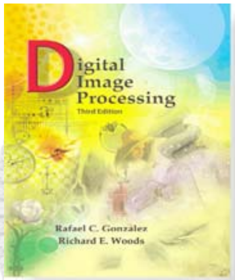
Lecture #17

- Morphology & set operations on images
- Structuring elements
- Erosion and dilation
- Opening and closing
- Morphological image processing, boundary extraction, region filling
- Connectivity: convex hull, thinning

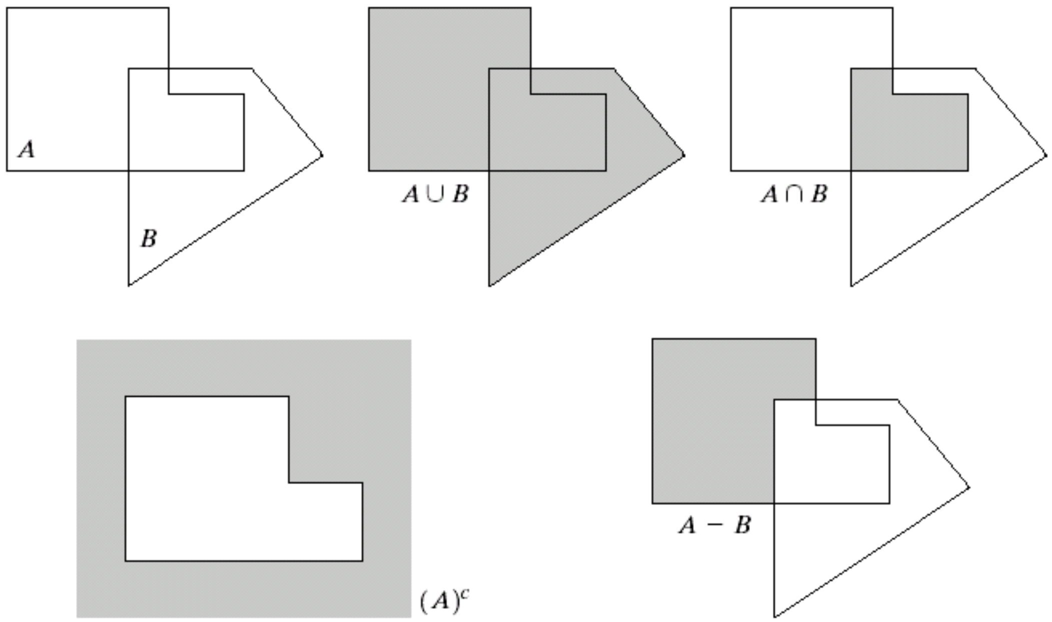
What is Mathematical Morphology?

It is:

- nonlinear,
- built on Minkowski set theory,
- part of the theory of finite lattices,
- for image analysis based on shape,
- extremely useful, yet not often used.



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*.

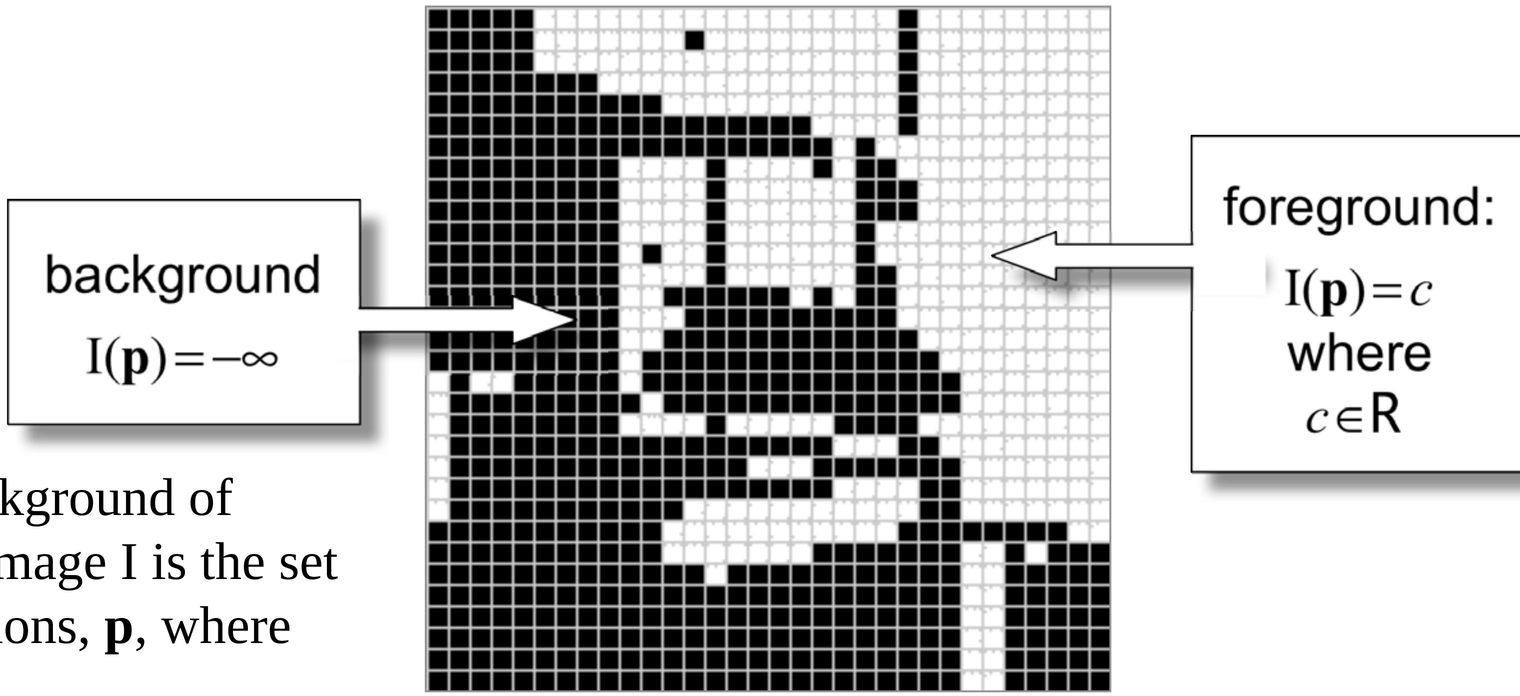
$$A^C = \{\omega \mid \omega \notin A\}$$

$$A - B = \{\omega \mid \omega \in A, \omega \notin B\} = A \cap B^C$$

A Binary Image

The foreground of binary image I is the set of locations, **p**, where I(**p**) = v_{fg}.

