IE404

Digital Image Processing

Dr. Manish Khare

Morphological Image Processing

Preview

- The word Morphology commonly denotes a branch of biology that deals with the form and structure of animals and plants

- We use mathematical morphological as a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries extraction, skeletons, convex hull, morphological filtering, thinning, pruning

- Binary images whose components are elements of Z^2 while in gray scale image elements belongs to Z^3

Z^2 and Z^3

- set in mathematic morphology represent objects in an image
  - binary image (0 = white, 1 = black) : the element of the set is the coordinates (x,y) of pixel belong to the object ⇨ Z^2

- gray-scaled image : the element of the set is the coordinates (x,y) of pixel belong to the object and the gray levels ⇨ Z^3

Basic Set Theory

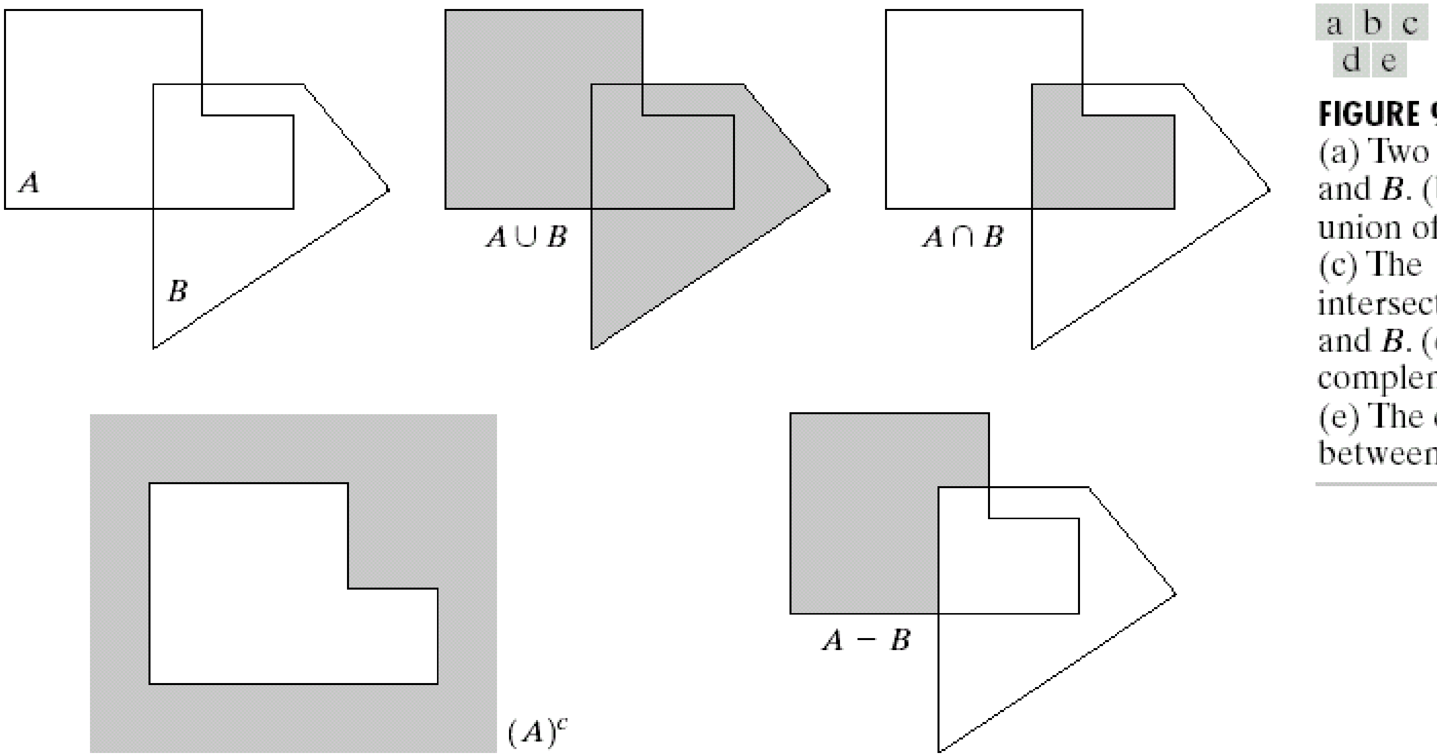


FIGURE 9.1 (a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

Preliminaries

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

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

- Translation: The translation of a set A by point z = (z_1,z_2), denoted $(A)_z$, is defined as

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

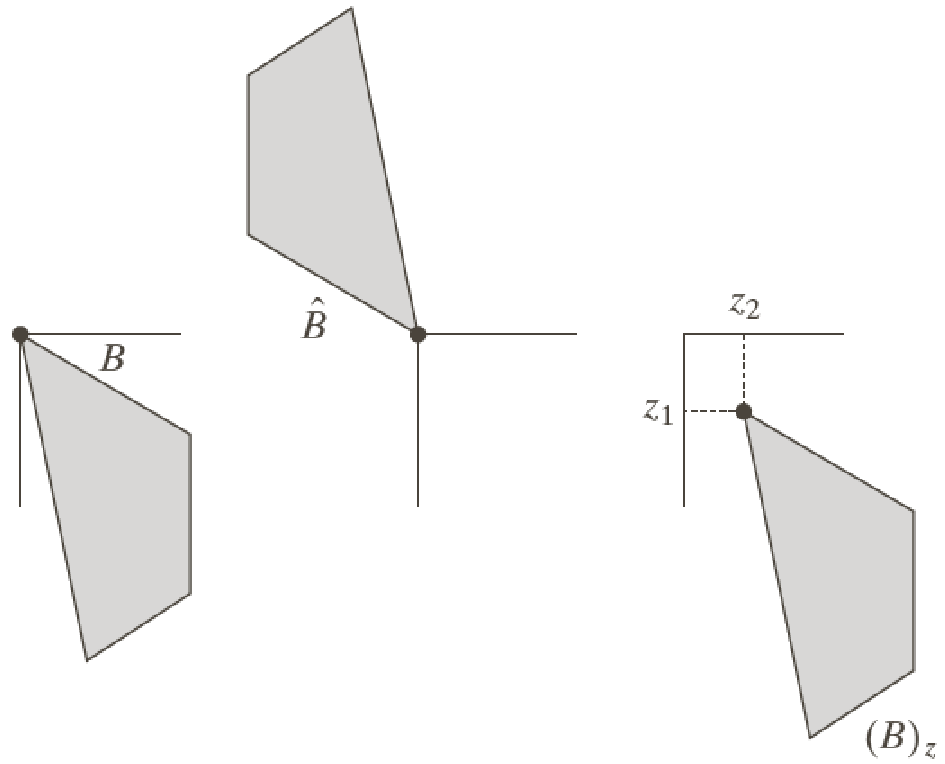


FIGURE 9.1 (a) A set, (b) its reflection, and (c) its translation by z.

Logic 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

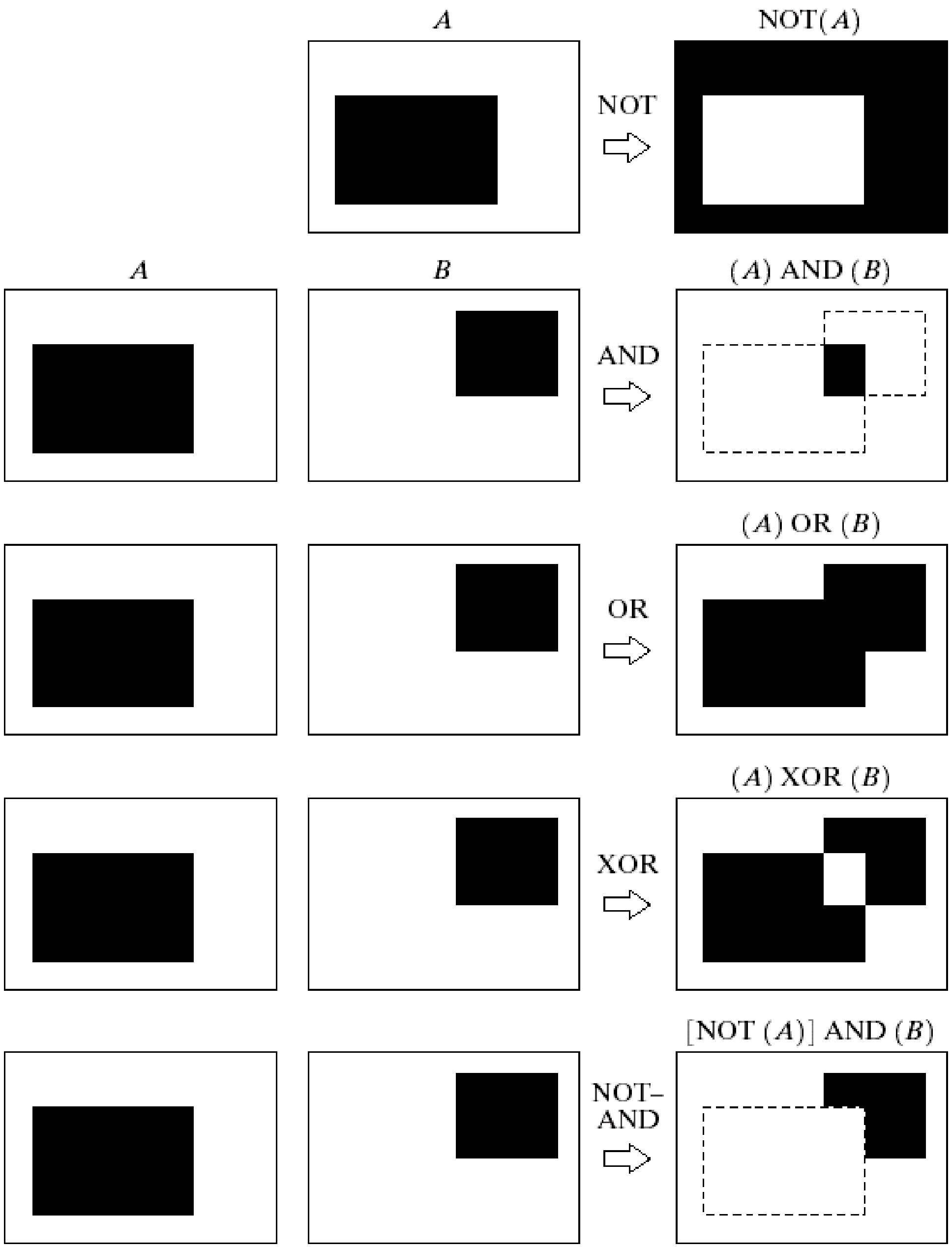


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Structuring Elements

- Small sets or sub-images used to probe an image under study for properties of interest is called SE.
- The **structuring element** is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one.

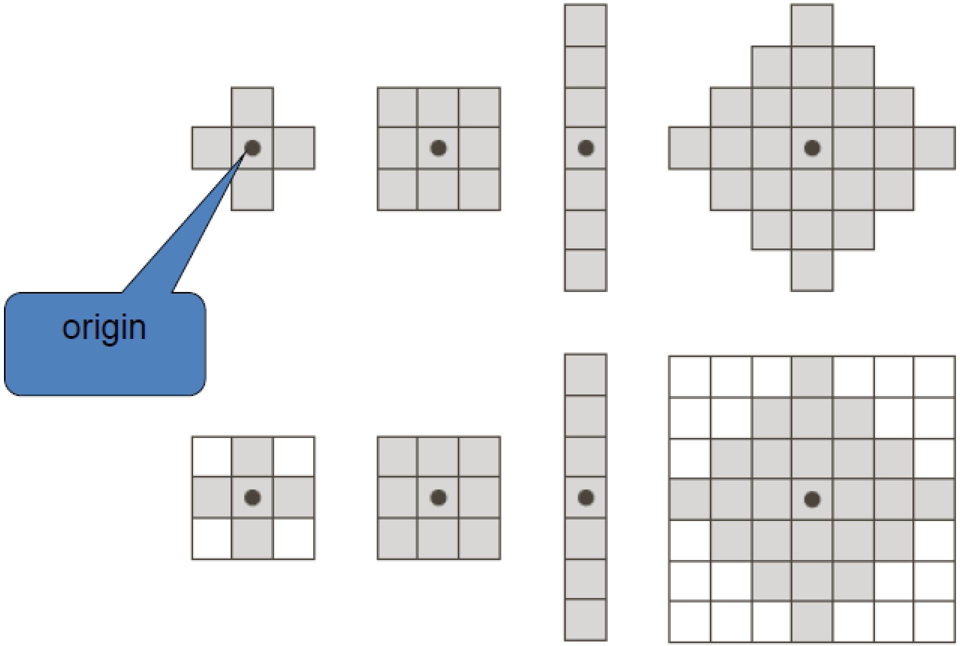
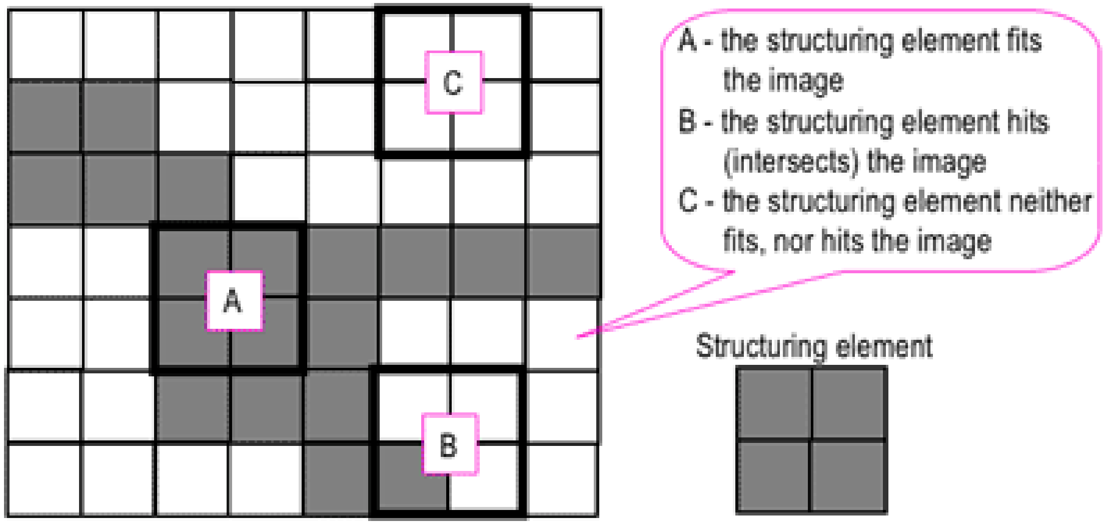


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

- A common practice is to have odd dimensions of the structuring matrix and the origin defined as the centre of the matrix. Structuring elements play in morphological image processing the same role as convolution kernels in linear image filtering.

