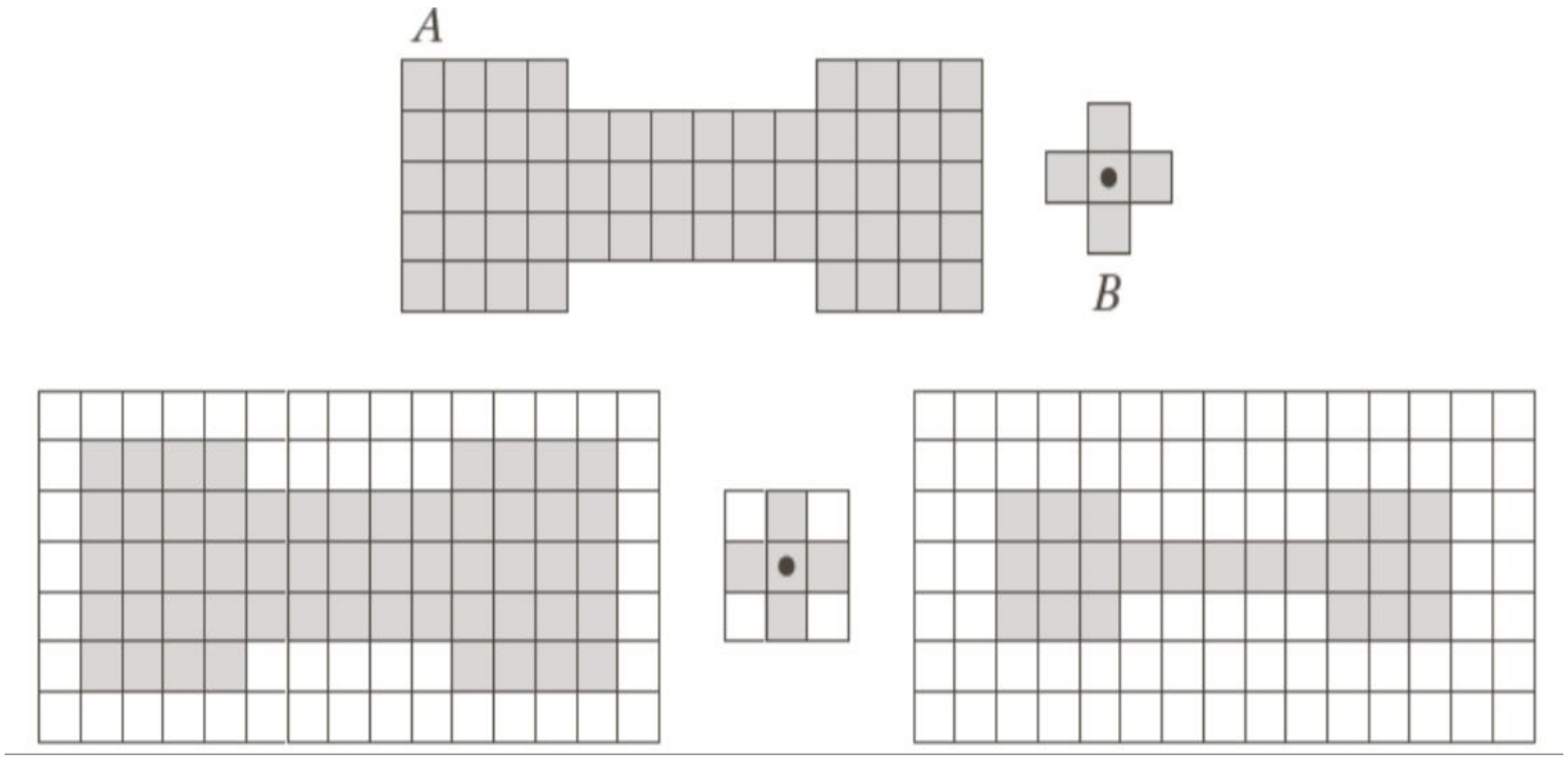


Fig. 10.2 *Some possibilities of 5 x 5 square structuring elements are shown. They are named as (a) N8 (8 –Neighbourhood centred) (b, N4 (4 –Neighbourhood centred) (c) Flat plus (d) Shifted version (e) 2 × 3 sized rectangular (f) Reflected structuring elemen. of Fig. (e) (g) L' Shaped structuring element (h) Reflected version of structuring element Fig. (g) (i) line structuring element of 45° (j) line structuring element of 135° (k) Horizontal structuring element with size 1 × 3 (l) Vertical structuring element with size 3 × 1*

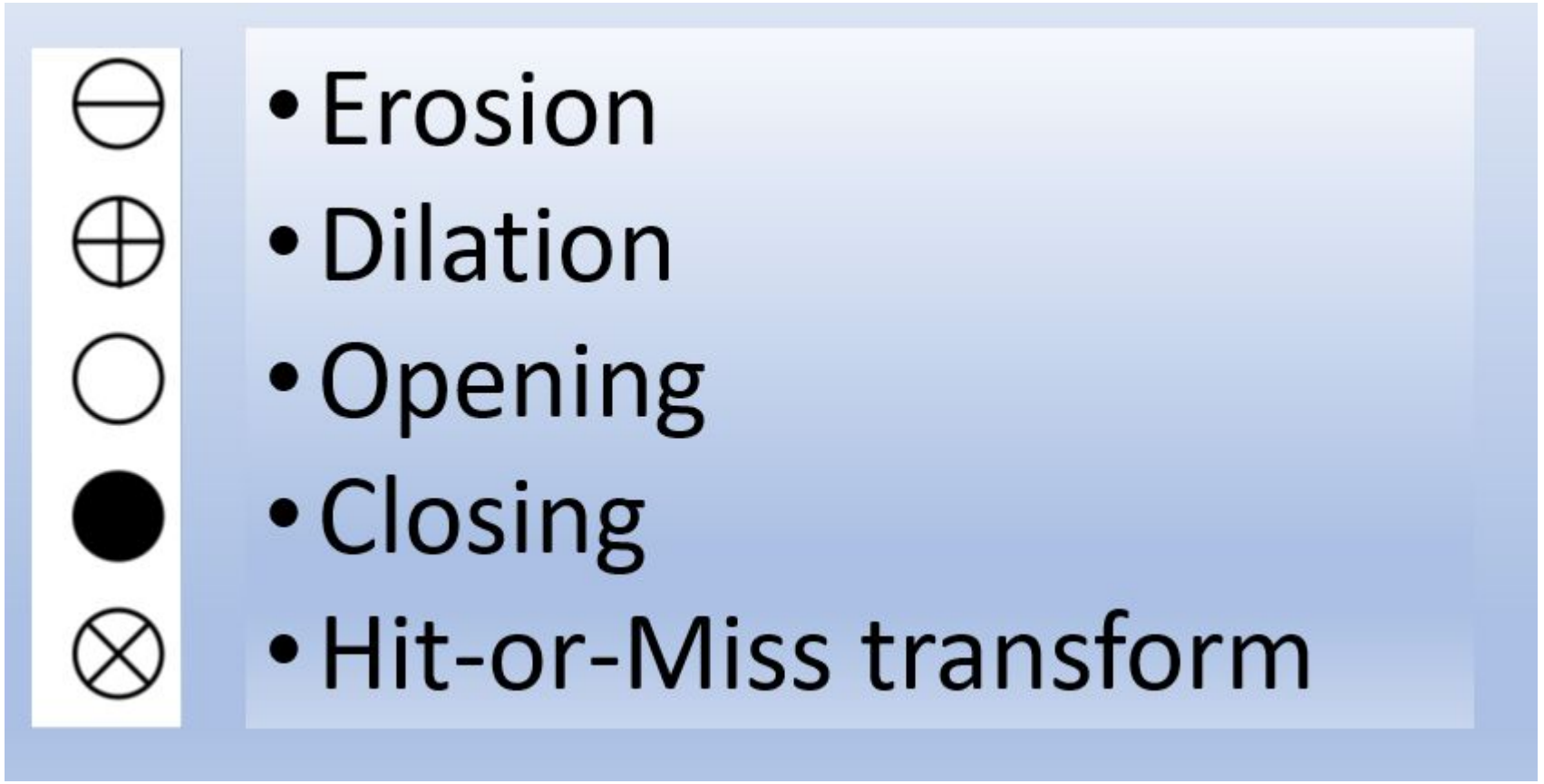
How Structuring Elements are used



The Background border is made large enough to accommodate the entire structuring element when its origin is on the border of the original set (Padding)

Suppose that we define an operation on set A using Structuring Element B, as follows: create a new set by running B over A such that the origin of B visits every elements of A. At each location of the origin of B, if B is completely contained in A, mark that location as a member of new set else mark it as not a member.