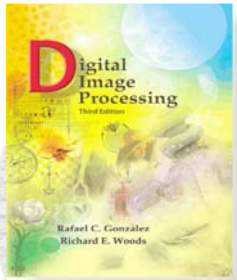
$$\mathrm{FG}\{\,\mathrm{I}\,\} = \{\ \mathrm{I}(\mathbf{p}),\ \mathbf{p} = (r,c) \in S_{\mathrm{P}}\ \big|\ \mathrm{I}(\mathbf{p}) = v_{\mathrm{fg}}\ \}$$



The background of binary image I is the set of locations, **p**, where I(**p**) = v_{bg}.

$$\mathrm{BG}\{\,\mathrm{I}\,\} = \{\ \mathrm{I}(\mathbf{p}),\ \mathbf{p} = (r,c) \in S_{\mathrm{P}}\ \big|\ \mathrm{I}(\mathbf{p}) = v_{\mathrm{bg}}\ \}.$$

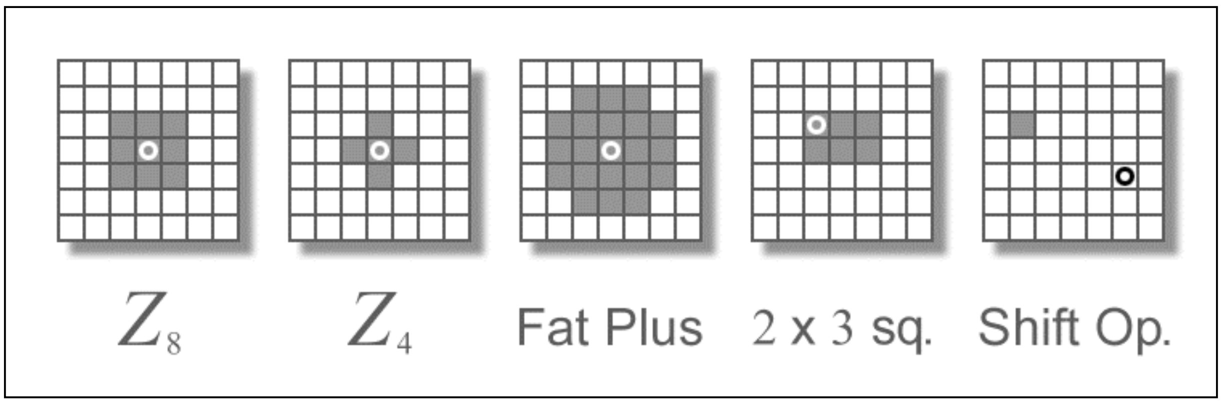
This represents a digital image. Each square is one pixel.



Structuring Elements

A structuring element is a small image – used as a moving window – whose support* delineates pixel neighborhoods in the image plane.

Example SEs



FG is gray ; BG is white

It can be of any shape, size, or connectivity (more than 1 piece, have holes). In the above figure the circles mark the location of the structuring element's origin which can be placed anywhere relative to its support.

*the support of a binary image is the set of foreground pixel locations within the image plane

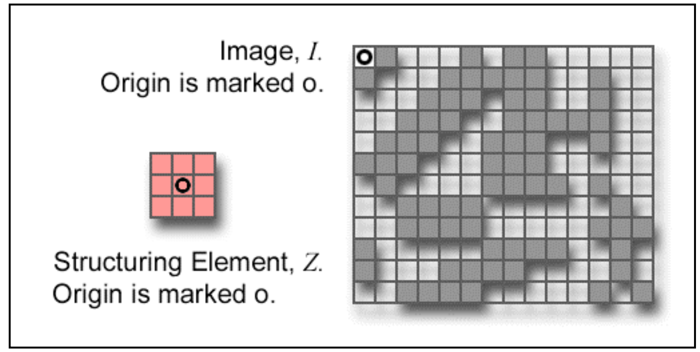
Structuring Elements

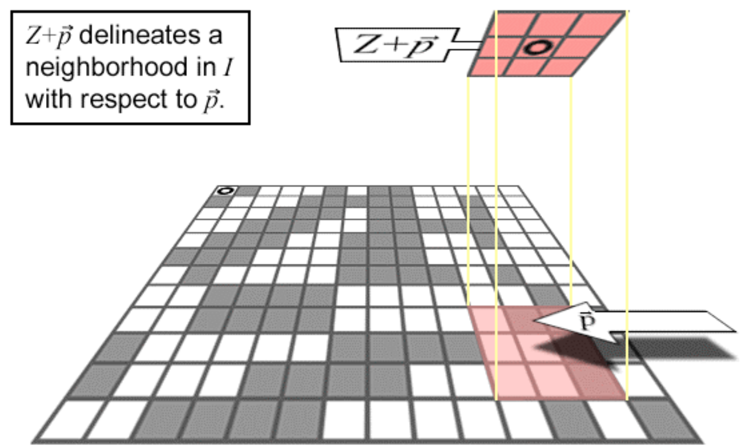
Let I be an image and *Z* a SE.

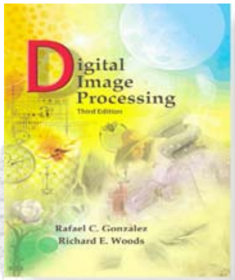
Z+**p** means that *Z* is moved so that its origin coincides with location **p** in S_p.

Z+**p** is the *translate* of *Z* to location **p** in S_p.

The set of locations in the image delineated by Z+**p** is called the *Z-neighborhood* of **p** in I denoted N{I,*Z*}(**p**).





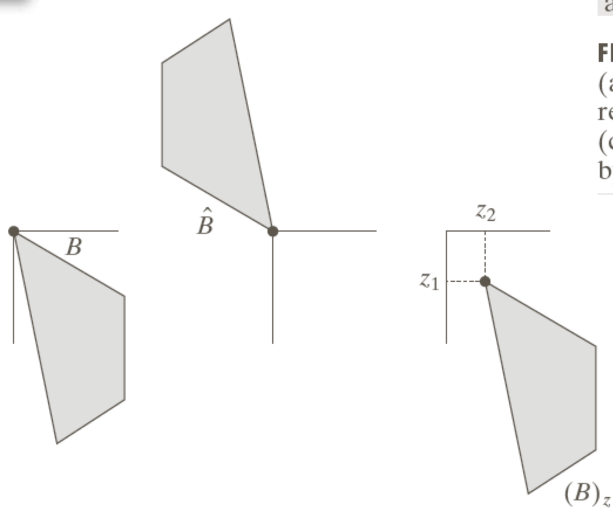


Basic Set Operations

Reflection of B about the origin

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

Original set B



a b c

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

Translation of B by $z=(z_1,z_2)$

$$(B)_Z = \{c \mid c = b + z, \text{ for } b \in B\}$$

Image morphology includes two operations (reflection and translation) not normally used in set theory.

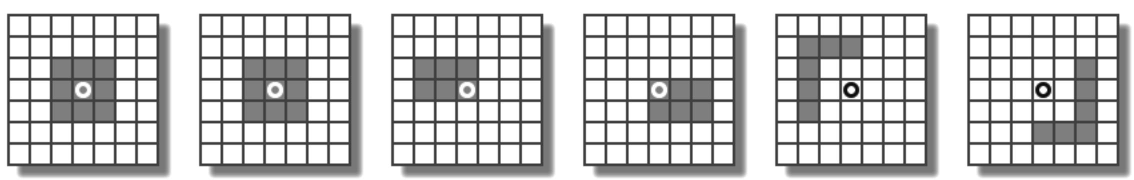
Structuring Elements

Let Z be a SE and let **S** be the square of pixel locations that contains the set {(r,c), (–r,–c) | (r,c)∈ support(Z)}. Then

$$\hat{Z}(\rho,\chi) = Z(-\rho,-\chi) \quad \text{for all} \quad (\rho,\chi) \in \mathbf{S}.$$

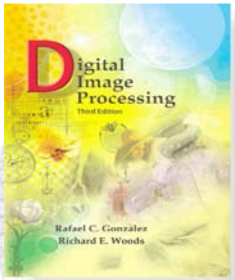
is the reflected structuring element.

$\hat{Z}$ is Z rotated by 180º around its origin.



