Morphological Processing

Morphological Processing

- Morphology: Study of form or structure
- Morphological operates are often applied to binary images or images that are converted to binary by thresholding or a similar segmentation technique
- Foreground pixels
- Background pixels
- Two main morphological operations are:
  - Erosion
  - Dilation

Structuring elements

- Using a structuring elements erosion $\ominus$ and dilation$\oplus$ operators determine the output image.

1
1
1
1

1	1	1	1	1

1	0	1

1	1	1
1	1	1
1	1	1

1	1
1	1

0	1	0
1	1	1
0	1	0

Erosion

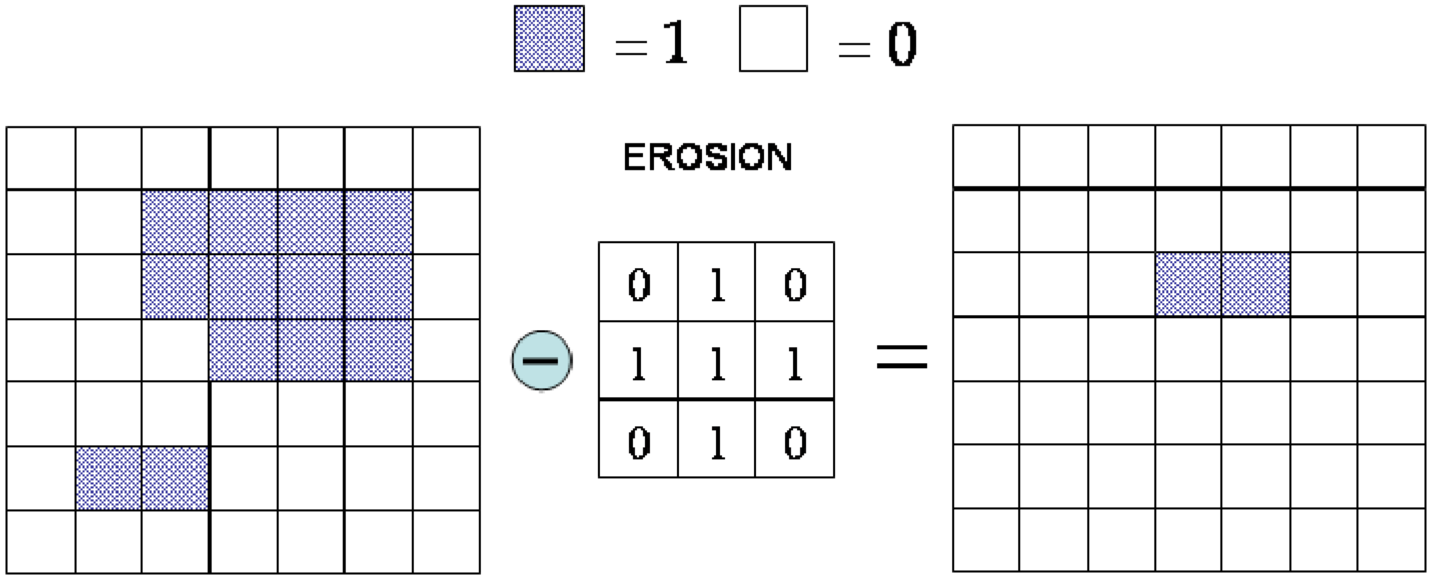
- Erosion $\ominus$

Place the center pixel of the structuring element on each foreground pixel. If *any* of the neighborhood pixels are background pixels, then the foreground is switched to background.

neighborhood pixels are pixels where structuring element entries are equal to 1.