Five Binary Morphological Operations



Erosion and Dilation

Many of the morphological algorithms are based on these two primitive operations

- Erosion (Shrink): With A and B as sets in Z^2, the erosion of A by B is defined as,
  - $A \ominus B = \{z \mid (B)_z \in A\}$

Erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

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

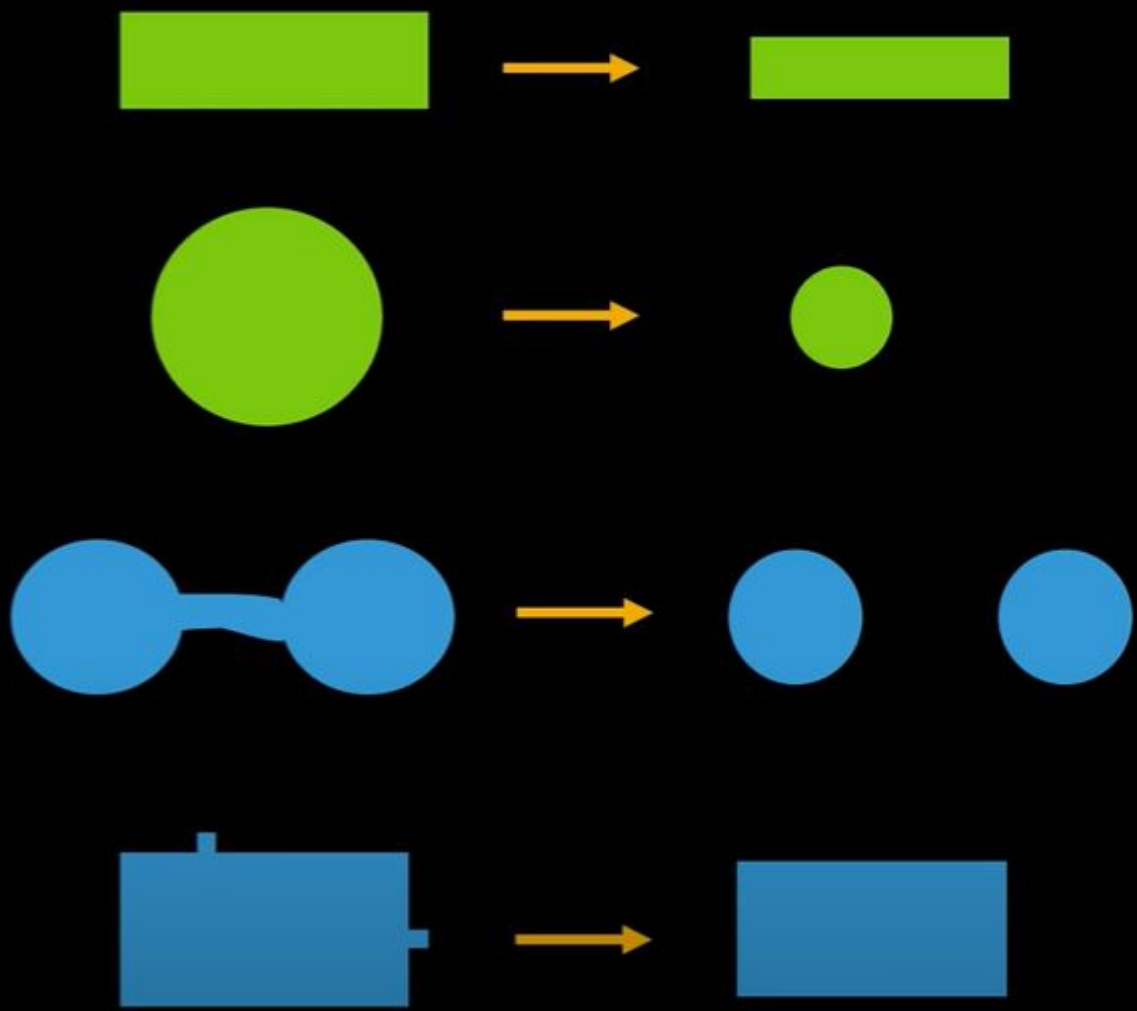
B has to be contained in A is equivalent to B not sharing any common elements with the background

Erosion

- The erosion process *enlarges* the number of pixels with *value zero* (background) and *shrinks* the number of pixels with *value one* (foreground).
- The erosion operation *removes* those structures which are *lesser in size than that of the structuring element*. So it can be used to remove the noisy ‘connection’ between two objects.

Erosion

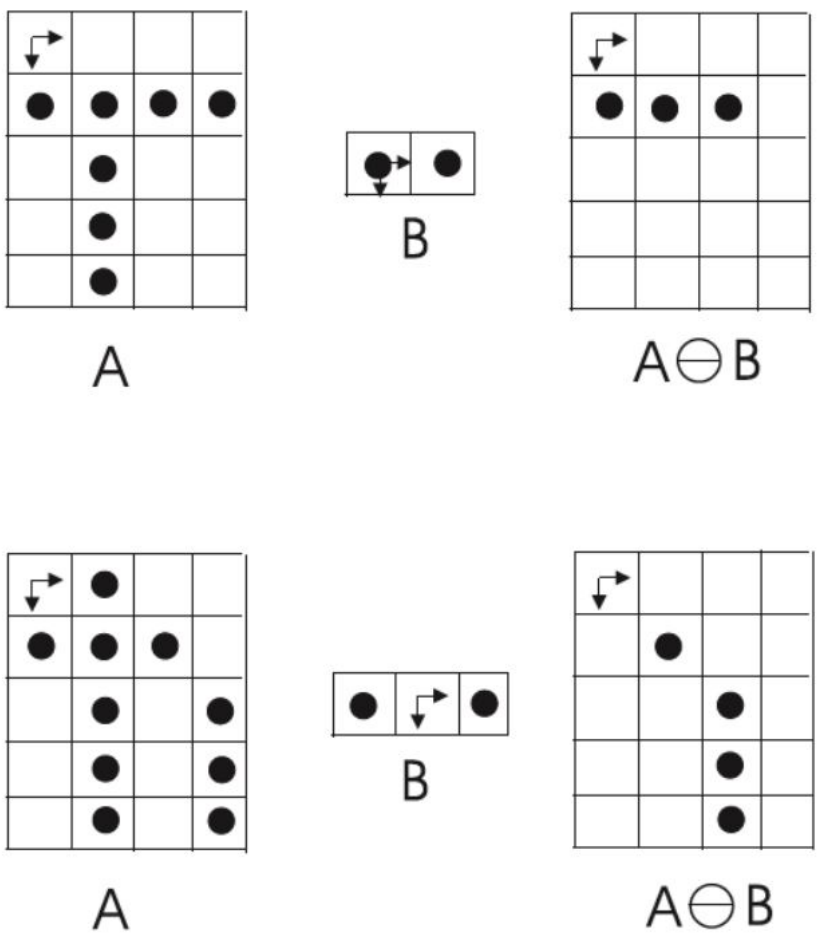
Erosion shrinks the connected sets of 1s of a binary image.

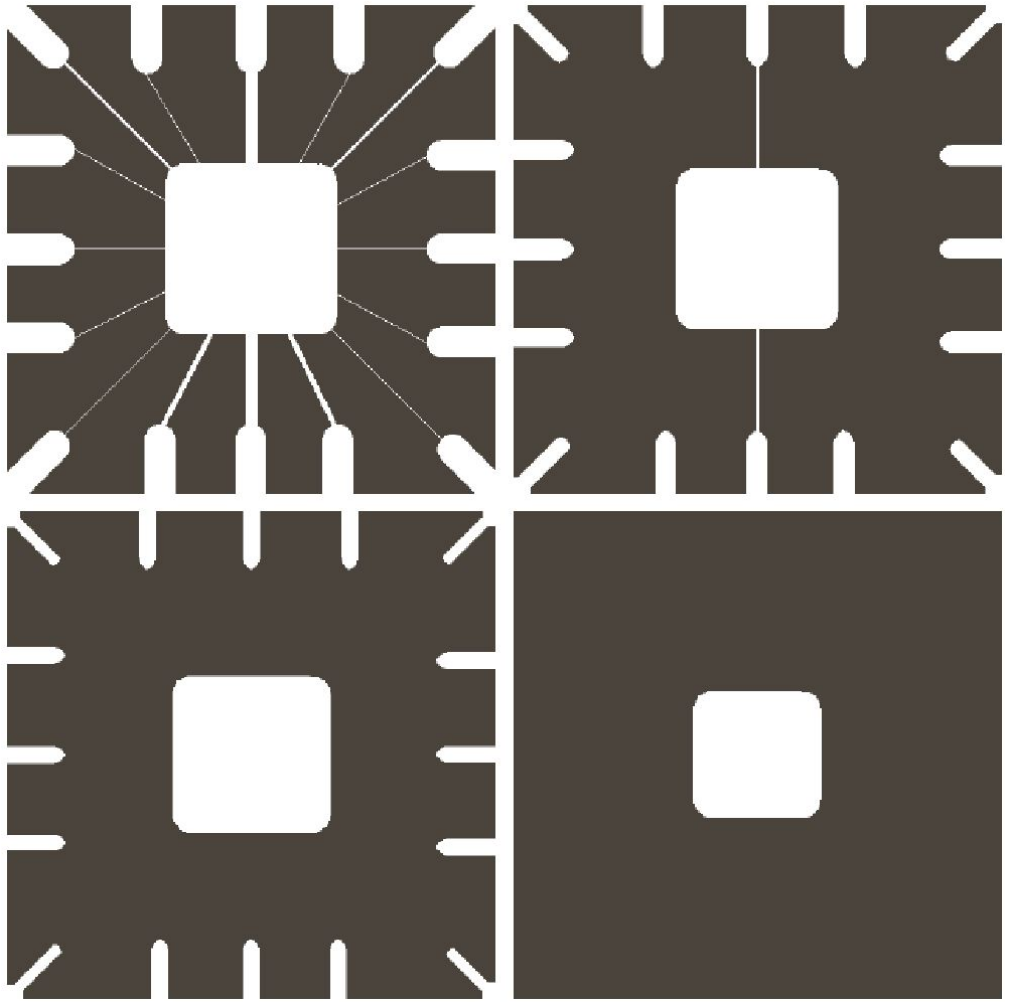


It can be used for:

- Shrinking features
- Removing bridges, branches, protrusions

Erosion





a b
c d

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

Erosion

- Erosion shrinks or thins objects in a binary image
- Erosion as a morphological filtering operation in which image details smaller than the structuring elements are filtered from the image
- Erosion performed the function of a “line filter”

Dilation

Dilation (makes an object to grow by size)-

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

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

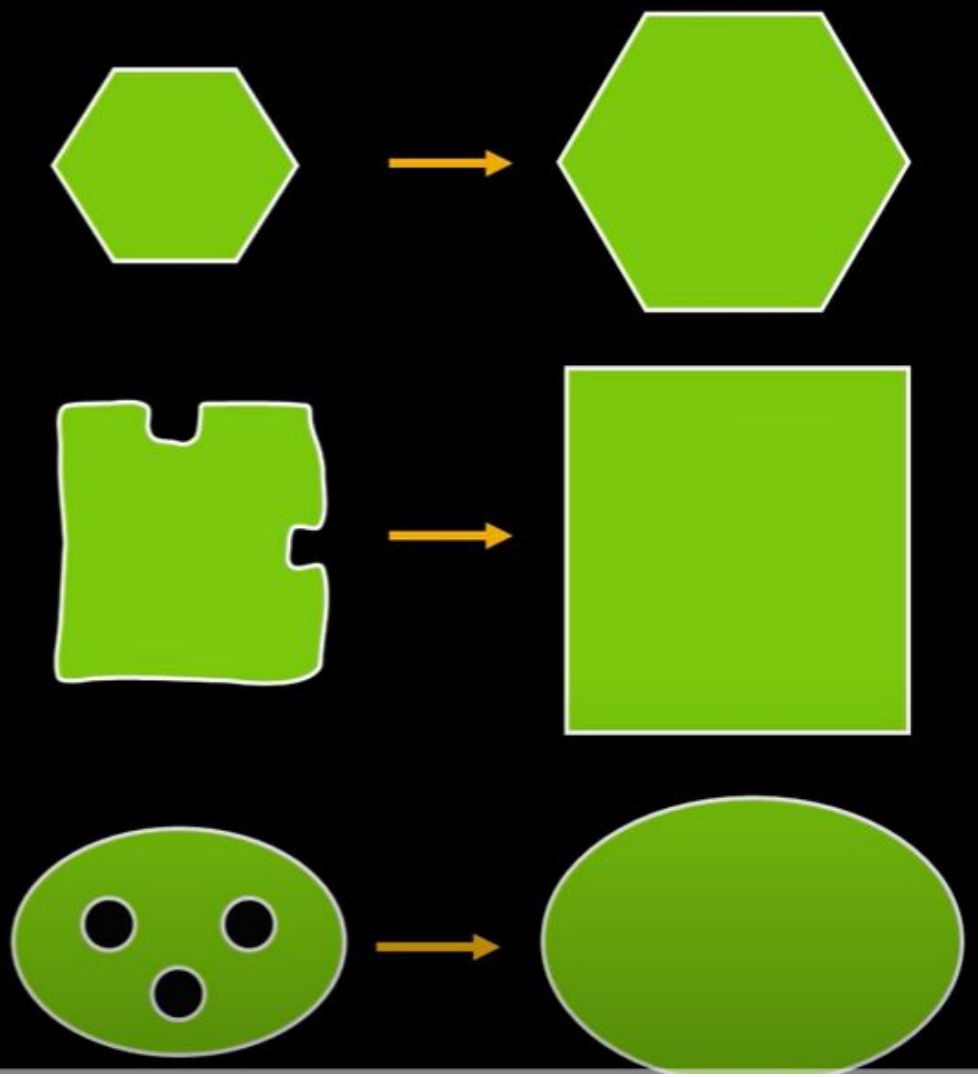
- This equation is based on reflecting B about its origin, and shifting this reflection by z.
- The dilation of A by B is the set of all displacements, z such that $\hat{B}$ and A overlap by at least one element.
- Dilation, as said above, adds pixels to the boundary elements.
- The dilation process enlarges the number of pixels with value one (foreground) and shrinks the number of pixels with value zero (background).

Dilation

Dilation expands the connected sets of 1s of a binary image.

It can be used for:

- Growing features
- Filling holes and gaps



Dilation

- A =[0 0 0 0 0
  0 1 1 1 0
  0 1 1 1 0
  0 1 1 1 0
  0 0 0 0 0]
- B=[0 1 0
  1 1 1
  0 1 0]

Opening and Closing

- Opening is so called because it can open up a gap between objects connected by a thin bridge of pixels. Any regions that have survived the erosion are restored to their original size by the dilation
- Closing also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour. Closing is so called because it can fill holes.

Opening

- The opening of an image is a combinational operation of erosion and dilation.
- Opening of set A by Structuring Element(B)is defined as –

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

- Thus, opening A by B is the erosion of A by B, followed by a dilation of the result by B
- Opening is an idempotent operation: once an image hasbeen opened, subsequent openings with the same structuring element have no further effect on that image:

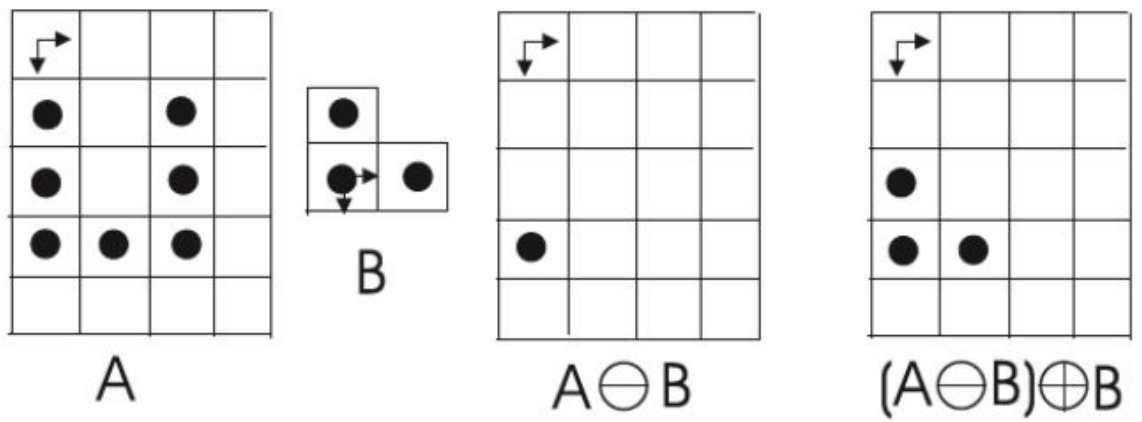
$$(A \circ B) \circ B = A \circ B$$

Opening

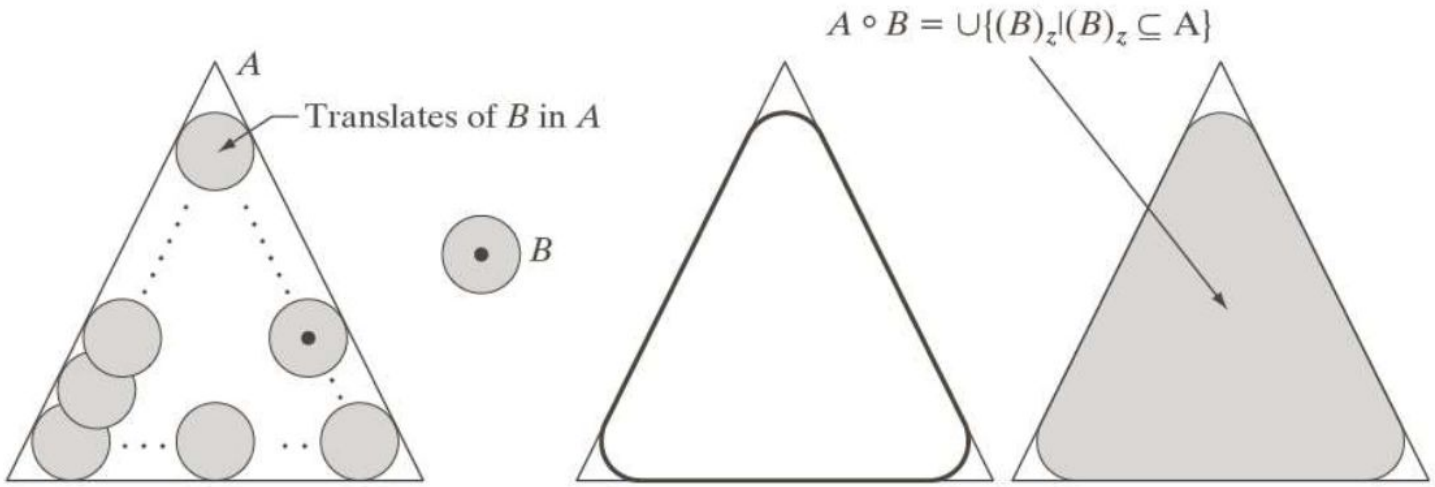
Erosion followed by dilation, denoted $\circ$

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

- eliminates protrusions
- breaks necks
- smoothes contour



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.