- When a structuring element is placed in a binary image, each of its pixels is associated with the corresponding pixel of the neighbourhood under the structuring element. 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.

Hits and Fits



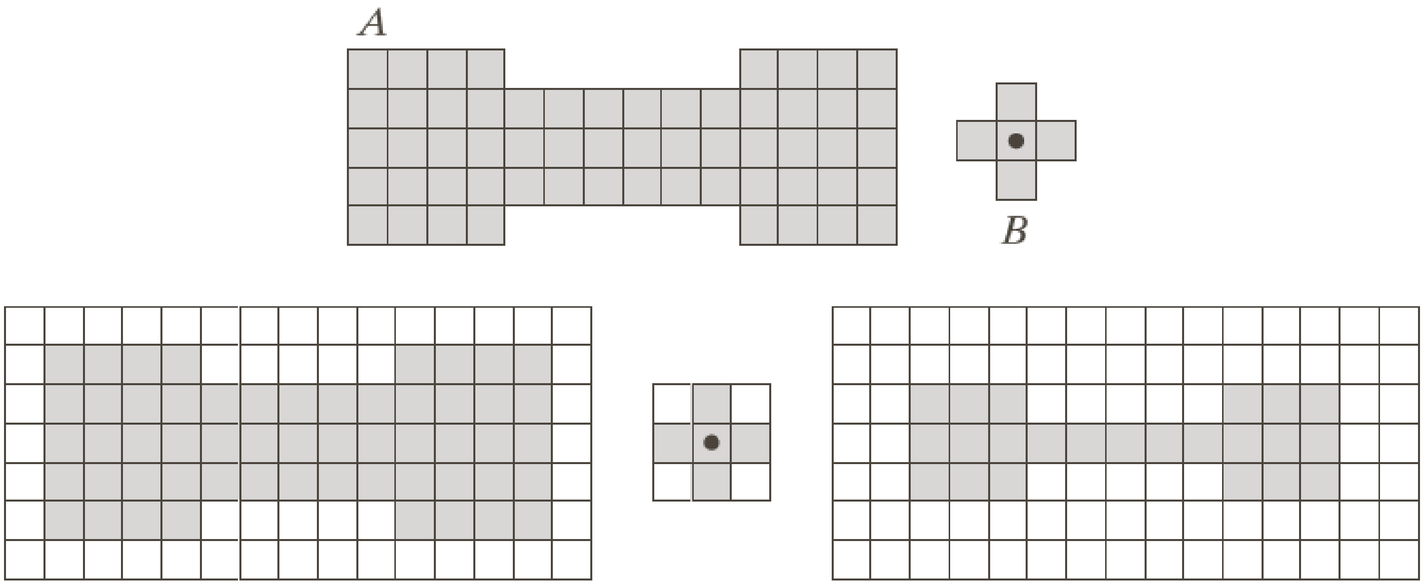
- Hit: Any on pixel in the structuring element covers an on pixel in the image.

- Fit: All on pixel in the structuring element cover on pixels in the image

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)

- SE is of size 3×3 with the origin in the center, so as a on element border that encompasses the entire set is sufficient



a b
c d e

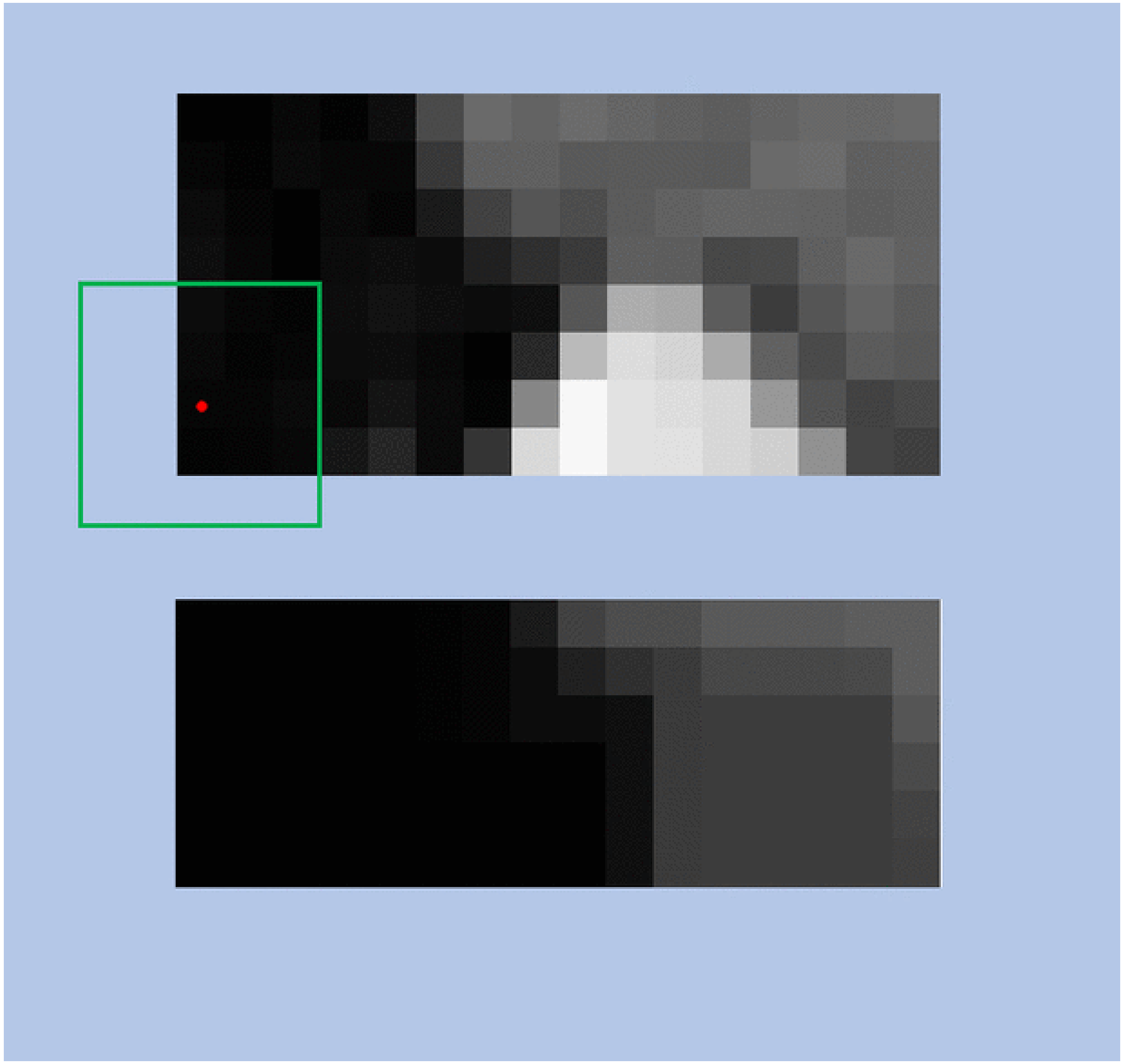
FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

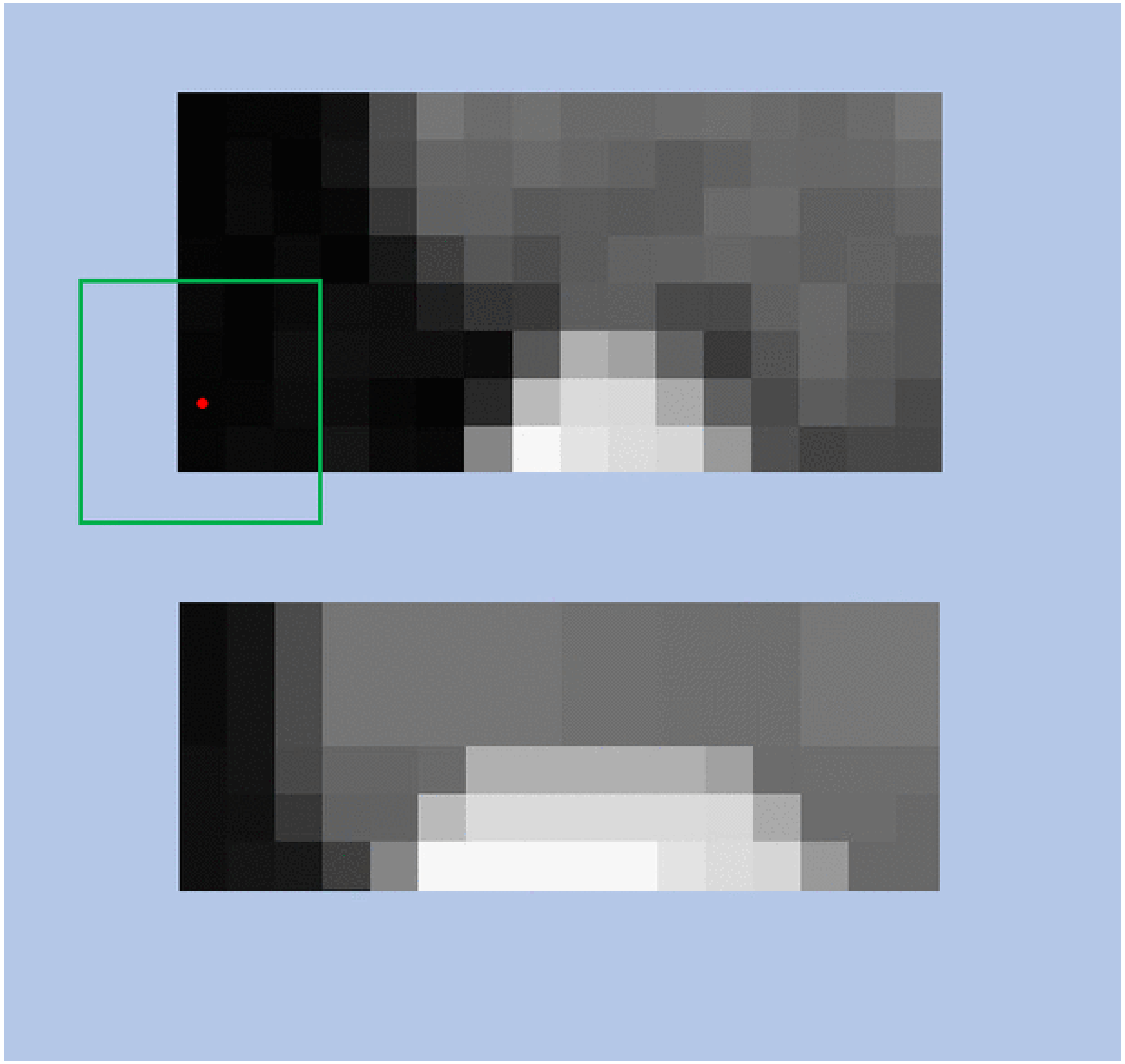
Erosion and Dilation

- **Erosion:**
  - The value of the output pixel is the minimum value of all pixels in the neighbourhood.
  - In a binary image, a pixel is set to 0 if any of the neighbouring pixels have the value 0. Morphological erosion removes islands and small objects so that only substantive objects remain.

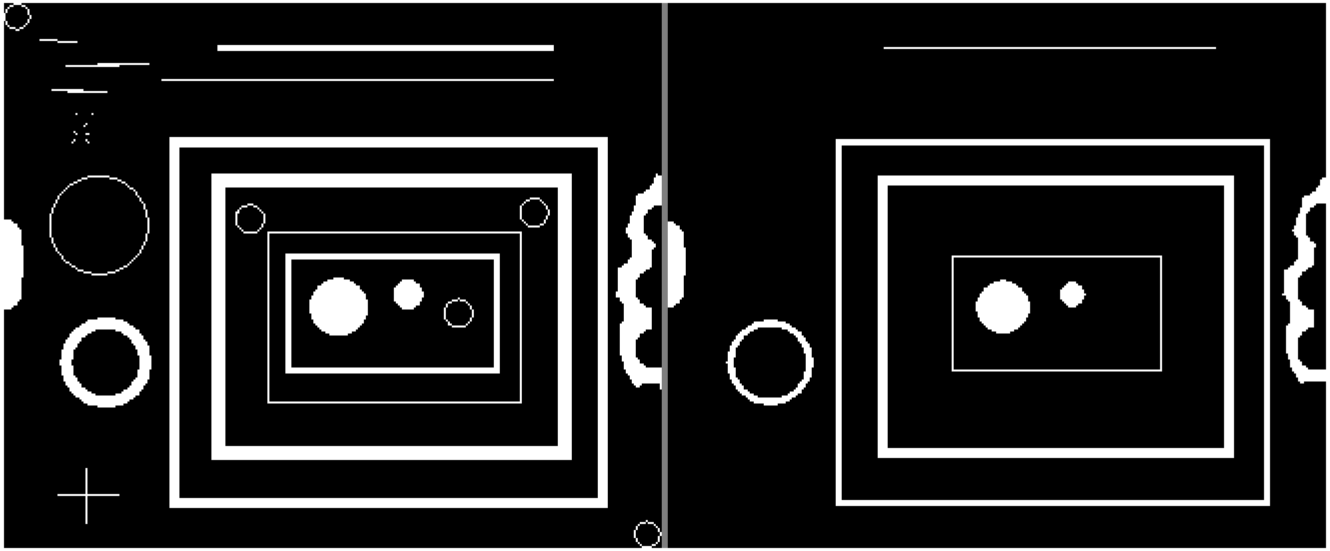
- **Dilation:**
  - The value of the output pixel is the maximum value of all pixels in the neighbourhood.
  - In a binary image, a pixel is set to 1 if any of the neighbouring pixels have the value 1. Morphological dilation makes objects more visible and fills in small holes in objects.

