Image Segmentation

Ayatullah Faruk Mollah, PhD
Department of Computer Science and Engineering,
Aliah University

Fundamentals

- Let R represent the entire spatial region occupied by an image. Image segmentation is a process that partitions R into *n* sub-regions, R_1, R_2, …, R_n, such that

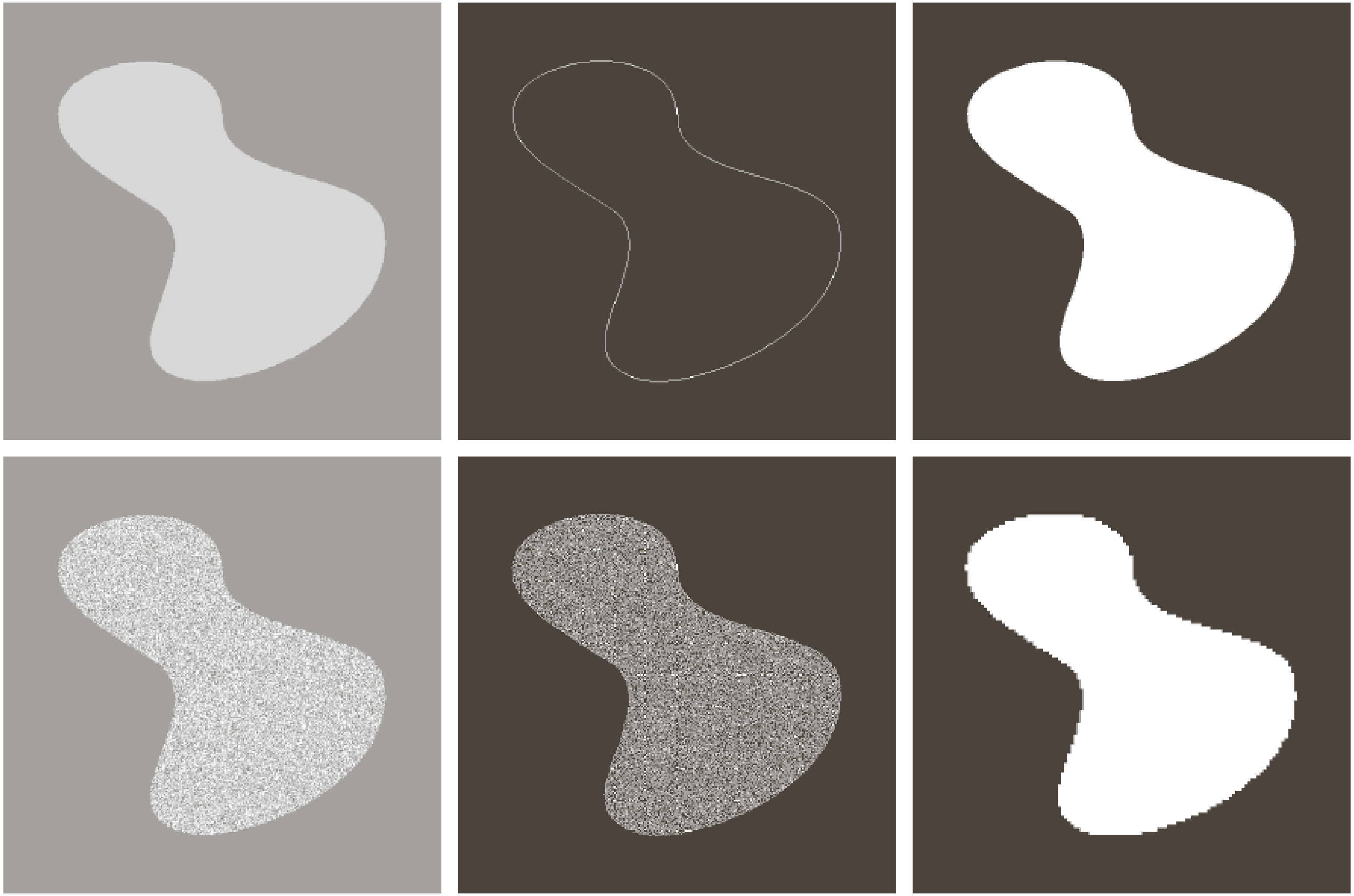
(a) $\bigcup_{i=1}^{n} R_i = R$.

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

(c) $R_i \cap R_j = \Phi$.

(d) $Q(R_i) = \text{TRUE}$ for $i = 1, 2, \ldots, n$.

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



a b c
d e f

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.

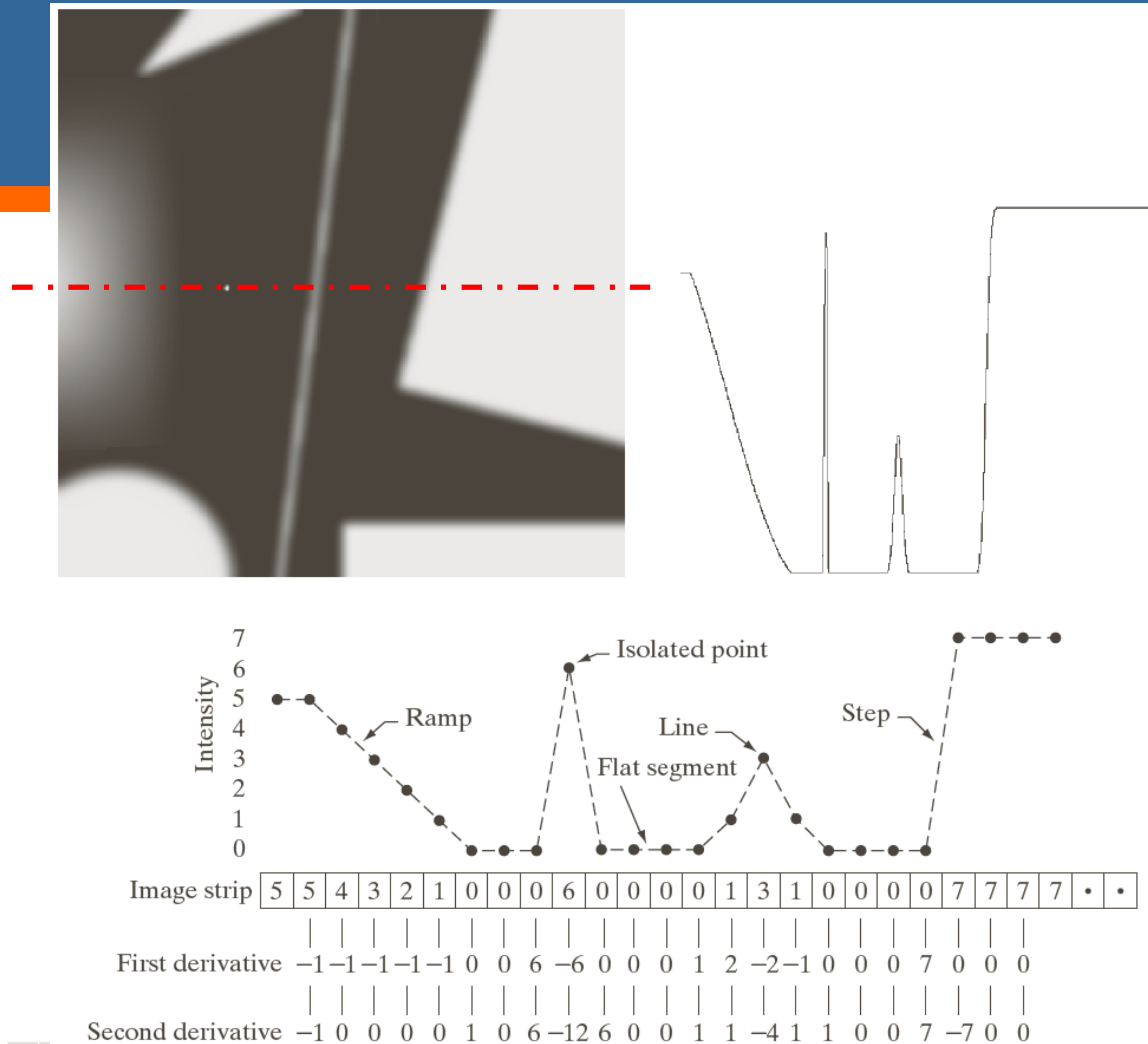
Background

- First-order derivative

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

- Second-order derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

Characteristics of First and Second Order Derivatives

- First-order derivatives generally produce thicker edges in image
- Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
- Second-order derivatives produce a double-edge response at ramp and step transition in intensity
- The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light

Detection of Isolated Points

- The Laplacian

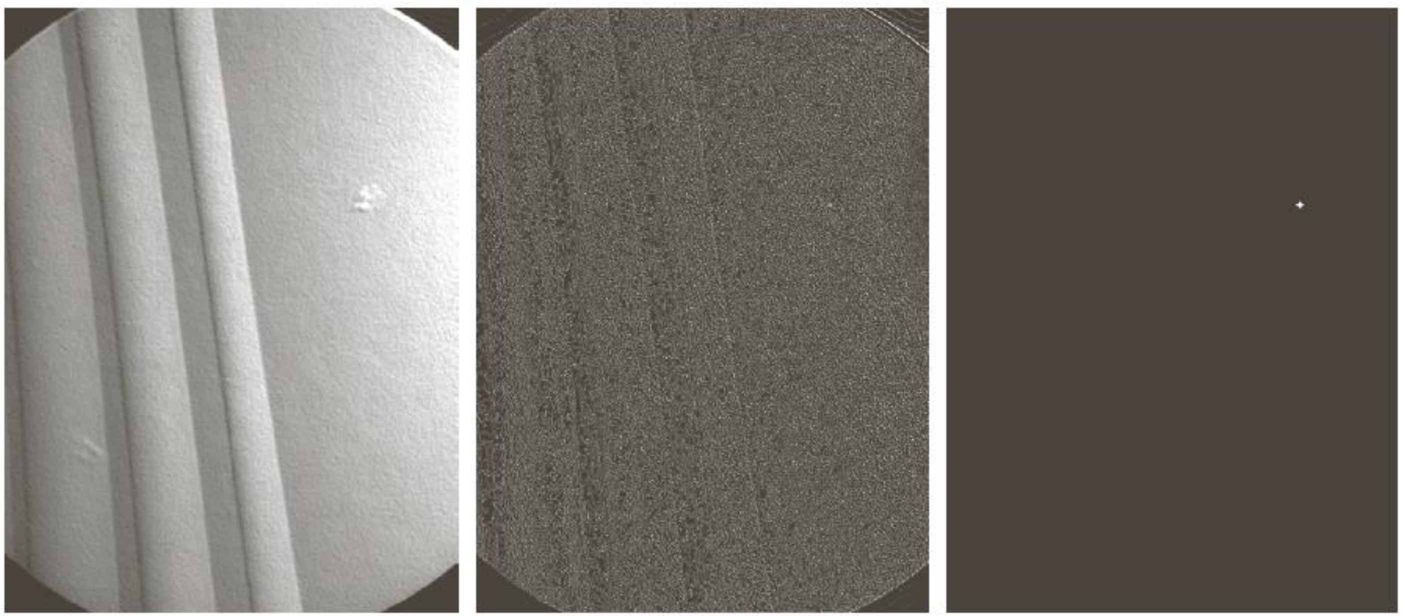
$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)$$

$$-4f(x,y)$$

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases} \qquad R = \sum_{k=1}^{9} w_k z_k$$

1	1	1
1	−8	1
1	1	1



a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Line Detection

- Second derivatives to result in a stronger response and to produce thinner lines than first derivatives
- Double-line effect of the second derivative must be handled properly

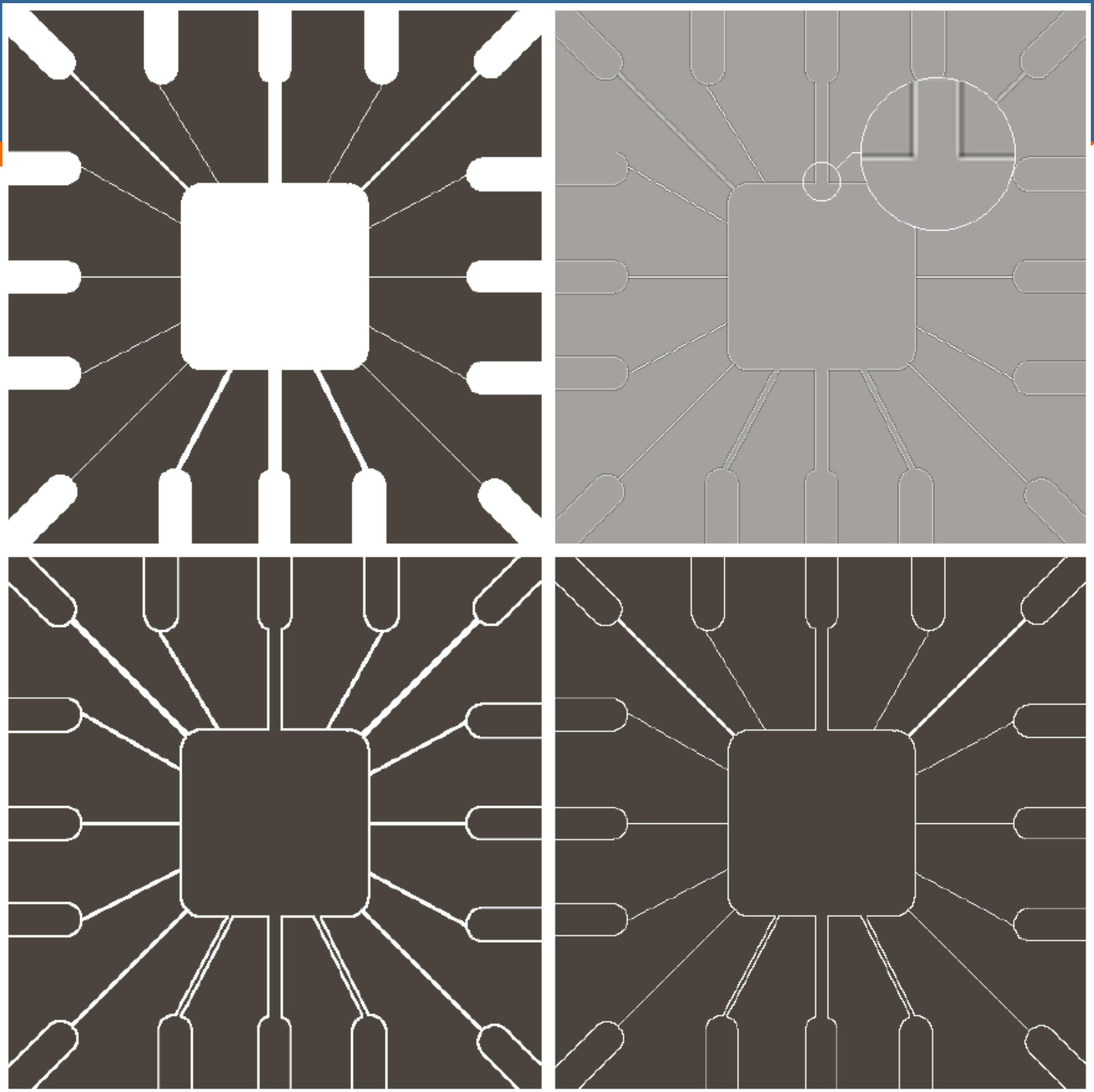


FIGURE 10.5
(a) Original image.
(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.
(c) Absolute value of the Laplacian.
(d) Positive values of the Laplacian.

Detecting Line in Specified Directions

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

- Let R_1, R_2, R_3, and R_4 denote the responses of the masks in Fig. 10.6. If, at a given point in the image, $|R_k|>|R_j|$, for all $j \neq k$, that point is said to be more likely associated with a line in the direction of mask k.