The opening operation has a simple geometric interpretation (Fig. 9.8). Suppose that we view the structuring element B as a (flat) "rolling ball." The *boundary* of $A \circ B$ is then established by the points in B that reach the *farthest* into the boundary of A as B is rolled around the *inside* of this boundary. This geometric *fitting* property of the opening operation leads to a set-theoretic formulation, which states that the opening of A by B is obtained by taking the union of all translates of B that fit into A. That is, opening can be expressed as a fitting process such that

$$A \circ B = \bigcup\{(B)_z | (B)_z \subseteq A\} \qquad (9.3\text{-}3)$$

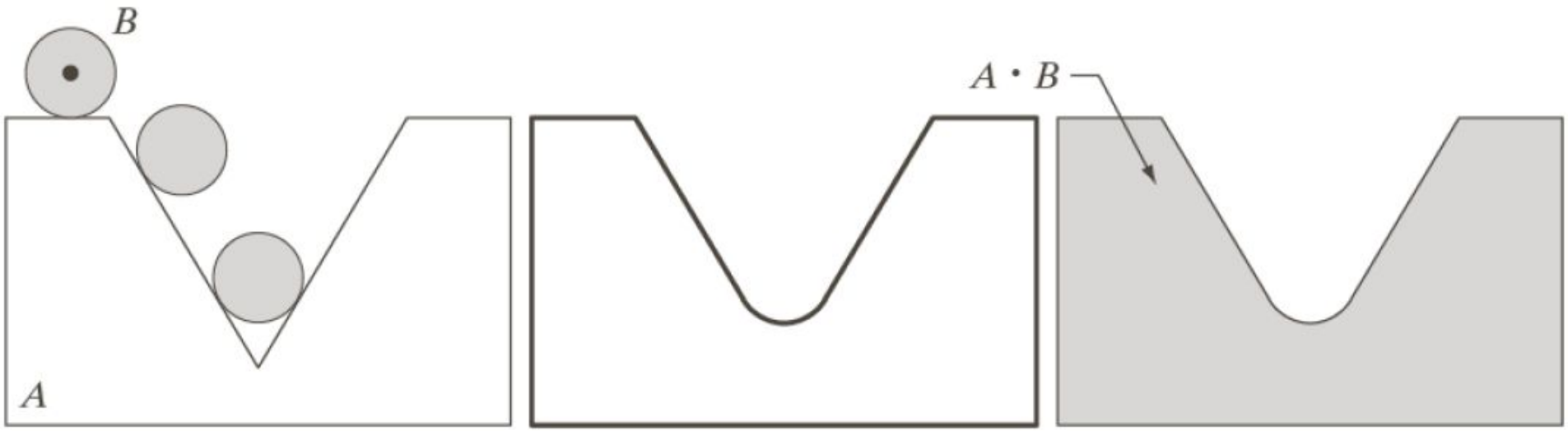
Closing

Dilation followed by erosion, denoted •

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

- smooth contour
- fuse narrow breaks and long thin gulfs
- eliminate small holes
- fill gaps in the contour

Closing



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.

Properties

Opening

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

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

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

Closing

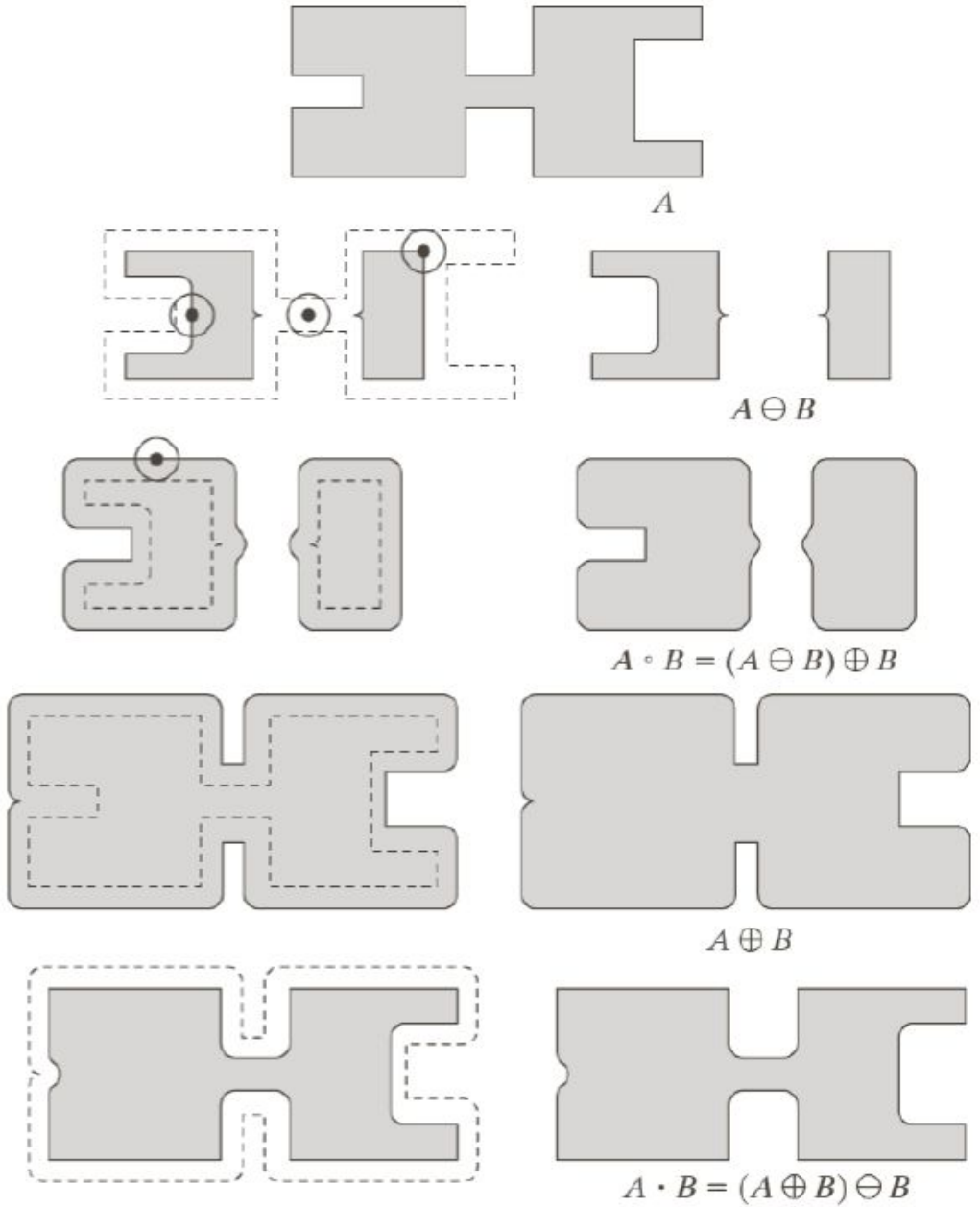
(i) A is a subset (subimage) of $A \bullet B$

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

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

Note: repeated openings/closings have no effect!

Opening and Closing



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.

Properties of Morphological operations

10.10.1 Increasing

Let X and Y be two images. Let the image X be smaller than the image Y, that is, $X(m, n) \leq Y(m, n)$, for all the values of m and n. Then

$$X \ominus B \leq Y \ominus B \tag{10.11}$$

$$X \oplus B \leq Y \oplus B \tag{10.12}$$

$$X \circ B \leq Y \circ B \tag{10.13}$$

$$X \bullet B \leq Y \bullet B \tag{10.14}$$

Erosion and dilation are said to be increasing operations. Since erosion and dilation are increasing operations, opening and closing are also increasing operations.

Properties of Morphological operations

10.10.2 Expansivity

The expansivity of the morphological operations is as follows:

$$X \ominus B \leq X \tag{10.15}$$

$$X \oplus B \geq X \tag{10.16}$$

$$X \circ B \leq X \tag{10.17}$$

$$X \bullet B \geq X \tag{10.18}$$

Erosion is anti-expansive, i.e the no of one's present in the output eroded image is less as compared to the original image.

Dilation is expansive.

