

---

# Lecture 16

## Image Segmentation

1. The basic concepts of segmentation
2. Point, line, edge detection
3. Thresh holding
4. Region-based segmentation
5. Segmentation with Matlab

# What is segmentation

---

- What is segmentation
  - Segmentation subdivides an image into its constituent regions or objects, until the objects of interest in an application have been isolated.

- Segmentation conditions

$R_0, \dots, R_n$

$$(a) \bigcup_{i=0}^n R_i = R$$

(b)  $R_i$  is a connected set  $i = 1, 2, \dots, n$

(c)  $R_i \cap R_j = \emptyset$  for all  $i$  and  $j$ ,  $i \neq j$

(d)  $Q(R_i) = \text{true}$  for  $i = 1, 2, \dots, n$

(e)  $Q(R_i \cup R_j) = \text{false}$  for any adjacent regions  $R_i$  and  $R_j$

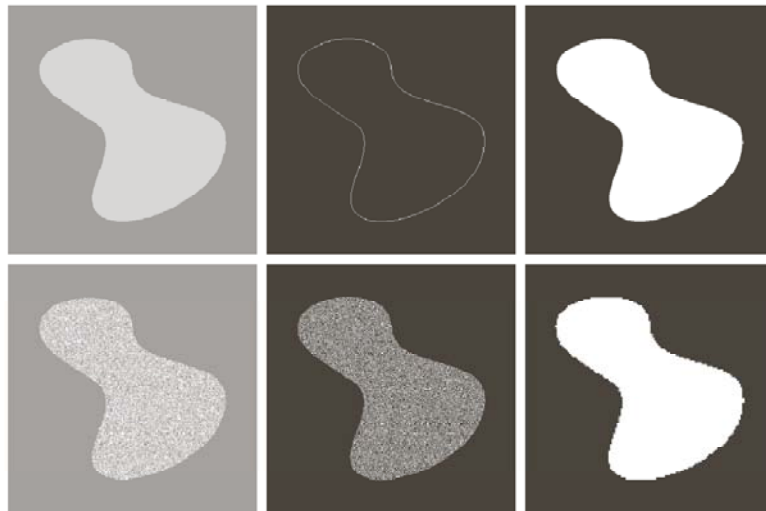
where  $Q(R_i)$  is a logical predicate defined over the points in set  $R_k$ .

- Segmentation problem: to partition the image into regions satisfying above conditions

# Two principal approaches

---

- Edge-based segmentation
  - partition an image based on abrupt changes in intensity (edges)
- Region-based segmentation
  - partition an image into regions that are similar according to a set of predefined criteria.



**FIGURE 10.1** (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

a	b	c
d	e	f

# Detection of Discontinuities

---

- Detect the three basic types of gray-level discontinuities
  - points , lines , edges
- Use the image sharpening techniques
  - The first order derivatives produce thicker edges
  - The second order derives have a strong response to fine detail, such as thin lines and isolated points, and noise
  - Laplasian operation
- Can be done by running a mask through the image

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$
$$= \sum_{k=1}^9 w_k z_k$$

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

# Point Detection

---

Steps for point detection

1. Apply Laplacian filter to the image to obtain  $R(x, y)$
2. Create binary image by threshold

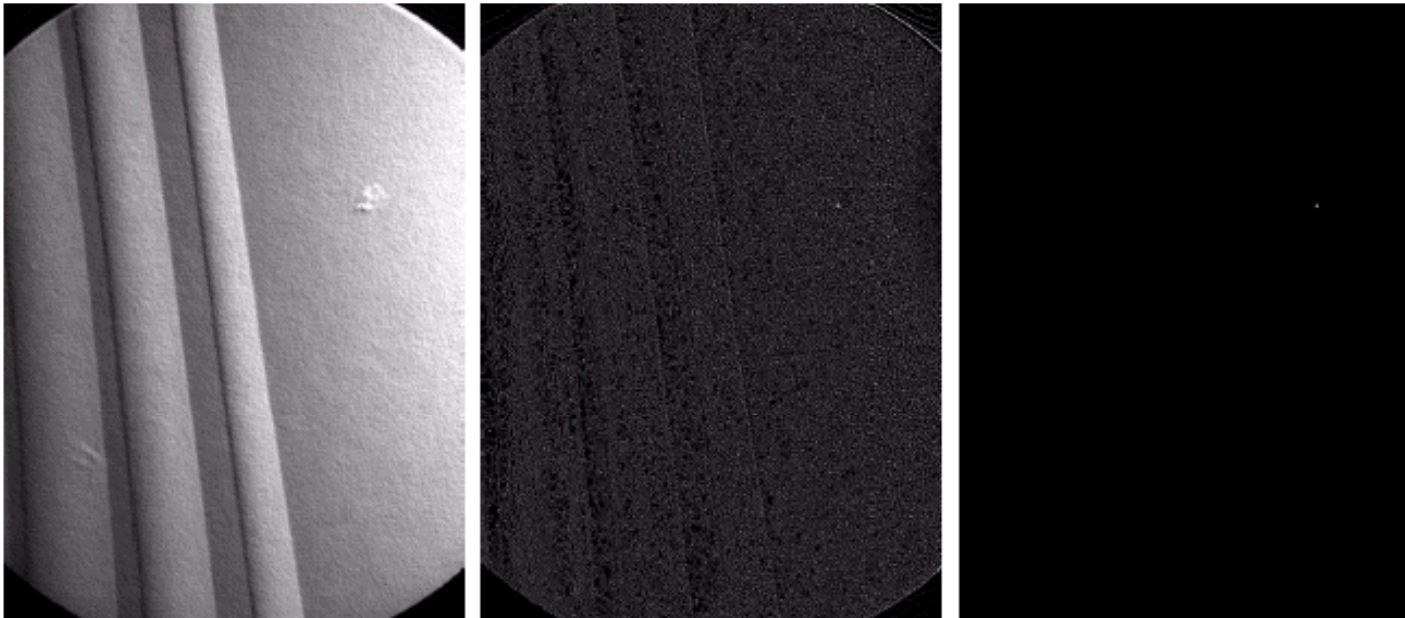
$$g(x, y) = \begin{cases} 1, & \text{if } |R(x, y)| \geq T \\ 0, & \text{otherwise} \end{cases}$$

where  $T$  is a nonnegative threshold

-1	-1	-1
-1	8	-1
-1	-1	-1

# Example

---



-1	-1	-1
-1	8	-1
-1	-1	-1

a  
b c d

**FIGURE 10.2**

(a) Point detection mask.  
(b) X-ray image of a turbine blade with a porosity.  
(c) Result of point detection.  
(d) Result of using Eq. (10.1-2).  
(Original image courtesy of X-TEK Systems Ltd.)

---

# Line Detection

---

- A special mask is needed to detect a special type of line
- Examples:
  - Horizontal mask has high response when a line passed through the middle row of the mask.

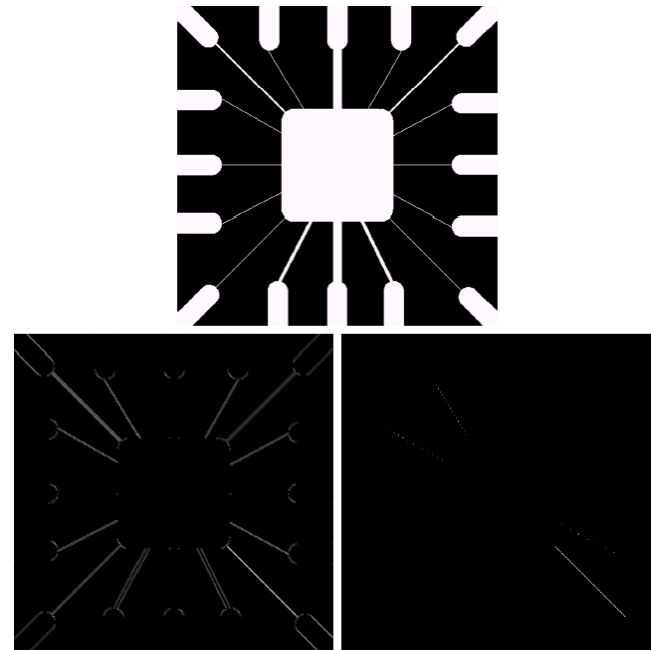
-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+45°			Vertical			-45°		

# Multiple Lines Detection

- Apply every masks on the image, find the maximum response.

Example:

Let  $R_1, R_2, R_3, R_4$  denotes the response of the horizontal, +45 degree, vertical and -45 degree masks, respectively. if, at a certain point in the image  $|R_i| > |R_j|$ , for all  $j \neq i$ , that point is said to be more likely associated with a line in the direction of mask  $i$ .



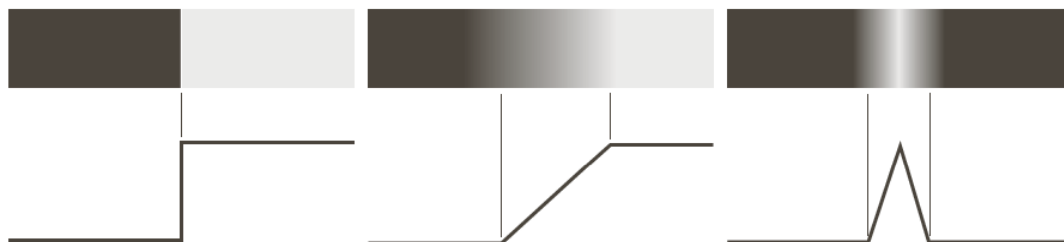
**FIGURE 10.4**  
Illustration of line detection.  
(a) Binary wire-bond mask.  
(b) Absolute value of result after processing with  $-45^\circ$  line detector.  
(c) Result of thresholding image (b).



# Edge Detection

---

- Edge detection is the approach for segmenting images based on abrupt changes in intensity.
- What is an edge
  - an edge is a set of connected pixels that lie on the boundary between two regions.
  - an edge is a “local” concept whereas a region boundary, owing to the way it is defined, is a more global idea.
- Edge models
  - Step edge (Idea edge), ramp edge (thick edge), and roof edge



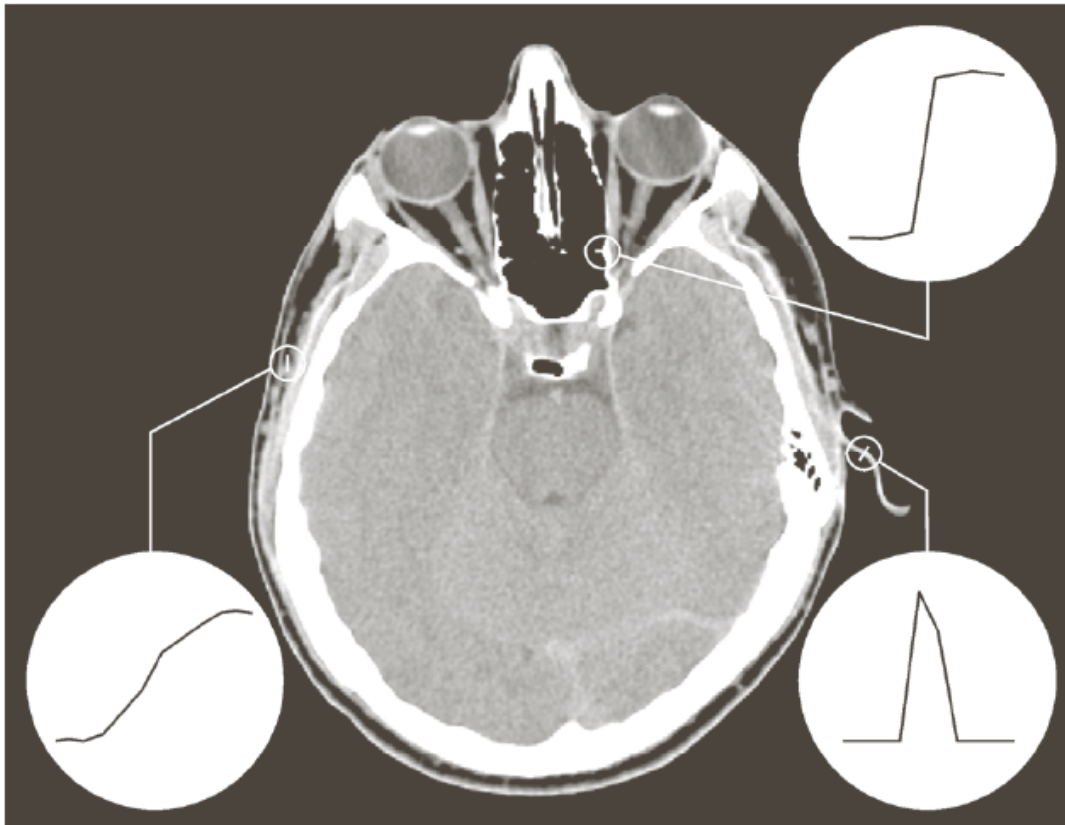
a b c

**FIGURE 10.8**  
From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

# Example of edges in an image

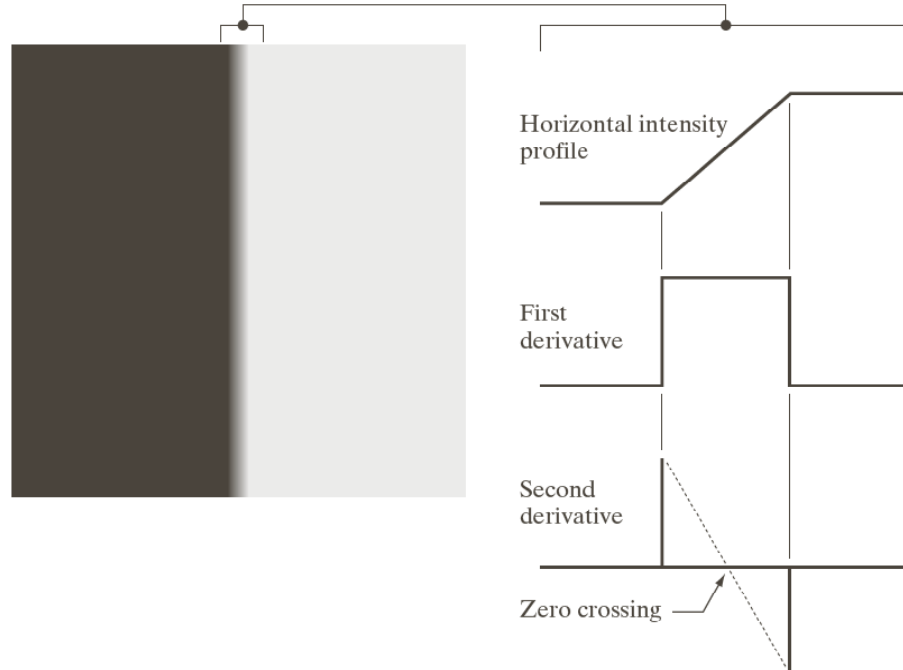
---

- An image may have all the three types of edges



**FIGURE 10.9** A  $1508 \times 1970$  image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and “step” profiles span 9 pixels and 2 pixels, respectively. The base of the roof edge is 3 pixels. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

