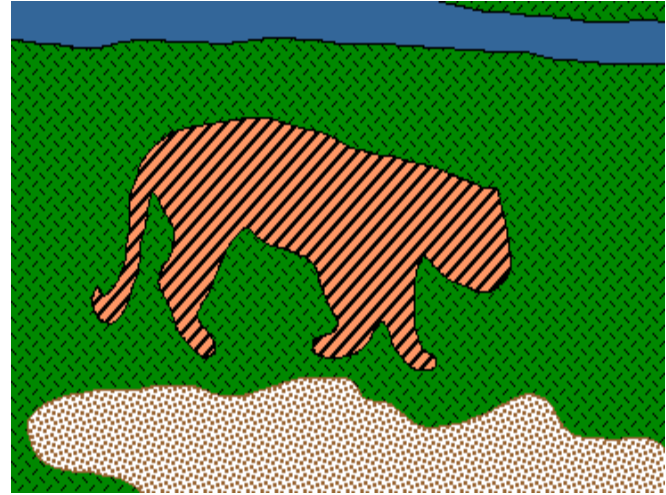


Goal: Extract “Blobs”



What are “blobs”?

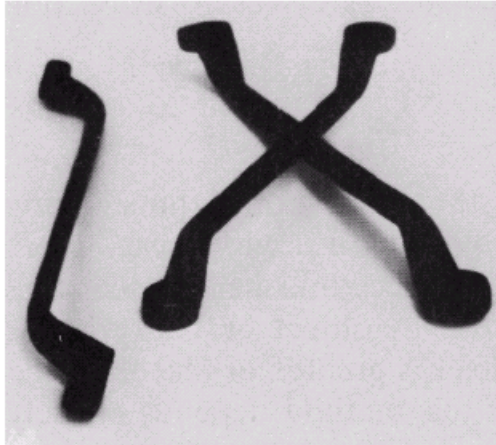
- Regions of an image that are somehow coherent

Why?

- Object extraction, object removal, compositing, etc.
- ...but are “blobs” objects?
- No, not in general

Blob's coherence

Simplest way to define blob coherence is as similarity in brightness or color:



The tools become blobs



The house, grass, and sky make different blobs

Thresholding

Basic segmentation operation:

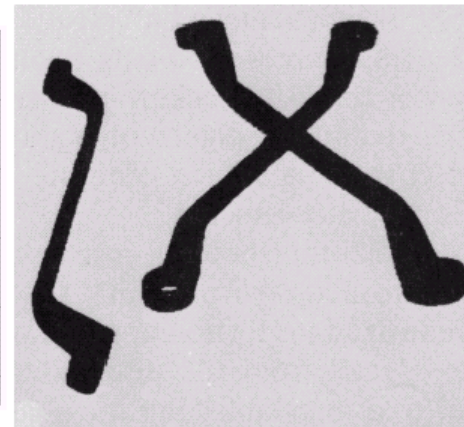
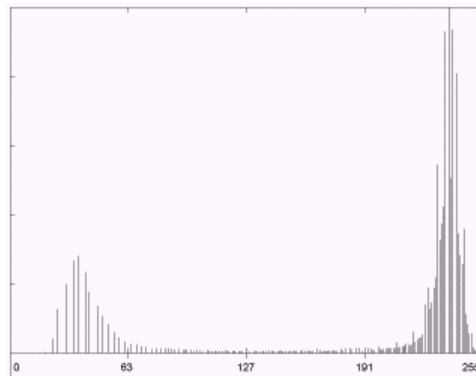
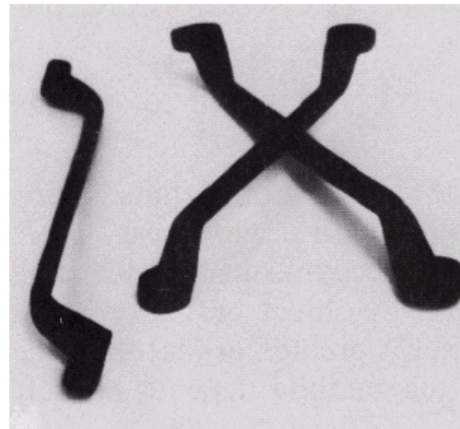
$$\text{mask}(x,y) = 1 \text{ if } \text{im}(x,y) > T$$

$$\text{mask}(x,y) = 0 \text{ if } \text{im}(x,y) < T$$

T is threshold

- User-defined
- Or automatic

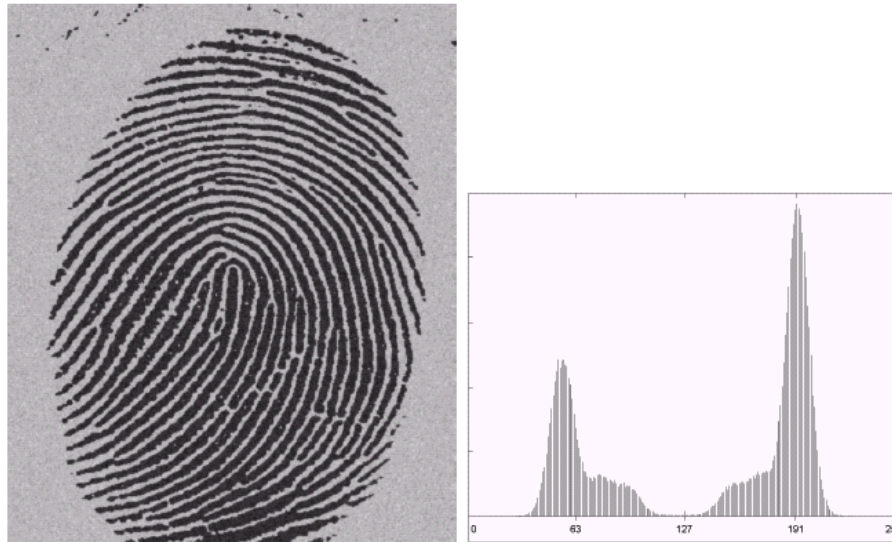
Same as
histogram
partitioning:



a
b c

FIGURE 10.28
(a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

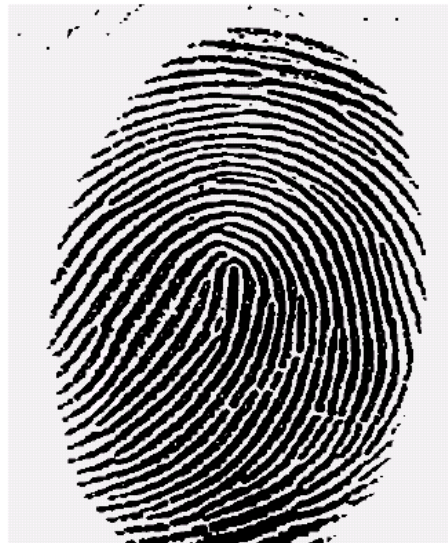
Sometimes works well...