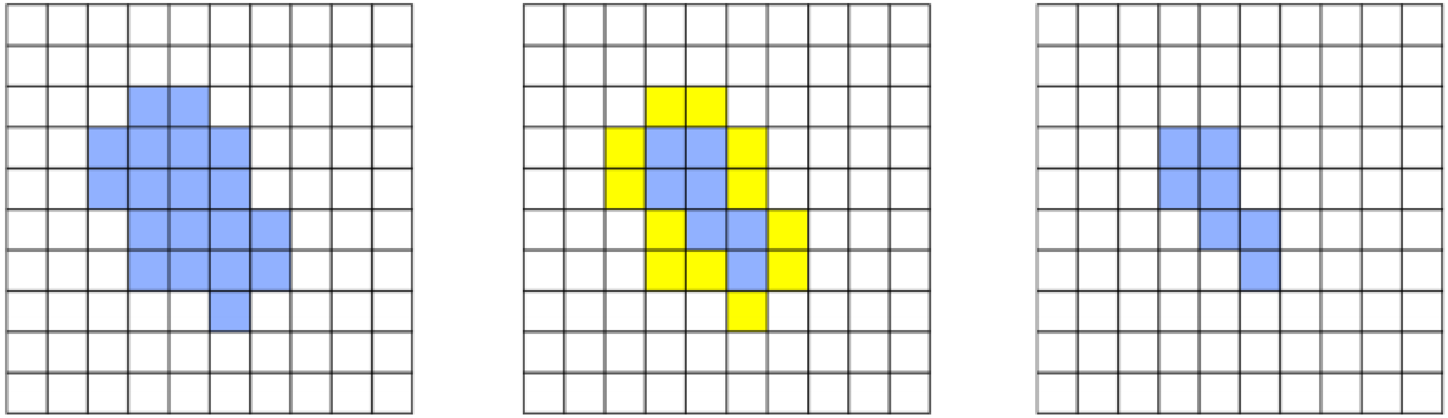
0	1	0
1	1	1
0	1	0

Structuring element

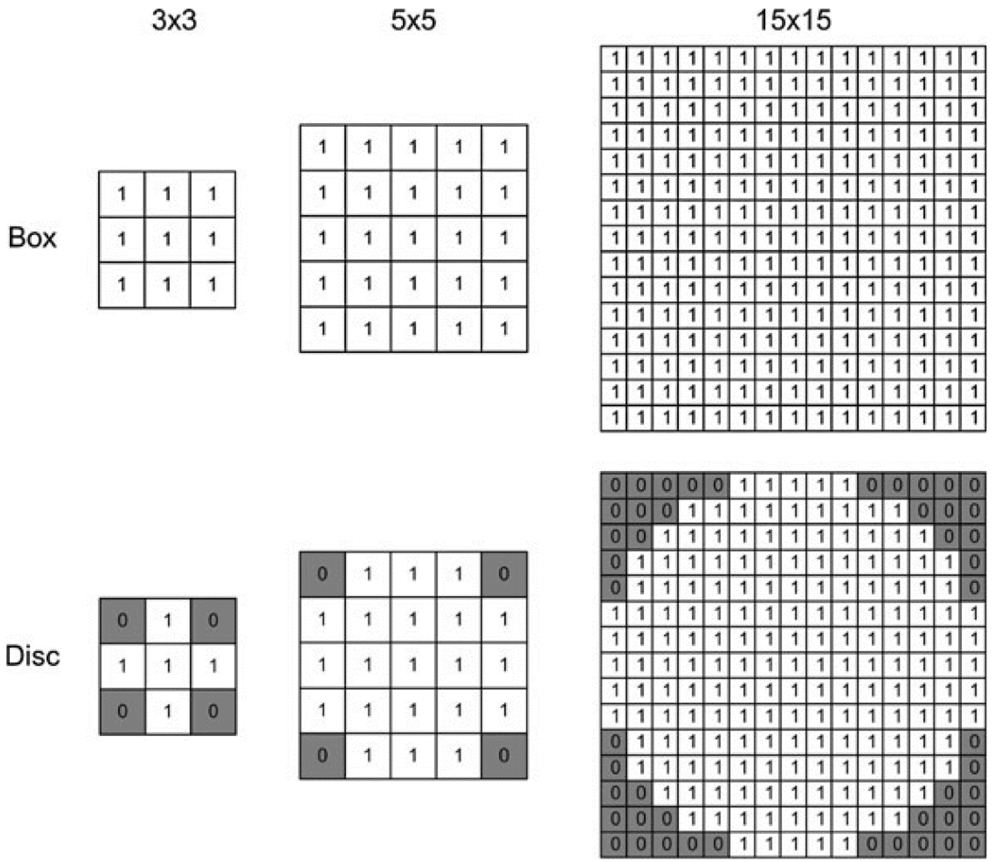


Erosion

- Second example of erosion using the same structuring element

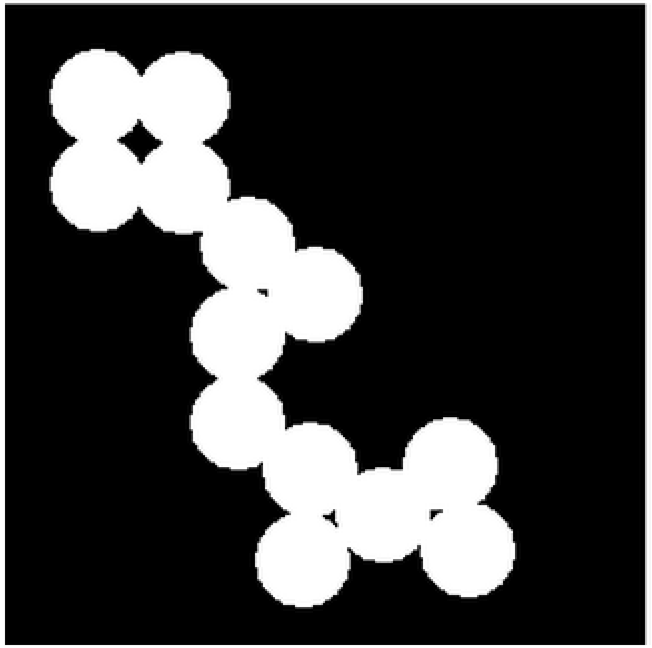


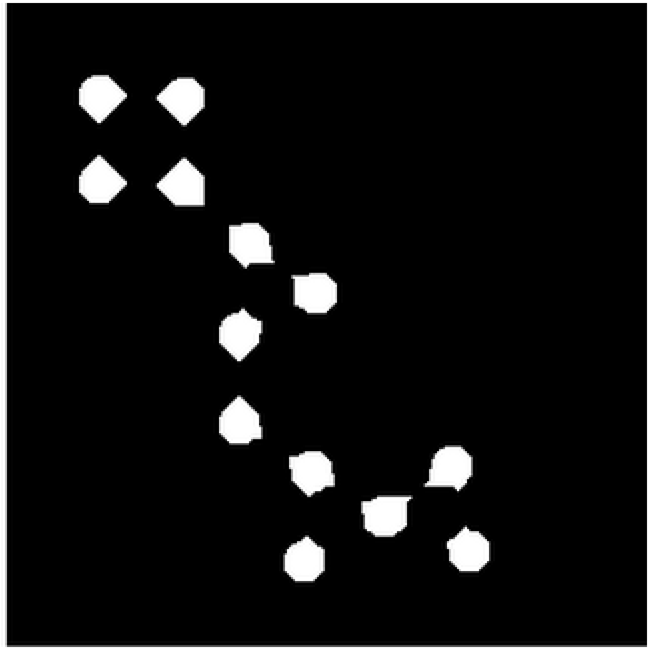
Structuring elements



Erosion

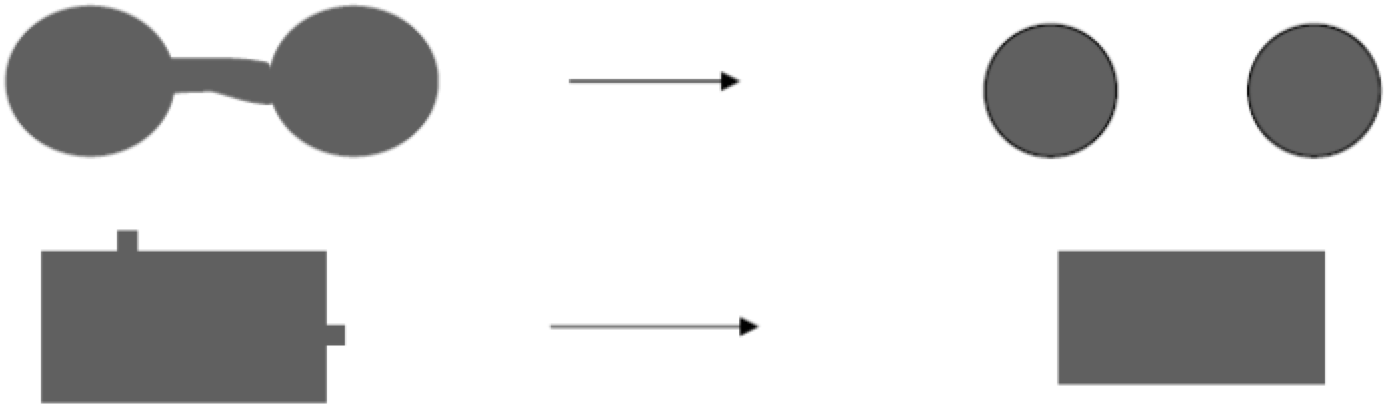
- Structuring element: Disk of size 11. Original circles are of radius 18





Erosion

- Erosion removes branches and edges.



What is the structuring element? What size should be chosen?

Size should be larger than the branches and edges that we plan to remove, and smaller than the object we like to preserve.

Erosion

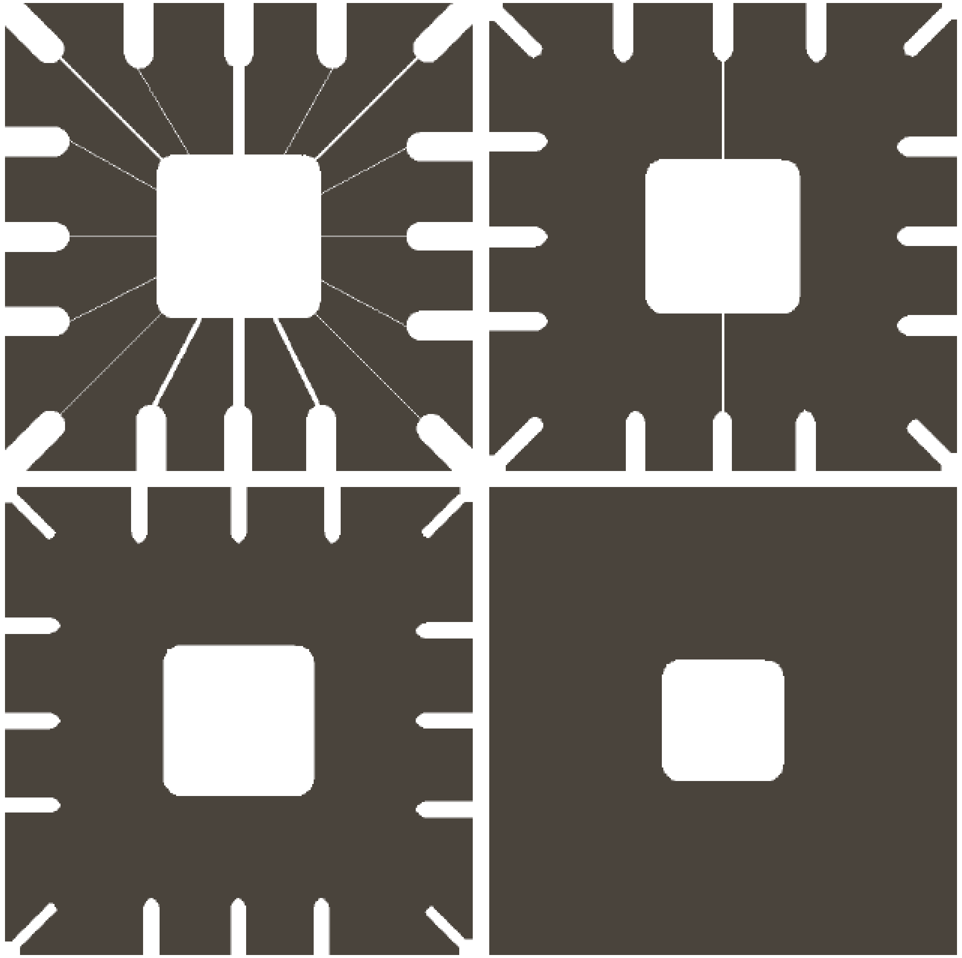
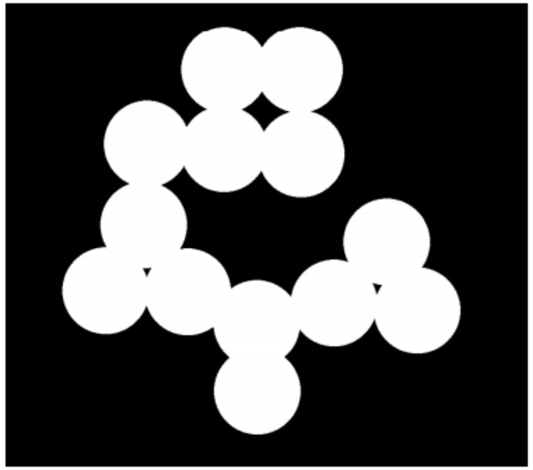
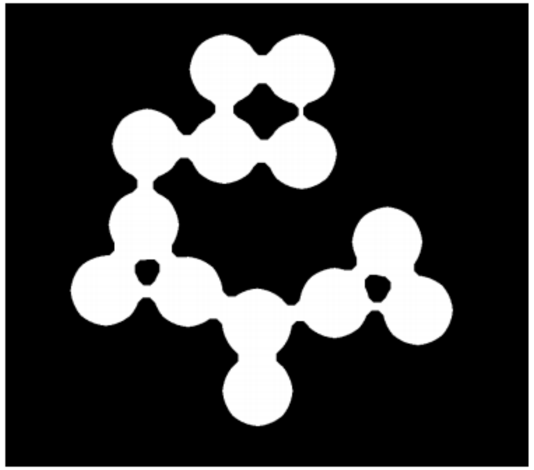


FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11×11, 15×15, and 45×45, respectively. The elements of the SEs were all 1s.

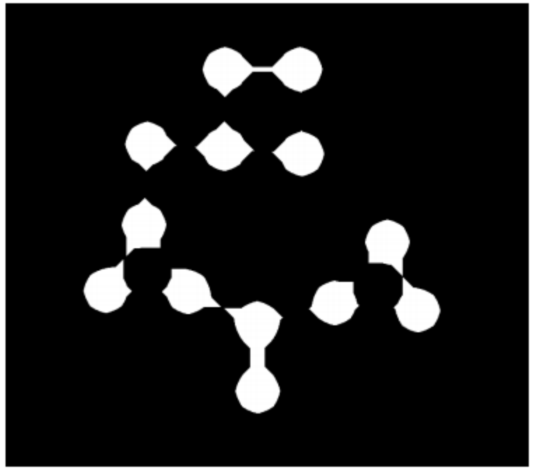
Erosion example



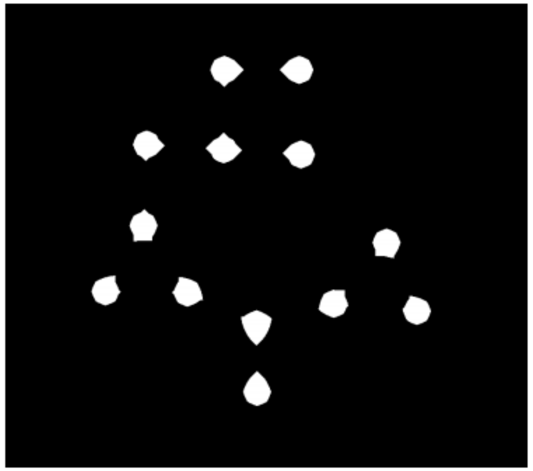
Original binary image
Circles (792x892)



Erosion by disk-shaped
structuring element
Diameter=15



Erosion by disk-shaped
structuring element
Diameter=35



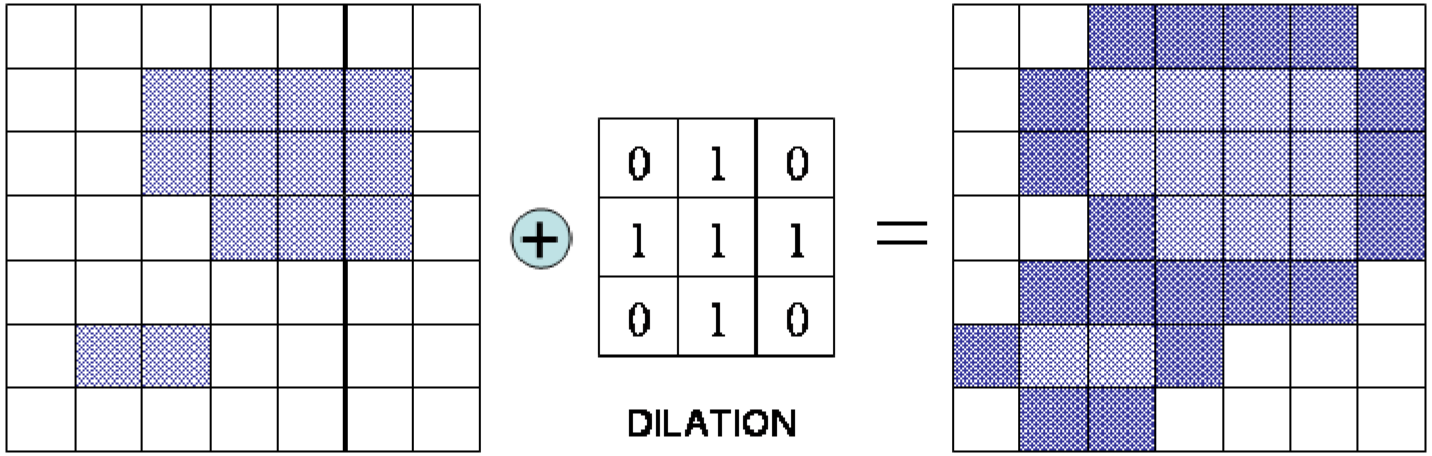
Erosion by disk-shaped
structuring element
Diameter=48

Dilation

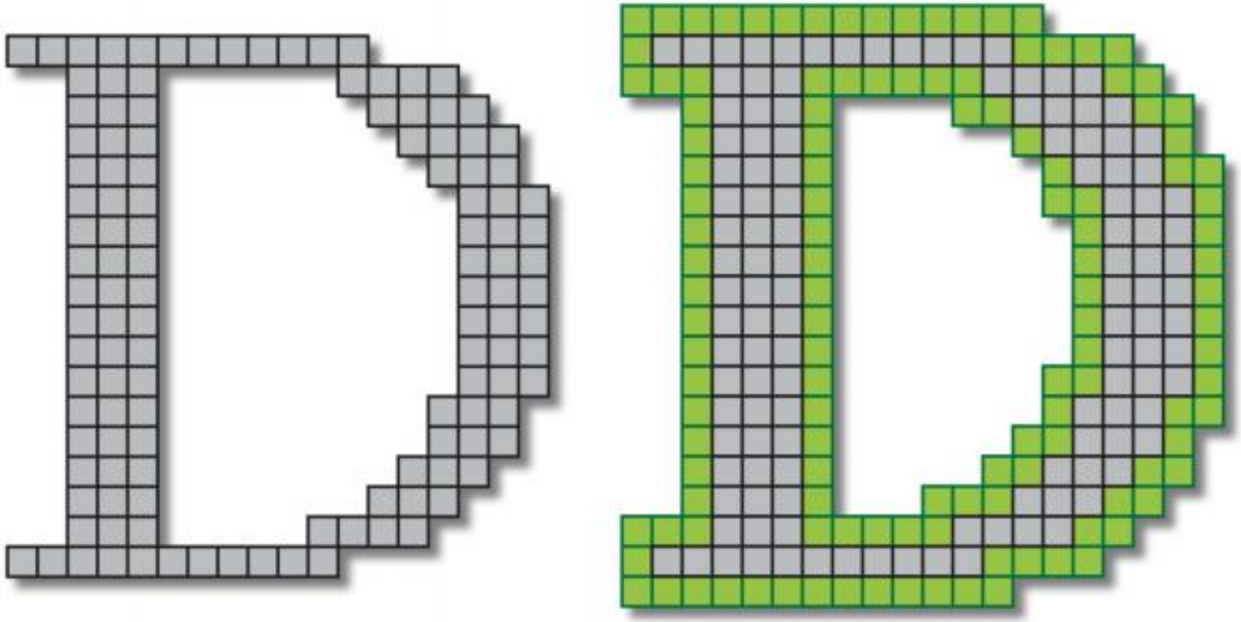
- Dilation $\oplus$

Place the center pixel of the structuring element on each background pixel. If *any* of the neighborhood pixels are foreground pixels, then the background is switched to foreground.

neighborhood pixels are pixels where structuring element entries are equal to 1.