# First and Second derivatives at the edge



a b

**FIGURE 10.10**

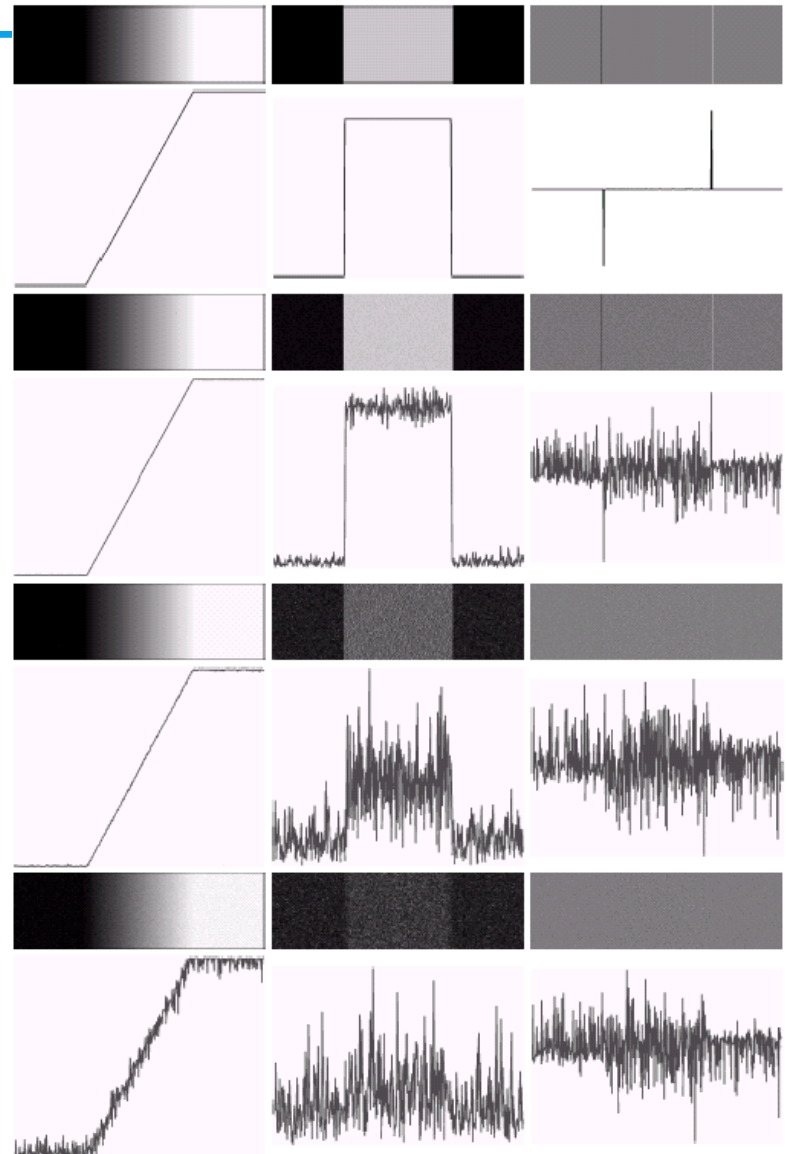
(a) Two regions of constant intensity separated by an ideal vertical ramp edge.  
(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

1. The magnitude of the first derivative can be used to detect the presence of an edge at a point
2. Second derivative produces two values for every edge in an image. An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. **zero-crossing point:** the center of thick edges

# Noise Images

---

- First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and  $\sigma = 0.0$ , 0.1, 1.0 and 10.0, respectively.
- Second column: first-derivative images and gray-level profiles.
- Third column : second-derivative images and gray-level profiles.



# Steps in edge detection

---

1. Image smoothing for noise reduction
2. Detection of edge points. Points on an edge
3. Edge localization

# Image gradient

---

- Gradient is a vector

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- The magnitude of the gradient

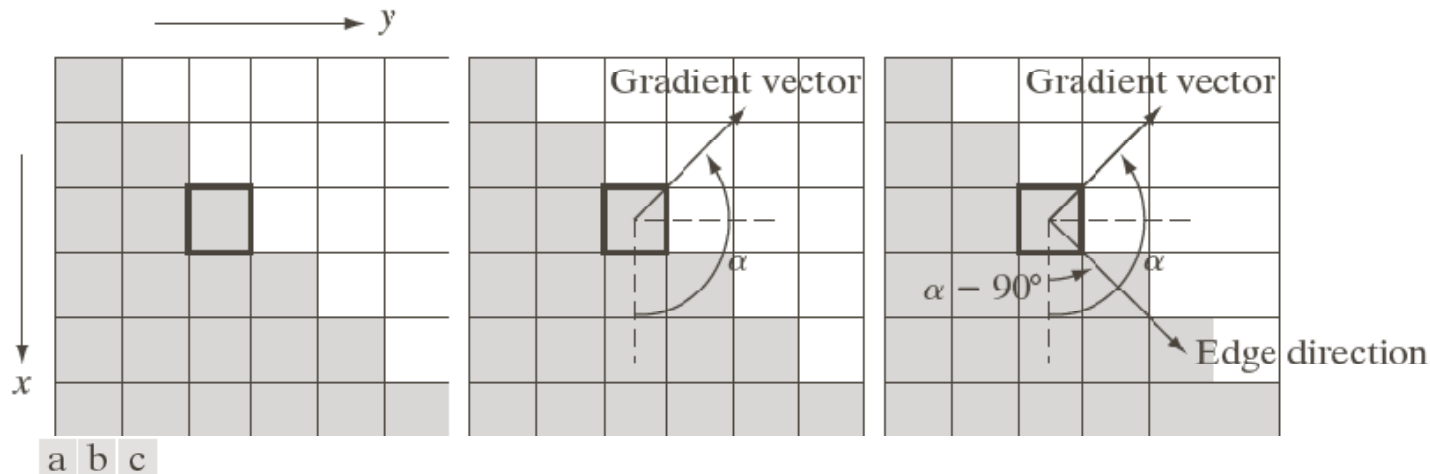
$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- The direction of the gradient vector

$$\alpha(x, y) = \tan^{-1} \left[ \frac{g_y}{g_x} \right]$$

- The direction of an edge at  $(x, y)$  is perpendicular to the direction of the gradient vector at that point

# Example



**FIGURE 10.12** Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, M(x, y) = 2\sqrt{2},$$

$$\alpha(x, y) = \tan^{-1}(-2/2) = -45^\circ$$

# Gradient Operators and Masks

$$g_x = \frac{\partial f}{\partial x} = f(x+1, y) - f(x, y)$$

$$g_y = \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y)$$

Roberts cross-gradient operators

$$g_x = \frac{\partial f(x, y)}{\partial x} = z_9 - z_5$$

$$g_y = \frac{\partial f(x, y)}{\partial y} = z_8 - z_6$$

Prewitt operators

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

Sobel operators

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

a b

**FIGURE 10.13**  
One-dimensional masks used to implement Eqs. (10.2-12) and (10.2-13).

-1
1

-1	1
----	---

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

-1	0	0	-1
0	1	1	0

Roberts

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Prewitt

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Sobel

a  
b c  
d e  
f g

**FIGURE 10.14**  
A  $3 \times 3$  region of an image (the  $z$ 's are intensity values) and various masks used to compute the gradient at the point labeled  $z_5$ .



# Prewitt and Sobel masks for Diagonal edges

---

0	1	1	-1	-1	0
-1	0	1	-1	0	1
-1	-1	0	0	1	1

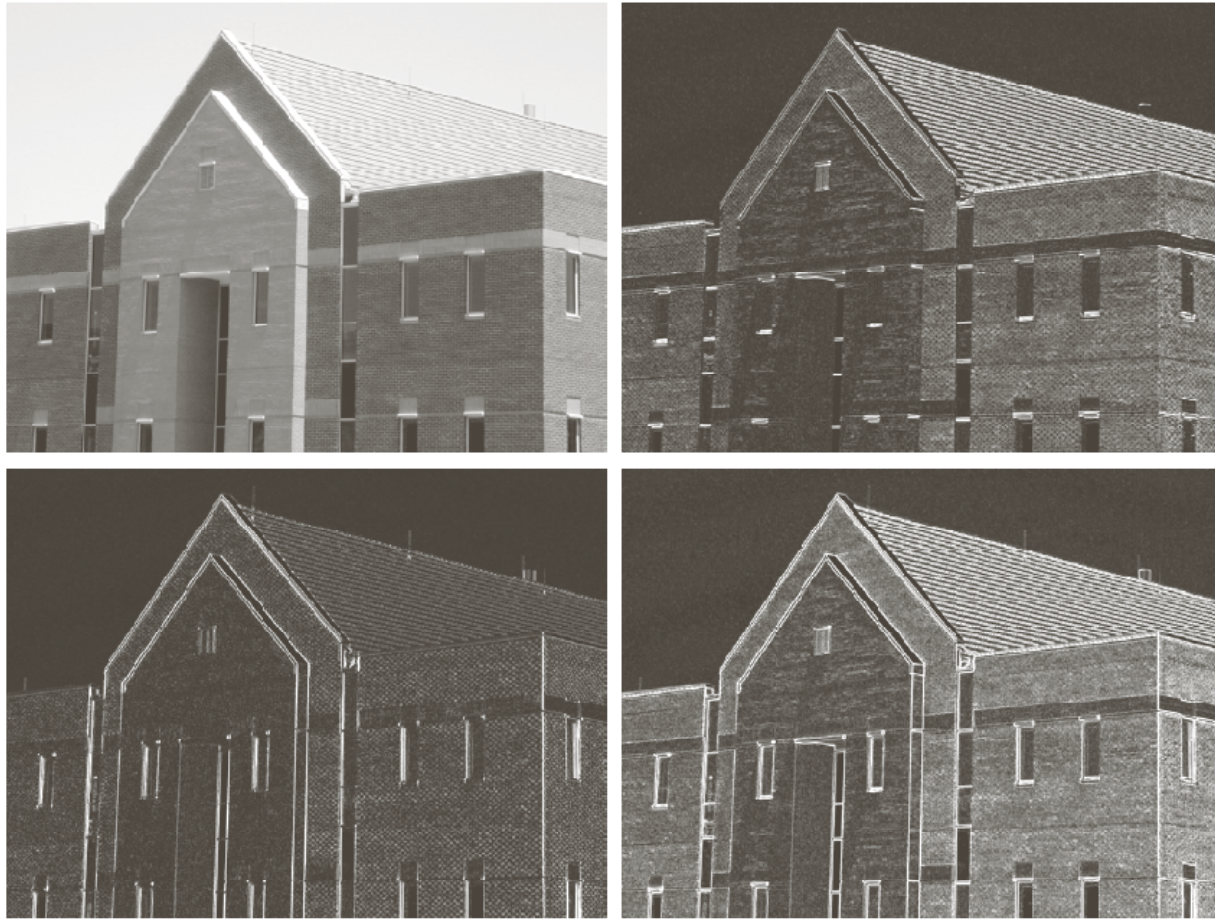
Prewitt

0	1	2	-2	-1	0
-1	0	1	-1	0	1
-2	-1	0	0	1	2

Sobel

Sobel masks have slightly superior noise-suppression characteristics which is an important issue when dealing with derivatives.

# Example



a	b
c	d

**FIGURE 10.16**  
(a) Original image of size  $834 \times 1114$  pixels, with intensity values scaled to the range  $[0, 1]$ .  
(b)  $|g_x|$ , the component of the gradient in the  $x$ -direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image.  
(c)  $|g_y|$ , obtained using the mask in Fig. 10.14(g).  
(d) The gradient image,  $|g_x| + |g_y|$ .

# Example

---

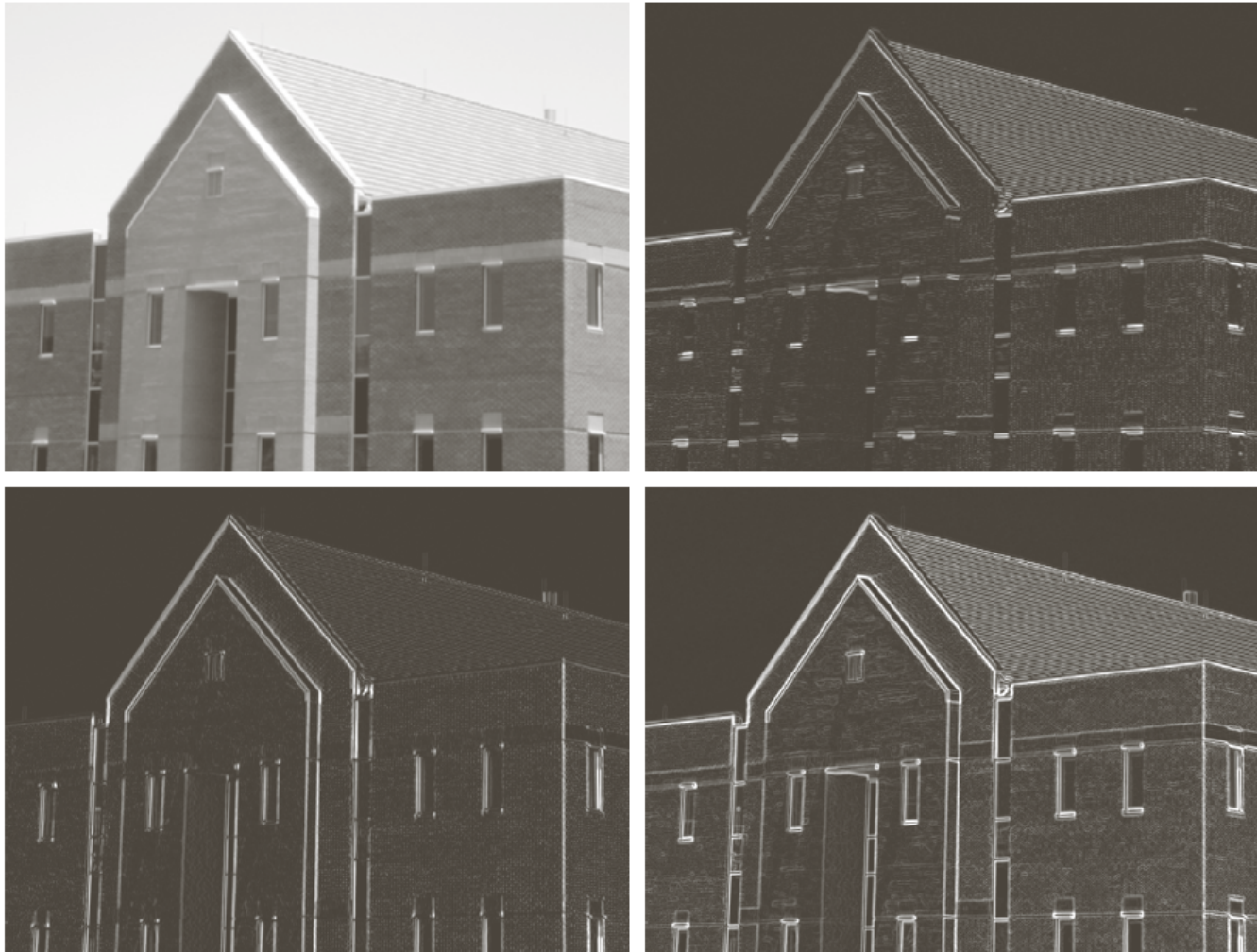


**FIGURE 10.17**  
Gradient angle  
image computed  
using  
Eq. (10.2-11).  
Areas of constant  
intensity in this  
image indicate  
that the direction  
of the gradient  
vector is the same  
at all the pixel  
locations in those  
regions.