a b
c

FIGURE 10.29

(a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)



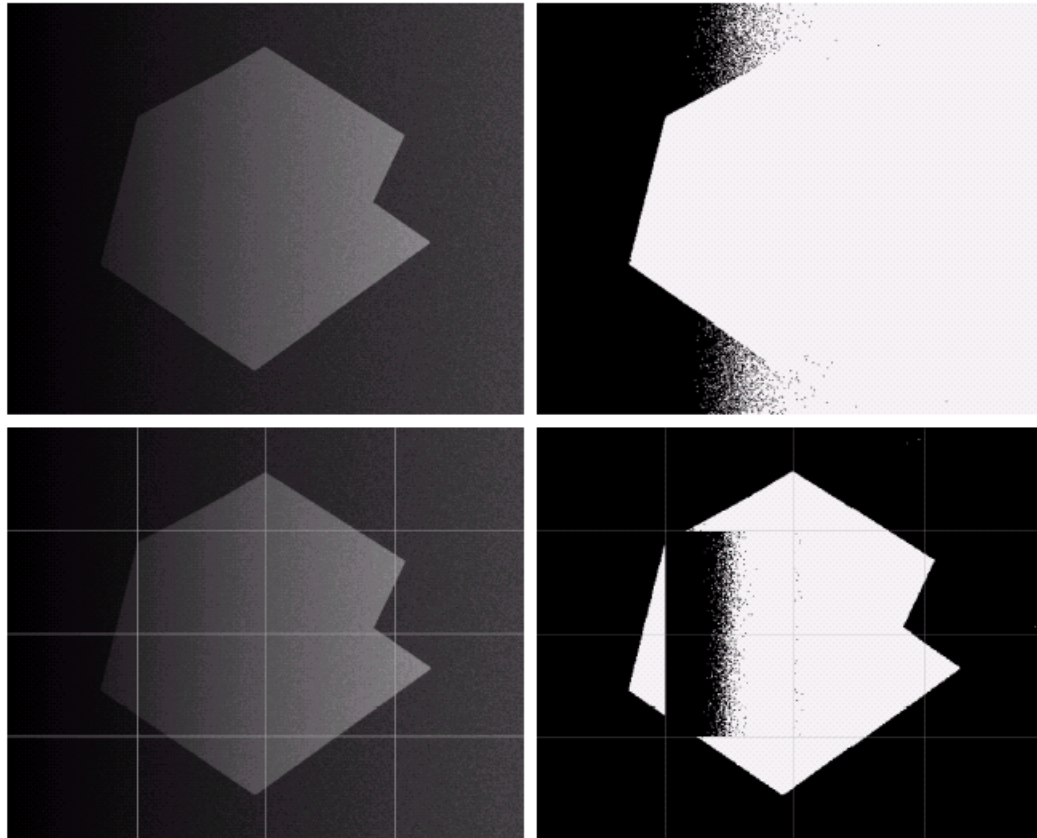
What are potential Problems?

...but more often not

a b
c d

FIGURE 10.30

(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



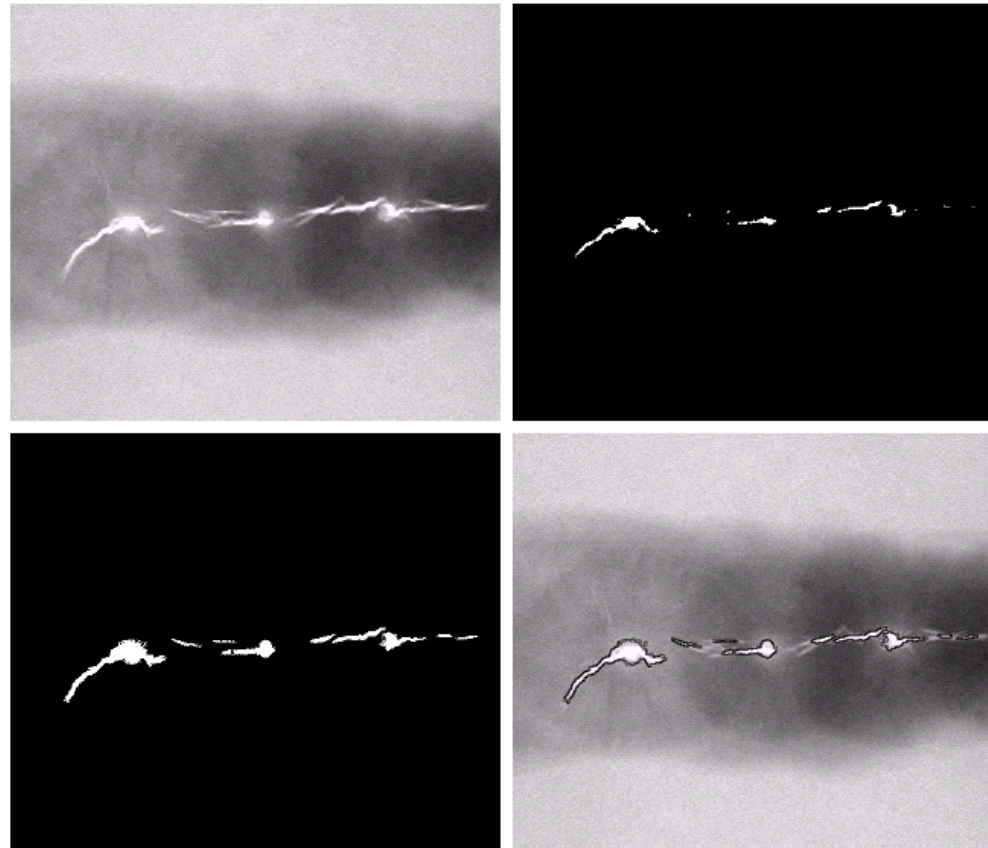
Adaptive thresholding

Region growing

a b
c d

FIGURE 10.40

(a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).



