Marwadi
University

IMAGE PROCESSING
01CE0507

Chap 7
Image Morphology

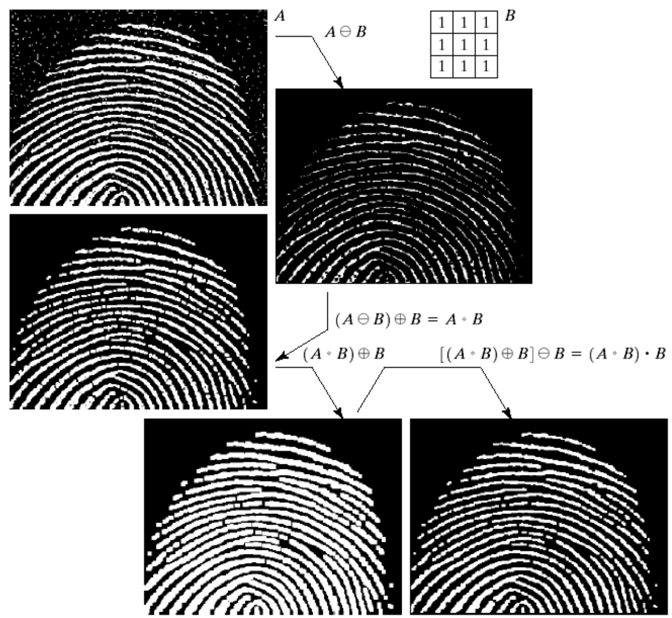
Prof. Urvi Y. Bhatt
Department of Computer Engineering

Key Steps Of Digital Image Processing

Image Restoration

Morphological Processing

Image Enhancement



Segmentation

Image Acquisition

Object Recognition

Problem Domain

Representation & Description

Colour Image Processing

Image Compression

Morphological Image Processing

- The word 'Morphology' generally represents a branch of biology that deals with the form and structure of animals and plants.
- However, we use the same term in 'mathematical morphology' to **extract image components useful in representing region shape, boundaries, etc**.

Morphological Image Processing (Cont.)

- Morphological Operations is a broad set of image processing operations that process digital images based on their shapes.
- In a morphological operation, each image pixel is corresponding to the value of other pixel in its neighborhood.

Morphological Image Processing (Cont.)

- Idea
  - An approach for processing digital image based on its shape
  - A mathematical tool to investigate geometric structure in image
- Morphological operators: are used to change image data to reflect new geometric structure.

Basic Concepts

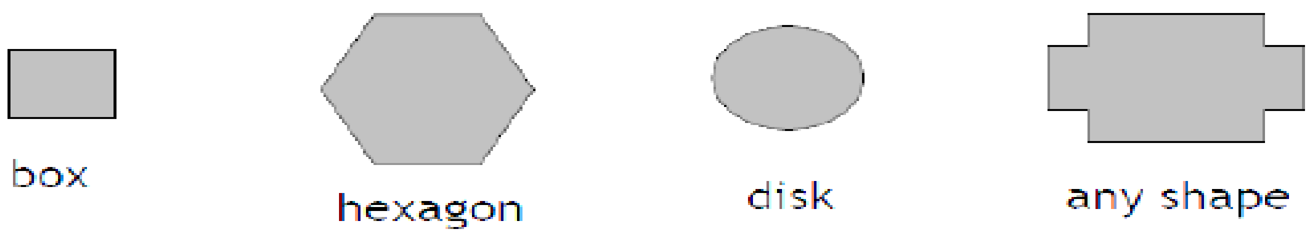
- All morphological processing operations are based on mentioned terms.
  - **Structuring Element:** It is a matrix or a small-sized template that is used to traverse an image. The structuring element is positioned at all possible locations in the image, and it is compared with the connected pixels. It can be of any shape.
  - **Fit:** When all the pixels in the structuring element cover the pixels of the object, we call it Fit.
  - **Hit:** When at least one of the pixels in the structuring element cover the pixels of the object, we call it Hit.
  - **Miss:** When no pixel in the structuring element cover the pixels of the object, we call it miss.

Basic Concepts (Cont.)

- Structure Element: The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one

A structuring element is a shape mask used in the basic morphological operations.

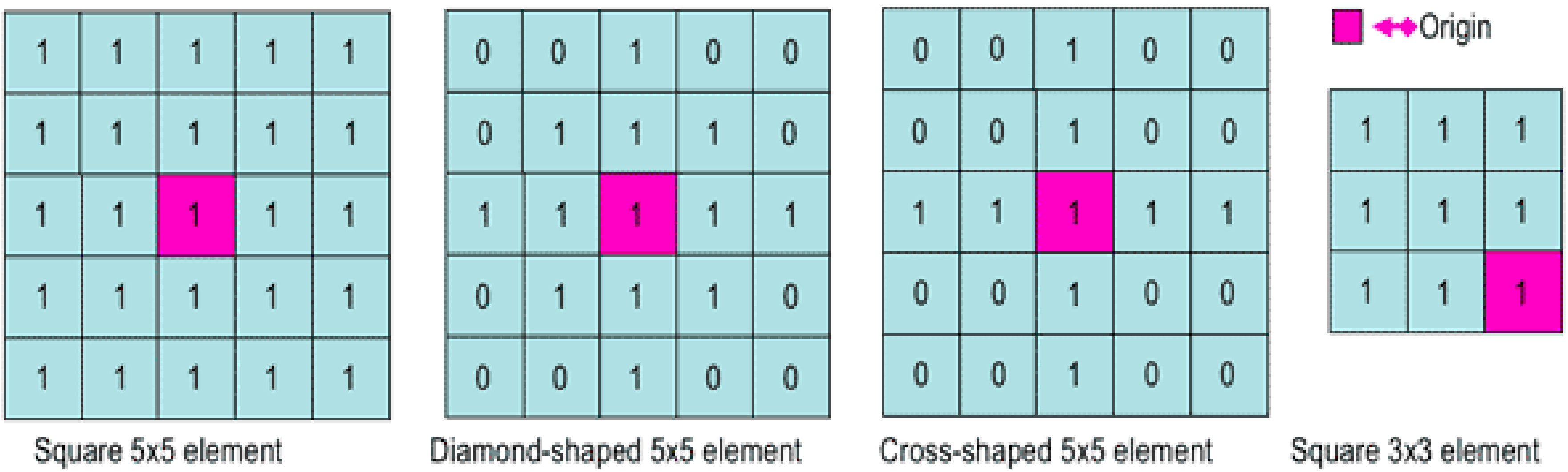
They can be any shape and size that is digitally representable, and each has an origin.



box(length,width)

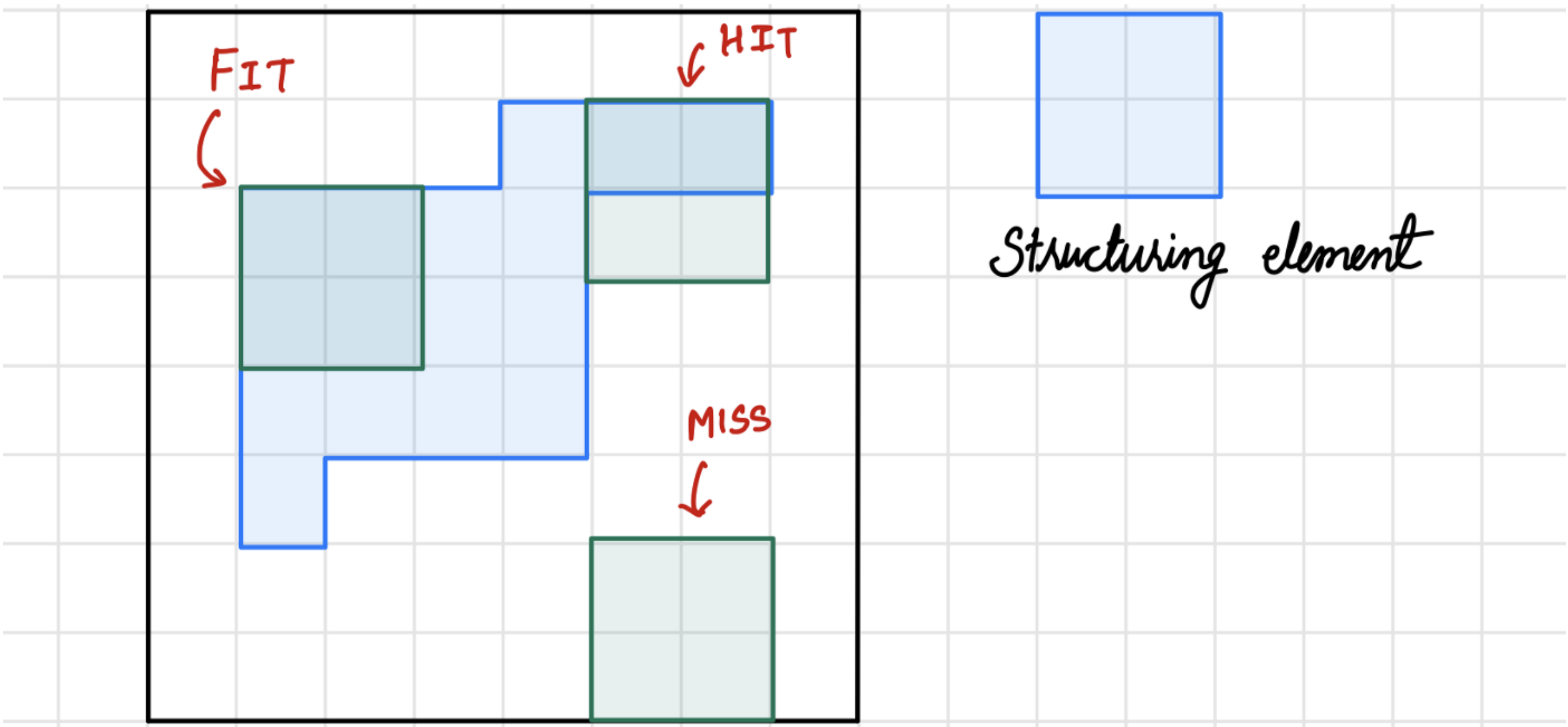
disk(diameter)

Basic Concepts (Cont.)

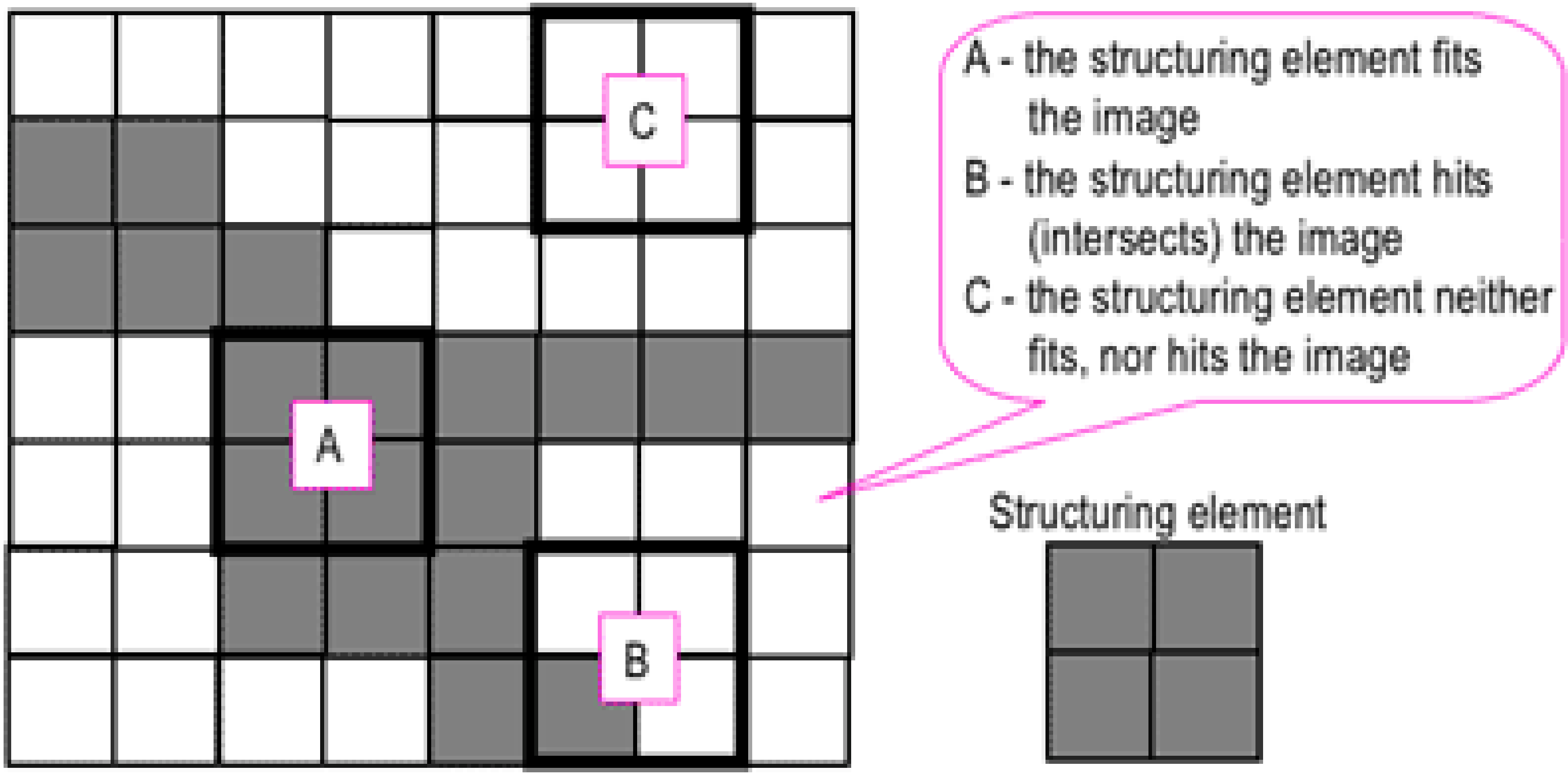


Examples of simple structuring elements.

Basic Concepts (Cont.)



Basic Concepts (Cont.)

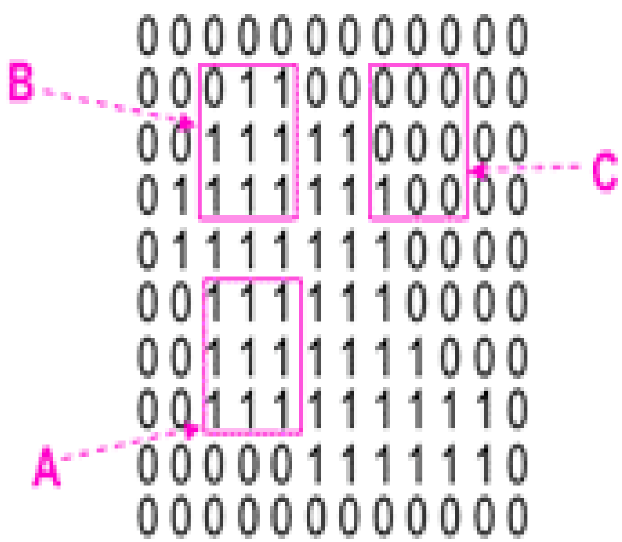


Basic Concepts (Cont.)

- Fitting and hitting of a binary image with structuring elements
  - The structuring element is said to **fit** the image if, for **each of its pixels set to 1, the corresponding image pixel is also 1**.
  - Similarly, a structuring element is said to **hit**, or intersect, an image if, **at least for one of its pixels set to 1 the corresponding image pixel is also 1.**

Basic Concepts (Cont.)

- Fit = all 1 should match
- Hit = At least one 1 should match
- Zero-valued pixels of the structuring element are ignore



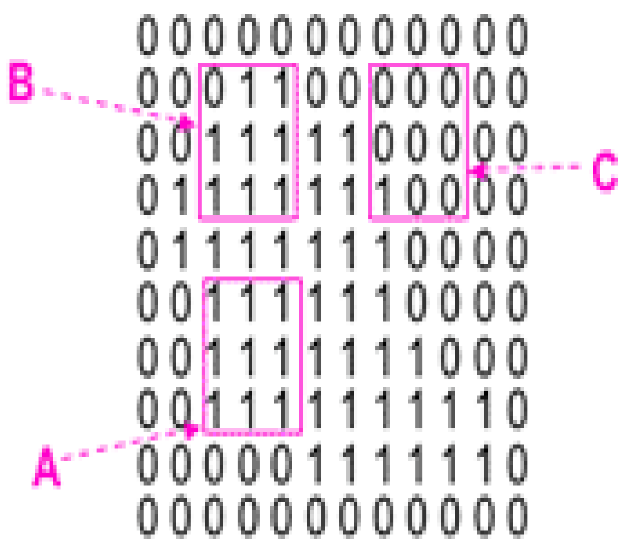
$$s_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$s_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

		A	B	C
fit	s_1	yes	no	no
	s_2	yes	yes	no
hit	s_1	yes	yes	yes
	s_2	yes	yes	no

Basic Concepts (Cont.)

- Fit = all 1 should match
- Hit = At least one 1 should match
- Zero-valued pixels of the structuring element are ignore

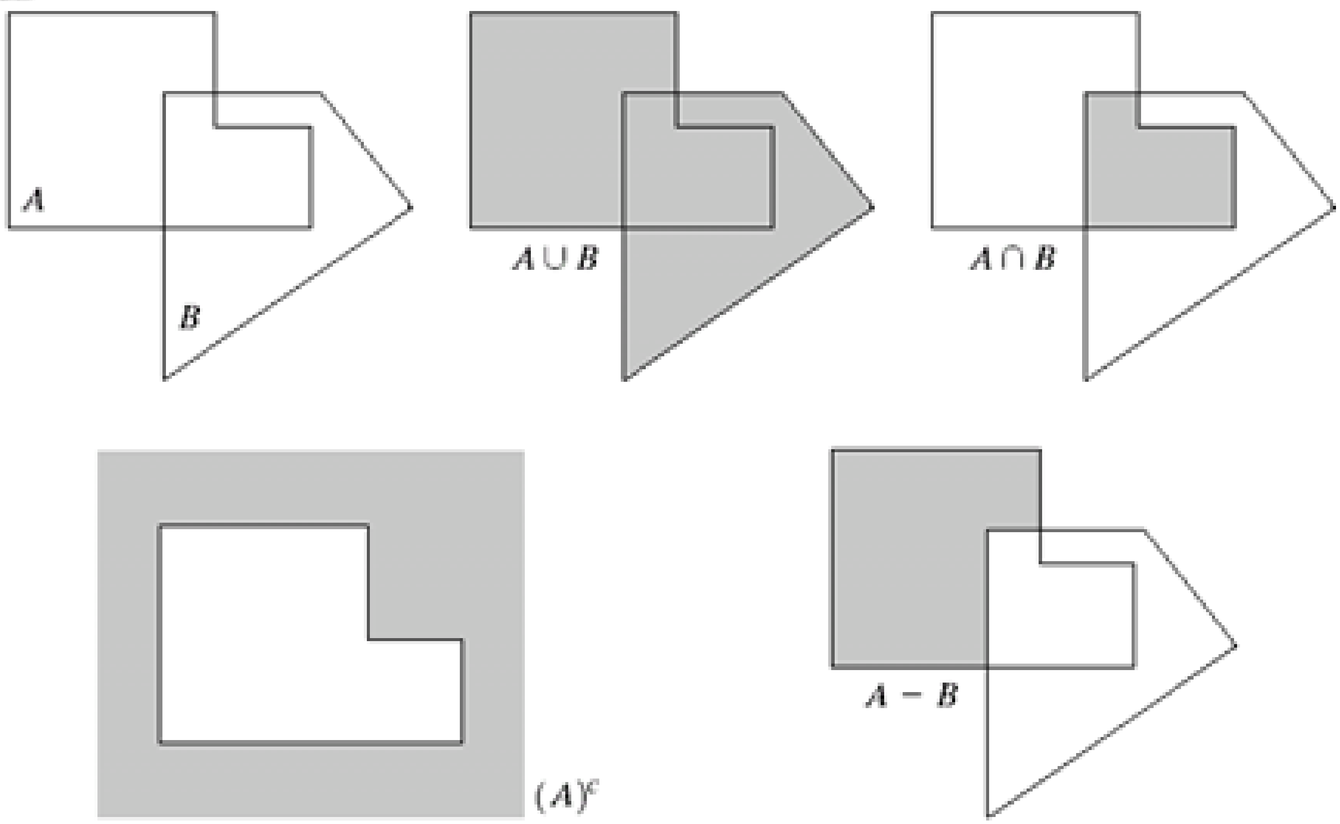


$$s_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$s_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

		A	B	C
fit	s_1	yes	no	no
	s_2	yes	yes	no
hit	s_1	yes	yes	yes
	s_2	yes	yes	no

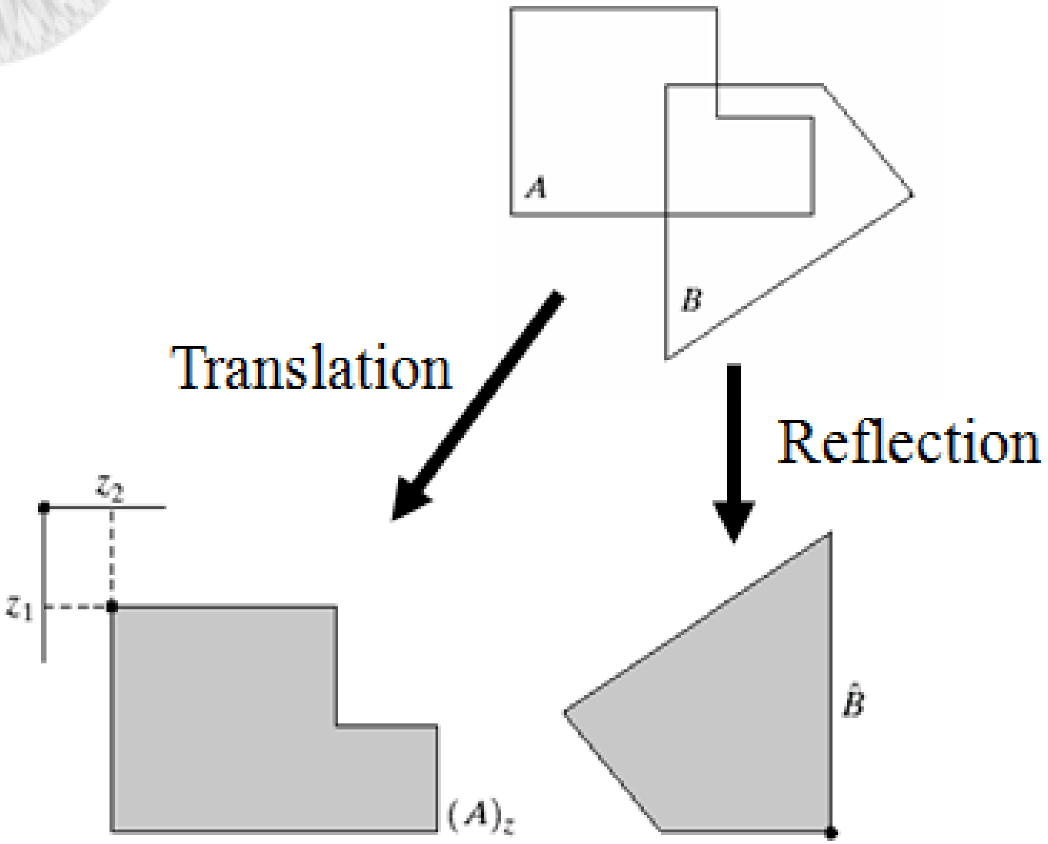
Some Basic Concepts from Set Theory



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*.