Erosion

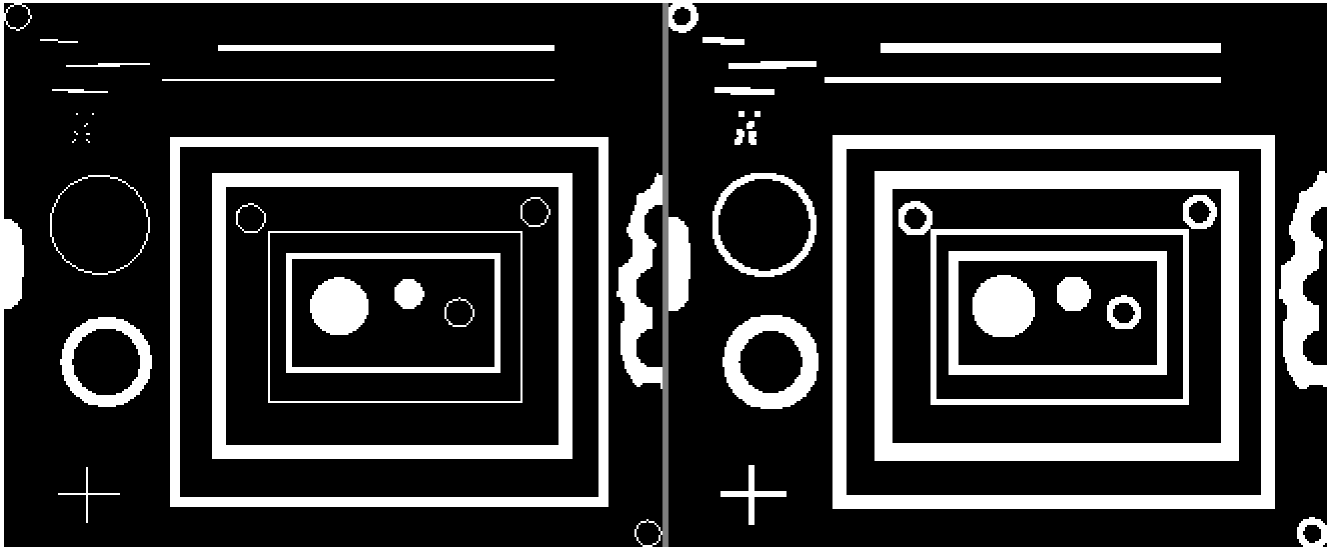




- Erosion



- Dilation



- 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”

- Unlike erosion, dilation “grows” or “thickens” objects in a binary image
- The specific manner and extent of this thickening is controlled by the shape of the structuring element used

Erosion

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

$$A \ominus B = \left\{ z \mid (B)_Z \subseteq A \right\}$$

- To compute the erosion of pixels (x,y) in image f with the structure element S, place the structuring elements such that its origin is at (x,y). Compute

- g(x,y) = {1, if S fits f and 0 otherwise}

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

$$\begin{matrix} 1 & 1 & 1 \\ 1 & \boxed{1} & 1 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 0 \\ 1 & \boxed{1} & 1 \\ 0 & 1 & 0 \end{matrix}$$

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

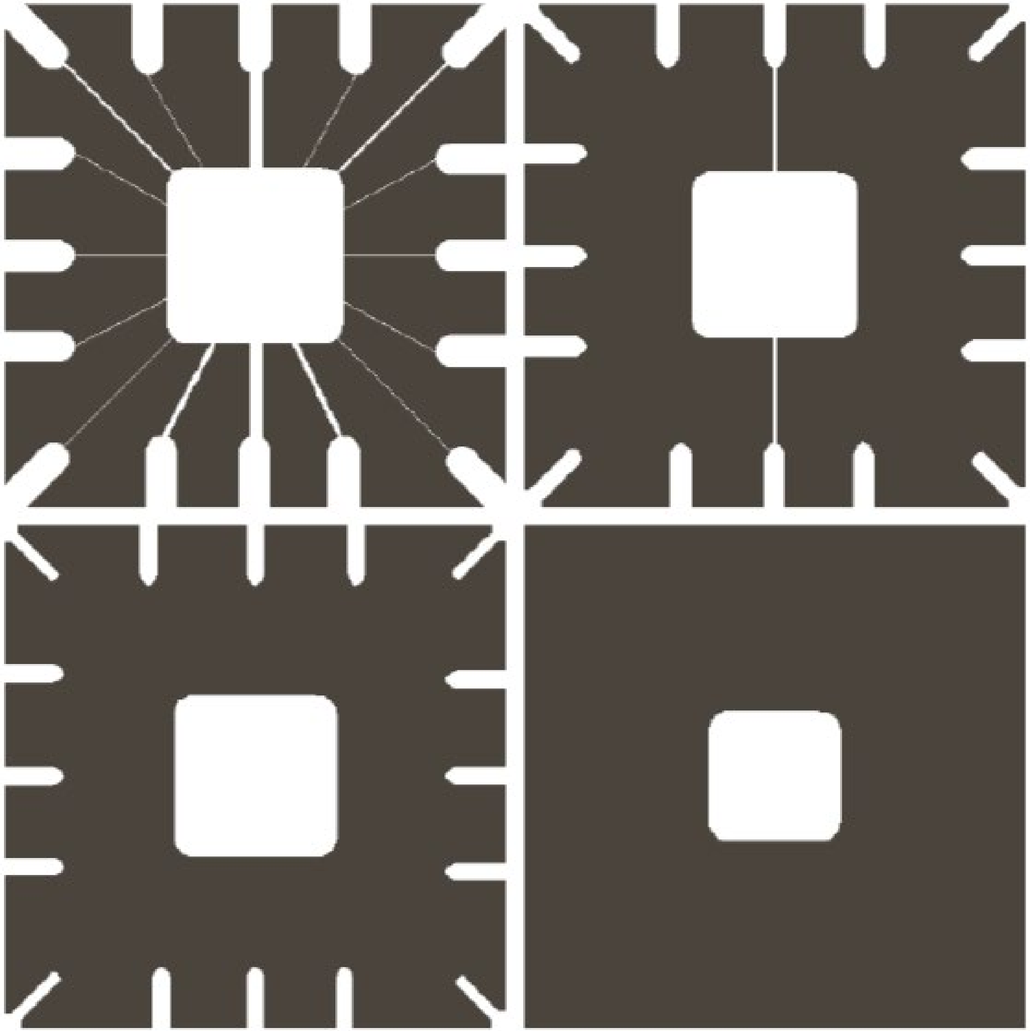


FIGURE 9.5 Using erosion to remove image components. (a) A 486 × 486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11 × 11, 15 × 15, and 45 × 45, respectively. The elements of the SEs were all 1s.

Edge Detection using Erosion

- Erosion removes pixels on the border of an object.
- We can find the border by subtracting and eroded image from the original image

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

gives

difference

Eroded by

0 1 0
1 1 1
0 1 0

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

=>

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

1 1 1
1 1 1
1 1 1

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

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

Example

Dilation

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

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

- To compute the dilation of pixels (x,y) in image f with the structure element S, place the structuring elements such that its origin is at (x,y). Compute

- g(x,y) = {1, if S hits f and 0 otherwise}

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

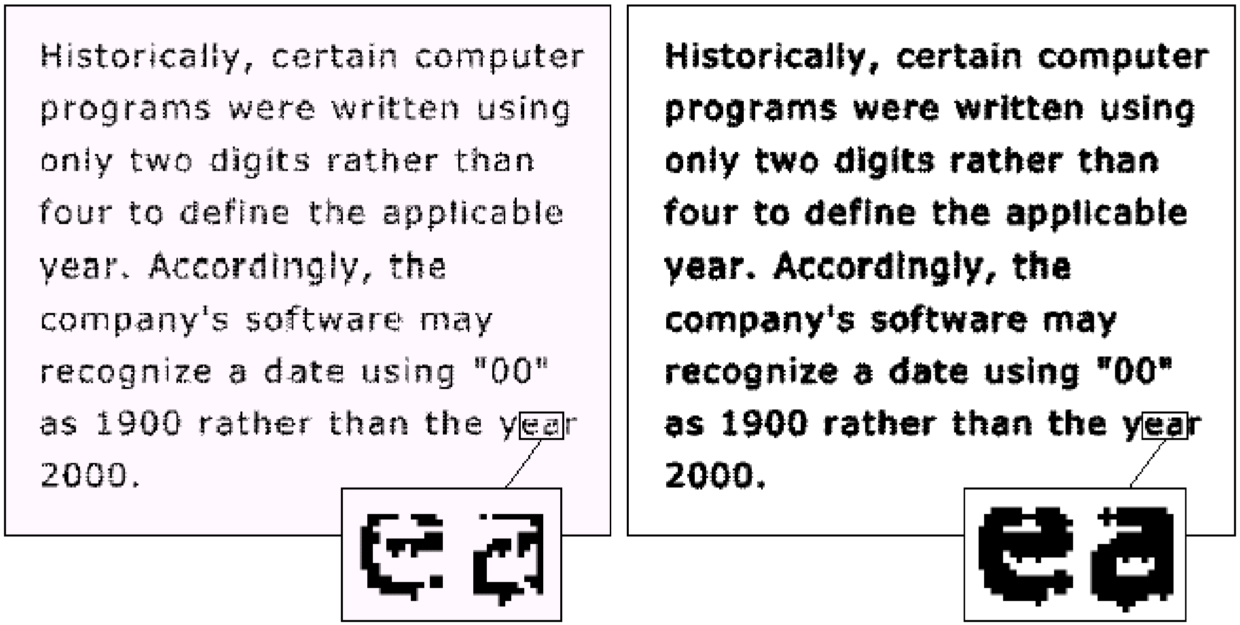
1	1	1
1	$\boxed{1}$	1
1	1	1

0	1	0
1	$\boxed{1}$	1
0	1	0

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

Dilation: Bridging Gaps



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.

Effect of Dilation

- Expand the object
- Both inside and outside borders of the object
- Dilation fills holes in the object
- Dilation smooths out the object contour
- Depends on the structure element
- Bigger structure element gives greater effect

Duality between Erosion and Dilation

- Erosion and dilation are duals of each other with respect to set complementation and reflection

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

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

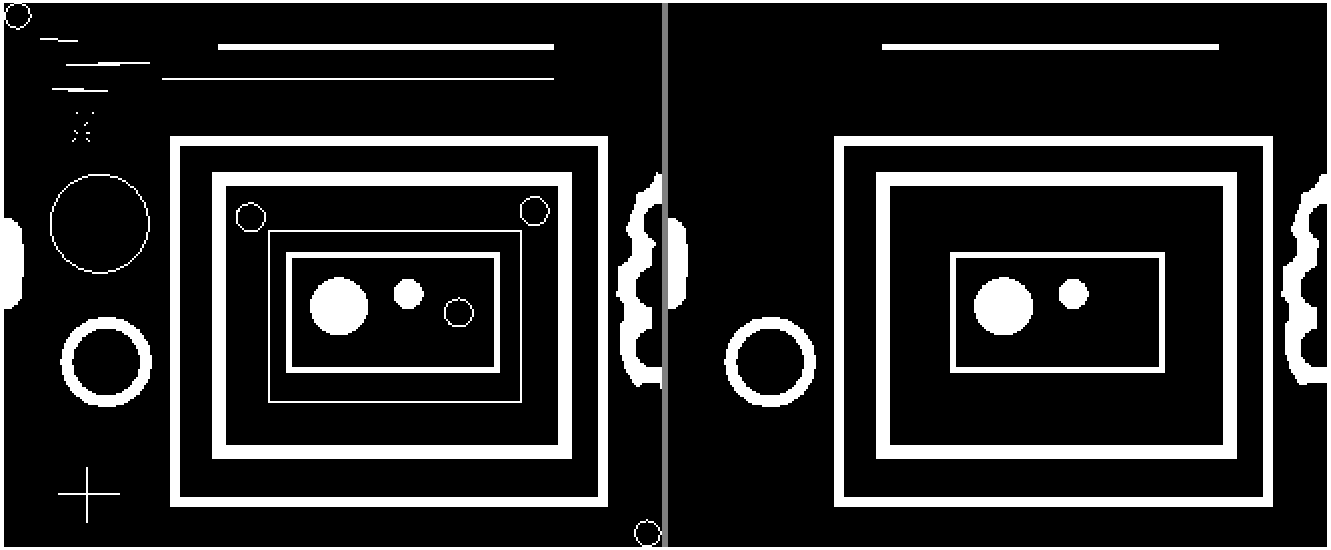
$$\begin{aligned}(A \ominus B)^c &= \{z \mid (B)_z \subseteq A^c = \Phi\}^c \\ &= \{z \mid (B)_z \subseteq A^c \neq \Phi\} \\ &= A^c \oplus \hat{B}\end{aligned}$$

Opening and Closing

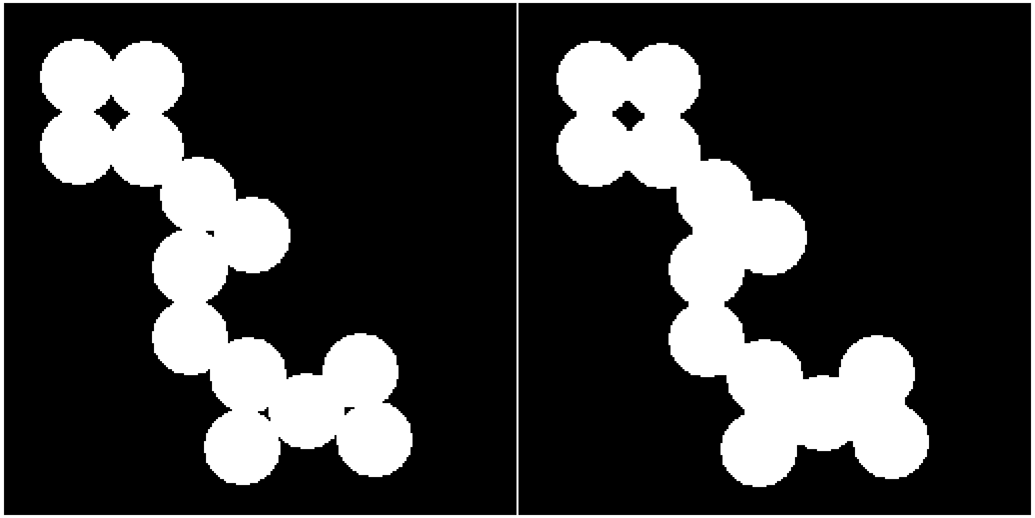
- Opening: First we perform erosion in an image, then we do dilation on the eroded image, using the same structuring element for both operations.