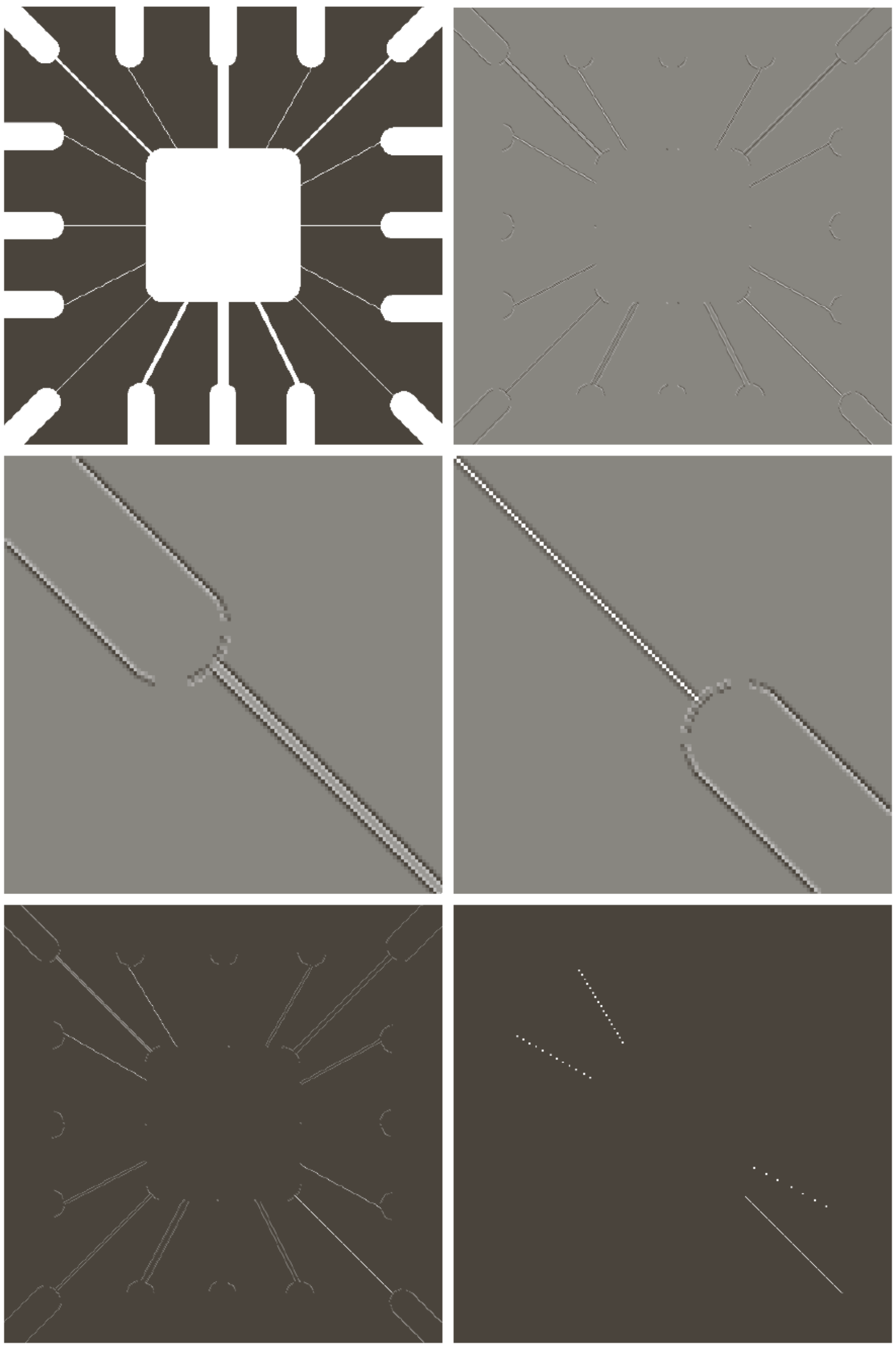
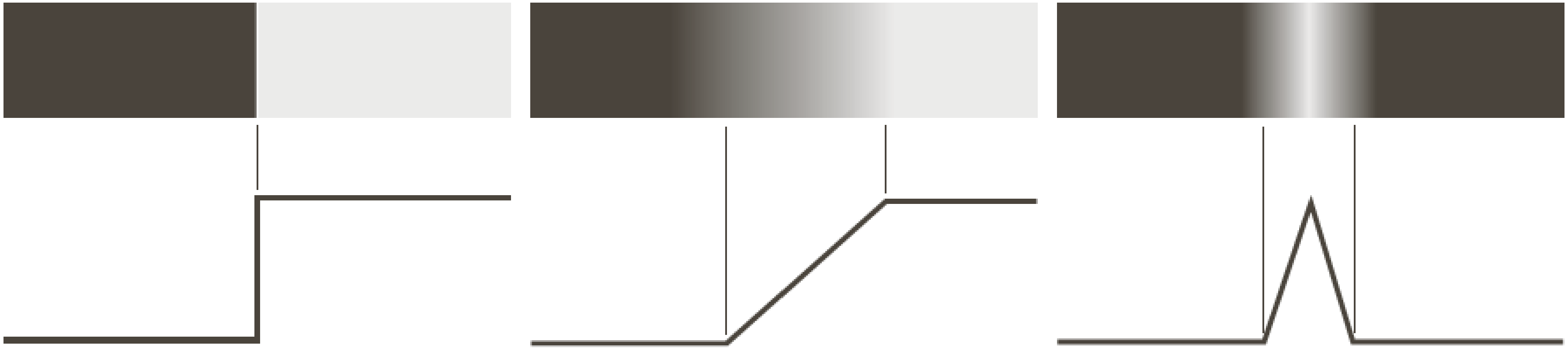


FIGURE 10.7
(a) Image of a wire-bond template.
(b) Result of processing with the $+45°$ line detector mask in Fig. 10.6.
(c) Zoomed view of the top left region of (b).
(d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Edge Detection

- Edges are pixels where the brightness function changes abruptly
- Edge models



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.

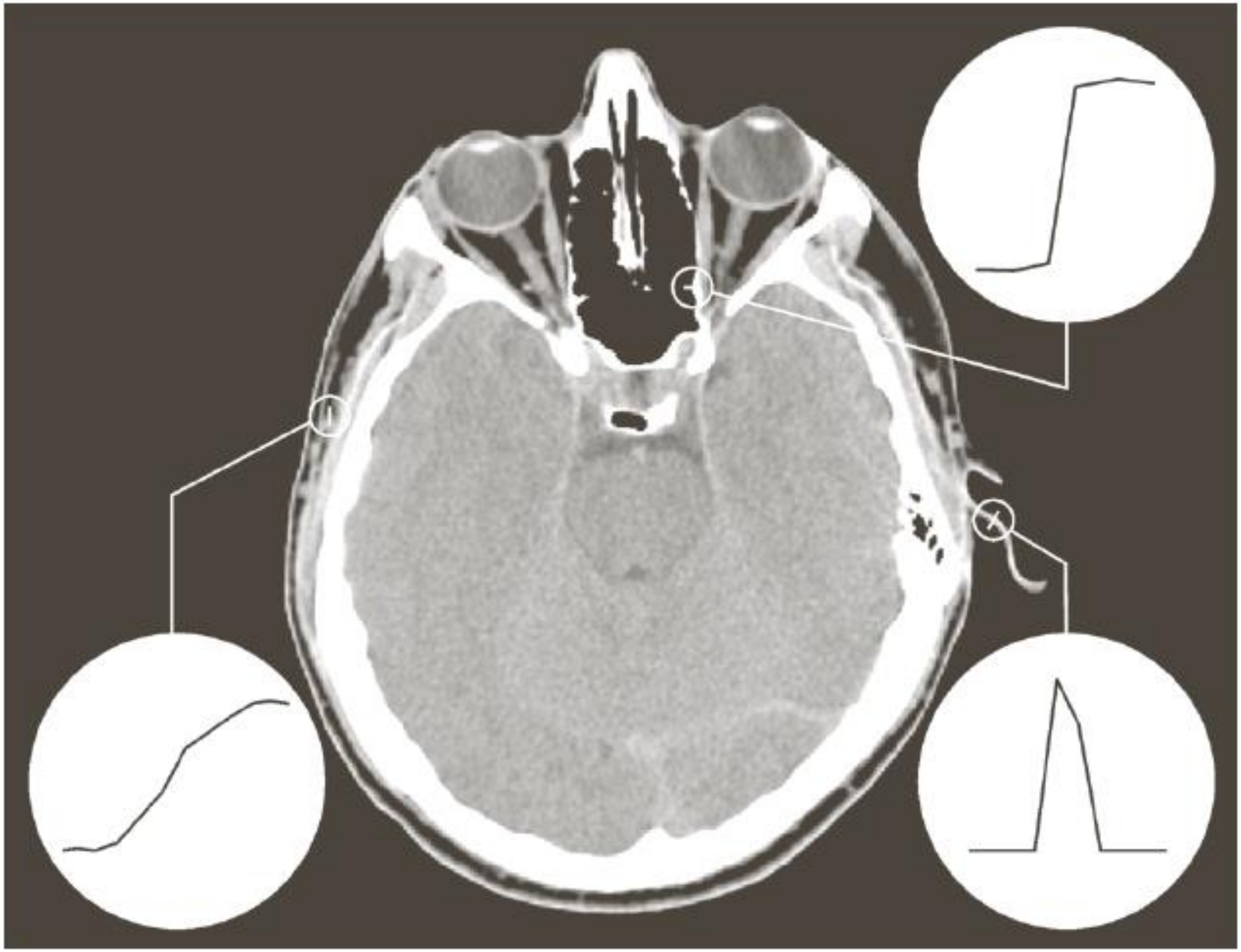
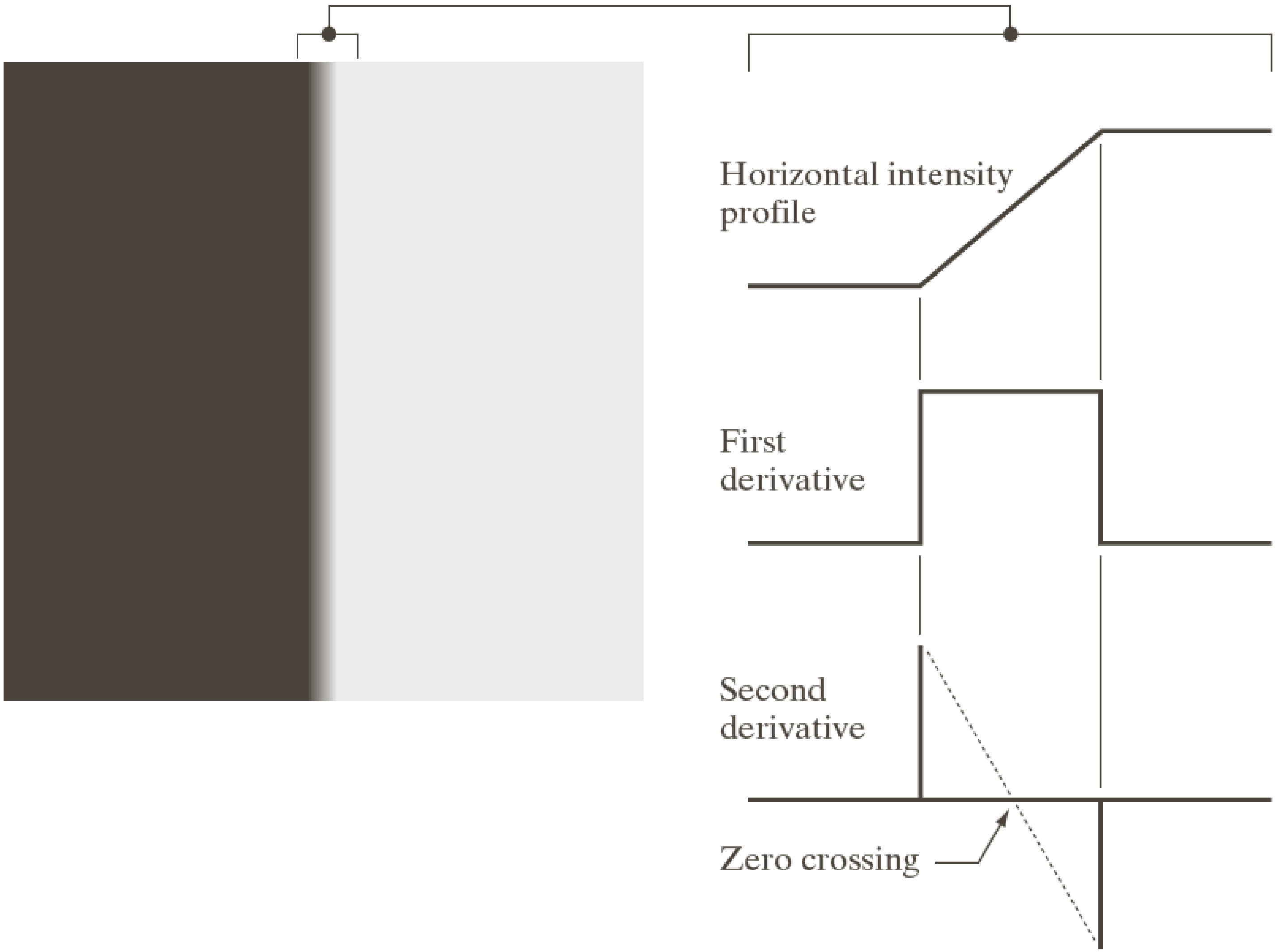


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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



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.

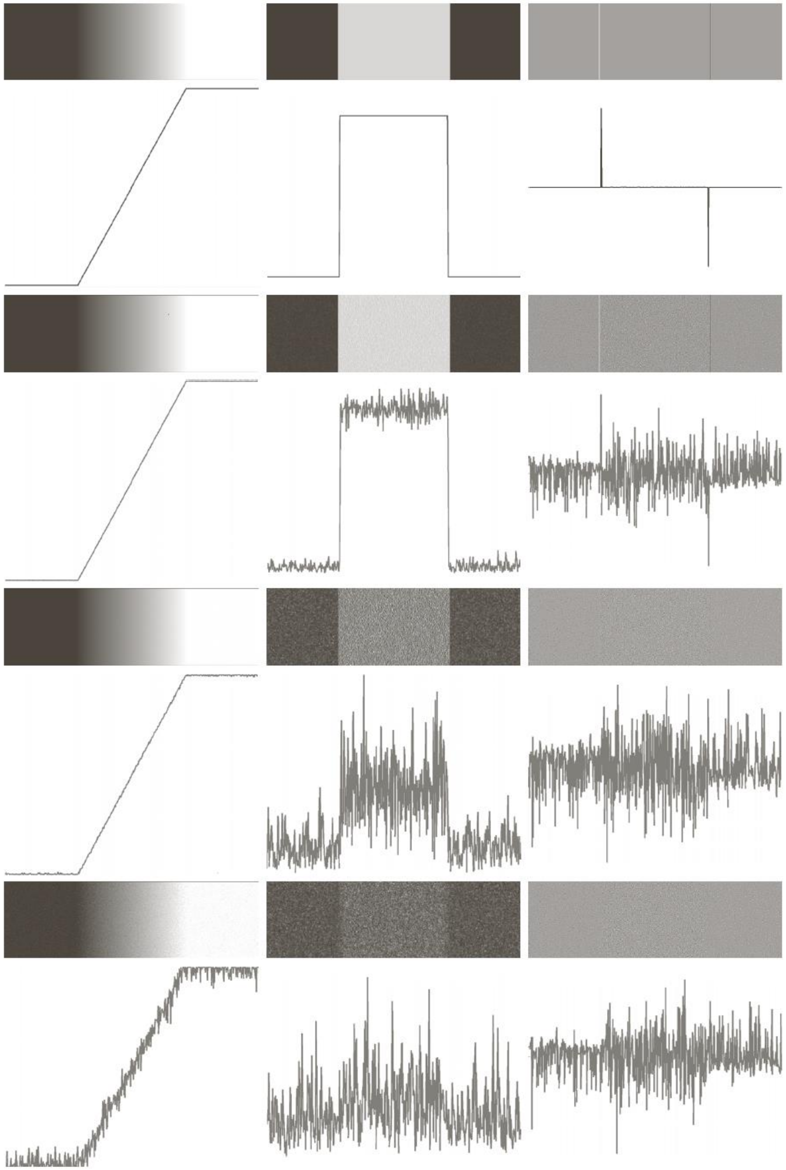


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

Basic Edge Detection by Using First-Order Derivative

$$\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of ∇f

$$M(x, y) = \mathrm{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$$

The direction of ∇f

$$\alpha(x, y) = \arctan(g_y / g_x)$$

The direction of the edge

$$\phi = \alpha - 90^\circ$$

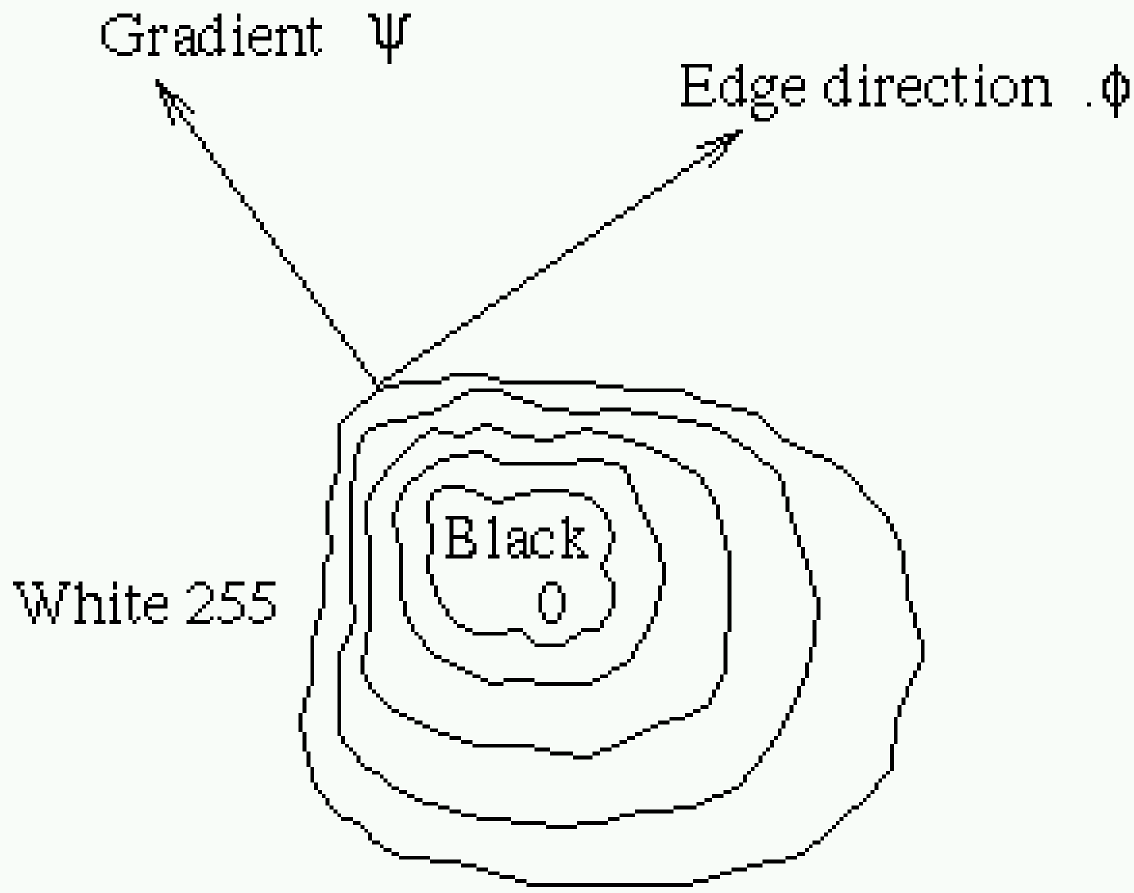
Basic Edge Detection by Using First-Order Derivative

Edge normal: $\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$

Edge unit normal: $\nabla f \,/\, \text{mag}(\nabla f)$

In practice,sometimes the magnitude is approximated by

$$\text{mag}(\nabla f)=\left|\frac{\partial f}{\partial x}\right|+\left|\frac{\partial f}{\partial y}\right| \text{ or } \text{mag}(\nabla f)=\max\left(\left|\frac{\partial f}{\partial x}\right|,\left|\frac{\partial f}{\partial y}\right|\right)$$


Gradient ψ
Edge direction ϕ
Black
0
White 255

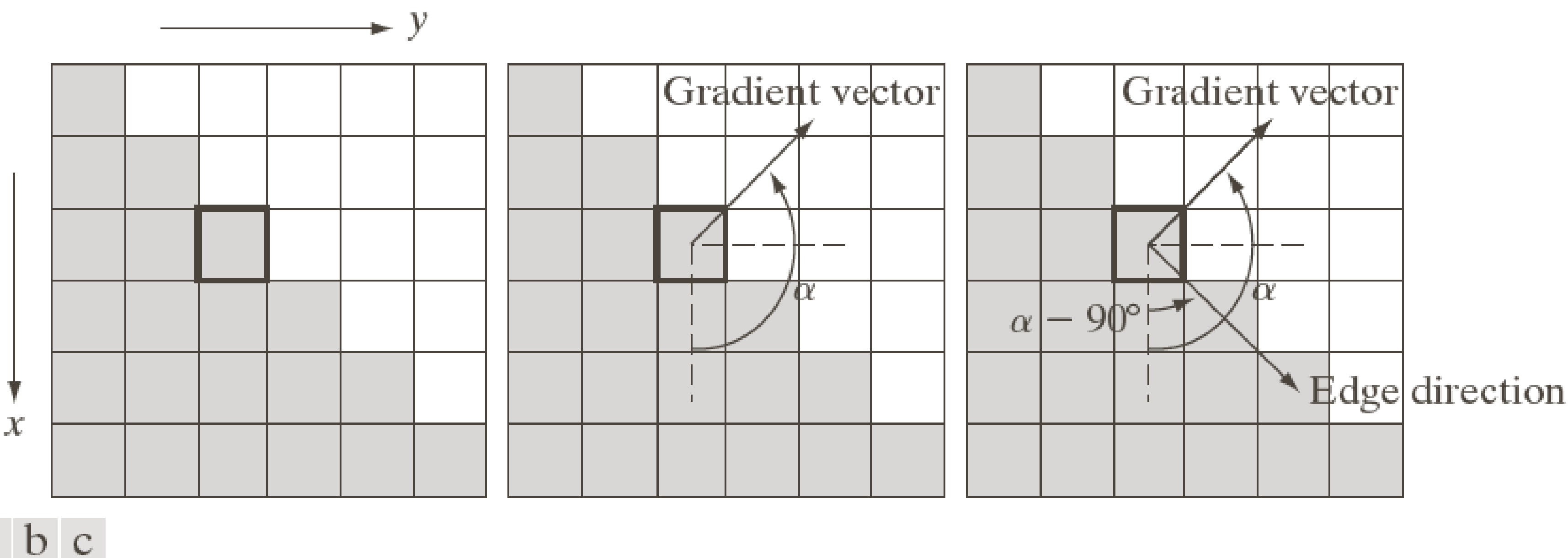


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

−1
1

−1	1

a b

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

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
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

a
b c
d e
f g

FIGURE 10.14 A 3 × 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.