Dilation



Region growth using dilation with a square element

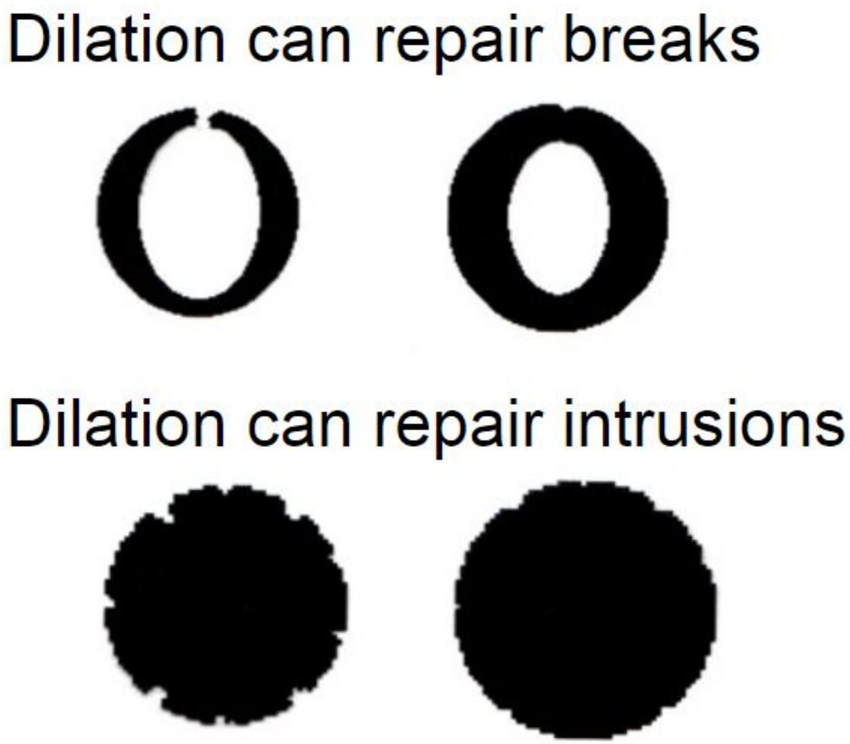
Dilation

Original image

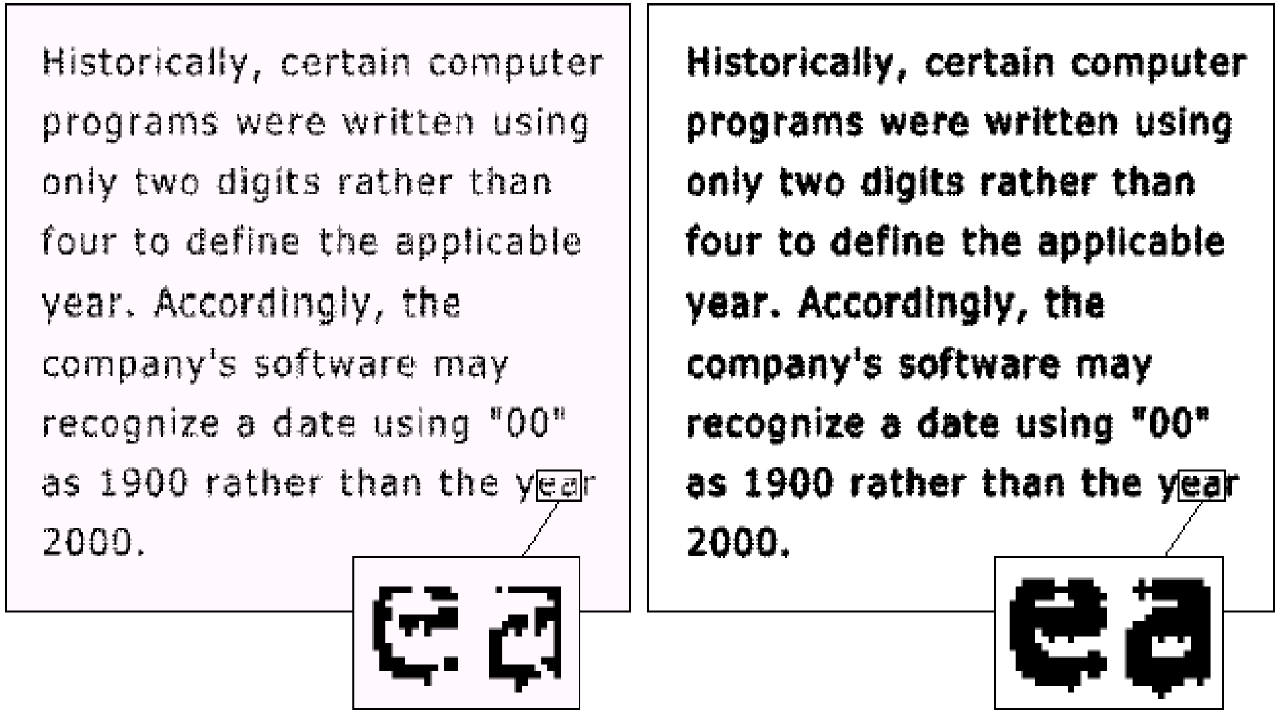
Dilation by 3*3 square structuring element

Dilation by 5*5 square structuring element

Dilation


Dilation can repair breaks
Dilation can repair intrusions

Dilation


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

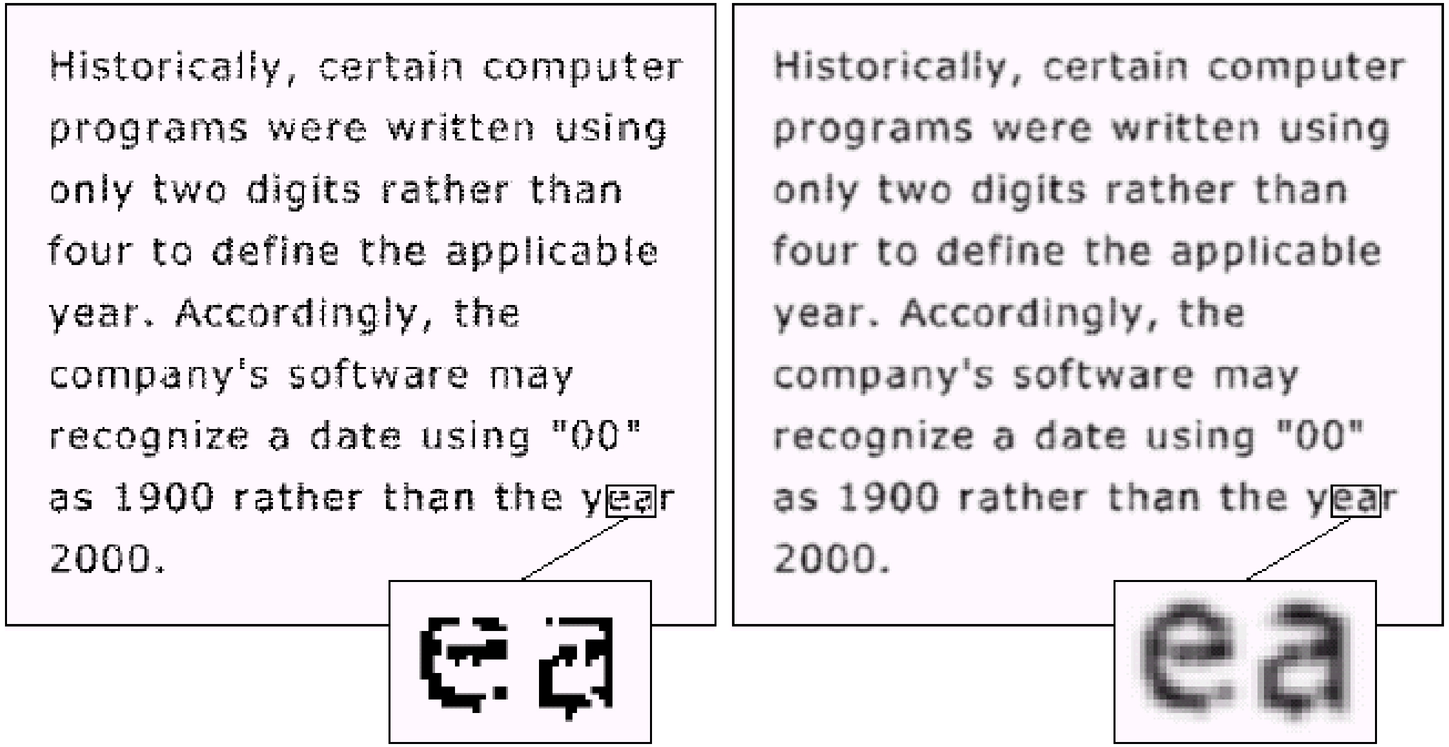
0	1	0
1	1	1
0	1	0

Connecting missing components

Dilation

a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).