---

# Example

---



a b  
c d

**FIGURE 10.18**  
Same sequence as in Fig. 10.16, but with the original image smoothed using a  $5 \times 5$  averaging filter prior to edge detection.



# Example

---



a b

**FIGURE 10.19**

Diagonal edge detection.

(a) Result of using the mask in Fig. 10.15(c).

(b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

# Laplacian

---

Laplacian operator  $\nabla^2 f = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)]$$

**FIGURE 10.13**  
Laplacian masks  
used to  
implement  
Eqs. (10.1-14) and  
(10.1-15),  
respectively.

---

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

# Laplacian of Gaussian

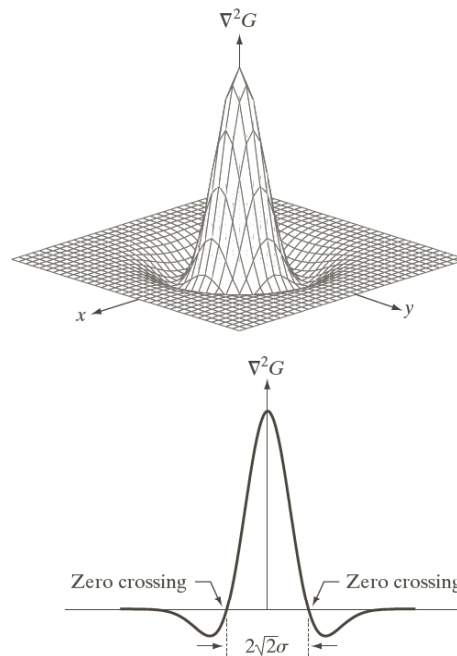
- Laplacian combined with smoothing as a precursor to find edges via zero-crossing.

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

$$\nabla^2 G(x, y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

$$= \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

$$g(x, y) = \nabla^2 G(x, y) * f(x, y)$$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

a b  
c d

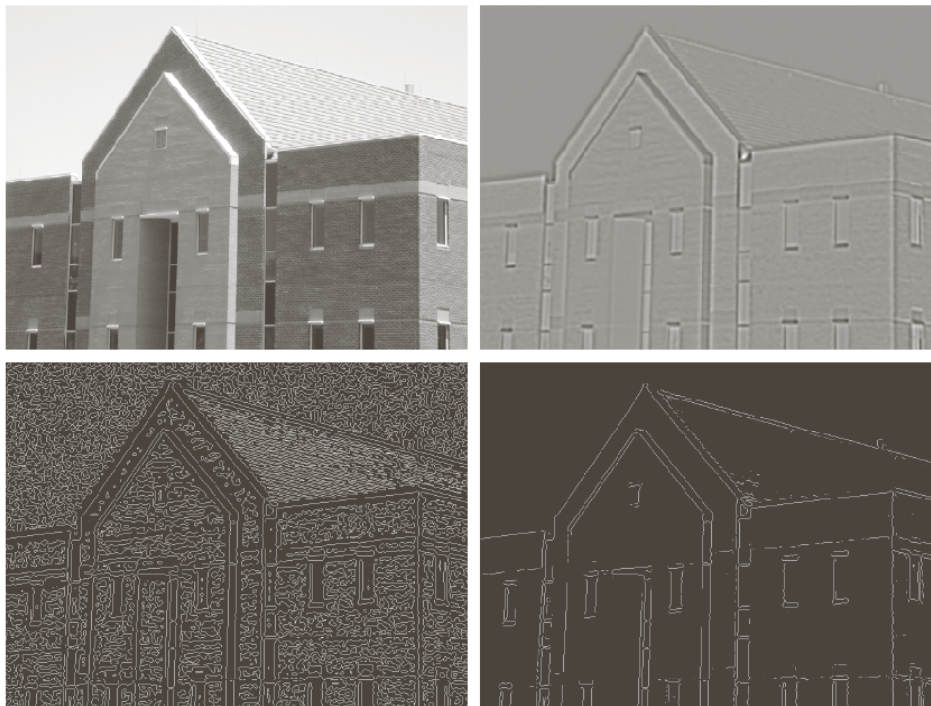
**FIGURE 10.21**  
 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d)  $5 \times 5$  mask approximation to the shape in (a). The negative of this mask would be used in practice.

# Marr-Hildreth edge detection algorithm

---

1. Filter the input with an  $n$  by an Gaussian lowpass filter
2. Compute the Laplacian of the image of step 1
3. Find the zero crossing of the image from step

Approximate the zero crossing from LoG image to threshold the LoG image by setting all its positive values to white and all negative values to black. the zero crossing occur between positive and negative values of the thresholded LoG.



a b  
c d

**FIGURE 10.22**

(a) Original image of size  $834 \times 1114$  pixels, with intensity values scaled to the range  $[0, 1]$ . (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using  $\sigma = 4$  and  $n = 25$ . (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.



# Zero crossing vs. Gradient

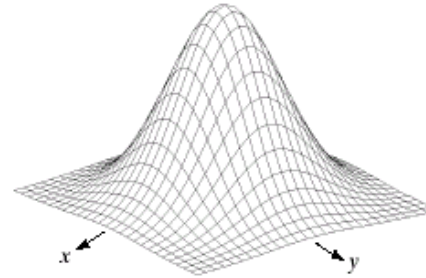
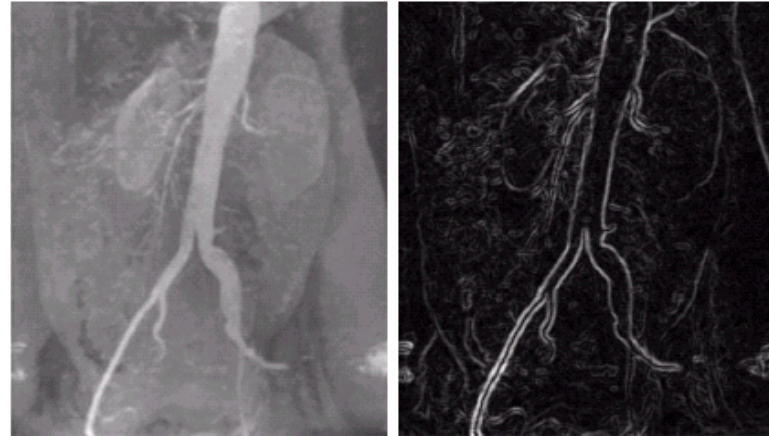
---

- Attractive
  - Zero crossing produces thinner edges
  - Noise reduction
- Drawbacks
  - Zero crossing creates closed loops. (spaghetti effect)
  - sophisticated computation.
- Gradient is more frequently used.

# Example

---

- a). Original image
- b). Sobel Gradient
- c). Spatial Gaussian smoothing function
- d). Laplacian mask
- e). LoG
- f). Threshold LoG
- g). Zero crossing



-1	-1	-1
-1	8	-1
-1	-1	-1



# Edge Linking and Boundary Detection

---

- An edge detection algorithm are followed by linking procedures to assemble edge pixels into meaningful edges.
- Basic approaches
  - Local Processing
  - Global Processing via the Hough Transform
  - Global Processing via Graph-Theoretic Techniques

# Local Processing

---

- Analyze the characteristics of pixels in a small neighborhood  $S_{xy}$  (say, 3x3, 5x5) about every edge pixels  $(x, y)$  in an image.
- All points that are similar according to a set of predefined criteria are linked, forming an edge of pixels that share those criteria.
- Criteria
  - $|M(s, t) - M(x, y)| \leq E$   
where  $E$  is a positive threshold
  - $|\alpha(s, t) - \alpha(x, y)| \leq A$   
where  $A$  is a positive angle threshold
- Algorithm steps
  1. compute  $\nabla f, M(x, y), \alpha(x, y)$
  2. Form a binary image  $g(x, y) = \begin{cases} 1, & M(x, y) > T_M \text{ and } \alpha(x, y) = A \pm T_A \\ 0, & \text{otherwise} \end{cases}$
  3. Scan the rows of  $g$  and fill all gaps in each row that do not exceed a specified length  $K$
  4. Detect the gap in any direction  $\theta$ , rotate  $g$  by this angle and apply the horizontal scanning procedure in Step 3. Rotate the result by  $-\theta$

# Example of local precessing

---

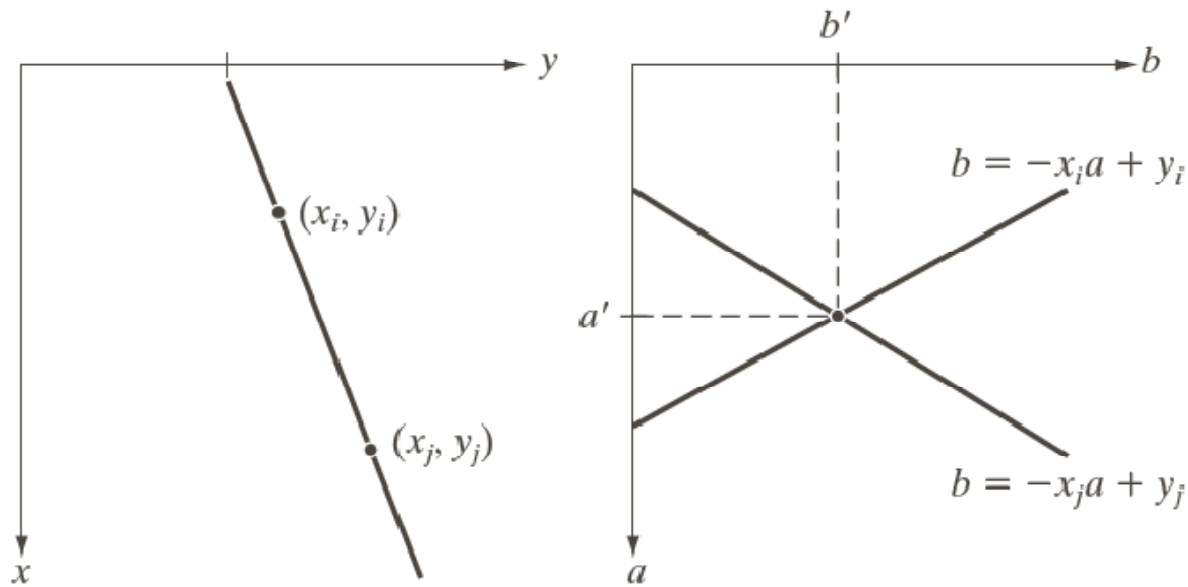


a	b	c
d	e	f

TM =30%,A =90,TA = 45, Gap =25

**FIGURE 10.27** (a) A  $534 \times 566$  image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)

# Hough Transformation (Line)



a b

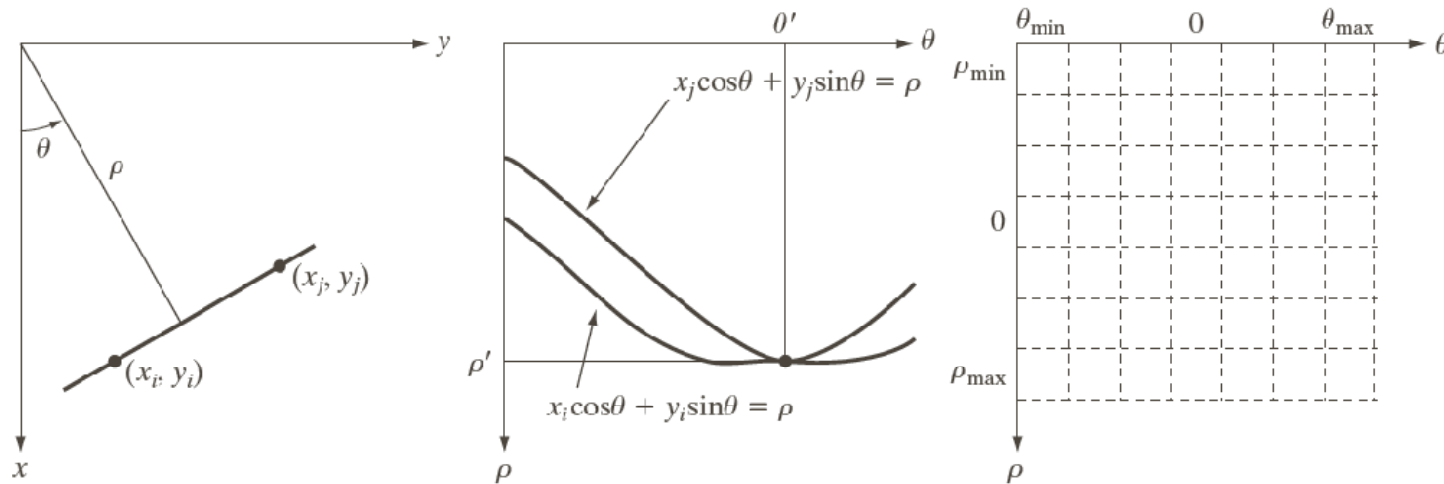
**FIGURE 10.31**  
(a)  $xy$ -plane.  
(b) Parameter space.

$$y_i = ax_i + b$$

$$b = -ax_i + y_i$$

all points  $(x_i, y_i)$  contained on the same line must have lines in parameter space that intersect at  $(a', b')$