- Start with initial set of pixels K
- Add to K any neighbors, if they are within similarity threshold
- Repeat until nothing changes

Is this same as global threshold?

What can go wrong?

Issues:

Although often useful, all these approaches work only some of the time, and are considered rather “hacky”.

Can't even handle our tiger:



Problem is that blobs \neq objects!

Binary Image Processing

Binary images are handy in many cases (sprite extraction, compositing, etc).

Binary image processing is a well-studied field, based on set theory, called Mathematical Morphology

consists of a set of image analysis operations that are used to produce or process binary images, usually images of 0's and 1's.

0 represents the background

1 represents the foreground

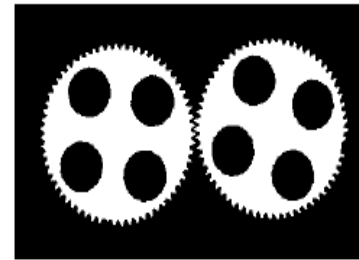
| | | | |
|-----|-----|------|------|
| 000 | 100 | 1000 | 1000 |
| 000 | 111 | 1000 | 1000 |
| 000 | 100 | 1000 | 1000 |

Application Areas

Document Analysis



Industrial Inspection



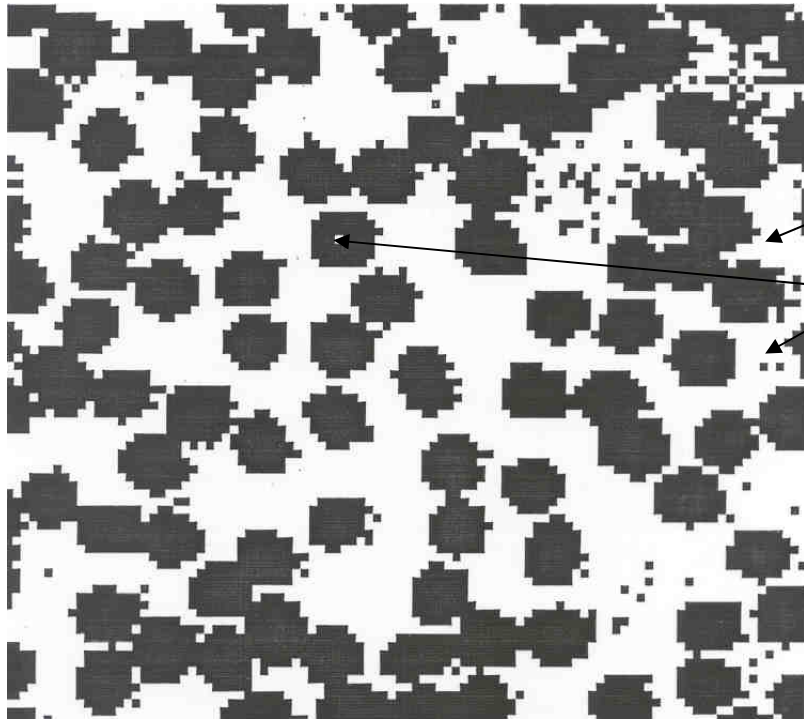
Medical Imaging



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 to each other – bad!
- Salt and pepper noise from thresholding
- How usable is this data?

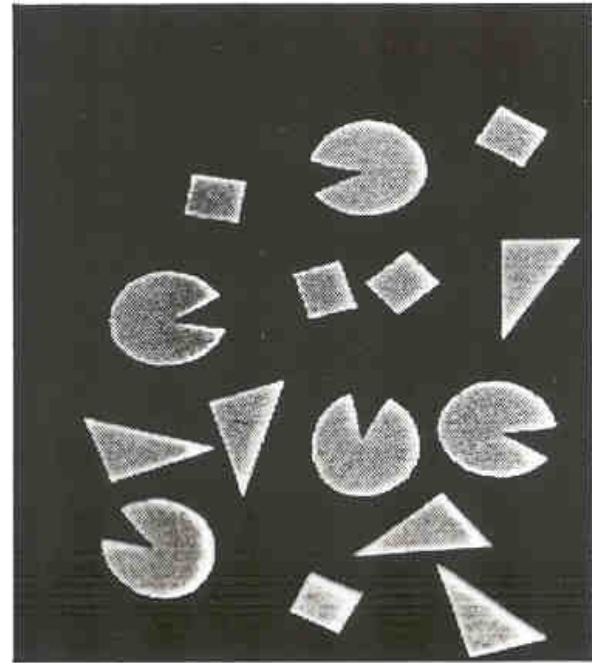
- 63 separate objects detected
- Single cells have area about 50
- Noise spots
- Gobs of cells

More controlled images

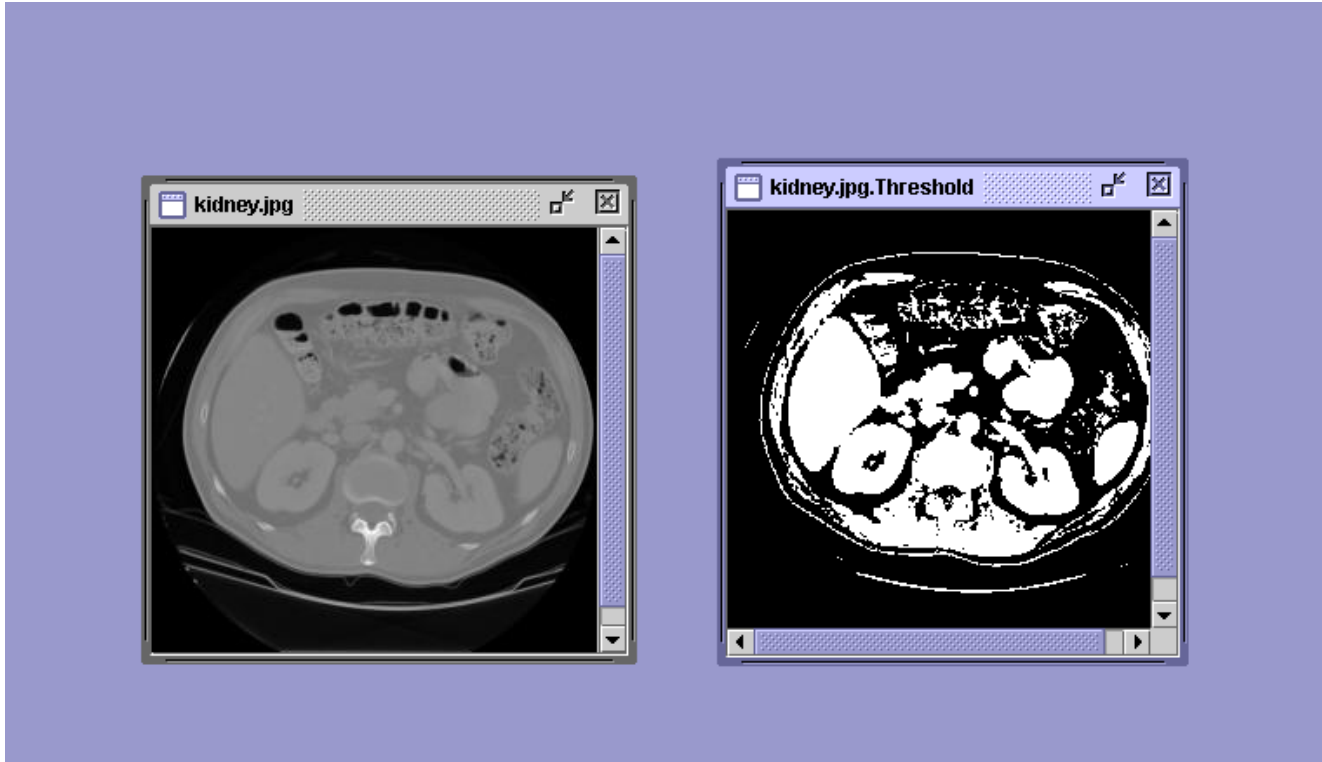
- More uniform objects
- More uniform background
- Objects actually separated