Some Basic Concepts from Set Theory (Cont.)



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.

Some Basic Concepts from Set Theory (Cont.)

TABLE 9.1 The three basic logical operations.

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

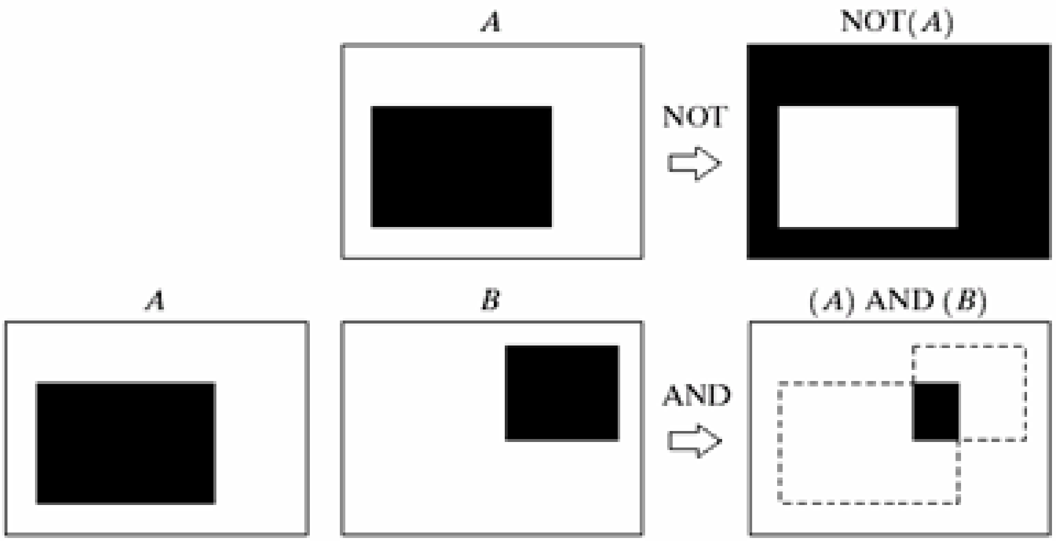
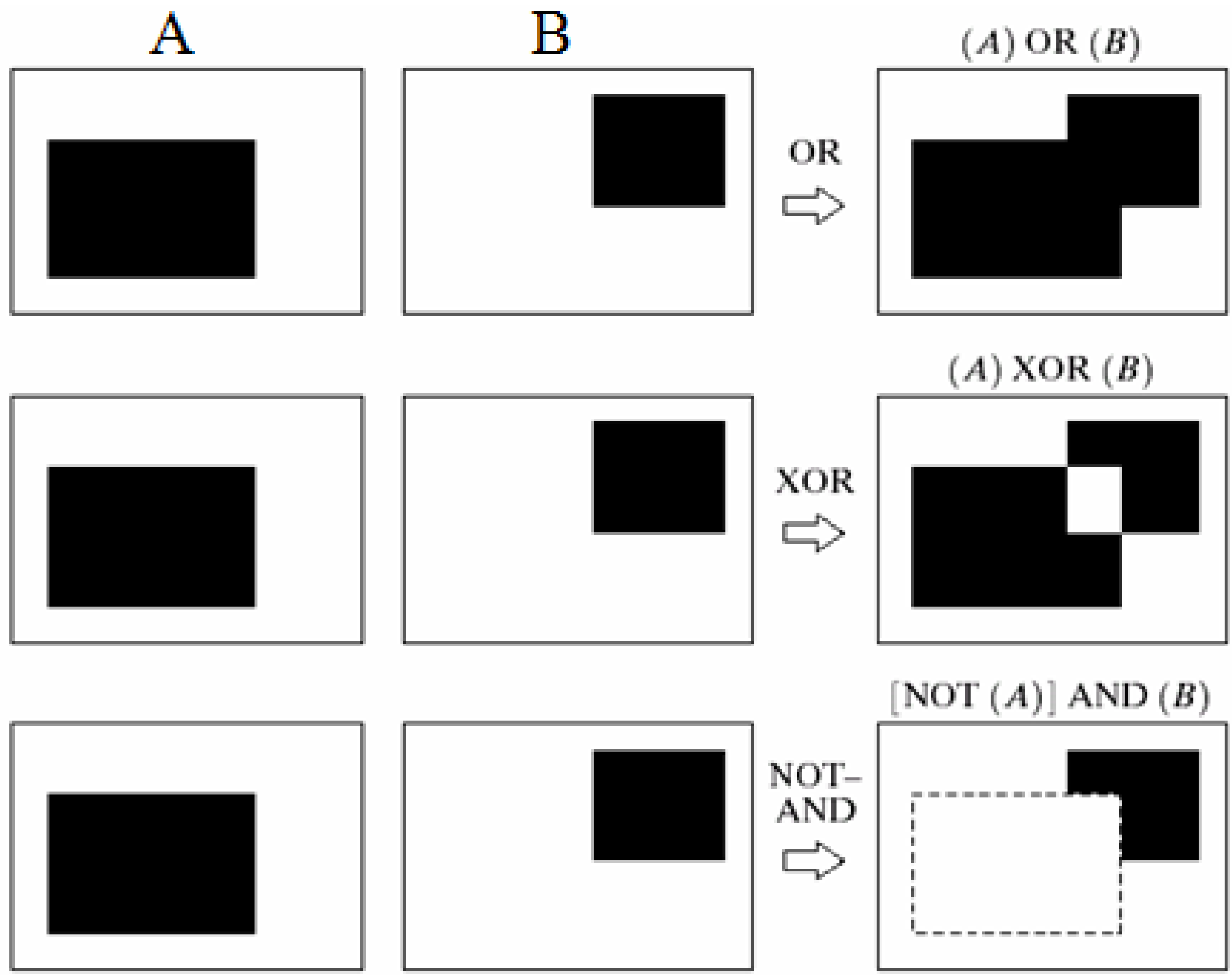


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Some Basic Concepts from Set Theory (Cont.)



Types Of Morphological Operations / Fundamental Operations

- **Dilation:**
  - Enlarge a region
  - Dilation adds pixels on the object boundaries.
- **Erosion:**
  - Shrink a region
  - Erosion removes pixels on object boundaries.
- **Open:**
  - The opening operation erodes an image and then dilates the eroded image, using the same structuring element for both operations.
- **Close:**
  - The closing operation dilates an image and then erodes the dilated image, using the same structuring element for both operations.

Types Of Morphological Operations / Fundamental Operations (Cont.)

$$dilation(W(x,y)) = \max(W(x,y))$$

$$erosion(W(x,y)) = \min(W(x,y))$$

$$opening = dilation \circ erosion$$

$$clo\sin g = erosion \circ dilation$$

Dilation

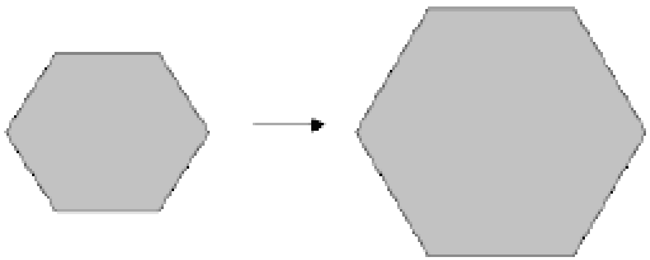
- Enlarge a region
- Dilation adds pixels on the object boundaries
- Dilation expands the image pixels, or it adds pixels on object boundaries.
- First, we traverse the structuring element over the image object to perform an dilation operation, as shown in Figure.
- The output pixel values are calculated using the following equation.
  - Pixel (output) = 1 {if HIT}
  - Pixel (output) = 0 {otherwise}
- **Dilation is A XOR B.**

Dilation (Cont.)

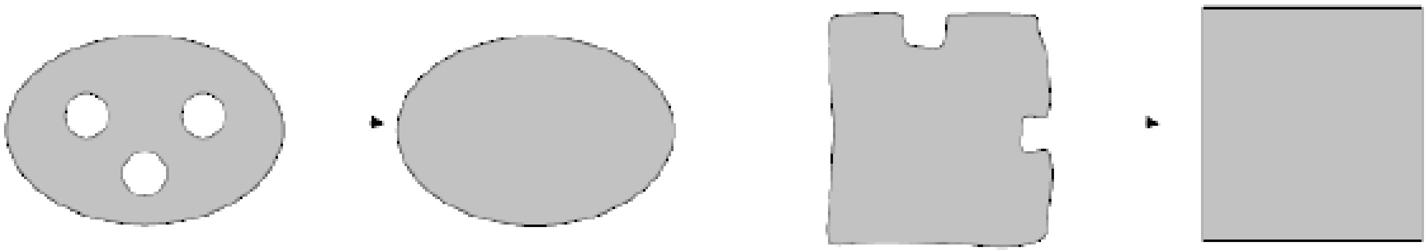
Dilation expands the connected sets of 1s of a binary image.

It can be used for

1. expanding shapes:

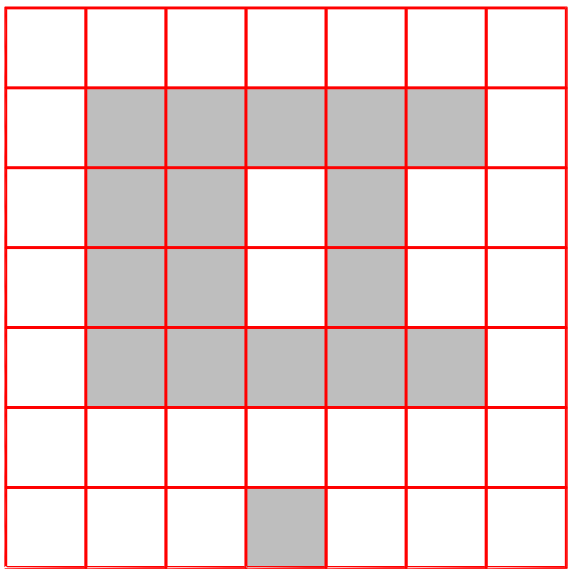


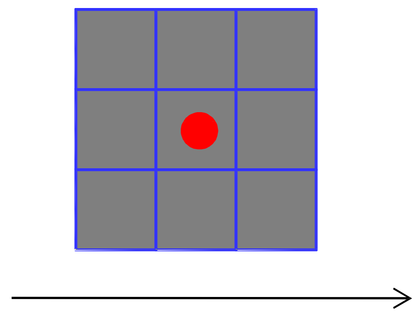
2. filling holes, gaps and gulfs:

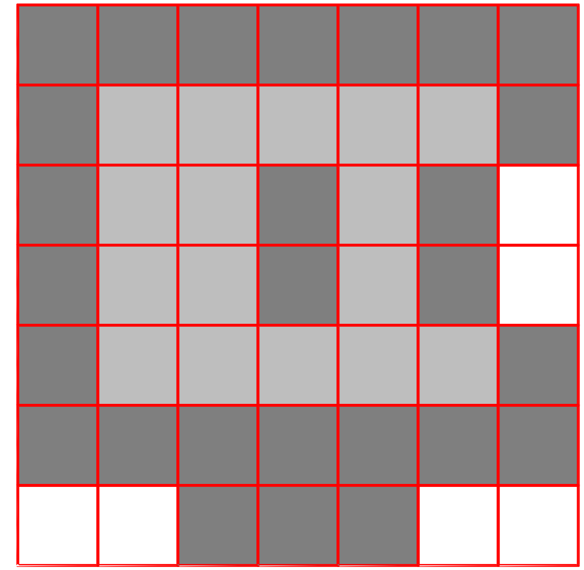


Dilation (Cont.)

Application







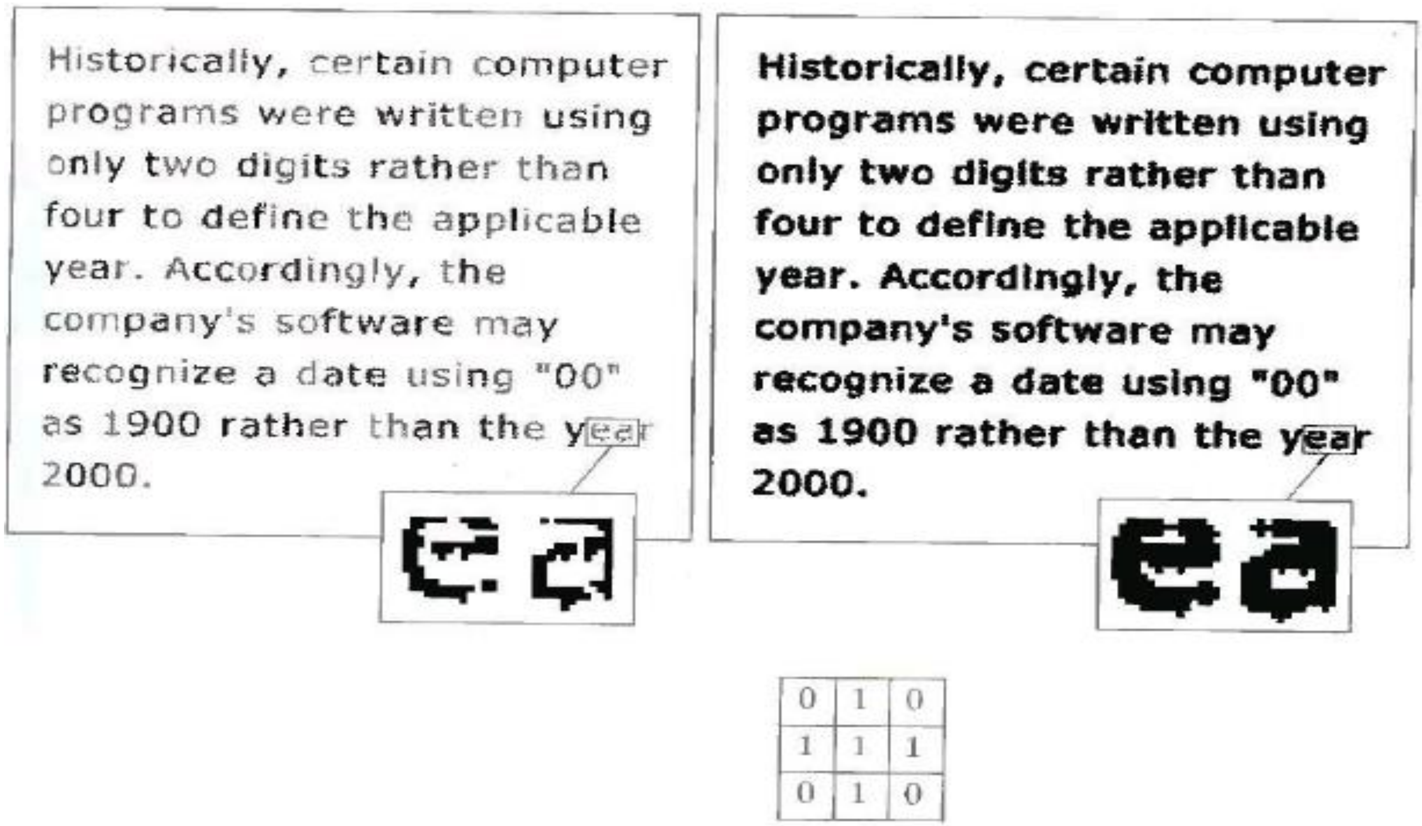
Expand object

Filling holes

Connecting disjoint components

Dilation (Cont.)

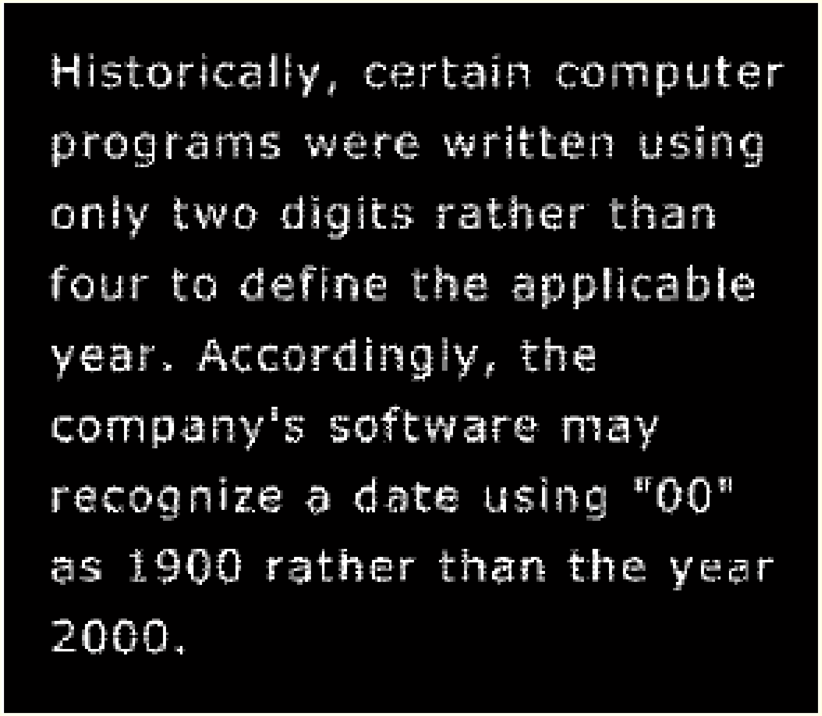
Marwadi University

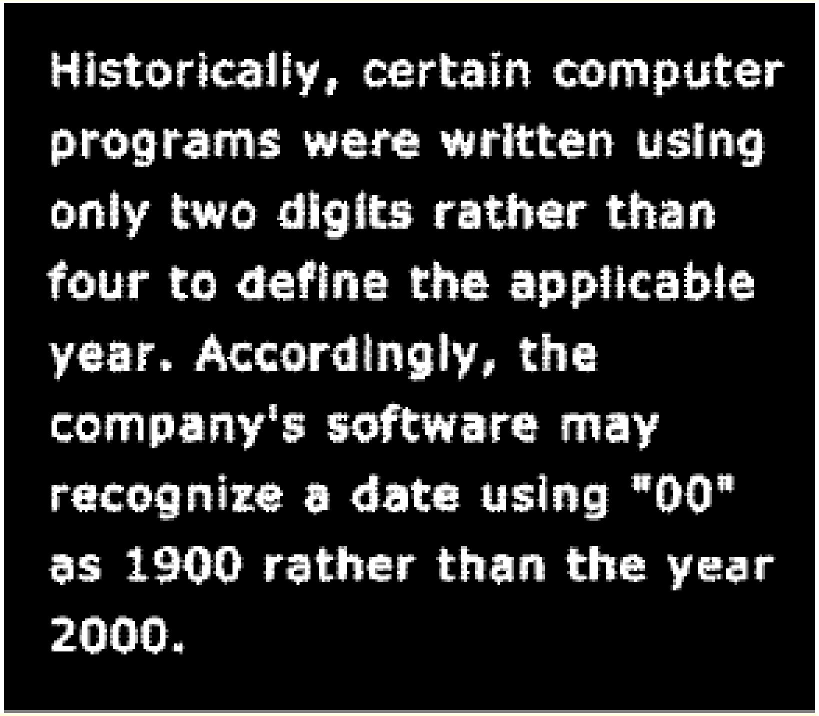


a b c

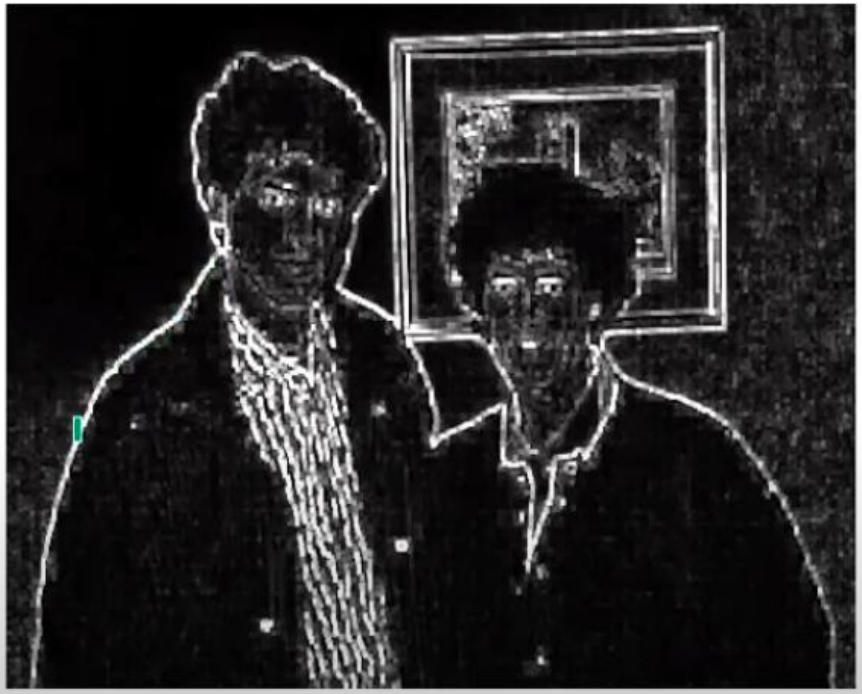
FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

Dilation (Cont.)