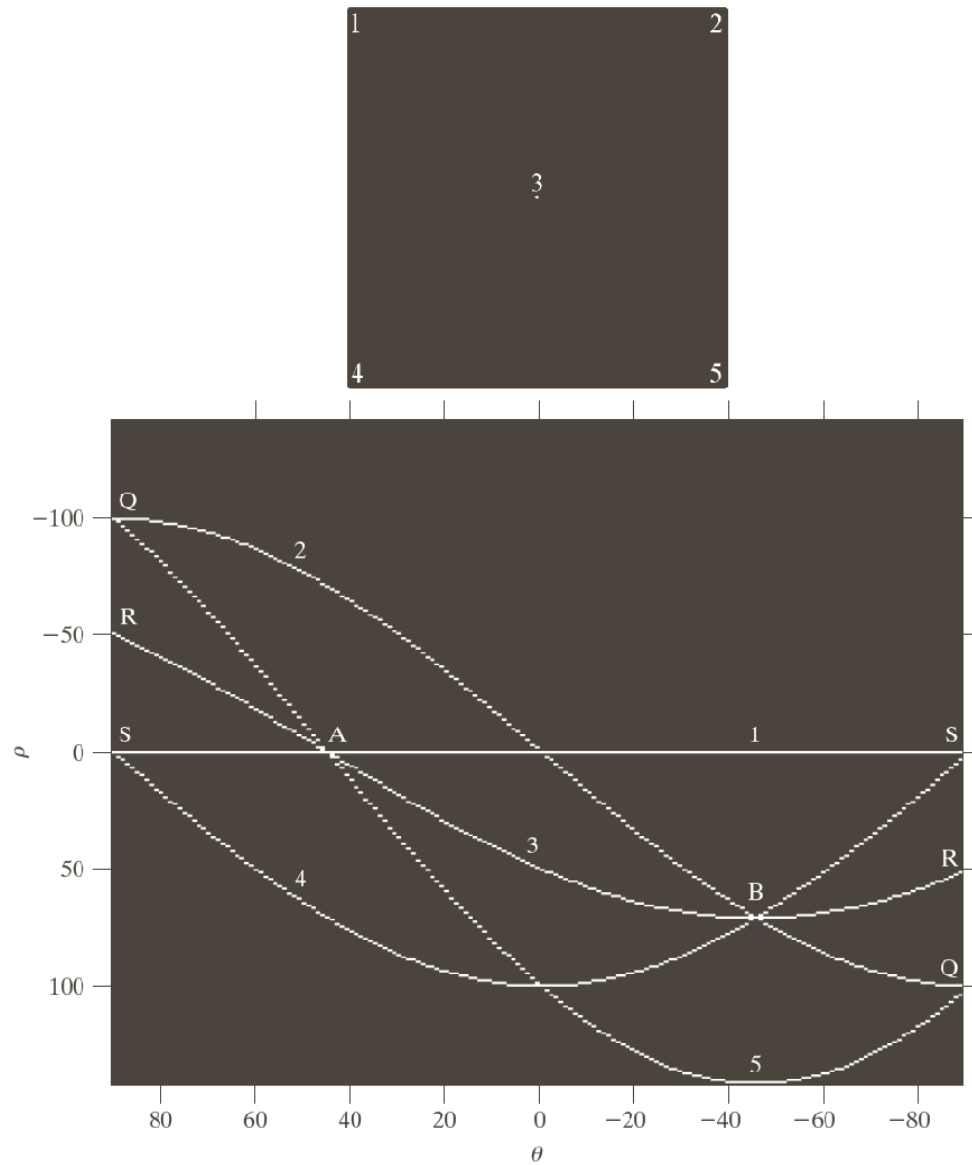
# $\rho\theta$ -plane



a b c

**FIGURE 10.32** (a)  $(\rho, \theta)$  parameterization of line in the  $xy$ -plane. (b) Sinusoidal curves in the  $\rho\theta$ -plane; the point of intersection  $(\rho', \theta')$  corresponds to the line passing through points  $(x_i, y_i)$  and  $(x_j, y_j)$  in the  $xy$ -plane. (c) Division of the  $\rho\theta$ -plane into accumulator cells.

- problem of using equation  $y = ax + b$  is that value of  $a$  is infinite for a vertical line.
- To avoid the problem, use equation  $x \cos \theta + y \sin \theta = \rho$  to represent a line instead.
- vertical line has  $\theta = 90^\circ$  with  $\rho$  equals to the positive  $y$ -intercept or  $\theta = -90^\circ$  with  $\rho$  equals to the negative  $y$ -intercept



a  
b

**FIGURE 10.33**

(a) Image of size  $101 \times 101$  pixels, containing five points.  
 (b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)



# Generalized Hough Transformation

---

The method can be used for any function of the form  $g(v, c) = 0$ , where  $v$  is a vector of coordinates,  $c$  is a vector of coefficients

Example: circles,  $(x-c_1)^2 + (y-c_2)^2 = c_3^2$

$(c_1, c_2, c_3)$  cube like cells, accumulators of the form  $A(i, j, k)$   
increment  $c_1$  and  $c_2$ , solve for  $c_3$  that satisfies the equation  
update the accumulator corresponding to the cell associated with triplet  $(c_1, c_2, c_3)$

# Edge-linking based on Hough Transformation

---

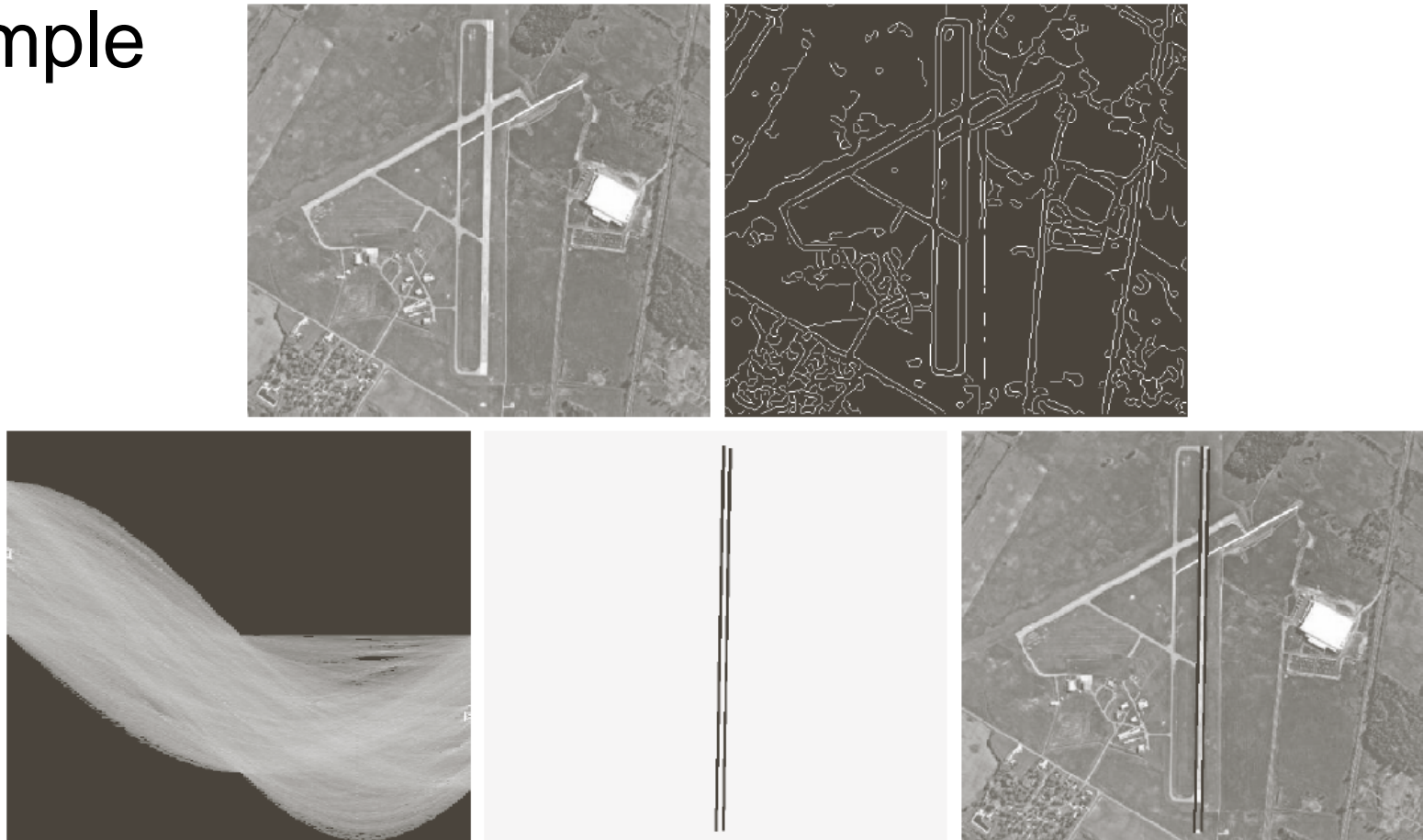
1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify subdivisions in the  $\rho\theta$ -plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

## Continuity:

1. based on computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell.
2. a gap at any point is significant if the distance between that point and its closet neighbor exceeds a certain threshold.

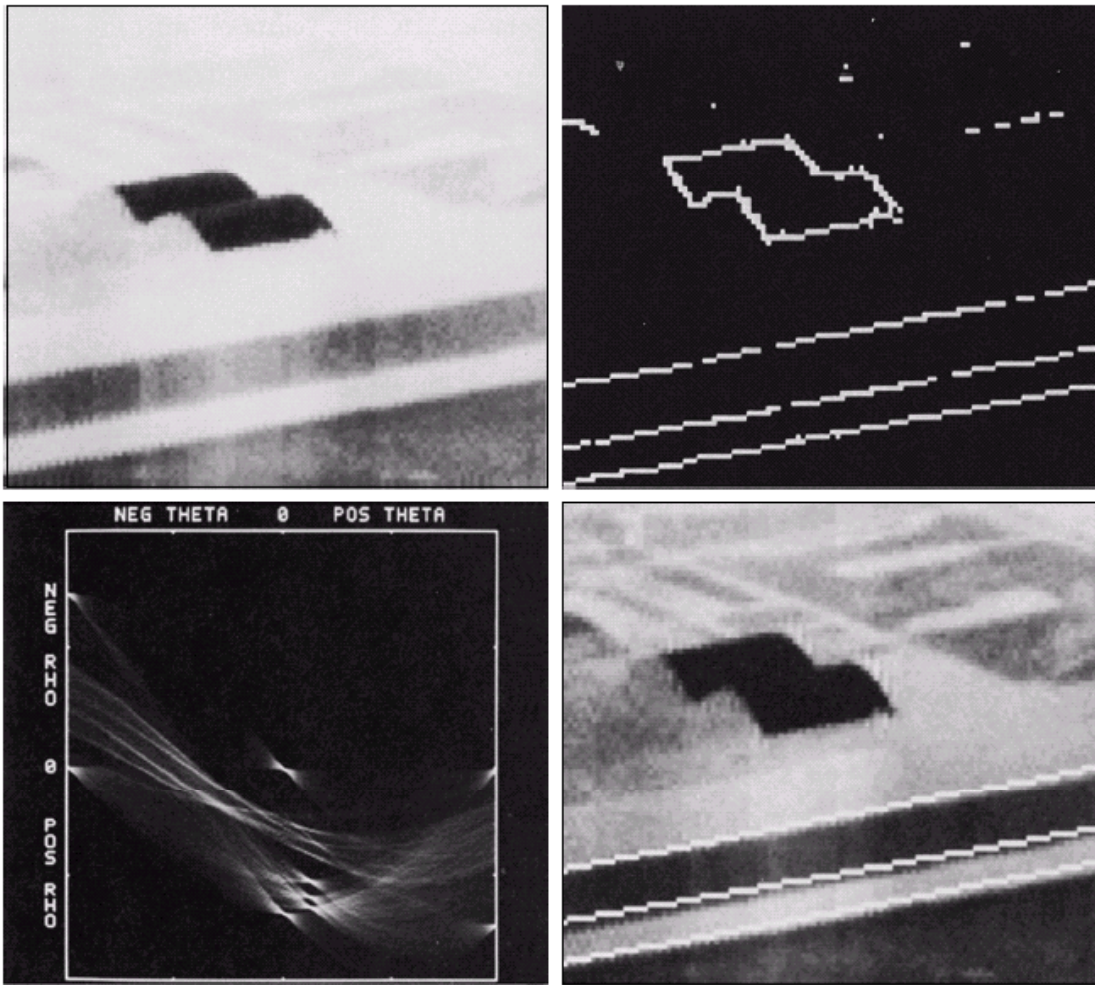
# Example

---



a b  
c d e

**FIGURE 10.34** (a) A  $502 \times 564$  aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.



a	b
c	d

**FIGURE 10.21**

(a) Infrared image.  
 (b) Thresholded gradient image.  
 (c) Hough transform.  
 (d) Linked pixels.  
 (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

link criteria:

- 1). the pixels belonged to one of the set of pixels linked according to the highest count
- 2). no gaps were longer than 5 pixels