- Closing: First we perform dilation in an image, then we do erosion on the dilated image, using the same structuring element for both operations.

- Opening



- Closing



Opening

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

$$A \circ B = \cup\{(B)_z \mid (B)_z \subseteq A\}$$

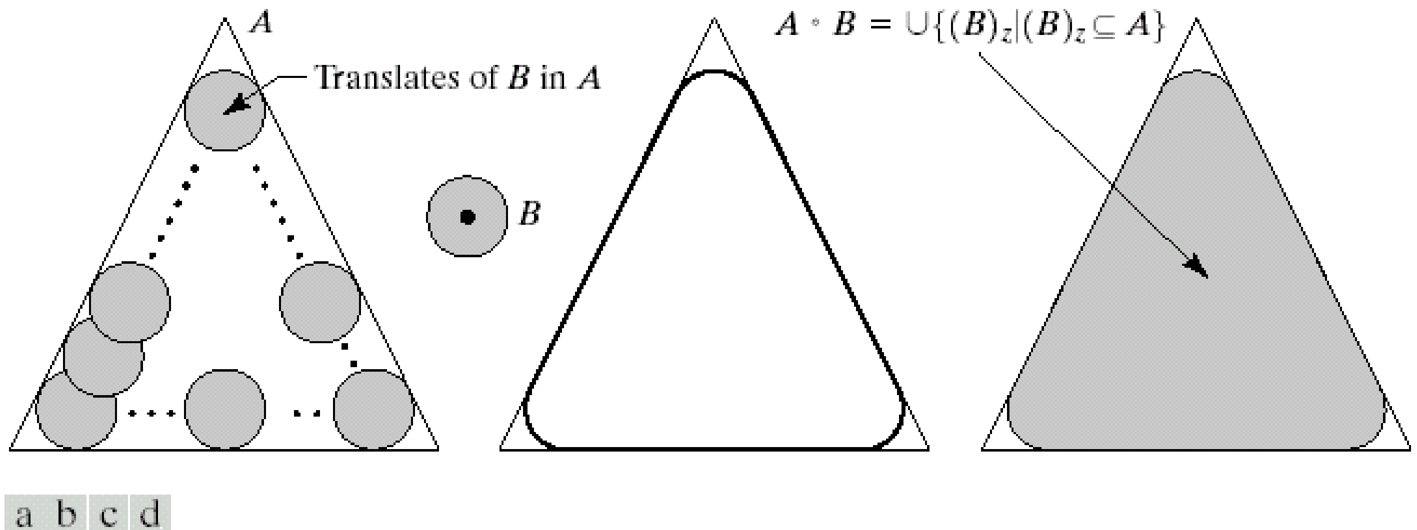
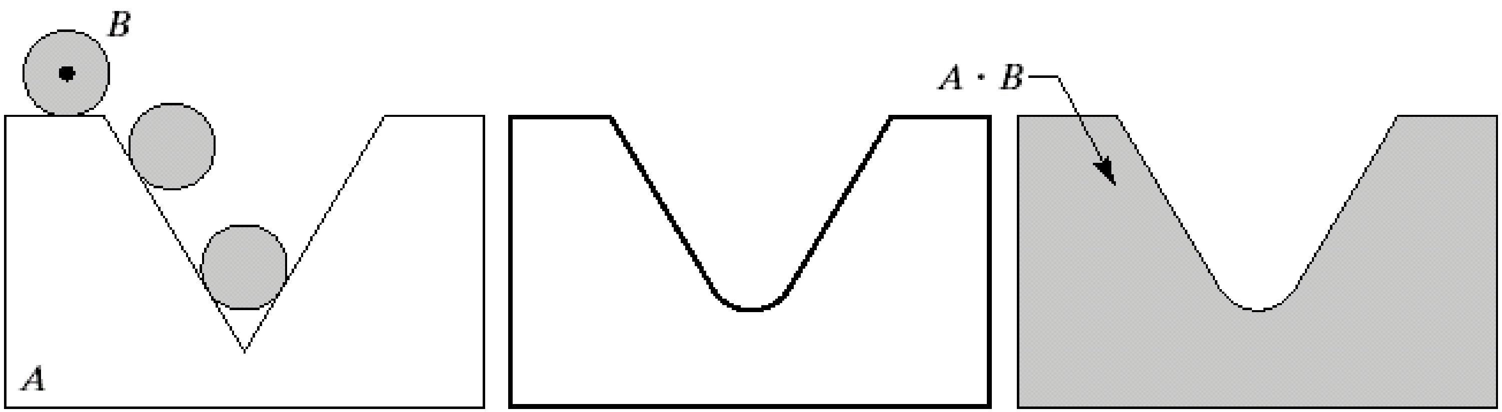


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

Closing

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



a b c

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).

Duality between Opening and Closing

- Opening and Closing are duals of each other with respect to set Complementation and reflections

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

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

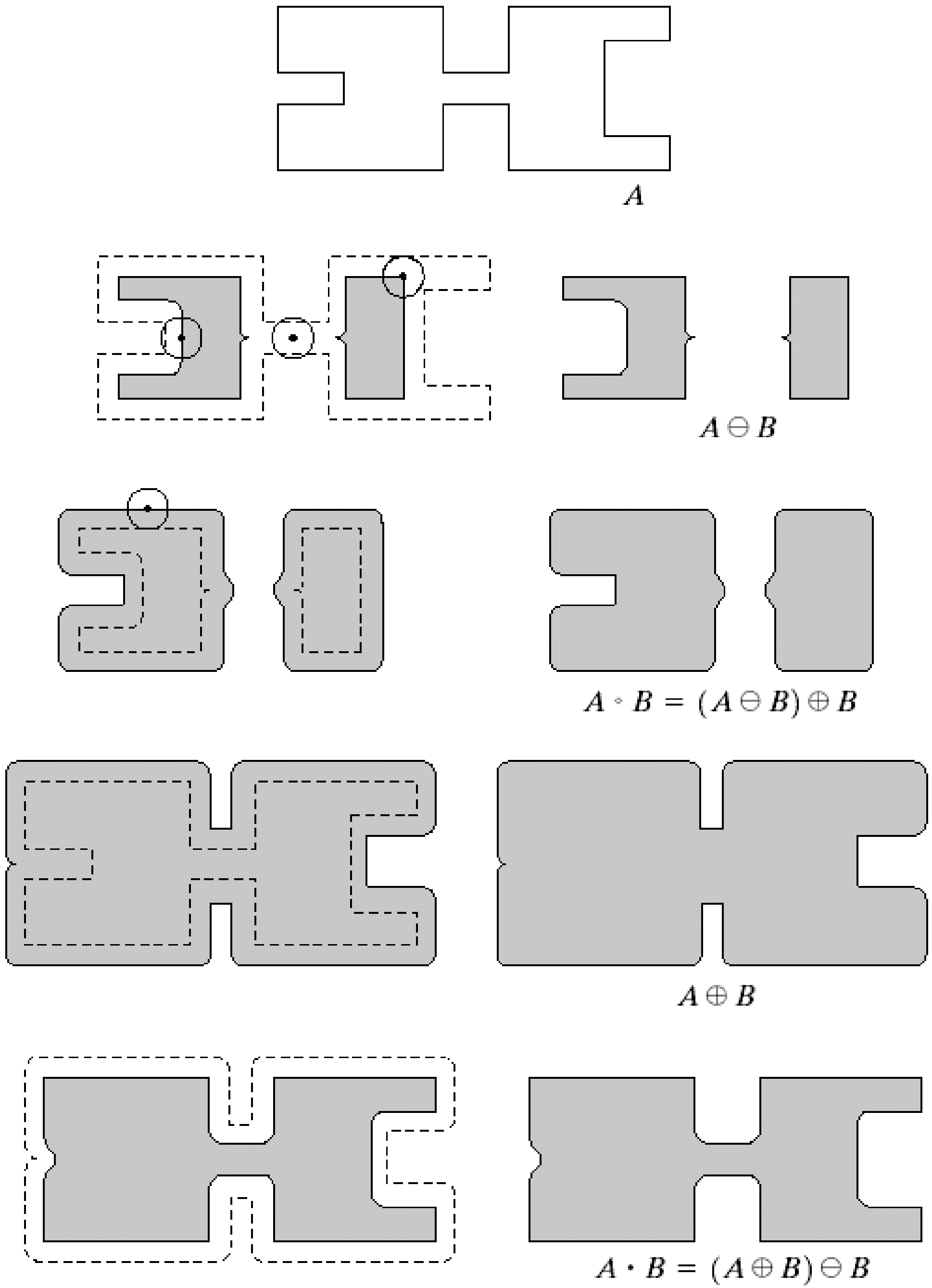
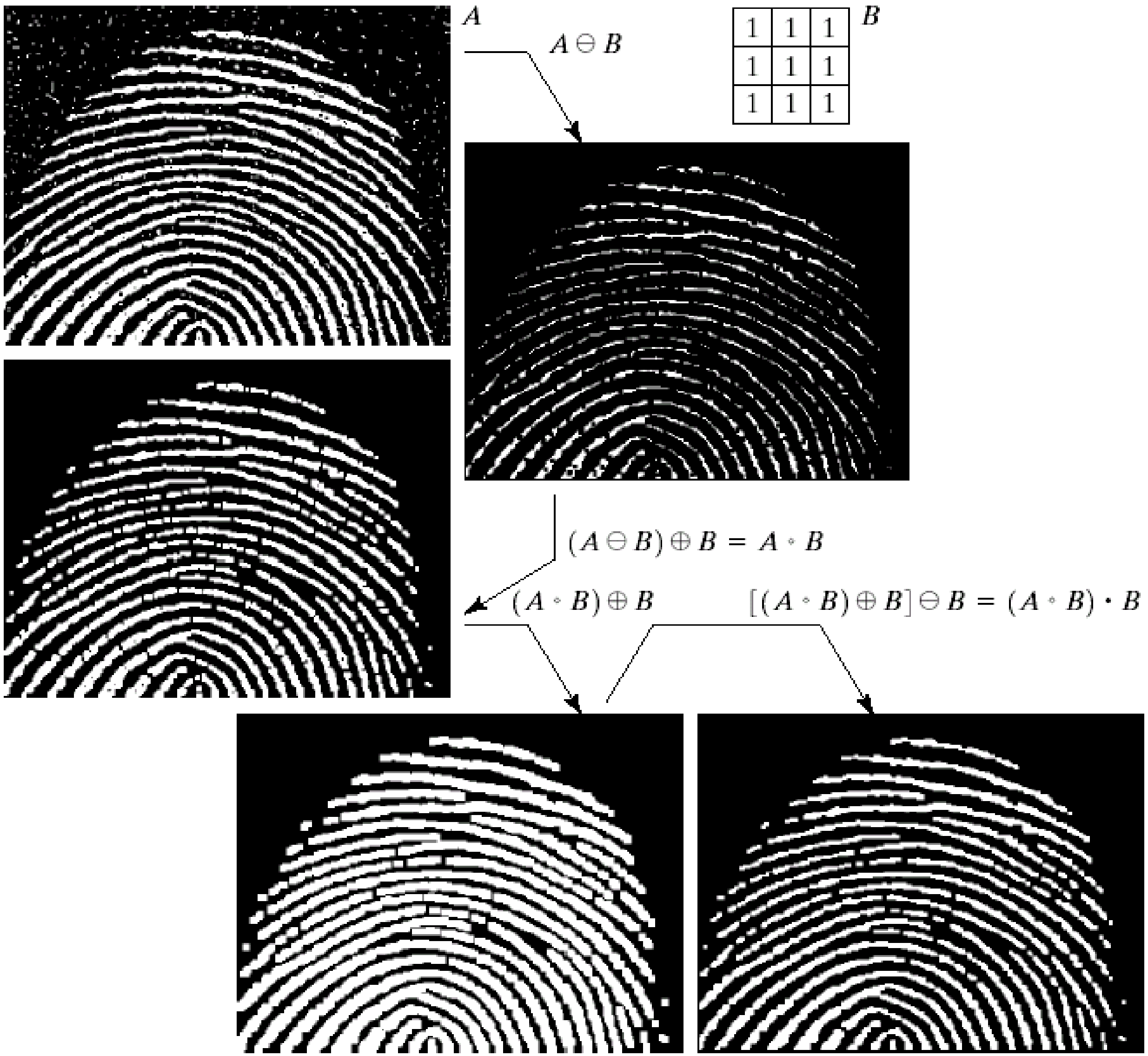


FIGURE 9.10 Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The dark dot is the center of the structuring element.



a b c d 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.)

