Computer Vision

Lecture 5: Binary Image Analysis

Outline

- Introduction to Binary Image Analysis
- Mathematical Morphology (part I)

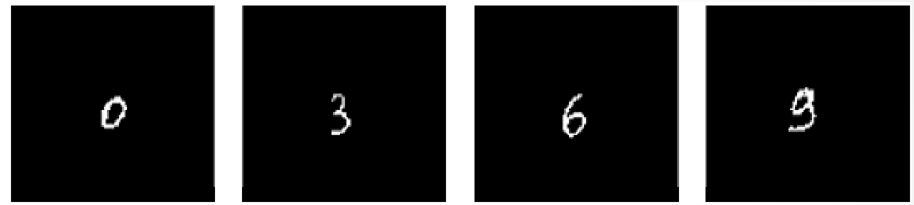
Binary Image Analysis

- Binary image analysis consists of a set of operations that are used to produce or process binary images, usually images of 0's and 1's where
  - 0 represents the background,
  - 1 represents the foreground.

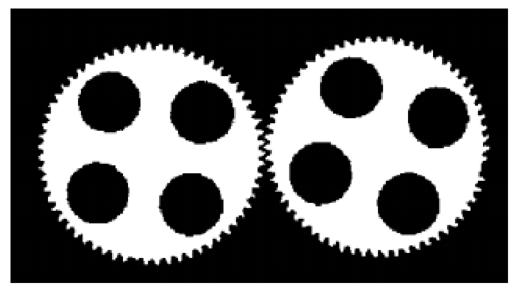
```
00010010001000
00011110001000
00010010001000
```

Application Areas

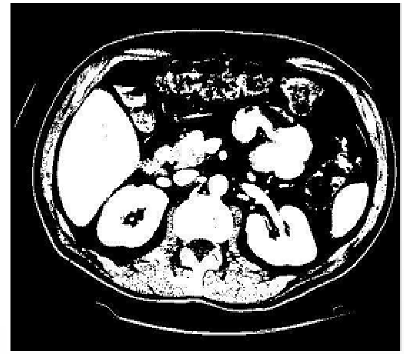
- Document Analysis



- Industrial Inspection



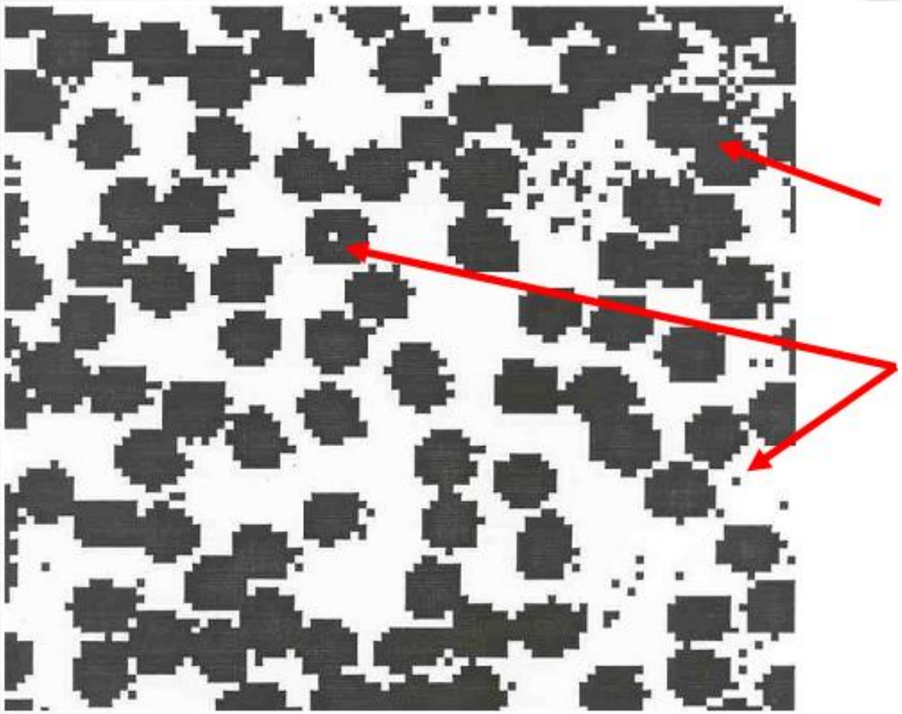
- Medical Imaging



Operations

- Separate objects from background and from one another.
- Aggregate pixels for each object.
- Compute features for each object.

Example: Red Blood Cell Image



- Many blood cells are separate objects..
- Many touch each other → bad!
- Salt and pepper noise is present.
- How useful is this data?

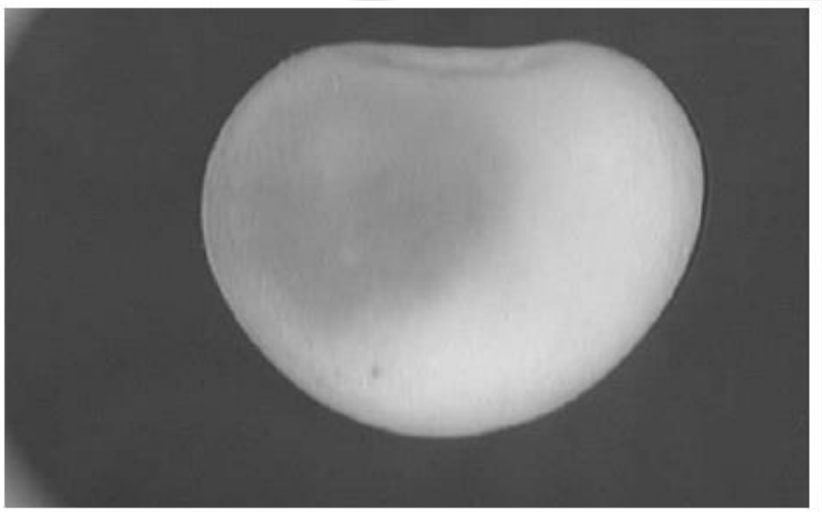
- 63 separate objects are detected.
- Single cells have area of about 50 pixels.

Threshoding

- Binary images can be obtained from gray level images by thresholding.
- Assumptions for thresholding:
  - Object region of interest has intensity distribution different from background.
  - Object pixels likely to be identified by intensity alone
    - intensity > a
    - intensity < b
    - a < intensity < b
- Works OK with flat-shaded scenes or engineered scenes..
- Does not work well with natural scenes.

