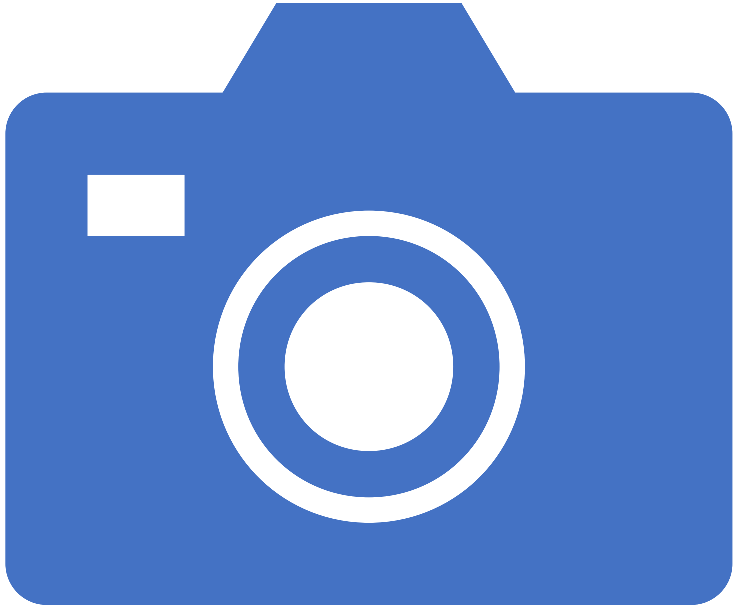


# Digital Image Processing

Dr. Mubashir Ahmad (Ph.D.)

# Introduction

- ✓ *Morphology is concerned with image analysis methods whose outputs describe image content (i.e. extract ‘meaning’ from an image).*
- ✓ ***Mathematical morphology*** *is a tool for extracting image components that can be used to represent and describe region shapes such as boundaries and skeletons.*
- ✓ *All morphology functions are defined for binary images, but most have natural extension to grey scale images.*
- ✓ *Morphological techniques are based on combinations of set operations and translation.*

# *Set operations and translation*

- In image processing, set operations and translation are two important concepts that are commonly used for various image manipulation and analysis tasks.

1. Set Operations: Set operations involve manipulating the pixel values or regions of interest in an image based on logical relationships or conditions. The most common set operations in image processing include union, intersection, difference, and complement.

- Union: The union of two images combines the pixel values from both images. It results in an image where the pixels correspond to the maximum value at each location.
- Intersection: The intersection of two images retains only the pixel values that exist in both images. It results in an image where the pixels correspond to the minimum value at each location.
- Difference: The difference between two images calculates the absolute or signed difference between corresponding pixel values. It highlights the dissimilarities or changes between the two images.

# *Set operations and translation*

- Complement: The complement operation inverts the pixel values of an image, where black becomes white and vice versa. It is useful for highlighting specific features or objects in an image.
- Set operations are often used for tasks such as image merging, image registration, region of interest selection, and image subtraction.

2. Translation: Translation refers to the process of shifting an image by a certain distance in the horizontal or vertical direction. It is a geometric transformation that preserves the shape and content of the image but changes its spatial position. Translation can be performed by applying a pixel shift to all the pixels in the image. Translation is used in various image processing applications, including image alignment, image registration, object tracking, and image stitching. It allows for aligning or matching corresponding features in different images or adjusting the position of objects within an image.

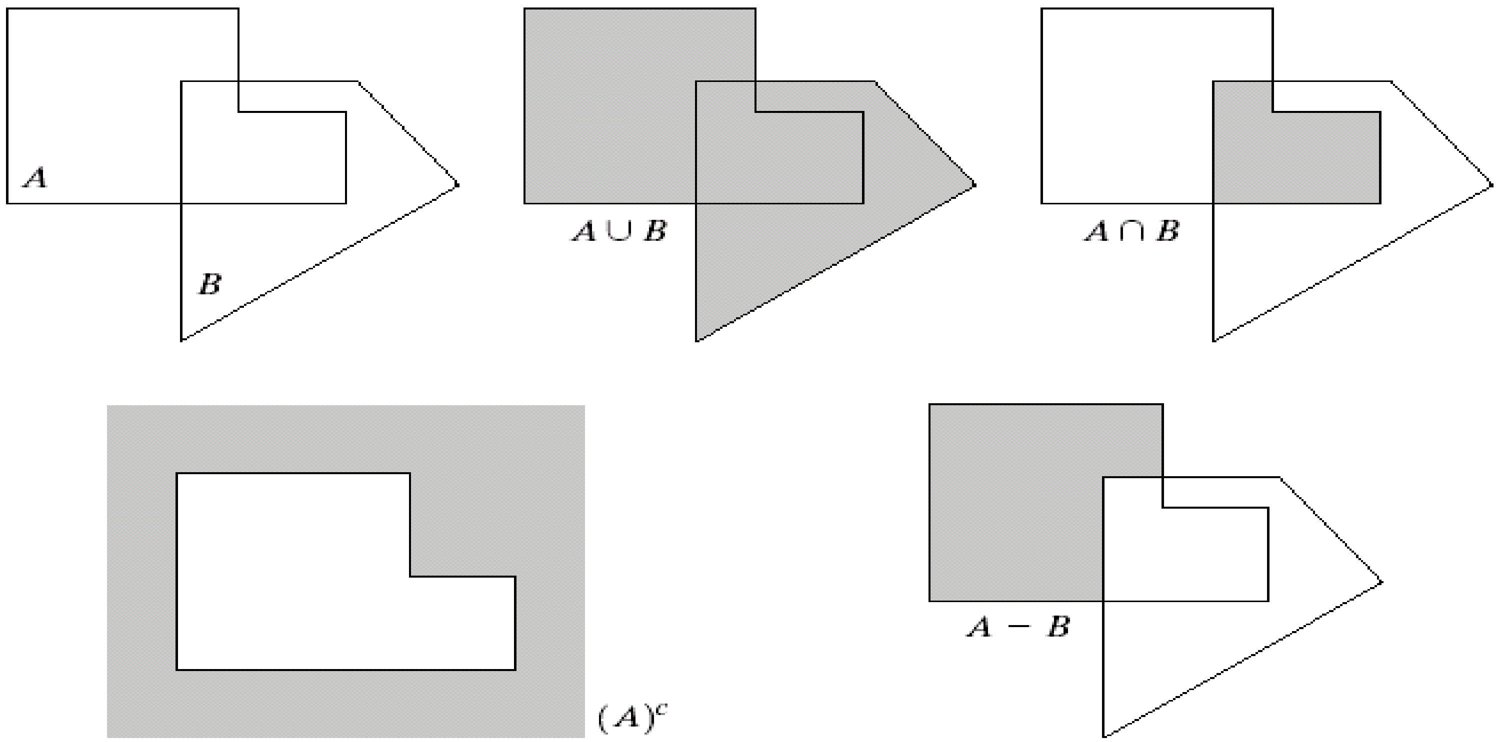
# Set Operations Binary Images

- ✓ ***A binary image is an image whose pixel values are 0 (representing black) or 1 (representing white, i.e. 255)***
- ✓ ***The usual set operations of complement, union, intersection, and difference can be defined easily in terms of the corresponding logic operations.***
- ✓ ***A & B stands for*** $A \cap B$; $A \mid B$ ***stands for*** $A \cup B$;