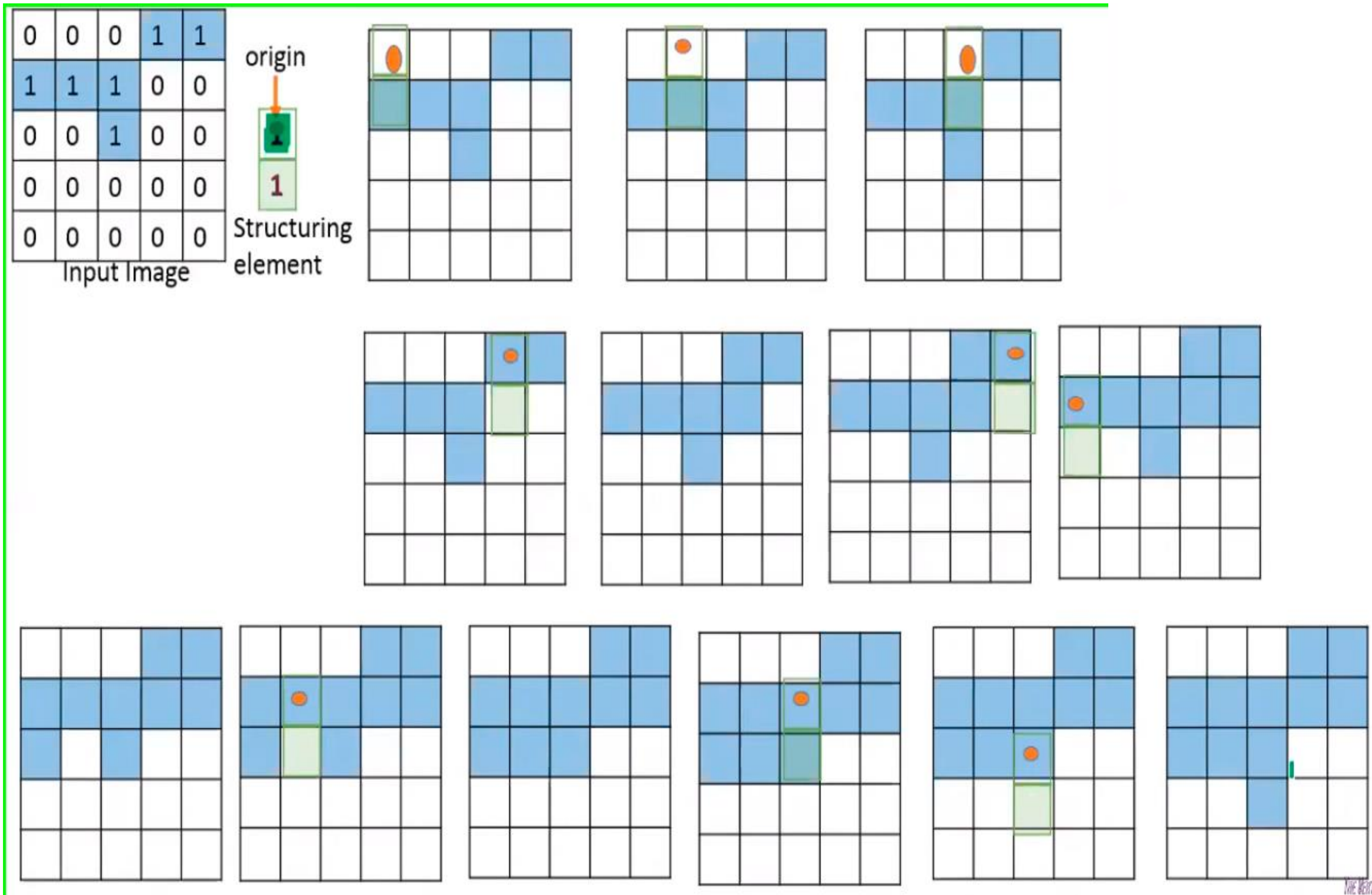
Dilation (Cont.)



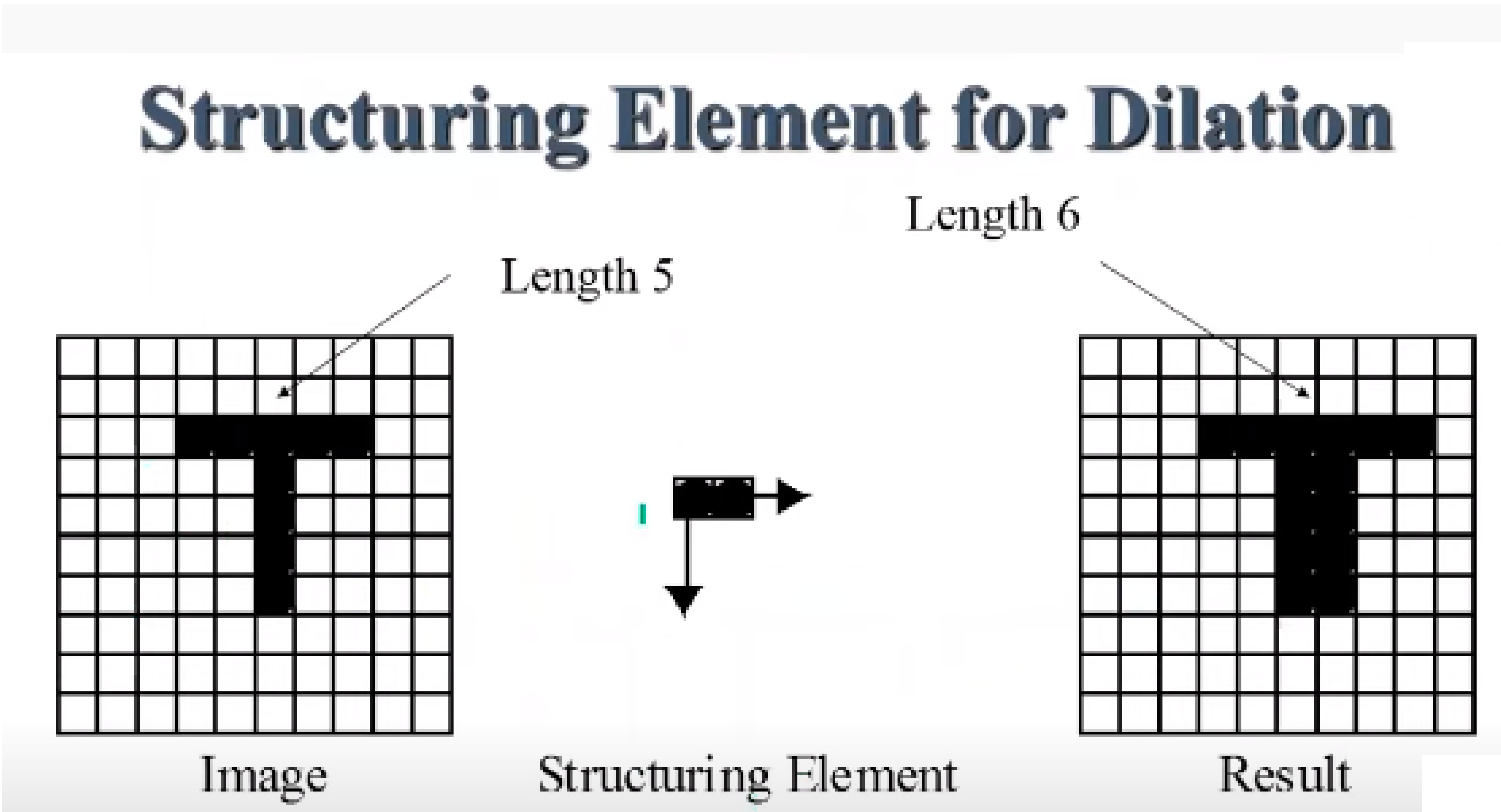
Dilation (Cont.)

- Pixel (output) = 1 {if HIT}
- Pixel (output) = 0 {otherwise}

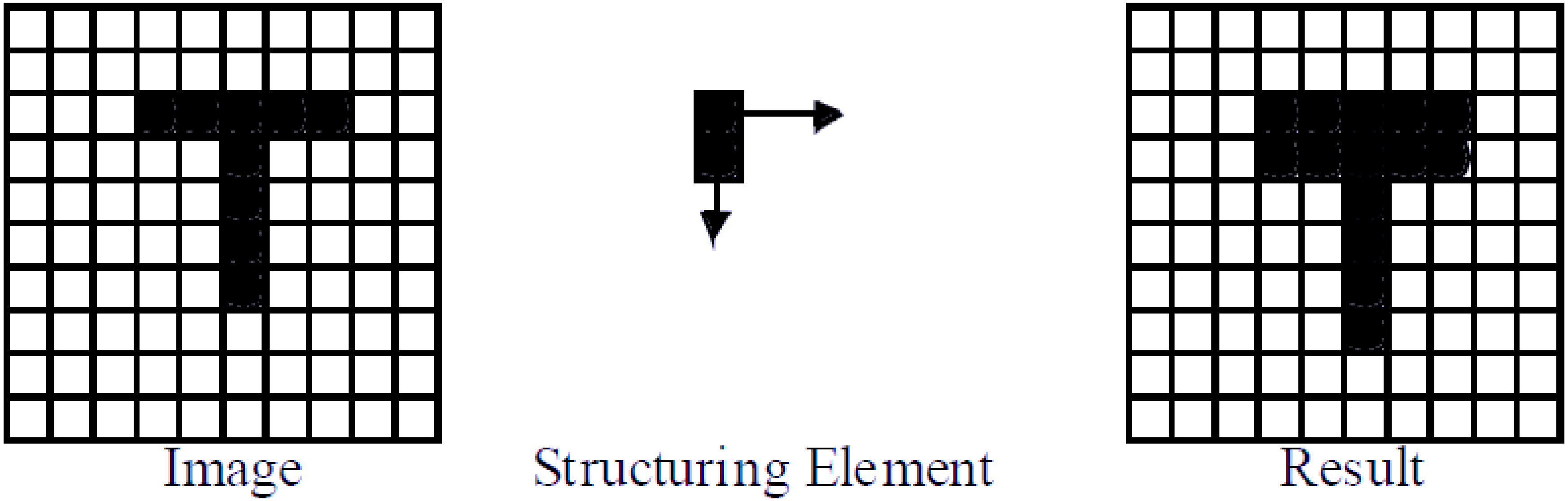
Dilation (Cont.)



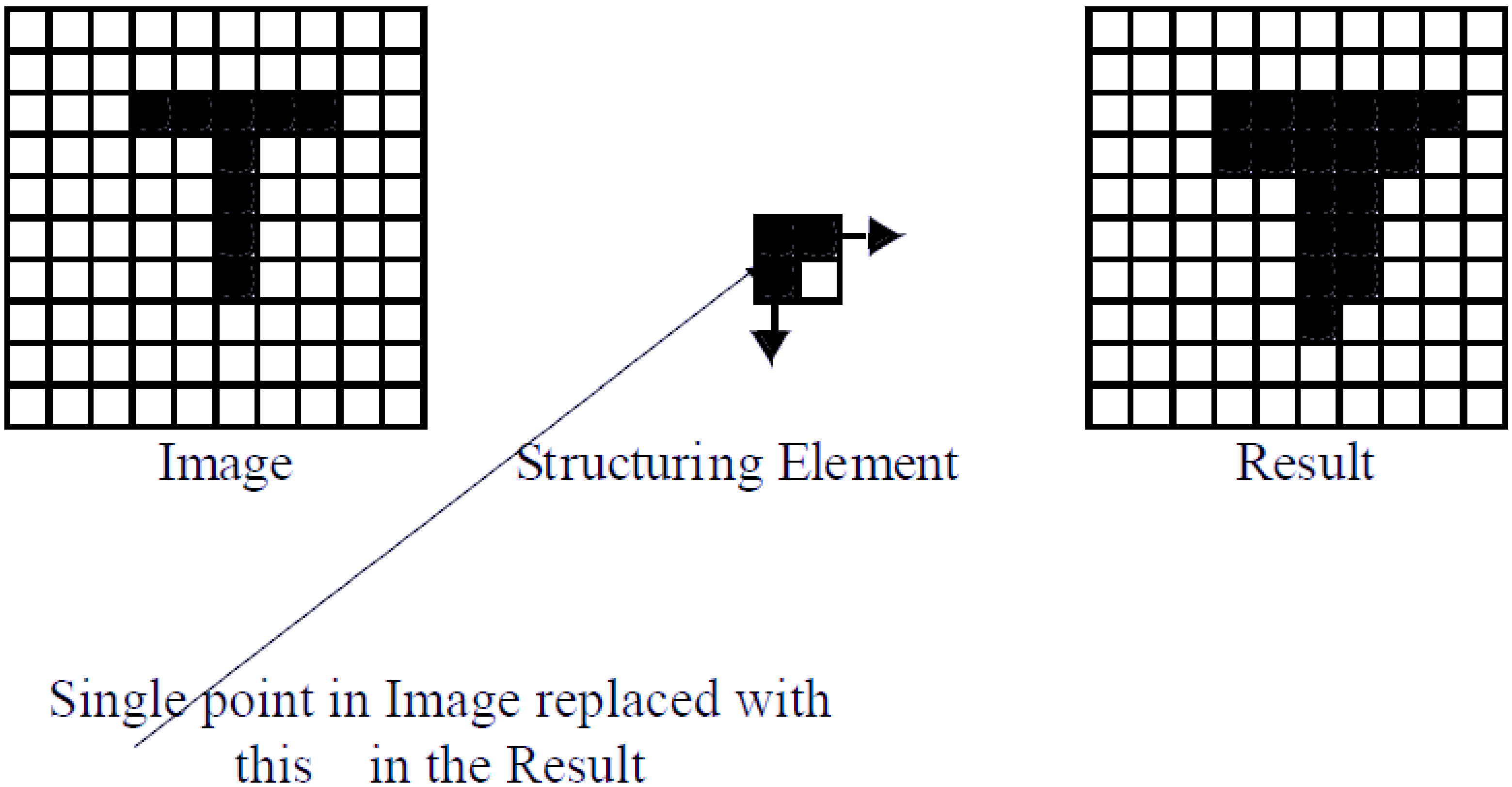
Dilation (Cont.)



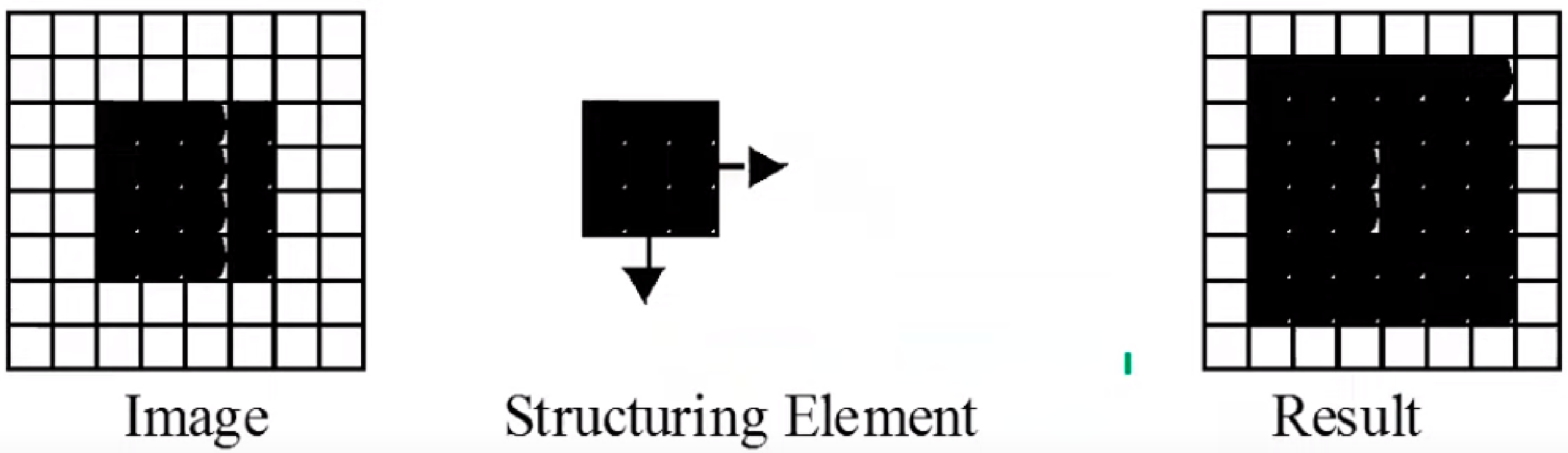
Dilation (Cont.)



Dilation (Cont.)

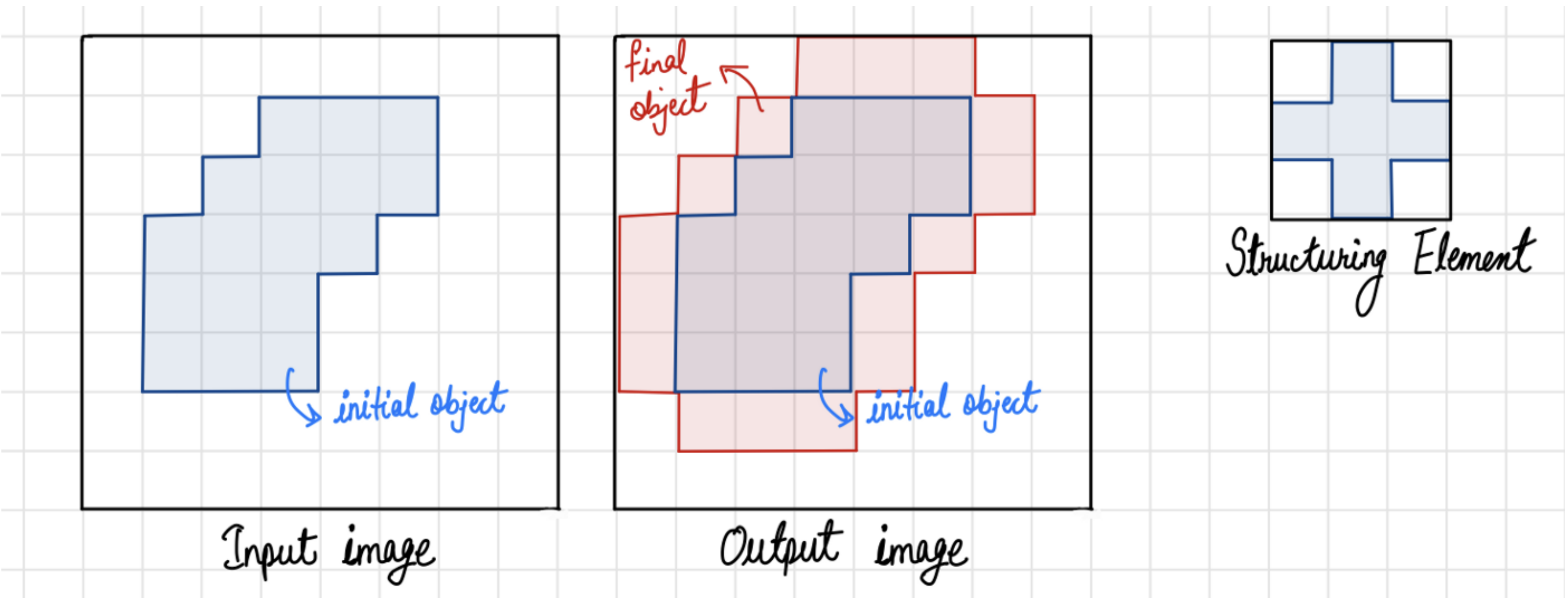


Dilation (Cont.)



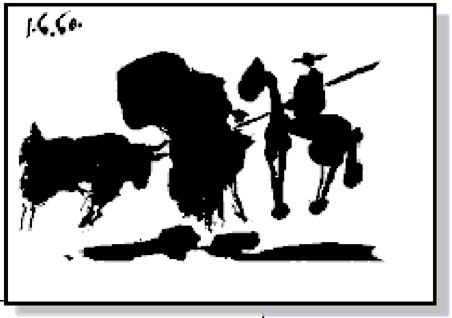
Dilation (Cont.)

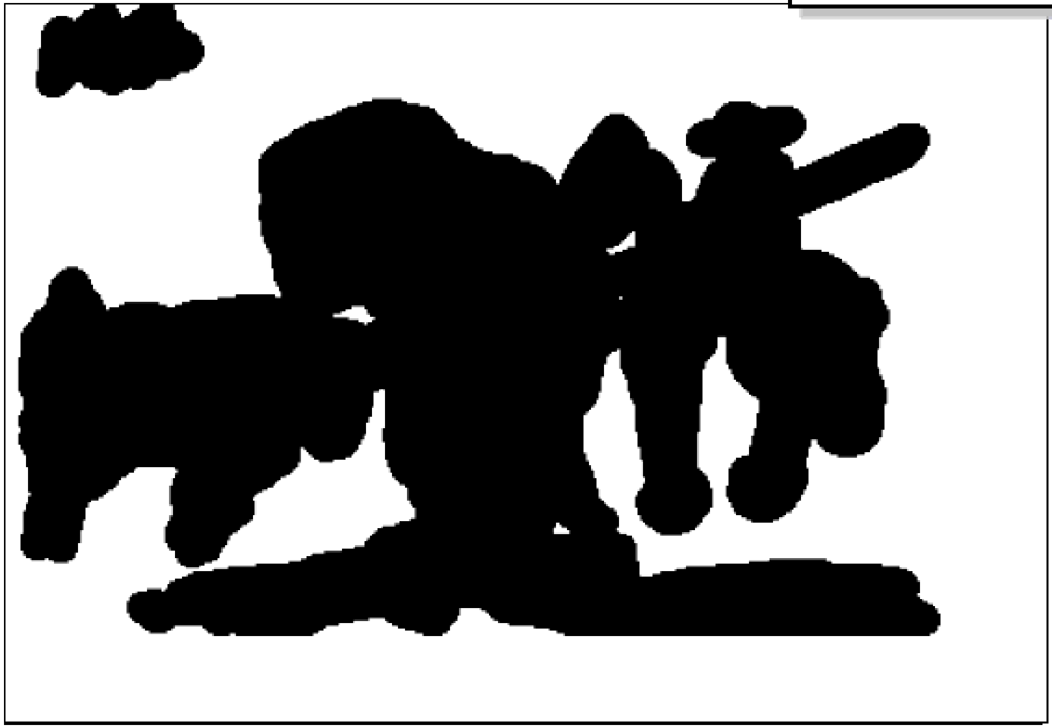
- Pixel (output) = 1 {if HIT}
- Pixel (output) = 0 {otherwise}

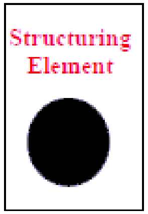


Dilation (Cont.)