Z_1 $\check{Z}_1$ Z_2 $\check{Z}_2$ Z_3 $\check{Z}_3$

*the support of a binary image is the set of foreground pixel locations within the image plane



Structuring Elements

Structuring Elements

Structuring Elements for images need to be rectangular arrays

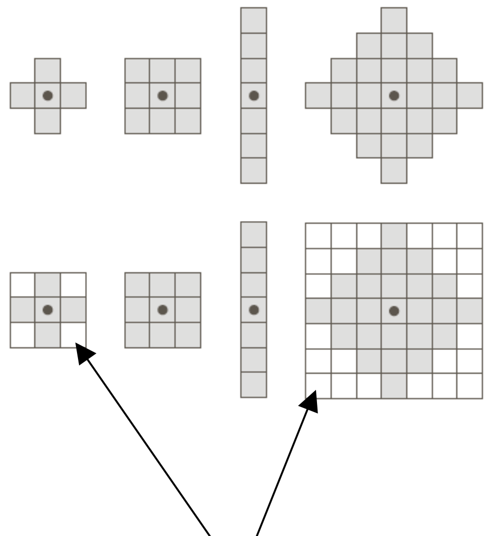


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Expanded with appropriate number of background elements to form a rectangle

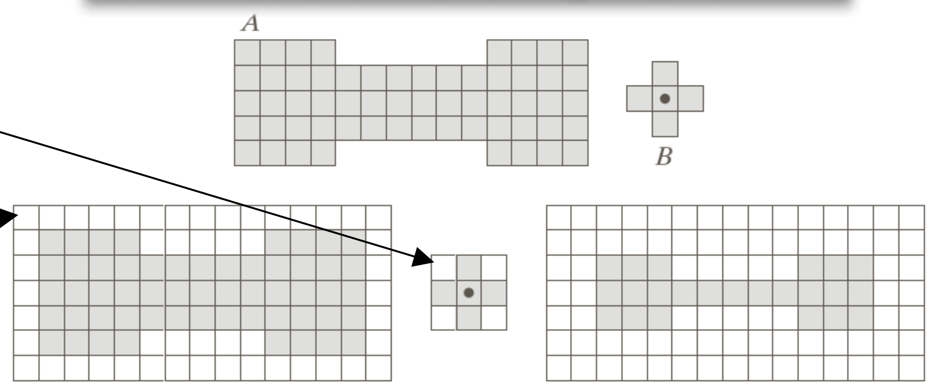
Origins of structuring elements shown as black dots

Erosion

Set A and structuring element B

1. Make structuring element rectangular

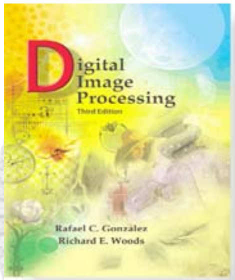
2. Pad set A with background elements such that the entire structuring element is in the image when its origin is on the border of the original set



3. Output set C is foreground only when the foreground of B is entirely contained in the foreground of A — this is called erosion

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

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

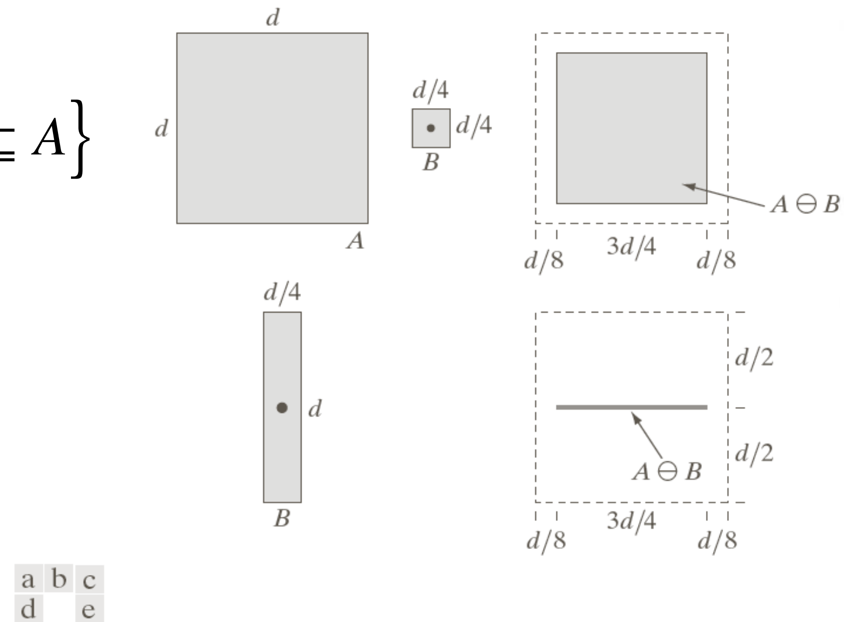


FIGURE 9.4 (a) Set A. (b) Square structuring element, B. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.

The erosion of A by B is the set of all points z such that the translation of B by z is contained in A

In this case the structuring element B is the same height as A so the erosion is a straight line.

Structuring element is NOT drawn to scale.

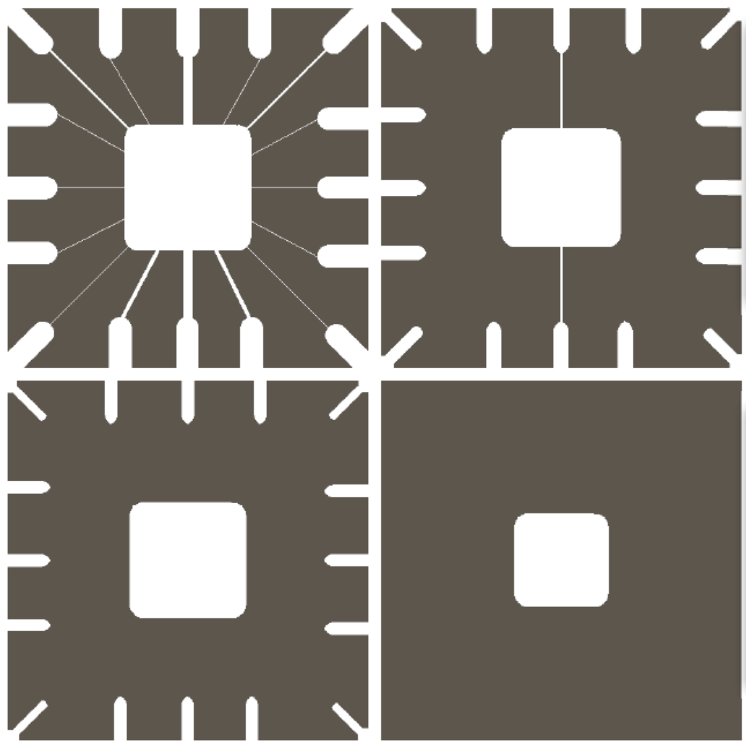
Erosion

Original image

Erosion by 11x11 structuring element of all 1's

Erosion by 15x15 structuring element of all 1's

Erosion by 45x45 structuring element of all 1's



a b
c d

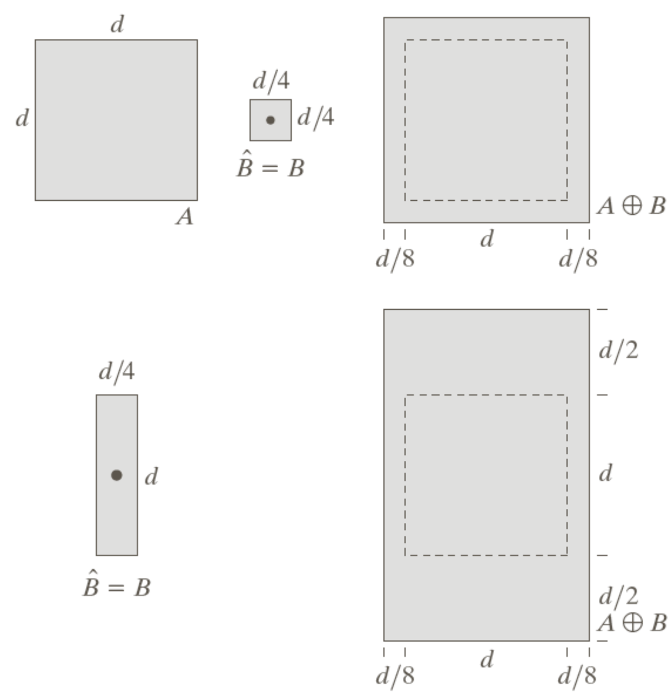
FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11×11, 15×15, and 45×45, respectively. The elements of the SEs were all 1s.