- 15 objects detected
- Location known
- Area known
- 3 distinct clusters of 5 values of area; 85, 145, 293

Results on Coloring Pacmen



Thresholding



Binary images can be obtained from gray level images by thresholding

Cleaning up thresholding results



- Delete object pixels on boundary to better separate parts.
- Fill small holes
- Delete tiny objects
- (last 2 are “salt-and-pepper” noise)

Mathematical Morphology

Binary mathematical morphology consists of two basic operations

dilation and erosion

and several composite relations

closing and opening
conditional dilation

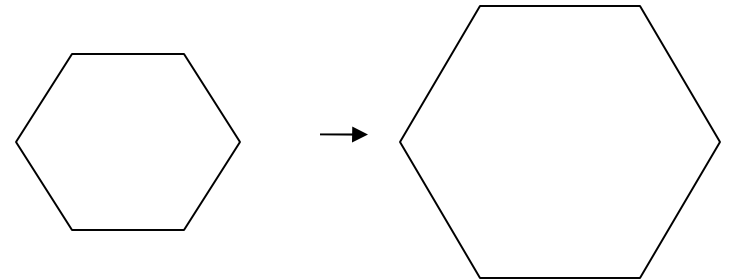
...

Dilation

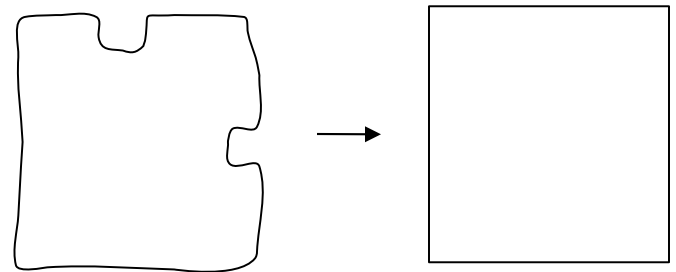
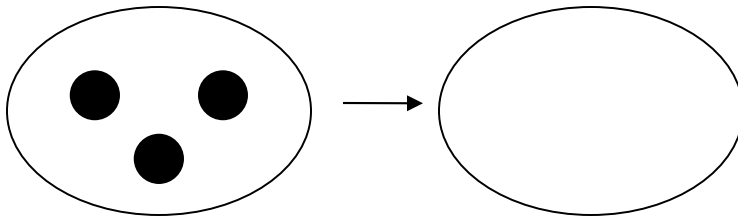
Dilation **expands** the connected sets of 1s of a binary image.

It can be used for

1. growing features



2. filling holes and gaps

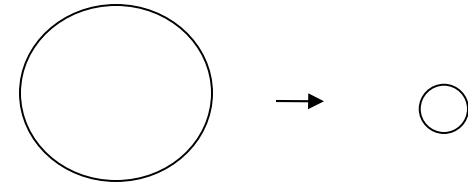
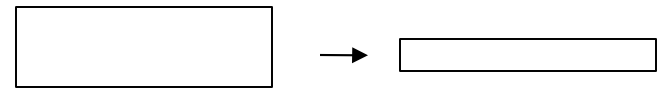


Erosion

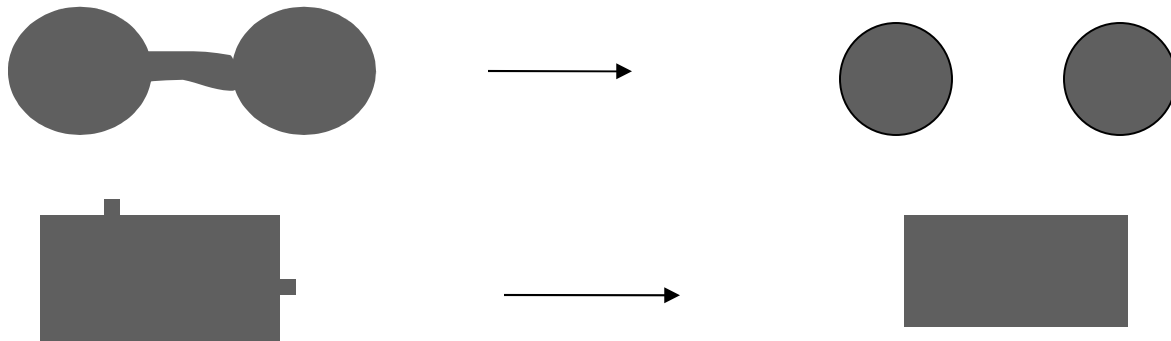
Erosion **shrinks** the connected sets of 1s of a binary image.

It can be used for

1. shrinking features



2. Removing bridges, branches and small protrusions



Structuring Elements

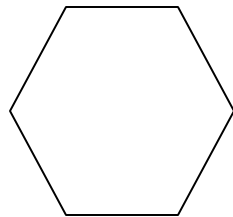
A **structuring element** is a shape mask used in the basic morphological operations.