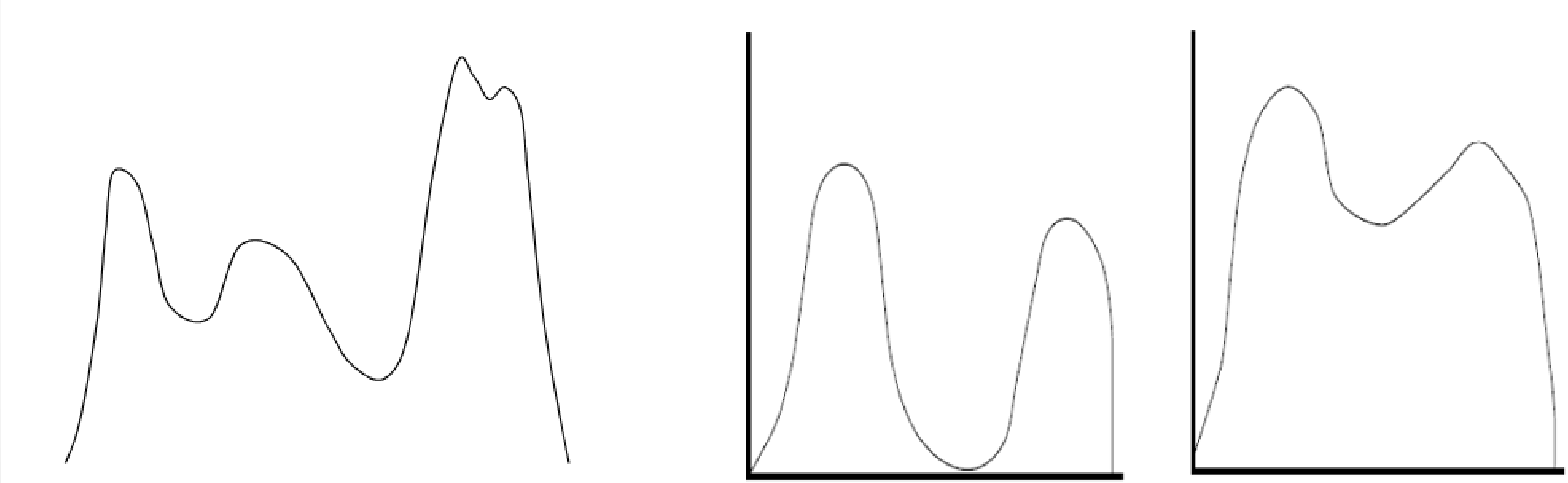
Use of Histograms for Thresholding

- Background is black.
- Healthy cherry is bright.
- Bruise is medium dark.
- Histogram shows two cherry regions (black background has been removed).



Automatic Thresholding

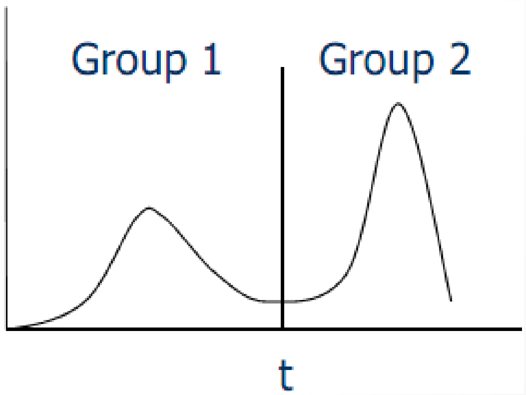
- How can we use a histogram to separate an image into 2 (or several) different regions?



Is there a single clear threshold? 2? 3? Two distinct modes Overlapped modes

Automatic Thresholding: Otsu's Method

- Assumption: the histogram is bimodal.
- Method: find the threshold t that minimizes the <u>weighted sum of within-group variances</u> for the two groups that result from separating the gray levels at value t.
- The best threshold t can be determined by a simple sequential search through all possible values of t.



- If the gray levels are strongly dependent on the location within the image, local or dynamic thresholds can also be used.

Mathematical Morphology

- The word morphology refers to form and structure.
- In computer vision, it is used to refer to the shape of a region.
- The language of mathematical morphology is set theory where sets represent objects in an image.
- We will discuss morphological operations on binary images whose components are sets in the 2D integer space Z^2.

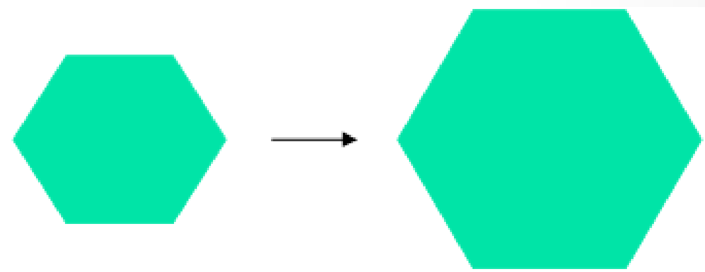
Mathematical Morphology

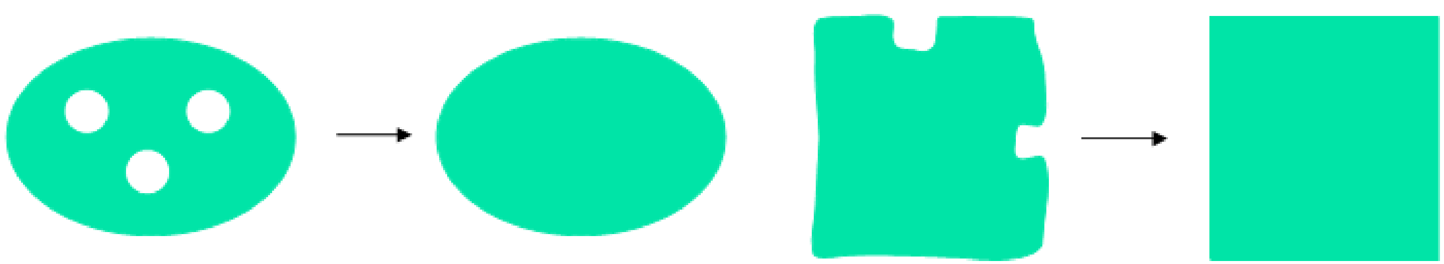
- Mathematical morphology consists of two basic operations:
  - dilation
  - erosion

  and several composite relations
  - opening
  - closing
  - conditional dilation
  - …

Dilation

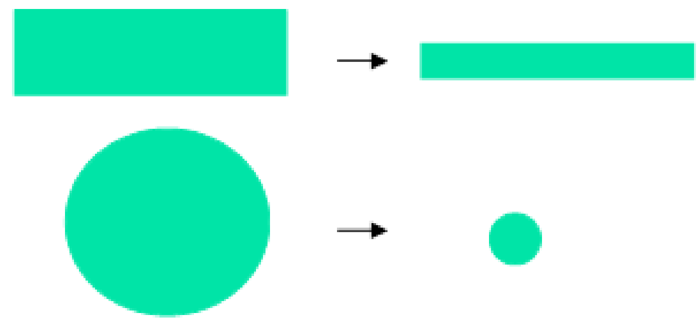
- Dilation expands the connected sets of 1s of a binary image.
- It can be used for
  - growing features
  - filling holes and gaps