***- A stands for*** $A^c$ ; ***and*** $A-B = A \cap B^c$.

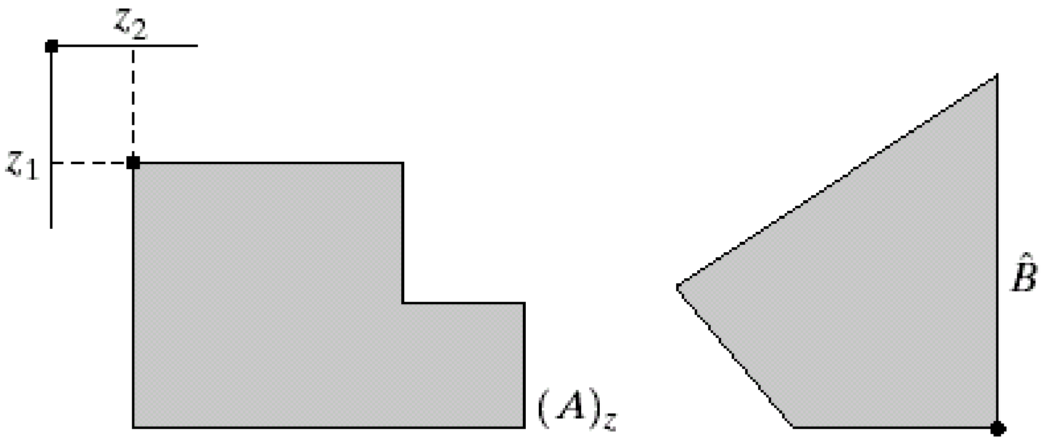
***Normally, Objects are represented in white, and background is black***

# Basic set operations




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 $B$.

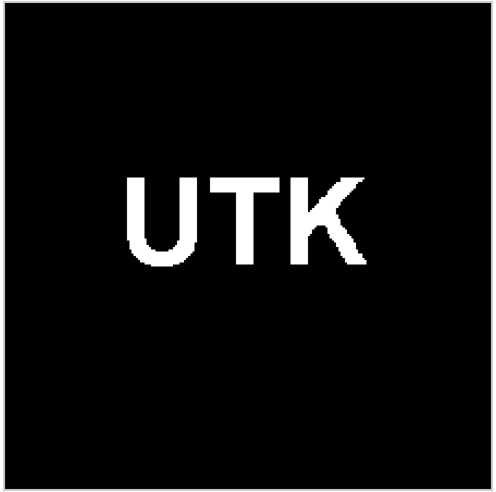



a b

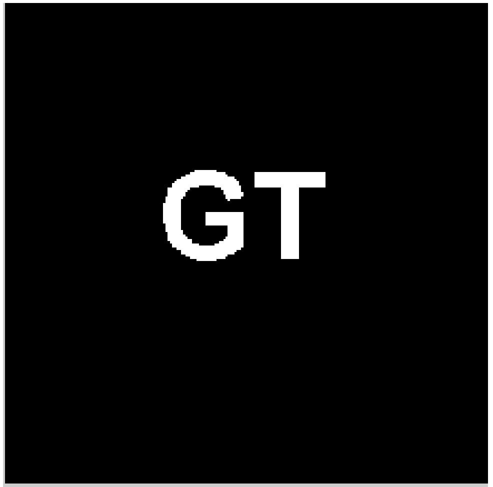
**FIGURE 9.2** (a) Translation of $A$ by $z$. (b) Reflection of $B$. The sets $A$ and $B$ are from Fig. 9.1.

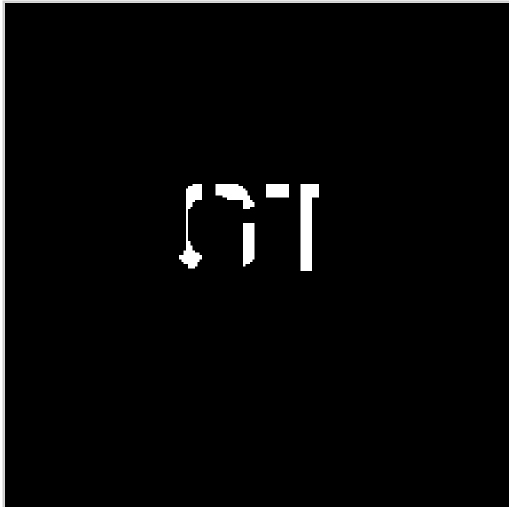
# Example

a =




b =



a & b

a | b

a - b

# Structuring Elements

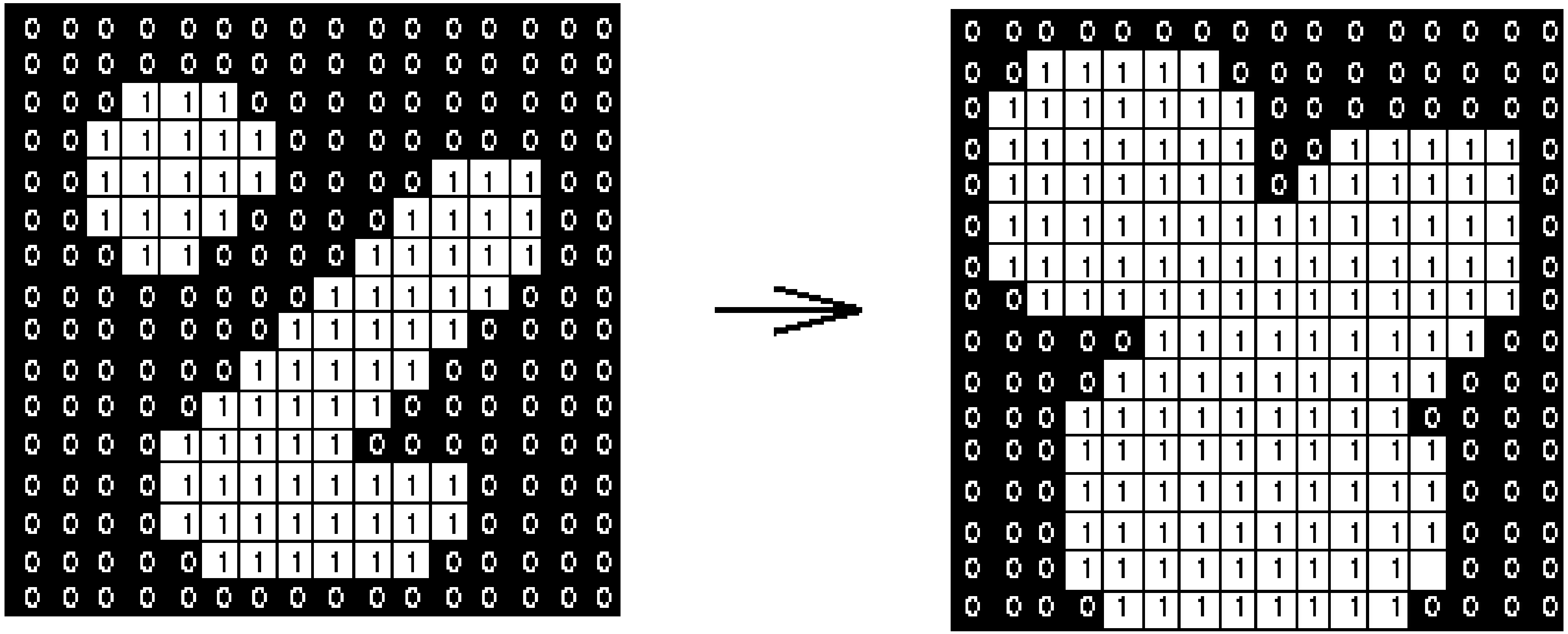
- ✓ *A morphing operation is based on the use of a filter-like binary pattern called the* __structuring element__ *(Mask) of the operation . For simplicity, the zero entries are often omitted:*

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| | 1 | |
| 1 | 1 | 1 |
| | 1 | |