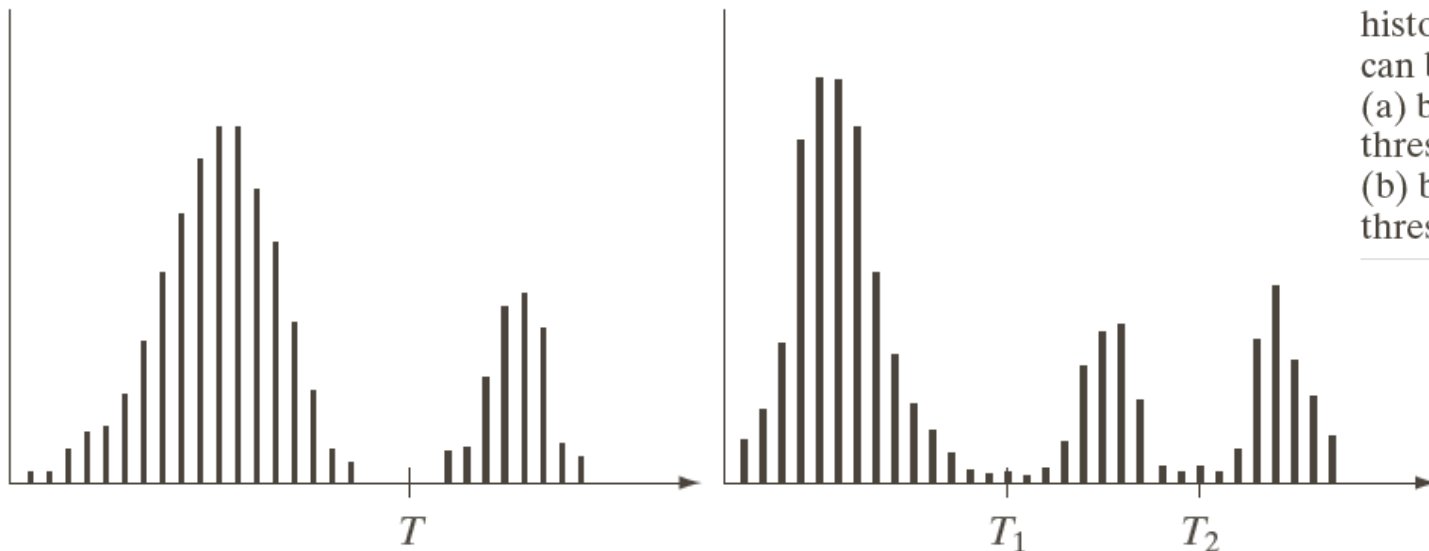
---

# Image Segmentation II

1. Threshold
2. Region-based segmentation
3. Segmentation using watersheds
4. Segmentation with Matlab

# Thresholding

- Problem: how to determine the threshold value of segmentation?
- Histogram



a b

**FIGURE 10.35**  
Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

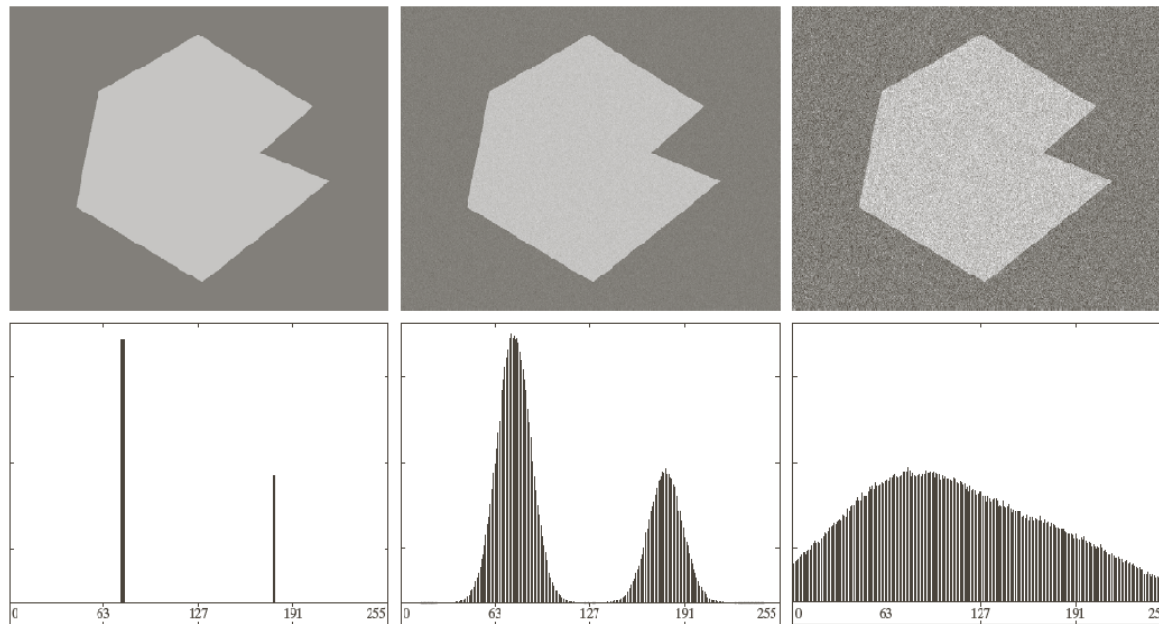
$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) > T \\ 0, & \text{if } f(x, y) \leq T \end{cases}$$

$$g(x, y) = \begin{cases} a, & \text{if } f(x, y) > T_2 \\ b, & \text{if } T_1 < f(x, y) \leq T_2 \\ c, & \text{if } f(x, y) \leq T_1 \end{cases}$$

# The role of noise in image thresholding

---

- When noise is small, the method work, otherwise it may not work. Noise reduction has to be done before choosing threshold value



a	b	c
d	e	f

**FIGURE 10.36** (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.



# The Role of Illumination

---

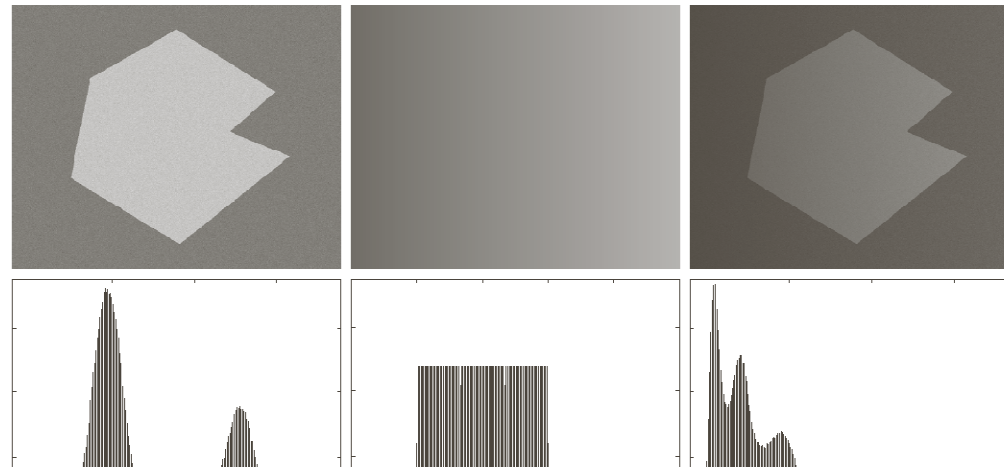
Illumination plays an important role

$$f(x,y) = i(x,y) r(x,y)$$

Approach:

1.  $g(x,y) = ki(x,y)$

2.  $h(x,y) = f(x,y)/g(x,y) = r(x,y)/k$



a	b	c
d	e	f

**FIGURE 10.37** (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.



# Basic Global Thresholding

---

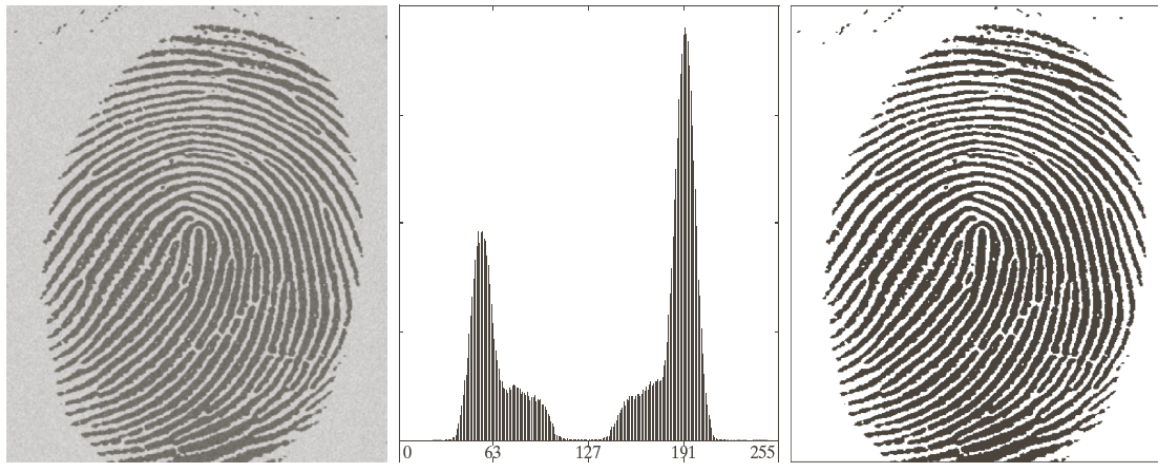
Based on visual inspection of histogram

1. Select an initial estimate for  $T$ .
2. Segment the image using  $T$ . This will produce two groups of pixels:  $G_1$  consisting of all pixels with gray level values  $> T$  and  $G_2$  consisting of pixels with gray level values  $\leq T$
3. Compute the average gray level values  $m_1$  and  $m_2$  for the pixels in regions  $G_1$  and  $G_2$
4. Compute a new threshold value  $T = 0.5 (m_1 + m_2)$
5. Repeat steps 2 through 4 until the difference in  $T$  in successive iterations is smaller than a predefined parameter  $T_o$ .

# Basic Global Thresholding

---

Use  $T$  midway between the max and min gray levels  
generate binary image



a b c

**FIGURE 10.38** (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

$T_0 = 1$  , 3 iterations, result  $T = 125$

# Optimum global thresholding

---

Choose thresholding values that maximize the between-class variance

Otsu's method

1. Compute the normalized histogram of the image  $p_i, i = 0, 1, 2, \dots, L-1$
2. Computer the cumulative sums  $m(k) = \sum_{i=0}^k ip_i, k = 0, 1, 2, \dots, L-1$
3. Computer the cumulative mean  $P_1(k) = \sum_{i=0}^k p_i, k = 0, 1, 2, \dots, L-1$
4. Computer the global intensity mean,  $m_G = \sum_{i=0}^{L-1} ip_i, k = 0, 1, 2, \dots, L-1$
5. Compute the between-class variance

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}, k = 0, 1, 2, \dots, L-1$$

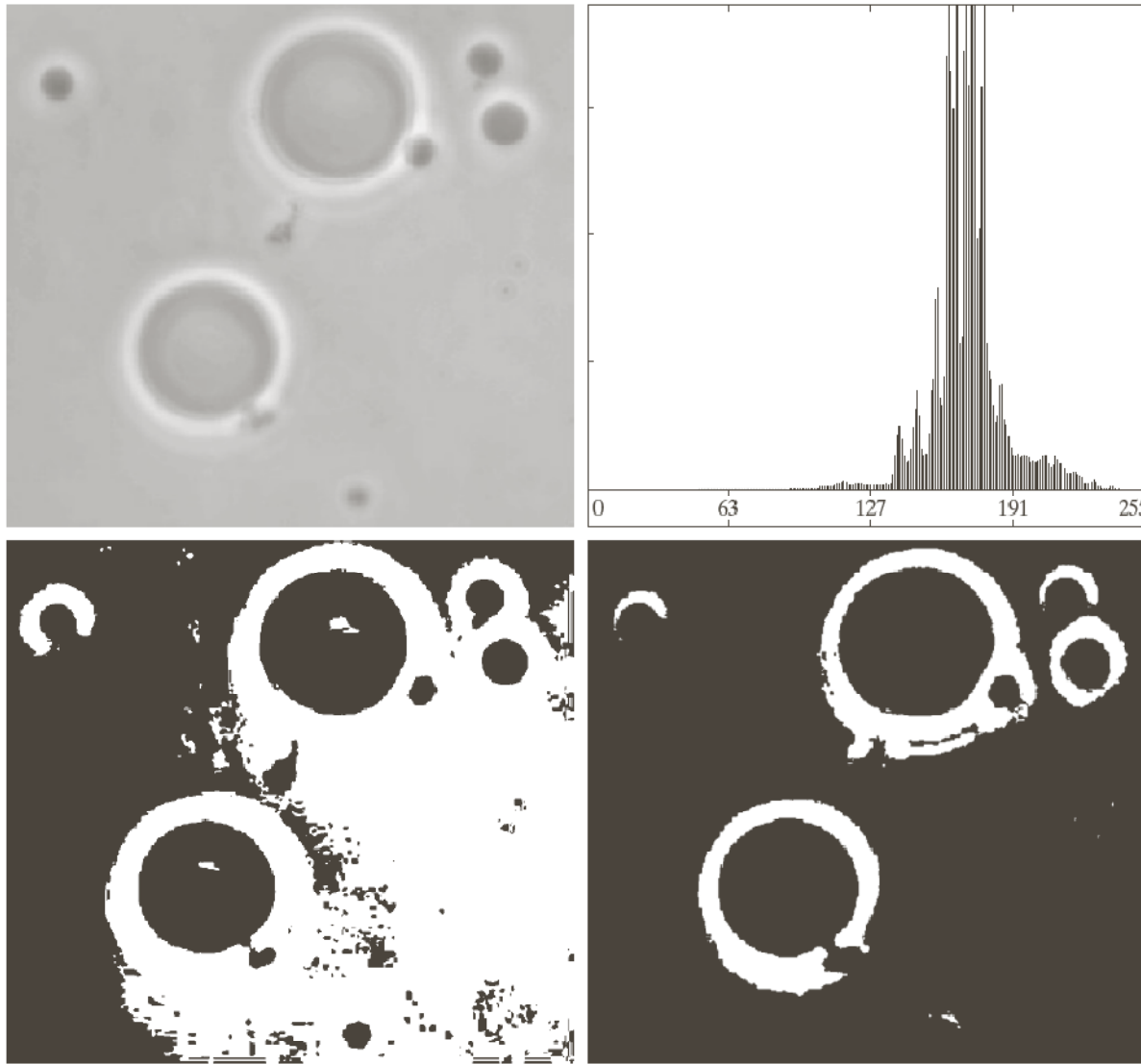
6. Obain the Otsu thereshold

$k^*$  the average of all  $k'$  such that  $\sigma_B^2(k') = \max_k \{ \sigma_B^2(k) \}$

7. Compute the separability measure

$$\eta^* = \frac{\sigma_B^2(k^*)}{\sigma_G^2}$$

# Example



a	b
c	d

**FIGURE 10.39**

(a) Original image.

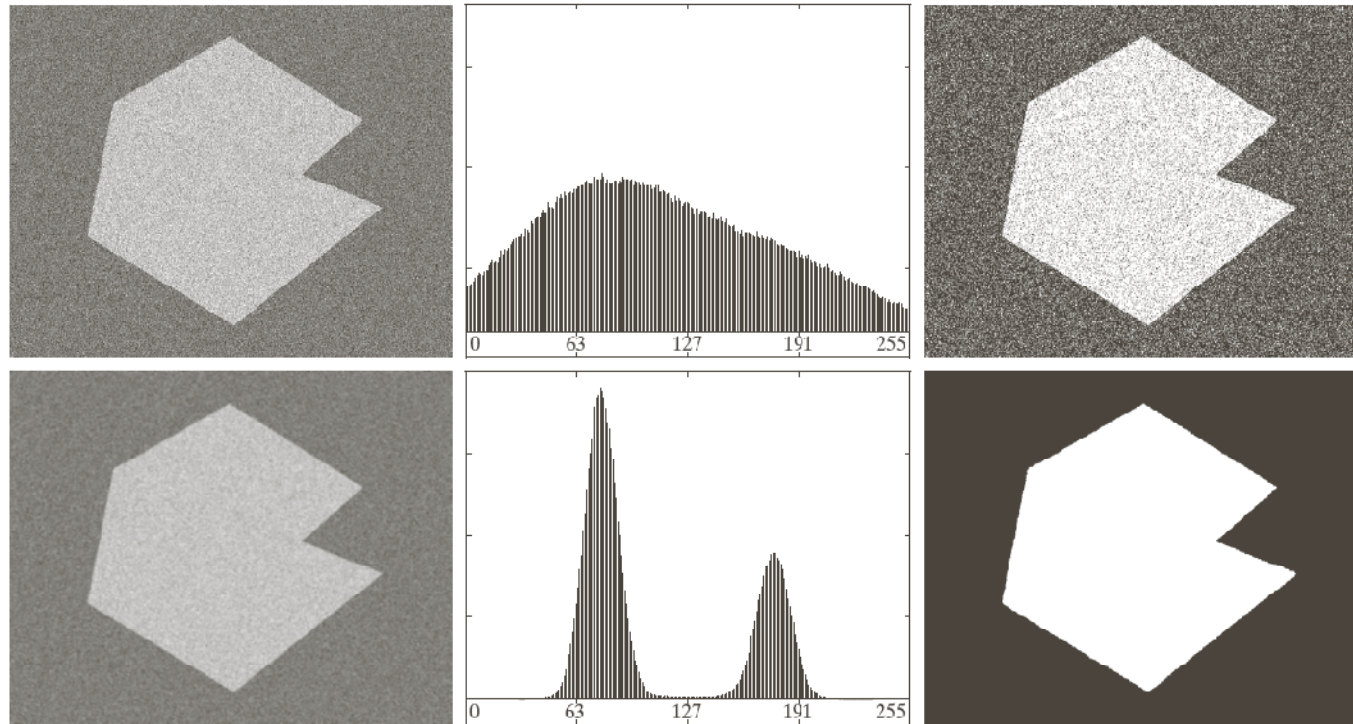
(b) Histogram (high peaks were clipped to highlight details in the lower values).