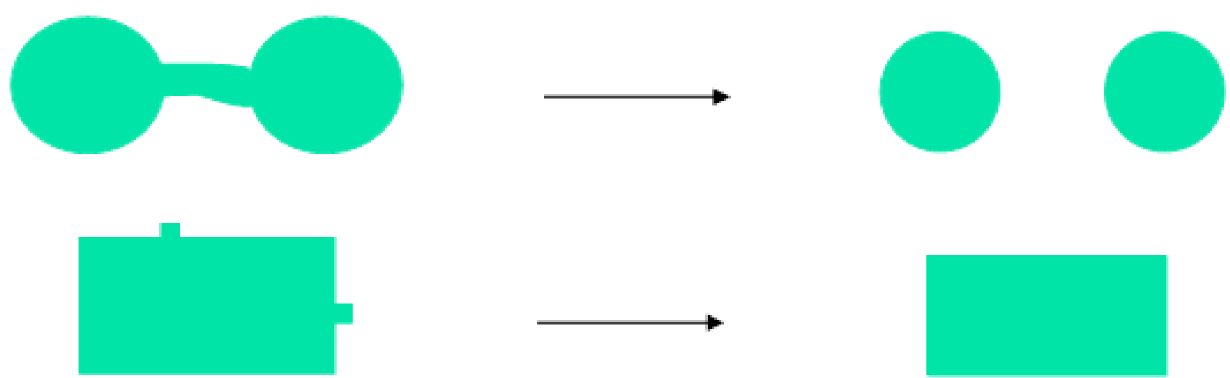


Erosion

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



- It can be used for
  - o shrinking features
  - o removing bridges, branches and small protrusions



Basic Concepts from Set Theory

- Let A be a set in Z^2. If $a = (a_1, a_2)$ is an element of A, we write $a \in A$; otherwise, we write $a \notin A$.
- Set A being a *subset* of set B is denoted by $A \subseteq B$.
- The *union* of two sets A and B is denoted by $A \cup B$.
- The *intersection* of two sets A and B is denoted by $A \cap B$.
- The *complement* of a set A is the set of elements not contained in A:

$$A^c = \{w | w \notin A\}.$$

- The *difference* of two sets A and B, denoted by $A - B$, is defined as

$$A - B = \{w | w \in A, w \notin B\} = A \cap B^c.$$

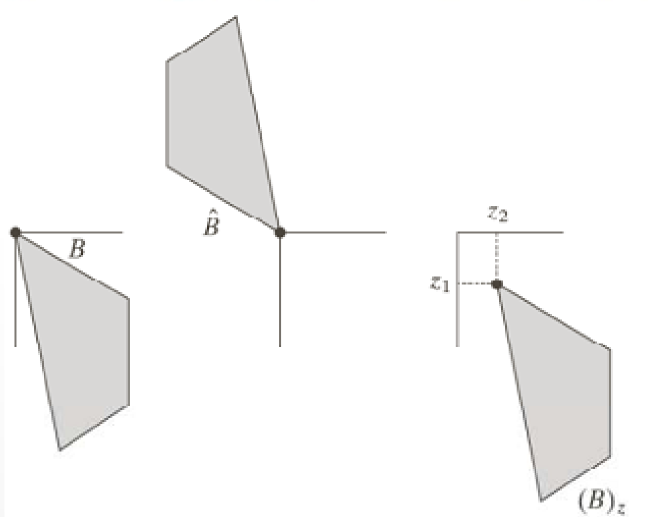
Basic Concepts from Set Theory

- The *reflection* of set *B*, denoted by $\hat{B}$, is defined as

$$\hat{B} = \{w \mid w = -b, \forall b \in B\}.$$

- The *translation* of set *A* by point $z=(z_1,z_2)$, denoted by A_z, is defined as

$$A_z = \{w \mid w = a + z, \forall a \in A\}.$$



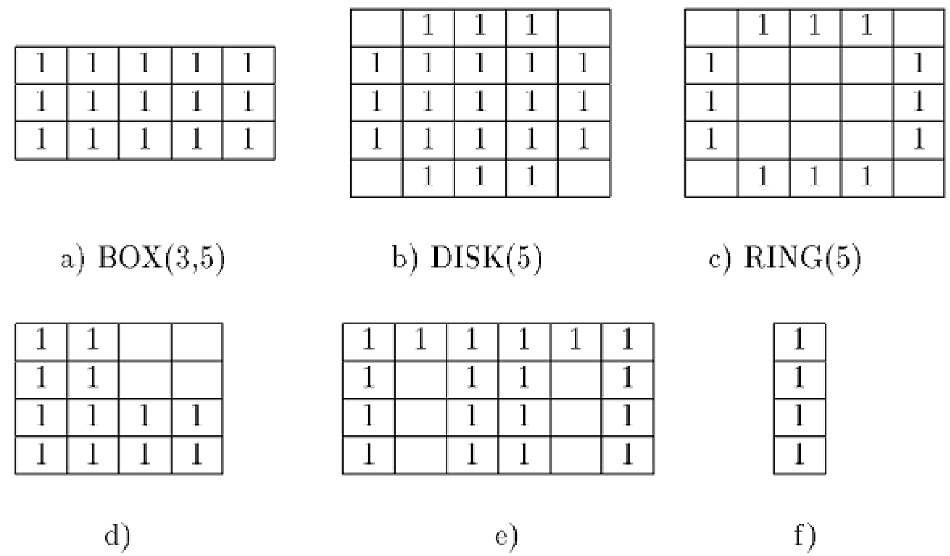
a b c

FIGURE 9.1 (a) A set, (b) its reflection, and (c) its translation by *z*.

Structuring Elements

- Structuring elements are small binary images used as shape masks in basic morphological operations.
- They can be any shape and size that is digitally representable.
- One pixel of the structuring element is denoted as its origin.
- Origin is often the central pixel of a symmetric structuring element but may in principle be any chosen pixel.

Structuring Elements



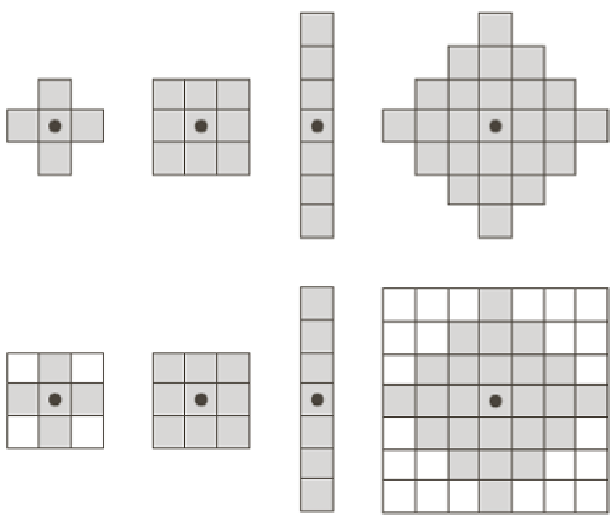


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Dilation

- The *dilation* of binary image A by structuring element B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{z | \check{B}_z \cap A \neq \emptyset\},$$
$$= \bigcup_{a \in A} B_a.$$

  - First definition: The dilation is the set of all displacements z such that $\check{B}_z$ and A overlap by at least one element.
  - Second definition: The structuring element is swept over the image. Each time the origin of the structuring element touches a binary 1-pixel, the entire translated structuring element is ORed to the output image, which was initialized to all zeros.