# Dilation ⊕

✓ *Dilation is an operation used to grow or thicken objects in binary images. depending on the mask's <u>shape</u> and <u>size</u>*

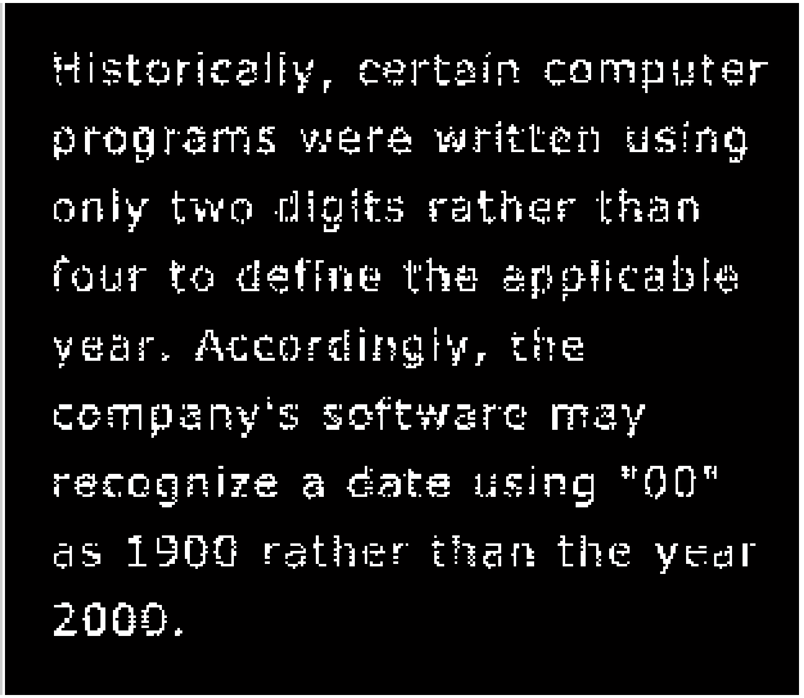


✓All neighbors should be <u>zeros</u>

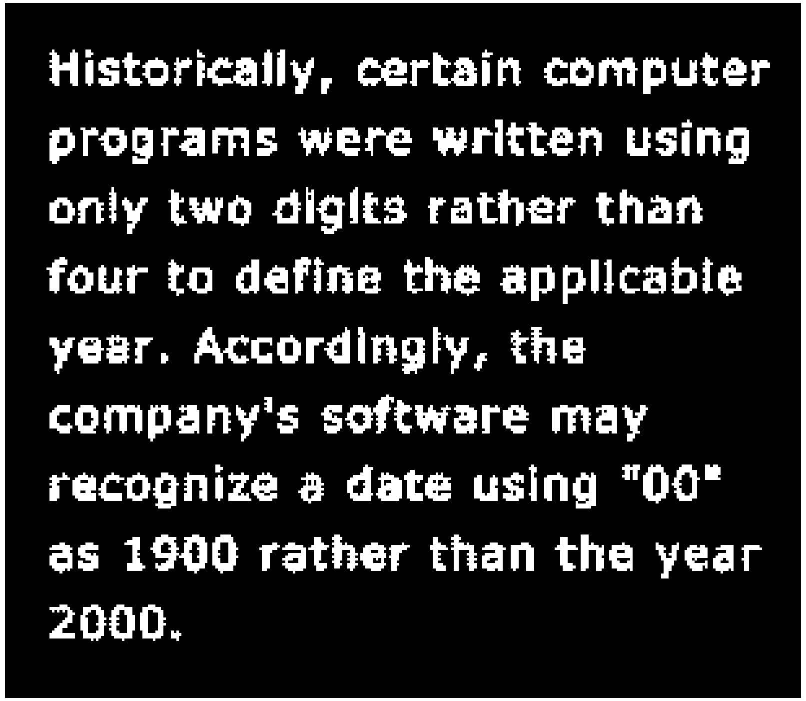
✓any <u>one</u> appears in the mask it makes the output <u>one</u> i.e.(255)

# Dilation applications - Example

- Dilation can be used for:
  - ✓ Enhancing broken/unclear textual errors.
  - ✓ Joint adjacent and closed objects




c




f

# Compute: A dilated ⊕ by B

Structuring element (SE) =

| |
|---|
| 1 |
| 1 |
| 1 |

1. Fully match = 1
2. Some match = 1
3. No match = 0

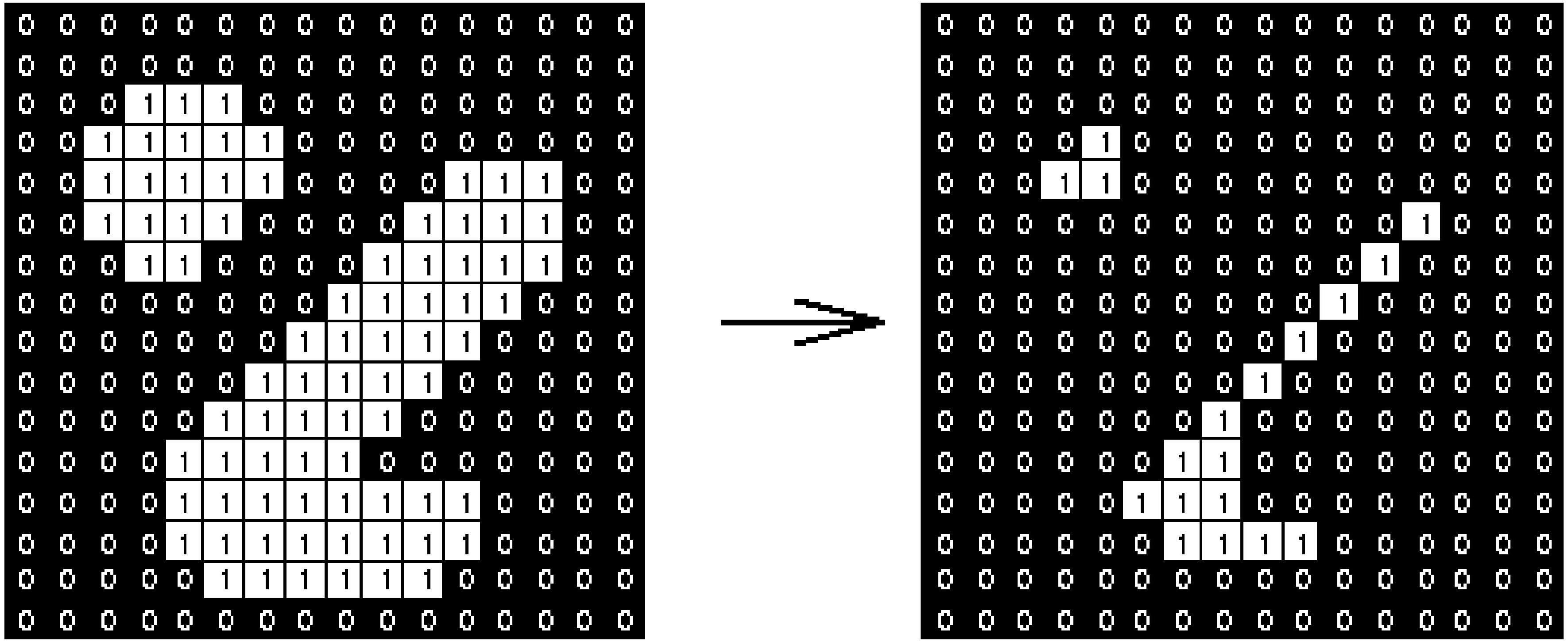
Here use the padding on top row.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |

# Erosion Θ

✓ *Erosion is an operation used to shrink or thinning objects in binary images depending on the mask's <u>shape</u> and <u>size</u>.*