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

a b
c d

FIGURE 10.16 (a) Original image of size 834×1114 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, $|g_x| + |g_y|$.

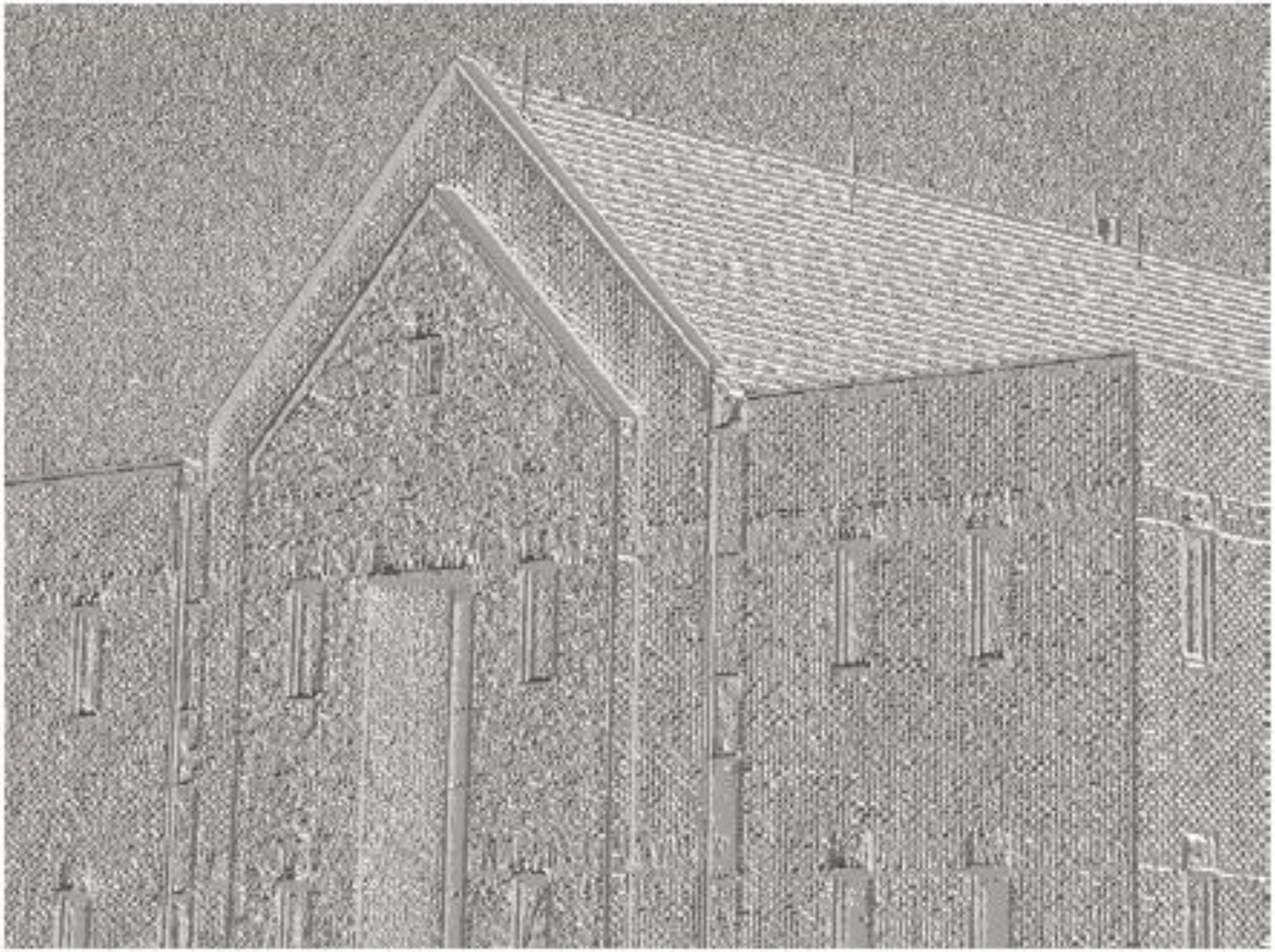


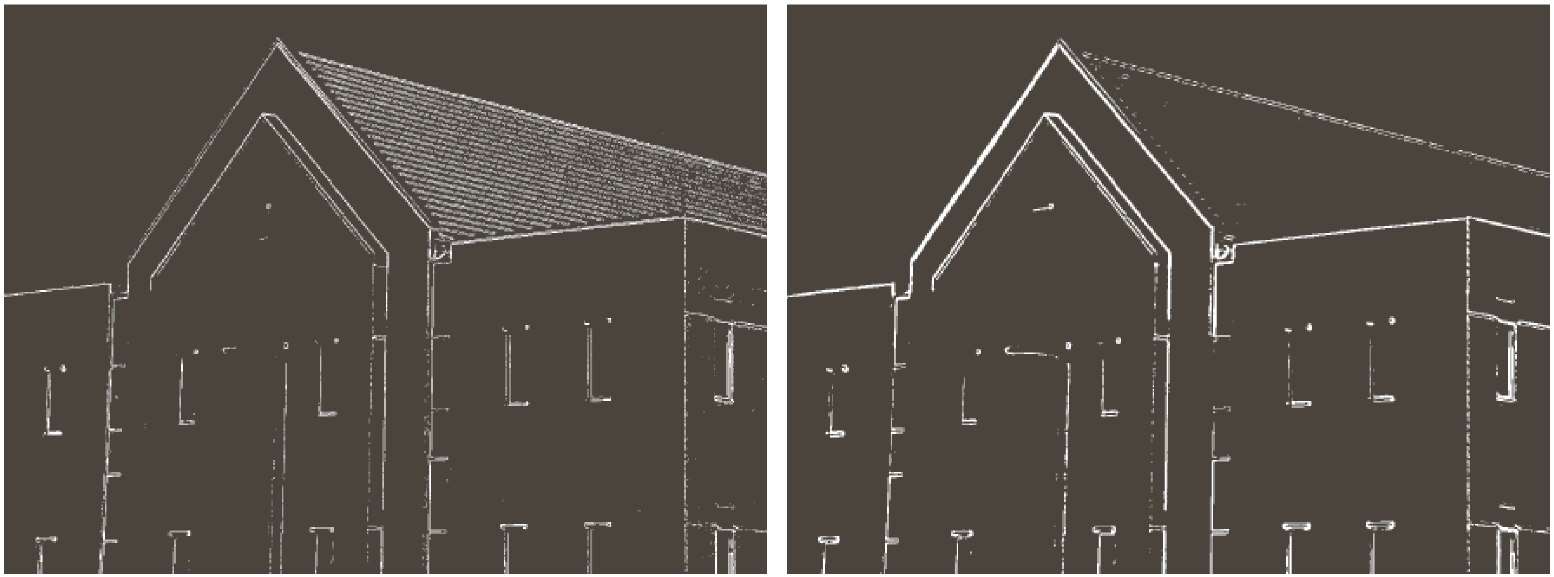
FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

a b
c d

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

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).



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

Advanced Techniques for Edge Detection

- The Marr-Hildreth edge detector

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}, \ \sigma : \text{space constant.}$$

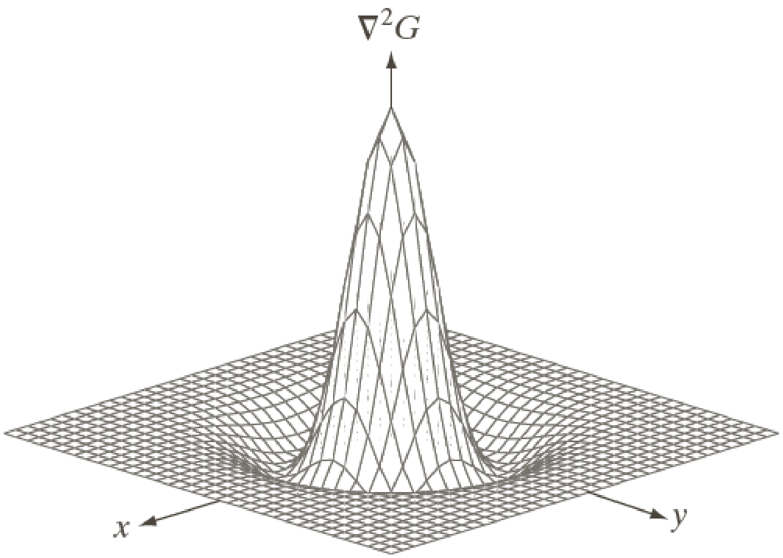
Laplacian of Gaussian (LoG)

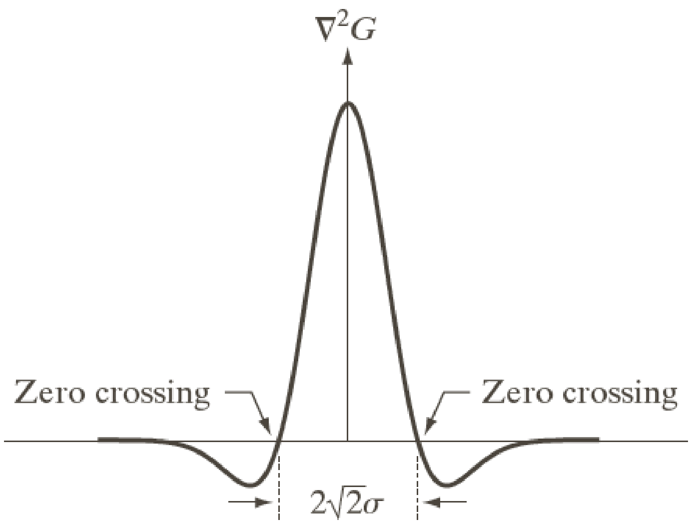
$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2}$$

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

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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





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) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

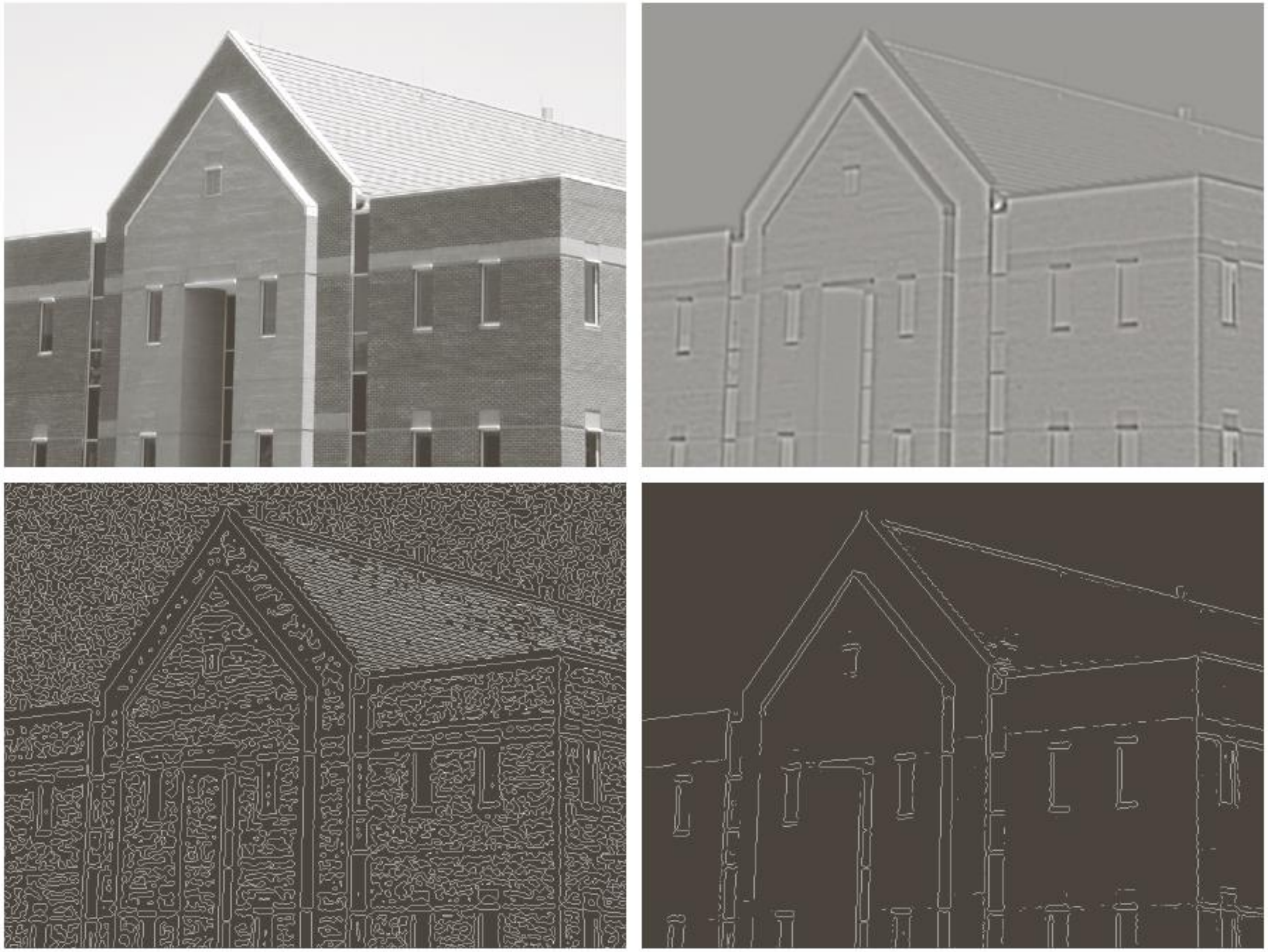
Marr-Hildreth Algorithm

1. Filter the input image with an nxn Gaussian lowpass filter. n is the smallest odd integer greater than or equal to 6σ

2. Compute the Laplacian of the image resulting from Step 1.

3. Find the zero crossing of the image from Step 2.

$$g(x,y) = \nabla^2 \left[G(x,y) \star f(x,y) \right]$$



a b
c d

FIGURE 10.22 (a) Original image of size 834 × 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.

The Canny Edge Detector

The Objectives

1. **Low error rate**
   All edges should be found and there should be no spurious responses.

2. **Edge points should be well localized**
   The edges located must be as close as possible to the true edges.

3. **Single edge point response**
   The number of local maxima around the true edge should be minimum.