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

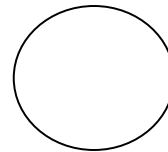


box

box(length,width)

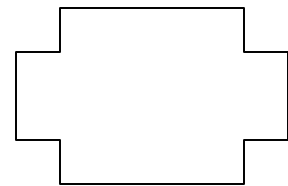


hexagon



disk

disk(diameter)



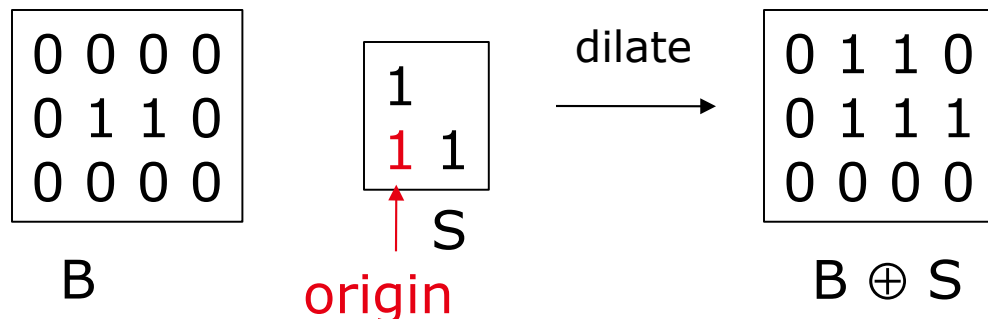
something

Dilation with structuring elements

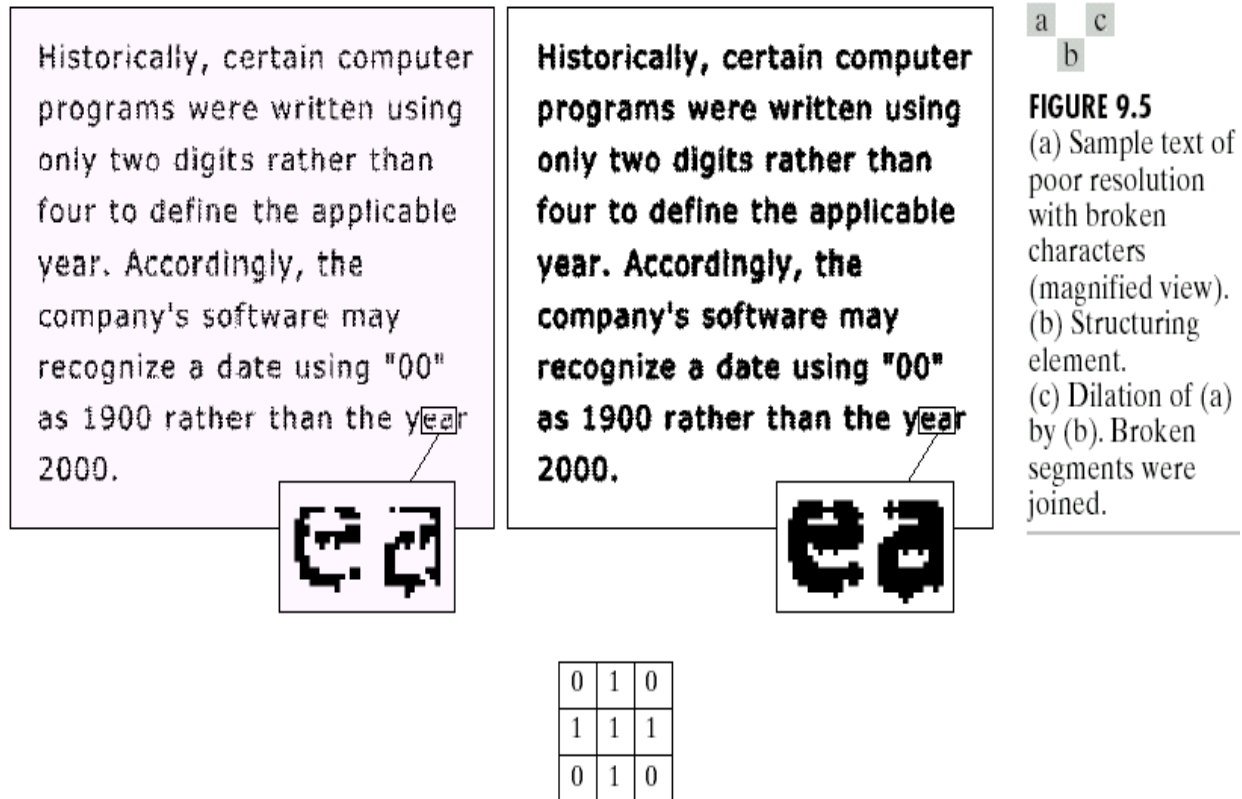
The arguments to dilation and erosion are

1. a binary image **B**
2. a structuring element **S**

`dilate(B,S)` takes binary image **B**, places the origin of structuring element **S** over each 1-pixel, and ORs the structuring element **S** into the output image at the corresponding position.