Hit or Miss Transformation

- Hit or Miss transform is a basic tool for shape detection.

- 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.

- Concept: To detect a shape
  - Hit object
  - Miss background

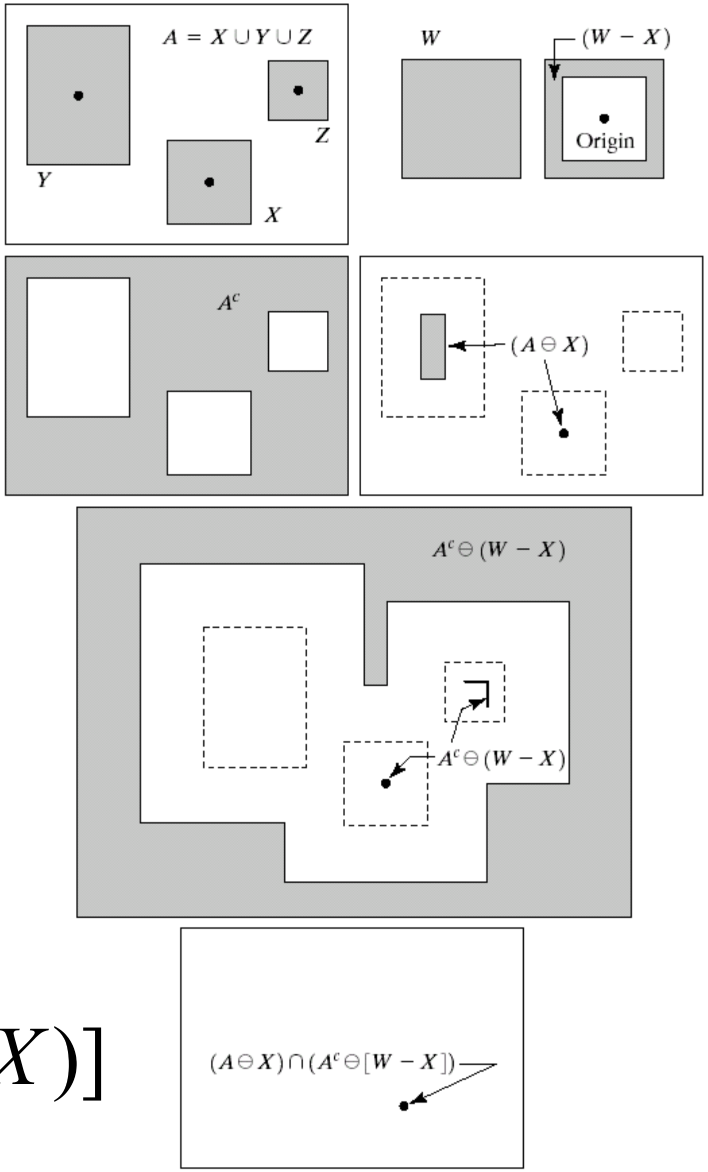
- The structural elements used for Hit-or-miss transforms are an extension to the ones used with dilation, erosion etc.

- The structural elements can contain both foreground and background pixels, rather than just foreground pixels, i.e. both ones and zeros.

- The structuring element is superimposed over each pixel in the input image, and if an exact match is found between the foreground and background pixels in the structuring element and the image, the input pixel lying below the origin of the structuring element is set to the foreground pixel value. If it does not match, the input pixel is replaced by the boundary pixel value.

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.



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

- The Hit or Miss transform is defined as:
  - Let B ={B1, B2}, where B1 is the set formed from elements of B associated with an object and B2 is the set of elements of B associated with the corresponding background, where B1 and B2 are disjoint.

$$A * B = (A \Theta X) \cap [A^c \Theta (W - X)]$$

$$A * B = (A \Theta B_1) \cap [A^c \Theta B_2]$$

B1: Object related

B2: Background related

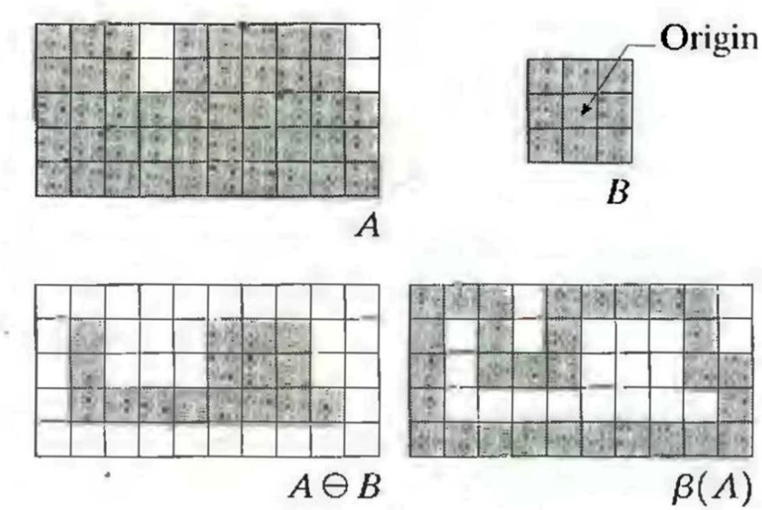
Some Basic Algorithms for Morphology

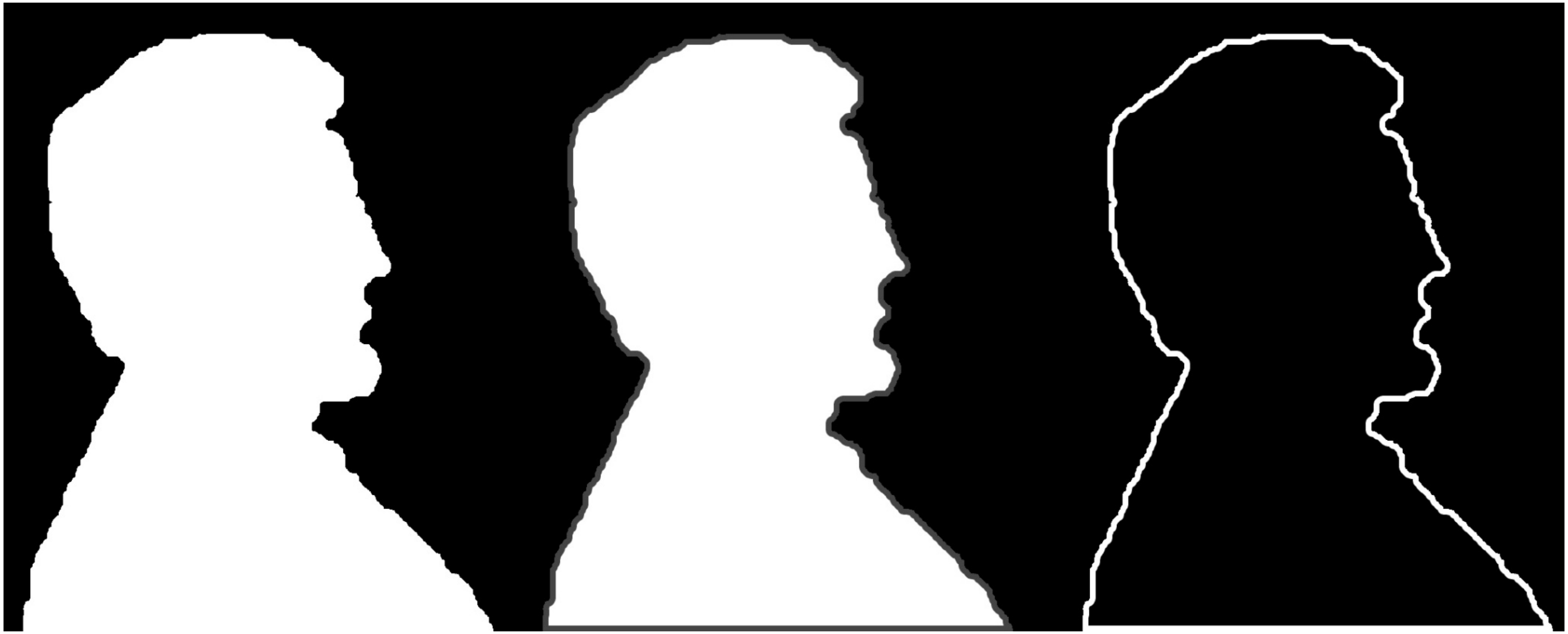
- Boundary Extraction
- Hole Filling
- Extraction of Connected Components
- Thinning and Thickening
- Convex Hull

Boundary Extraction

- First, erode A by B, then make set difference between A and the erosion
- The thickness of the contour depends on the size of constructing object – B

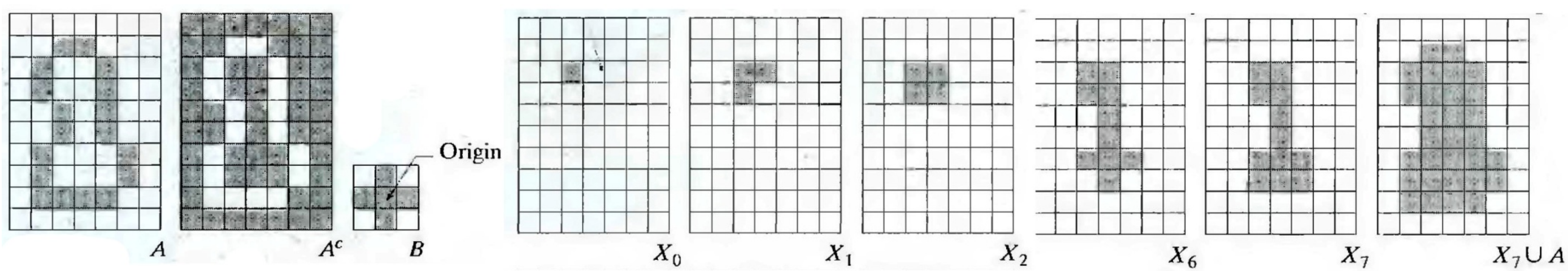
$$\beta(A) = A - (A \ominus B)$$

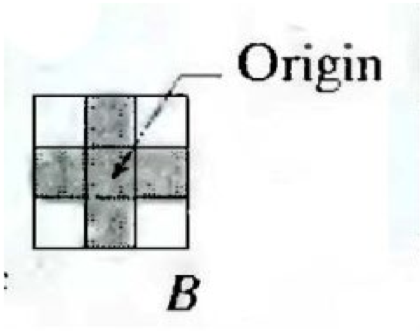


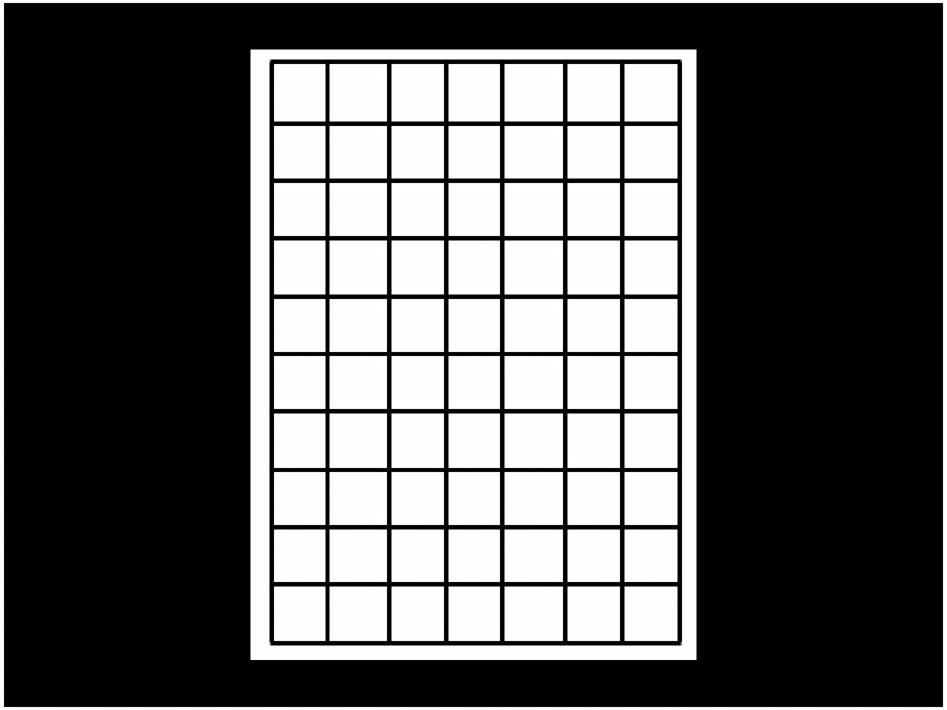


Region Filling

- This algorithm is based on a set of dilations, complementation and intersections
- p is the point inside the boundary, with the value of 1
- $X_{(k)} = (X_{(k-1)}$ xor B) conjunction with complemented A
- The process stops when $X_{(k)} = X_{(k-1)}$
- The result that given by union of A and $X_{(k)}$, is a set contains the filled set and the boundary