The Canny Edge Detector: Algorithm (1)

Let $f(x, y)$ denote the input image and $G(x, y)$ denote the Gaussian function:

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

We form a smoothed image, $f_s(x, y)$ by convolving G and f:

$$f_s(x, y) = G(x, y) \star f(x, y)$$

The Canny Edge Detector: Algorithm(2)

Compute the gradient magnitude and direction (angle):

$$M(x,y) = \sqrt{{g_x}^2 + {g_y}^2}$$

and

$$\alpha(x,y) = \arctan(g_y / g_x)$$

where $g_x = \partial f_s / \partial x$ and $g_y = \partial f_s / \partial y$

Note: any of the filter mask pairs in Fig.10.14 can be used to obtain g_x and g_y

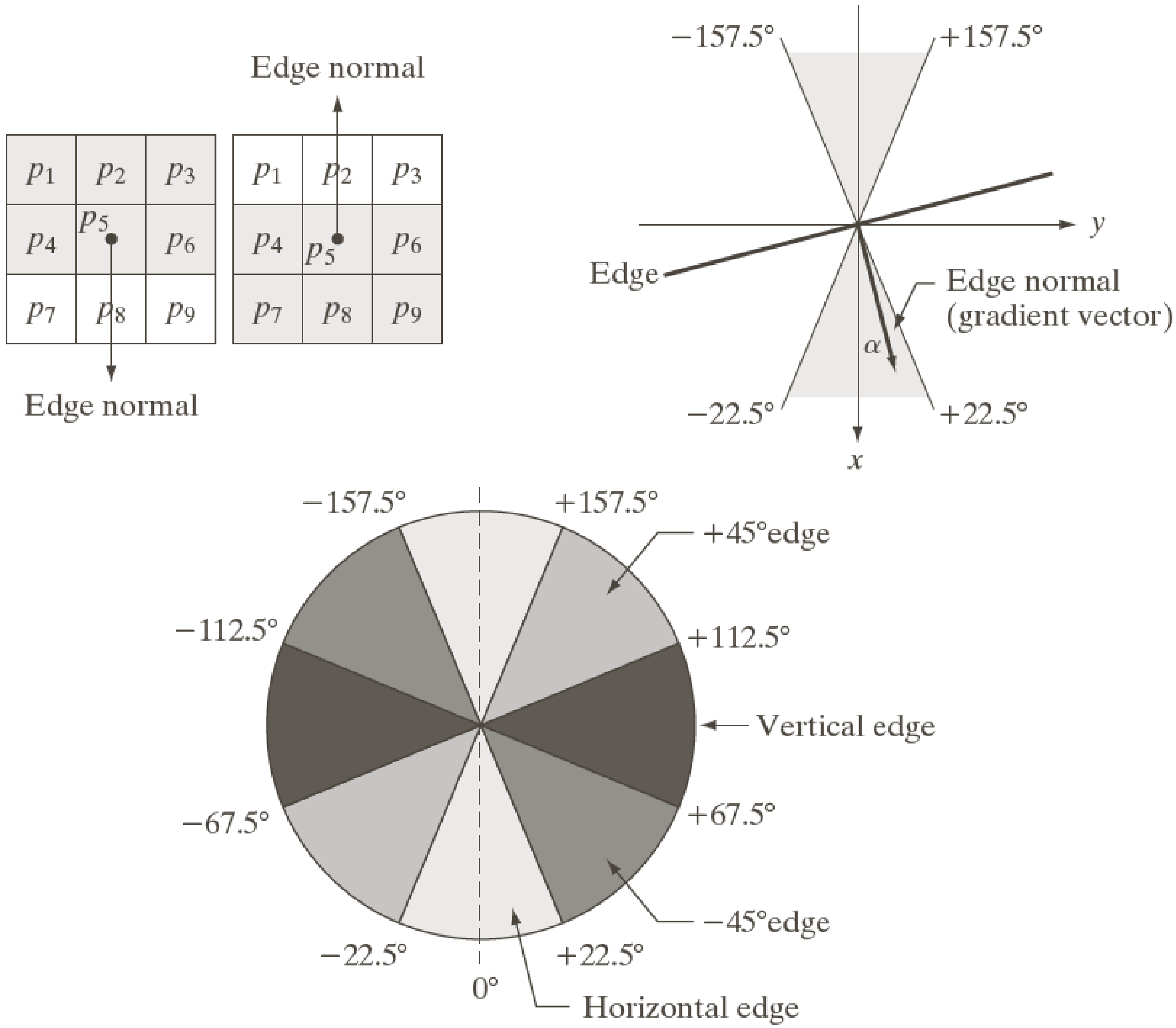
The Canny Edge Detector: Algorithm(3)

The gradient $M(x, y)$ typically contains wide ridge around local maxima. Next step is to thin those ridges.

Nonmaxima suppression:

Let d_1, d_2, d_3, and d_4 denote the four basic edge directions for a 3×3 region: horizontal, -45°, vertical, +45°, respectively.

1. Find the direction d_k that is closest to $\alpha(x, y)$.
2. If the value of $M(x, y)$ is less than at least one of its two neighbors along d_k, let $g_N(x, y) = 0$ (suppression); otherwise, let $g_N(x, y) = M(x, y)$



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3×3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3×3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

The Canny Edge Detector: Algorithm(4)

The final operation is to threshold $g_N(x, y)$ to reduce false edge points.

Hysteresis thresholding:

$$g_{NH}(x, y) = g_N(x, y) \geq T_H$$

$$g_{NL}(x, y) = g_N(x, y) \geq T_L$$

and

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$$

The Canny Edge Detector: Algorithm(5)

Depending on the value of T_H, the edges in $g_{NH}(x, y)$ typically have gaps. Longer edges are formed using the following procedure:

(a). Locate the next unvisited edge pixel, p, in $g_{NH}(x, y)$.

(b). Mark as valid edge pixel all the weak pixels in $g_{NL}(x, y)$ that are connected to p using 8-connectivity.

(c). If all nonzero pixel in $g_{NH}(x, y)$ have been visited go to step (d), esle return to (a).

(d). Set to zero all pixels in $g_{NL}(x, y)$ that were not marked as valid edge pixels.

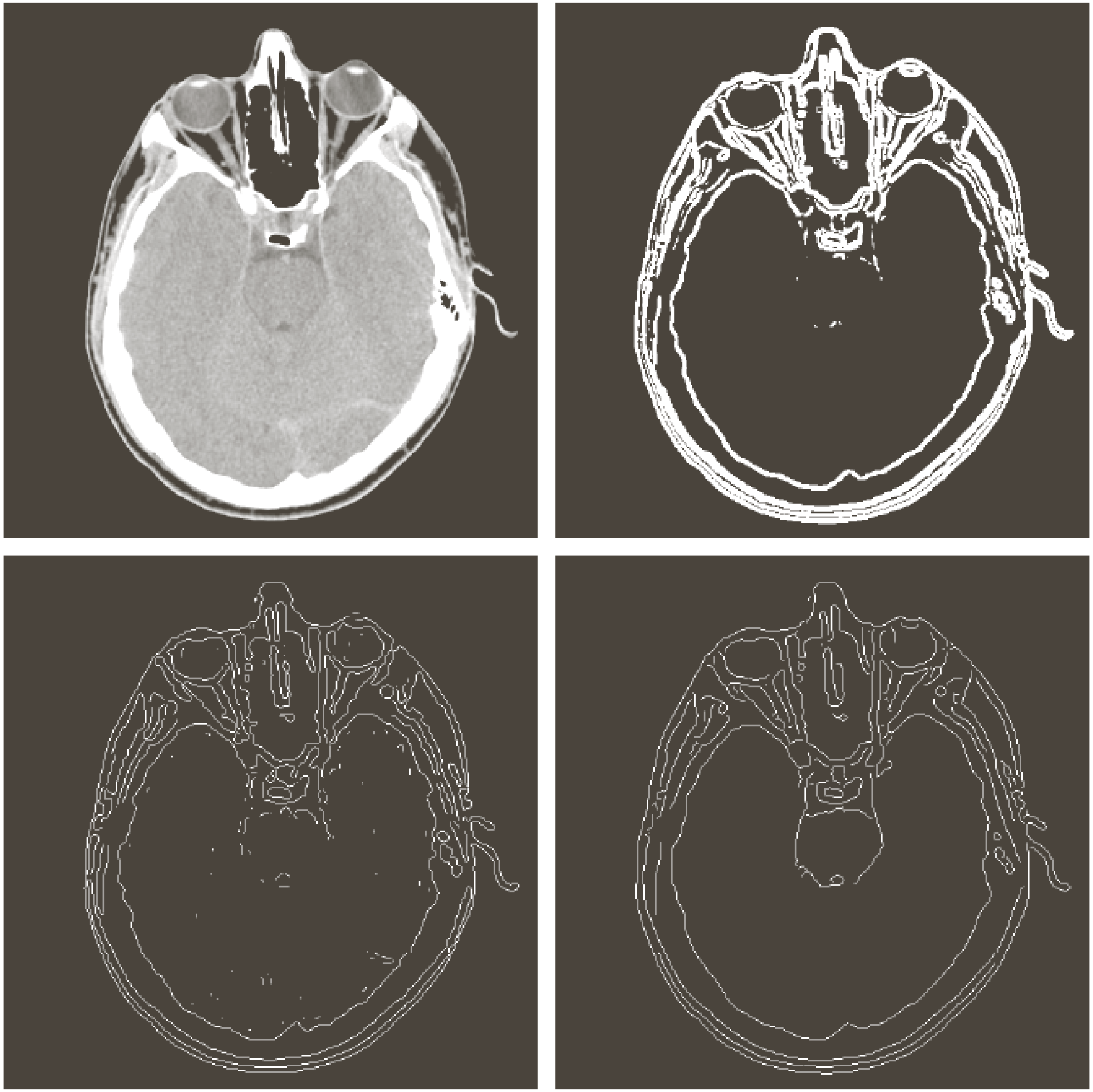
The Canny Edge Detection: Summary

- Smooth the input image with a Gaussian filter
- Compute the gradient magnitude and angle images
- Apply nonmaxima suppression to the gradient magnitude image
- Use double thresholding and connectivity analysis to detect and link edges

a b
c d

FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

$T_L = 0.04; T_H = 0.10; \sigma = 4$ and a mask of size 25×25



a	b
c	d

FIGURE 10.26 (a) Original head CT image of size 512 × 512 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

$T_L = 0.05; T_H = 0.15; \sigma = 2$ and a mask of size 13×13

Edge Linking and Boundary Detection

- Edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries

- Three approaches to edge linking

  Local processing

  Regional processing

  Global processing

Local Processing

- Analyze the characteristics of pixels in a small neighborhood about every point (x,y) that has been declared an edge point

- All points that are similar according to predefined criteria are linked, forming an edge of pixels.

Establishing similarity: (1) the strength (magnitude) and (2) the direction of the gradient vector.

A pixel with coordinates (s,t) in S_{xy} is linked to the pixel at (x,y) if both magnitude and direction criteria are satisfied.

Local Processing

Let S_{xy} denote the set of coordinates of a neighborhood centered at point (x, y) in an image. An edge pixel with coordinate (s,t) in S_{xy} is similar in *magnitude* to the pixel at (x, y) if

$$\left|M(s,t) - M(x,y)\right| \le E$$

An edge pixel with coordinate (s,t) in S_{xy} is similar in *angle* to the pixel at (x, y) if

$$\left|\alpha(s,t) - \alpha(x,y)\right| \le A$$

Local Processing: Steps (1)

1. Compute the gradient magnitude and angle arrays, M(x,y) and $\alpha(x, y)$, of the input image f(x,y)

2. Form a binary image, g, whose value at any coordinate (x,y) is given by

$$g(x,y) = \begin{cases} 1 & \text{if } M(x,y) > T_M \text{ and } \alpha(x,y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

T_M : threshold $\quad$ A: specified angle direction

T_A: a "band" of acceptable directions about A

Local Processing: Steps (2)

3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K.

4. To detect gaps in any other direction, rotate g by this angle and apply the horizontal scanning procedure in step 3.

FIGURE 10.27 (a) A 534 × 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.)

Regional Processing

- The location of regions of interest in an image are known or can be determined

- Polygonal approximations can capture the essential shape features of a region while keeping the representation of the boundary relatively simple

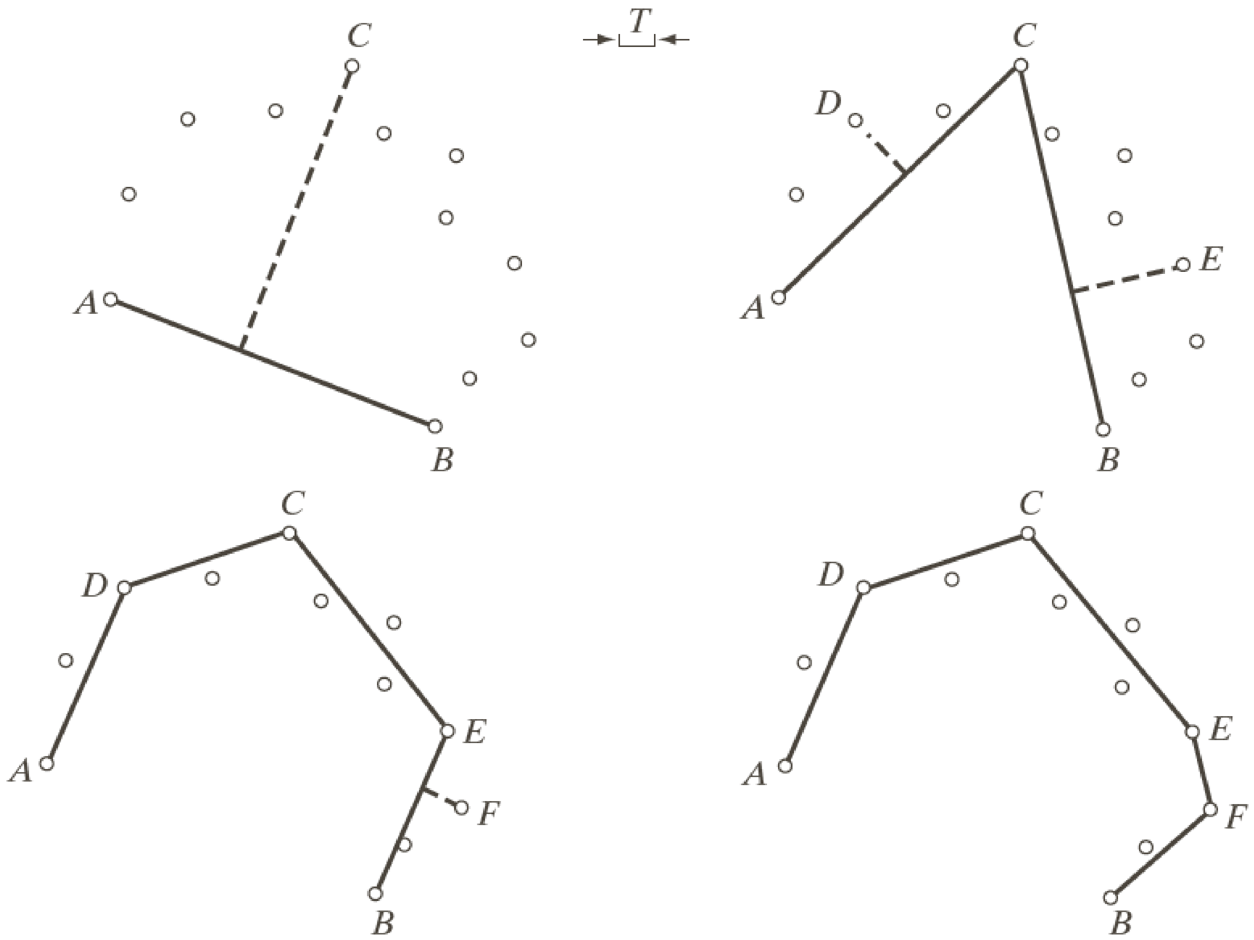


FIGURE 10.28
Illustration of the iterative polygonal fit algorithm.

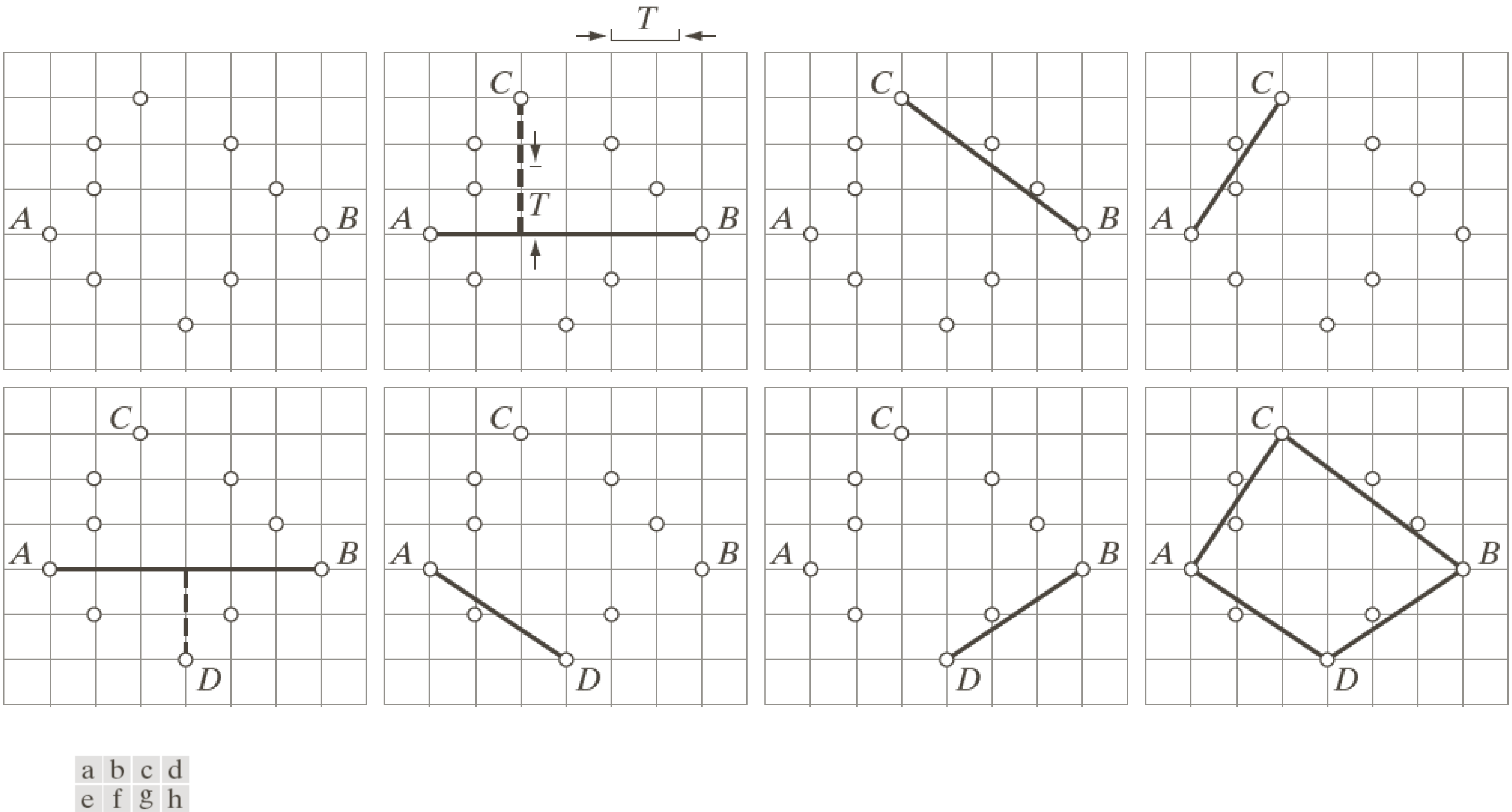


FIGURE 10.29 (a) A set of points in a clockwise path (the points labeled A and B were chosen as the starting vertices). (b) The distance from point C to the line passing through A and B is the largest of all the points between A and B and also passed the threshold test, so C is a new vertex. (d)–(g) Various stages of the algorithm. (h) The final vertices, shown connected with straight lines to form a polygon. Table 10.1 shows step-by-step details.