Erosion using a variety of square structuring elements to remove image elements

Dilation

Dilation by a small square structuring element extends the set A by 1/2 the width of the structuring element

Dilation by a rectangular structuring element also extends the set A by 1/2 the width of the structuring element but not uniformly in x and y



a b c
d e

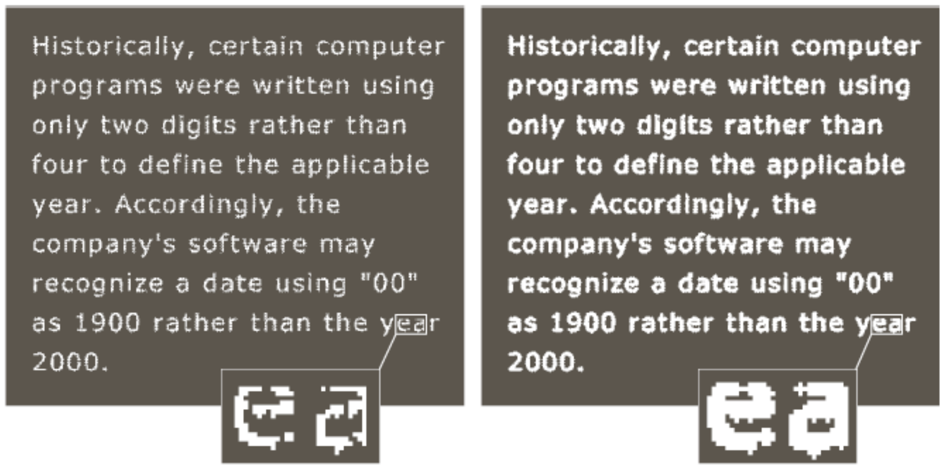
FIGURE 9.6 (a) Set A. (b) Square structuring element (the dot denotes the origin). (c) Dilation of A by B, shown shaded. (d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference

$$A \oplus B = \left\{ z \mid \left(\hat{B}\right)_z \cap A \neq 0 \right\}$$

Dilation is the set of all translations z such that the translated reflection of B is in the foreground of A

Dilation

Text with broken characters


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

a c
b

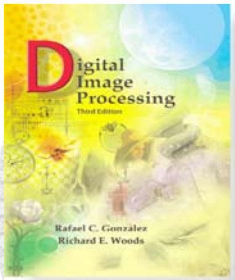
FIGURE 9.7 (a) Sample text of poor resolution with broken characters (see magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

0	1	0
1	1	1
0	1	0

Dilation of A by B joins broken segments

4-connected structuring element B

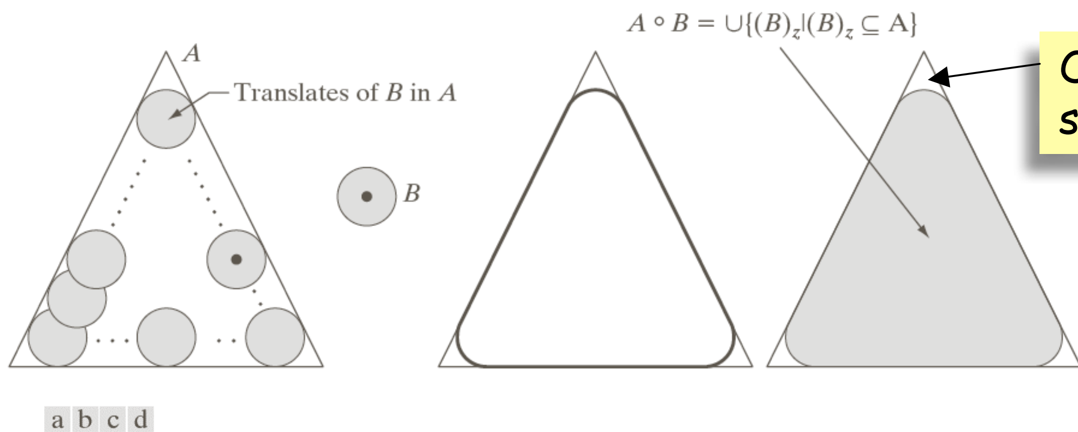
Dilation works on binary images (in this case) and the result is binary; correlation would produce gray scale images



Opening

An opening is the set of all translates of B that fit inside A

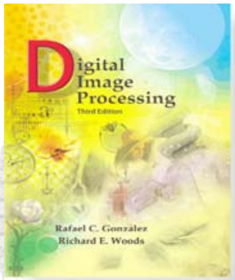
$$A \circ B = \cup \left\{ (B)_z \mid (B)_z \subset A \right\} = \left(A \ominus B \right) \oplus B$$



Opening removes sharp corners

FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

This is the geometric interpretation of opening for a round structuring element and is usually called the "rolling ball"



Closing

An closing is the set of <u>all</u> ω such that the translate of B does not intersect A for any translate of B that contains ω

$$A \bullet B = \left\{ \omega \mid (B)_z \cap A \neq 0 \right\} = (A \oplus B) \ominus B$$

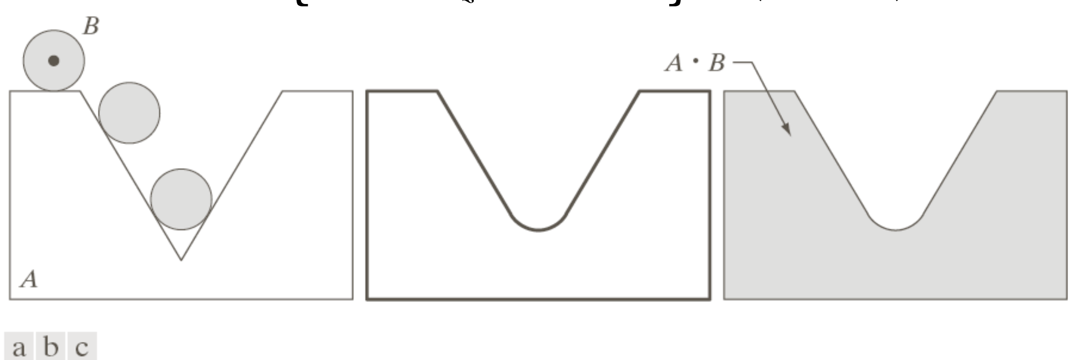
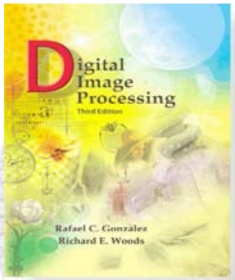


FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

The geometric interpretation of closing for a round structuring element involves "rolling the ball" on the outside of the set A to define the boundary of A•B