Example of Dilation with various sizes of structuring elements

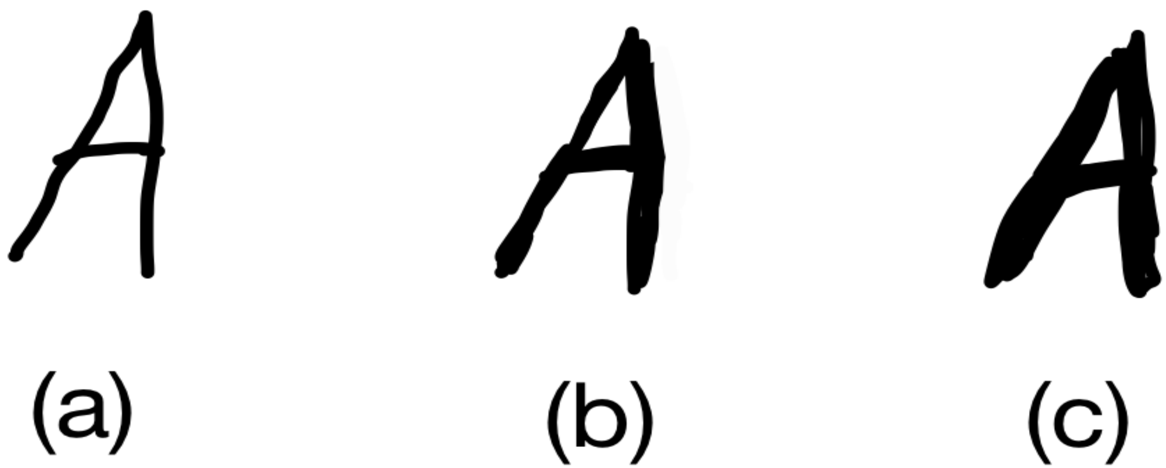






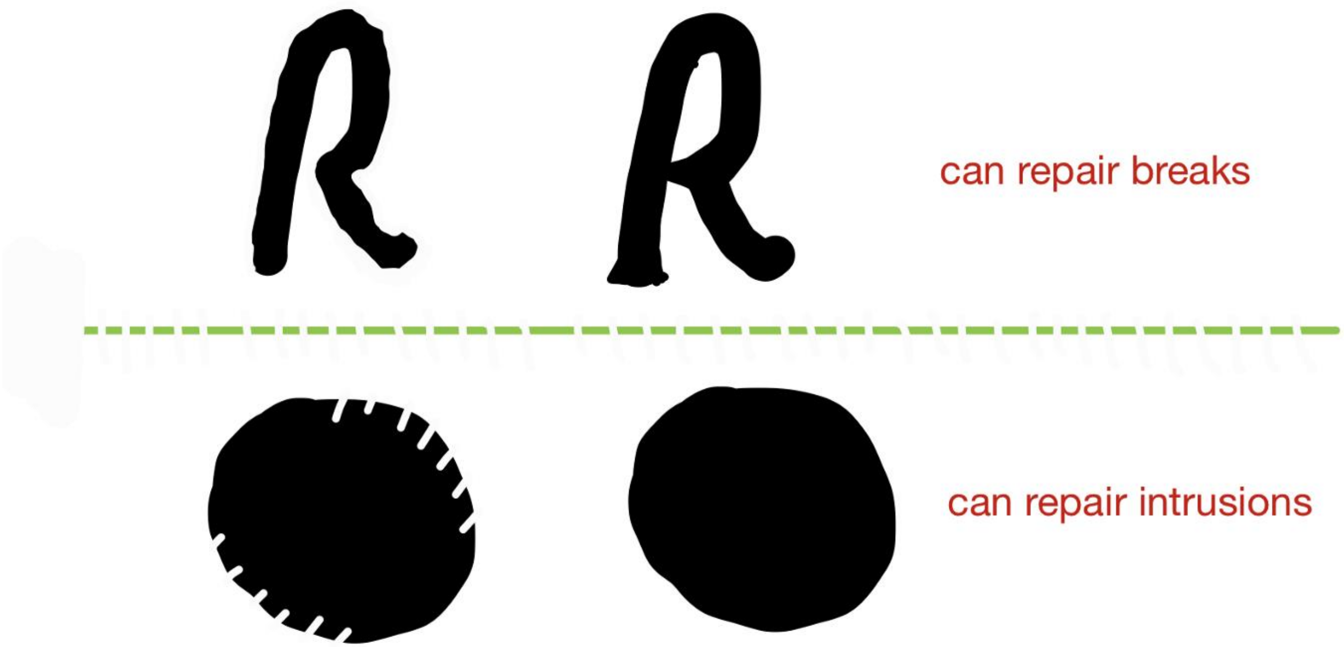
Dilation (Cont.)

- An example of Dilation is shown in Figure . Figure (a) represents original image, (b) and (c) shows processed images after dilation using 3x3 and 5x5 structuring elements respectively.



Dilation (Cont.)

- Properties:
  - It can repair breaks
  - It can repair intrusions

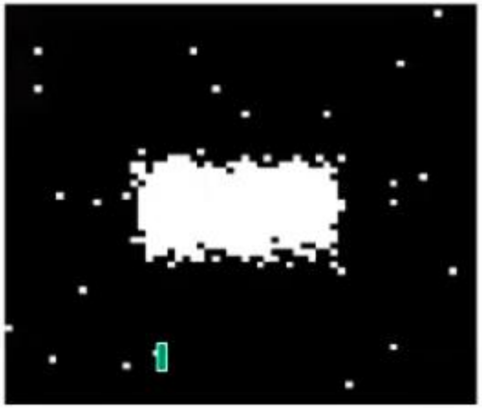


Erosion

- Erosion shrinks the image pixels, or erosion removes pixels on object boundaries.
- First, we traverse the structuring element over the image object to perform an erosion operation, as shown in Figure.
- The output pixel values are calculated using the following equation.
  - Pixel (output) = 1 {if FIT}
  - Pixel (output) = 0 {otherwise}

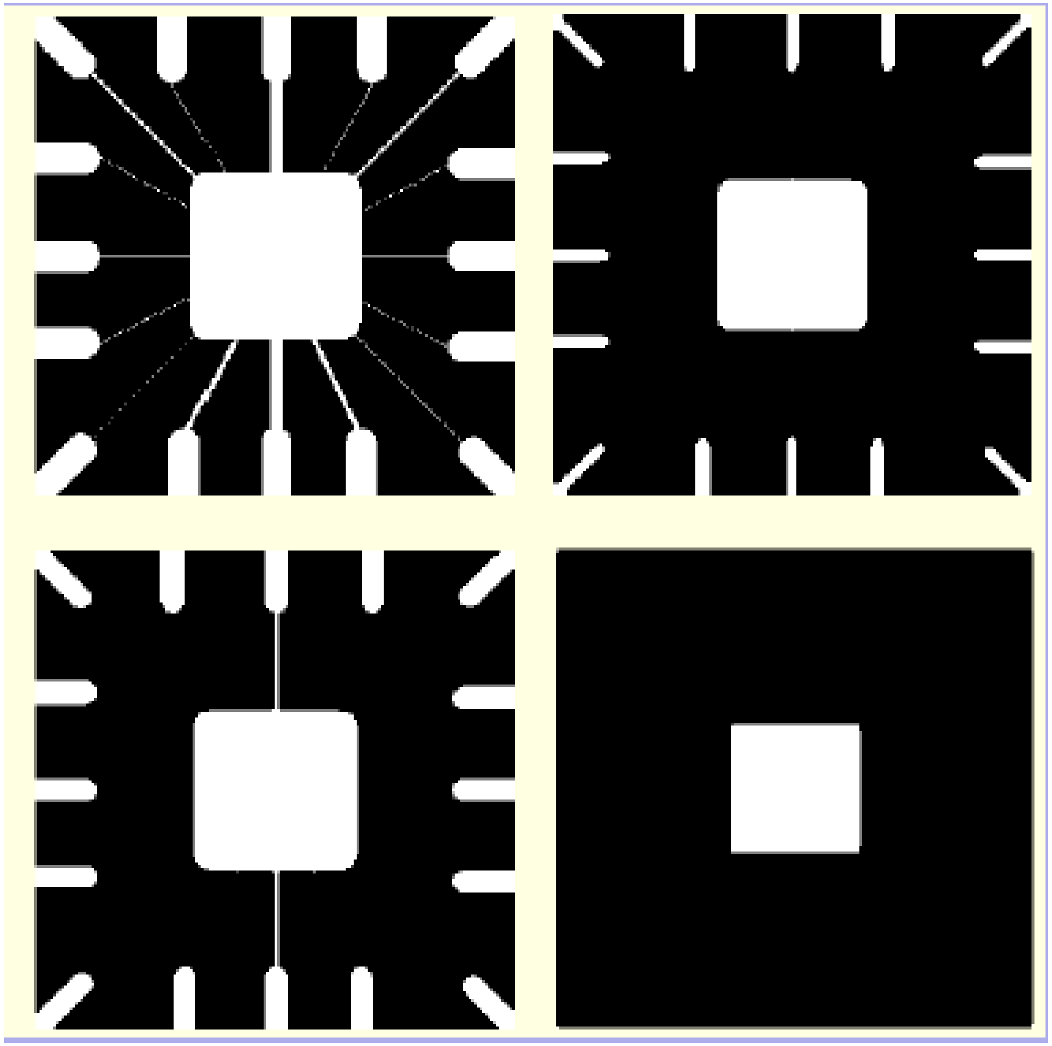
Erosion (Cont.)

1. Removes isolated noisy pixels.
2. Smoothes object boundary.
3. Removes the outer layer of object pixels, ie, object becomes slightly smaller.

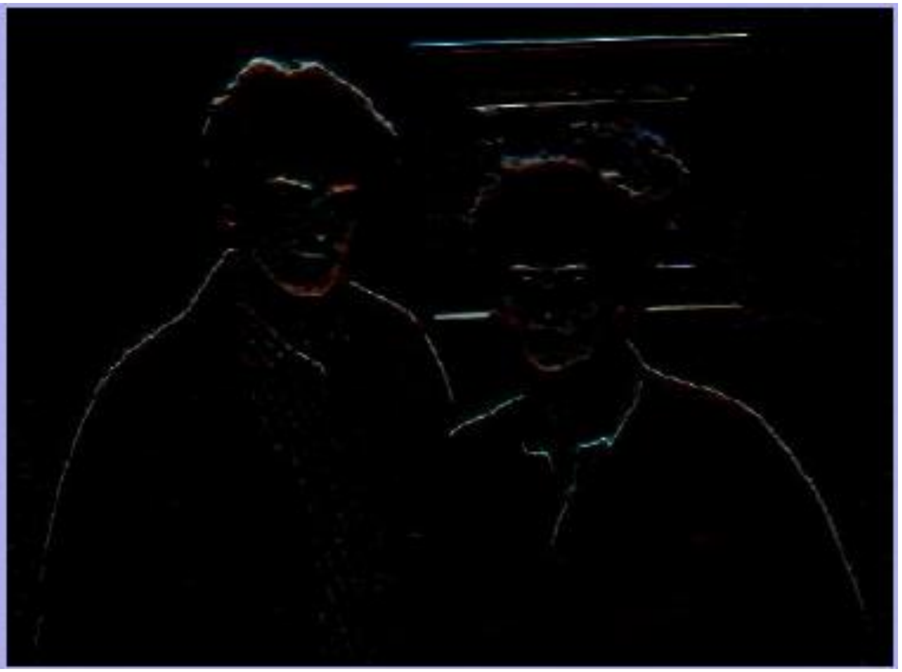


Erosion (Cont.)

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.

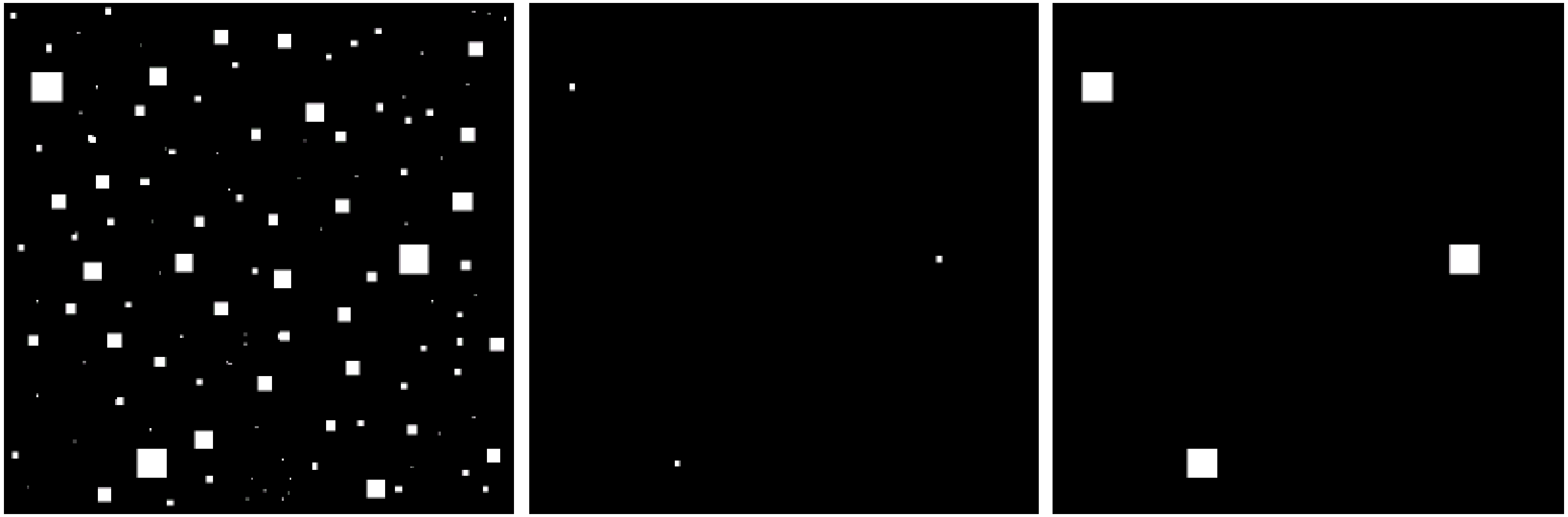


Erosion (Cont.)



Erosion (Cont.)

- Erosion can Eliminating Irrelevant Details



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.

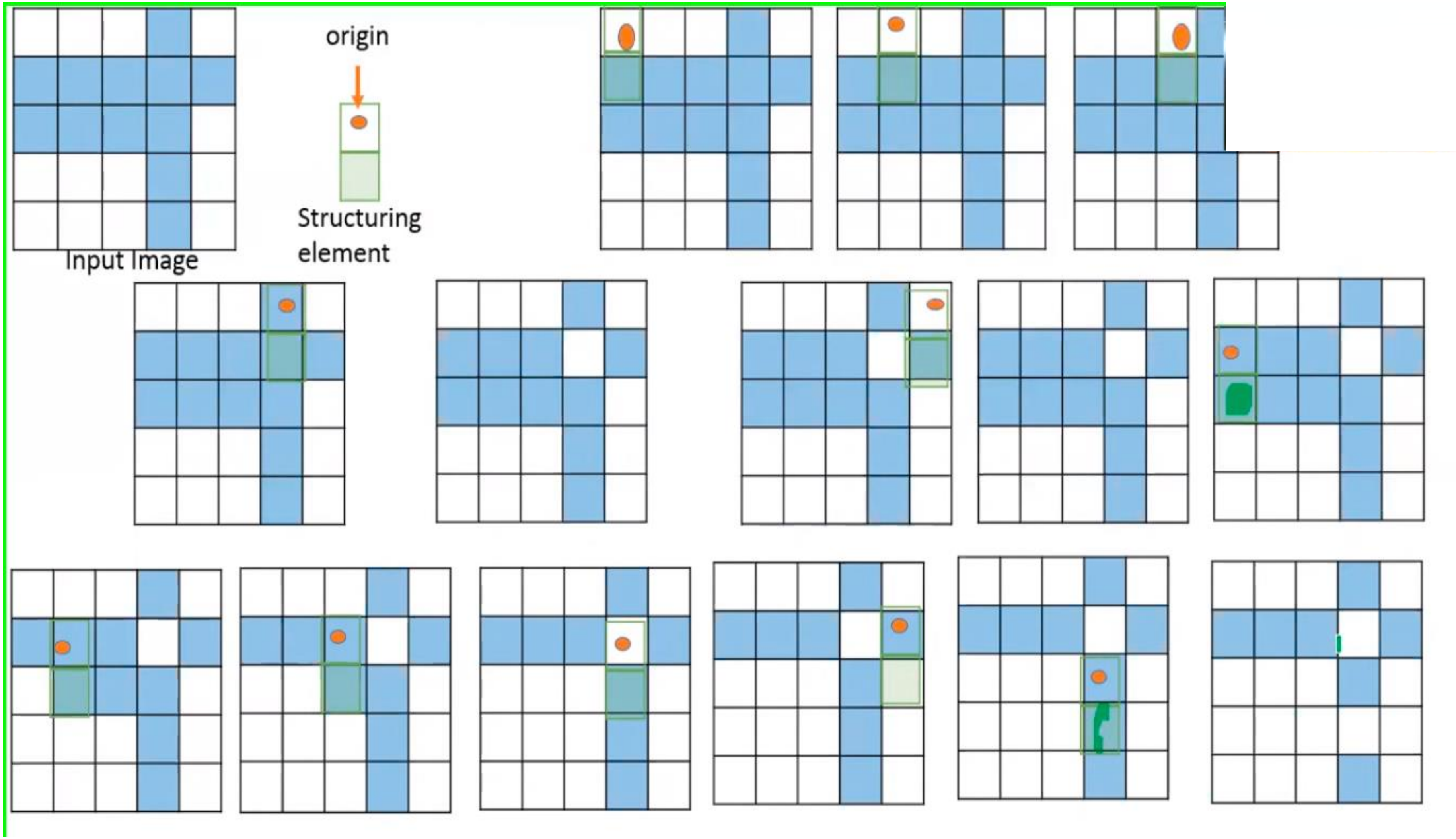
Erosion (Cont.)

- Pixel (output) = 1 {if FIT}
- Pixel (output) = 0 {otherwise}

Or

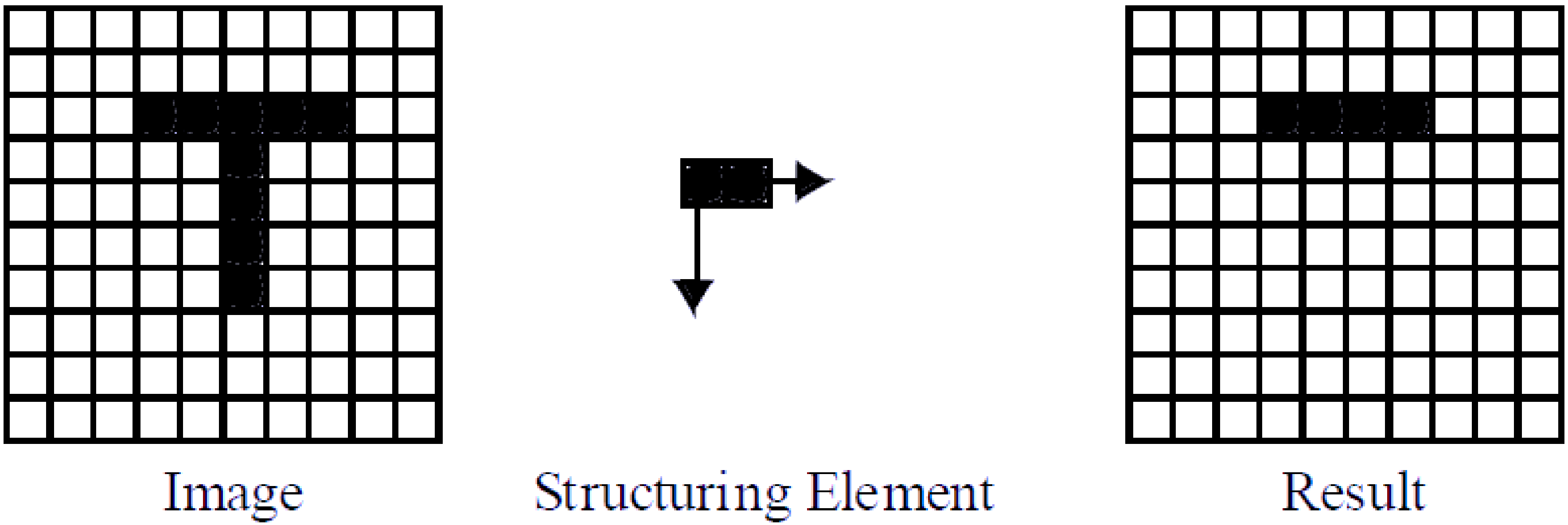
- Overlapping with Origin = remove pixel

Erosion (Cont.)



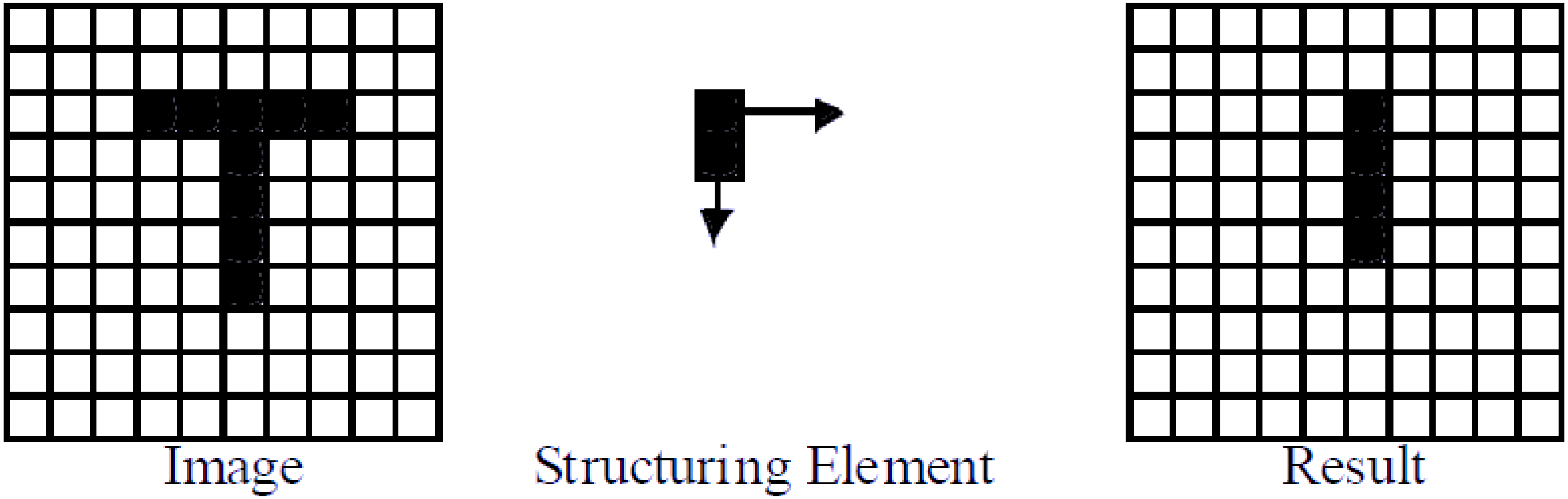
Erosion (Cont.)

- If not match – remove that pixel



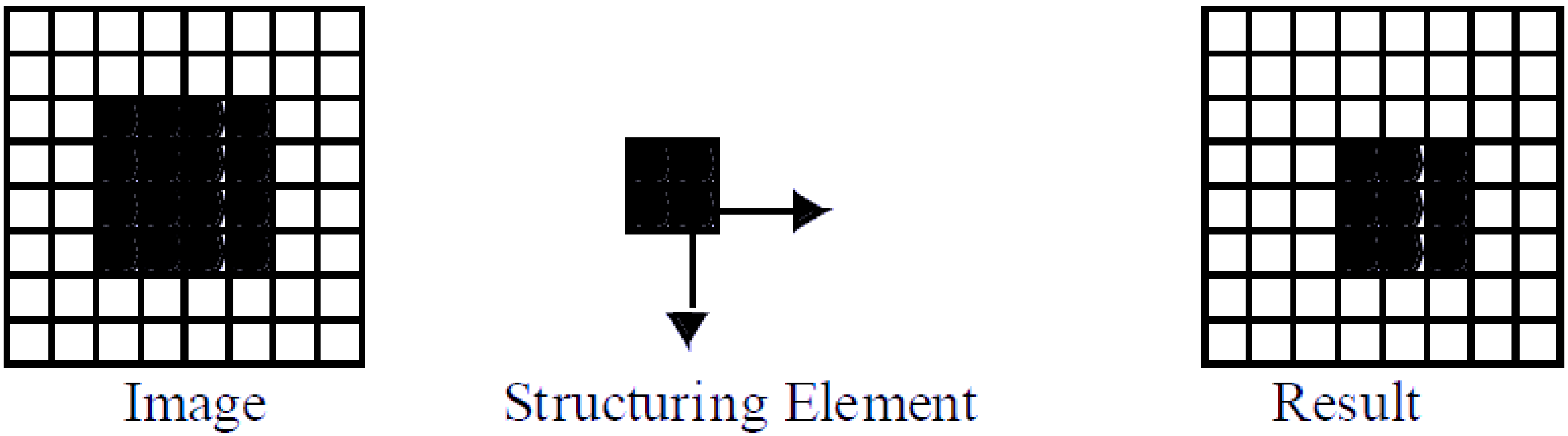
Erosion (Cont.)