Properties of Morphological operations

10.10.3 Duality

The next property of binary morphological operations is duality. The property of duality is as follows:

$$X \ominus B \leq \left(X^c \oplus B\right)^c \tag{10.19}$$

$$X \oplus B \leq \left(X^c \ominus B\right)^c \tag{10.20}$$

$$X \circ B \leq \left(X^c \bullet B\right)^c \tag{10.21}$$

$$X \bullet B \leq \left(X^c \circ B\right)^c \tag{10.22}$$

From the Eqs (10.19) and (10.20), dilation is the dual of erosion. Thus, dilation is the erosion of the complemented function and vice versa. Here, the input image and the structuring element are shown in Fig. 10.28. The MATLAB code and the resultant of this property are shown in Figs. 10.29 and 10.30 respectively.

Similarly, opening and closing are dual operations. The closing of X corresponds to the opening of the complemented image X^c. The dual operation of the opening and closing MATLAB code is shown in Fig. 10.29, and the resultant image is shown in Fig. 10.31(c).

Properties of Morphological operations

10.10.4 Chain Rule

The chain rule of the morphological operations is as follows:

$$(X \ominus B_1) \ominus B_2 = X \ominus (B_1 \oplus B_2) \tag{10.23}$$

$$(X \oplus B_1) \oplus B_2 = X \oplus (B_1 \oplus B_2) \tag{10.24}$$

Equations (10.23) and (10.24) imply that the erosion or dilation of an image X by a wide or complex structural element $B(\equiv B_1 \oplus B_2)$ can be performed using the basic components of B, namely, B_1 and B_2. The input image X and the structuring elements B_1 and B_2 are shown in Fig. 10.32. The MATLAB code for the chain-rule property is shown in Fig. 10.33

The resultant images of the chain-rule property are illustrated in Fig. 10.34. Figure 10.34(a) indicates the result of the LHS of Eq. 10.23, and 10.35(b) indicates the result of the RHS of Eq. 10.23. Figure 10.34(c) indicates the result of the LHS of Eq. 10.24, and 10.35(d) indicates the result of the RHS of Eq. 10.24.

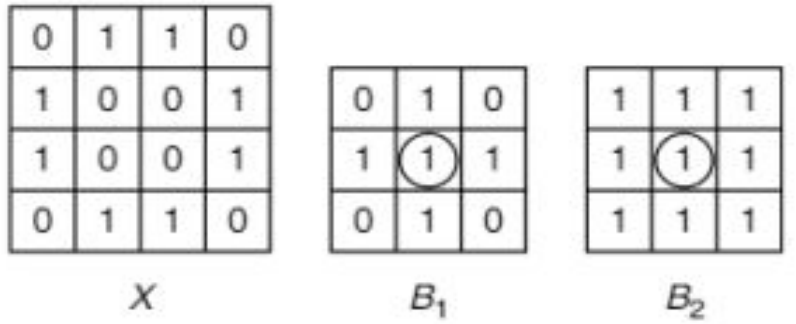


Fig. 10.32 *Binary image and structuring element*

Properties of Morphological operations

10.10.5 Idempotency

The last property of binary morphological operations is idempotency. The idempotency property is defined as follows:

$$(X \circ B) \circ B = X \circ B \tag{10.25}$$

$$(X \bullet B) \bullet B = X \bullet B \tag{10.26}$$

Hit-or-Miss Transformation

- The hit-or-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image.
- The hit-and-miss transform is a basic tool to detect or locate an object of given size or shape in an image.
- Concept: To detect a shape:
  - Hit object
  - Miss background

Hit-or-Miss Transformation

- The **hit and miss transform** allows to derive information on how objects in a binary image are related to their surroundings.
- The operation requires a matched pair of structuring elements, $\{B_1, B_2\}$, that probe the inside and outside, respectively, of objects in the image:
- composite SE: object part (B_1) and background part (B_2)
- the reason for using structuring element B_1 associated with object and B_2 associated with the background is based on the assumed definition that two or more objects are distinct only if they form disjoint sets

Hit - or - Miss transform is given as

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

where A = *Set in which we want to find the location of object* X

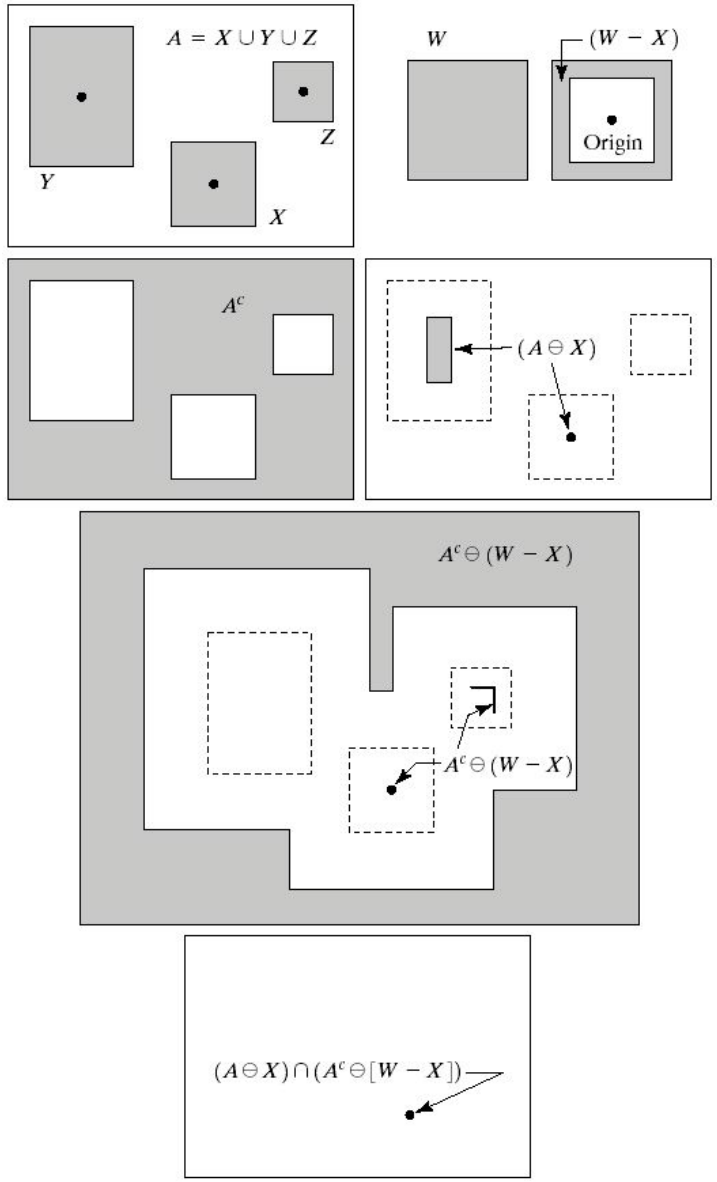
B = *Set composed of* X *and its background* W

We can also write Hit - or - Miss transform as

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

where B_1 = *Object* *and* B_2 = *background*

Hit-or-Miss Transformation



a b
c d
e
f

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

Hit and Miss - Example 1

Given a following image segment, use Hit-or-Miss transform to find the top edge of the square.

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

Use two structuring elements shown below.