✓All neighbors should be <u>ones</u>

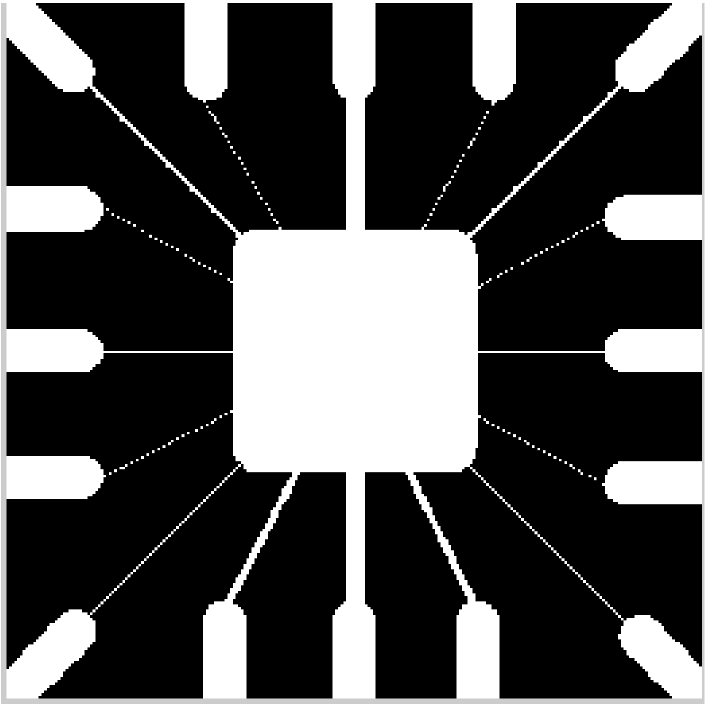
✓any zero appears makes the output zero

# Erosion applications - Example

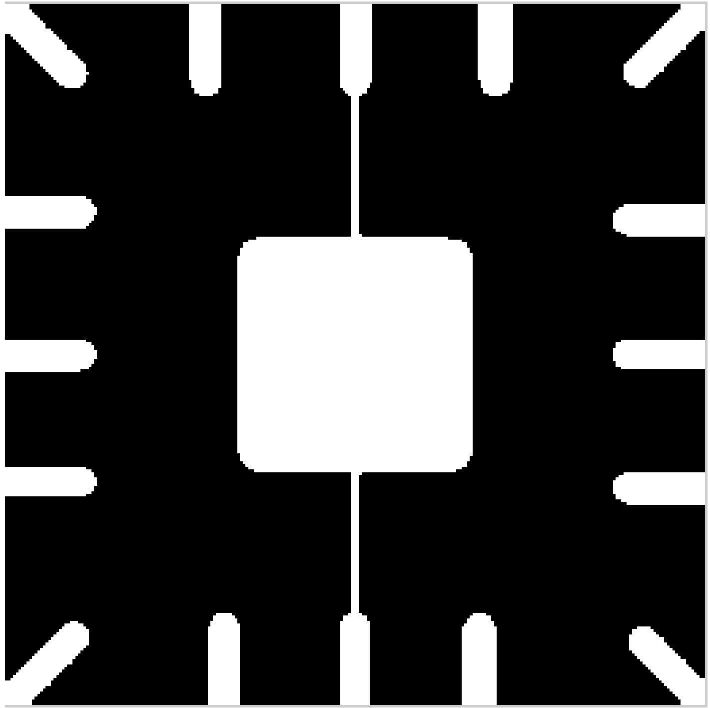
Erosion can be used to:

✓remove isolated features which may include noise or thin edges.

✓Disjoint objects



c



f

Structuring element (SE) =

| |
|---|
| 1 |
| 1 |
| 1 |

1. Fully match = 1
2. Some match = 0
3. No match = 0

Compute: $A^c$ eroded by B

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Now $A^c$

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |

# Combining Erosion & Dilation - Opening Morphology

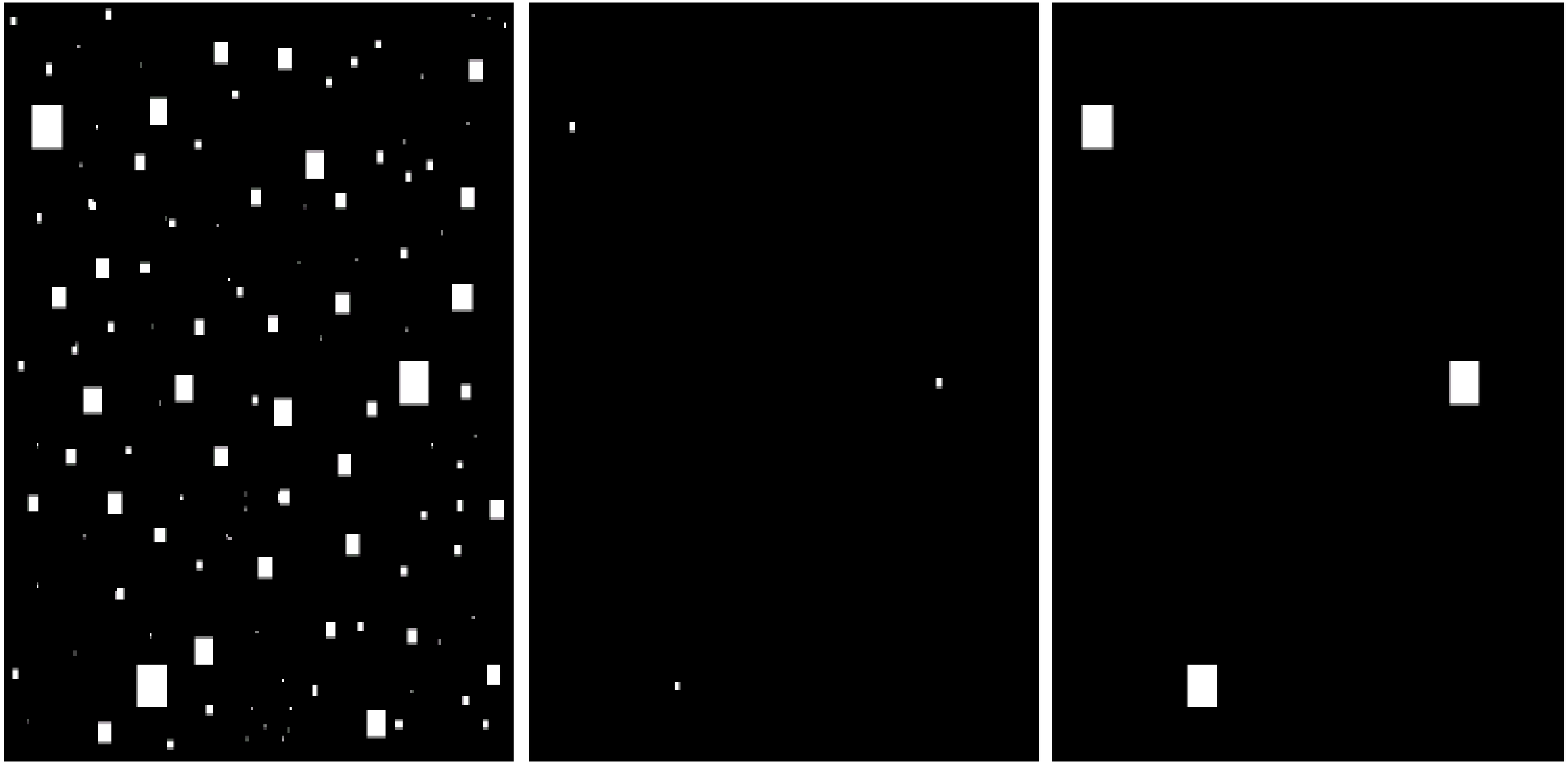
- ✓ *The* *opening* *operation* *<u>erodes</u>* *an image and then* *<u>dilates</u>* *the eroded image using the same structuring element for both operations, i.e.*