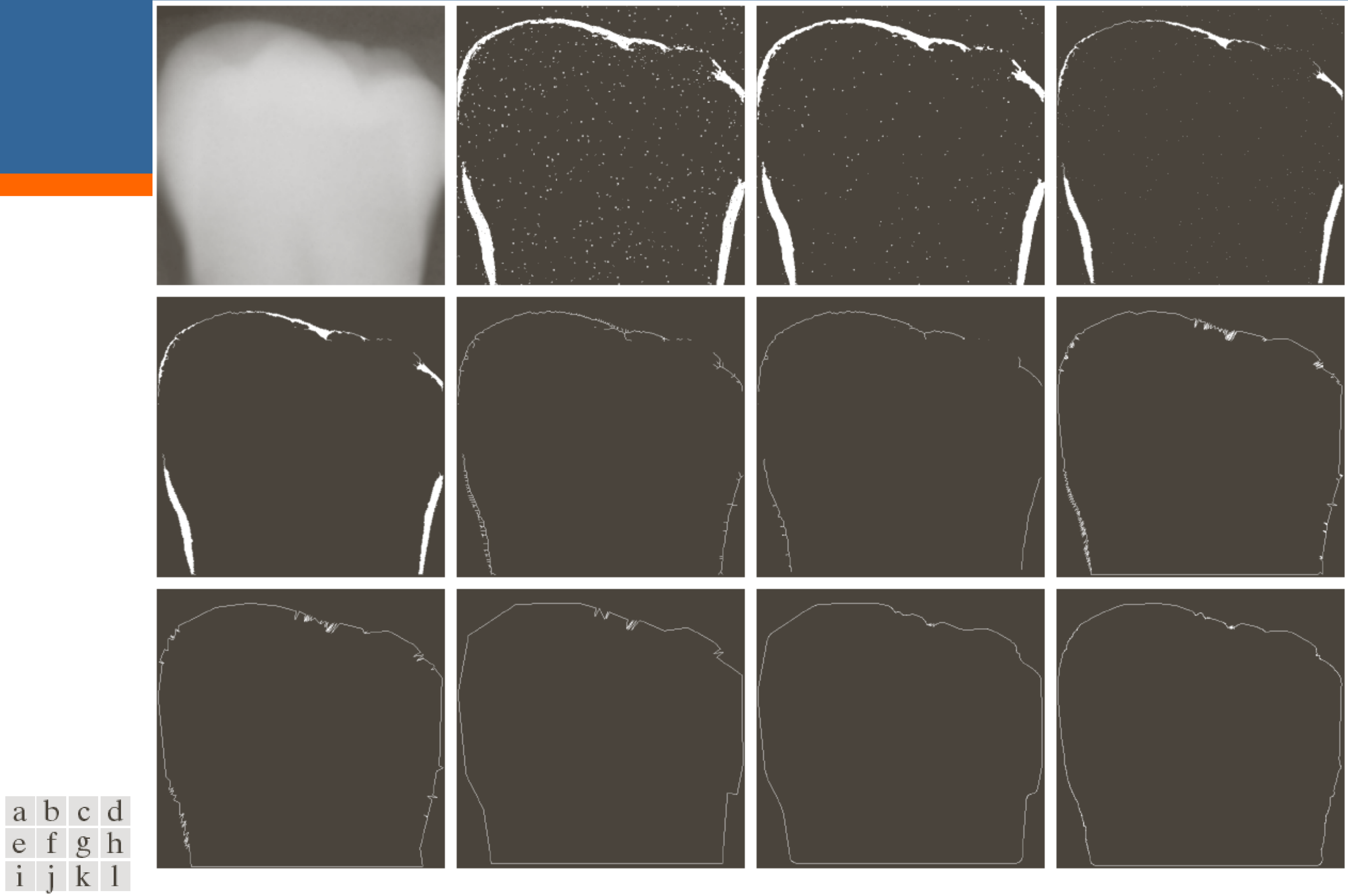
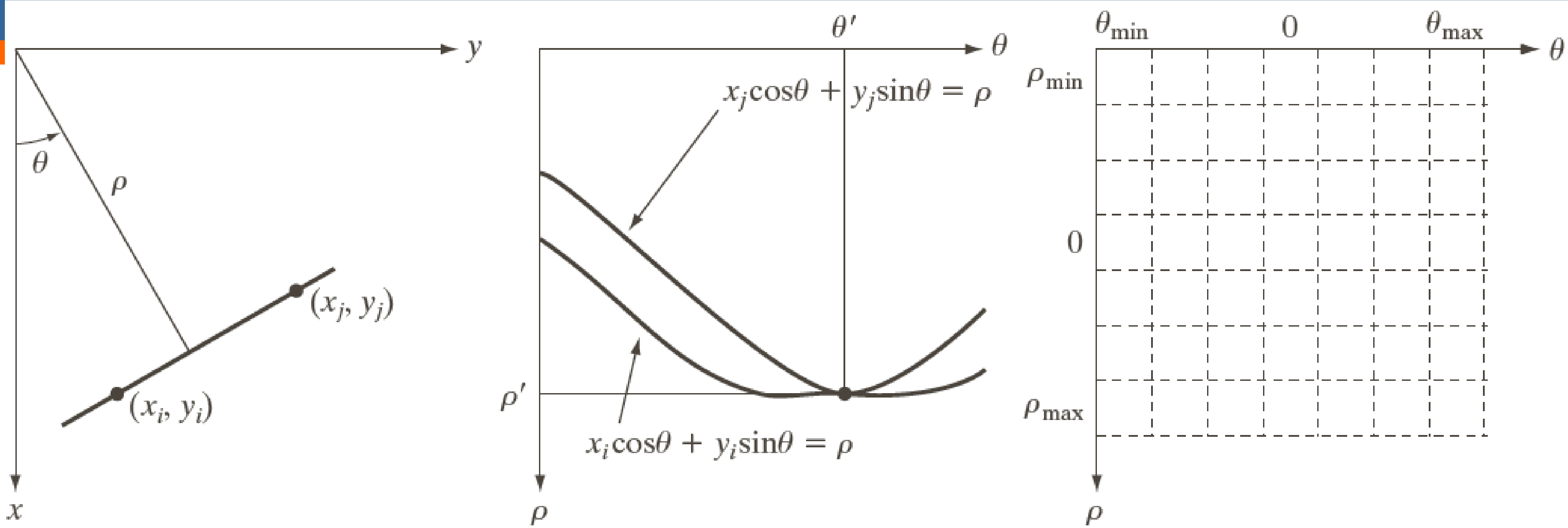


FIGURE 10.30 (a) A 550 × 566 X-ray image of a human tooth. (b) Gradient image. (c) Result of majority filtering. (d) Result of morphological shrinking. (e) Result of morphological cleaning. (f) Skeleton. (g) Spur reduction. (h)–(j) Polygonal fit using thresholds of approximately 0.5%, 1%, and 2% of image width ($T = 3$, 6, and 12). (k) Boundary in (j) smoothed with a 1-D averaging filter of size 1×31 (approximately 5% of image width). (l) Boundary in (h) smoothed with the same filter.

Global Processing Using the Hough Transform

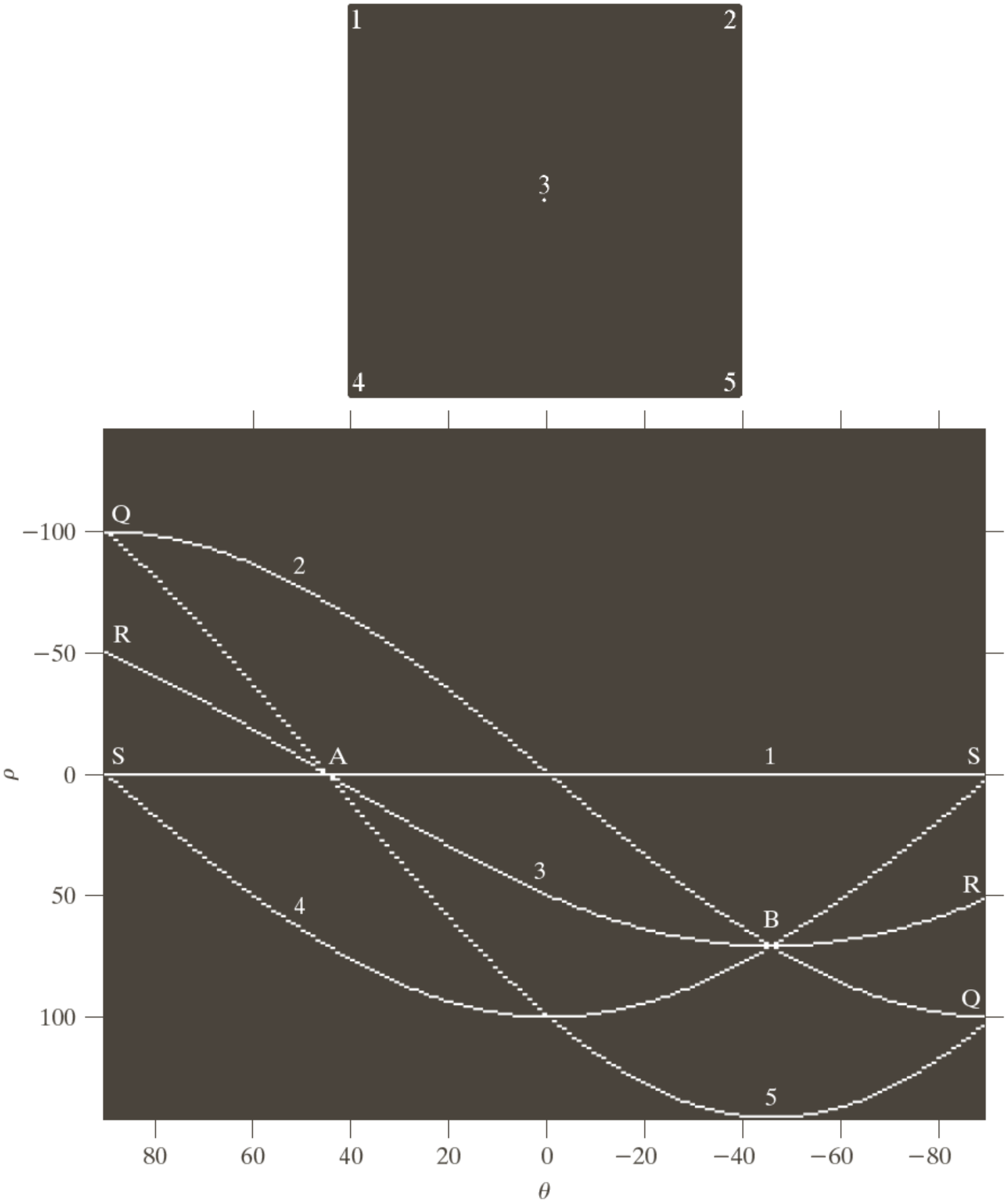
- "**The Hough transform** is a general technique for identifying the locations and orientations of certain types of features in a digital image. Developed by Paul Hough in 1962 and patented by IBM, the transform consists of parameterizing a description of a feature at any given location in the original image's space. A mesh in the space defined by these parameter is then generated, and at each mesh point a value is accumulated, indicating how well an object generated by the parameters defined at that point fits the given image. Mesh points that accumulate relatively larger values then describe features that may be projected back onto the image, fitting to some degree the features actually present in the image."

http://planetmath.org/encyclopedia/HoughTransform.html



a b c

FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') 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.



a
b

FIGURE 10.33 (a) Image of size 101 × 101 pixels, containing five points. (b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

Edge-linking Based on the Hough Transform

1. Obtain a binary edge image

2. Specify subdivisions in $\rho\theta - plane$

3. Examine the counts of the accumulator cells for high pixel concentrations

4. Examine the relationship between pixels in chosen cell

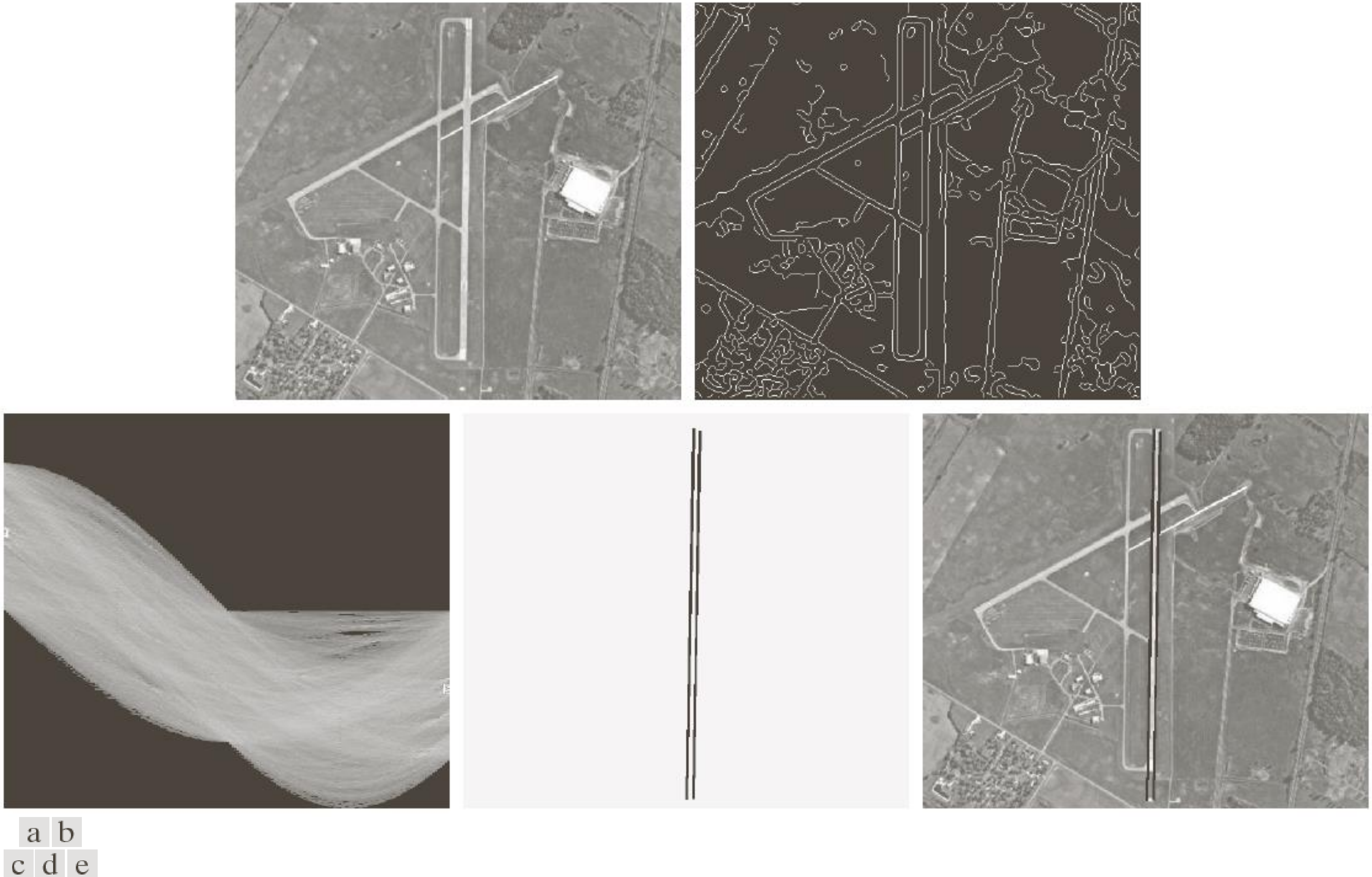


FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed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

Thresholding

$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)>T \quad \text{(object point)}\\ 0 & \text{if } f(x,y)\le T \quad \text{(background point)}\end{cases}$$