B1 =

0	0	0
0	1	0
0	1	0

B2 =

0	1	0
0	0	0
0	0	0

Step 1
Erode A by B_1

Step 2
Find A^c and erode A^c with B_2

Step 3
Find inter-section of results in Step 1 and 2

Hit and Miss - Example 1

STEP 1: Erode A by B_1

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

A

0	0	0
0	1	0
0	1	0

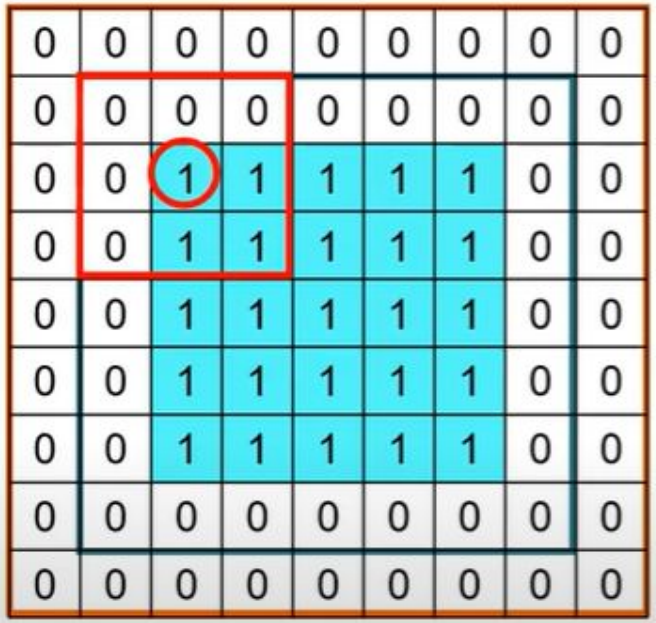
B_1

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

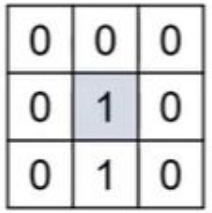
A_{Pad}

Hit and Miss - Example 1

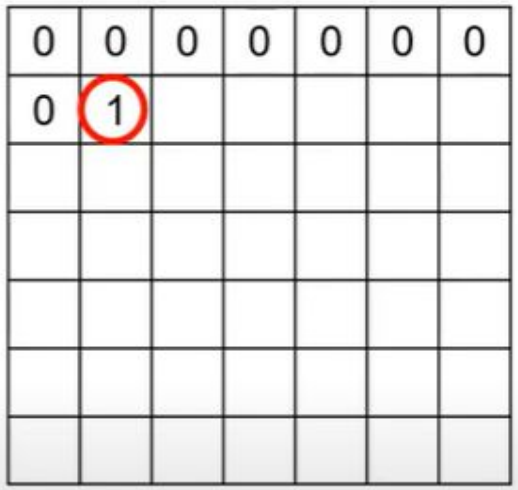
STEP 1: Erode A by B_1



A_{Pad}

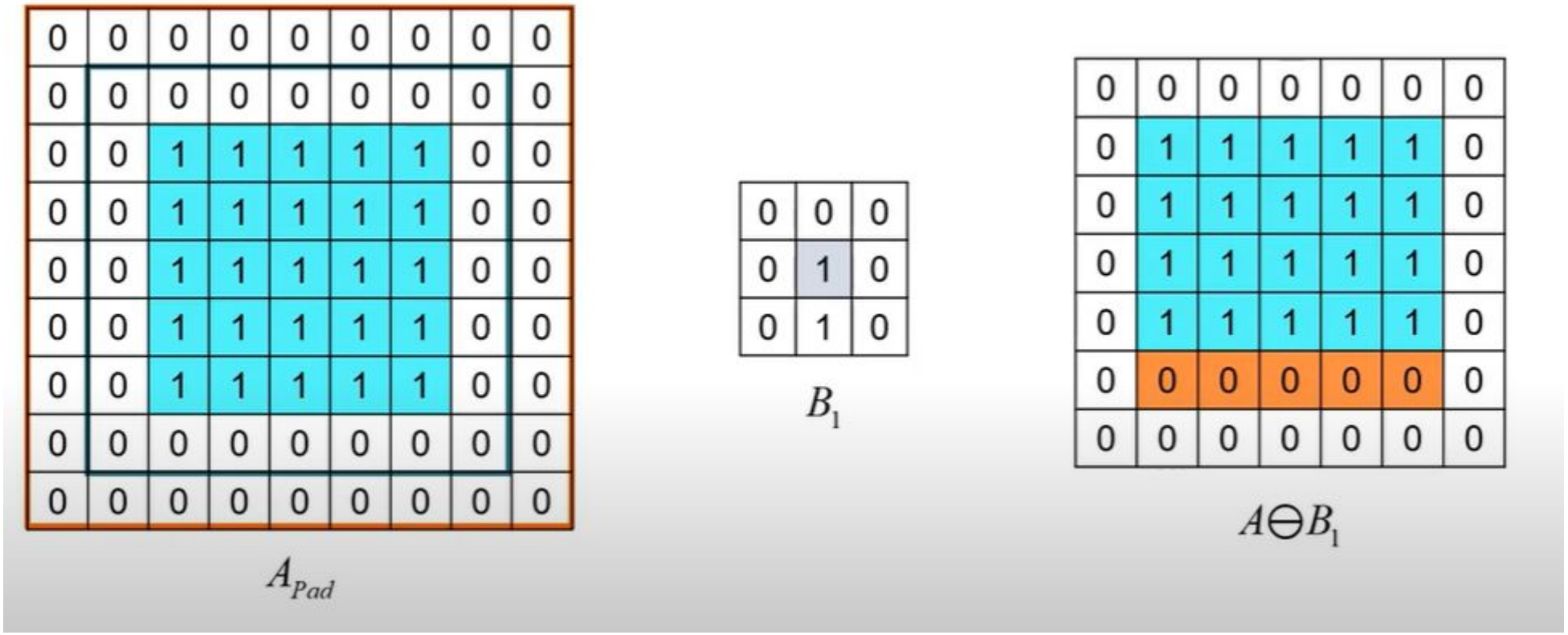


B_1



$A \ominus B_1$

Hit and Miss - Example 1



Hit and Miss - Example 1

STEP 2: Erode A^c by B_2

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

A^c

0	1	0
0	0	0
0	0	0

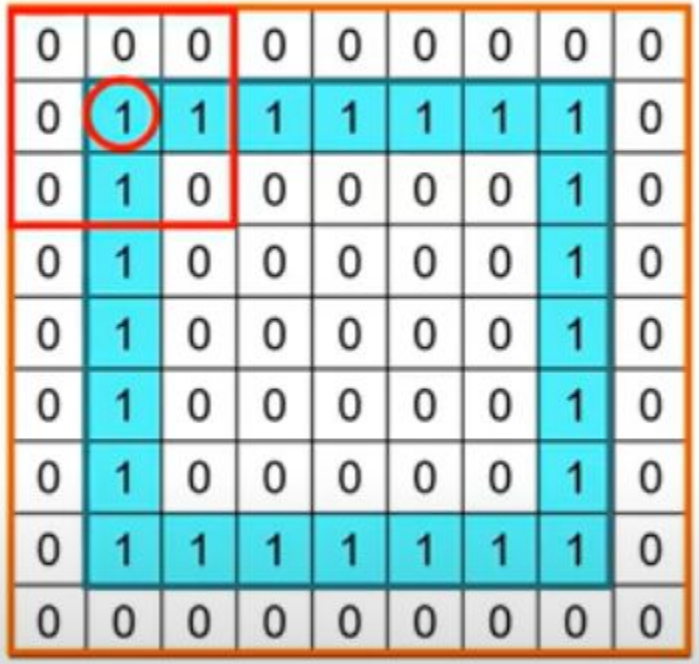
B_2

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

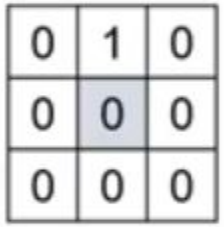
A^c_{Pad}

Hit and Miss - Example 1

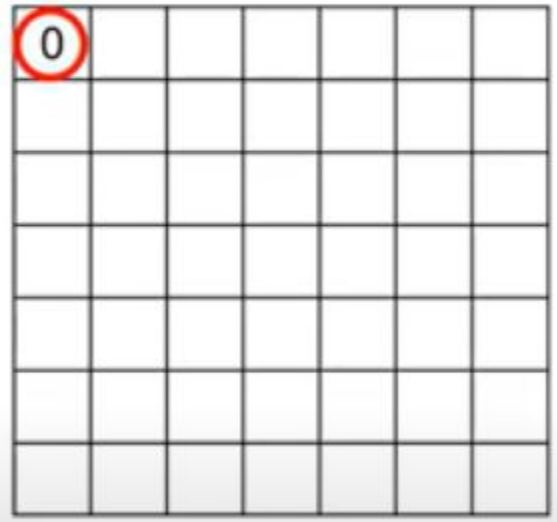
STEP 2: Erode A^c by B_2



$A^c{}_{Pad}$



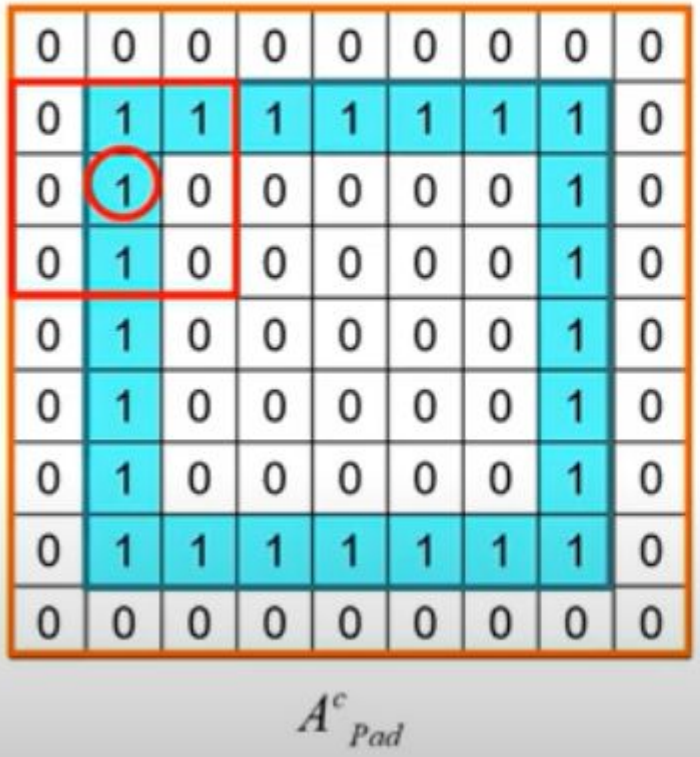
B_2



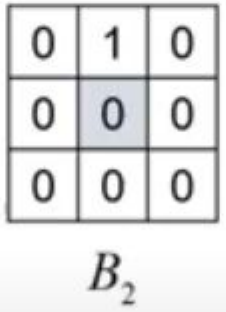
$A^c \ominus B_2$

Hit and Miss - Example 1

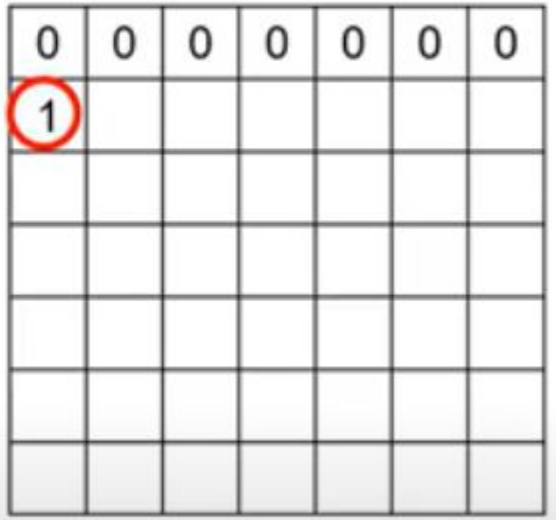
STEP 2: Erode A^c by B_2



A^c_{Pad}



B_2



$A^c \ominus B_2$

Hit and Miss - Example 1

STEP 2: Erode A^c by B_2

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

A^c_{Pad}

0	1	0
0	0	0
0	0	0

B_2

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