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

*A is the original image and B is the structuring element*

- ✓ *It removes regions of an object that cannot contain the structuring element, smoothes objects' contours, and breaks thin connections.*
- ✓ *Can be used to remove small objects in an image while preserving the shape and size of larger objects.*

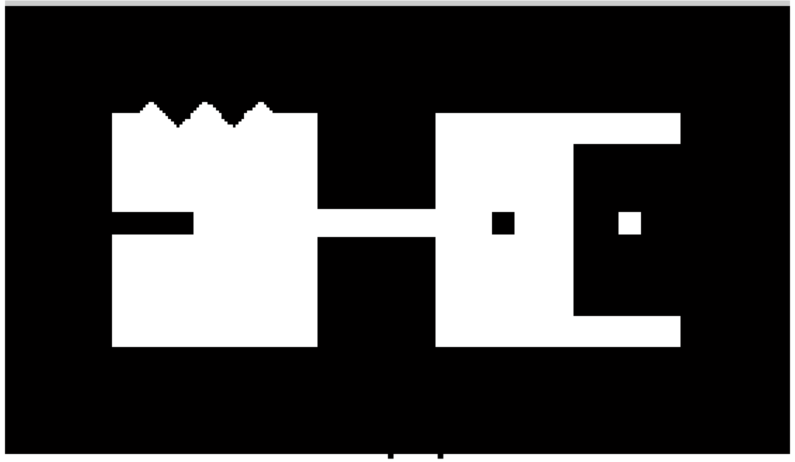
# Example Erosion + Dilation



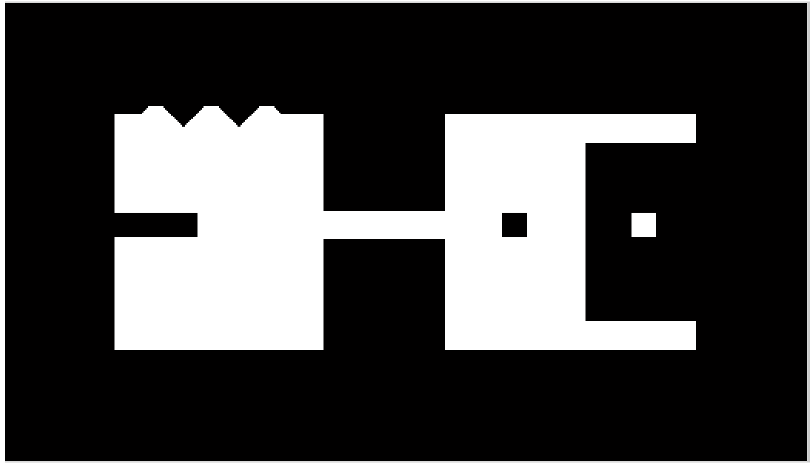
a b c

**FIGURE 9.7** (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

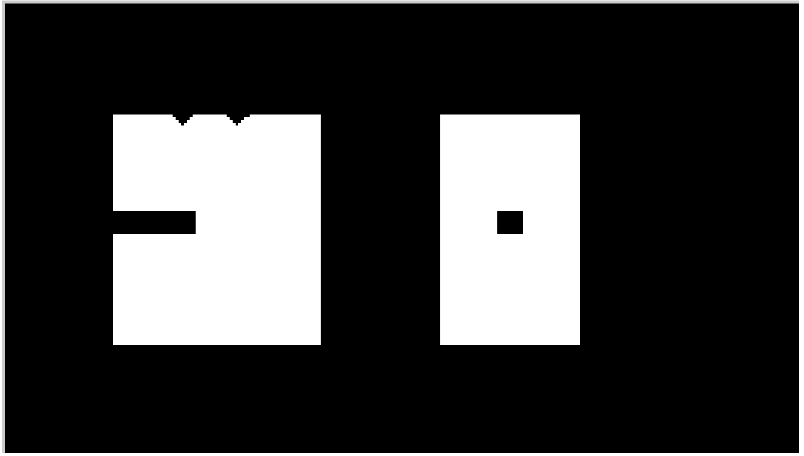
# Openning with the same element but different parameters



origin

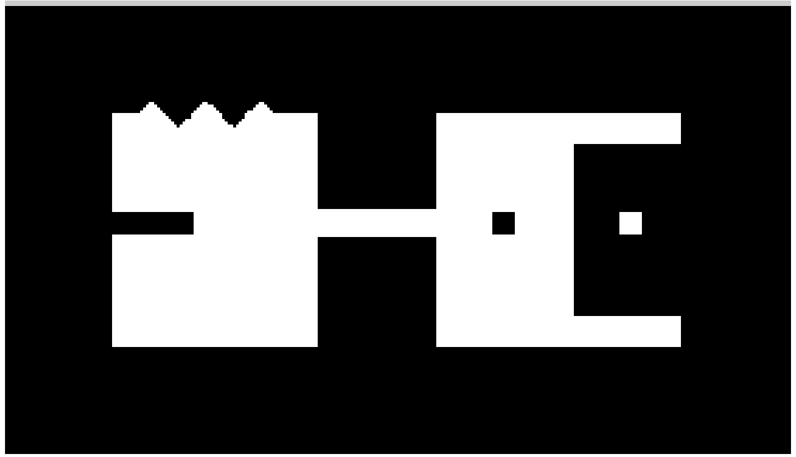


open, when we use SE=10x10

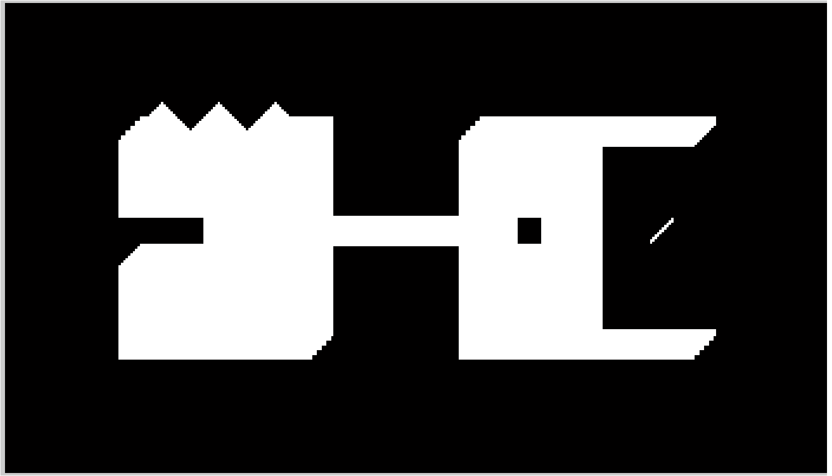


open, when we use SE=20x20

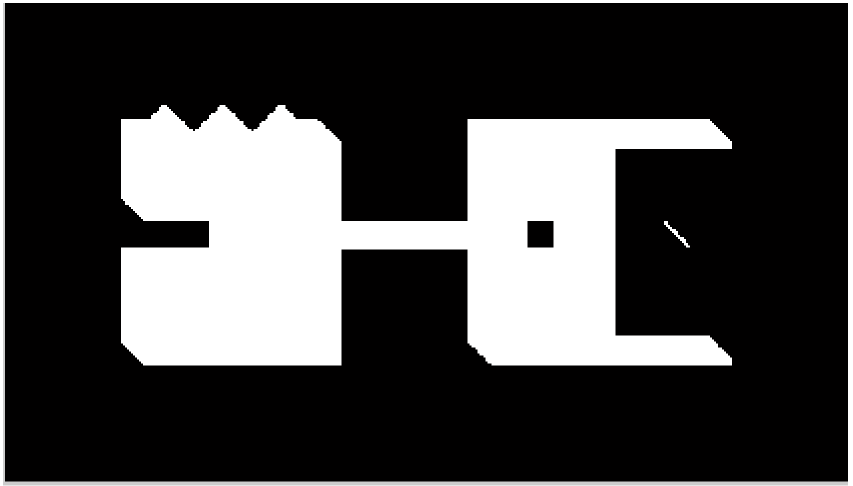
# *Opening with different structure elements*



origin



se=('line',20, 45)



se= ('line',20, -45)