- If not match – remove that pixel



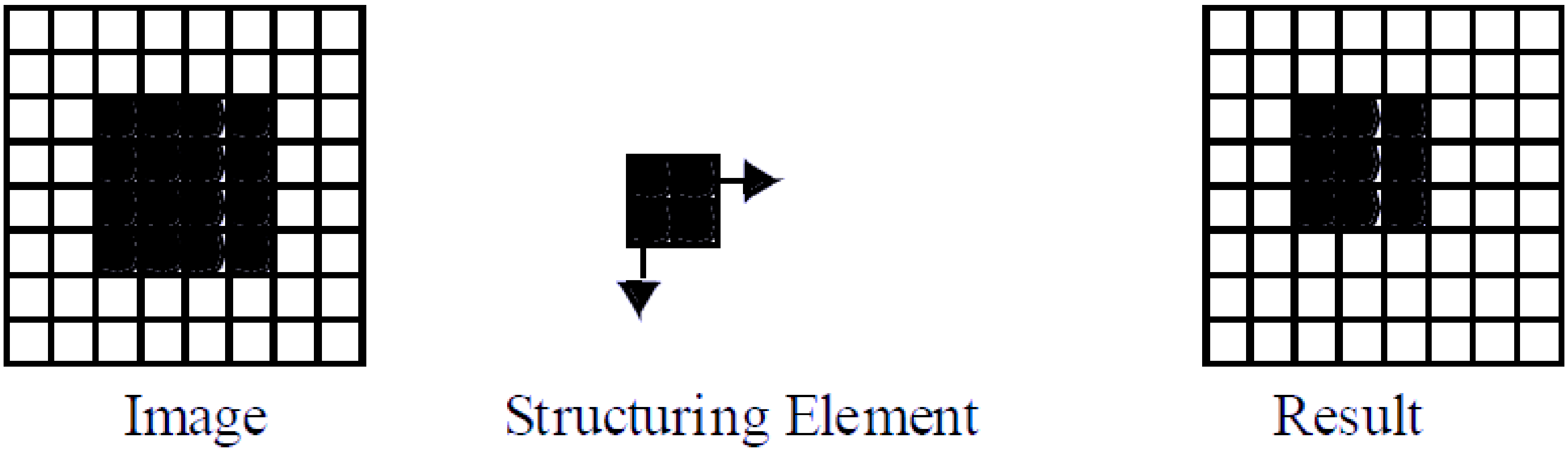
Erosion (Cont.)

- If not match – remove that pixel



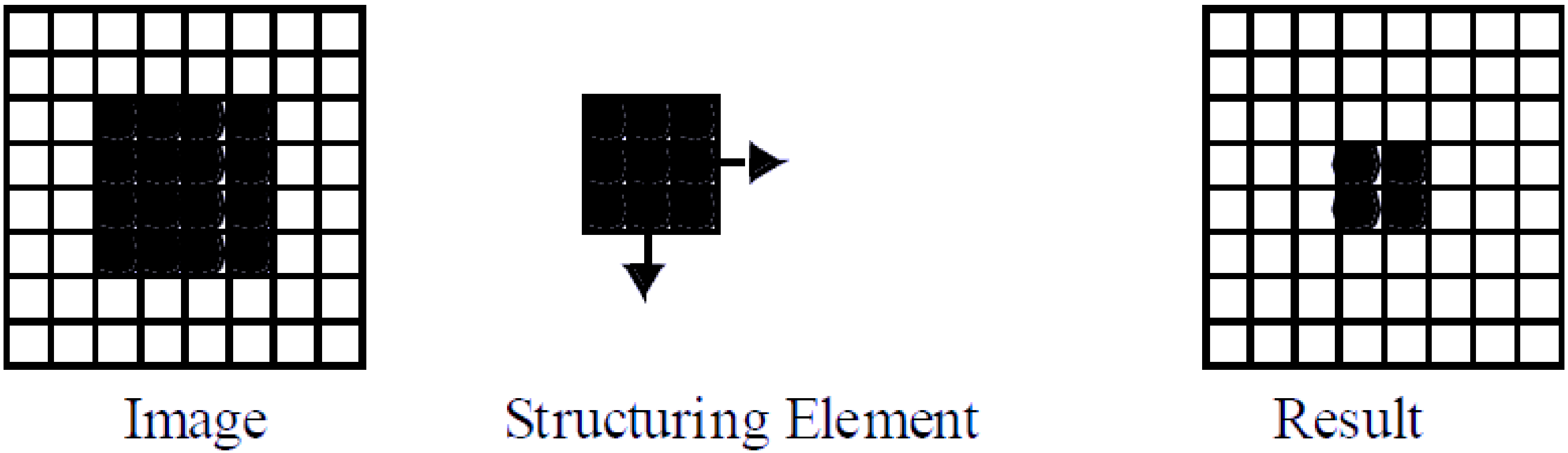
Erosion (Cont.)

- If not match – remove that pixel



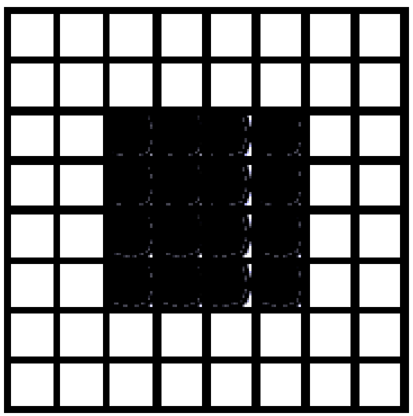
Erosion (Cont.)

- If not match – remove that pixel

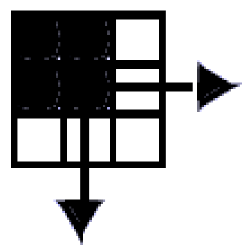


Erosion (Cont.)

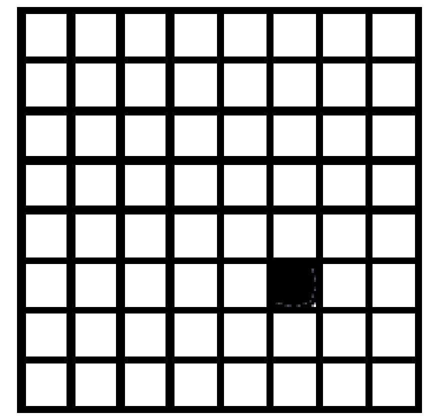
- If not match – remove that pixel



Image



Structuring Element

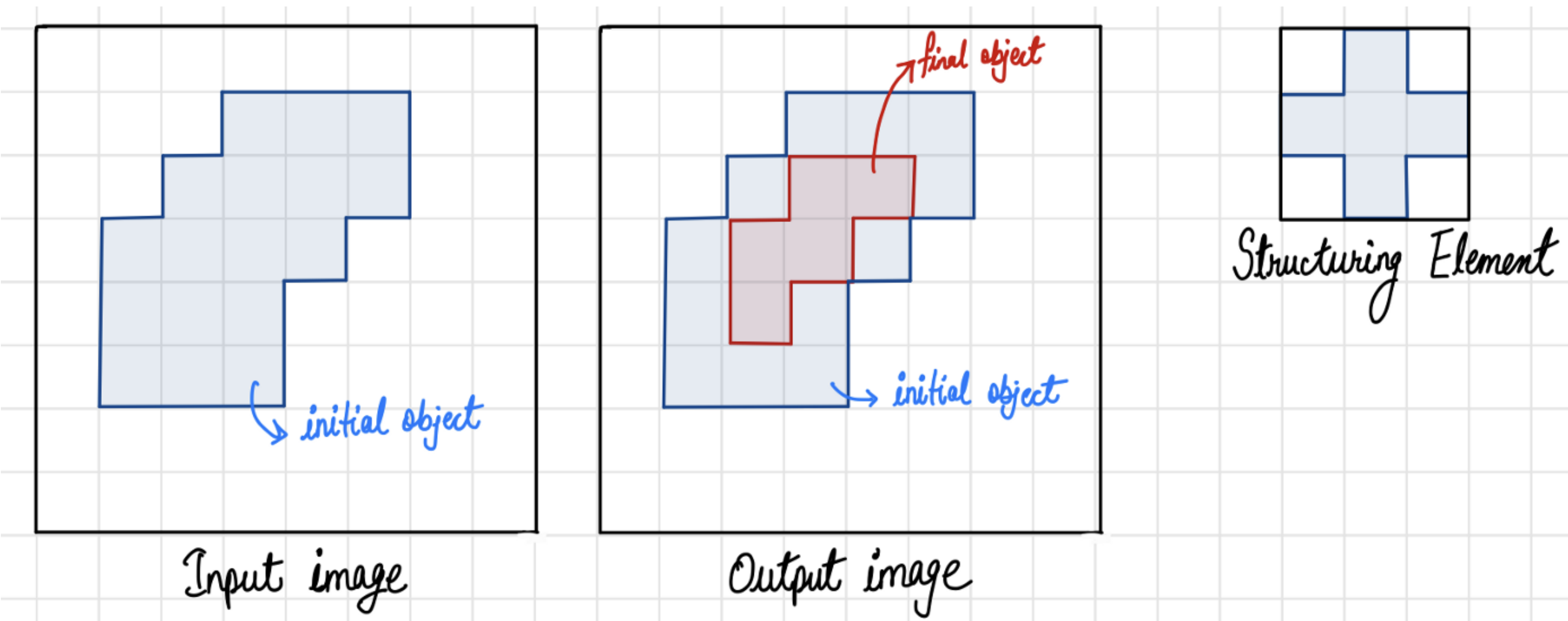


Result

Erosion (Cont.)

– Pixel (output) = 1 {if FIT}

– Pixel (output) = 0 {otherwise}



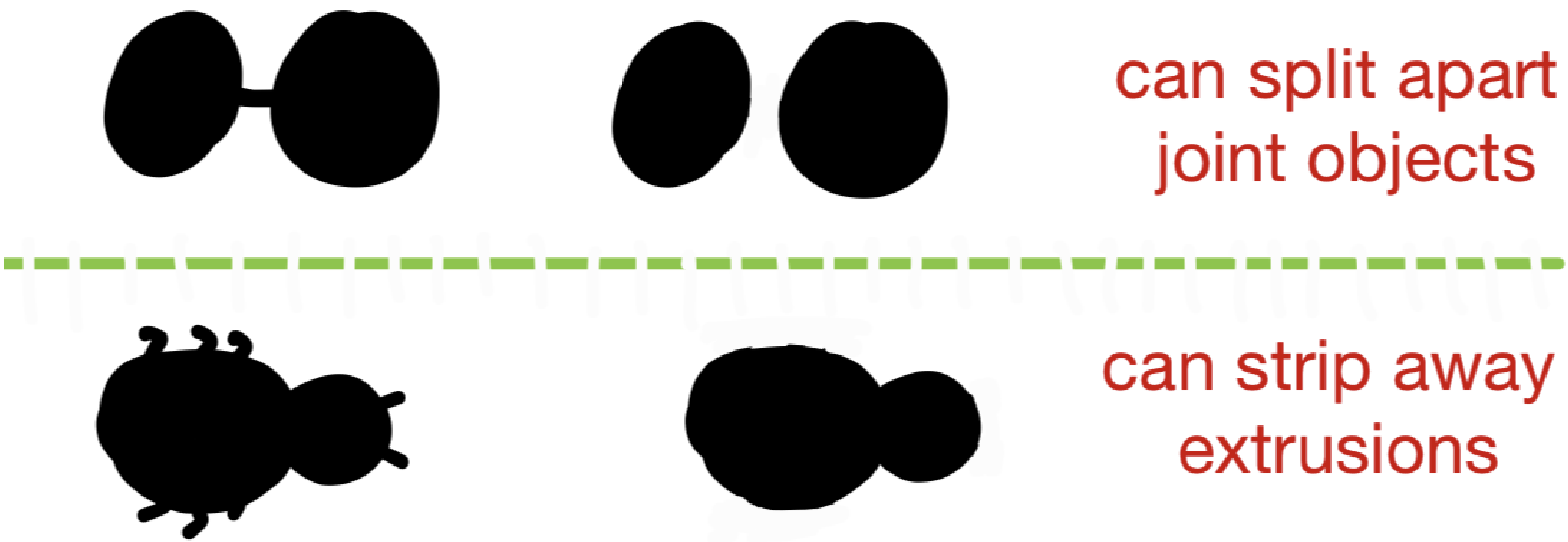
Erosion (Cont.)

- An example of Erosion is shown in Figure 5. Figure 5(a) represents original image, 5(b) and 5(c) shows processed images after erosion using 3x3 and 5x5 structuring elements respectively.

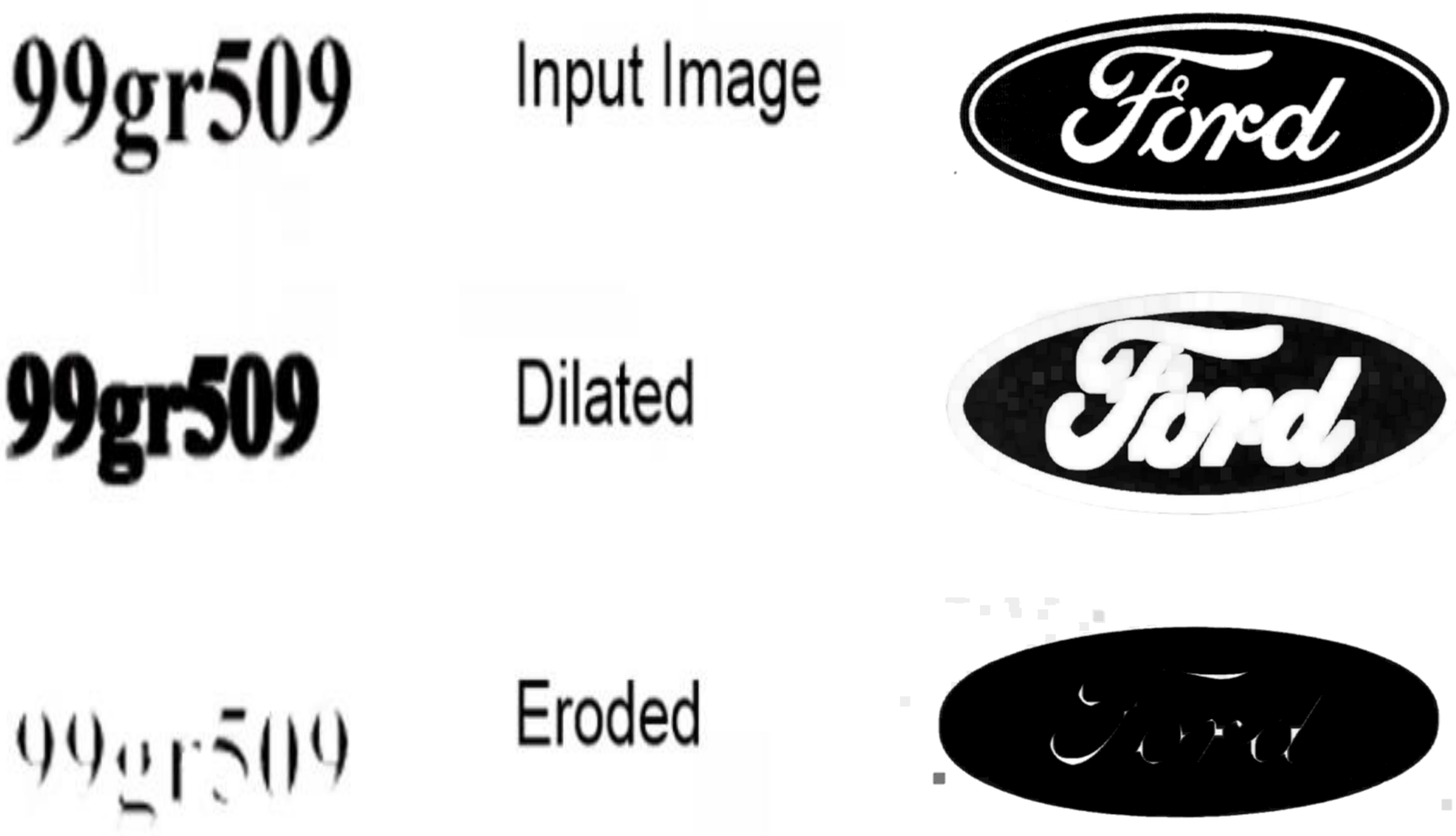
(a) (b) (c)

Erosion (Cont.)

- Properties:
  - It can split apart joint objects.
  - It can strip away extrusions.



Dilation & Erosion



Dilation VS Erosion

Dilation	Erosion
It increases the size of the objects.	It decreases the size of the objects.
It fills the holes and broken areas.	It removes the small anomalies.
It connects the areas that are separated by space smaller than structuring element.	It reduces the brightness of the bright objects.
It increases the brightness of the objects.	It removes the objects smaller than the structuring element.
Distributive, duality, translation and decomposition properties are followed.	It also follows the different properties like duality etc.
It is XOR of A and B.	It is dual of dilation.
It is used prior in Closing operation.	It is used later in Closing operation.
It is used later in Opening operation.	It is used prior in Opening operation.

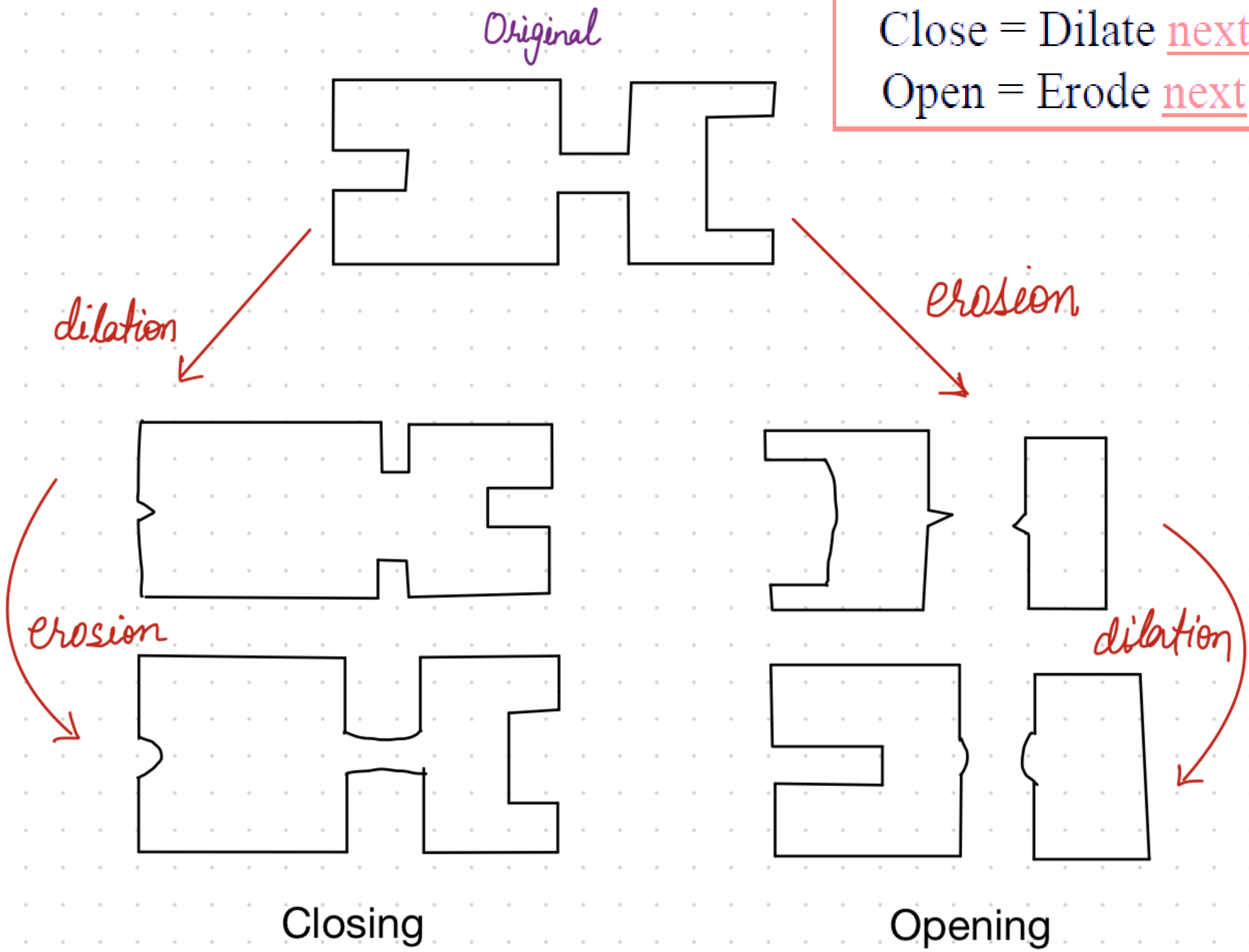
Compound Operations / Closing & Opening

- Most morphological operations are not performed using either dilation or erosion; instead, they are performed by using both.
- Two most widely used compound operations are:
  - Closing - by first performing dilation and then erosion)
  - Opening - by first performing erosion and then dilation)

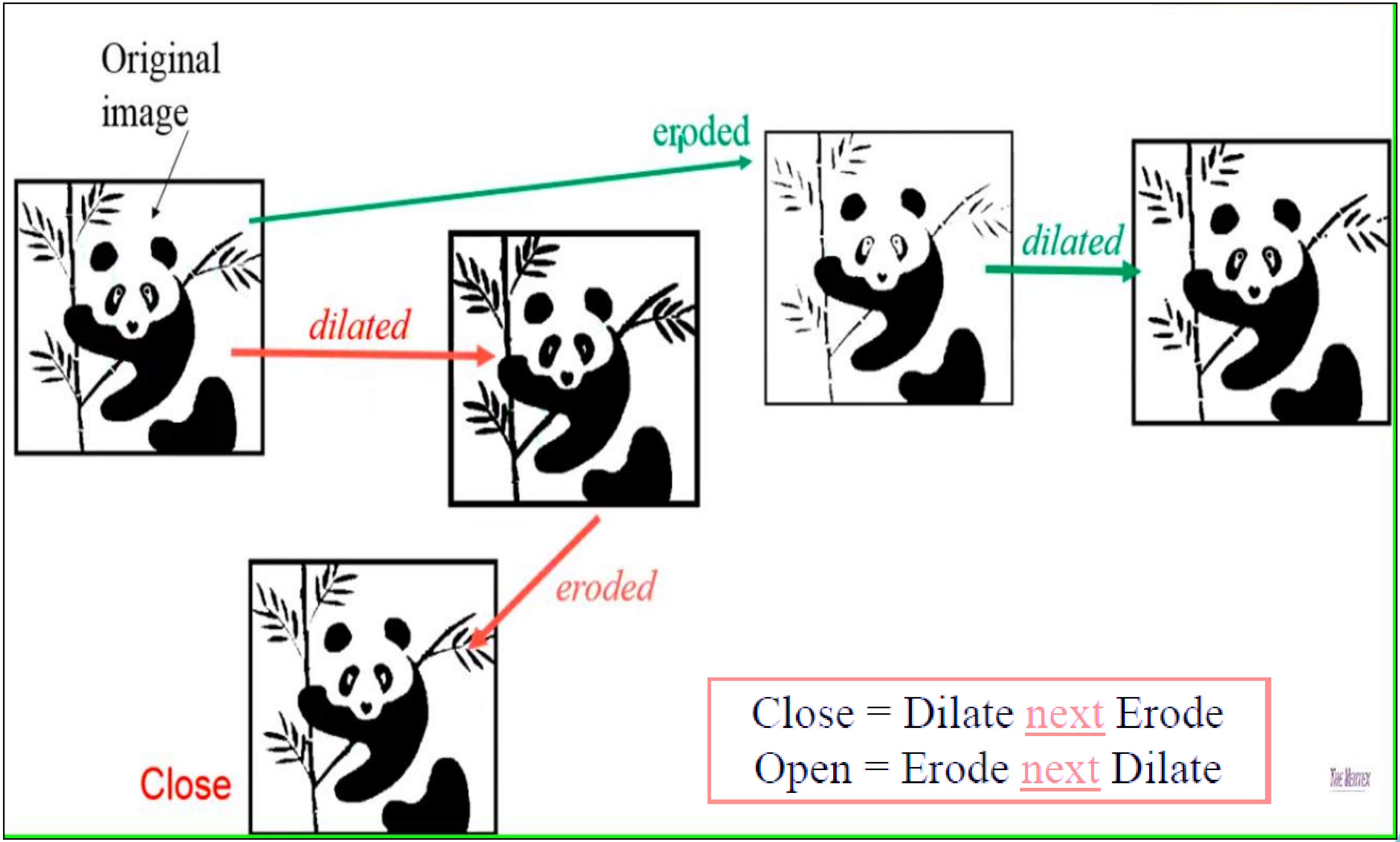
$$\boxed{\begin{array}{c}\text{Close} = \text{Dilate } \underline{\text{next}} \text{ Erode} \\ \text{Open} = \text{Erode } \underline{\text{next}} \text{ Dilate}\end{array}}$$

Closing & Opening (Cont.)