(c) Segmentation result using the basic global algorithm from Section 10.3.2.

(d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

# Example

---

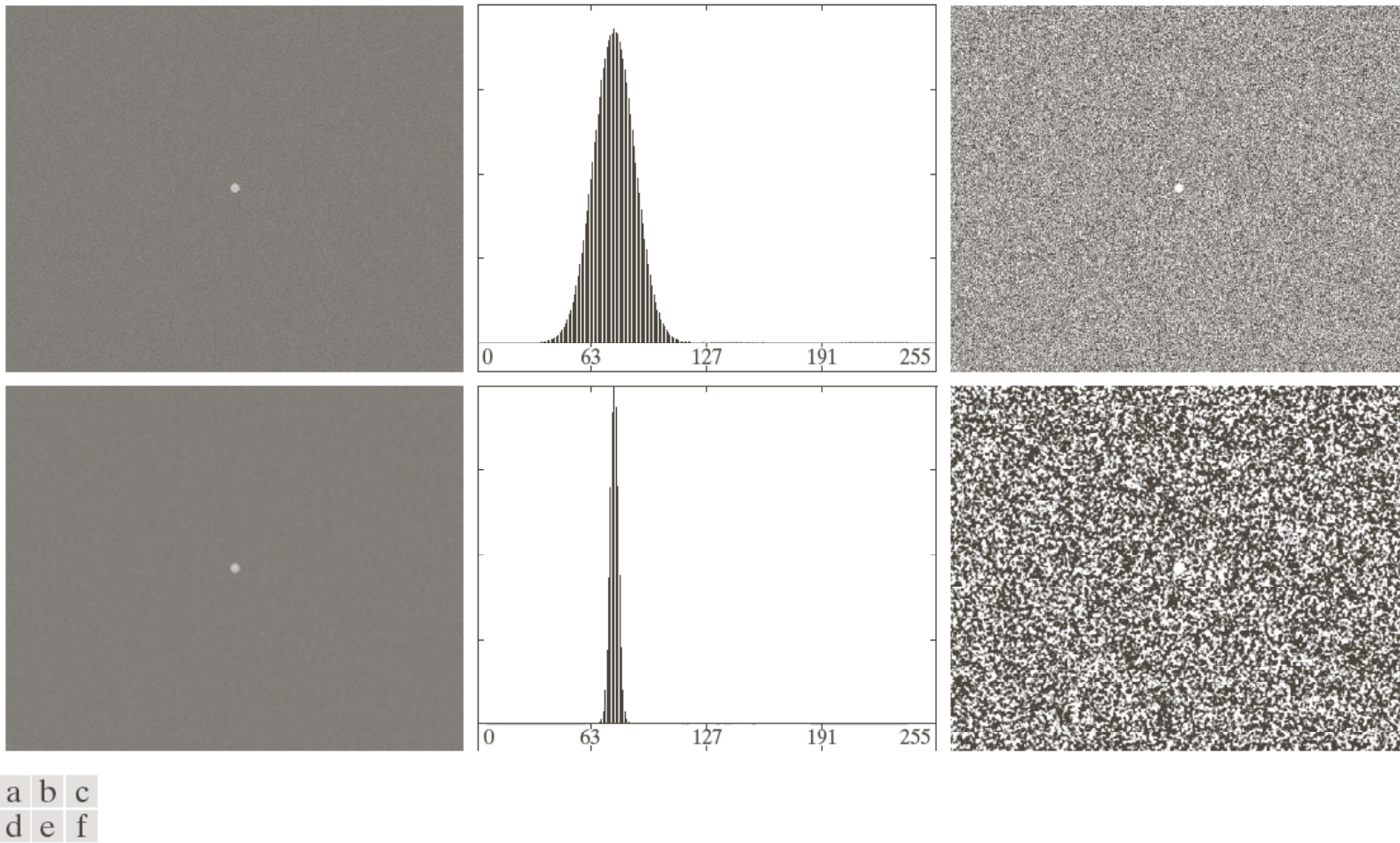


a	b	c
d	e	f

**FIGURE 10.40** (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a  $5 \times 5$  averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.



# Example



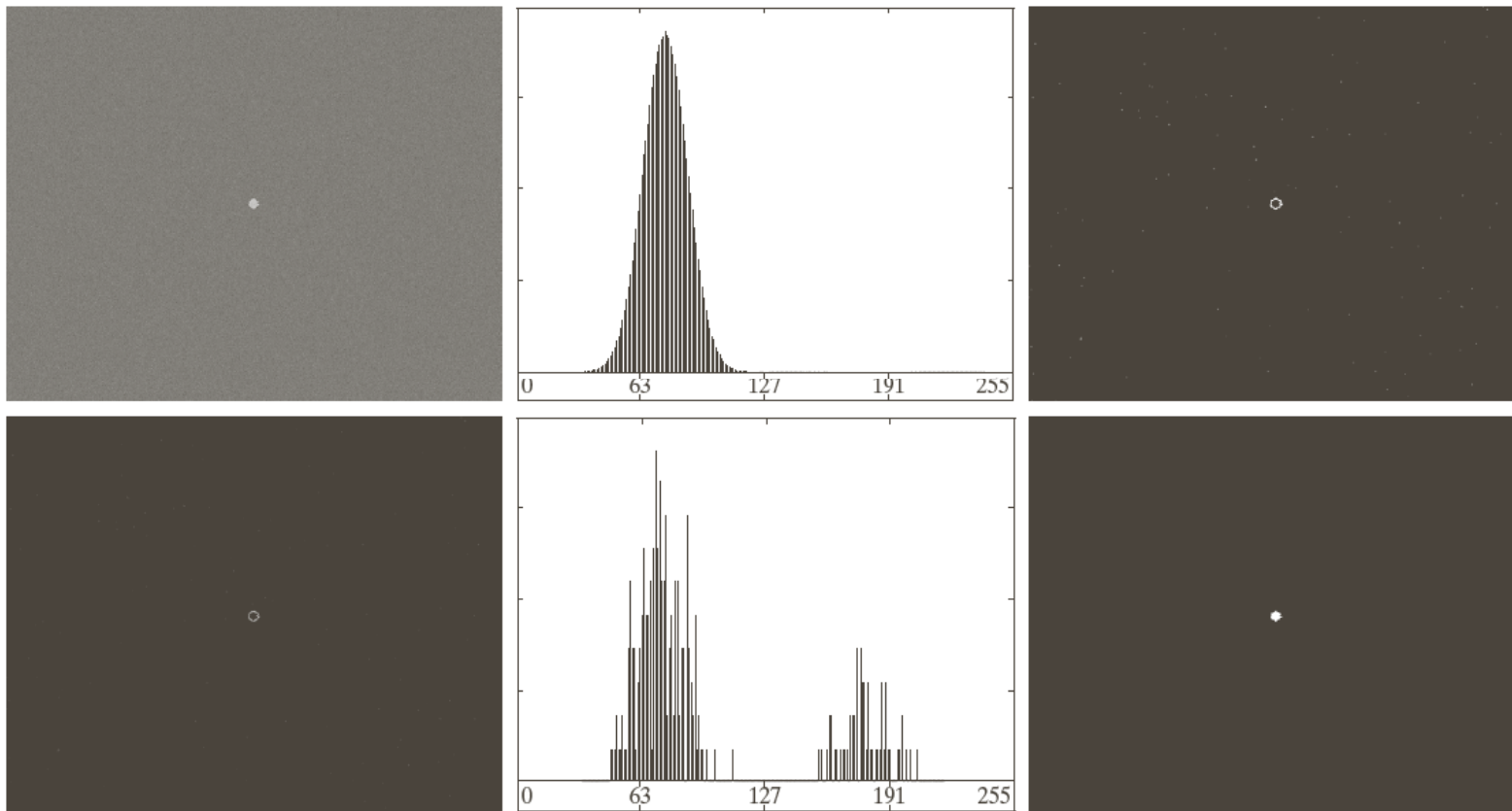
**FIGURE 10.41** (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a  $5 \times 5$  averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

# Using edges to improve global thresholding

---

1. Compute the edge image  $g(x, y)$  of  $f(x, y)$
2. Specify a threshold value  $T$  of  $g(x, y)$
3. Threshold the image  $f(x, y)$  using  $T$ , obtain  $g_t(x, y)$
4. Compute a histogram of  $f(x, y)$  using pixels  $(x, y)$  with  $g_t(x, y) = 1$
5. Use the above histogram to segment  $f(x, y)$  globally.

# Example



a b c  
d e f

**FIGURE 10.42** (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.



# Multiple thresholding

---

Otsu's method can be extended to arbitrary number of thresholds.

1. Compute the normalized histogram of the image  $p_i, i = 0, 1, 2, \dots, L-1$
2. Computer the cumulative sums  $P_k = \sum_{i \in C_k} p_i, m_k = \frac{1}{P_k} \sum_{i \in C_k} ip_i, C_1, \dots, C_K$  are classes
3. Computer the cumulative mean

$$m_G = \sum_{i=1}^K P_i m_i, \sigma_B^2(k_1, \dots, k_K) = \sum_{i=1}^K P_i [m_i - m_G]^2$$

4. Obain theresold

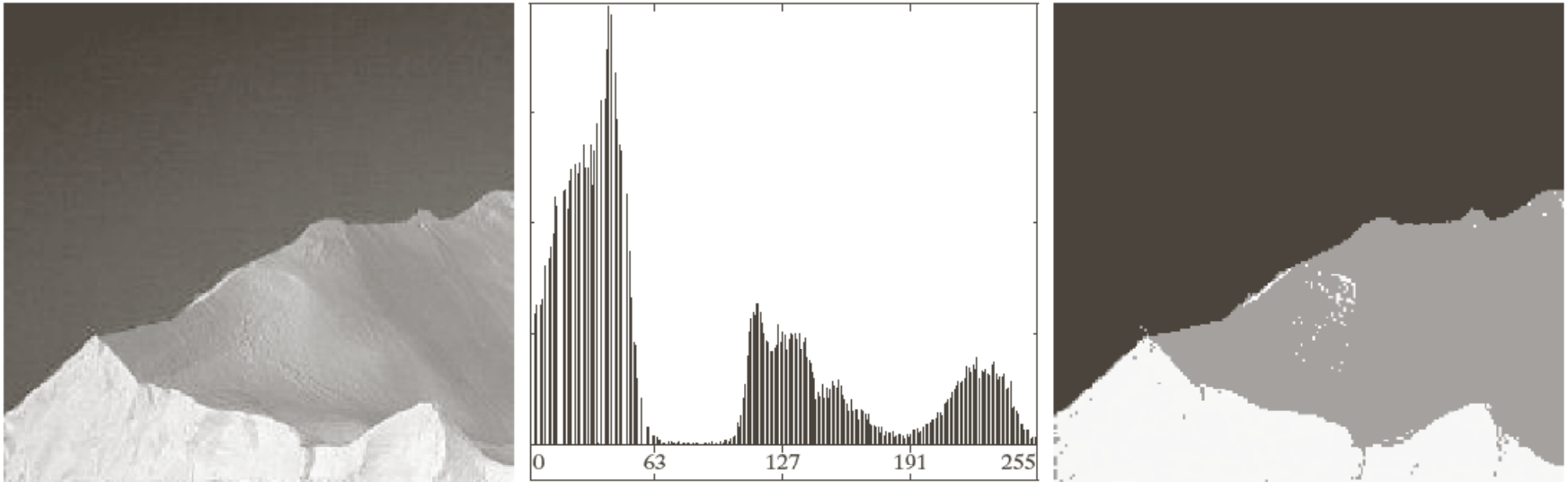
$$k_1^*, \dots, k_{K-1}^* \text{ such that } \sigma_B^2(k_1^*, \dots, k_{K-1}^*) = \max_k \{ \sigma_B^2(k_1, \dots, k_{K-1}) \}$$

5. Threshold image

$$g(x, y) = \begin{cases} a_1 & \text{if } f(x, y) \leq k_1^* \\ a_2 & \text{if } k_1^* < f(x, y) \leq k_2^* \\ & \vdots \\ a_K & \text{if } f(x, y) > k_K^* \end{cases}$$

# Example

---



a b c

**FIGURE 10.45** (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

---

# Basic Adaptive Thresholding

---

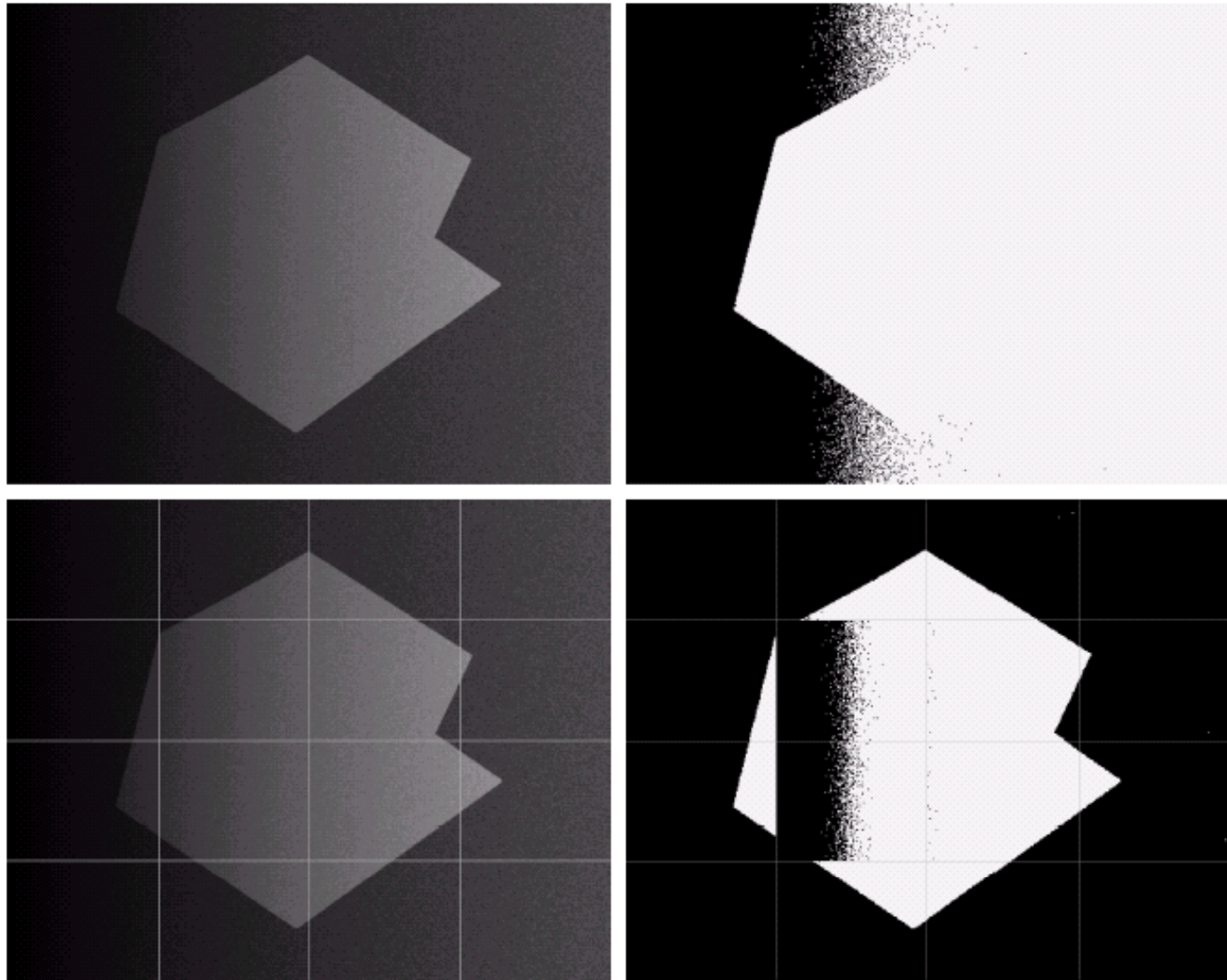
- Subdivide original image into small areas.
- Utilize a different threshold to segment each subimages.
- The threshold used for each pixel depends on the location of the pixel in terms of the subimages.