Opening, Closing and Topology

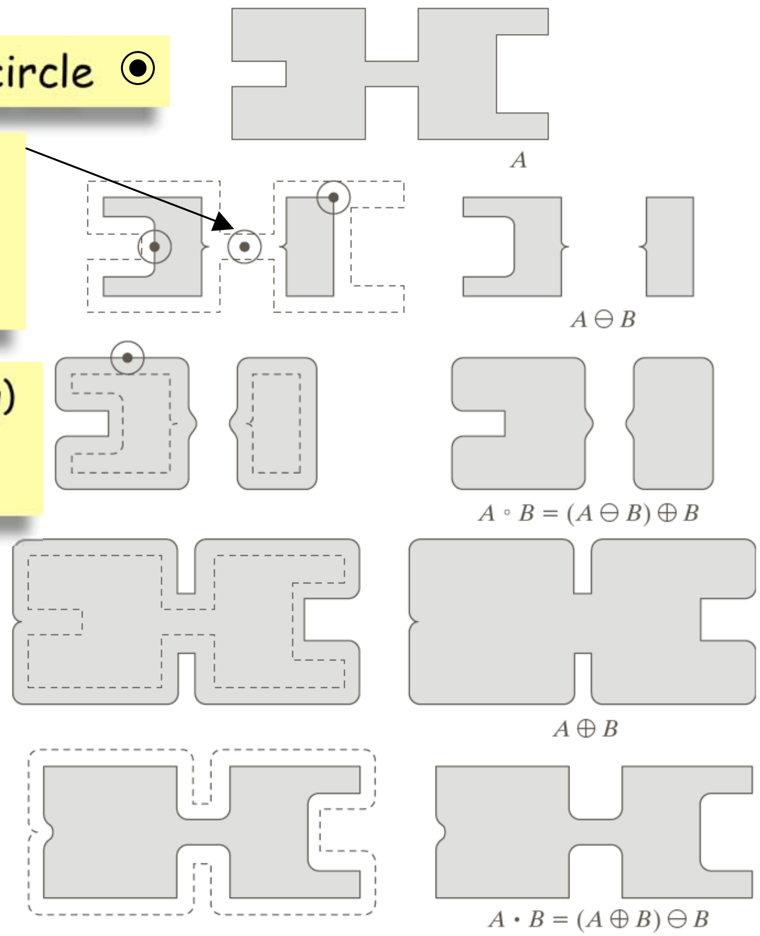
The structuring element is a small circle ◉

Erosion by this structuring element removes the narrow bridge joining the fingers (B must fit inside A to preserve topology)

A subsequent dilation (forming a closing) cannot restore the bridge so a closing can remove small features.

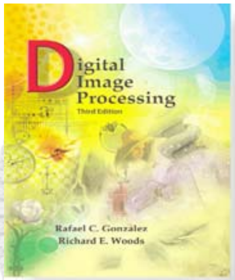
Dilation of the original image removes the notch between the left fingers.

A subsequent erosion (forming an opening) cannot restore the fingers so an opening can remove small indentations.



a	
b	c
d	e
f	g
h	i

FIGURE 9.10 Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.

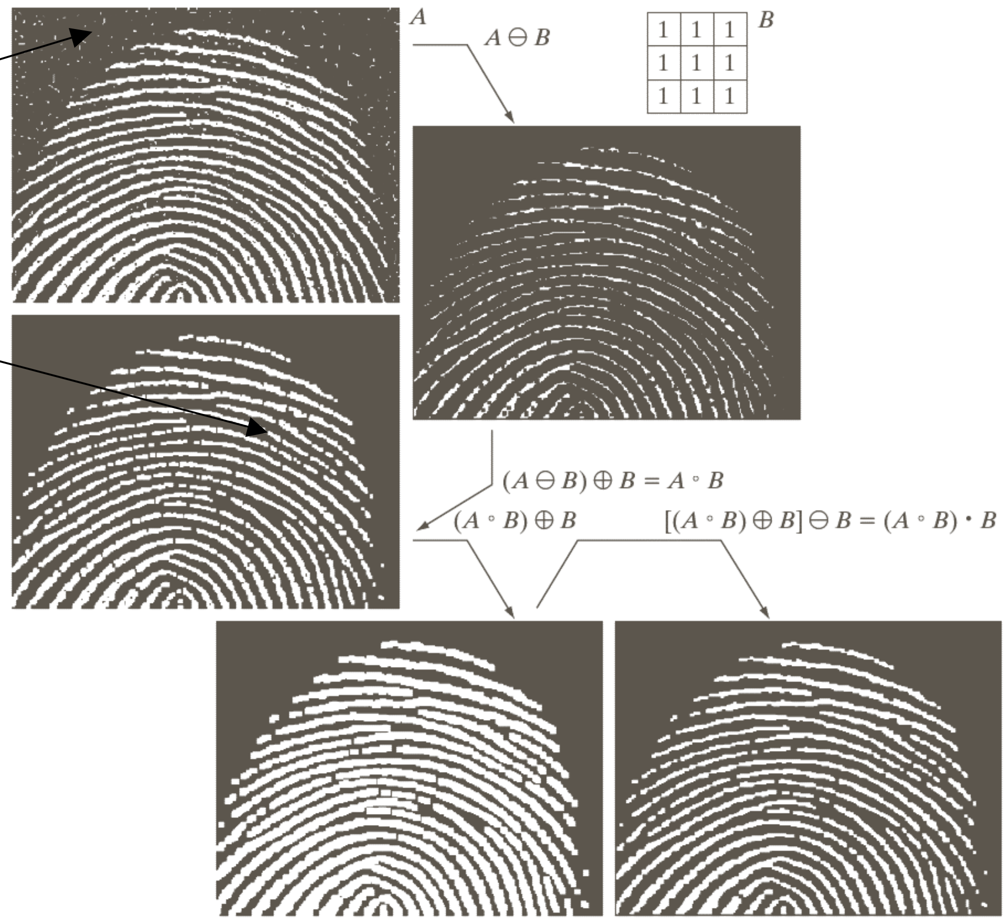


Morphological Image Processing

<u>Remove noise in fingerprints:</u>

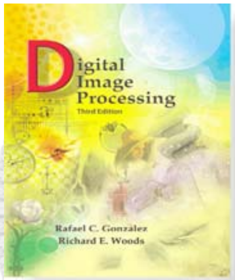
1. Erode A by B to eliminate white on black noise
2. Dilate by B (completing a closing) to restore the size of the ridges
3. Dilate by B to eliminate gaps (black on white noise) in the ridges
4. Erode by B (completing an opening) to restore the size of the thickened ridges

There is some remaining noise because the process (closing followed by an opening) does not maintain connectivity.



a b
d c
e f

FIGURE 9.11 (a) Noisy image. (b) Structuring element. (c) Eroded image. (d) Opening of A. (e) Dilation of the opening. (f) Closing of the opening. (Original image courtesy of the National Institute of Standards and Technology.)