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1								

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0							

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1						

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1					

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1				

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1			

Example for Dilation

Input image	1	0	0	0	1	1	1	0	1	1
Structuring Element							1	1	1	
Output Image		1	0	1	1	1	1	1		

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

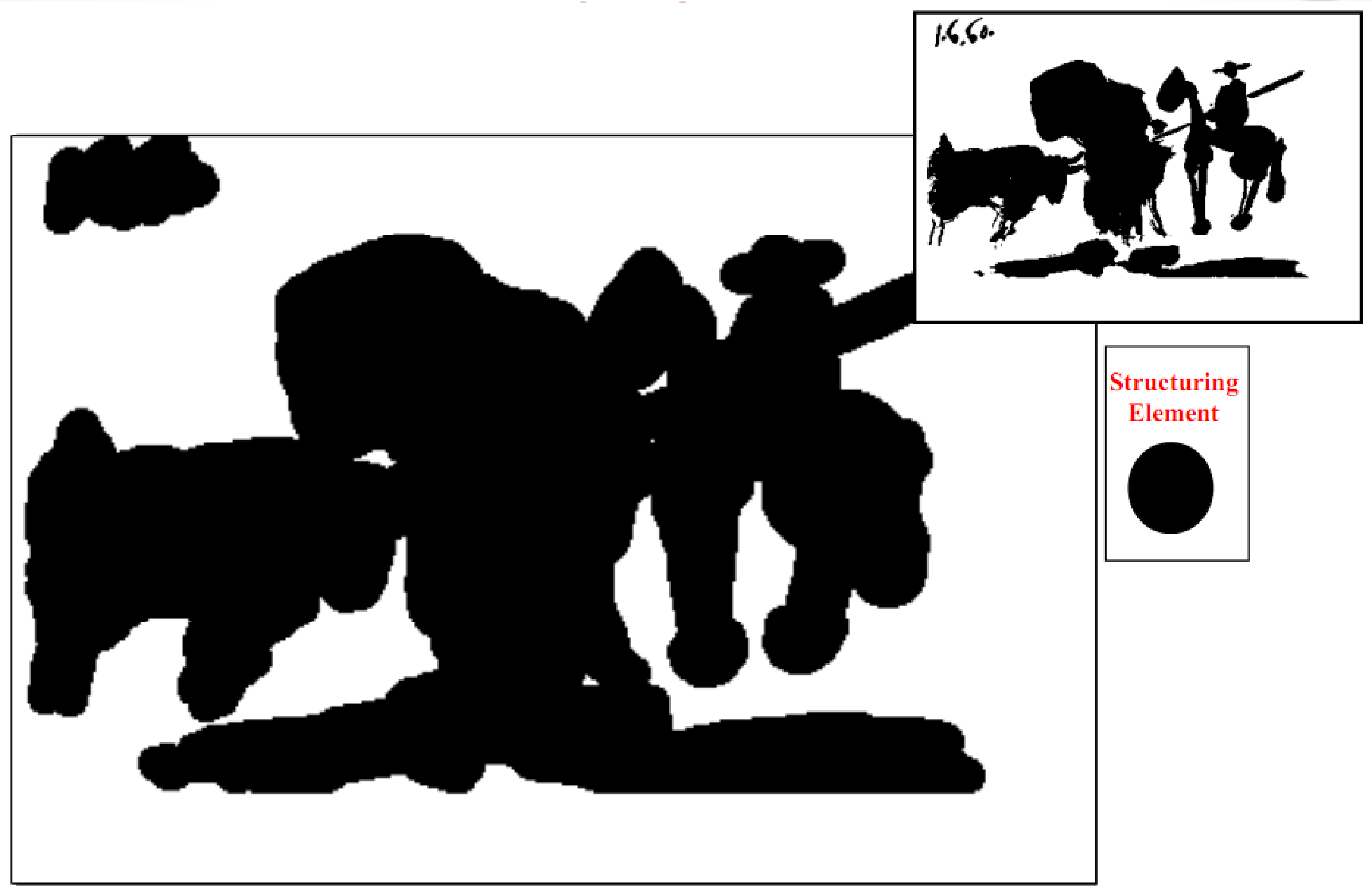
Structuring Element

1	1	1

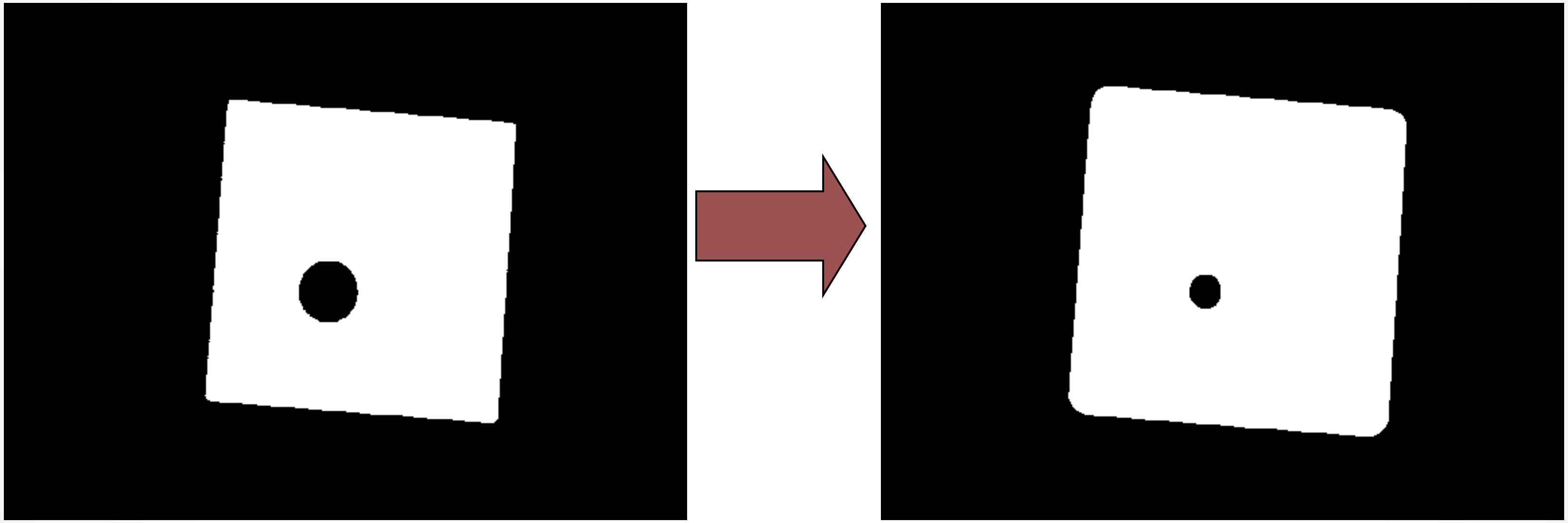
Output Image

	1	0	1	1	1	1	1	1	

Another Dilation Example

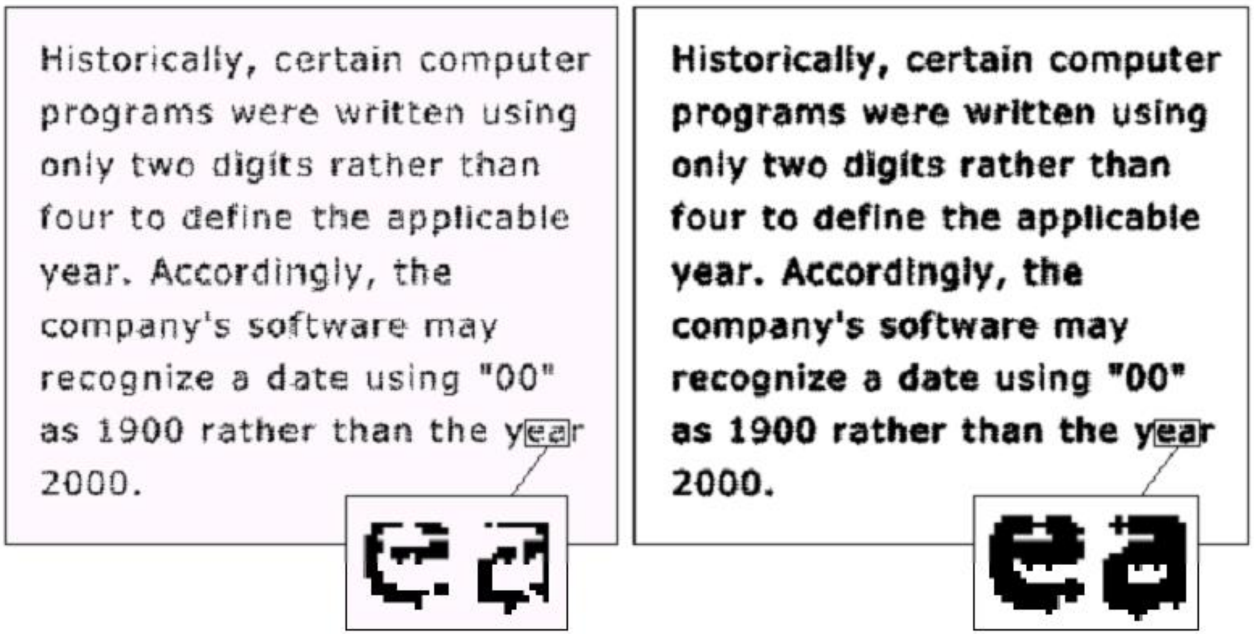


Another Dilation Example



- Image get lighter, more uniform intensity

Another Dilation Example



0	1	0
1	1	1
0	1	0

a c
b

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.

Erosion

- The *erosion* of binary image A by structuring element B is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{z | B_z \subseteq A\},$$
$$= \{a | a + b \in A, \forall b \in B\}.$$

  - First definition: The erosion is the set of all points z such that B, translated by z, is contained in A.
  - Second definition: The structuring element is swept over the image. At each position where every 1-pixel of the structuring element covers a 1-pixel of the binary image, the binary image pixel corresponding to the origin of the structuring element is ORed to the output image.

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0								

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0							

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0						

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0					

Example for Erosion

Input image	1	0	0	0	1	1	1	0	1	1
Structuring Element					1	1	1			
Output Image		0	0	0	0	1				

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1	0			

Example for Erosion

Input image	1	0	0	0	1	1	1	0	1	1
Structuring Element							1	1	1	
Output Image		0	0	0	0	1	0	0		