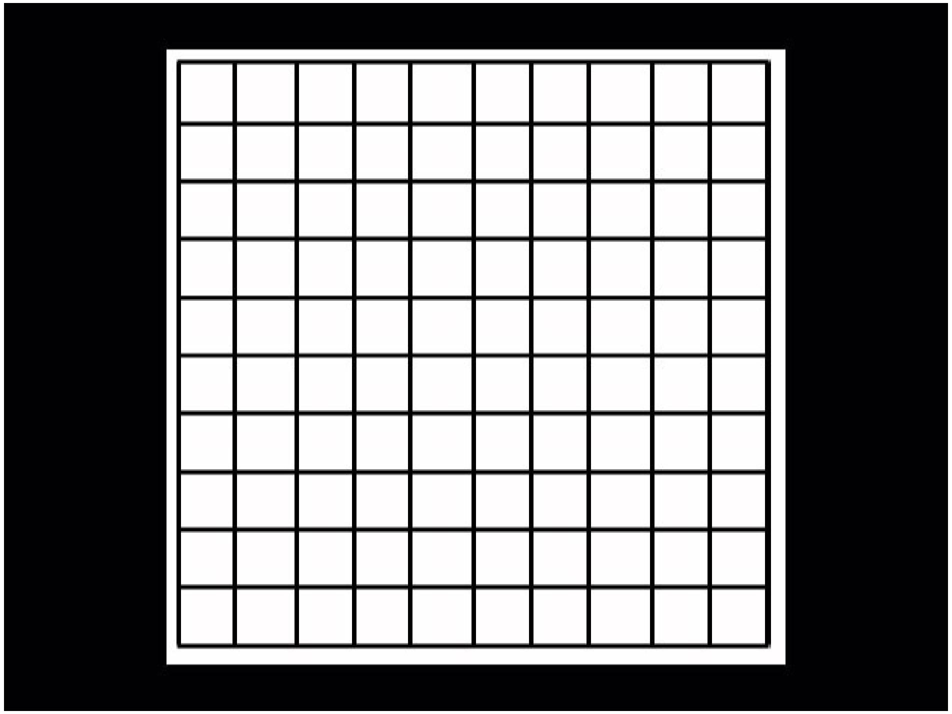


$$X_k = (X_{k-1} \oplus B) \cap A^c$$

Extraction of Connected Components

- This algorithm extracts a component by selecting a point on a binary object A
- Works similar to region filling, but this time we use in the conjunction the object A, instead of it's complement

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

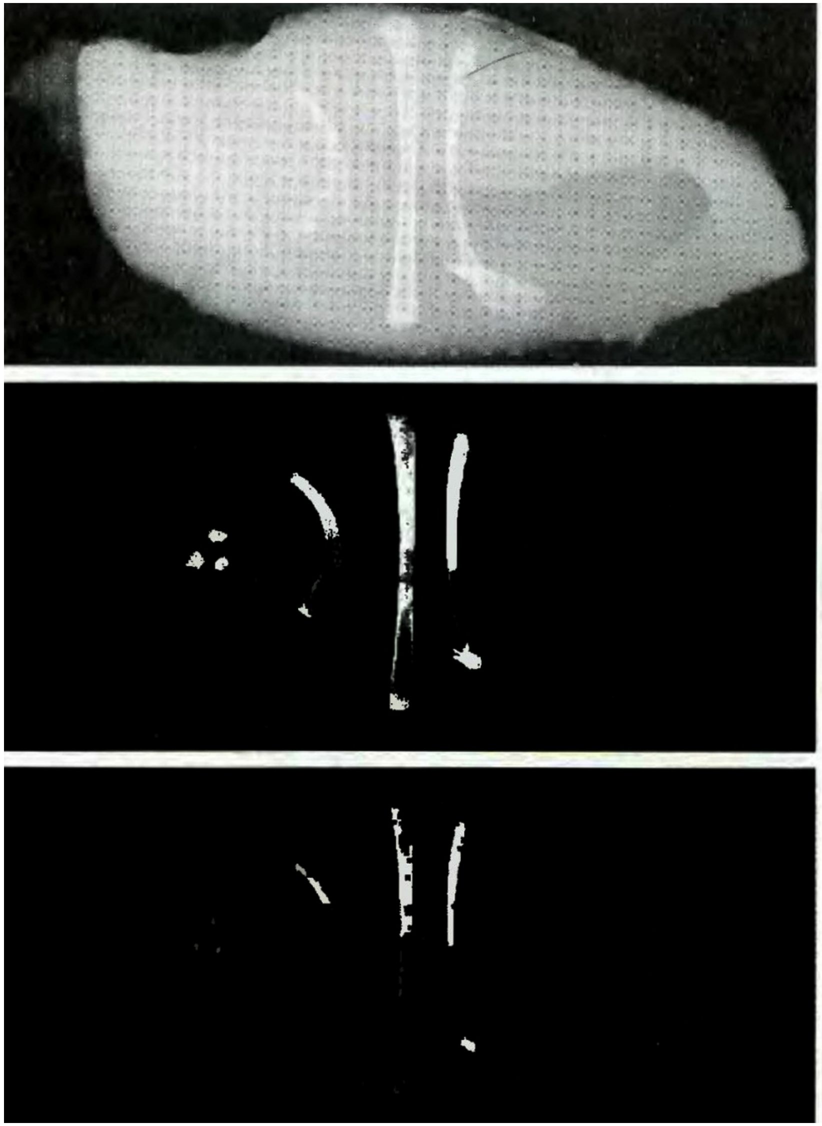


This shows automated inspection of chicken-breast, that contains bone fragment

The original image is thresholded

We can get by using this algorithm the number of pixels in each of the connected components

Now we could know if this food contains big enough bones and prevent hazards



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

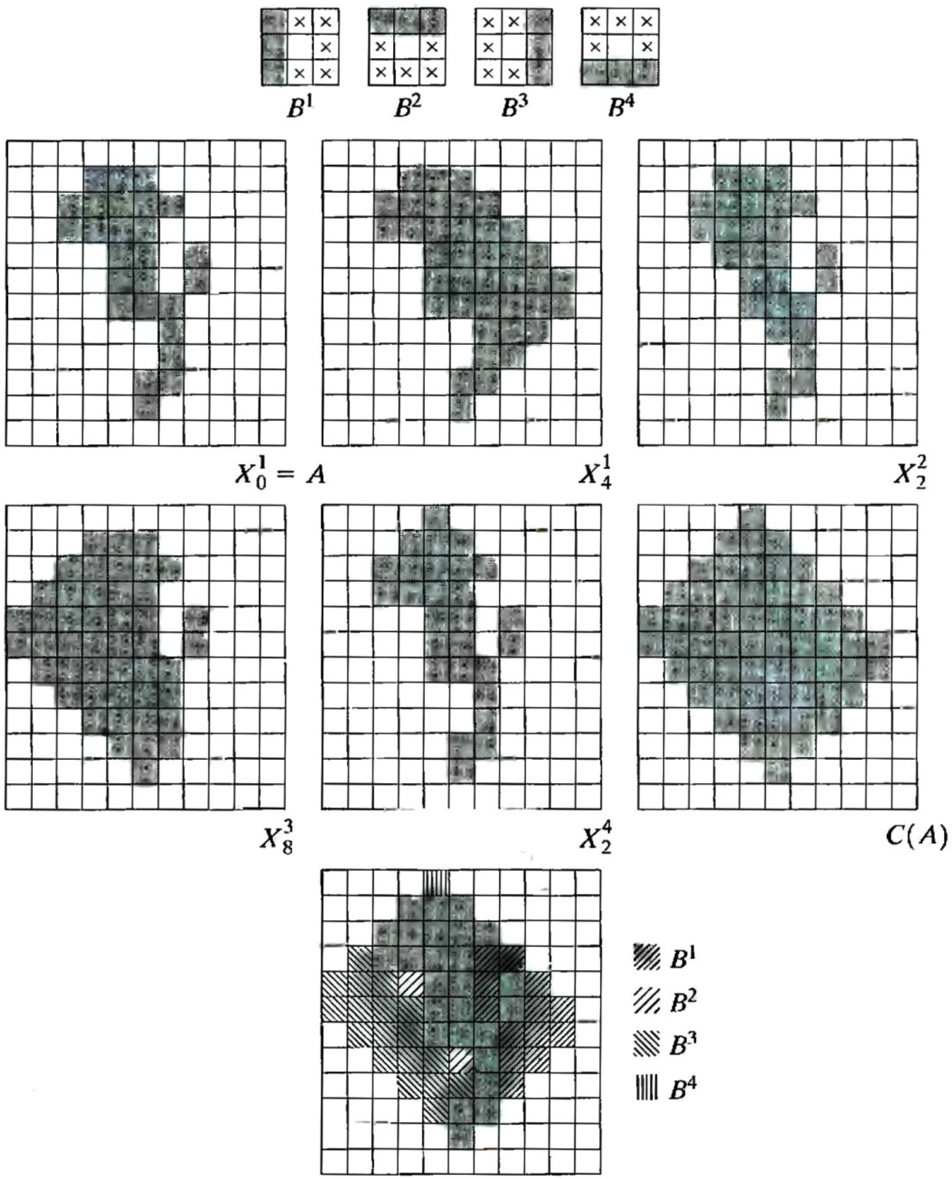
$\bar{A}$

Convex Hull

- A is said to be convex if a straight line segment joining any two points in A lies entirely within A
- The convex hull H of set S is the smallest convex set containing S
- Convex deficiency is the set difference H-S
- Useful for object description
- This algorithm iteratively applying the hit-or-miss transform to A with the first of B element, unions it with A, and repeated with second element of B

$$X_k^i = (X_{k-1} \circledast B^i) \cup A$$

$\bar{A}$



Thinning

- The thinning of a set A by a structuring element B, can be defined by terms of the hit-and-miss transform:

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

- A more useful expression for thinning A symmetrically is based on a sequence of structuring elements:

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

- Where B^i is a rotated version of B^{i-1}. Using this concept we define thinning by a sequence of structuring elements:

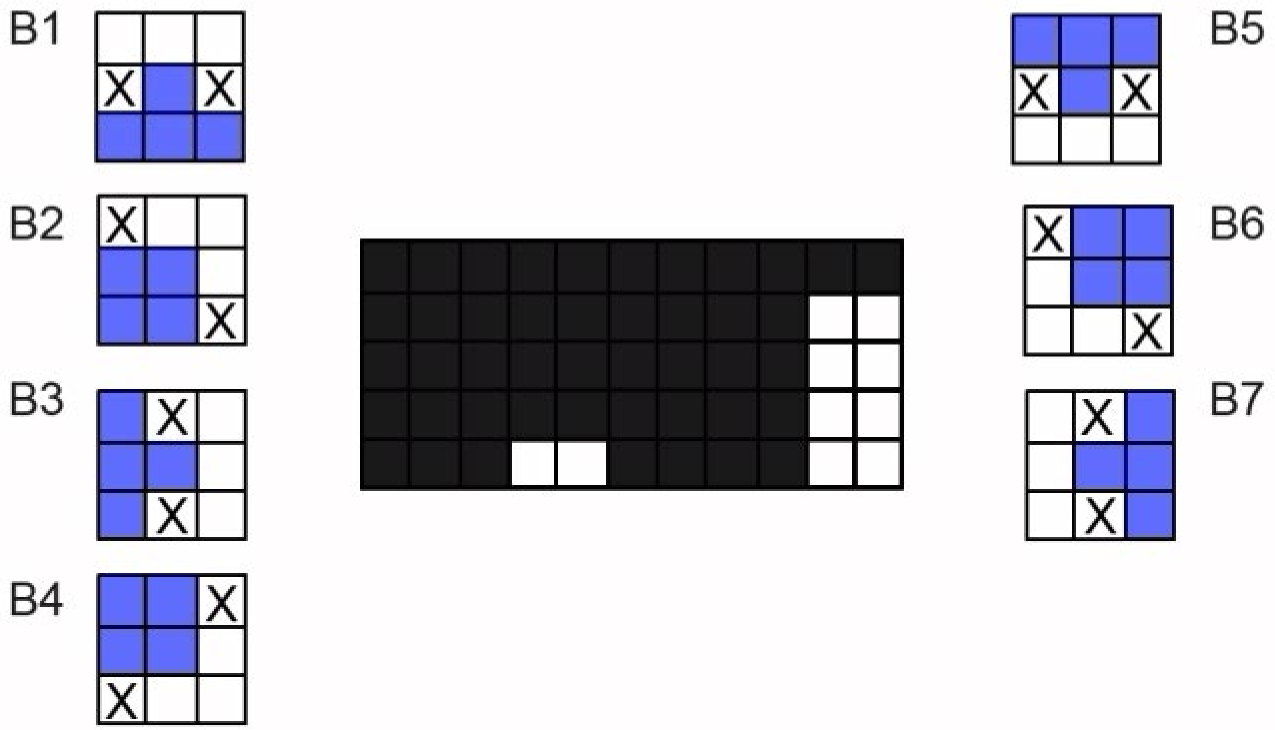
$$A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$$

Thinning cont

- The process is to thin by one pass with B^1 , then thin the result with one pass with B^2, and so on until A is thinned with one pass with B^n.
- The entire process is repeated until no further changes occur.
- Each pass is preformed using the equation:

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

Thinning example



Thickening

- Thickening is a morphological dual of thinning.
- Definition of thickening $A \odot B = A \cup (A \circledast B)$
- As in thinning, thickening can be defined as a sequential operation:

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

- the structuring elements used for thickening have the same form as in thinning, but with all 1's and 0's interchanged.

Thickening - cont

- A separate algorithm for thickening is often used in practice, Instead the usual procedure is to thin the background of the set in question and then complement the result.

- In other words, to thicken a set A, we form $C=A^c$, thin C and than form C^c.