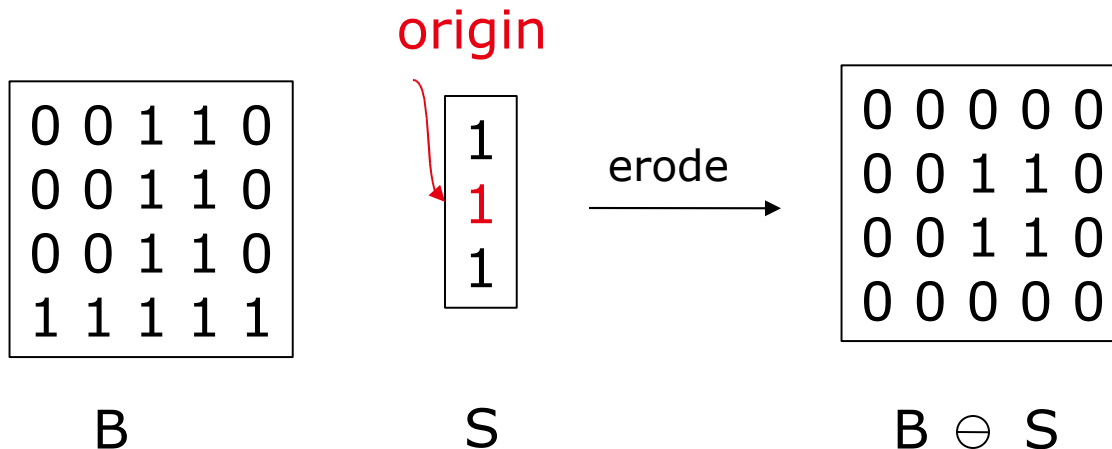


Dilation



Erosion with structuring elements

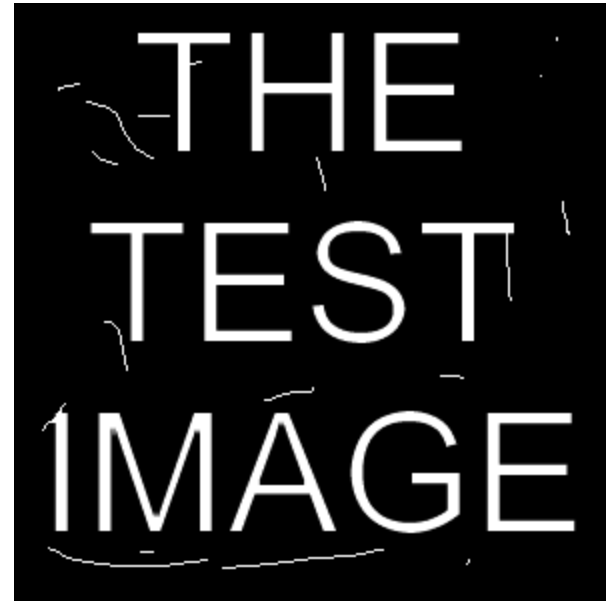
`erode(B,S)` takes a binary image B, places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image only if every position of S (with a 1) covers a 1 in B.



Erosion



Original image



Eroded image

Erosion



Eroded once



Eroded twice

Opening and Closing

- **Closing** is the compound operation of dilation followed by erosion (with the same structuring element)
 - **Opening** is the compound operation of erosion followed by dilation (with the same structuring element)
-

Opening and Closing

Opening : smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions

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

Closing : smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour

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

Prove to yourself that they are not the same thing. Play around with `bwmorph` in Matlab.

Example 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 | | | | | |
| | | | | | | | |

a) Binary image B

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

b) Structuring Element S

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 | 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 | | | |

c) Dilation $B \oplus S$

| | | | | | | | |
|--|---|---|---|---|---|---|--|
| | | | | | | | |
| | 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 | | | | |
| | | | | | | | |

e) Closing $B \bullet S$

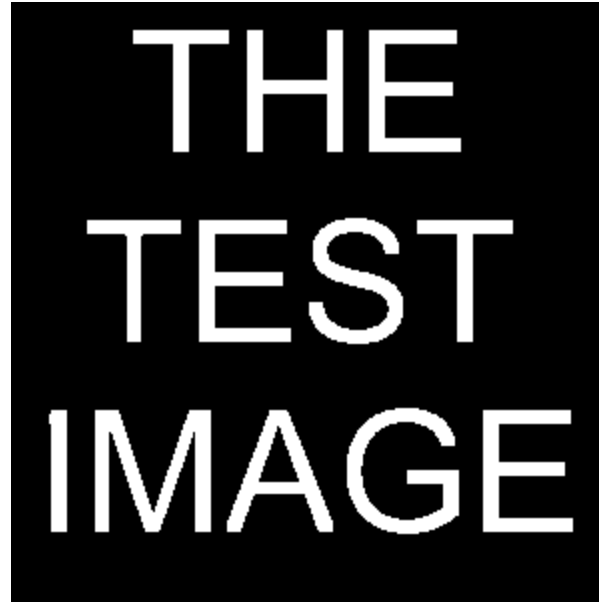
| | | | | | | | |
|--|--|--|---|---|--|--|--|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | 1 | 1 | | | |
| | | | 1 | 1 | | | |
| | | | 1 | 1 | | | |
| | | | | | | | |
| | | | | | | | |

d) Erosion $B \ominus S$

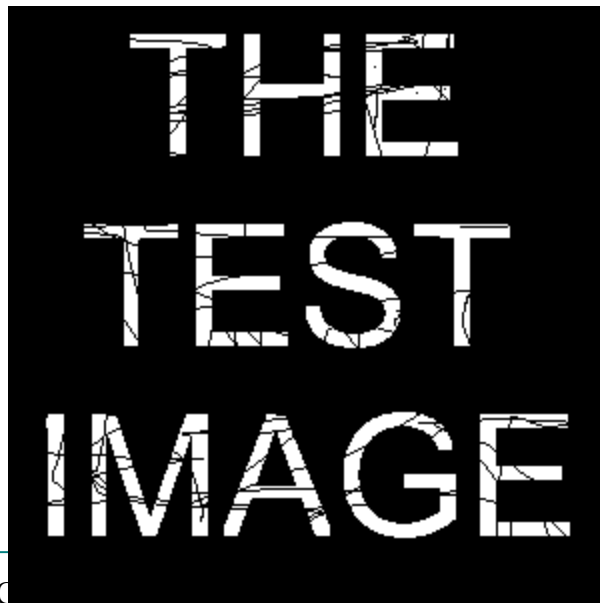
| | | | | | | | |
|--|--|--|---|---|---|---|--|
| | | | | | | | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | 1 | 1 | 1 | 1 | |
| | | | | | | | |
| | | | | | | | |

f) Opening $B \circ S$

Opening and Closing



OPENING: The original image eroded twice and dilated twice (opened). Most noise is removed