# Example : Adaptive Thresholding

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.

---



## 2. Region-Based Segmentation

---

- Basic Formulation

$$(a) \bigcup_{i=1}^n R_i = R$$

(b)  $R_i$  is a connected region,  $i = 1, 2, \dots, n$

(c)  $R_i \cap R_j = \phi$  for all  $i$  and  $j$ ,  $i \neq j$

(d)  $P(R_i) = \text{TRUE}$  for  $i = 1, 2, \dots, n$

(e)  $P(R_i \cup R_j) = \text{FALSE}$  for  $i \neq j$

$P(R_i)$  is a logical predicate property defined over the points in set  $R_i$

ex.  $P(R_i) = \text{TRUE}$  if all pixel in  $R_i$  have the same gray level

# Two basic approaches

---

- Region Growing

- start with a set of “**seed**” points
- growing by appending to each seed those neighbors that have similar properties such as specific ranges of gray level

e.g

$$Q = \begin{cases} \text{TRUE} & \text{if the absolute difference of the} \\ & \text{intensities between the seed and} \\ & \text{the pixel at (x,y) is } \leq T \\ \text{FALSE} & \text{otherwise} \end{cases}$$

- Region splitting and merging

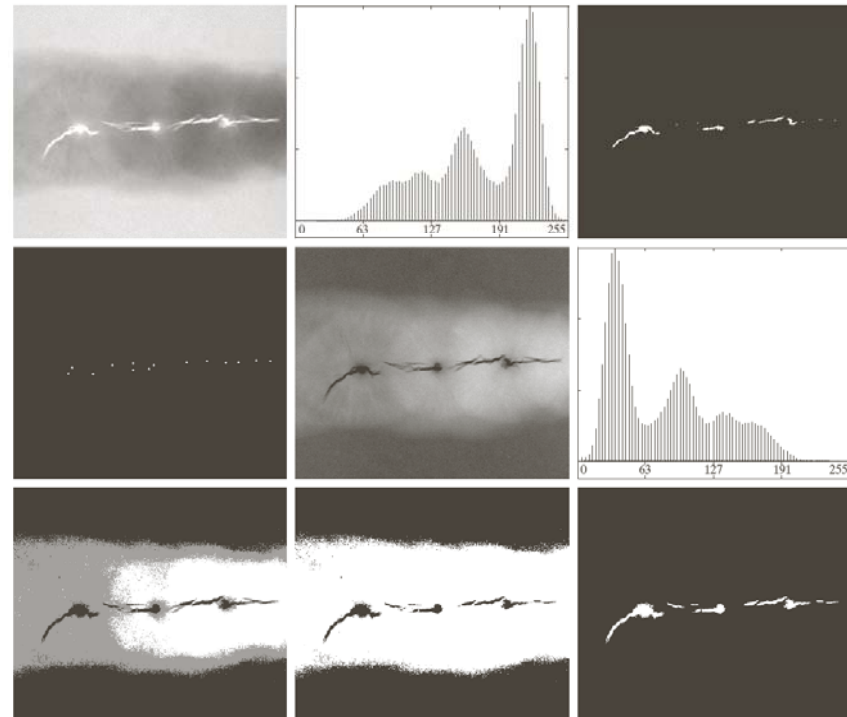
- Iteratively divide a region into smaller regions until all regions become TRUE
- Merge adjacent regions as long as the resulting region is still TRUE

# Region Growing

Select all seed points with gray level 255

criteria:

1. the absolute gray-level difference between any pixel and the seed has to be less than 65
2. the pixel has to be 8-connected to at least one pixel in that region (if more, the regions are merged)



a b c  
d e f  
g h i

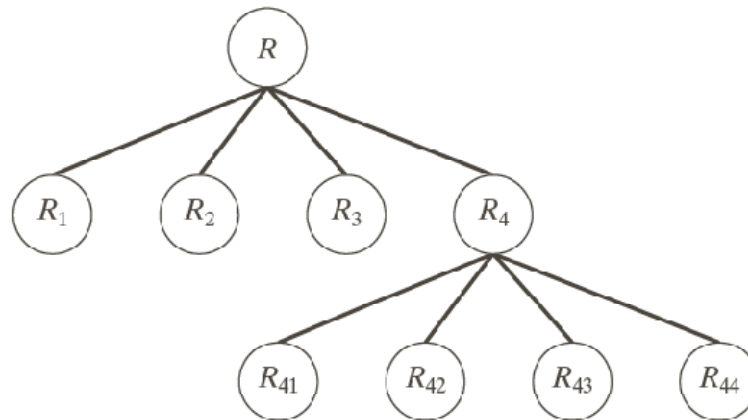
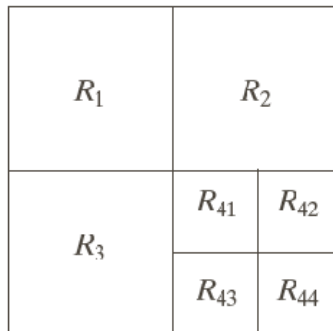
**FIGURE 10.51** (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)

# Region splitting and merging

---

## Quadtree

1. Split into 4 disjoint quadrants any region  $R_i$  for which  $P(R_i) = \text{FALSE}$
2. Merge any adjacent region  $R_j$  and  $R_k$  for which  $P(R_j \cup R_k) = \text{TRUE}$
3. Stop when no further merging or splitting is possible.



a b

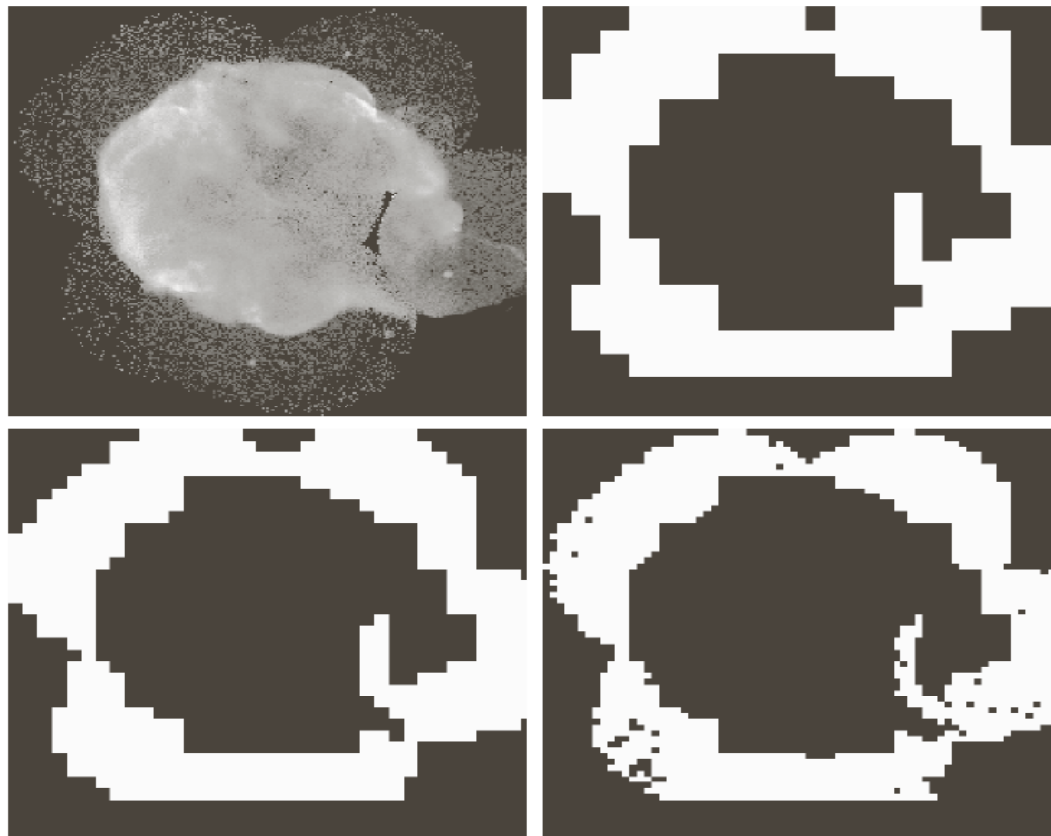
**FIGURE 10.52**

(a) Partitioned image.

(b) Corresponding quadtree.  $R$  represents the entire image region.



# Example



a b  
c d

**FIGURE 10.53**

(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of  $32 \times 32$ ,  $16 \times 16$ , and  $8 \times 8$  pixels, respectively. (Original image courtesy of NASA.)

$$Q = \begin{cases} \text{TRUE} & \text{if } \sigma > a \text{ and } 0 < m < b \\ \text{FALSE} & \text{otherwise} \end{cases}$$

where  $m$  and  $\sigma$  the mean and standard deviation of pixels in a quadregion

# Example

---

a b c

**FIGURE 10.43**

(a) Original image. (b) Result of split and merge procedure. (c) Result of thresholding (a).



$P(R_i) = \text{TRUE}$  if at least 80% of the pixels in  $R_i$  have the property  $|z_j - m_i| \leq 2\sigma_i$ ,  
where

$z_j$  is the gray level of the  $j^{\text{th}}$  pixel in  $R_i$

$m_i$  is the mean gray level of that region

$\sigma_i$  is the standard deviation of the gray levels in  $R_i$

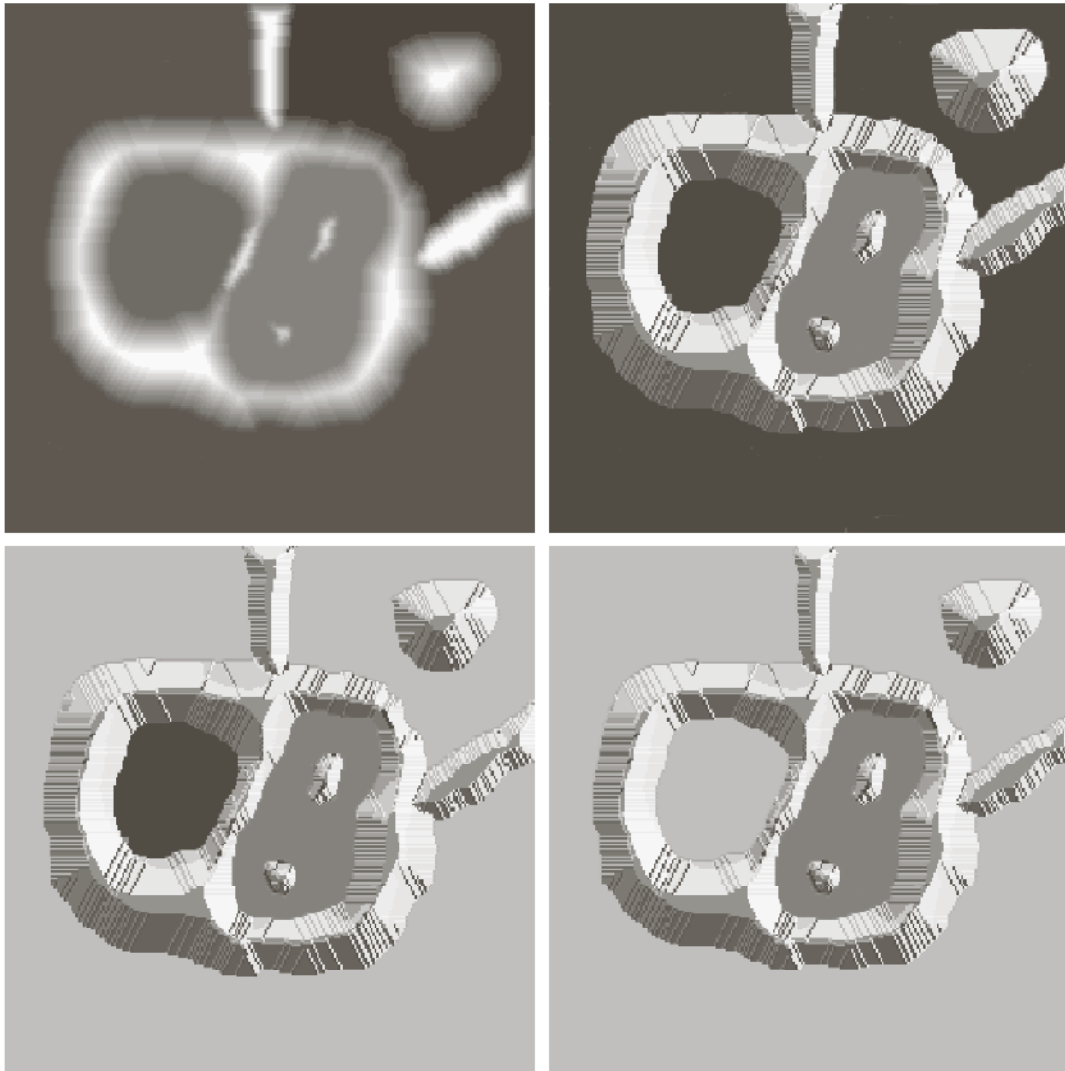
# Segmentation using Watersheds

---

- The concept of watersheds
  - View an image as 3-D graphics
  - Three types of points
    - Local minimum
    - Point at which water will flow to a local minimum, also called catchment basin, or watershed.
    - Point at which water can equally flow to more than one local minimum points, also called divide lines, or watershed lines
- Segmentation by watershed lines