$A^c \ominus B_2$

Hit and Miss - Example 1

STEP 3: Find inter-section of results in Step 1 and 2

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

$A \ominus B_1$

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

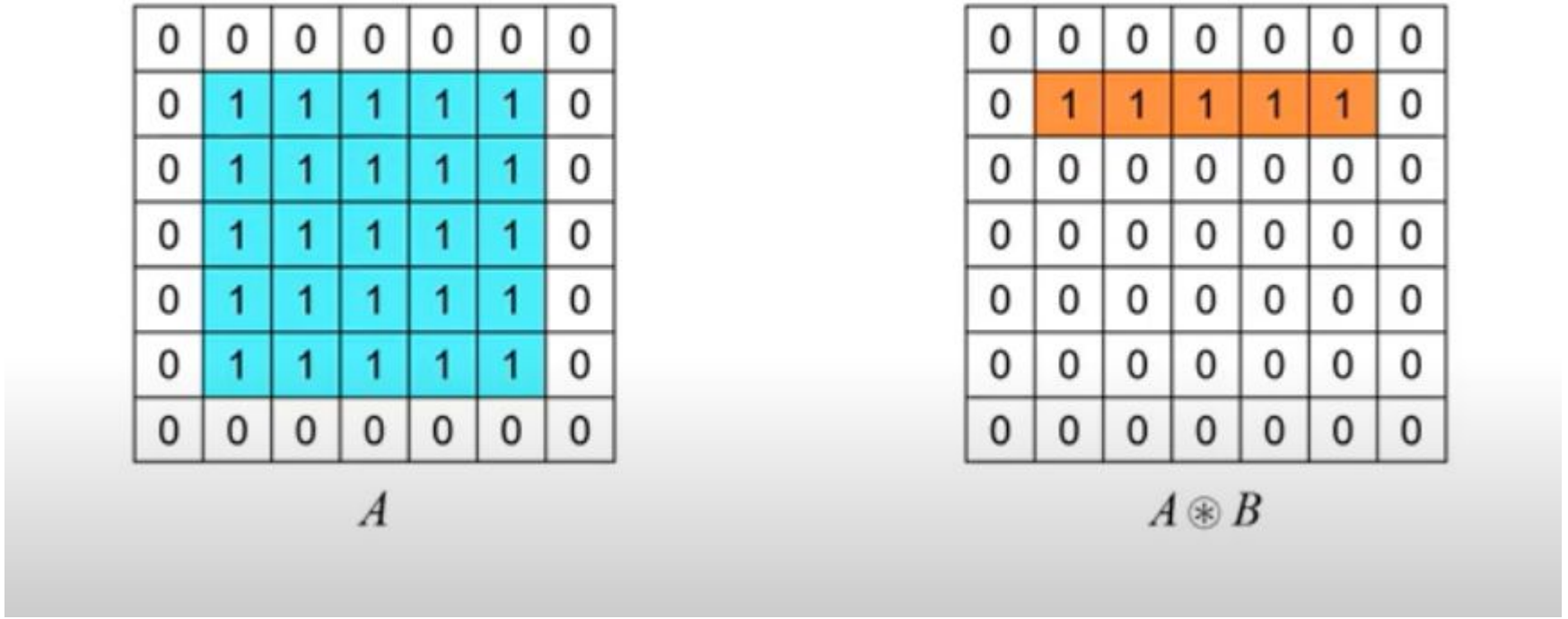
$A^c \ominus B_2$

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

$A \circledast B$

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

Hit and Miss - Example 1



Hit and Miss - Example 2

Perform the Hit-or-Miss transform on the image shown below.

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

Use the structuring element shown below.

x	1	x
1	1	0
x	0	0

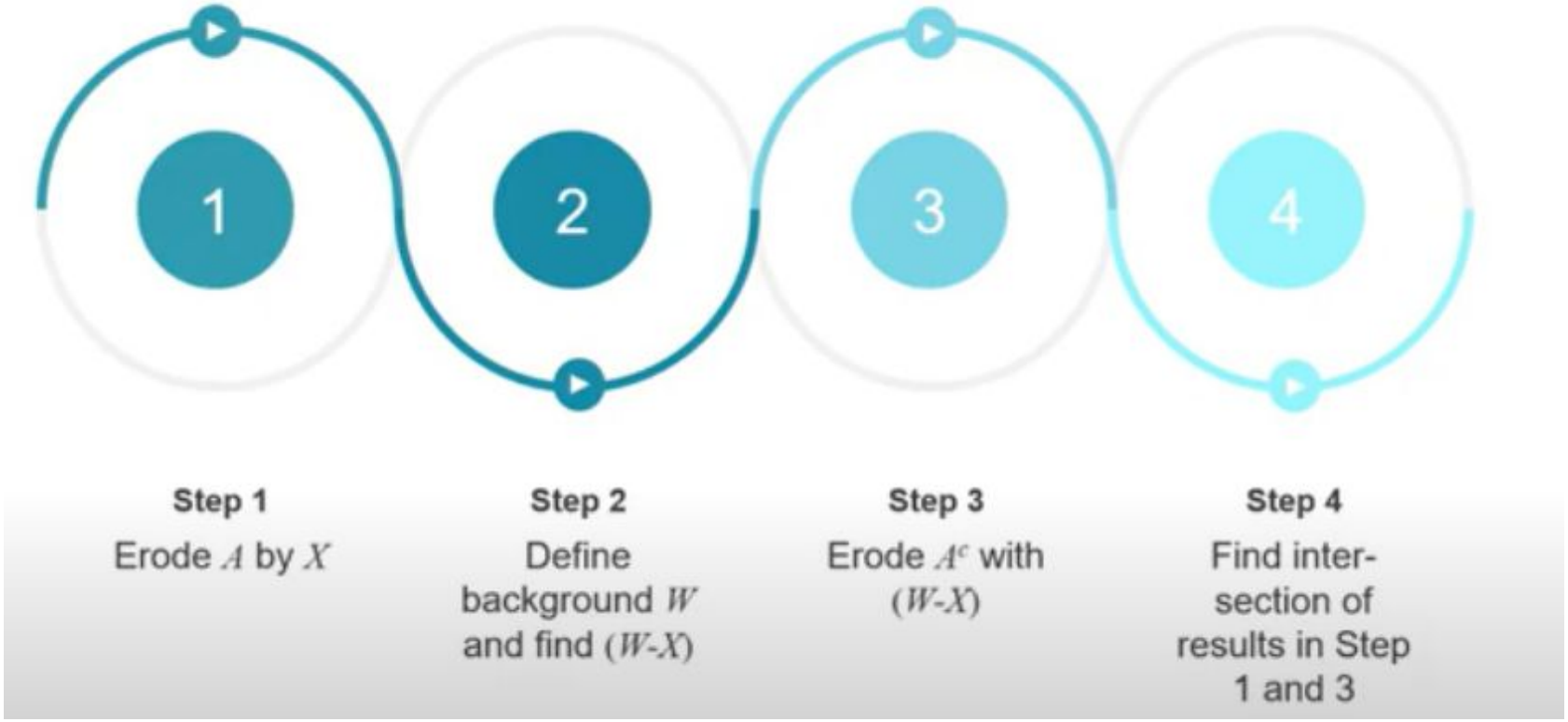
Hit and Miss - Example 2

Hit - or - Miss transform is given as

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

where A = *Set in which we want to find the location of object* X

B = *Set composed of* X *and its background* W



Hit and Miss Hit & Miss- Example 2

STEP 1: Erode A by B_1

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

A_{Pad}

x	1	x
1	1	0
x	0	0

X

0						

$A \ominus X$

Hit and Miss - Example 2

STEP 1: Erode A by B_1

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

A_{Pad}

x	1	x
1	1	0
x	0	0

X

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

$A \ominus X$

Hit and Miss - Example 2

Step-2 Define background W and find (W-X)

NOTE

In Hit-or-Miss transform, 1 represents foreground and 0 represents background for structuring element.