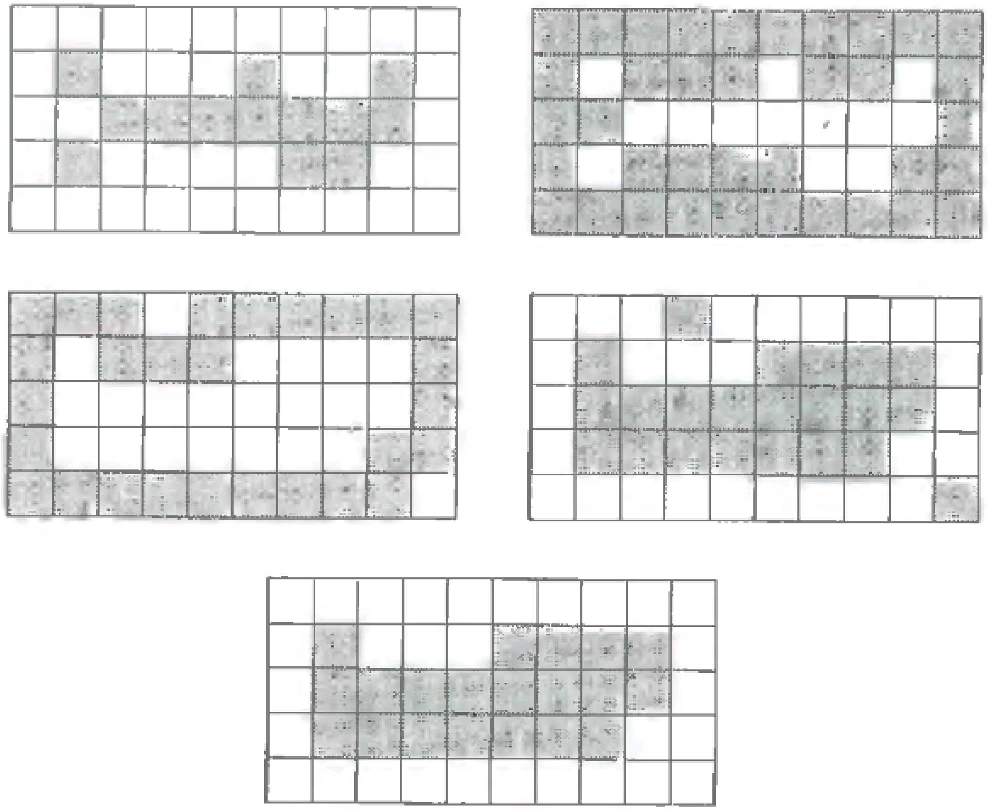
- depending on the nature of A, this procedure may result in some disconnected points. Therefore thickening by this procedure usually require a simple post-processing step to remove disconnected points.

Thickening example preview

- We will notice in the next example 9.22(c) that the thinned background forms a boundary for the thickening process, this feature does not occur in the direct implementation of thickening

- This is one of the reasons for using background thinning to accomplish thickening.

Thickening example



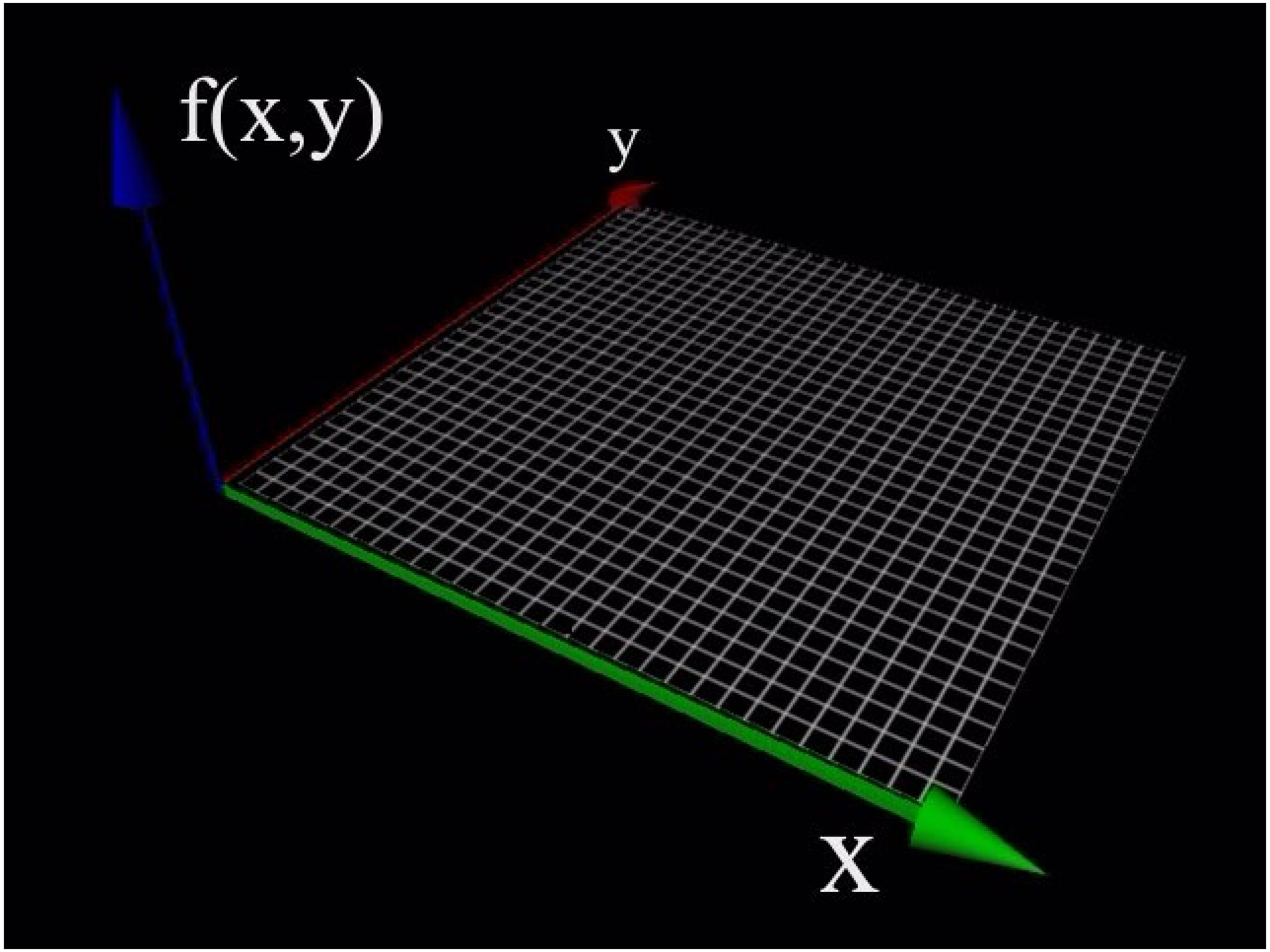
a b
c d
e

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

Gray-Scale Images

- In gray scale images on the contrary to binary images we deal with digital image functions of the form f(x,y) as an input image and b(x,y) as a structuring element.
- (x,y) are integers from Z*Z that represent a coordinates in the image.
- f(x,y) and b(x,y) are functions that assign gray level value to each distinct pair of coordinates.
- For example the domain of gray values can be 0-255, whereas 0 – is black, 255- is white.

Gray-Scale Images



Dilation – Gray-Scale

- Equation for gray-scale dilation is:

$$(f \oplus b)(s,t) = \max\{f(s-x,t-y) + b(x,y) \mid (s-x),(t-y) \in D_f,(x,y) \in D_b\}$$

- D_f and D_b are domains of f and b.
- The condition that (s-x),(t-y) need to be in the domain of f and x,y in the domain of b, is analogous to the condition in the binary definition of dilation, where the two sets need to overlap by at least one element.

Dilation – Gray-Scale (cont)

- We will illustrate the previous equation in terms of 1-D. and we will receive an equation for 1 variable:

  $$(f \oplus b)(s) = \max\{f(s-x) + b(x) \mid (s-x) \in D_f \; and \; x \in D_b\}$$

- The requirements the (s-x) is in the domain of f and x is in the domain of b imply that f and b overlap by at least one element.
- Unlike the binary case, f, rather than the structuring element b is shifted.
- Conceptually f sliding by b is really not different than b sliding by f.

Dilation – Gray-Scale (cont)

- The general effects of performing dilation on a gray scale image is twofold:

1. If all the values of the structuring elements are positive than the output image tends to be brighter than the input.
2. Dark details either are reduced or elimanted, depending on how their values and shape relate to the structuring element used for dilation

Dilation – Gray-Scale example

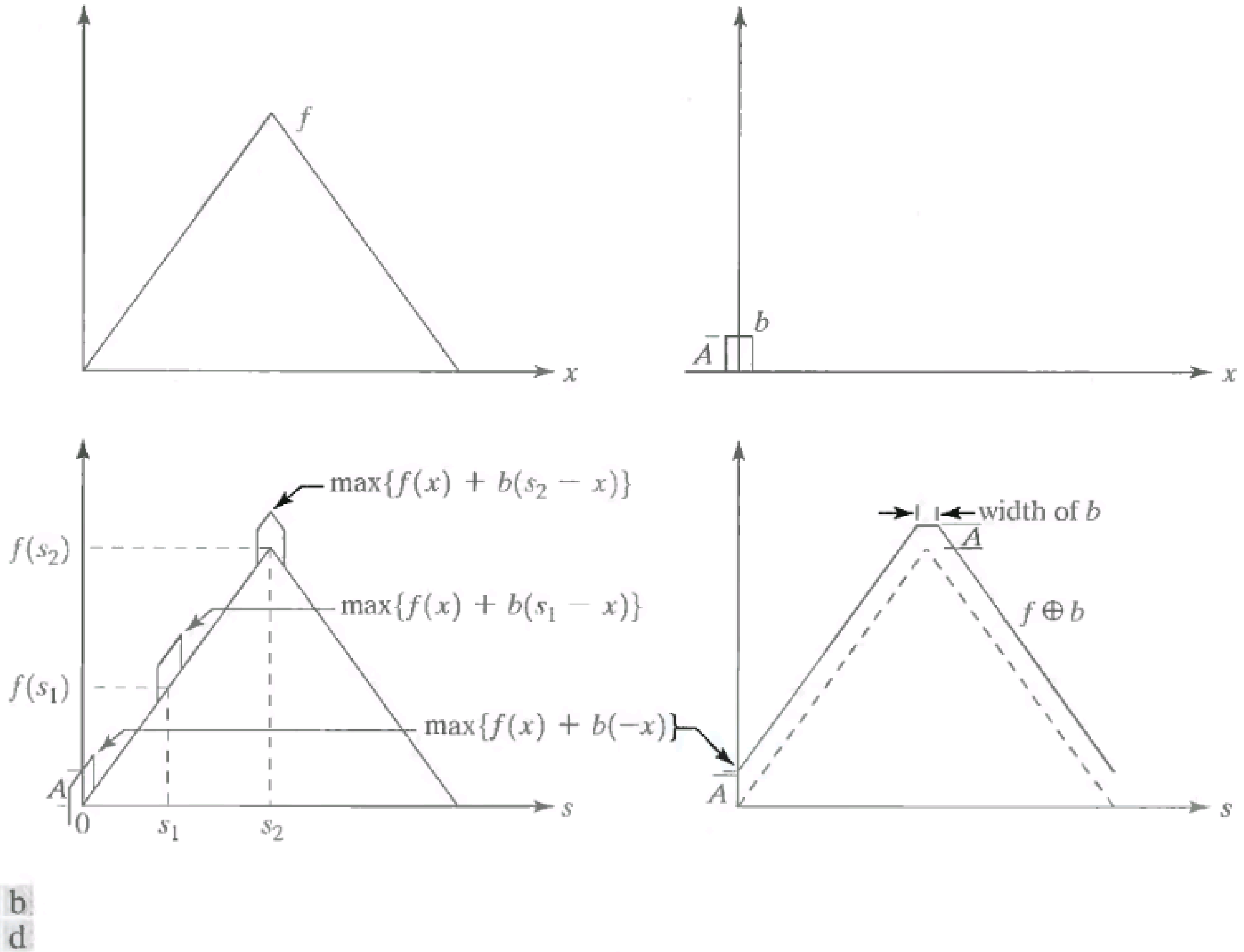


FIGURE 9.27 (a) A simple function. (b) Structuring element of height A. (c) Result of dilation for various positions of sliding b past f. (d) Complete result of dilation (shown solid).

Erosion – Gray-Scale

- Gray-scale erosion is defined as:

$$(f \ominus b)(s,t) = \min\{f(s+x,t+y) - b(x,y) \mid (s+x),(t+y) \in D_f, (x,y) \in D_b\}$$

- The condition that (s+x),(t+y) have to be in the domain of f, and x,y have to be in the domain of b, is completely analogous to the condition in the binary definition of erosion, where the structuring element has to be completely combined by the set being eroded.
- The same as in erosion we illustrate with 1-D function

$$(f \ominus b)(s) = \min\{f(s+x) - b(x) \mid (s+x) \in D_f \; and \; x \in D_b\}$$

Erosion– Gray-Scale example 1

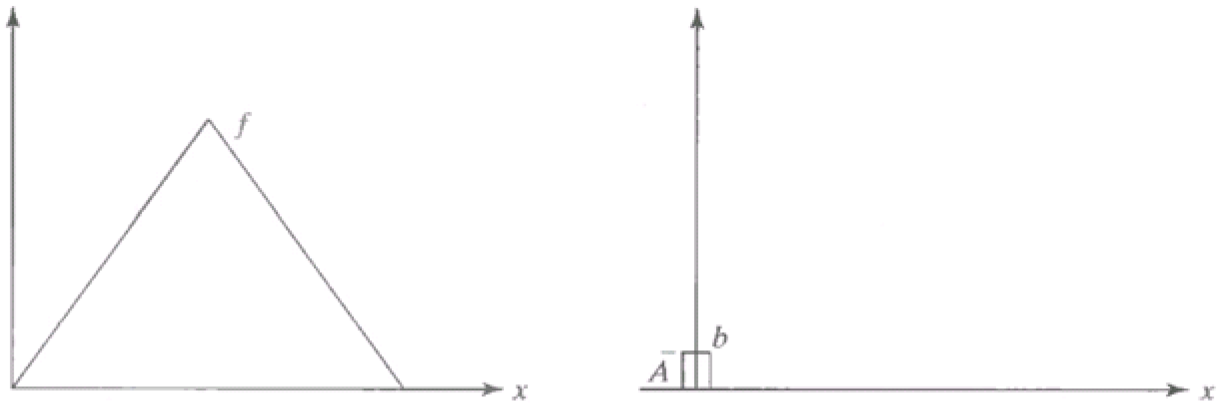
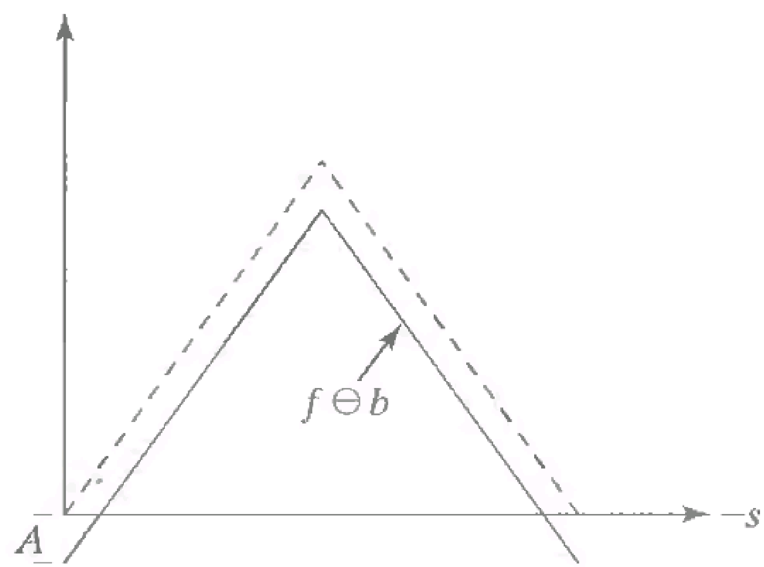


FIGURE 9.28
Erosion of the function shown in Fig. 9.27(a) by the structuring element shown in Fig. 9.27(b).