"Hit or Miss" Algorithm

a) We want to find the location of "D"

b) Let D be enclosed by a small background W. Define the local background W-D as shown

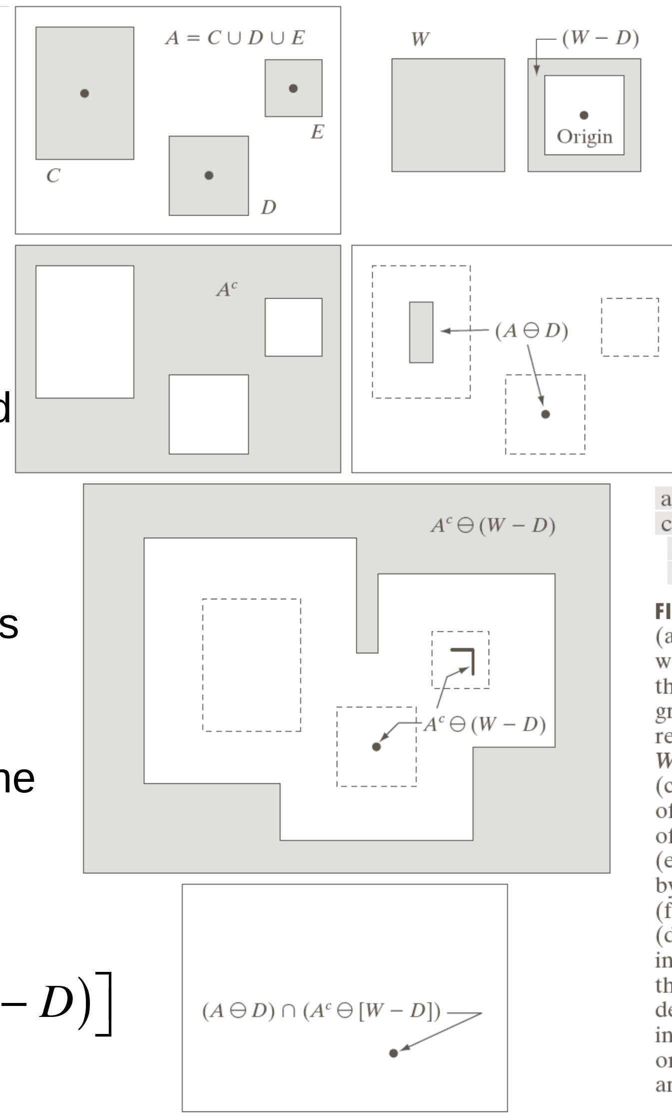
c) Consider the complement A^C of the set of all shapes, i.e., A=CUDUE

d) Erode A by D. Since E is smaller than D it disappears. D eroded by D is a single point and C eroded by D is a rectangular region of all locations of D inside C.

e) Now erode of A^C by W-D to give the set of all translates of W-D such that the center of W-D is in A^C. Note that W-D fits around D giving a single point. Since E is smaller than D there is also a set of points inside E corresponding to the translates pf W-D containing E.

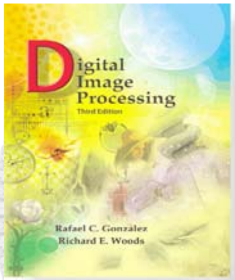
f) The intersection of $A \ominus D$ and $A^C \ominus (W\text{-}D)$ is the location of D.

This can be written as: $A \circledast B = \left(A \ominus D\right) \cap \left[A^C \ominus \left(W - D\right)\right]$



a b
c d
e
f

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



Morphological Image Processing

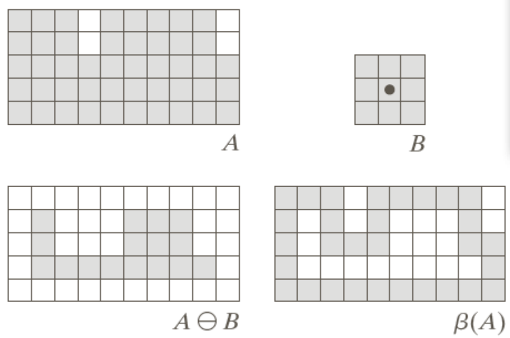
Boundary extraction:

1. Erode I by an appropriate structuring element B
2. Subtract this from A to get the boundary β(A)

NOTE: A 3x3 structuring element will give 1-pixel boundaries; a 5x5 will give two pixel boundaries

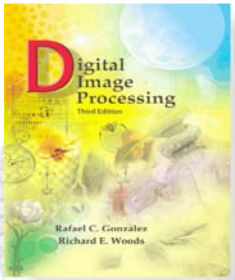
A⊖B gives the interior of A

β(A)=A-(A⊖B) gives the boundary of A



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.

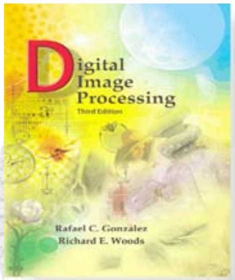


Morphological Image Processing

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).

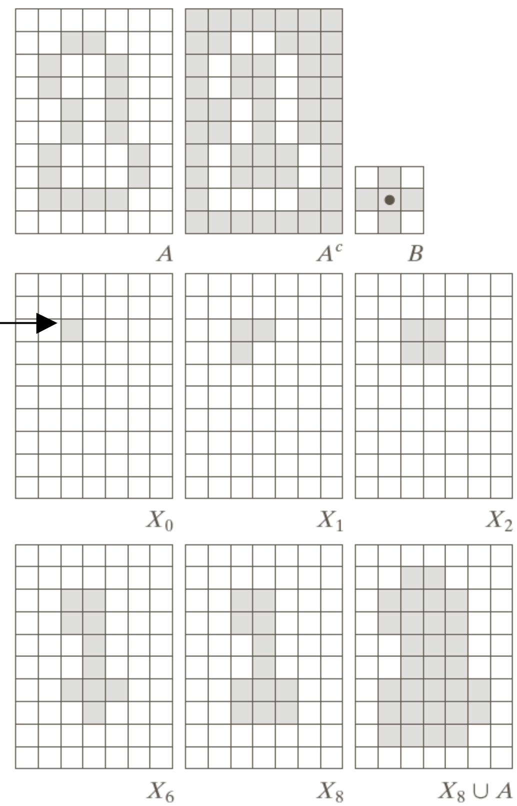
Erode the image using a 3x3 uniform structuring element and subtract from the original to get the boundary image on the left.



Region Filling

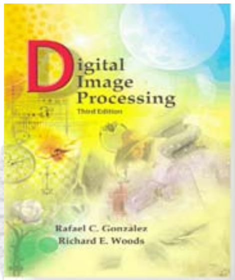
Original image A, its complement A^C, and structuring element B

1. Let X_0=P, an initial point inside the object
2. $X_1=(P\oplus B)\cap A^C$, a dilation which stops at all pixels in the boundary — this is called a conditional dilation
3. Repeat $X_k=(X_{k-1}\oplus B)\cap A^c$ until $X_k=X_{k-1}$

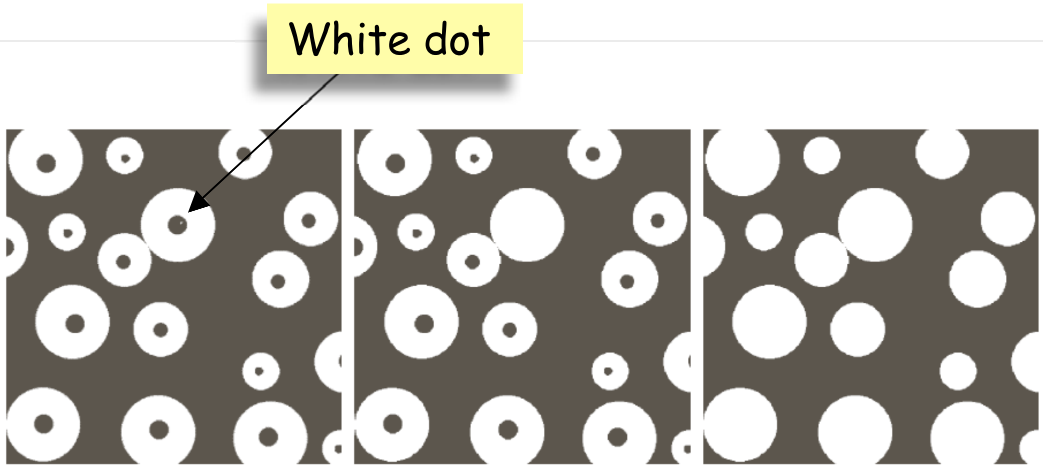


a b c
d e f
g h i

FIGURE 9.15 Hole filling. (a) Set A (shown shaded). (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)].