# Combining Dilation & Erosion - Closing Morphology

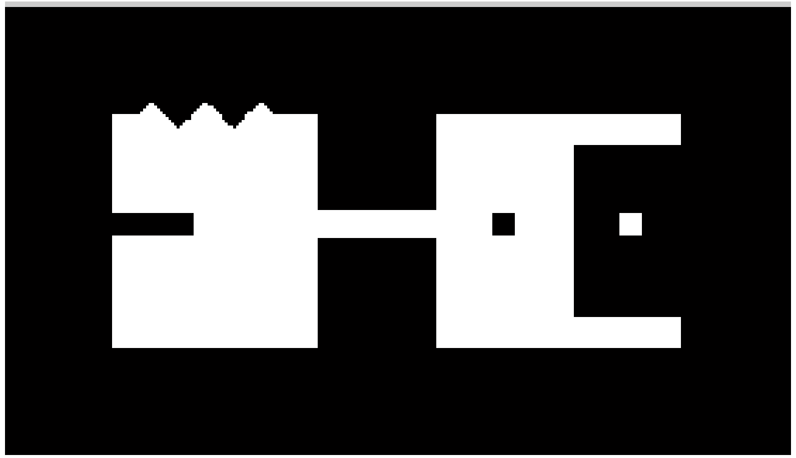
- ✓ *The closing operation dilates an image and then erodes the dilated image using the same structuring element for both operations, i.e.*

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

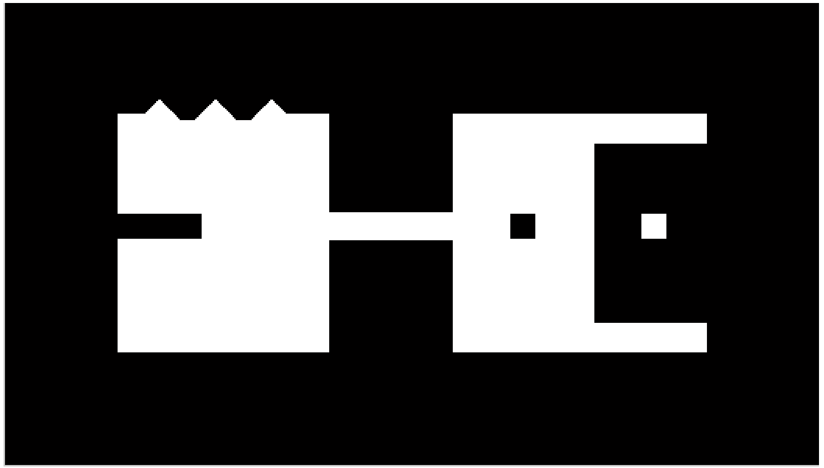
*A is the original image and B is the structuring element*

- ✓ *Fills holes that are larger than the structuring element, joins narrowing breaks, and fill gaps in contours.*
- ✓ *But like opening the closing operation smoothes objects contours.*

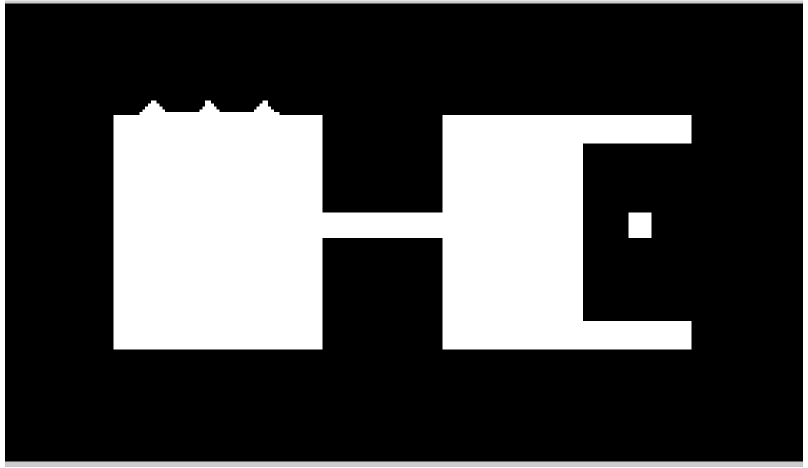
# Closing with the same element but different parameters



origin

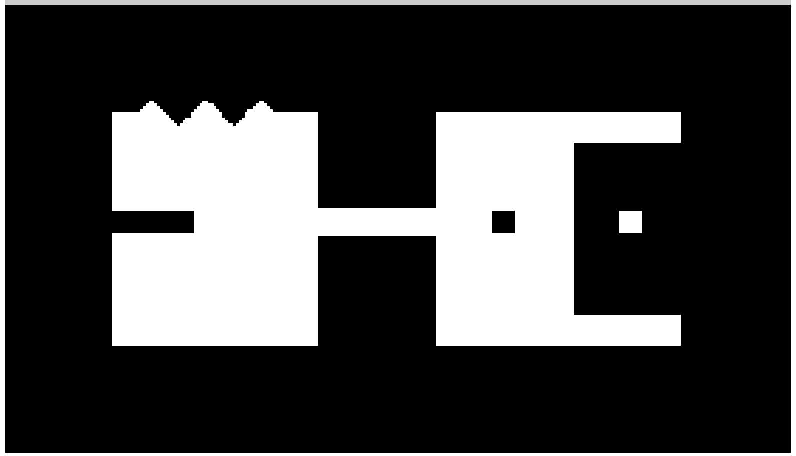


close, when SE=10x10

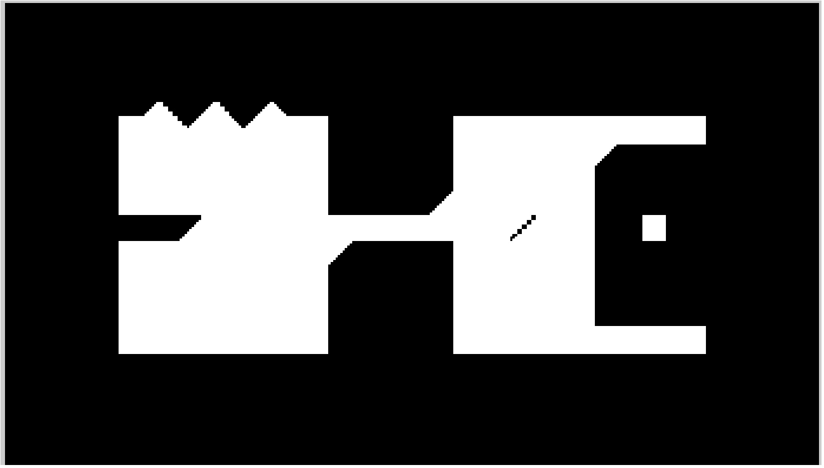


close, when SE=20x20

# *Closing with different structure elements*



origin



se=('line',20, 45)

se= ('line',20, -45)

# *Example – Combining open and close*

```
clear all
c=imread('noisy_fingerprint.tif');
se= strel('square',3);
f=imopen(c,se);
g=imclose(f,se);
```

c

f

g

Note, noise has been removed completely.