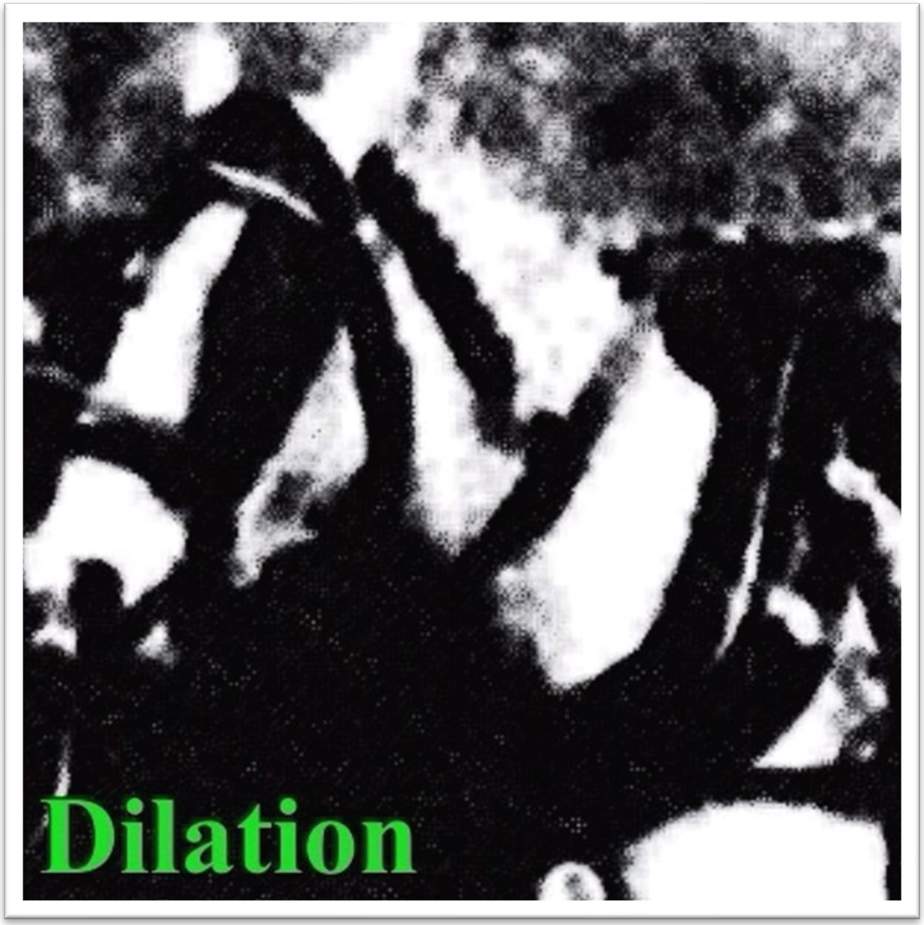


Erosion– Gray-Scale (cont)

- General effect of performing an erosion in grayscale images:
  1. If all elements of the structuring element are positive, the output image tends to be darker than the input image.
  2. The effect of bright details in the input image that are smaller in area than the structuring element is reduced, with the degree of reduction being determined by the grayscale values surrounding by the bright detail and by shape and amplitude values of the structuring element itself.
- Similar to binary image grayscale erosion and dilation are duals with respect to function complementation and reflection.

Dilation & Erosion– Gray-Scale

Over Applying the Filter



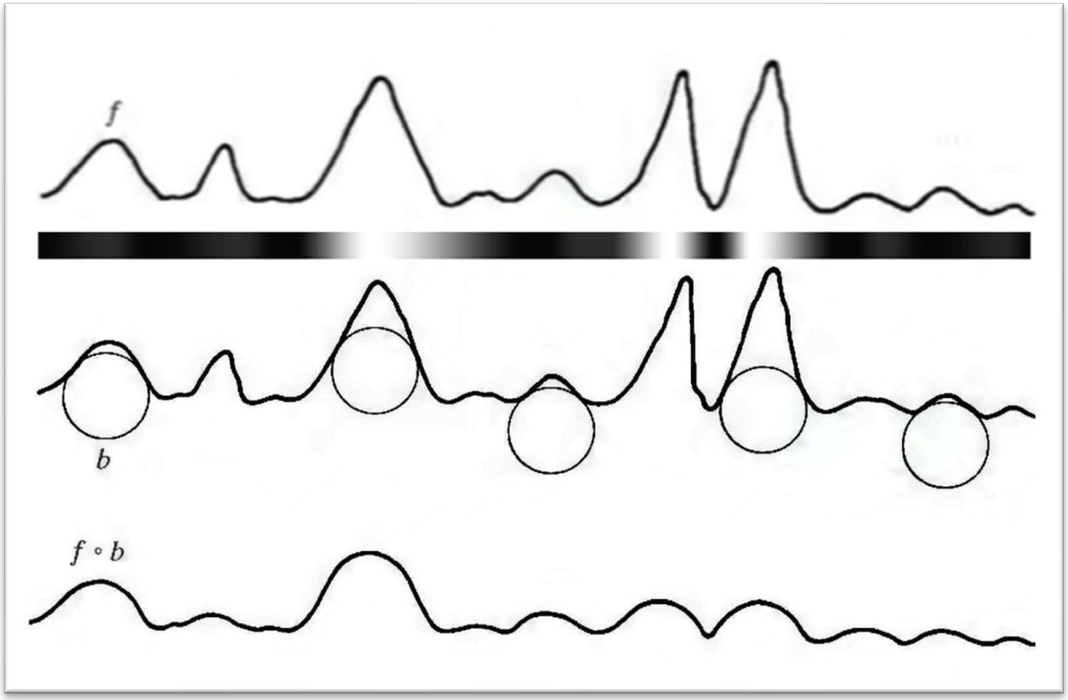
Filter Demonstration

Opening And Closing

- Similar to the binary algorithms
- Opening – $f \circ b = (f \ominus b) \oplus b.$
- Closing – $f \bullet b = (f \oplus b) \ominus b.$
- In the opening of a gray-scale image, we remove small light details, while relatively undisturbed overall gray levels and larger bright features
- In the closing of a gray-scale image, we remove small dark details, while relatively undisturbed overall gray levels and larger dark features

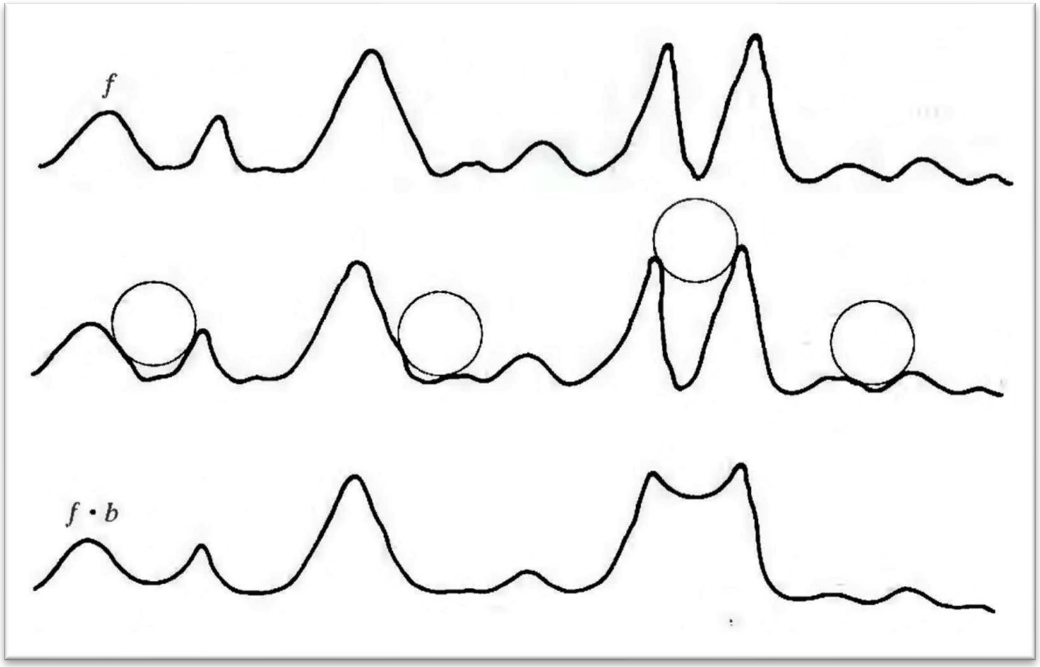
Opening And Closing

- Opening a G-S picture is describable as pushing object B under the scan-line graph, while traversing the graph according the curvature of B

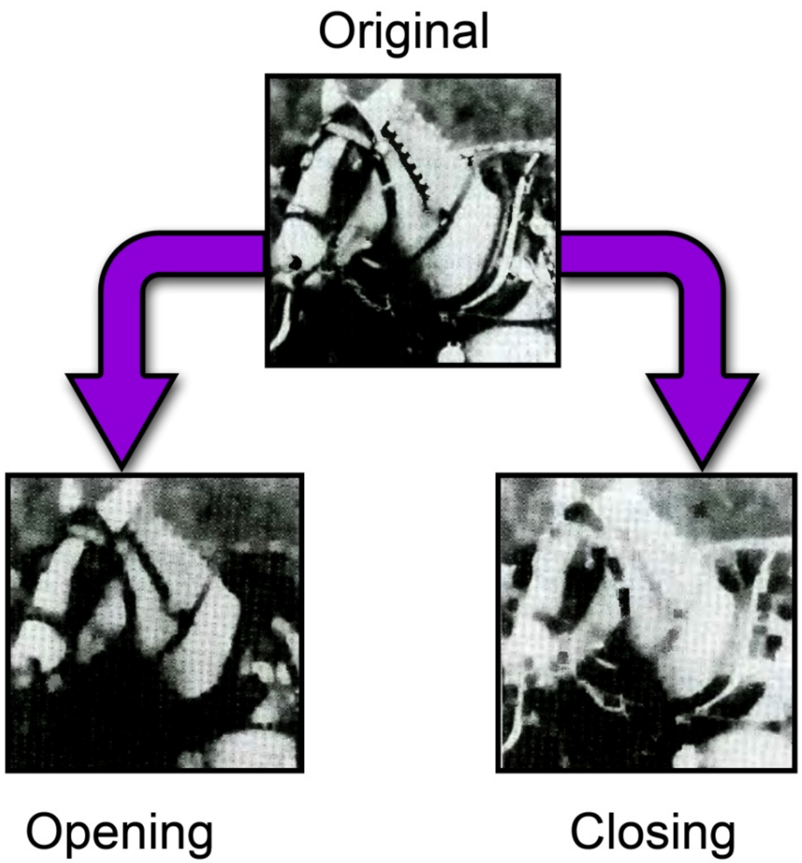


Opening And Closing

- Closing a G-S picture is describable as pushing object B on top of the scan-line graph, while traversing the graph according the curvature of B
- The peaks are usually remains in their original form



Opening And Closing



A	64	23	33	35	32	24
	34	255	24	0	26	23
	23	21	32	31	28	26
	15	20	100	90	43	12
	24	88	70	23	50	66
	88	45	29	51	67	39

B	1	1	1
	1	1	1
	1	1	1

Erosion

23	23	0	0	0	23
21	21	0	0	0	23
15	15	0	0	0	12
15	15	20	23	12	12
15	15	20	23	12	12
24	24	23	23	23	39

Dilation

255	255	255	35	35	32
255	255	255	35	35	32
255	255	255	100	90	43
88	100	100	100	90	66
88	100	100	100	90	67
88	88	88	70	67	67

Opening

23	23	23	0	23	23
23	23	23	0	23	23
21	21	23	23	23	23
15	20	23	23	23	12
24	24	24	23	39	39
24	24	24	23	39	39

Closing

64	35	35	35	32	24
34	255	35	35	32	24
24	88	35	35	32	26
24	88	100	90	43	26
24	88	70	67	66	66
88	51	51	51	67	39