Gaussian filtering result

Opening and closing

- One problem with dilation and erosion is that they grow or shrink the binary objects.
- Using a 3 x 3 box structuring element, dilation extends the object by one pixel around the border.
- Similarly, erosion removes a one-pixel of thickness from the border of the object.
- As a solution to this problem, the morphological operations **opening** and **closing** are used.

Opening and closing

Opening: erosion followed by a dilation

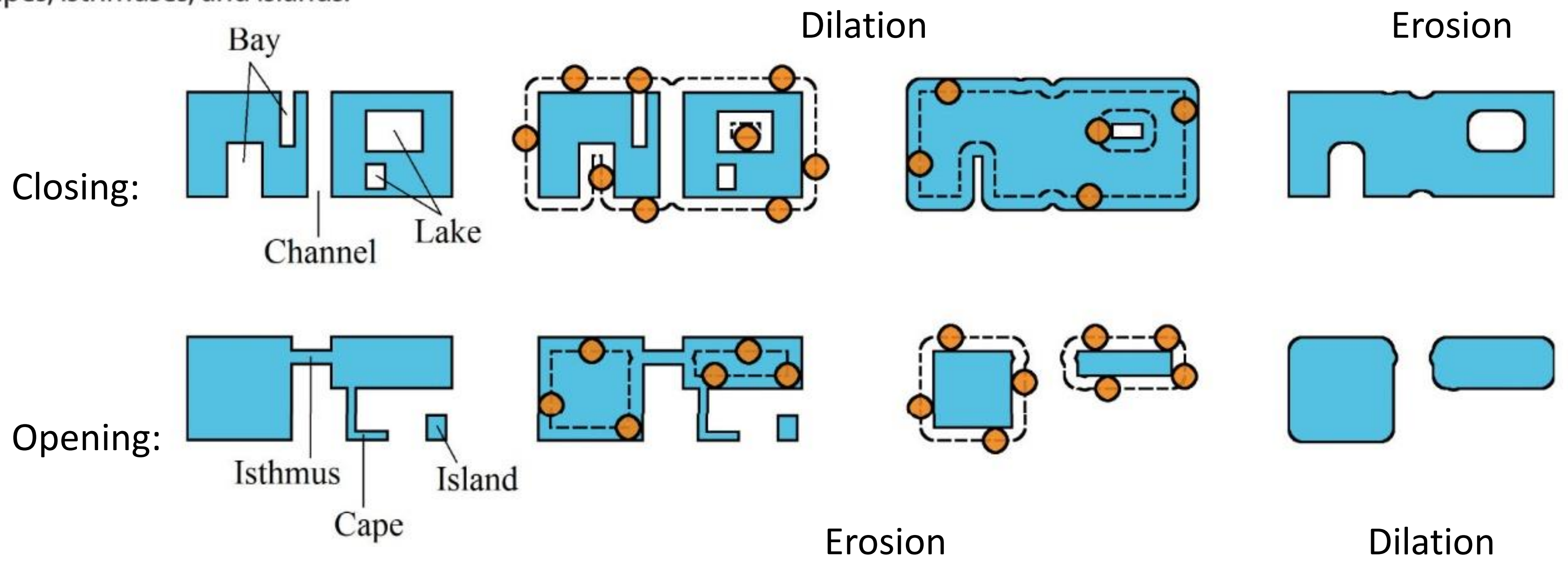
$$A \circ \mathrm{B} = (\mathrm{A} \ominus \mathrm{B}) \oplus \mathrm{B}$$

Closing: dilation followed by an erosion

$$A \bullet \mathrm{B} = (\mathrm{A} \oplus \mathrm{B}) \ominus \mathrm{B}$$

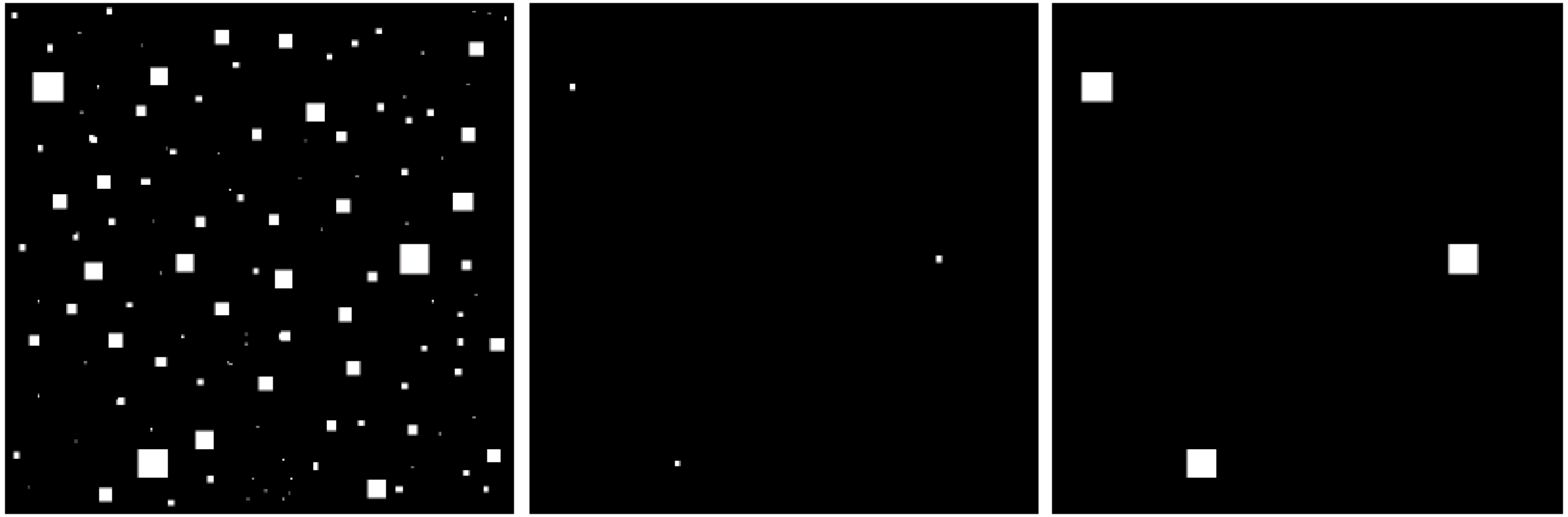
Closing and Opening

Figure 4.10 TOP: Dilation of a binary region by a circular structuring element can be visualized as rolling the disk along the outside of the region; the result is enclosed by the path of the center of the disk. The right column shows the result of closing (dilation followed by erosion), which fills lakes, bays, and channels. BOTTOM: Erosion can be visualized as rolling the disk along the inside of the region; the result is again enclosed by the path of the center of the disk. The right column shows the result of opening (erosion followed by dilation), which removes capes, isthmuses, and islands.



From Image processing and analysis, Birchfield

Opening (erosion followed by dilation)



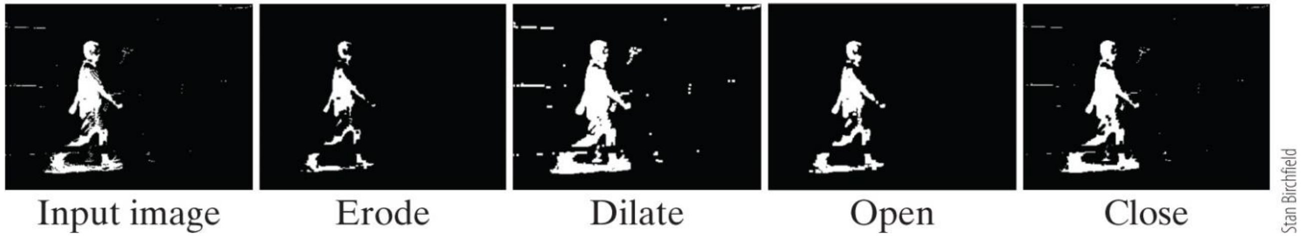
a b c

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

The largest squares are identified by this process.

Opening and closing

Figure 4.11 A binary image and the result of morphological operations: Erode, dilate, open, and close. Erosion removes salt noise but shrinks the foreground. Dilate fills pepper noise but expands the foreground. Opening and closing removes the respective types of noise while retaining the overall size of the foreground.