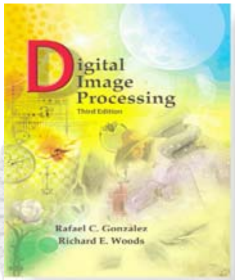
Region Filling



a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

An application of region filling to thresholded images of ball bearings: start with white dot in indicated bearing image and use repeated conditional dilations. Filling all bearing images would require an algorithm such as morphological reconstruction to identify starting points.



Connectivity

Structuring element assumes 8-connectivity

Initial starting point P

1. Dilate P to give all adjacent connected points, i.e., $X_o = P \oplus B$
2. Then $X_1 = X_o \cap A$ gives all the 1's in A connected to P, i.e., the elements of the object. This is a conditional dilation.
3. Repeat $X_k = (X_{k-1} \oplus B) \cap A$ until $X_k = X_{k-1}$. This gives all the 8-connected elements of the object in A.

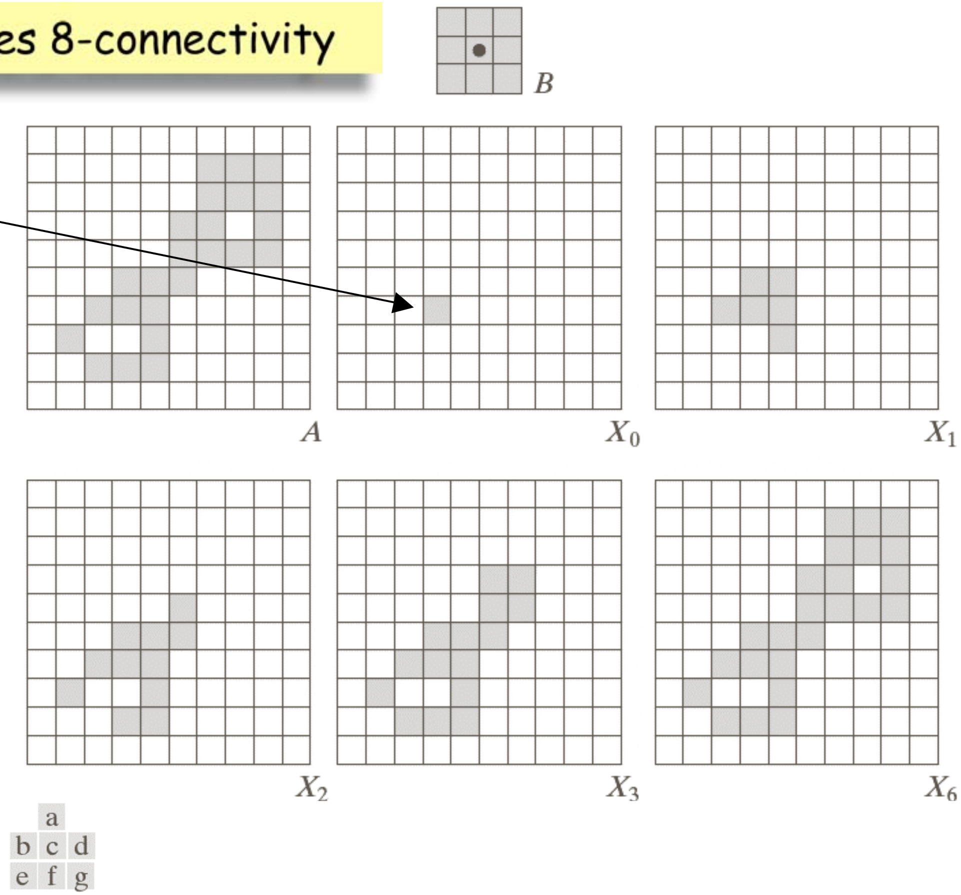


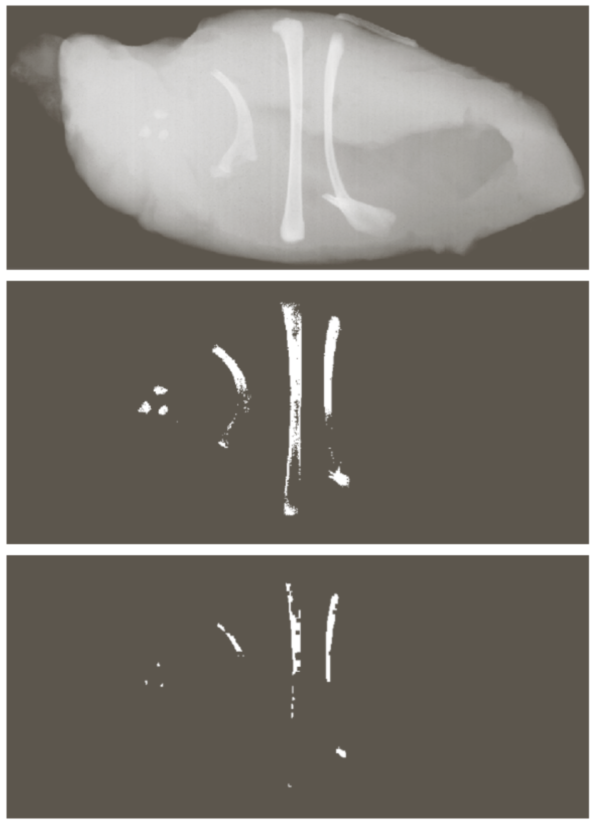
FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

Connectivity

X-ray image of bones in chicken fillet

Connected components found by repeated dilation and intersection with thresholded binary image

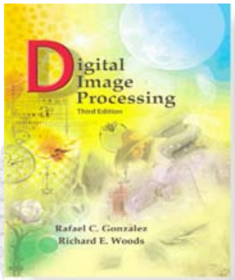
Final image eroded by 5x5 structuring element of 1's



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

a
b
c d

FIGURE 9.18 (a) X-ray image of chicken filet with bone fragments. (b) Thresholded image. (c) Image eroded with a 5×5 structuring element of 1s. (d) Number of pixels in the connected components of (c). (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)



Connectivity

Structuring elements. X's indicate "don't cares".

Use a repetitive "hit or miss" algorithm to construct a convex hull by computing $X_0^1 \circledast B^1$ which finds the locations of B^1 and then adds them to A, i.e., $X_1^1 = (X_0^1 \circledast B^1) \cup A$. Continue this process until nothing changes. Repeat for each B^i.