Close = Dilate next Erode
Open = Erode next Dilate



Closing & Opening (Cont.)



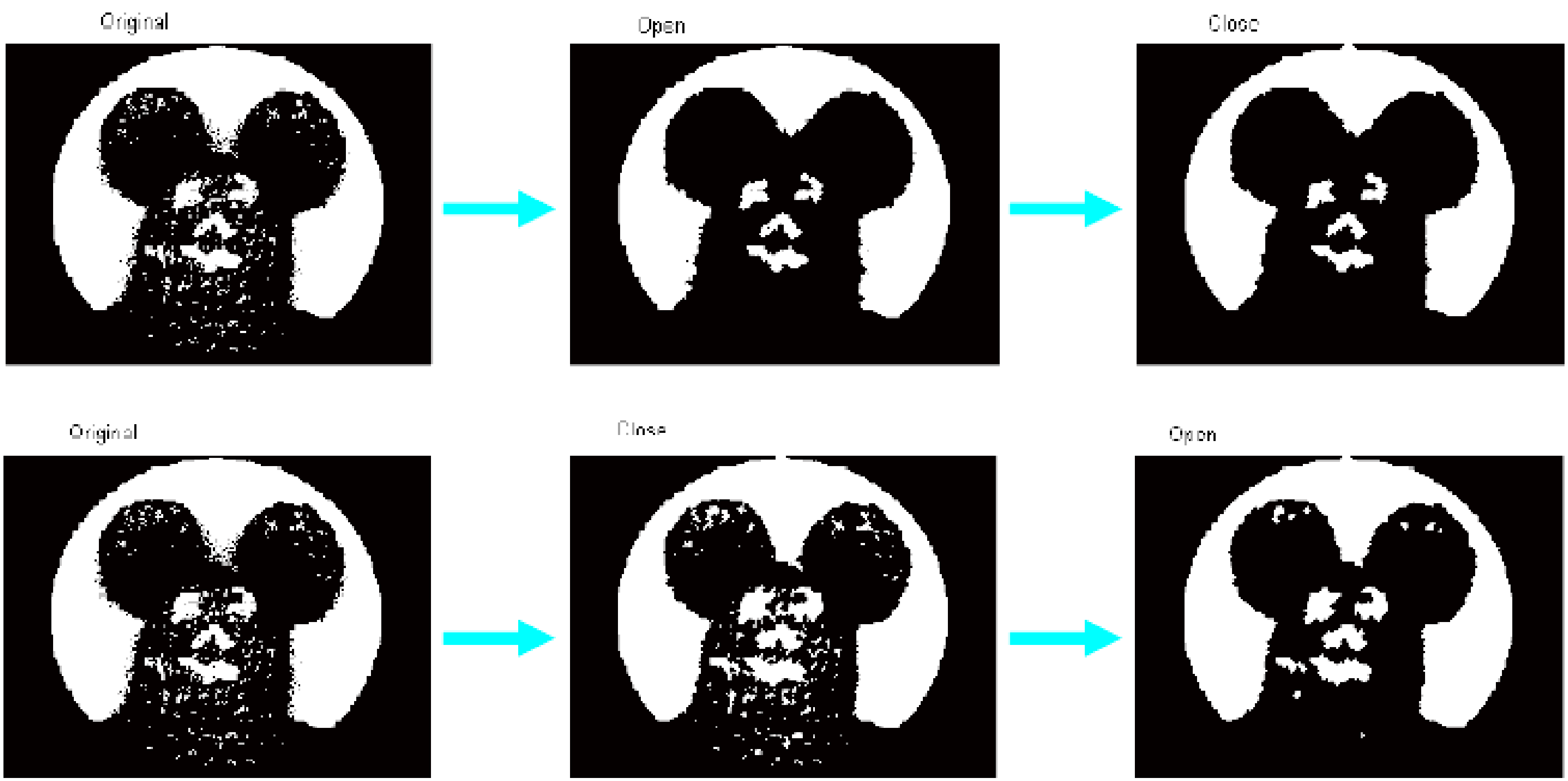
Closing & Opening (Cont.)

Original Image

Image after Closing

Image after opening

Closing & Opening (Cont.)



Closing & Opening (Cont.)

- Closing
  - Closing consists of a dilation followed by an erosion and connects objects that are close to each other. It can be used to fill in holes and small gaps.

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

  - Use
    - Smooth contour
    - Fuse narrow breaks and long thin gulfs
    - Eliminate small holes
    - Fill gaps in the contour

Closing & Opening (Cont.)

- Opening
  - Opening consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are to small to contain the structuring element.

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

  - Use
    - Eliminates protrusions
    - Breaks necks
    - Smoothes contour

Morphological Operations and Their Properties

Operation	Equation	Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Translation	$(A)_z = \{w \mid w = a + z, \quad \text{for } a \in A\}$	Translates the origin of A to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	"Contracts" the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)

Morphological Operations and Their Properties (Cont.)

Hit-or-miss transform	$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$ $= (A \ominus B_1) - (A \oplus \hat{B}_2)$	The set of points (coordinates) at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c.
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Region filling	$X_k = (X_{k-1} \oplus B) \cap A^c$; $X_0 = p$ and $k = 1, 2, 3, \ldots$	Fills a region in A, given a point p in the region. (II)
Connected components	$X_k = (X_{k-1} \oplus B) \cap A$; $X_0 = p$ and $k = 1, 2, 3, \ldots$	Finds a connected component Y in A, given a point p in Y. (I)
Convex hull	$X_k^i = (X_{k-1}^i \circledast B^i) \cup A$; $i = 1, 2, 3, 4$; $k = 1, 2, 3, \ldots$; $X_0^i = A$; and $D^i = X_{\text{conv}}^i$.	Finds the convex hull $C(A)$ of set A, where "conv" indicates convergence in the sense that $X_k^i = X_{k-1}^i$. (III)

Morphological Operations and Their Properties (Cont.)

Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	Thins set A. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)
Thickening	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} =$ $((\ldots(A \odot B^1) \odot B^2 \ldots) \odot B^n)$	Thickens set A. (See preceding comments on sequences of structuring elements.) Uses IV with 0's and 1's reversed.

Application of Morphology

- Dilate an Image to Enlarge a Shape
  - Dilation adds pixels to boundary of an object. Dilation makes objects more visible and fills in small holes in the object.
- Remove Thin Lines Using Erosion
  - Erosion removes pixels from the boundary of an object. Erosion removes islands and small objects so that only substantive objects remain.
- Use Morphological Opening to Extract Large Image Features
  - You can use morphological opening to remove small objects from an image while preserving the shape and size of larger objects in the image.
- Flood-Fill Operations
  - A flood fill operation assigns a uniform pixel value to connected pixels, stopping at object boundaries.
- Find Image Peaks and Valleys
  - You can use neighborhood processing to find global and regional minima and maxima in images.

Structuring Element in MATLAB

- Function to Create Structuring element

Strel()

Syntax: strel(shape, parameter)

```
SE = strel(nhood)
```

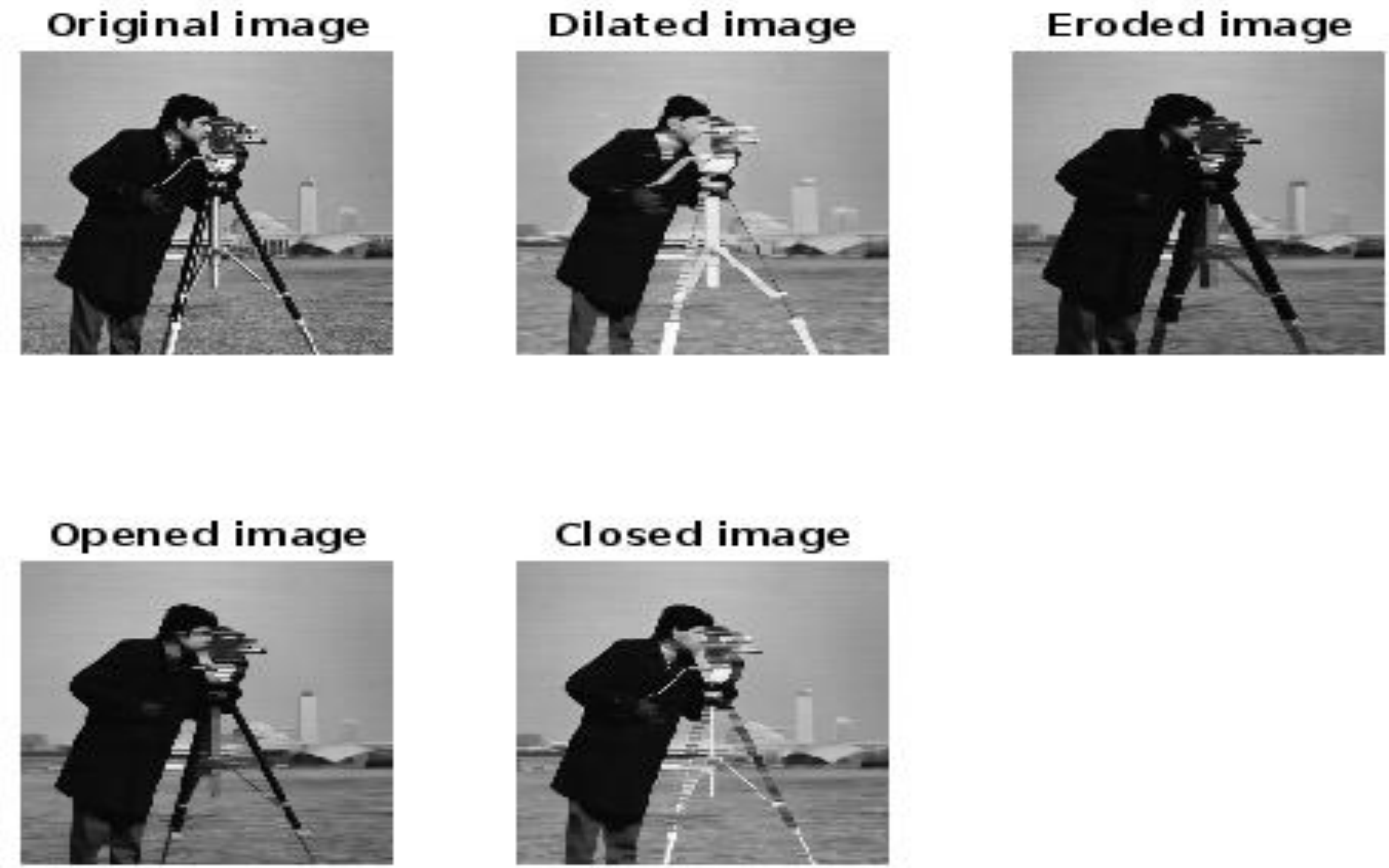
```
SE = strel("diamond",r)
SE = strel("disk",r)
SE = strel("disk",r,n)
SE = strel("octagon",r)
SE = strel("line",len,deg)
SE = strel("rectangle",[m n])
SE = strel("square",w)
```

```
SE = strel("cube",w)
SE = strel("cuboid",[m n p])
SE = strel("sphere",r)
```

Morphological Operations in MATLAB

```
# Importing the image
I = imread("cameraman.tif");
subplot(2, 3, 1),
imshow(I);
title("Original image");

% Dilated Image
se = strel("line", 7, 7);
dilate = imdilate(I, se);
subplot(2, 3, 2),
imshow(dilate);
title("Dilated image");

% Eroded image
erode = imerode(I, se);
subplot(2, 3, 3),
imshow(erode);
title("Eroded image");

% Opened image
open = imopen(I, se);
subplot(2, 3, 4),
imshow(open);
title("Opened image");

% Closed image
close = imclose(I, se);
subplot(2, 3, 5),
imshow(close);
title("Closed image");
```

Morphological Operations in MATLAB (Cont.)



Morphological Operations in MATLAB (Cont.)

Morphological Algorithms

- Boundary Extraction
- Hole Filling / Region Filling
- Thinning
- Thickening
- Skeletonization / Skeletons
- Convex Hull
- Extraction of Connected Components

Boundary Extraction

Image (A)

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

Structure Element

1	1	1
1	1	1
1	1	1

Erosion

$A \ominus B$

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

Boundary

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

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

Boundary Extraction (Cont.)

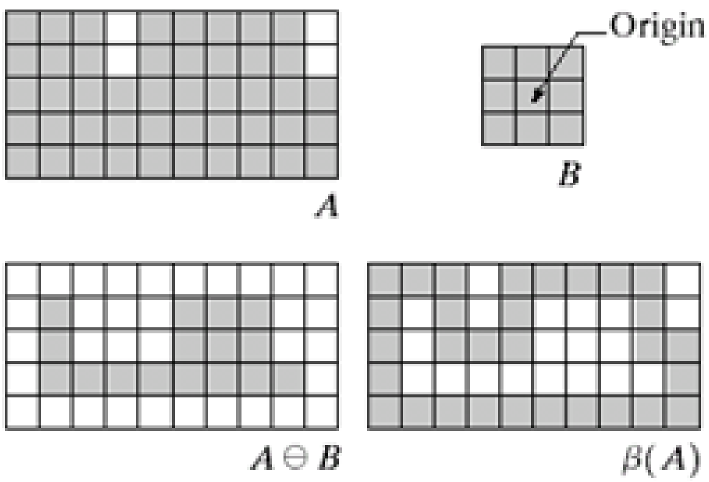
- The boundary of a set A,

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

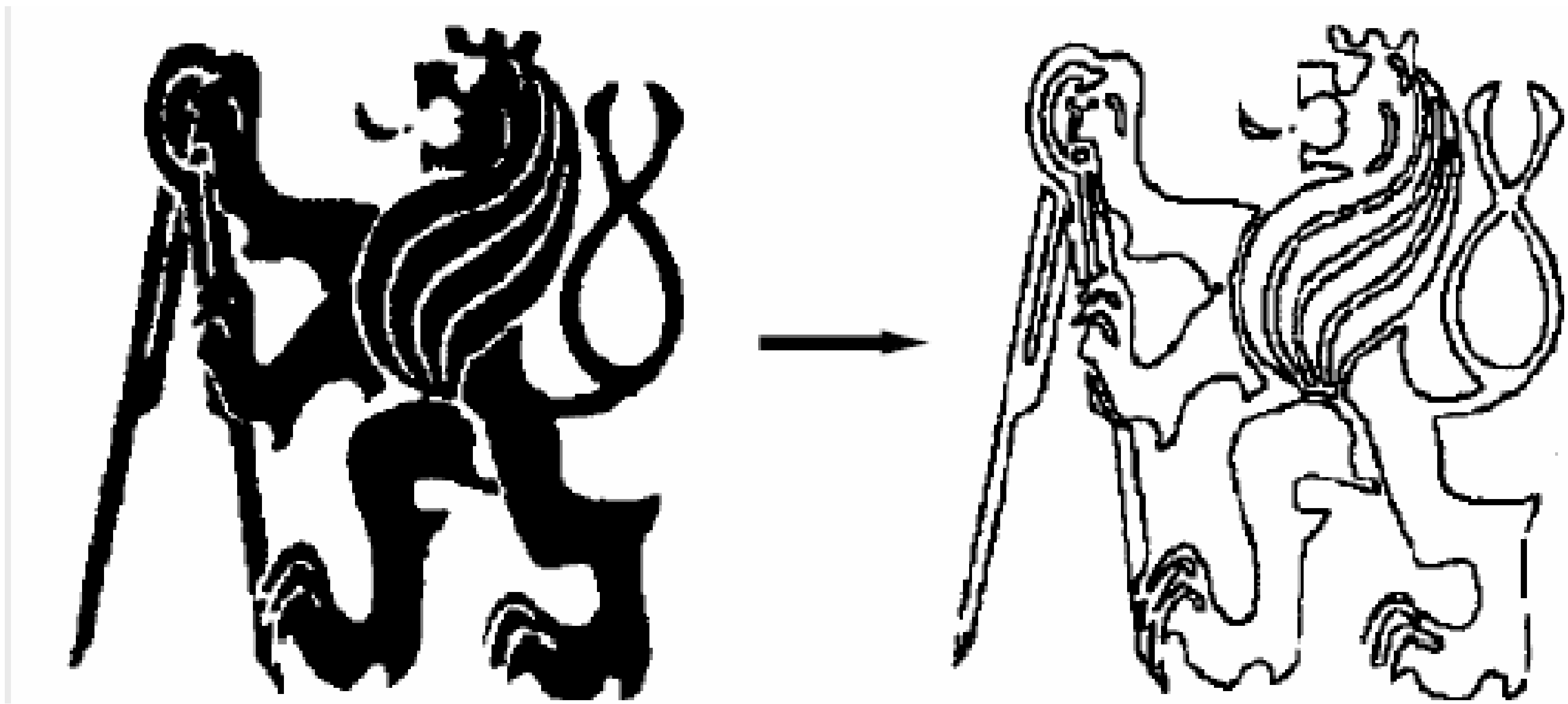
where B is a suitable structuring element.

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.



Boundary Extraction (Cont.)



Boundary Extraction (Cont.)

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).

Boundary Extraction (Cont.)

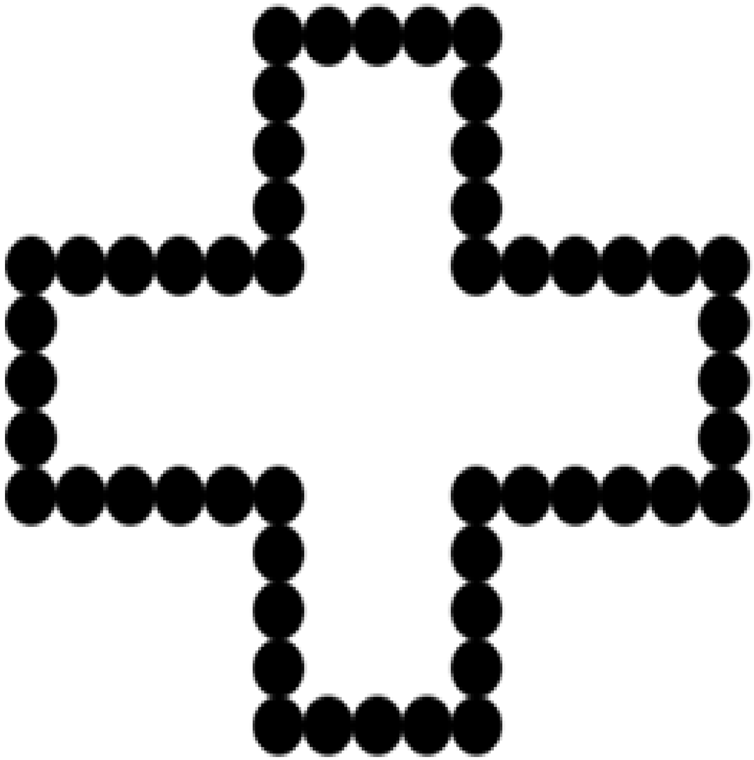
- **Boundary Extraction with the help of Dilation:**

```
A=imread('linkon.tif');
s=strel('disk',3);%Structuring element
F=imdilate(A,s); %Dialte the image by structuring element
 figure,imshow(A);title('Original Image');
figure,imshow(F);title('Imdilate Image');
figure,imshow(F-A);title('Boundary extracted Image with using imdilate');
```

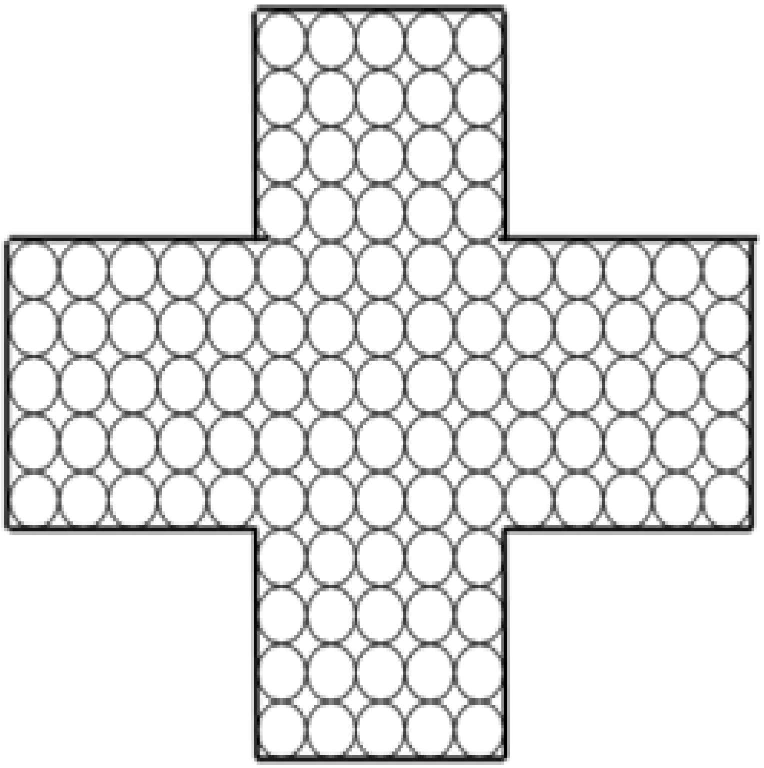
- **Boundary Extraction with the help of Erosion:**

```
A=imread('linkon.tif');
 s=strel('disk',3); %Structuring element
F=imerode(A,s); %Erode the image by structuring element
figure,imshow(A); title('Original Image');
figure,imshow(A-F); title('Boundary extracted Image with using imerode');
```

Hole Filling / Region Filling



Boundary Filled Region



Interior or Flood Filled Region

Hole Filling / Region Filling (Cont.)