# 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.

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

# Example 1

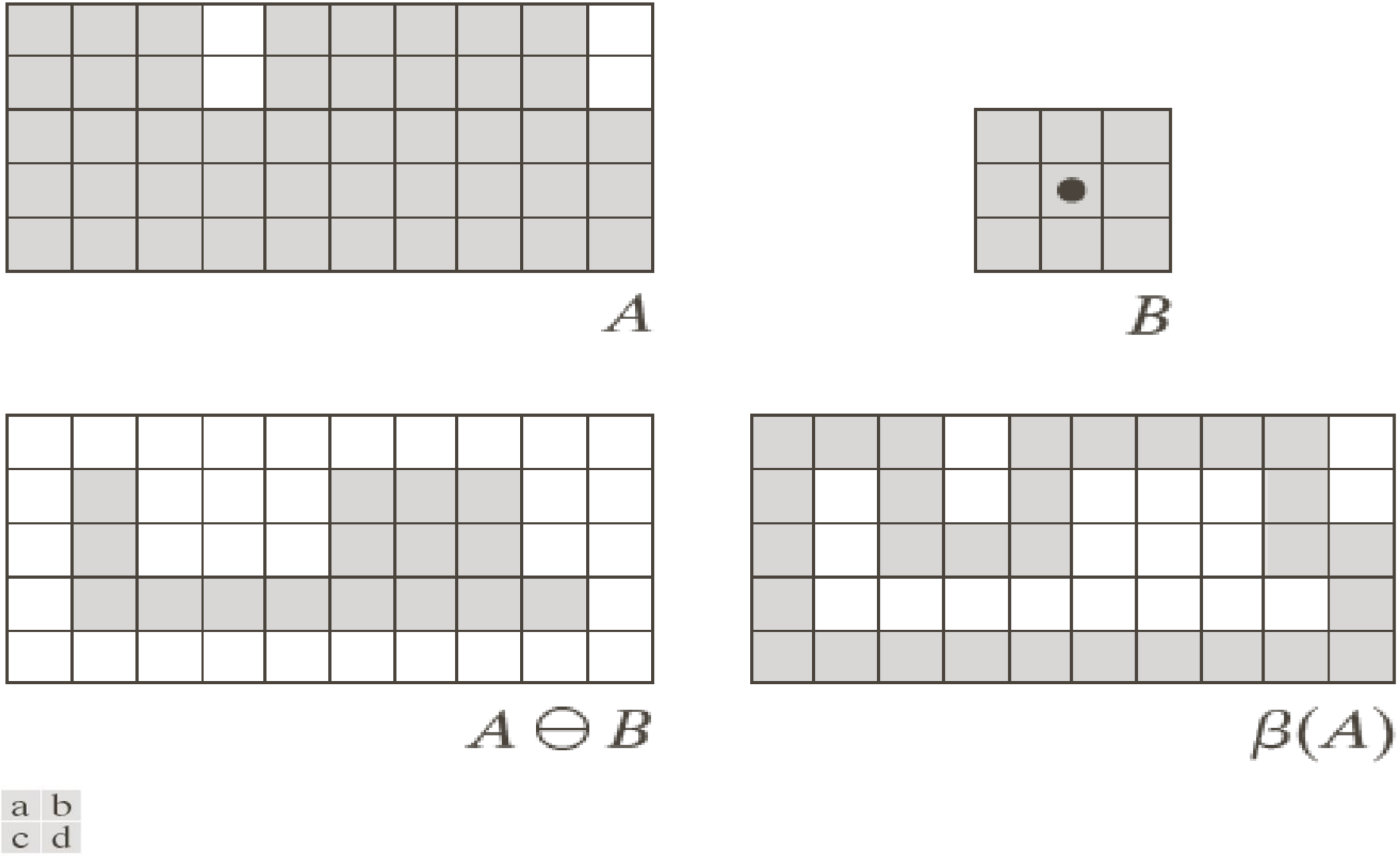



**FIGURE 9.13** (a) Set $A$. (b) Structuring element $B$. (c) $A$ eroded by $B$. (d) Boundary, given by the set difference between $A$ and its erosion.

# 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).

# Morphology operators on grey images

**Erosion dilation, open and close can be used on grey images.**

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

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

# Morphology operators on grey images

**Erosion dilation, open and close can be used on grey images.**

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

F =

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 7 | 4 | 5 | 6 | 7 | 0 |
| 0 | 6 | 4 | 7 | 8 | 6 | 0 |
| 0 | 7 | 9 | 8 | 7 | 7 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B =

| | | |
|---|---|---|
| 0 | 3 | 0 |
| 3 | 3 | 3 |
| 0 | 3 | 0 |

Due to clamping effect -ve values 0

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 3 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# Morphology operators on grey images

**Erosion dilation, open and close can be used on grey images.**

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

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 7 | 4 | 5 | 6 | 7 | 0 |
| 0 | 6 | 4 | 7 | 8 | 6 | 0 |
| 0 | 7 | 9 | 8 | 7 | 7 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B =

| | | |
|---|---|---|
| 0 | 3 | 0 |
| 3 | 3 | 3 |
| 0 | 3 | 0 |

flipping to 180°

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 10 | 7 | 8 | 9 | 10 | 3 |
| 10 | 10 | 10 | 10 | 11 | 10 | 10 |
| 9 | 10 | | | | | 9 |
| 10 | | | | | | 10 |
| 3 | 10 | 12 | 11 | 10 | 10 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 |

# Morphology operators on grey images

A

A dilated

A eroded

B=A opened

B closed

# Morphological Smoothing

- Opening suppresses bright details smaller than the specified SE, and closing suppresses dark details.

- Opening and closing are used often in combination as morphological filters for image smoothing and noise removal.