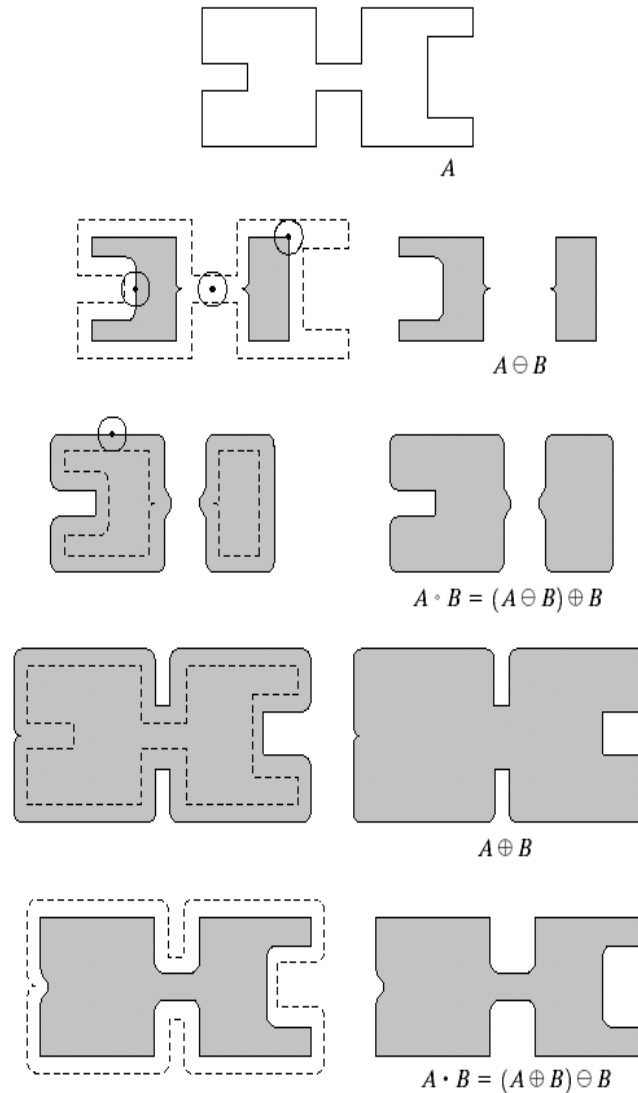


CLOSING: The original image dilated and then eroded. Most holes are filled.

Opening and Closing

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 dark dot is the center of the structuring element.

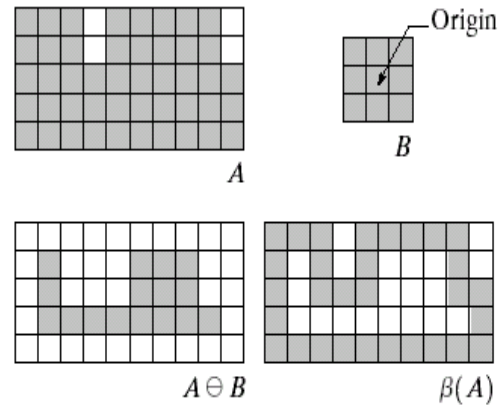


Boundary Extraction

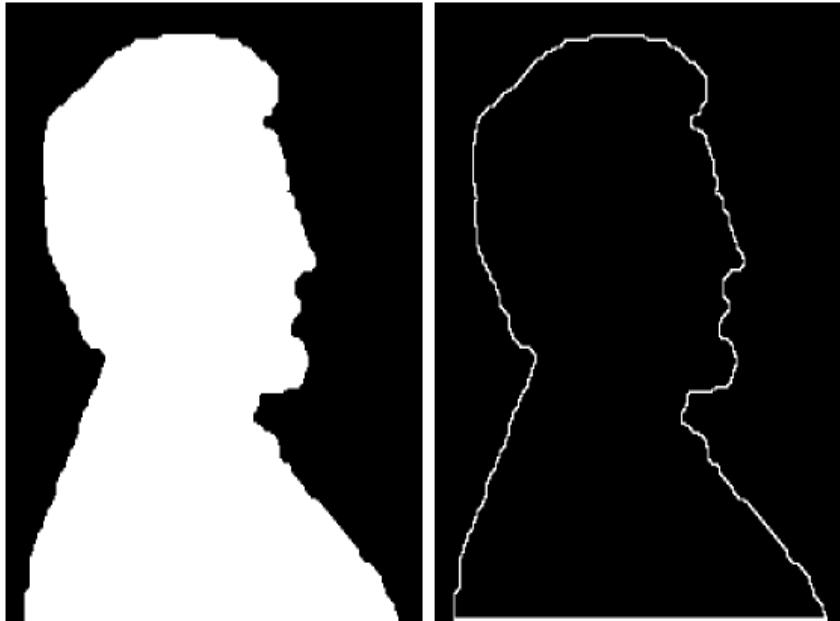
$$\beta(A) = A - (A \ominus B)$$

| | |
|---|---|
| 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



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).

Region Filling

| | | |
|---|---|---|
| a | b | c |
| d | e | f |
| g | h | i |

FIGURE 9.15

Region filling.

(a) Set A .

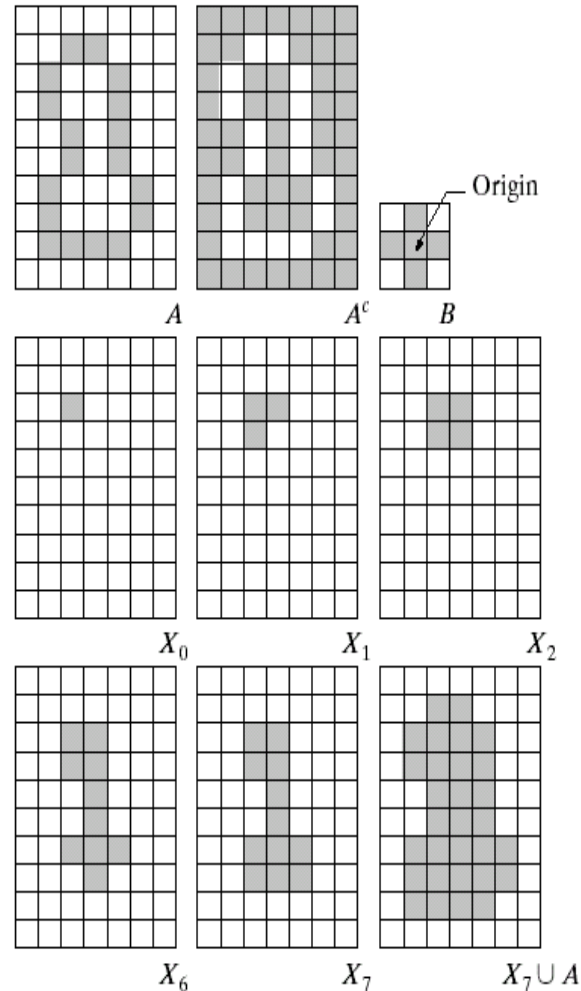
(b) Complement
of A .

(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)].