Hence, to find (W-X) complement each 1's and 0's of X. Keep don't care conditions as it is.

x	1	x
1	1	0
x	0	0

X

1	1	1
1	1	1
1	1	1

W

x	0	x
0	0	1
x	1	1

$W - X$

Hit and Miss - Example 2

STEP 3: Erode A^c with $(W\text{-}X)$

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

A^c

x	0	x
0	0	1
x	1	1

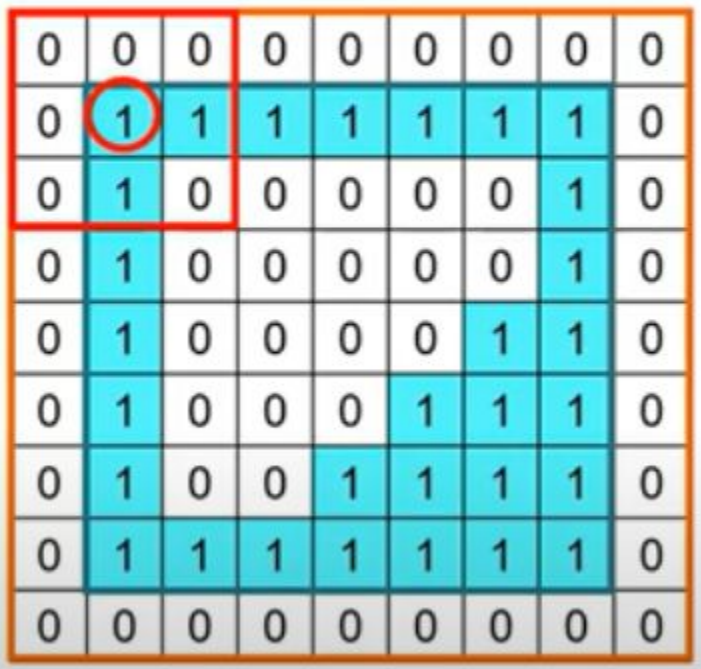
$W - X$

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

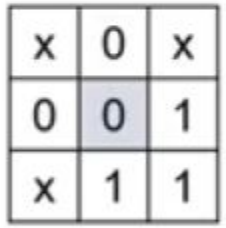
A^c_{Pad}

Hit and Miss - Example 2

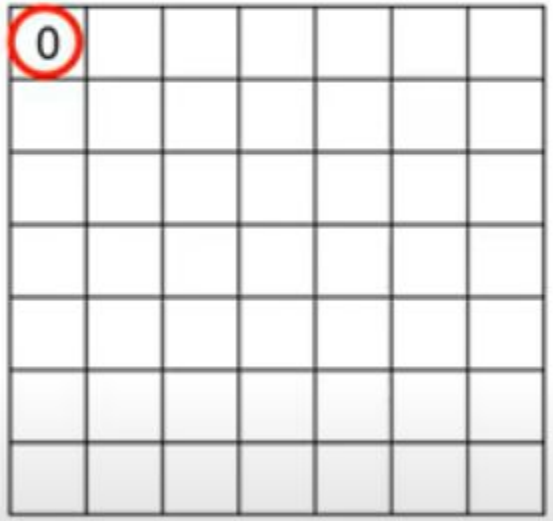
STEP 3: Erode A^c with $(W\text{-}X)$



A^c_{Pad}



$W - X$



$A^c \ominus (W - X)$

Hit and Miss - Example 2

STEP 3: Erode A^c with $(W\text{-}X)$

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

A^c_{Pad}

x	0	x
0	0	1
x	1	1

$W - X$

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

$A^c \ominus (W - X)$

Hit and Miss - Example 2

STEP 4: Find inter-section of results in Step 1 and 3

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

$A \ominus X$

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

$A^c \ominus (W - X)$

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

$A \circledast B$

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

Hit and Miss - Example 2

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

A

x	1	x
1	1	0
x	0	0

X

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

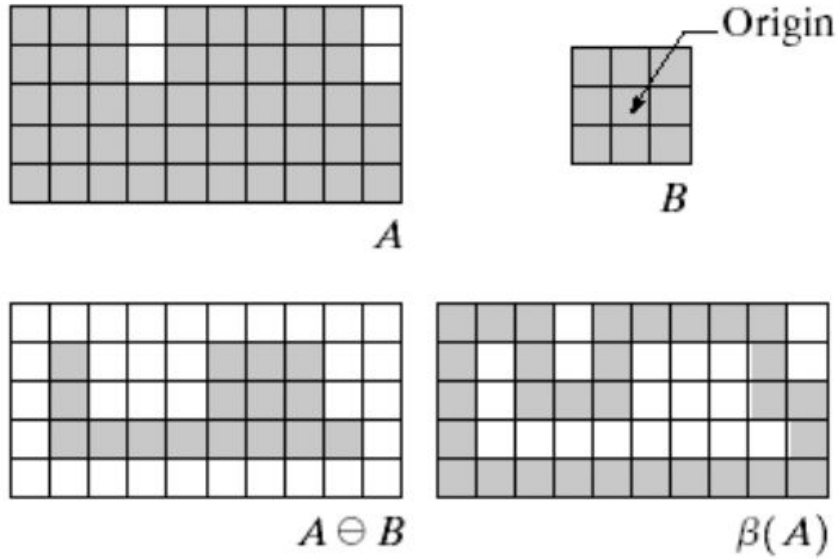
$A \circledast B$

Some basic Morphological algorithms

1. Boundary Extraction-

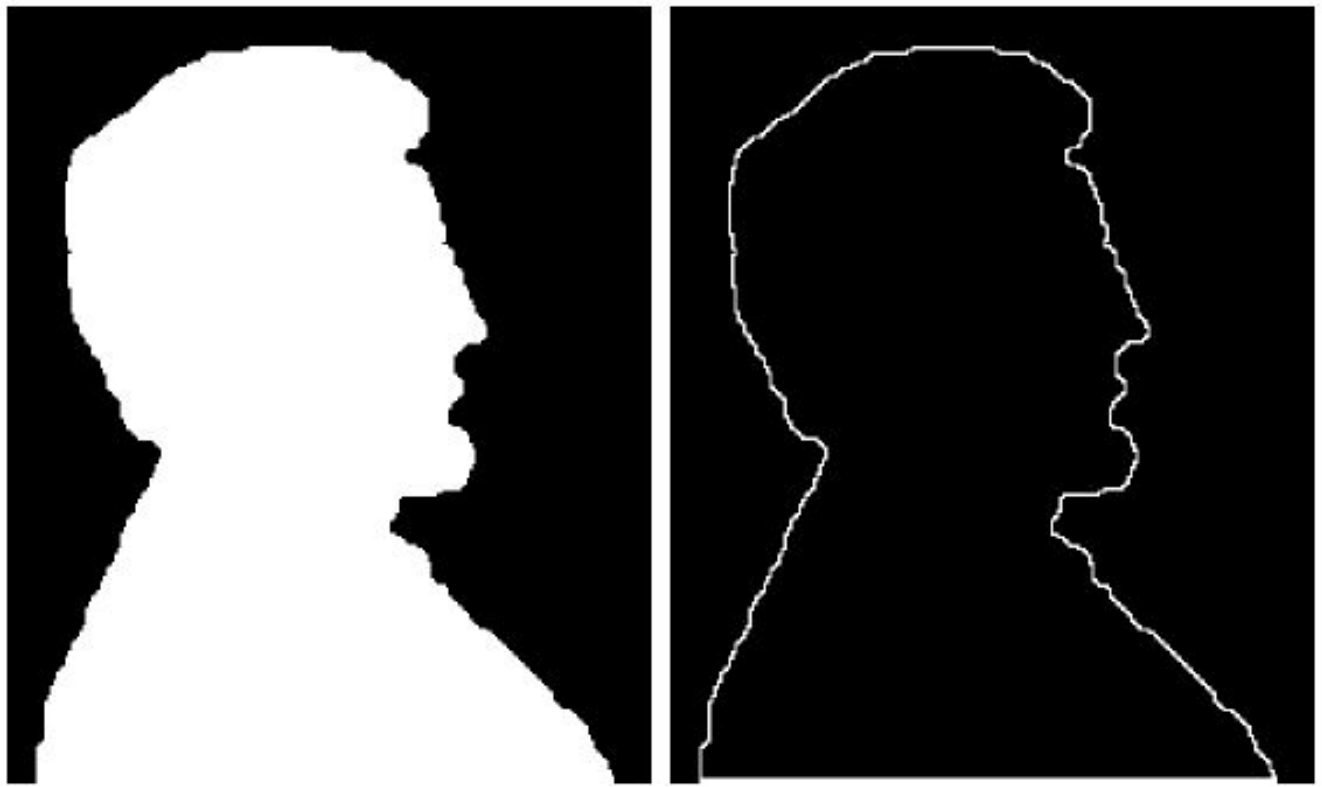
a b
c d

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.



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

Boundary Extraction-



a b

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

Matlab Code- Boundary extraction

```
pkg load image;
#Read input image
org_im = imread(coins.png');
figure 1, imshow(org_im); title('Original Image');
#Convert RGB image to Binary Image and Displayed as Input Image
A=im2bw(org_im);
se=strel('disk',1,0);
F=imerode(A,se);
figure 2, imshow(F);title("eroded");
figure 3, imshow(A-F); title("boundary");
```

Boundary extraction - Output