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

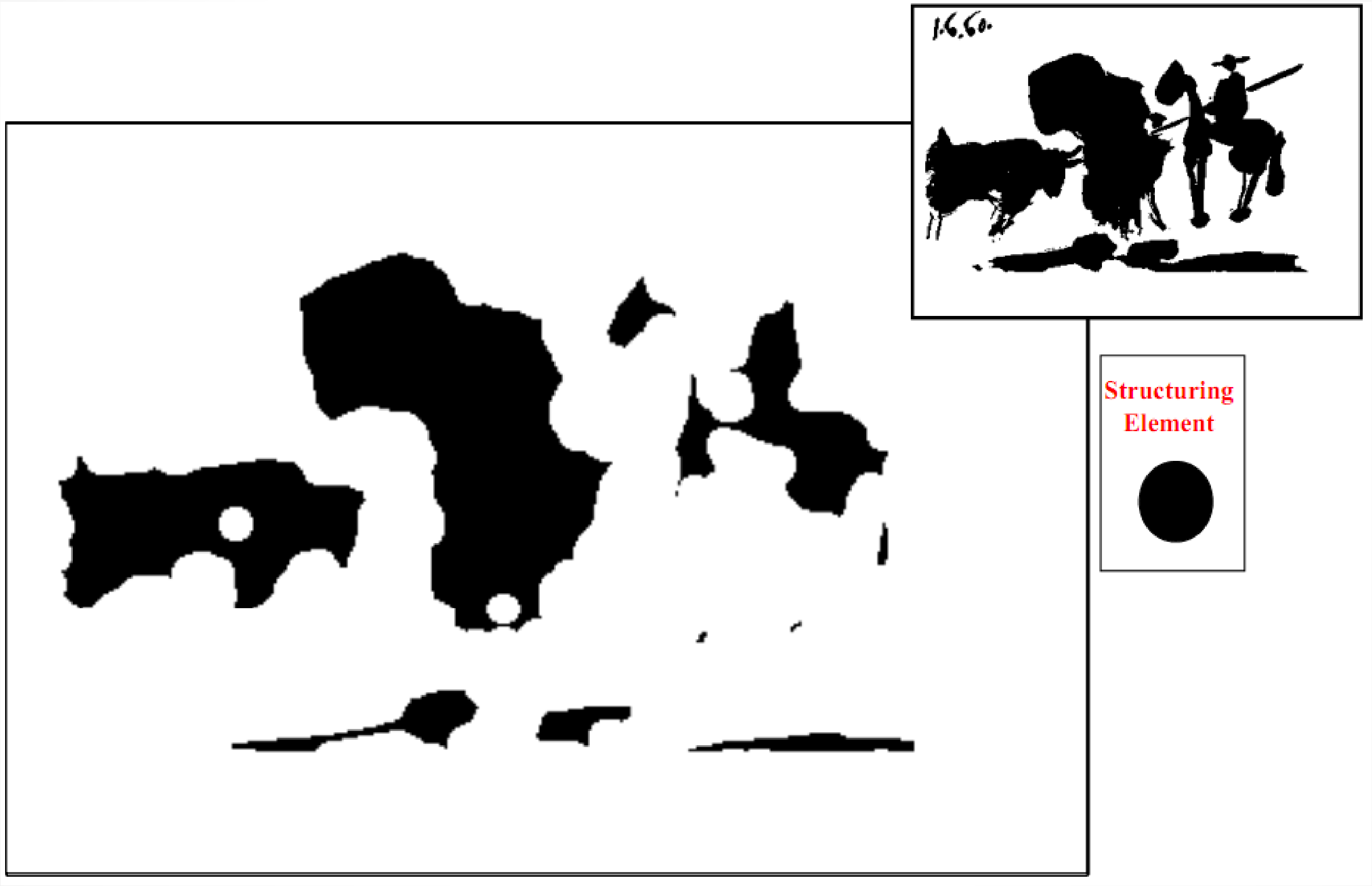
Structuring Element

1	1	1

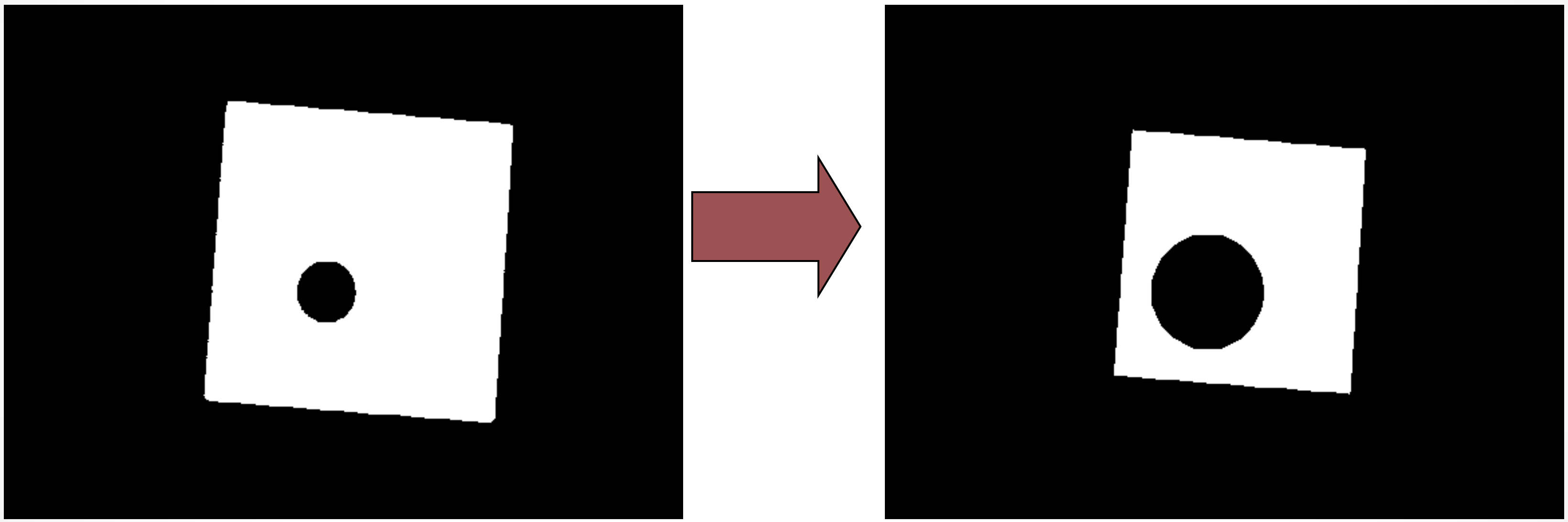
Output Image

	0	0	0	0	1	0	0	0	

Another Erosion Example

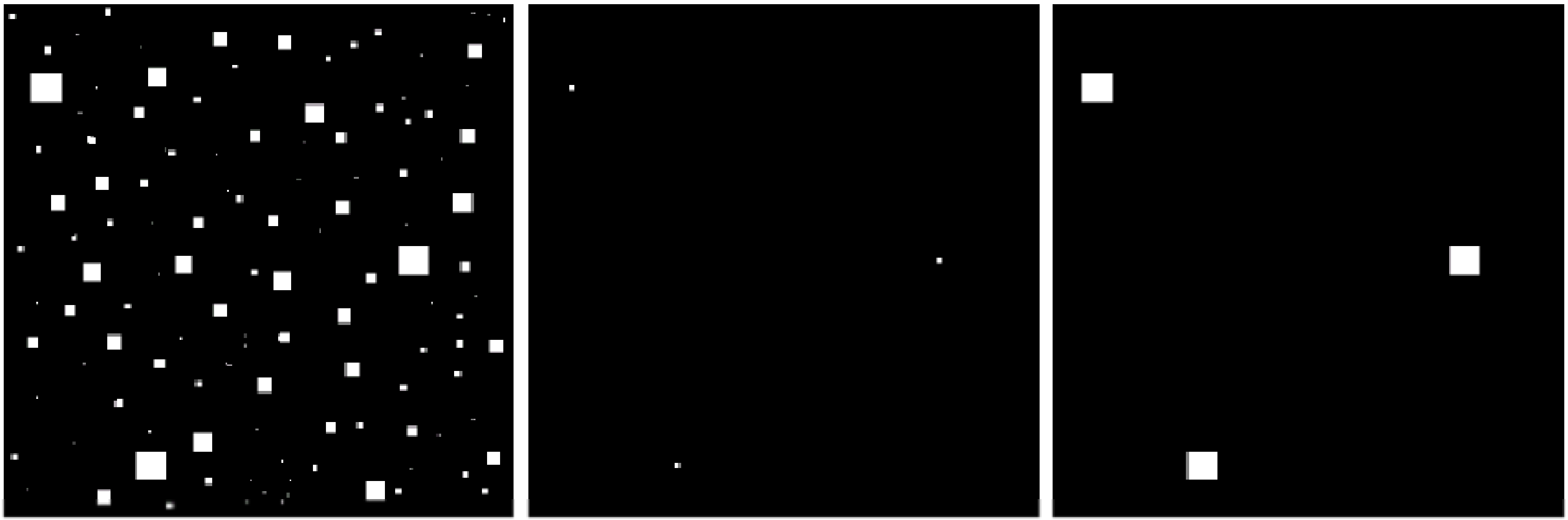


Another Erosion Example



- White = 0, black = 1, dual property, image as a result of erosion gets darker

Another Erosion Example



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.

Opening

- The *opening* of binary image A by structuring element B is denoted by $A \circ B$ and is defined by