From Image processing and analysis, Birchfield

Opening example

Original Image

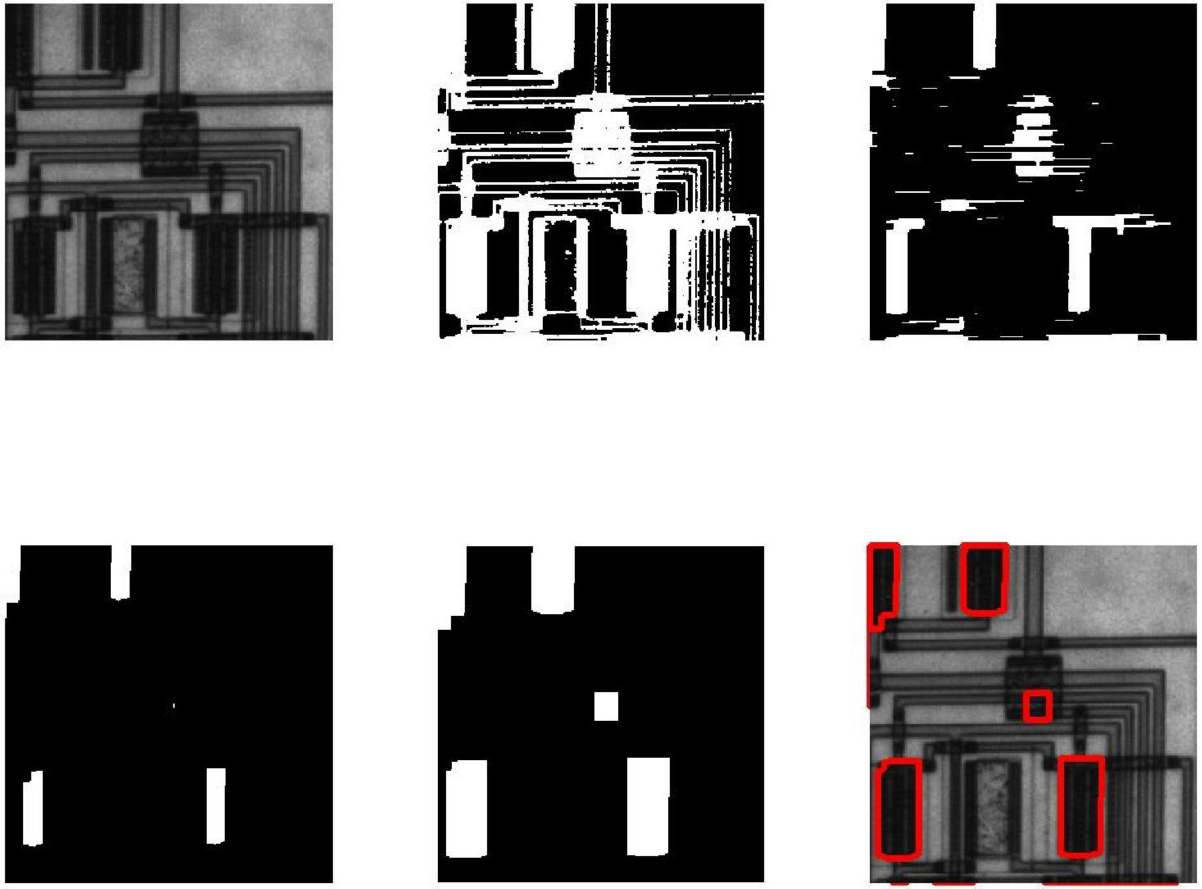
Image After Opening

Closing example

Original Image

Image After Closing

Opening



(a) Original image (b) result after thresholding (c) after erosion horiz. Line length 18 width 3 (d) After erosion with a similar vertical line (e) after dilation horiz. And vertical (f) boundary of remaining objects superimposed on the image

Boundary Extraction

- Using morphological operators, boundaries can be extracted using

$$\boxed{A_B = A - (A \ominus B)}$$

Where A is the binary image, B is the structuring element and $\ominus$ is the erosion operation

Boundary Extraction

Original image: erosion using a disk as a structuring element

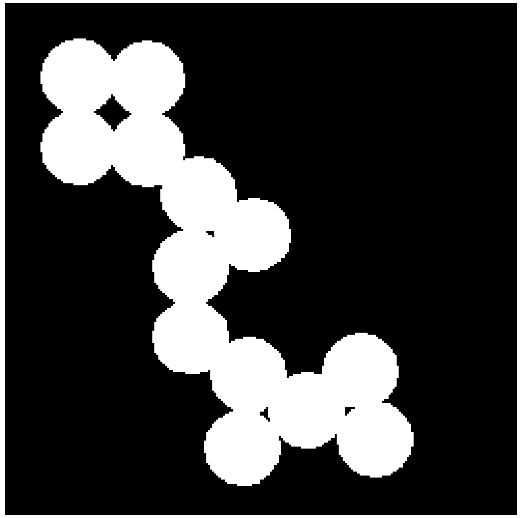
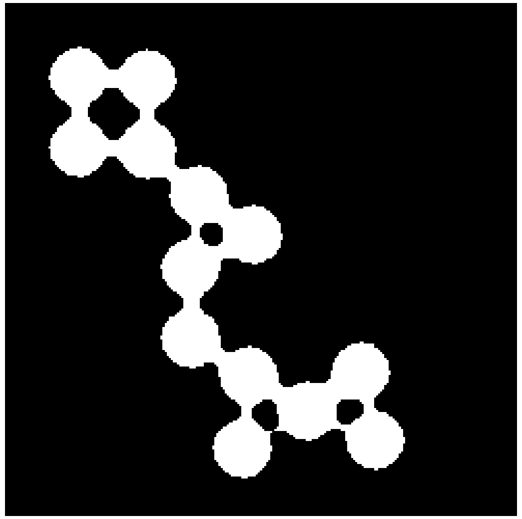
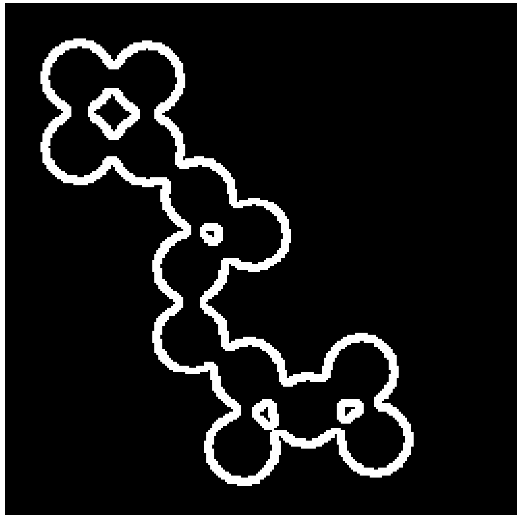


Image A



$A \ominus B$



$A - (A \ominus B)$

Boundary Extraction

Original Image　　Extracted Boundary

Boundary Extraction

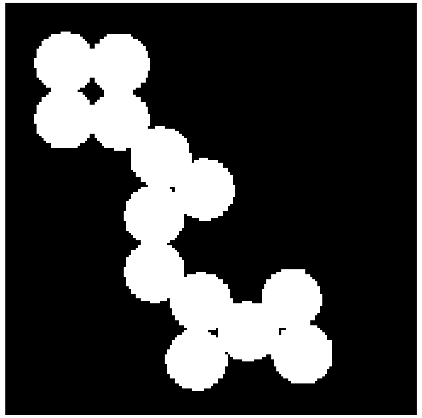
- Boundaries can also be extracted using dilation

$$\boxed{A_B = (A \oplus B) - A}$$

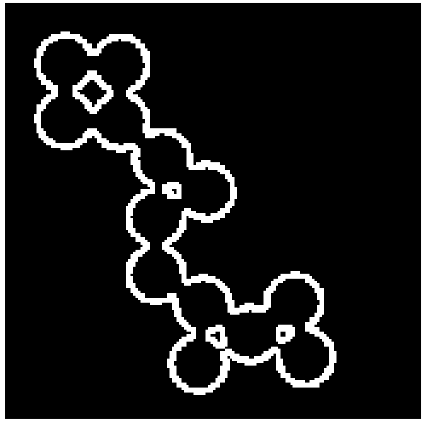
- Or a combination of dilation and erosion