Original Image - A

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

Complement of Original Image - B

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

Take Inside Pixels only – Image C

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

Add Original Image A and Image C

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

Hole Filling / Region Filling (Cont.)

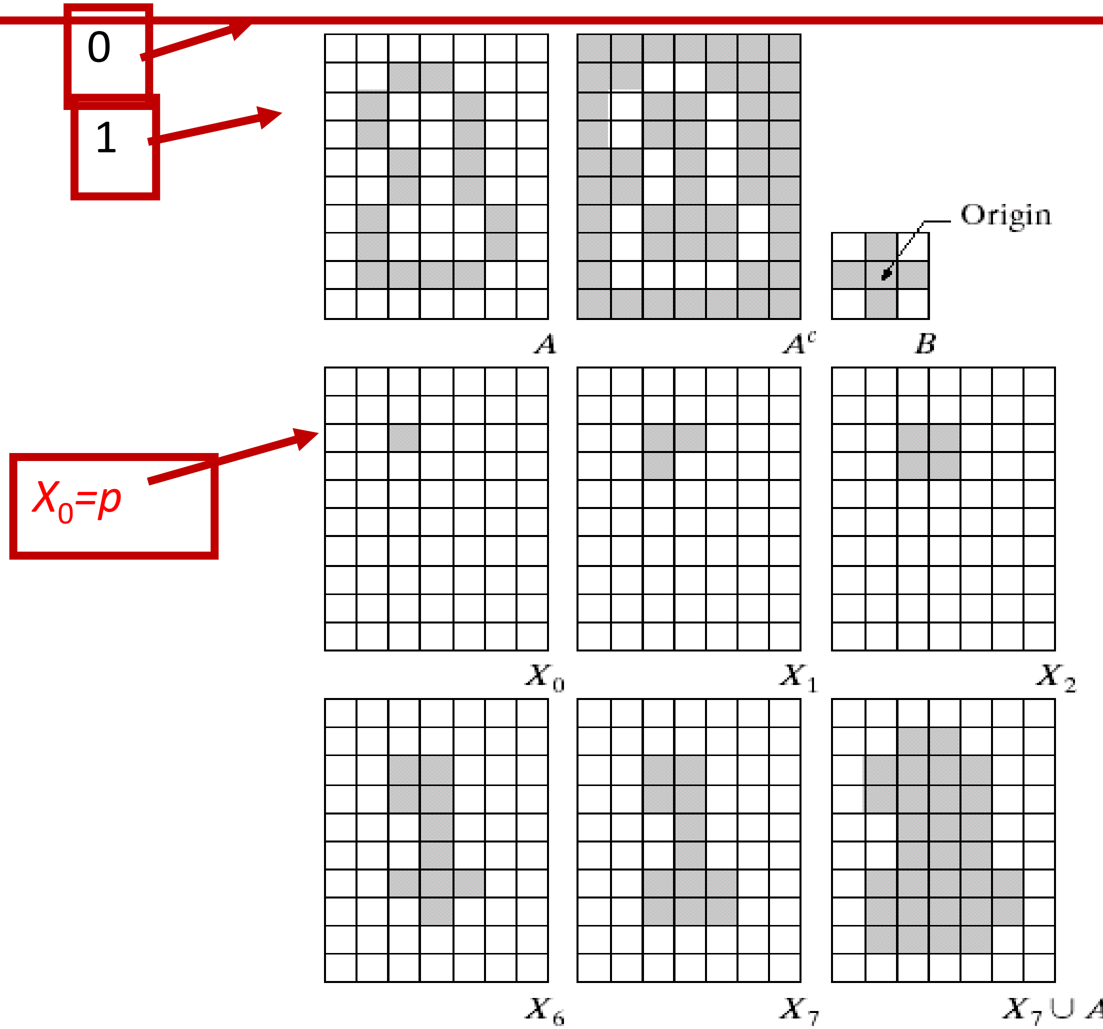
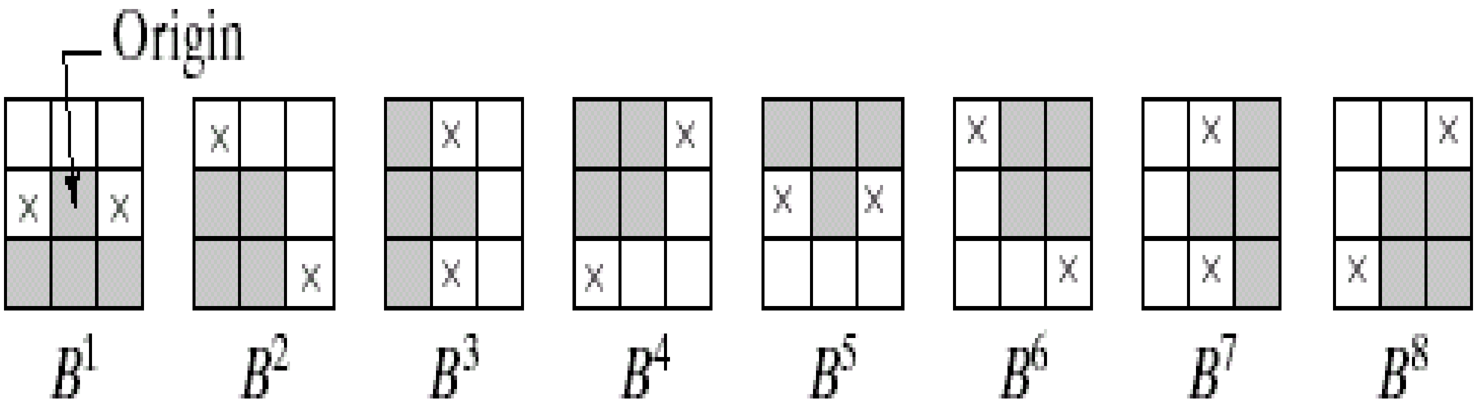


FIGURE 9.15 Region filling. (a) Set A. (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

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

Thinning & Thickening



$$\mathrm{thin}(I, J) = I - \text{hit-and-miss}(I, J)$$

$$\mathbf{thicken}(I, J) = I \cup \textbf{hit-and-miss}(I, J)$$

Thinning & Thickening (Cont.)

- Thinning
  - (Center Element) 0 - If Completely Match (All 0 and All 1)
  - As it is - If Not Match

- Thickeing
  - (Center Element) 1 - If Completely Match (All 0 and All 1)
  - As it is - If Not Match

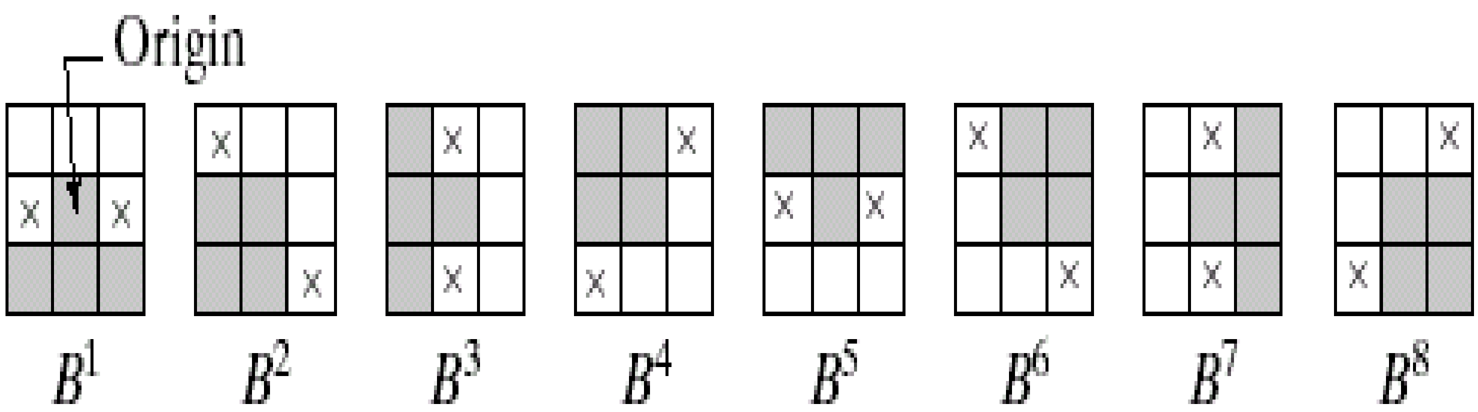
Note: Do Not Consider don't care condition

Thinning & Thickening (Cont.)

Thinning

- **0 - If Completely Match (All 0 and All 1)**
- **As it is - If Not Match**

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

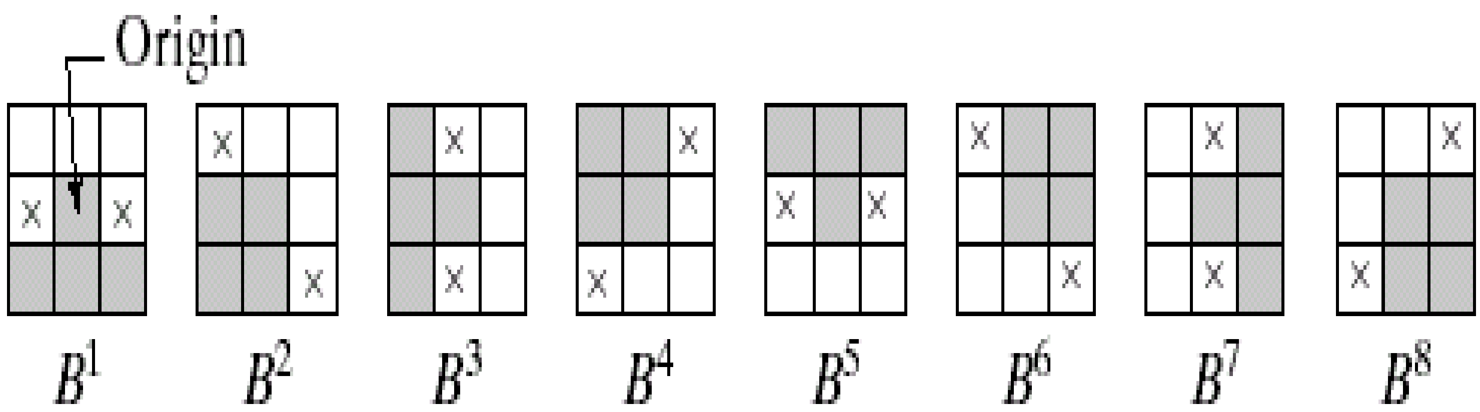


Thinning & Thickening (Cont.)

Thinning

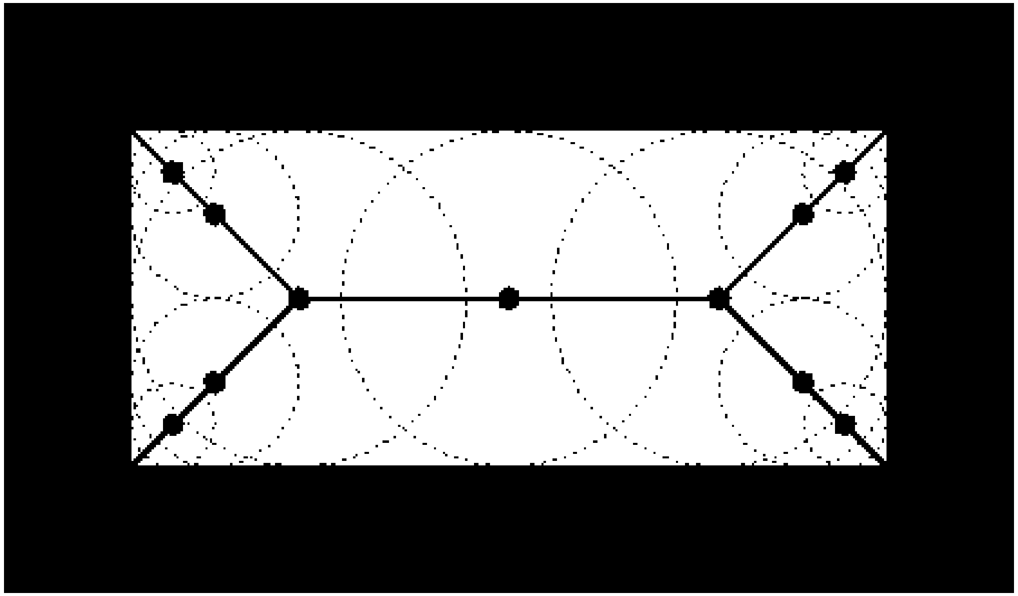
- **1 - If Completely Match (All 0 and All 1)**
- **As it is - If Not Match**

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



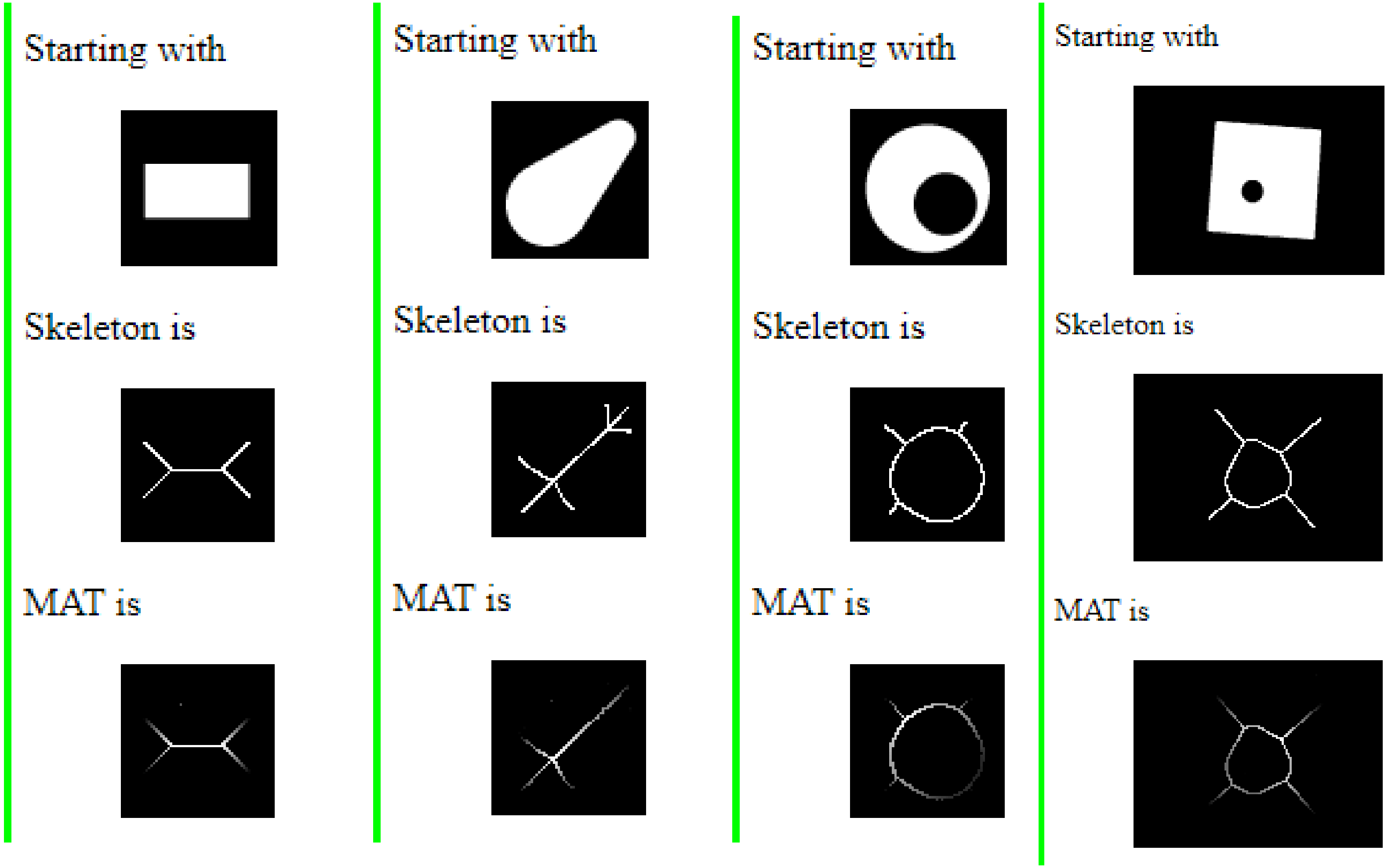
Skeletonization / Skeletons

- Skeletonization is a way to reduce dimensionality of digital objects



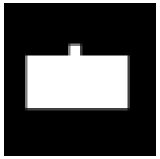
Skeletonization / Skeletons (Cont.)

- Medial axis transform (MAT)

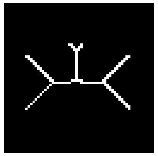


Skeletonization / Skeletons (Cont.)

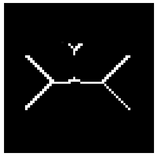
- The skeleton and the MAT are often very sensitive to small changes in the object. If, for example, the above rectangle changes to

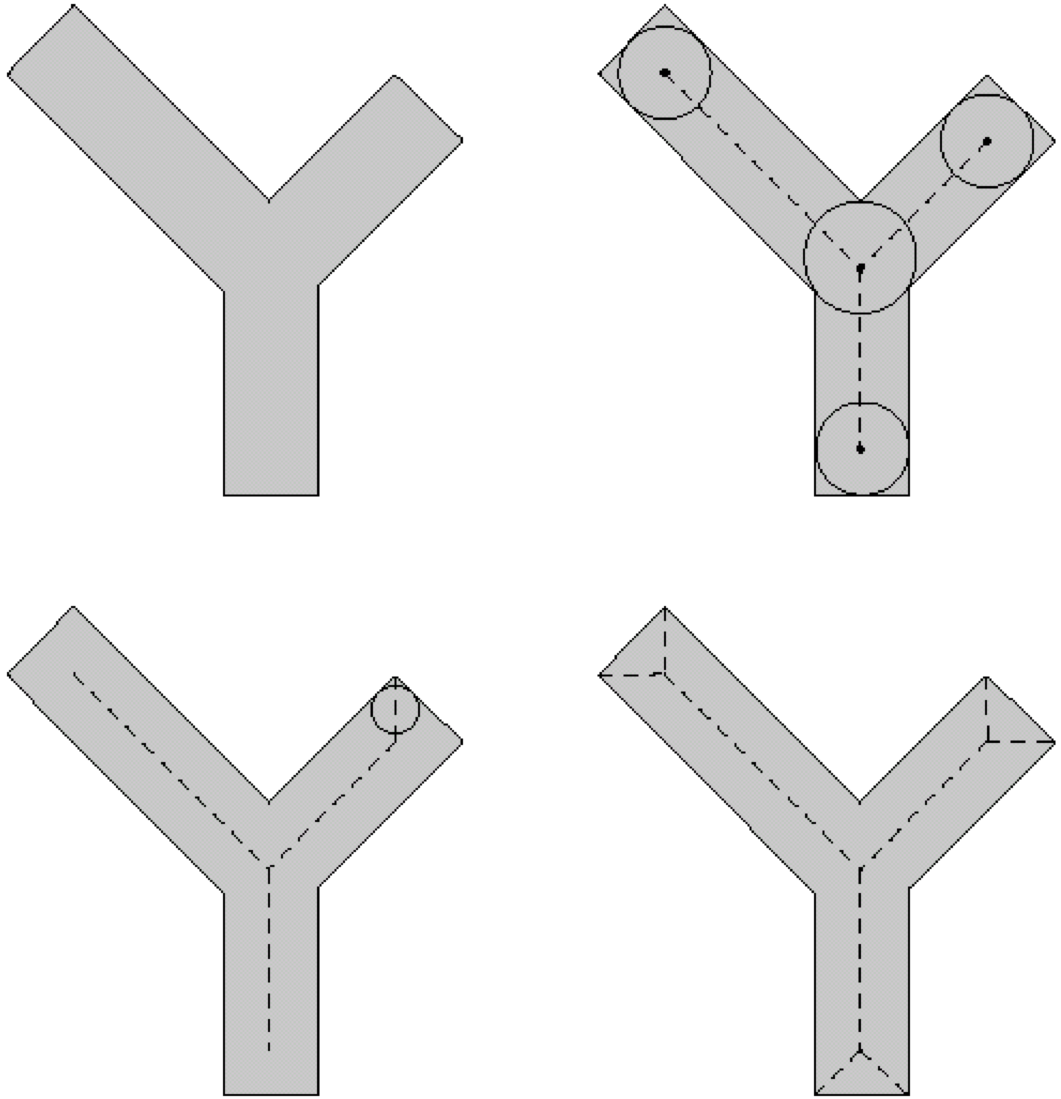


the corresponding skeleton becomes



Using a different algorithm which does not guarantee a connected skeleton yields



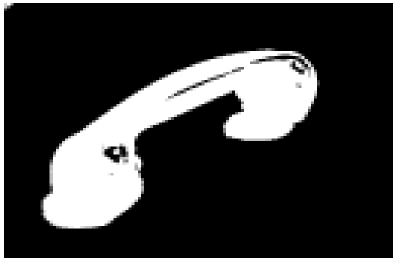


a b
c d

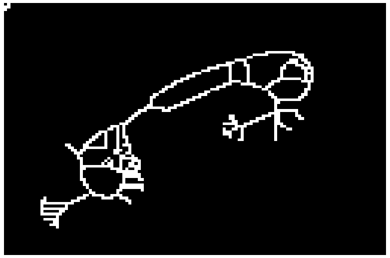
FIGURE 9.23
(a) Set A.
(b) Various positions of maximum disks with centers on the skeleton of A.
(c) Another maximum disk on a different segment of the skeleton of A.
(d) Complete skeleton.

Skeletonization / Skeletons (Cont.)