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

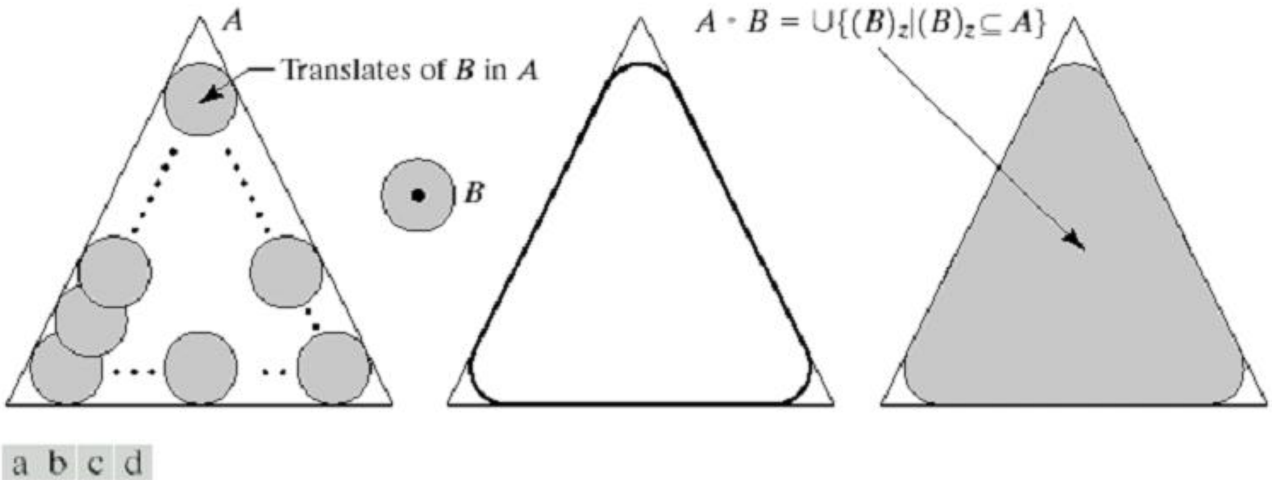


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

Opening

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

Binary image A

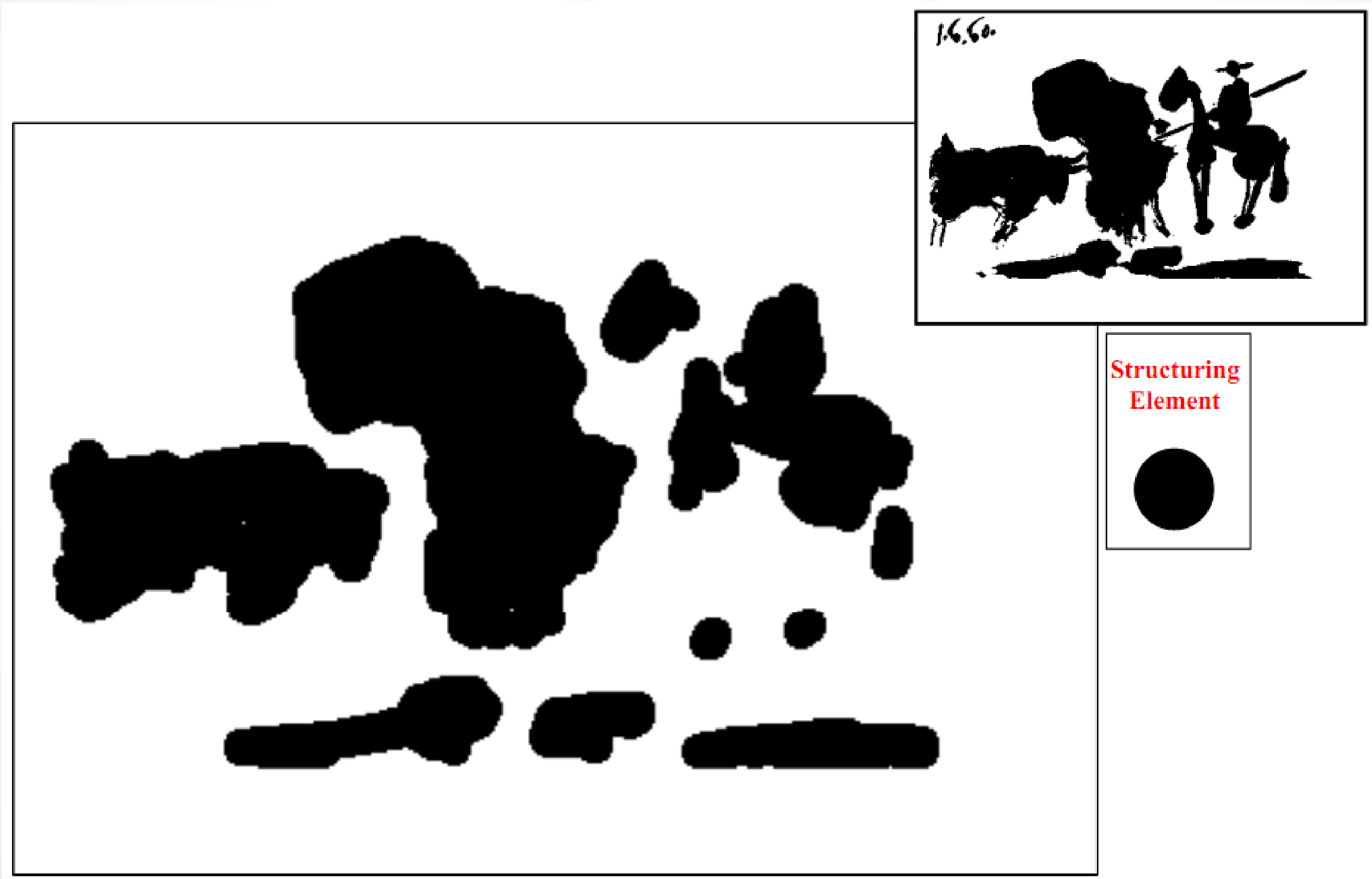
1	1	1
1	1	1
1	1	1

Structuring element B

			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	

Opening result

Opening



Closing

- The *closing* of binary image A by structuring element B is denoted by $A \bullet B$ and is defined by