# Example

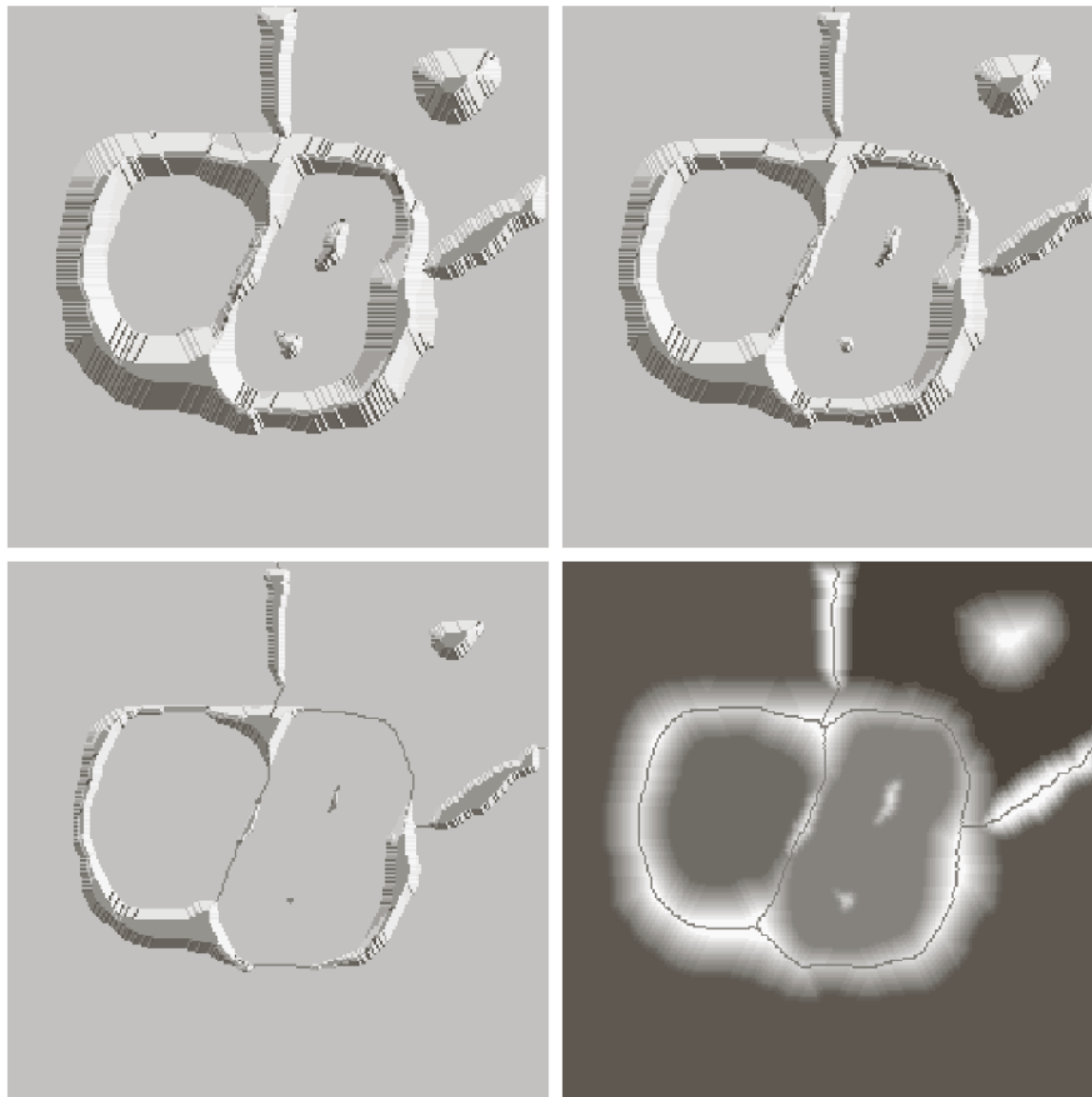
---



a	b
c	d

**FIGURE 10.54**  
(a) Original image.  
(b) Topographic  
view. (c)–(d) Two  
stages of flooding.

---



e f  
g h

**FIGURE 10.54**

*(Continued)*

(e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines.

(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

# Watershed algorithm

---

Let  $M_1, M_2, \dots, M_R$  be sets denoting the coordinates of the regional minima of image  $g(x, y)$ .  $C(M_i)$  is the set of pixels of catchment basin of  $M_i$

$$T[n] = \{(s, t) \mid g(s, t) < n\}, n = \min+1 \text{ to } \max+1$$

$$C_n(M_i) = C(M_i) \cap T[n]$$

$Q[n]$  denotes the set of connected components of  $M_i$

For  $n = \min+1$  to  $\max+1$  do

For each connected component  $q$  of  $Q[n]$

if  $q \cap C[n-1] = \emptyset$

Let  $C[n] \leftarrow C[n-1] \cup q$

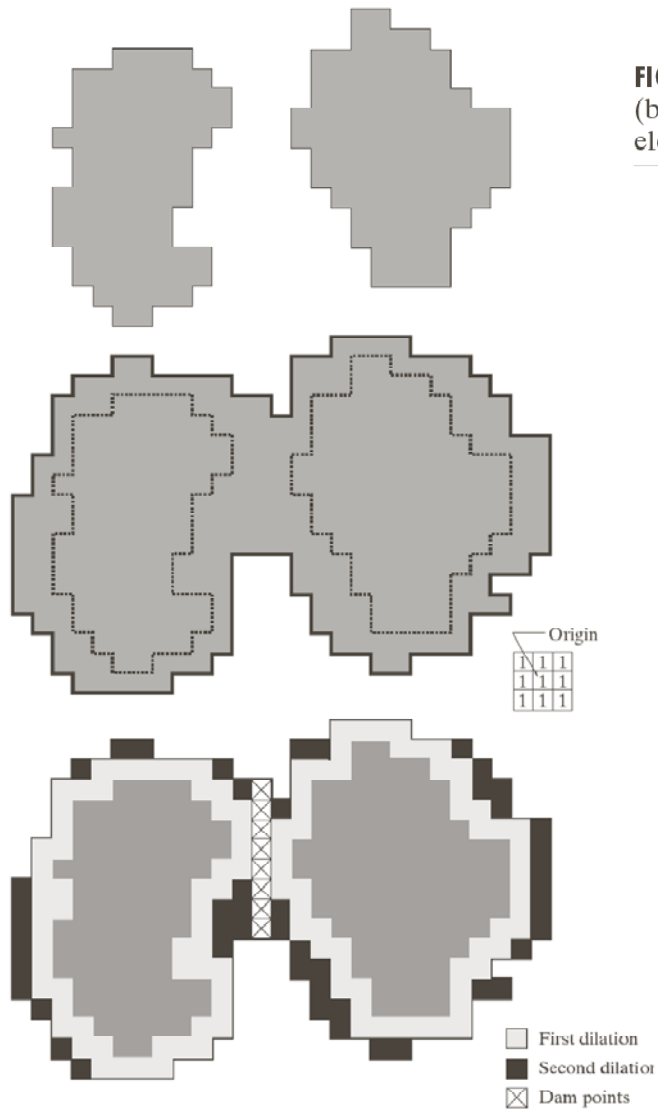
else if  $q \cap C[n-1]$  contains one connected component of  $C[n-1]$

$C[n] \leftarrow C[n-1] \cup q$

else  $q \cap C[n-1]$  contains more than one components of  $C[n-1]$

The build dam within  $q$

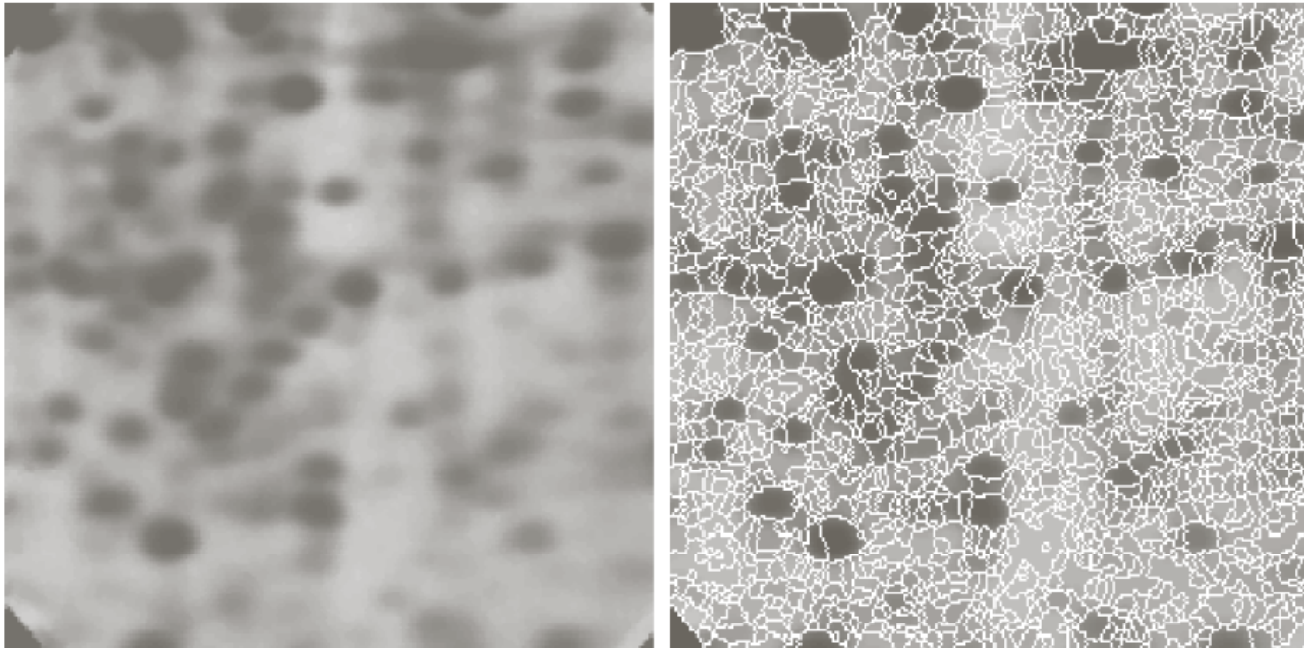
# Dam construction



**FIGURE 10.55** (a) Two partially flooded catchment basins at stage  $n - 1$  of flooding. (b) Flooding at stage  $n$ , showing that water has spilled between basins. (c) Structuring element used for dilation. (d) Result of dilation and dam construction.

# Example

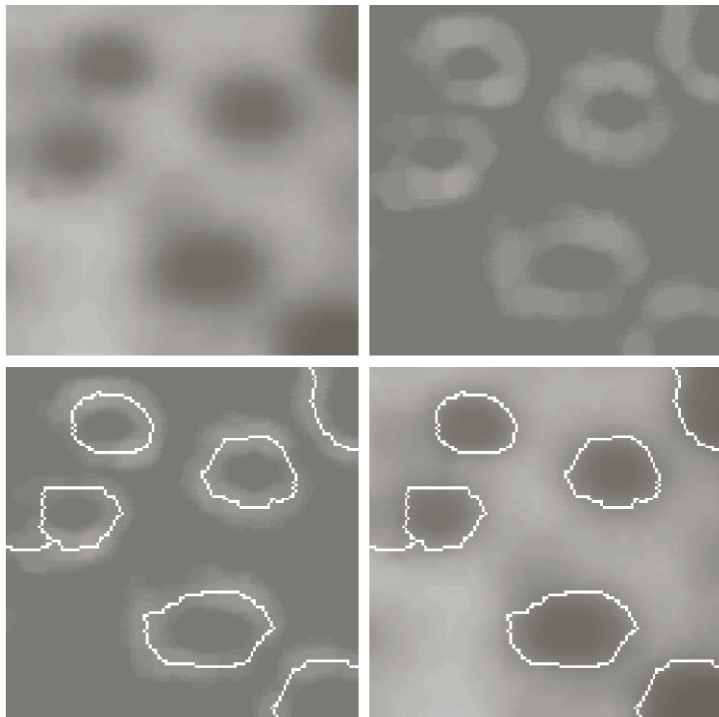
---



a b

**FIGURE 10.57**  
(a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)



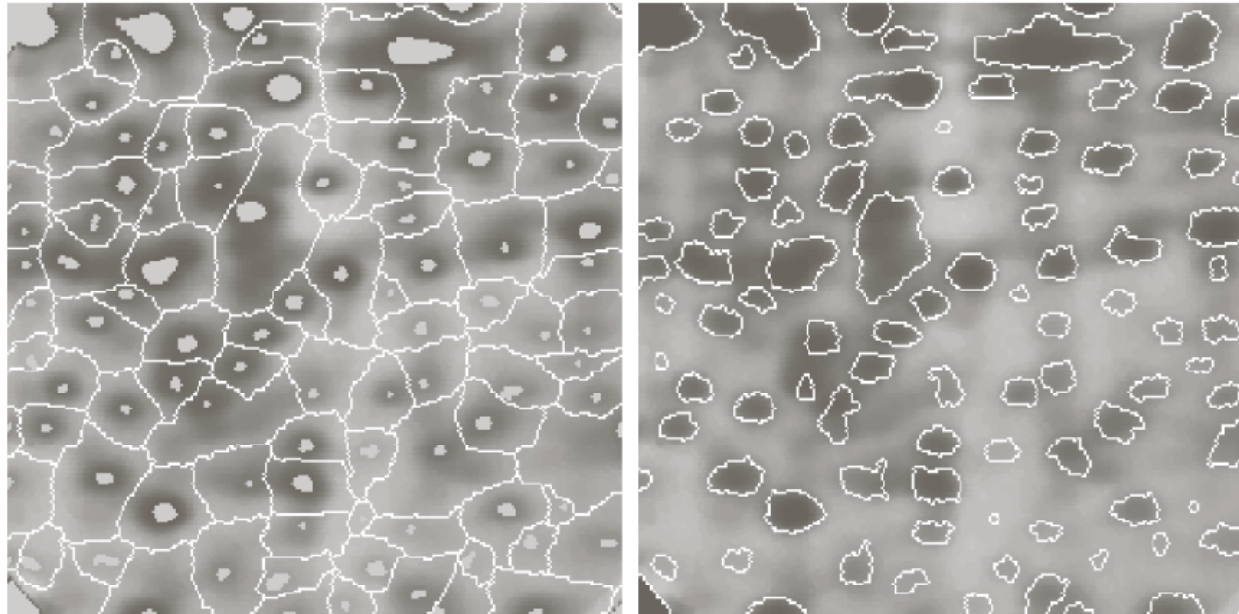


a	b
c	d

**FIGURE 10.56**

(a) Image of blobs.  
(b) Image gradient.  
(c) Watershed lines.  
(d) Watershed lines superimposed on original image.  
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

---



a b

**FIGURE 10.58** (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

---