1. $X_k^i = (X_{k-1}^i \circledast B^i) \cup A$ for i=1,2,3,…, k=1,2,3,4,
2. The convex hull is then $C(A) = D_1 \cup D_2 \cup D_3 \cup D_4$ where $D_i = X^i_{converged}$

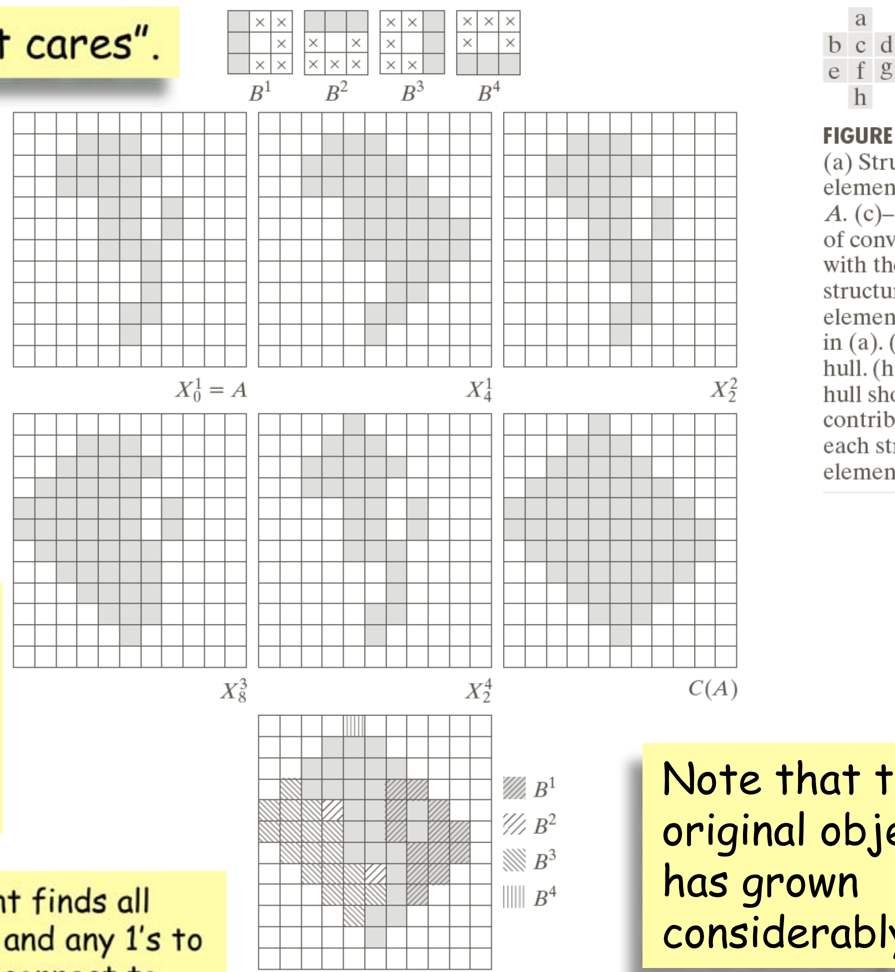
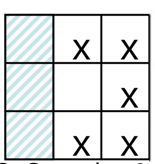


FIGURE 9.19 (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.



In this case the structuring element finds all locations with three 1's on the left and any 1's to the right adding an object pixel to connect to the object. This makes objects bigger.

Note that the original object has grown considerably

Connectivity

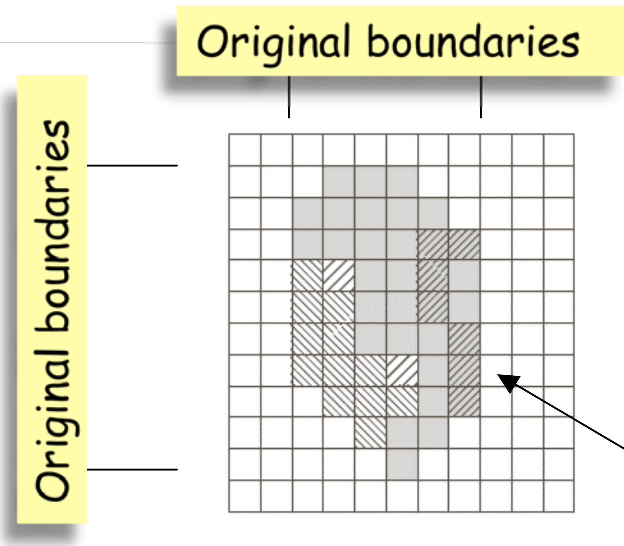
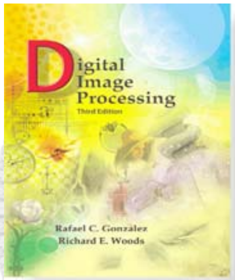


FIGURE 9.20 Result of limiting growth of the convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.

Bounding prevents growth of the convex hull by B^2 beyond the original dimensions of the object

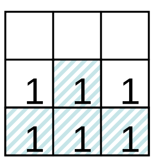


Connectivity

Structuring elements. X's indicate "don't cares".

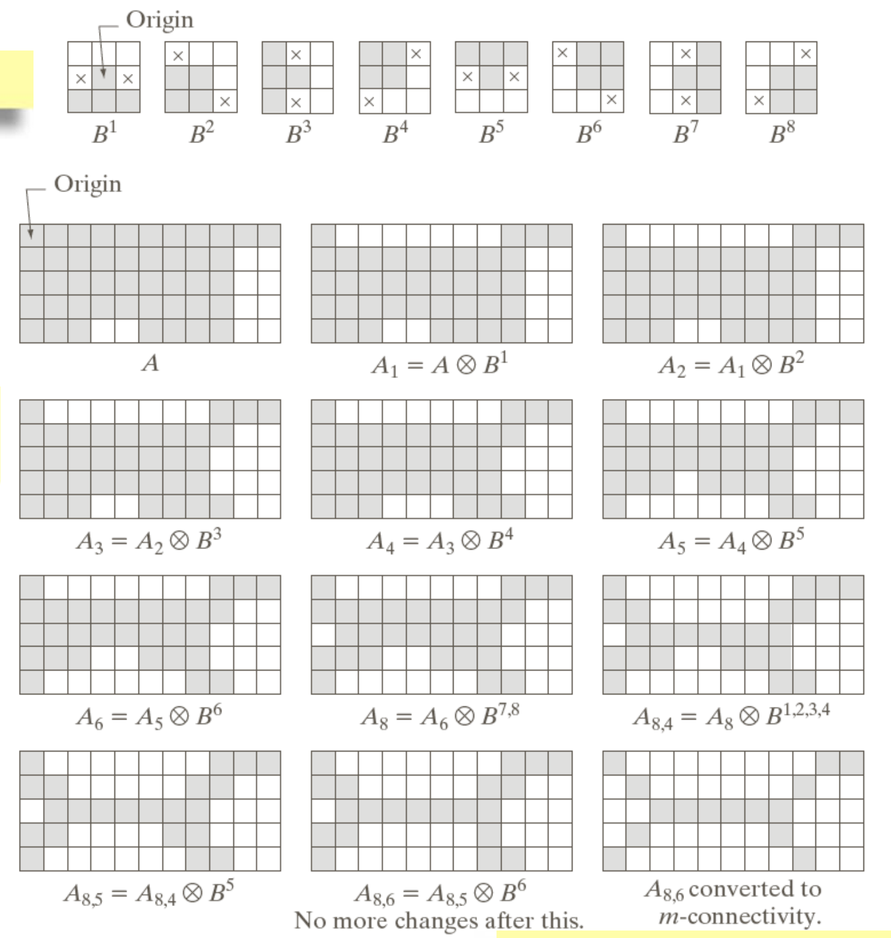
The convex hull built objects up. We use a thinning operator based upon 8-connectivity to reduce objects to their basic structure.

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



1	1	1
1	1	1

Basically, if the structuring element matches we remove that pixel. In this case of 6 1's B^1 matches so we remove the center pixel.



Converted to m-connectivity

	a	
b	c	d
e	f	g
h	i	j
k	l	m

FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to m-connectivity.