$$g = ((f \circ b) \bullet b)$$

# Opening

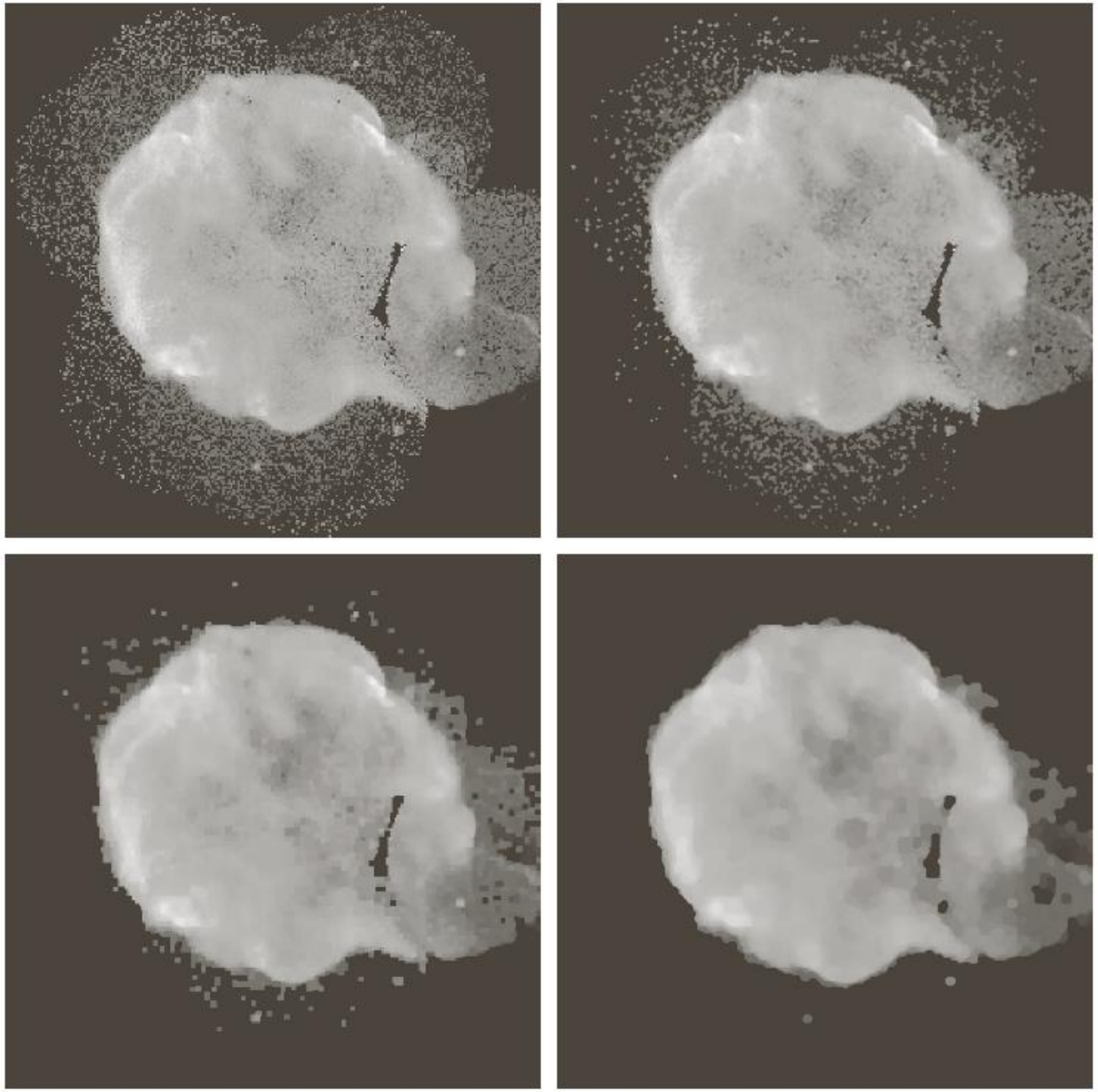
- Opening is a morphological operation that consists of two consecutive operations: erosion followed by dilation. It is typically used to remove small objects, smooth the boundaries of larger objects, and separate objects that are touching or overlapping.
- The opening operation involves the following steps:
- The image is eroded by a structuring element, which causes the foreground regions to shrink and thin.
- The eroded image is then dilated using the same structuring element, which expands the remaining regions but tends to keep them smaller than their original size.
- Opening is effective in removing noise, filling small holes, and breaking thin connections between objects. It preserves the overall shape and structure of larger objects while removing smaller details.

# Closing

2.Closing is the reverse operation of opening. It also consists of two consecutive operations: dilation followed by erosion. Closing is commonly used to close small gaps or holes in objects, connect broken lines or curves, and merge nearby objects that are separated by narrow gaps.

- The closing operation involves the following steps:
- The image is dilated by a structuring element, which causes the foreground regions to expand and fill gaps.
- The dilated image is then eroded using the same structuring element, which reduces the expanded regions but tends to keep them larger than their original size.
- Closing helps in smoothing object boundaries, connecting broken or disconnected structures, and filling in small gaps or holes within objects.

# Morphological Smoothing



a b
c d

**FIGURE 9.38** (a) $566 \times 566$ 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.)

# Hit-and-Miss Transform

- The hit-and-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.
- Very simple object recognition.
- Input:
  - Binary image
  - Structuring element, containing 0s and 1s!!

- 1s should match foreground
- 0s should match background
- Blanks "don't care"

| | | |
|---|---|---|
| | 1 | |
| 0 | 1 | 1 |
| 0 | 0 | |

# Hit-and-miss Transform

- Similar to Pattern Matching:
- 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-and-miss transformation

- General notation

structure element (SE): $B = (B_1, B_2)$

e.g., $B_1 = X$ (object);
$B_2 = W\text{-}X$ (i.e. complement of $B_1$ SE) (background)

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

This set contains all the (origin) points, at which, $B_1$ found a match ("hit") in A and $B_2$ found a match in $A^c$, simultaneously.

# Thinning

1. Used to remove selected foreground pixels/particular pattern 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

- Let ***B*** be a particular pattern or SE and ***I*** be an image then

$$\mathrm{thin}(I,B) = I - \mathrm{HitAndMiss}\,(I,B)$$

- If foreground and background fit the structuring element exactly, then the pixel at the origin of the SE is set to 0

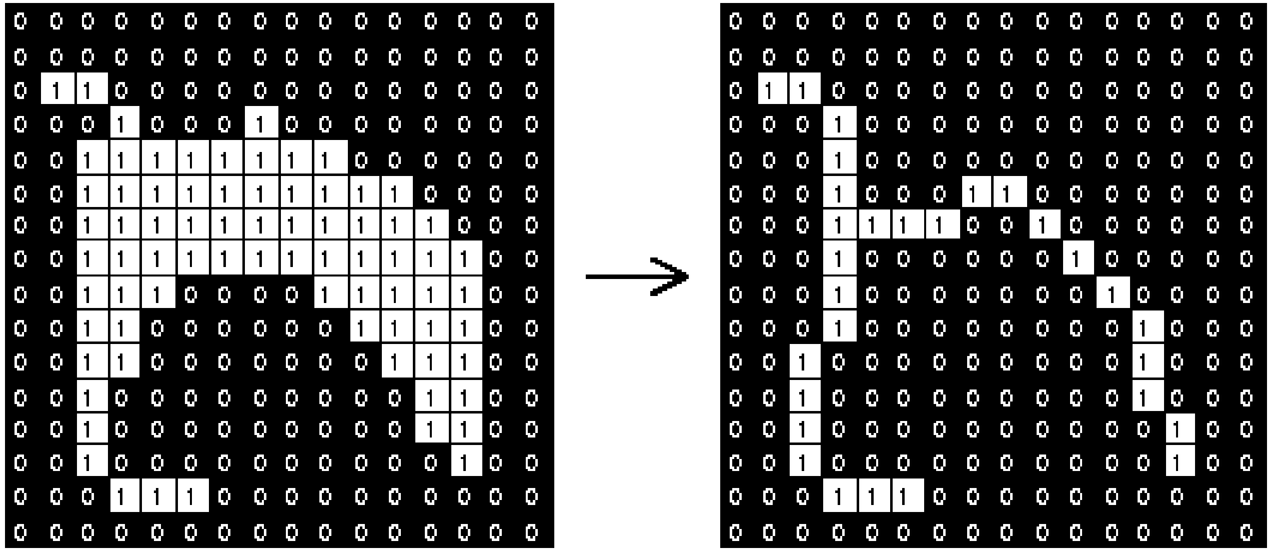
- Note that the value of the SE at the origin is 1 or *don't care*!

# Example Thinning

| | | |
|---|---|---|
| 0 | 0 | 0 |
| | 1 | |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| | 0 | 0 |
| 1 | 1 | 0 |
| | 1 | |

We use two Hit-and-miss Transforms



# Thickening

- Used to grow selected regions of foreground pixels.

# Definition Thickening

- Let ***B*** be a particular pattern or SE and ***I*** be an image then;

$$\text{thicken}\left(I, B\right) = I + \text{HitAndMiss}\left(I, B\right)$$

- 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*!

# Example Thickening

| | | |
|---|---|---|
| 1 | 1 | |
| 1 | 0 | |
| 1 | | 0 |

| | | |
|---|---|---|
| | 1 | 1 |
| | 0 | 1 |
| 0 | | 1 |