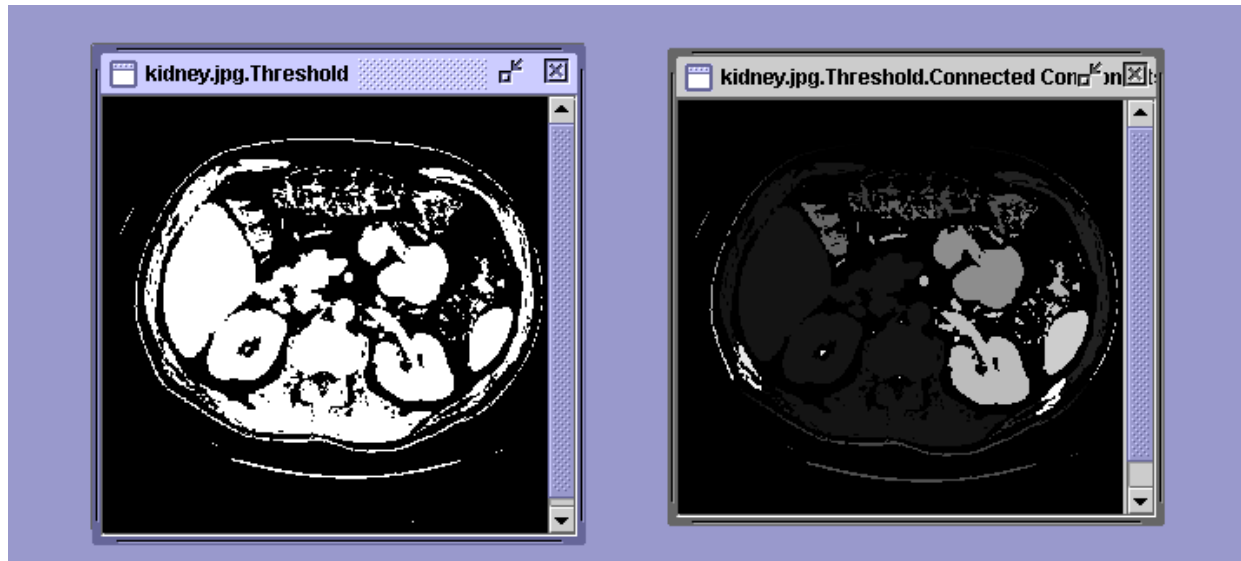


$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$

Connected Component Labeling

Once you have a binary image, you can identify and then analyze each **connected set of pixels**.

The connected components operation takes in a binary image and produces a **labeled image** in which each pixel has the integer label of either the background (0) or a component.



Extraction of Connected Components

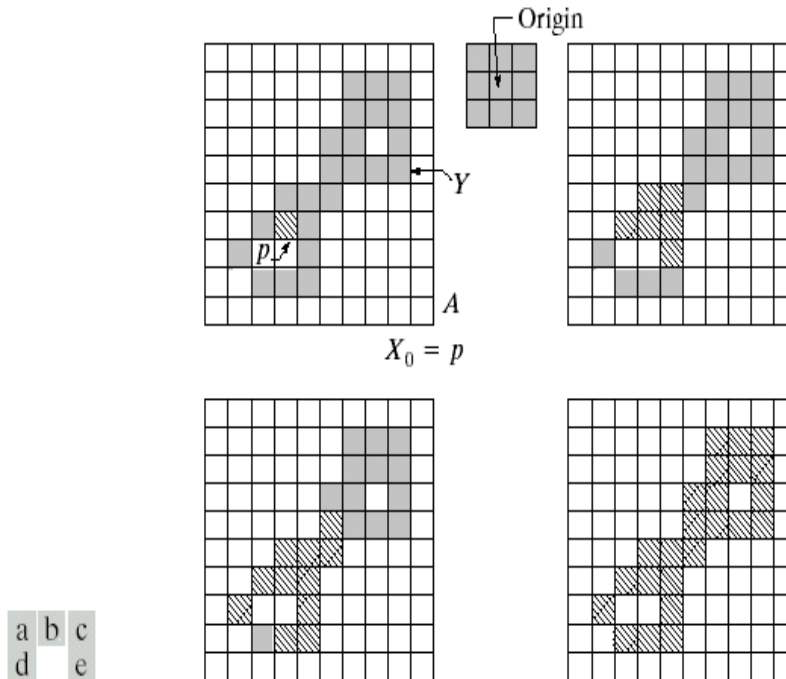
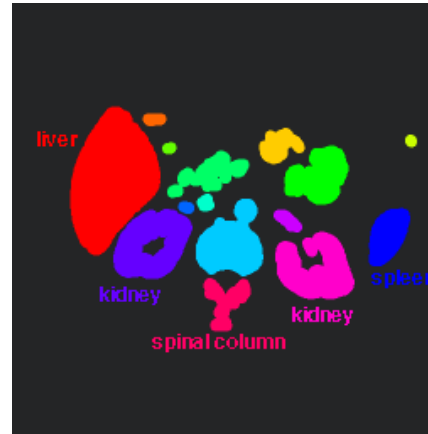
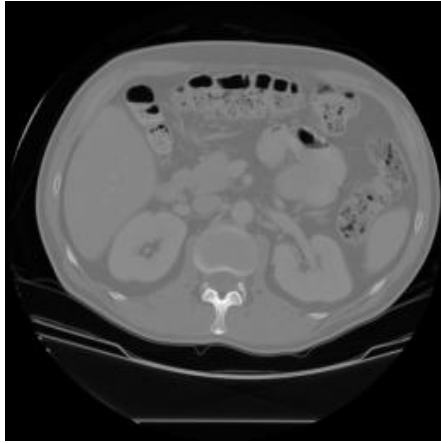


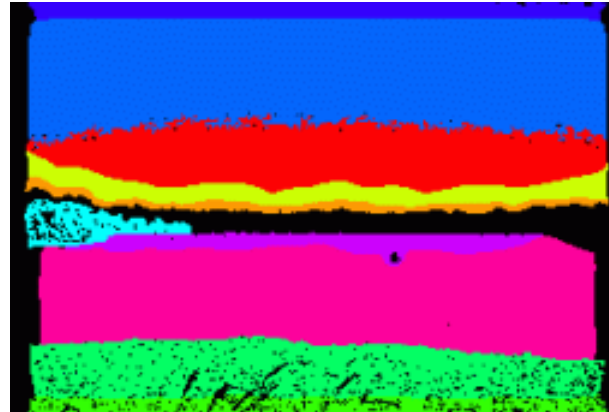
FIGURE 9.17 (a) Set A showing initial point p (all shaded points are valued 1, but are shown different from p to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots$$

Labeling shown as pseudo-color



connected
components
of 1's from
thresholded
image



connected
components
of cluster
labels