To obtain a binary image we threshold the image at a value of *100*, thus obtaining

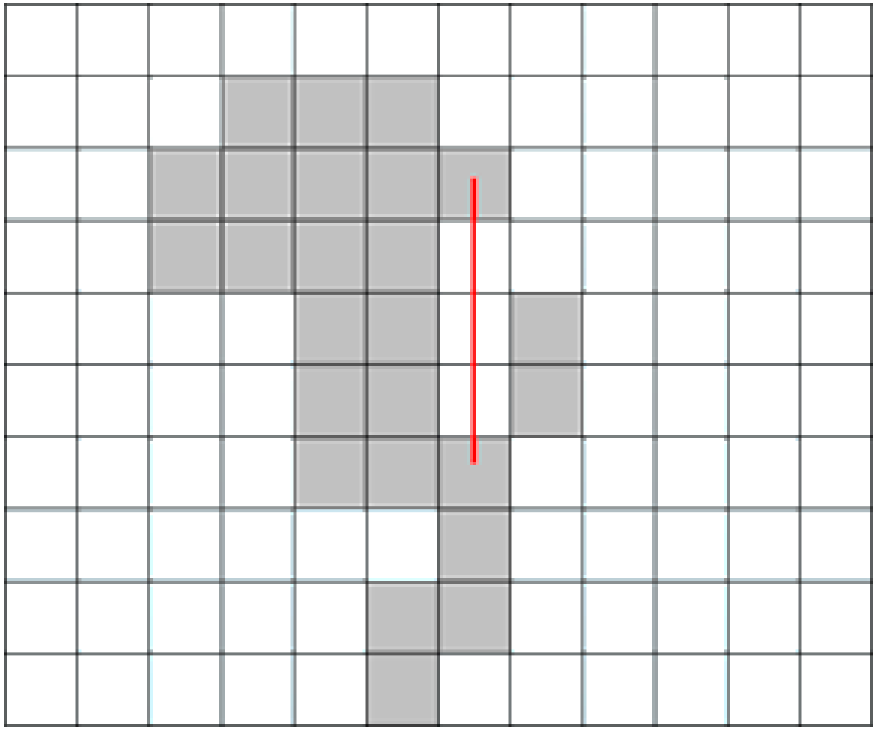


The skeleton of the binary image, shown in



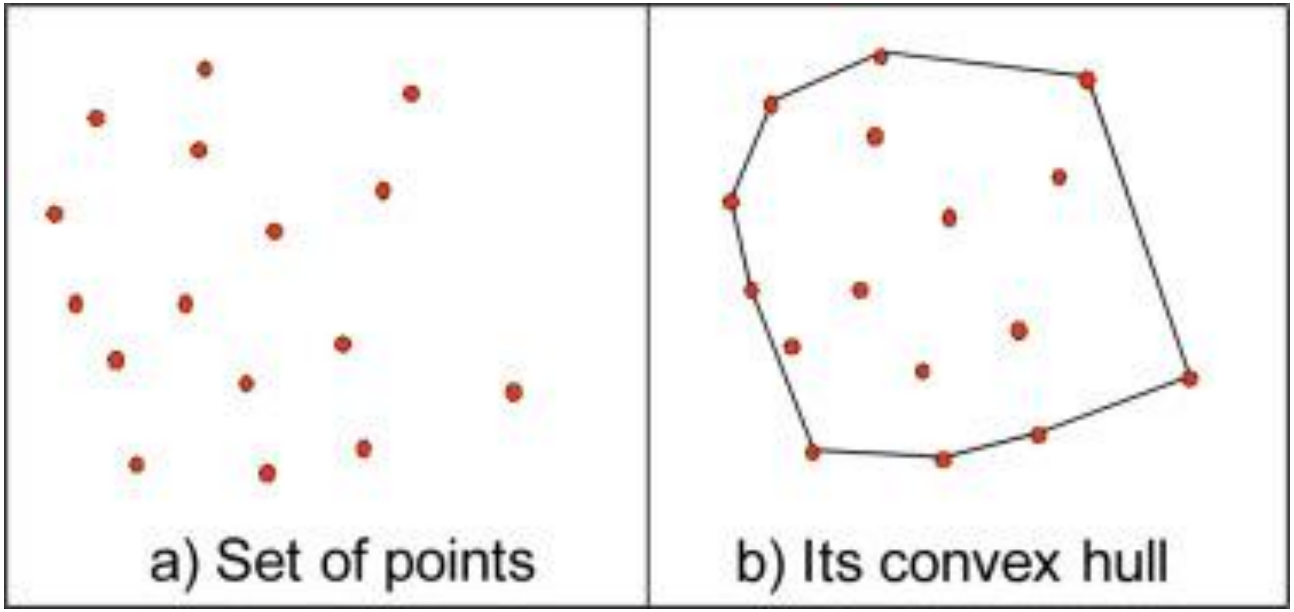
Convex Hull

- A set A is said to be convex if the straight line segment joining any two points in A lies entirely within A.

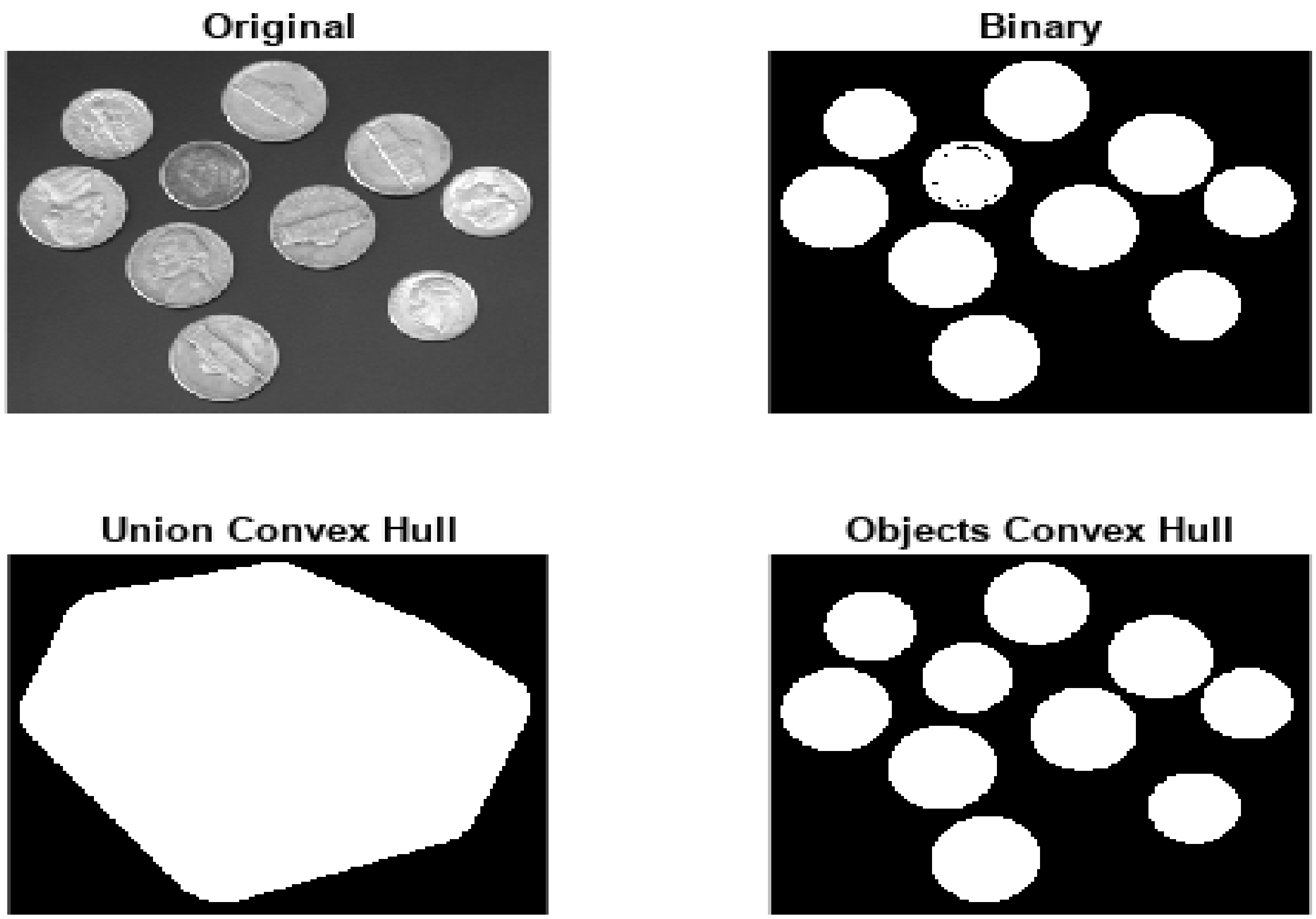


Convex Hull (Cont.)

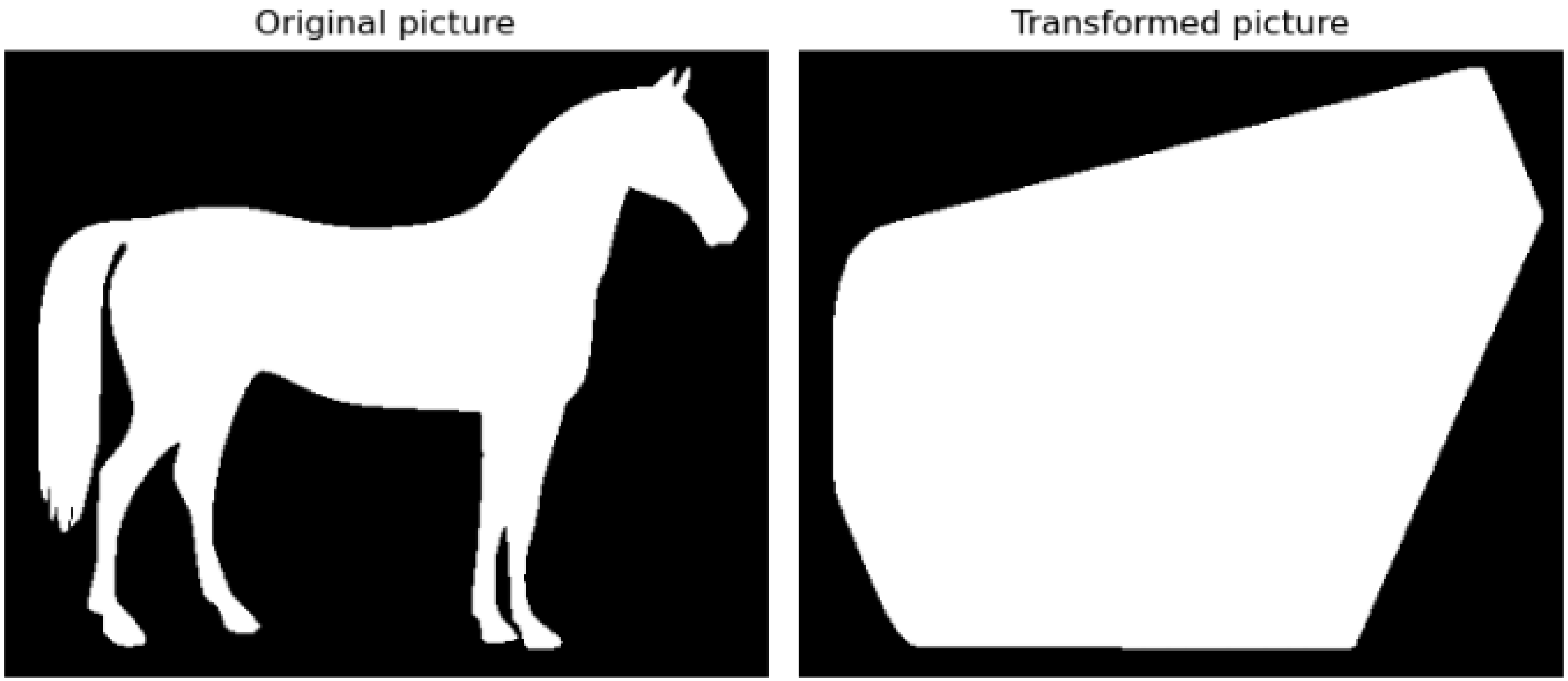
- The convex hull of a binary image is the set of pixels included in the smallest convex polygon that surround all white pixels in the input.



Convex Hull (Cont.)

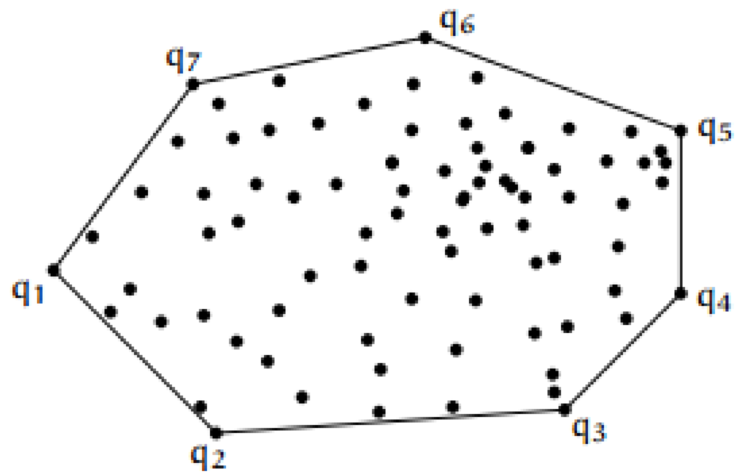


Convex Hull (Cont.)



Convex Hull (Cont.)

(a) Input.



(b) Output.

Extraction of Connected Components

- As its name might suggest, we use it to extract a particular object from an image.
- It is also one of the most important processes to many automated image analysis applications.
- It actually works very similar to hole filling process.
- However, the difference between the two is that we seek foreground pixels here while hole filling seeks background pixels.

Morphological Algorithms in MATLAB

```
%Read binary image and display it.
BW = imread('circles.png');
imshow(BW);
title('Original Image')

%Boundary Extraction
BW2 = bwmorph(BW,'remove');
figure
imshow(BW2)
title('Boundary Extraction')

%Get the image skeleton.
%Skeletonization
BW3= bwskel(BW);
%BW3= bwmorph(BW,'skel',Inf);
figure
imshow(BW3)
title('Skeleton of Image');

%Thickening
BW4 = bwmorph(BW,'thicken');
figure
imshow(BW4)
title('Thickening ');

%Thinning
BW4 = bwmorph(BW,'thin');
figure
imshow(BW4)
title('Thinning');

%Convex Hull of Image
BW5 = bwconvhull(BW);
figure
imshow(BW5);
title('Convex Hull');

%Extraction of Connected Components
%CC = struct with fields:
%Connectivity: 8
% ImageSize: [256 256]
% NumObjects: 88
% PixelIdxList: {1x88 cell}
CC = bwconncomp(BW);
disp(CC.Connectivity);
disp(CC.NumObjects);
```

Morphological Algorithms in MATLAB (Cont.)

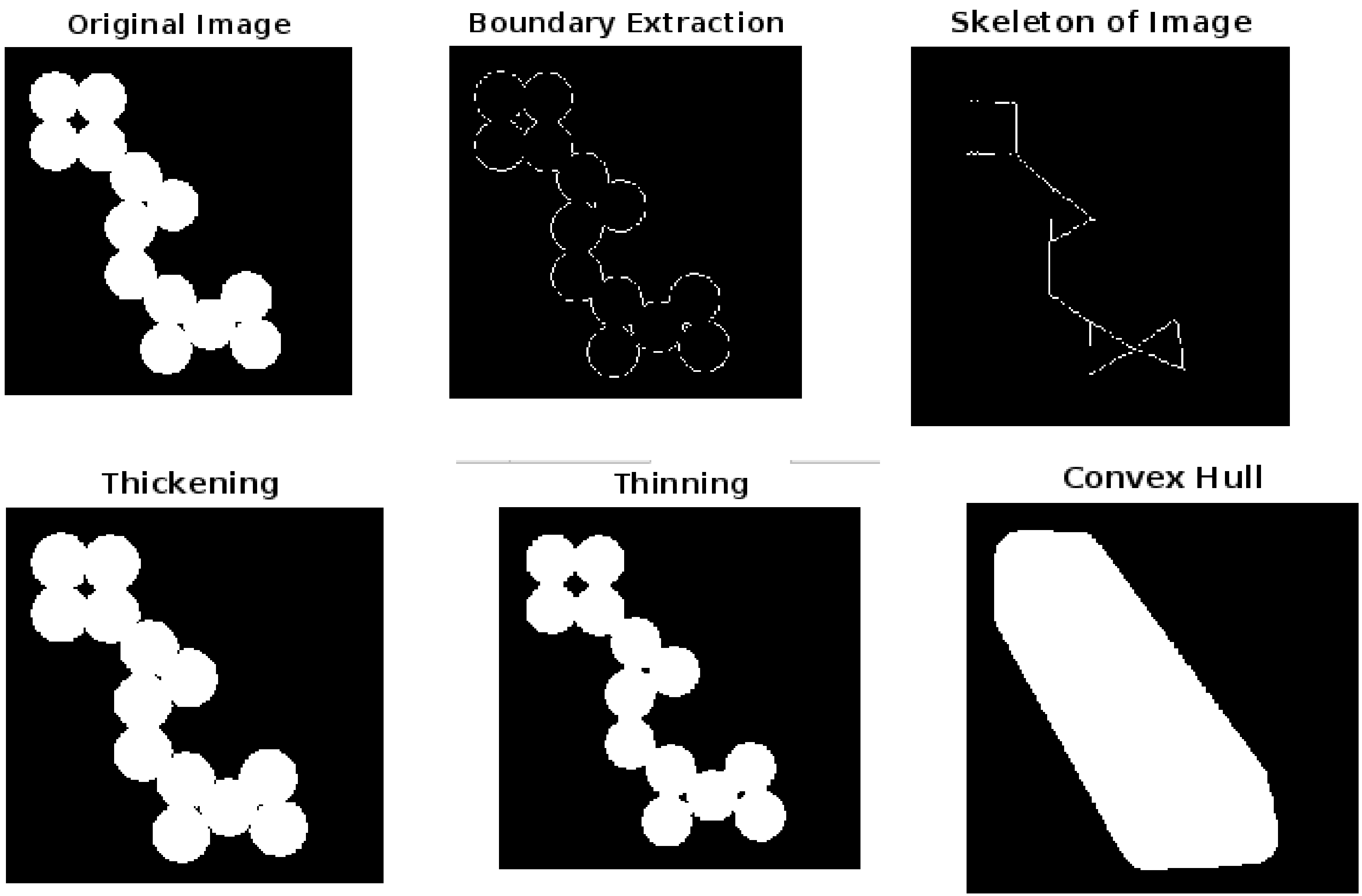


Image Reconstruction

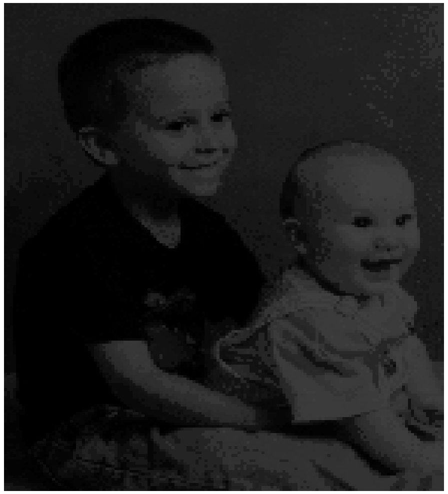
- Facial Images
- Texture Images

Image Reconstruction (Facial Images)

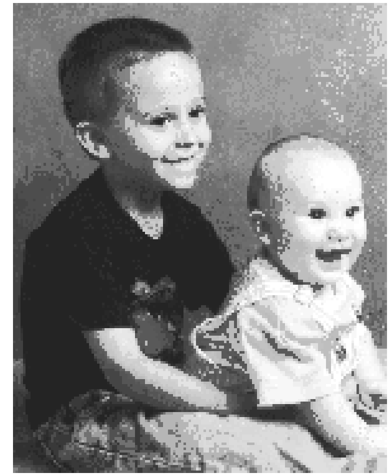
```
%Read and display a grayscale image.
I = imread('kids.tif');
imshow(I)
title ('original image')

%Adjust the contrast of the image to create the mask image and display results.
%mask = adapthisteq(I);
mask=histeq(I);
figure;
imshow(mask)
title ('Histogram Equalization')

%Create a marker image that identifies high-intensity objects in the image using morphological erosion and display results.
se = strel('disk',5);
marker = imerode(mask,se);
figure;
imshow(marker)
title ('Erosion');

%Perform morphological opening on the mask image, using the marker image to identify high-intensity objects in the mask. Display the result.
obr = imreconstruct(marker,mask);
figure;
imshow(obr,[])
title ('Reconstruction of Image');
```

Image Reconstruction (Facial Images)

original image



Histogram Equalization



Erosion

Reconstruction of Image

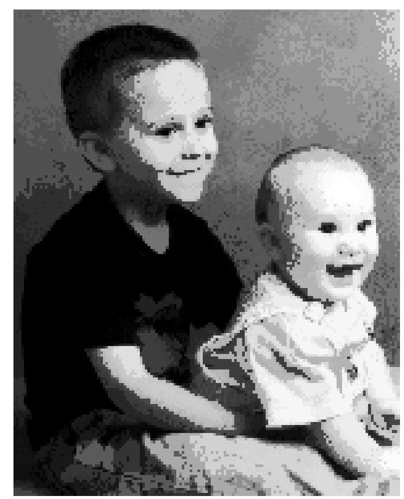
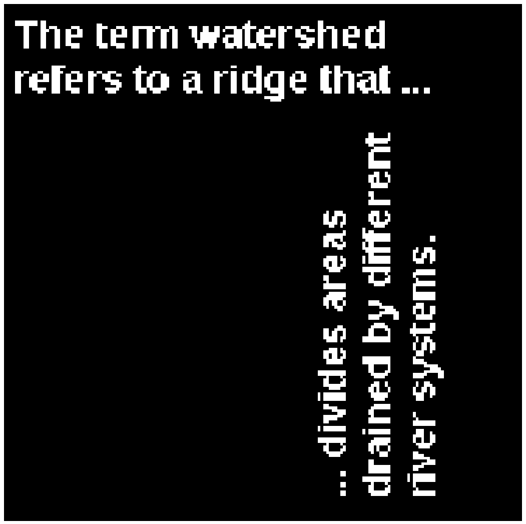


Image Reconstruction (Texture Images)

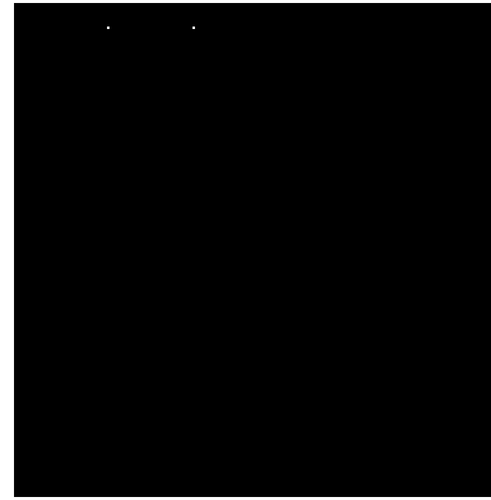
```
I = imread('text.png');
imshow(I)
title('Oiginal Image');

marker = false(size(I));
marker(13,50) = true;
marker(13,94) = true;
marker(13,150) = true;

figure
imshow(marker)
title('Marker Image');

im = imreconstruct(marker,I);
figure
imshow(im)
title('Restored Image');
```

Image Reconstruction (Texture Images)

Oiginal Image



Marker Image



Restored Image


Thank you