T : global thresholding

Multiple thresholding

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

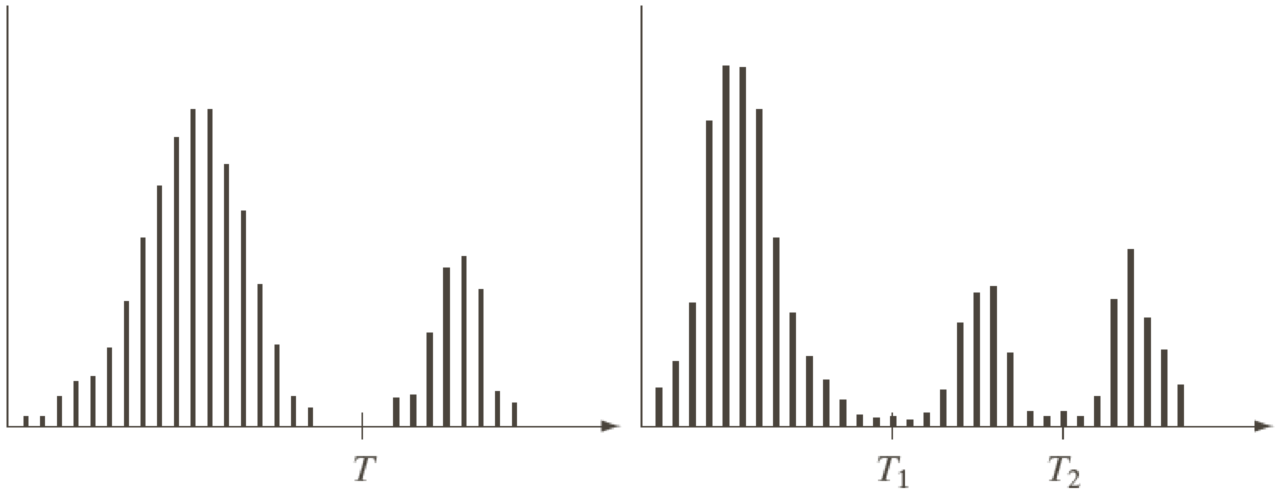


FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

The Role of Noise in Image Thresholding

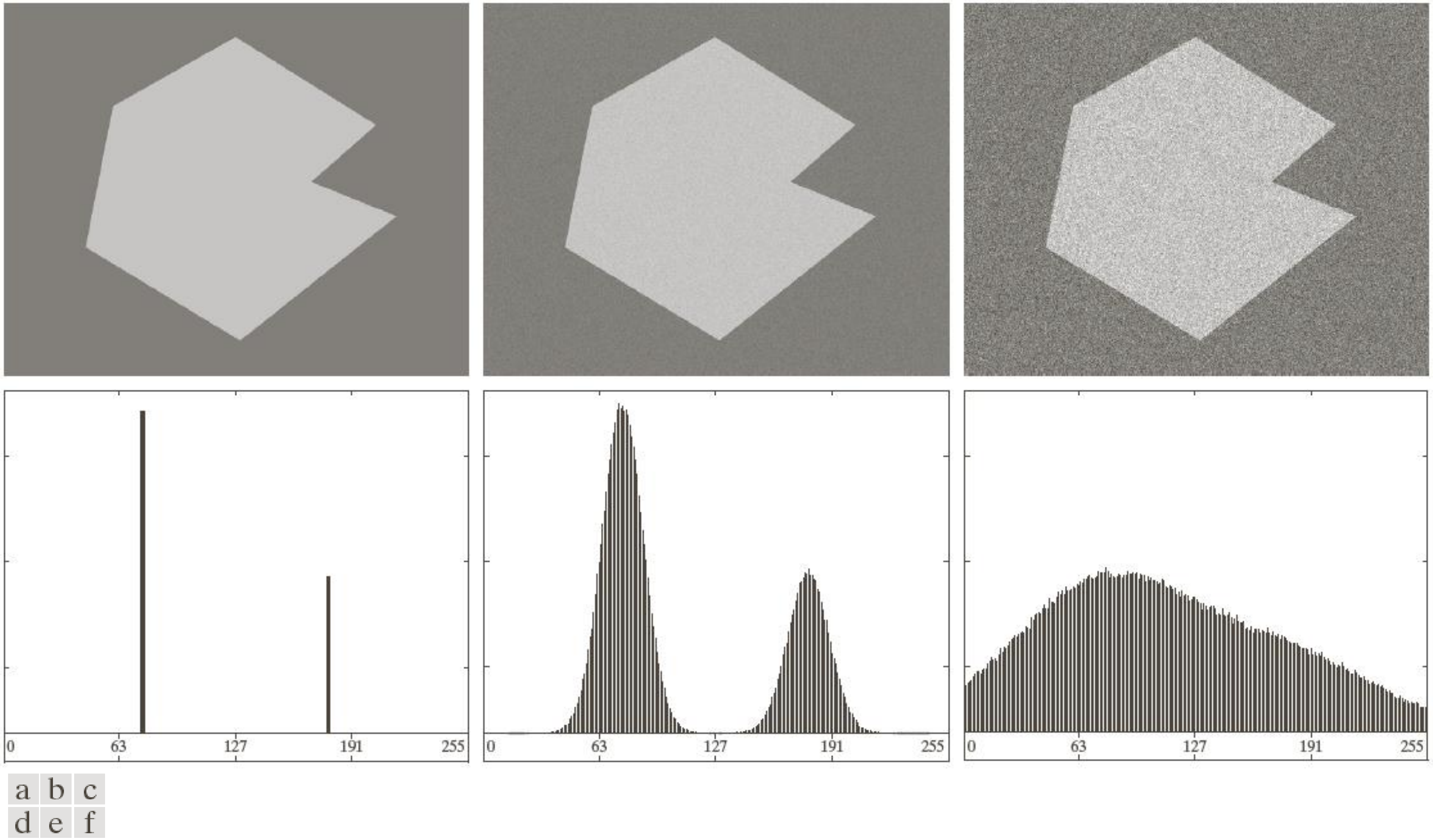


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

The Role of Illumination and Reflectance

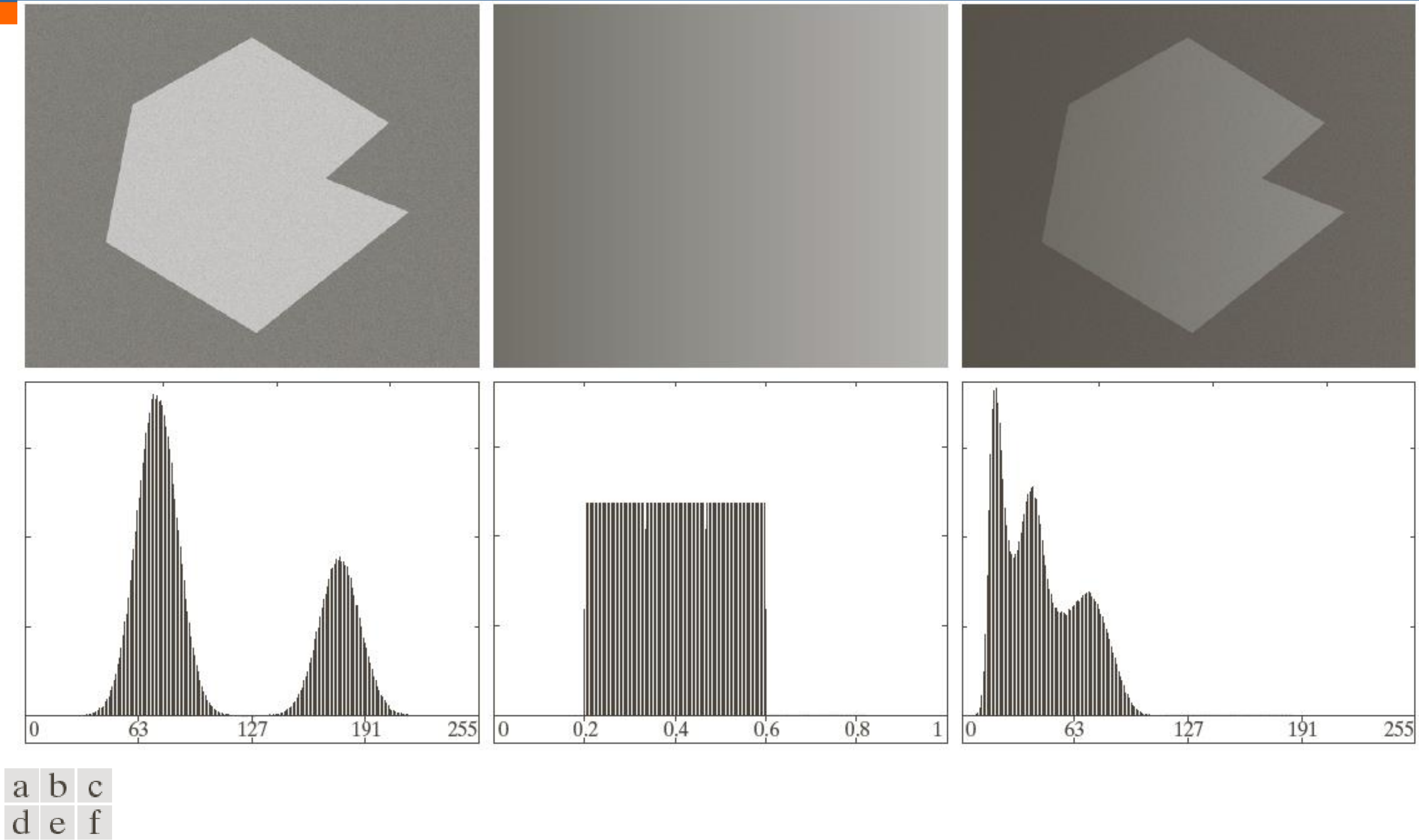


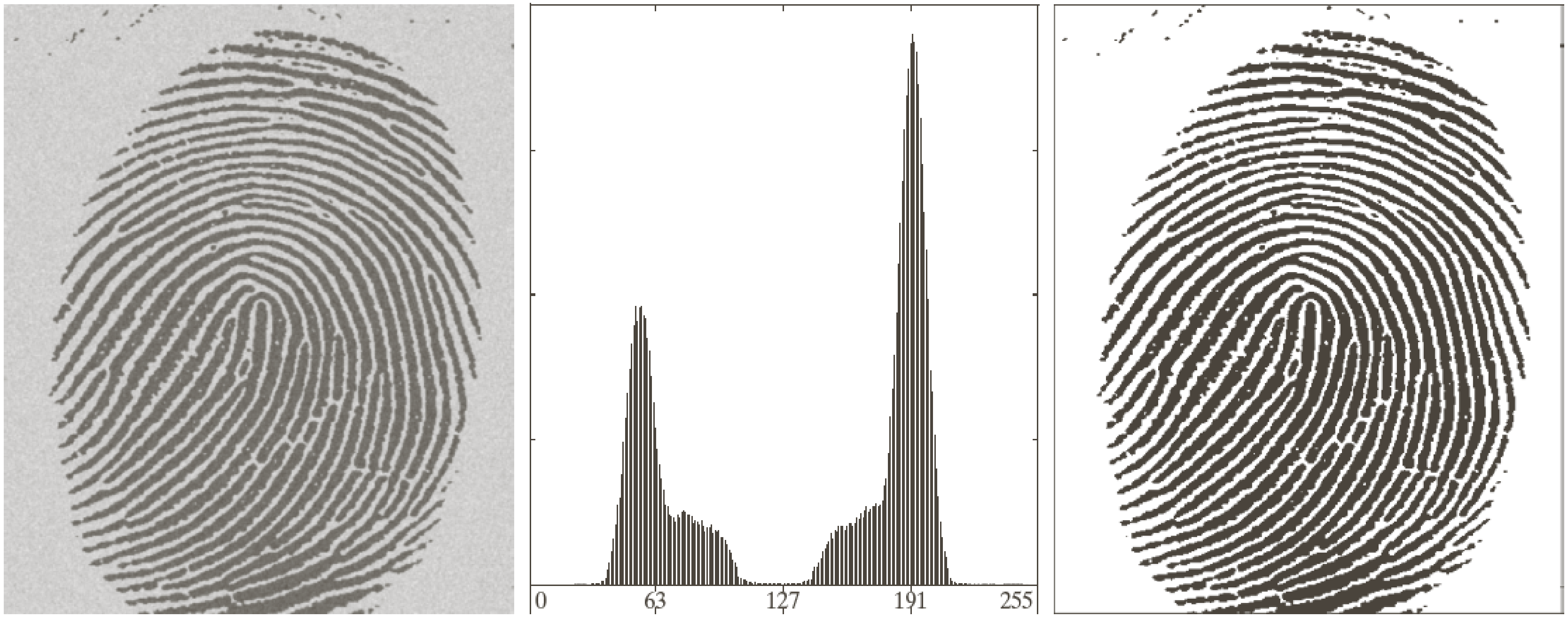
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Basic Global Thresholding

1. Select an initial estimate for the global threshold, T.
2. Segment the image using T. It will produce two groups of pixels: G1 consisting of all pixels with intensity values > T and G2 consisting of pixels with values $\leq$ T.
3. Compute the average intensity values m1 and m2 for the pixels in G1 and G2, respectively.
4. Compute a new threshold value.

$$T = \frac{1}{2}\left(m1 + m2\right)$$

5. Repeat Steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT .



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.)

Optimum Global Thresholding Using Otsu's Method

- Principle: maximizing the between-class variance

Let $\{0,\ 1,\ 2,\ ...,\ L\text{-}1\}$ denote the L distinct intensity levels in a digital image of size $M \times N$ pixels, and let n_i denote the number of pixels with intensity i.

$$p_i = n_i / MN \quad \text{and} \quad \sum_{i=0}^{L-1} p_i = 1$$

k is a threshold value, $C_1 \to [0, k]$, $C_2 \to [k+1, L\text{-}1]$

$$P_1(k) = \sum_{i=0}^{k} p_i \quad \text{and} \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

Optimum Global Thresholding Using Otsu's Method

The mean intensity value of the pixels assigned to class C_1 is

$$m_1(k) = \sum_{i=0}^{k} iP(i / C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$$

The mean intensity value of the pixels assigned to class C_2 is

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i / C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

$$P_1 m_1 + P_2 m_2 = m_G \quad \text{(Global mean value)}$$

Optimum Global Thresholding Using Otsu's Method

Between-class variance, σ_B^2 is defined as

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$
$$= P_1 P_2 (m_1 - m_2)^2$$

Optimum Global Thresholding Using Otsu's Method

The optimum threshold is the value, k*, that maximizes

$\sigma_B^2(k^*)$, $$\sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > k^* \\ 0 & \text{if } f(x,y) \le k^* \end{cases}$$

Separability measure $\eta = \dfrac{\sigma_B^2}{\sigma_G^2}$

Otsu's Algorithm: Summary

1. Compute the normalized histogram of the input image. Denote the components of the histogram by p_i, i=0, 1, …, L-1.
2. Compute the cumulative sums, $P_1(k)$, for k = 0, 1, …, L-1.
3. Compute the cumulative means, m(k), for k = 0, 1, …, L-1.
4. Compute the global intensity mean, m_G.
5. Compute the between-class variance, for k = 0, 1, …, L-1.

Otsu’s Algorithm: Summary

6. Obtain the Otsu’s threshold, k*.

7. Obtain the separability measure.

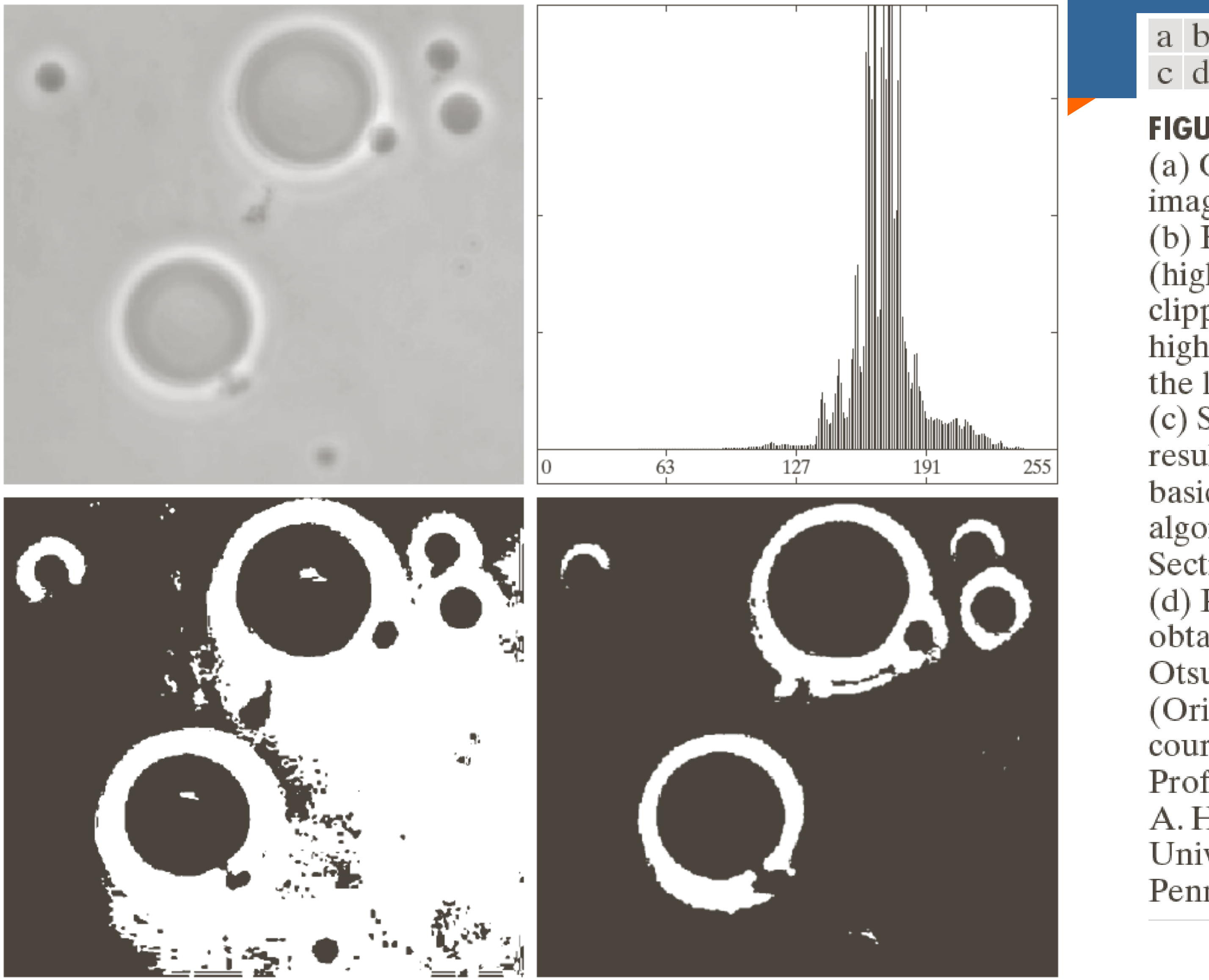
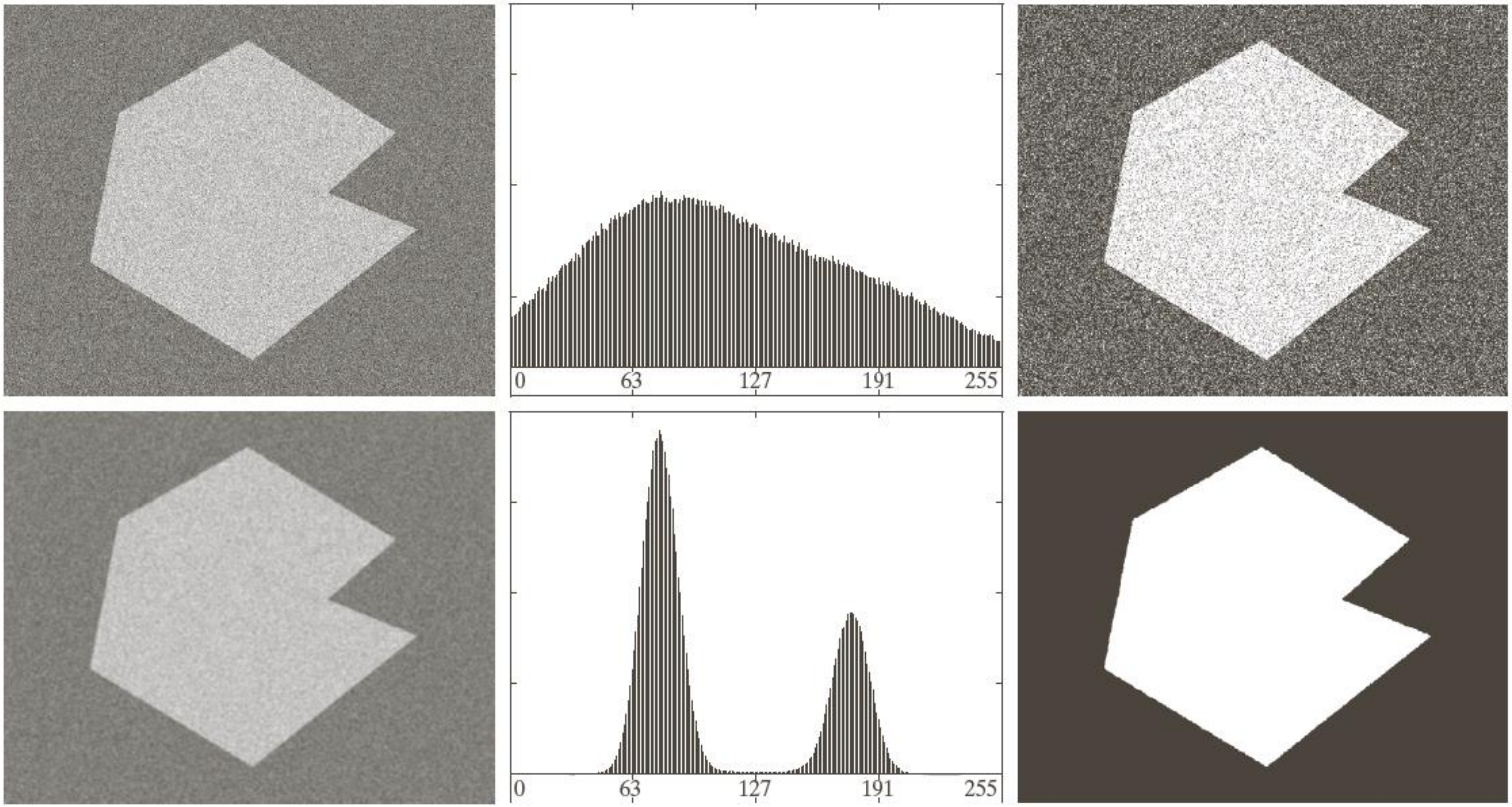


FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Using Image Smoothing to Improve Global Thresholding



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 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Using Edges to Improve Global Thresholding

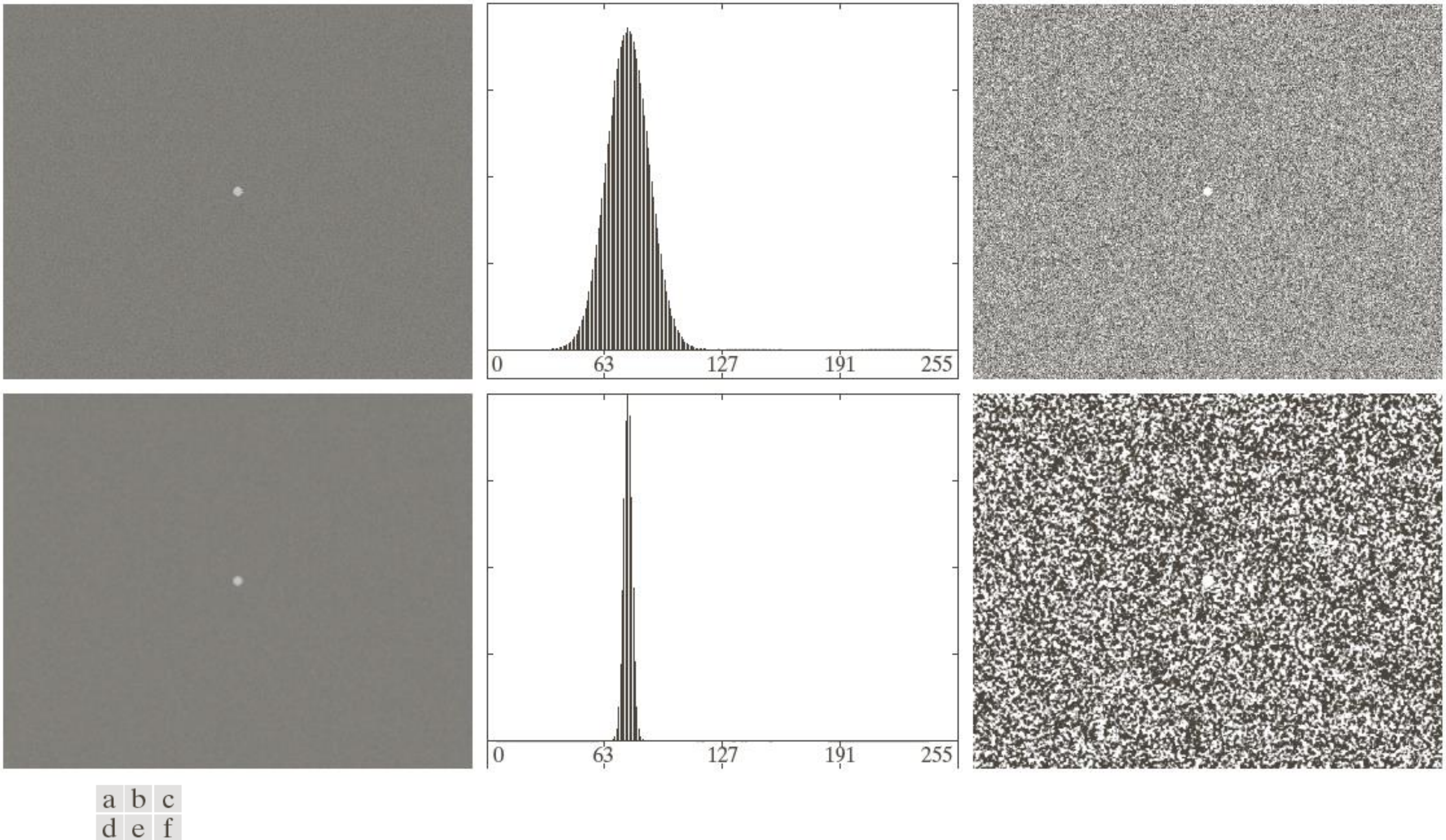


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

Using Edges to Improve Global Thresholding

1. Compute an edge image as either the magnitude of the gradient, or absolute value of the Laplacian of f(x,y)
2. Specify a threshold value T
3. Threshold the image and produce a binary image, which is used as a mask image; and select pixels from f(x,y) corresponding to “strong” edge pixels
4. Compute a histogram using only the chosen pixels in f(x,y)
5. Use the histogram from step 4 to segment f(x,y) globally

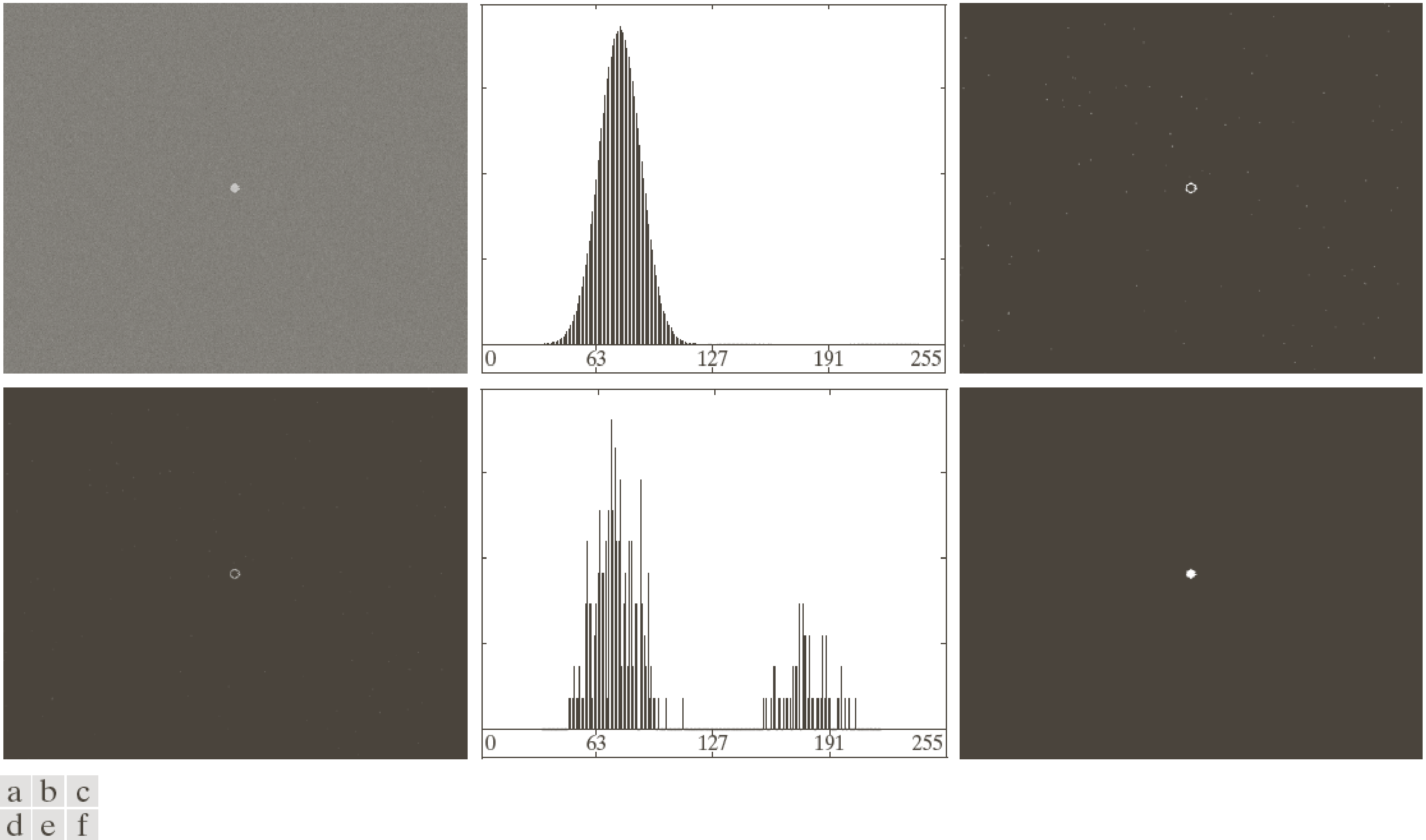


FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based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

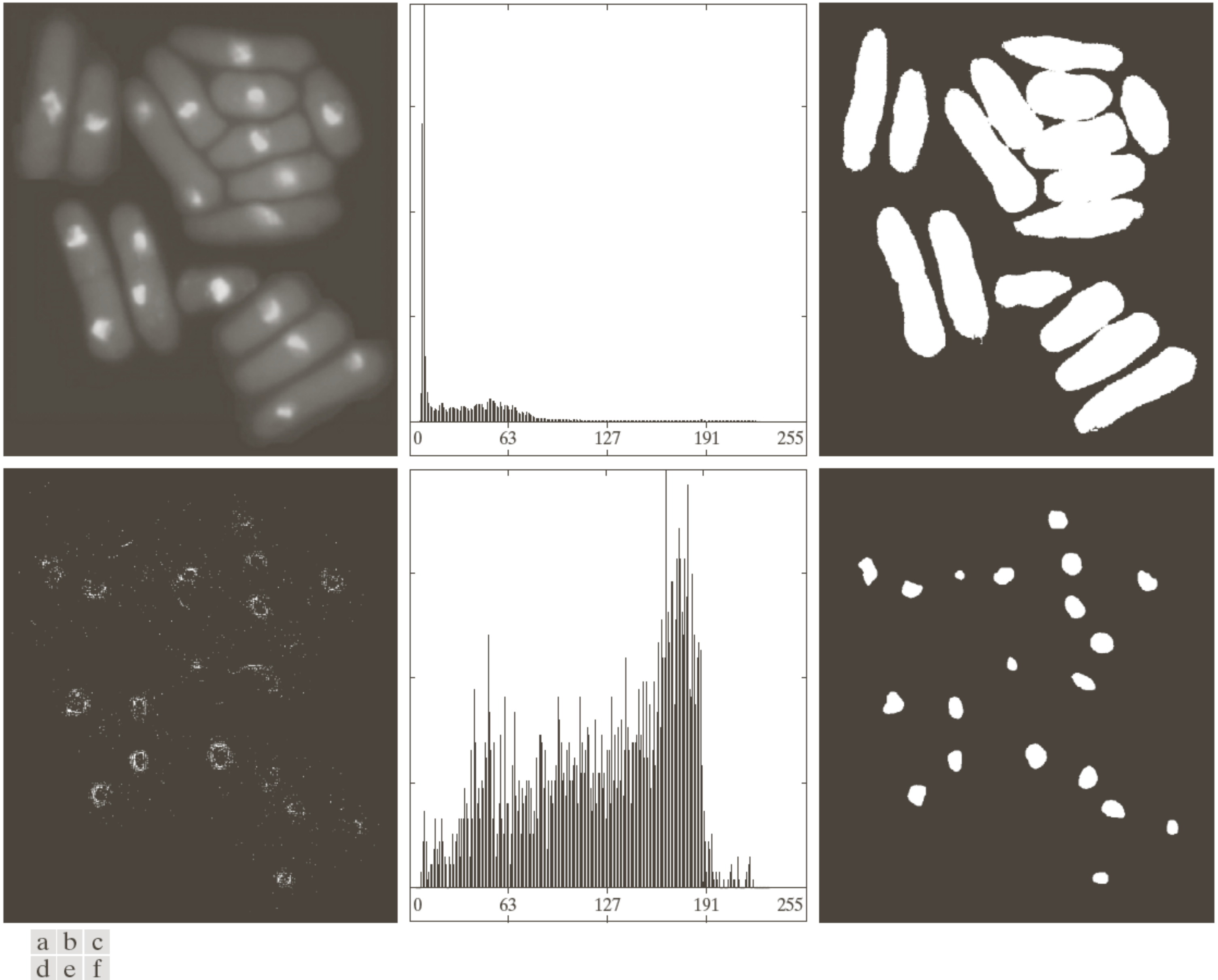


FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

Multiple Thresholds

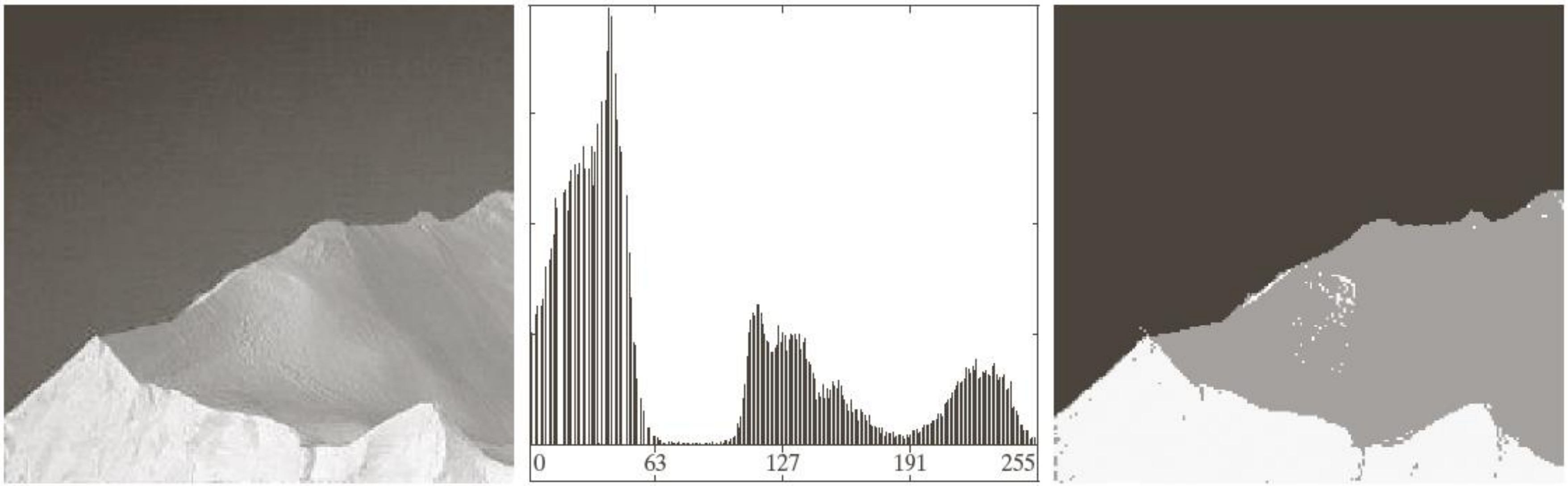
In the case of K classes, C_1, C_2, ..., C_K, the between-class variance is

$$\sigma_B^2 = \sum_{k=1}^{K} P_k \left(m_k - m_G\right)^2$$

where $$P_k = \sum_{i \in C_k} p_i \quad \text{and} \quad m_k = \frac{1}{P_k} \sum_{i \in C_k} i p_i$$

The optimum threshold values, k_1^*, k_2^*, ..., k_{K-1}^* that maximize

$$\sigma_B^2(k_1^*, k_2^*, \ldots, k_{K-1}^*) = \max_{0 \le k \le L-1} \sigma_B^2(k_1, k_2, \ldots, k_{K-1})$$