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

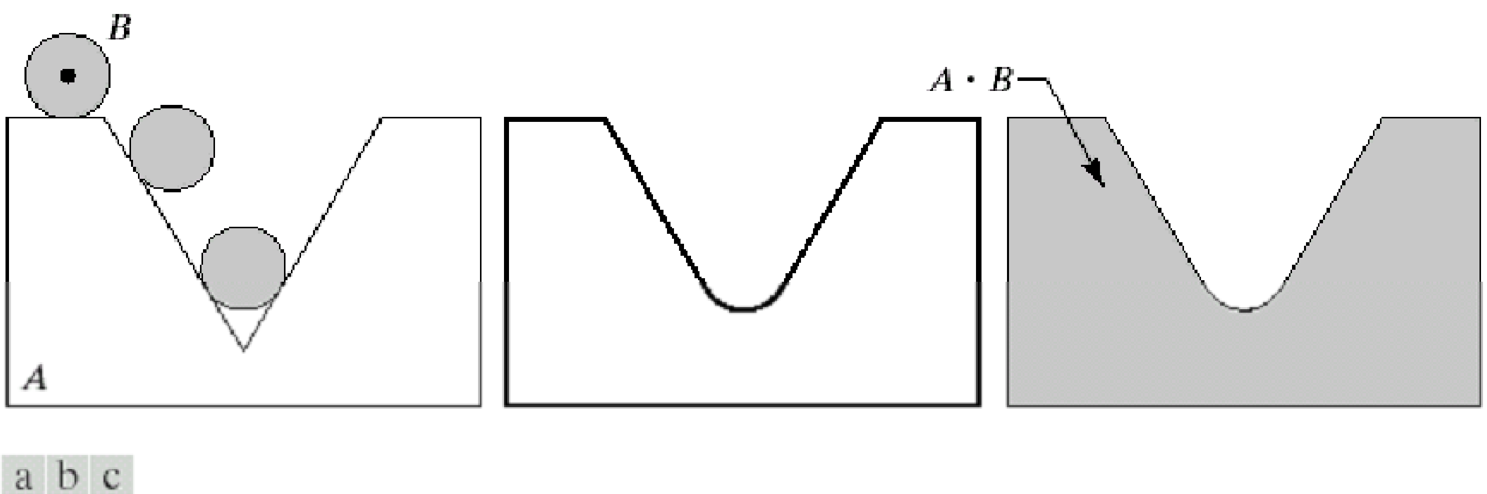


FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

Closing

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

Binary image A

1	1	1
1	1	1
1	1	1

Structuring element B

1	1	1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1				

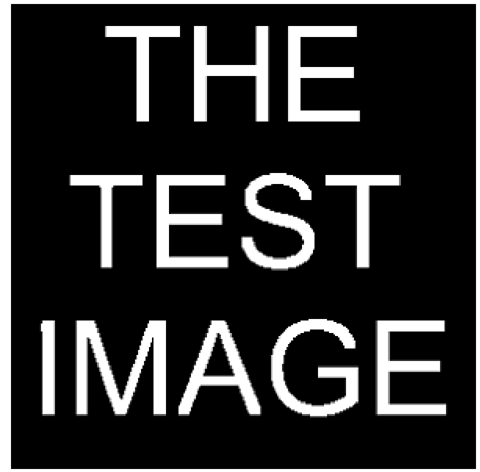
Closing result

Examples

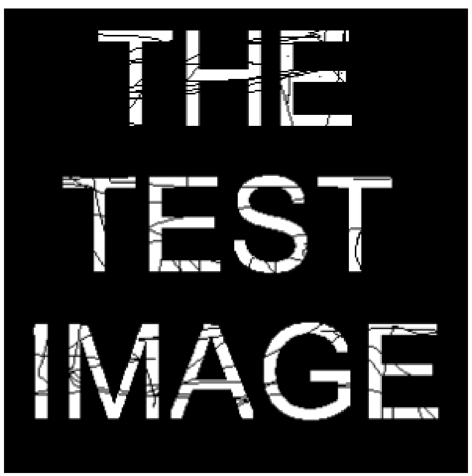
Original image | Eroded once | Eroded twice

Examples

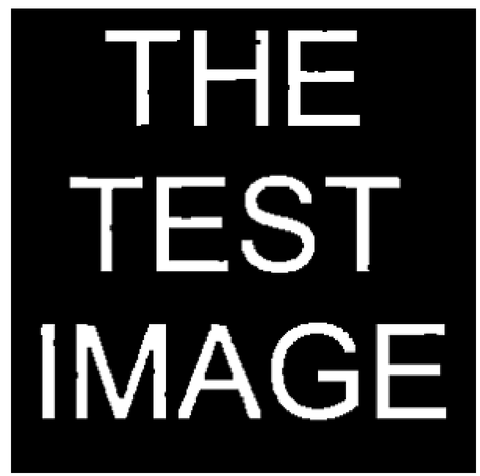
Original image



Opened twice

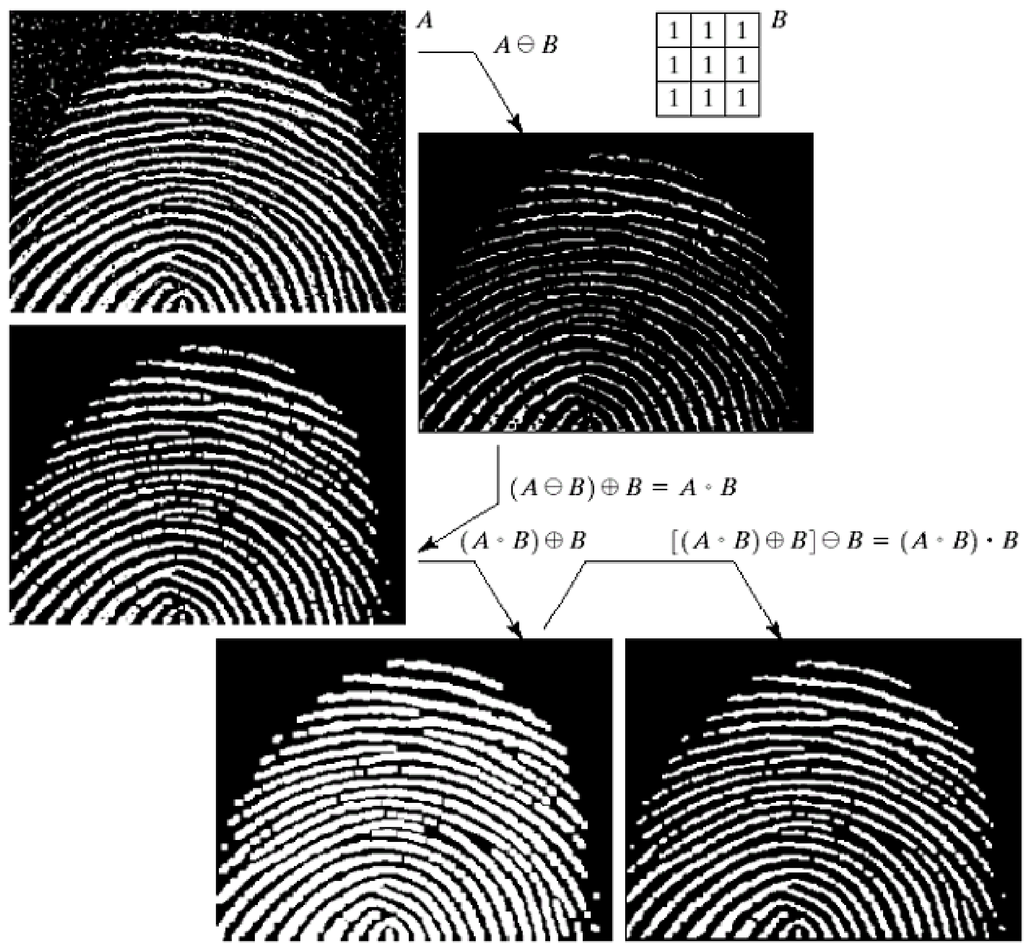


Original image



Closed once

Examples



a b
d c
e f

FIGURE 9.11
(a) Noisy image.
(c) Eroded image.
(d) Opening of A.
(d) Dilation of the opening.
(e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

Properties

- Dilation and erosion are duals of each other with respect to set complementation and reflection, i.e.,

$$(A \ominus B)^c = A^c \oplus \check{B}.$$

- Opening and closing are duals of each other with respect to set complementation and reflection, i.e.,

$$(A \bullet B)^c = A^c \circ \check{B}.$$

Properties

- Opening satisfies the following properties:
  - $A \circ B$ is a subset of A.
  - If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.
  - $(A \circ B) \circ B = A \circ B$.
- Closing satisfies the following properties:
  - A is a subset of $A \bullet B$.
  - If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.
  - $(A \bullet B) \bullet B = A \bullet B$.

• • • • • • • • •

For further advanced morphological operations, to be continued in the next lecture…

Assignment 1

Write your own implementations of the morphological dilation and erosion operations. Your programs should input a binary image (as a matrix) and a structuring element (also as a matrix), and produce a binary image (another matrix) as the result of the operation.

You can generate the structuring element as a binary image with an arbitrary shape or use a predefined structure (such as a square or a disc) with a user-defined parameter for its size (such as the length of the side of the square or the diameter of the disc). Given the structuring element, your code should implement the dilation and erosion operations using the definitions given in the course slides. Note that the structuring element is created outside and given as an input to the dilation/erosion codes so that these codes can work with any kind of structuring element. You are free to use any programming language.

Submit: Well-documented source code in ASCII format for dilation and erosion operations. Also cite the definition you used for the implementation in the code documentation.
for the implementation. The representation of the image data and the structuring element data (using data structures such as arrays, lists, etc.) will depend on your choice of the language. You MUST write your own implementations of these two morphological operations. Code from other sources is NOT allowed for this part of the assignment (as an exception, you can use the strel function in Matlab to generate the arrays containing the structuring elements). Contact the lecturer or TA.

Thank you for your attention and attendance.