$$\boxed{A_B = (A \oplus B) - (A \ominus B)}$$

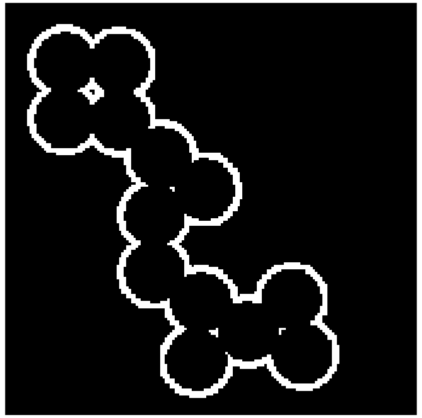
Boundary Extraction



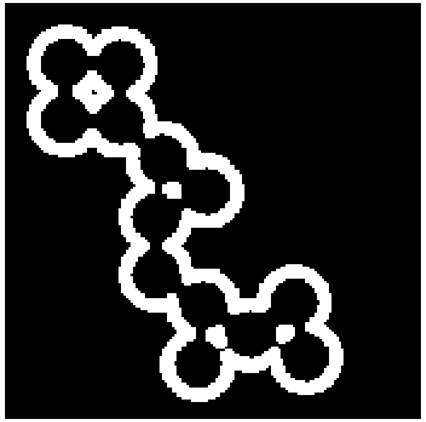
A



$A - (A \ominus B)$



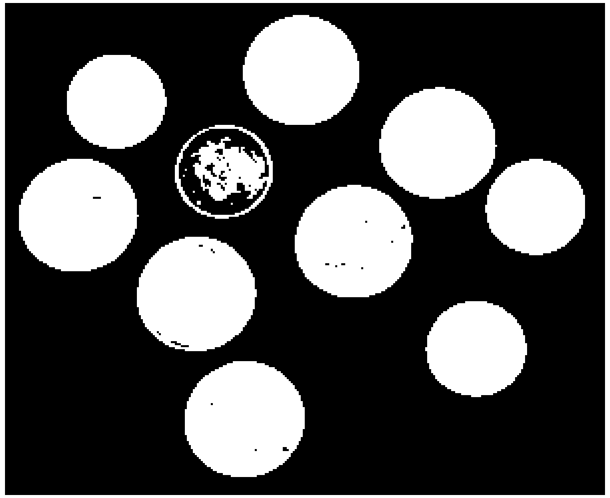
$(A \oplus B) - A$



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

B is the structuring element, a disk of size 5.

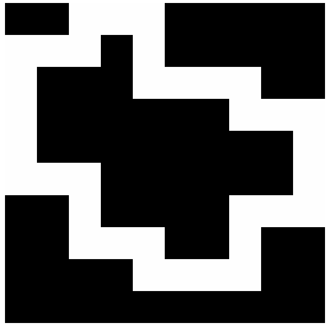
Region filling



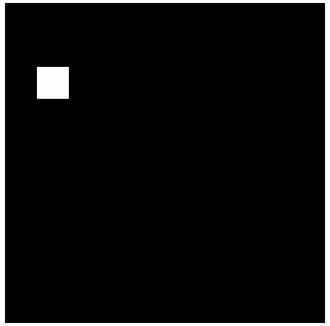
- Thresholding procedures are typically used to generate BW images.
- These images are rarely perfect and in many cases they discontinuities in the form of holes

Region filling

- Design a dilation based method that can fill the hole inside the structure shown in the image below. Start with using a seed pixel inside the region.



A



x_0

We need a function that dilates x_0 without crossing the boundary

Region filling

- Let x_0 be an image that has a 'seed' pixel that is set to a value of 1. The location of the seed pixel is inside the region to be filled. Start Dilating:

$$\boxed{\begin{array}{l}\text{Do } x_k = (x_{k-1} \oplus B)\ \&\ \bar{A} \\ \text{Until } x_k = x_{k-1}\end{array}}$$

This term grows the seed

This term makes sure we are not growing outside the hole

Region filling

Original Image | One Region Filled | All Regions Filled

Hit-or-miss transformation $A \otimes B$

- The hit-and-miss operation is performed in much the same way as other morphological operators, by translating the origin of the structuring element to all points in the image.
- The procedure compares the structuring element with the underlying image pixels. If the foreground and background pixels in the structuring element exactly match foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color.
- If it doesn't match, then that pixel is set to the background color.
- The equation $A \otimes B$ is used to denote applying the hit-or-miss transform to image A using the structuring element A.

Hit-or-miss transformation

- A structuring element and its rotations can be used to detect corners.

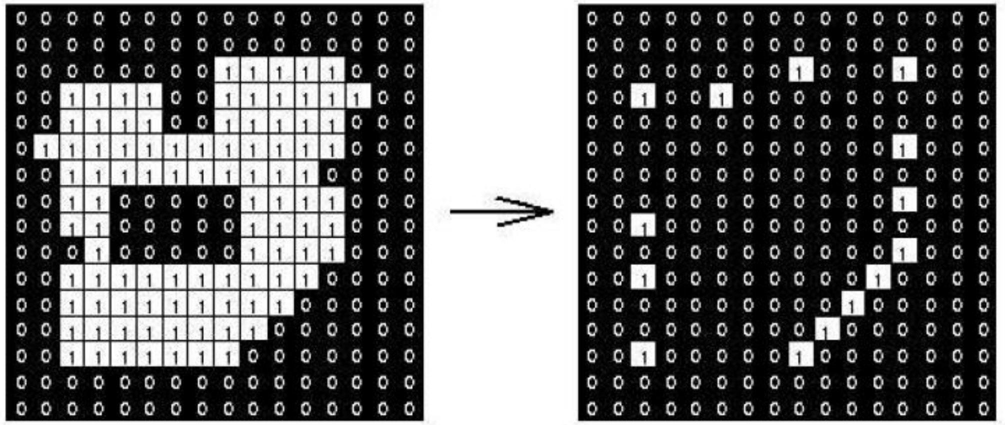
	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

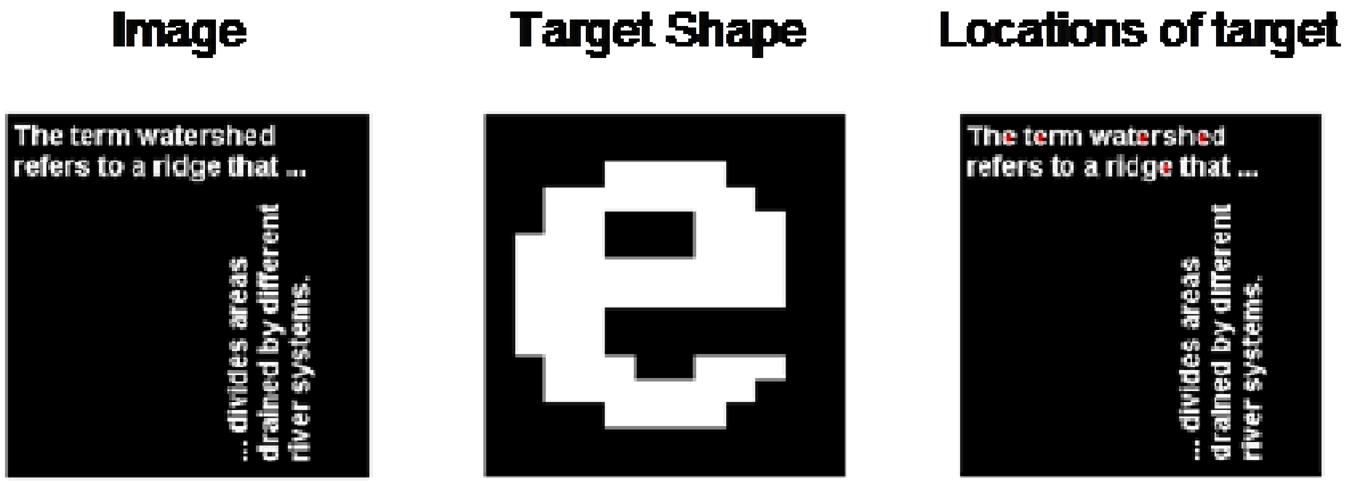
	0	0
1	1	0
	1	

0	0	
0	1	1
	1	

Don’t cares



Hit-or-miss transformation



Hit-or-miss transformation used to detect the letter e. Target shape is of the same size and font used in the text

A hit-or-miss transform $A \otimes B$ finds the structuring element B in A

Thinning

Structure to be removed

$$\text{Thin}(A,B) = A \cap (\overline{A \otimes B})$$

Image

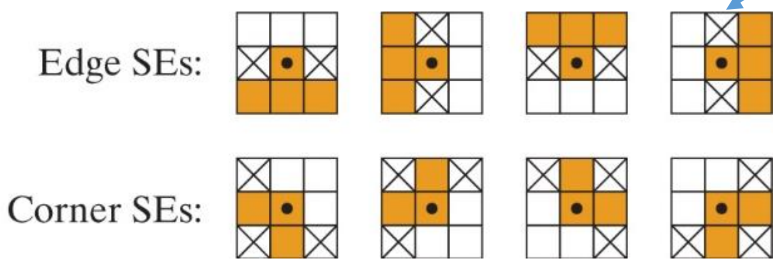
What should the structuring element B represent ?

Repeated application of the thinning operation generates the ‘skeleton’ of an object

Thinning

$$\text{Thin}(A,B) = A \cap (\overline{A \otimes B})$$

X's are don't cares



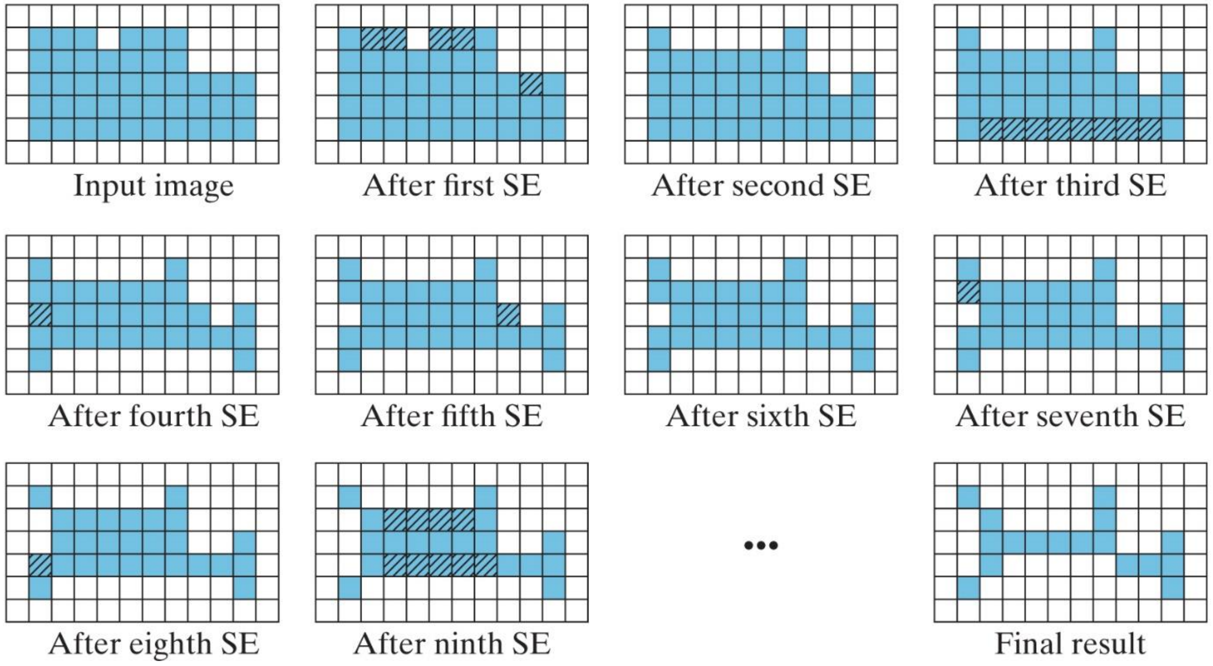
An example of commonly used structuring elements. The first row detects edges, while the second detects corners.

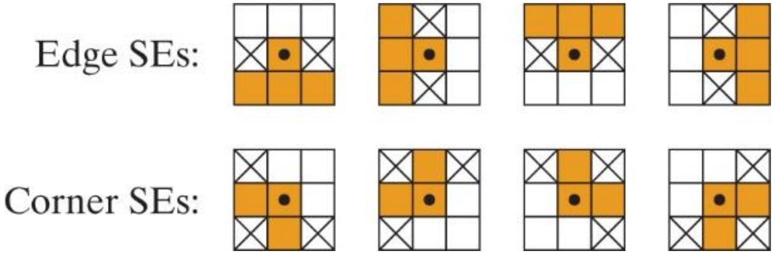
- Thinning works be consecutive application of such structuring elements until convergence (i.e. the process of applying thinning operations has no further effects on the image).
- Note that the central pixel is equal to 1 in all structuring elements.

Thinning

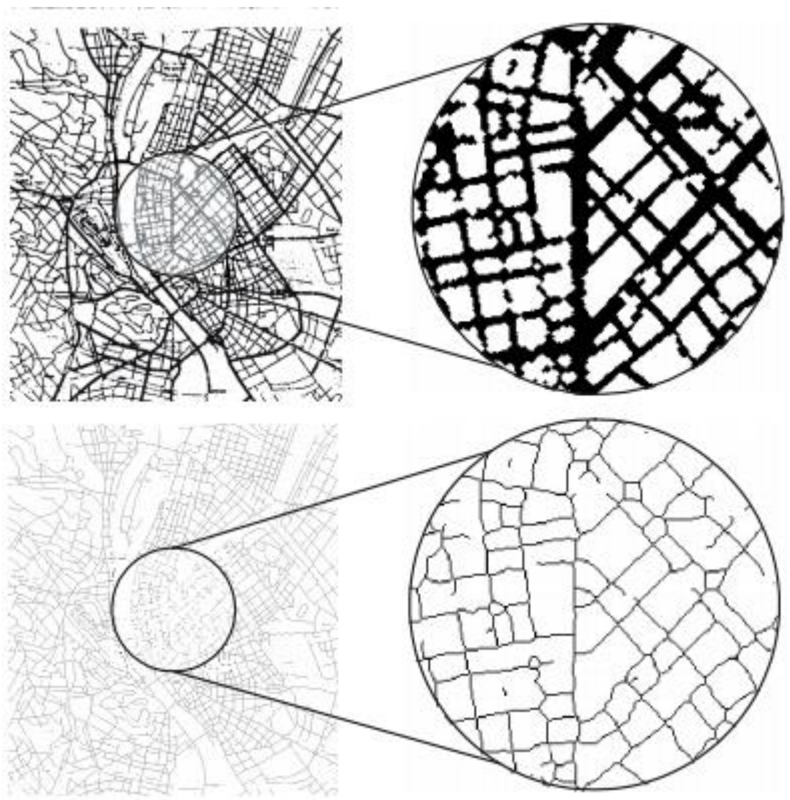
$$\text{Thin(A,B)} = A \cap (\overline{A \otimes B})$$

Figure 4.15 Morphological thinning of a binary image using the SEs of Figure 4.14 treated as a sequence. The first SE matches pixels (indicated by hashed squares) along the top of the region, which are then removed. The second SE matches no pixels, while the third SE matches pixels along the bottom, which are then removed. This process continues until convergence, yielding an approximate skeleton of the original image.





Thinning Example



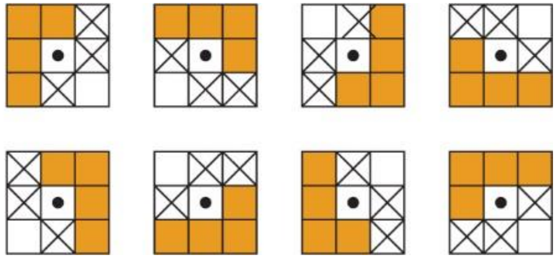
Thinning city map

Thickening

- Thickening a binary image involves adding pixels to the foreground while maintaining as much as possible the overall shape of the foreground regions.

$$\text{Thicken(A,B)} = A \cup (A \otimes B)$$

Figure 4.17 Structuring elements commonly used for morphological thickening.



- Note that the central pixel is equal to 0 in all thickening structuring elements.

Thickening as reverse thinning

Thickening can be implemented as a reverse thinning operation as follows:

1. Take the complement of the image.
2. Thin the complement.
3. Complement the result obtained in step 2.
4. Remove disconnected pixels.

a b
c d
e

FIGURE 9.24
(a) Set A.
(b) Complement of A.
(c) Result of thinning the complement.
(d) Thickened set obtained by complementing (c).
(e) Final result, with no disconnected points.