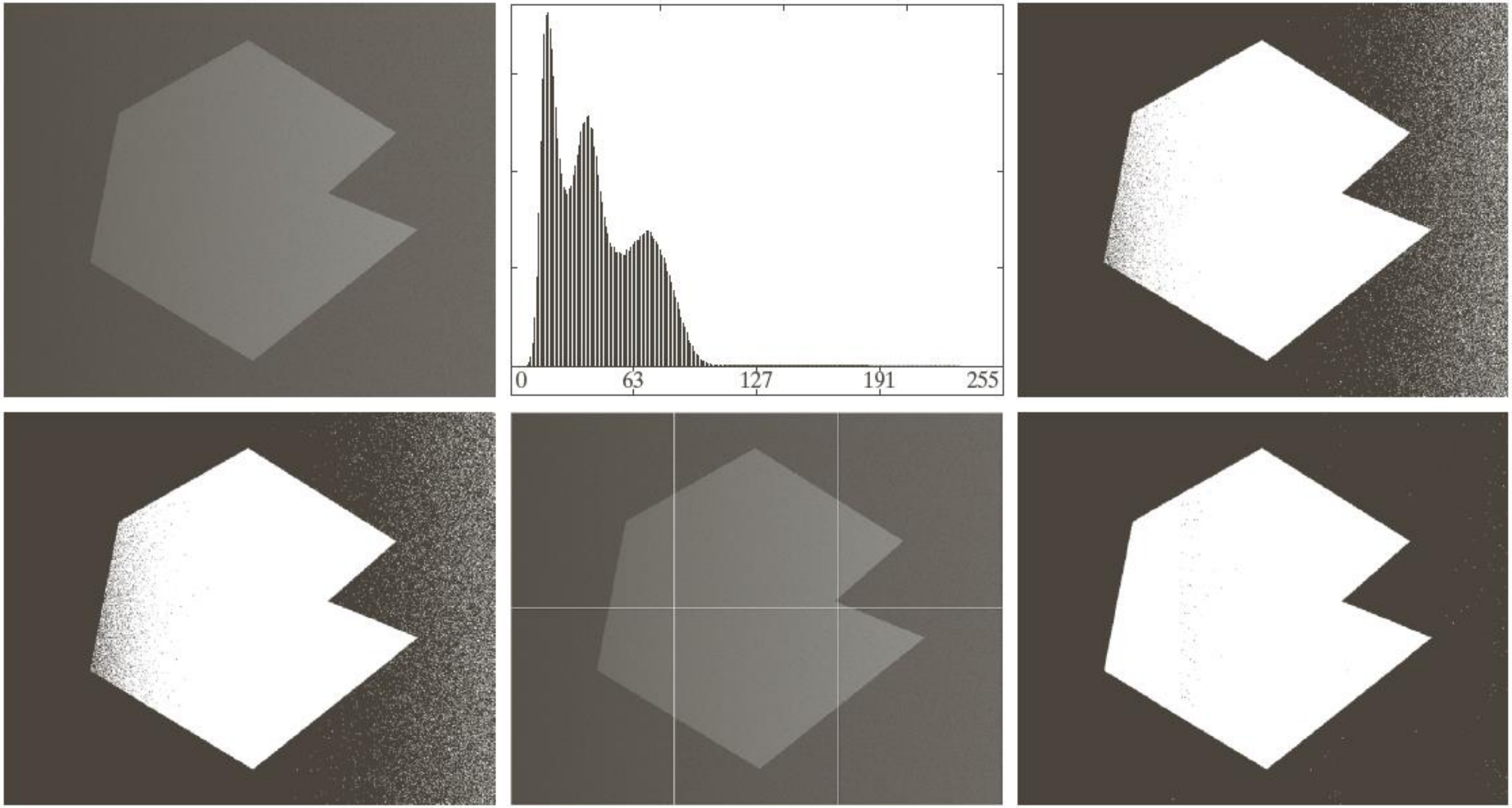
a b c

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

Variable Thresholding: Image Partitioning

- Subdivide an image into nonoverlapping rectangles

- The rectangles are chosen small enough so that the illumination of each is approximately uniform.



a b c
d e f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

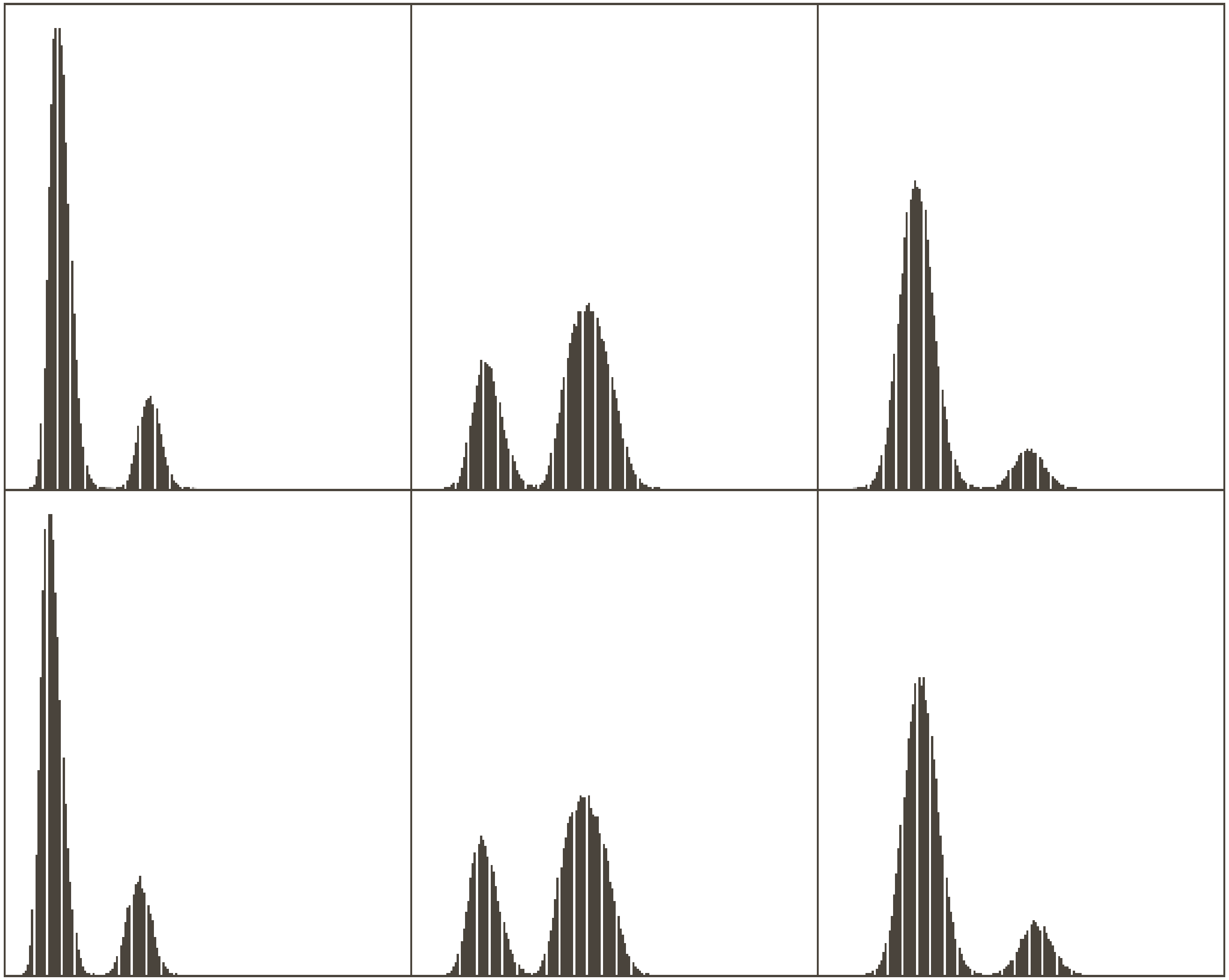


FIGURE 10.47
Histograms of the six subimages in Fig. 10.46(e).

Variable Thresholding Based on Local Image Properties

Let σ_{xy} and m_{xy} denote the standard deviation and mean value of the set of pixels contained in a neighborhood S_{xy}, centered at coordinates (x, y) in an image. The local thresholds,

$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

If the background is nearly constant,

$$T_{xy} = a\sigma_{xy} + bm$$

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

Variable Thresholding Based on Local Image Properties

A modified thresholding

$$g(x, y) = \begin{cases} 1 & \text{if } Q(\text{local parameters}) \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

$e.g.,$

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{true} & \text{if } f(x, y) > a\sigma_{xy} \text{ AND } f(x, y) > bm_{xy} \\ \text{false} & \text{otherwise} \end{cases}$$

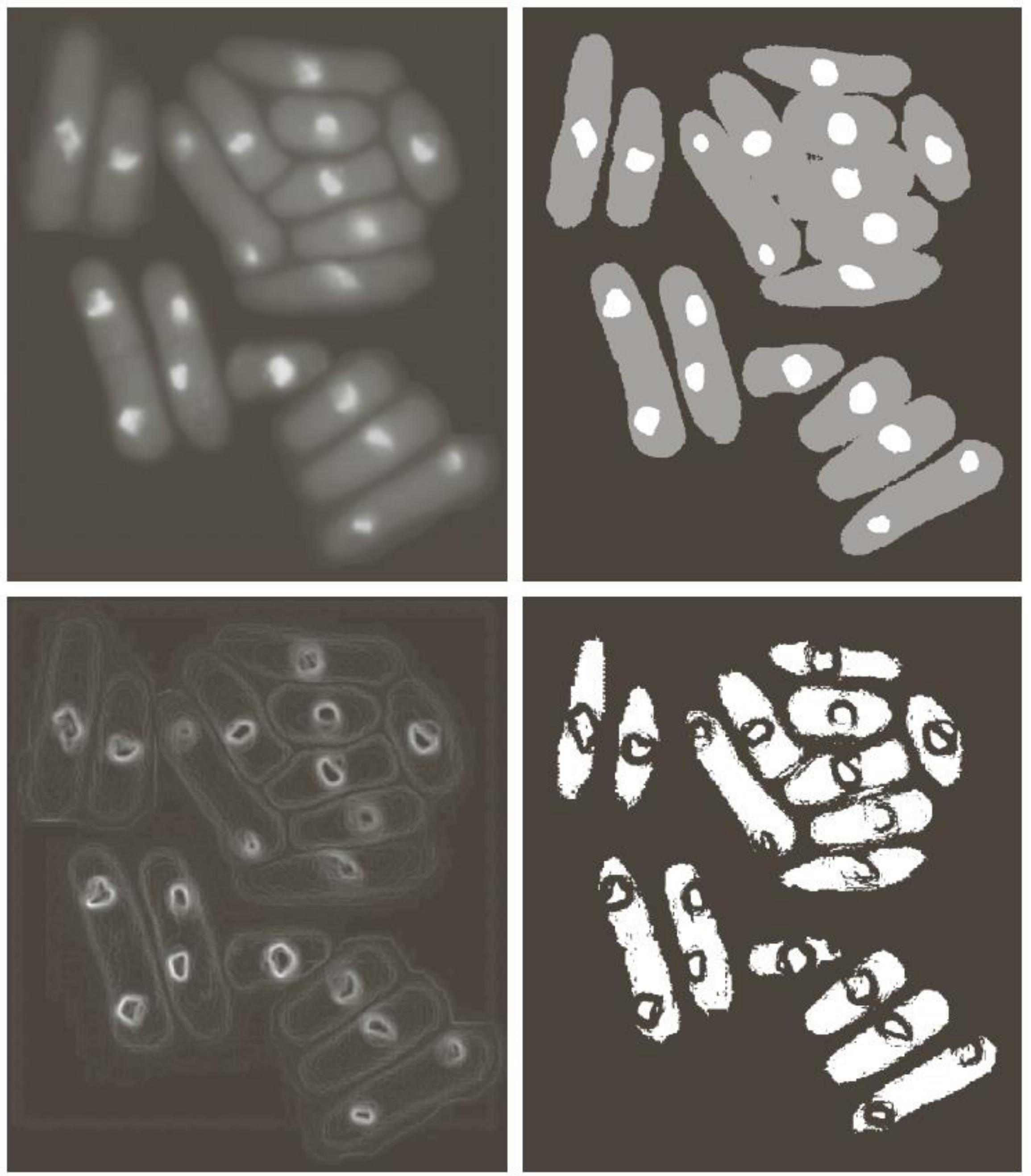


FIGURE 10.48 (a) Image from Fig. 10.43. (b) Image segmented using the dual thresholding approach discussed in Section 10.3.6. (c) Image of local standard deviations. (d) Result obtained using local thresholding.

a=30
b=1.5
$m_{xy} = m_G$

Variable Thresholding Using Moving Averages

- Thresholding based on moving averages works well when the objects are small with respect to the image size

- Quite useful in document processing

- The scanning (moving) typically is carried out line by line in zigzag pattern to reduce illumination bias

Variable Thresholding Using Moving Averages

Moving average at the pixel k is formed by averaging the intensities of that pixel and its n-1 preceding neighbors

Suppose we have a 5*5 image, n = 4

a_1	a_2	a_3	a_4	a_5
b_1	b_2	b_3	b_4	b_5
c_1	c_2	c_3	c_4	c_5
d_1	d_2	d_3	d_4	d_5
e_1	e_2	e_3	e_4	e_5

Step 1: reform it in a line following a zigzag way

a_1	a_2	a_3	a_4	a_5	b_5	b_4	b_3	b_2	b_1	c_1	c_2	c_3	c_4	c_5	...

Variable Thresholding Using Moving Averages

Step 2: for each position, compute the local average as the threshold

a_1	a_2	a_3	a_4	a_5	b_5	b_4	b_3	b_2	b_1	c_1	c_2	c_3	c_4	c_5	...

						m									...

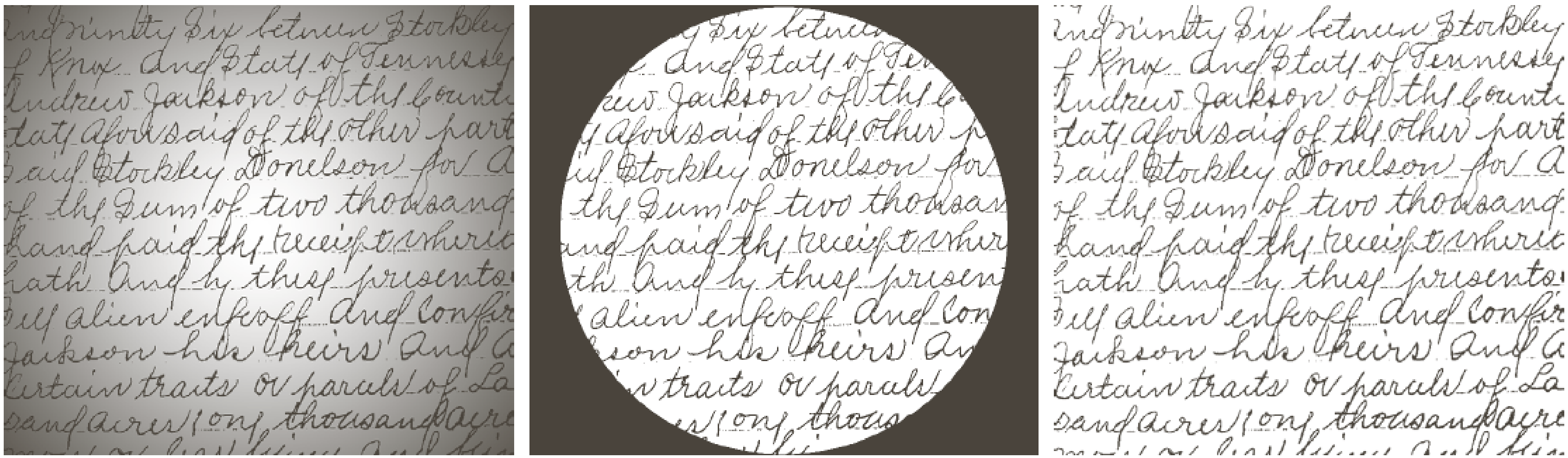
$$\left(a_4 + a_5 + b_5 + b_4\right) / 4$$

Variable Thresholding Using Moving Averages

Step 3: reform the local average "line" into the original matrix form to get the local threshold map $m(x, y)$

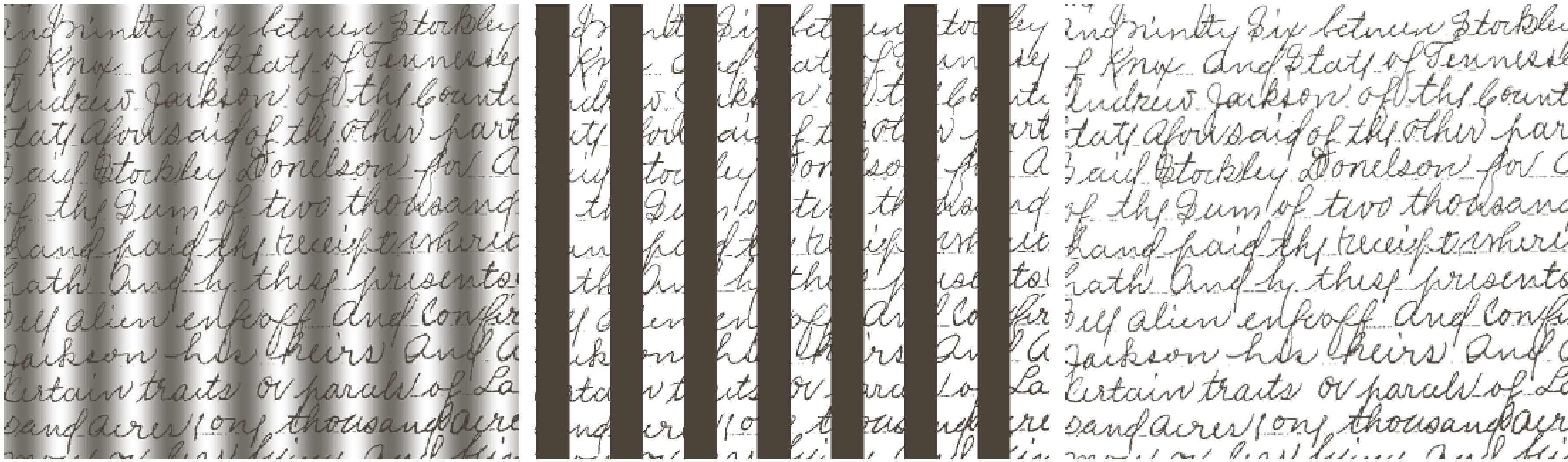
Segment the image as

$$g(x,y)=\begin{cases}1, if \;\; f(x,y)>K\cdot m(x,y)\\ 0, otherwise\end{cases}$$



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.



a b c

FIGURE 10.50 (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

Region-Based Segmentation

- Region Growing

1. Region growing is a procedure that groups pixels or subregions into larger regions.

2. The simplest of these approaches is ***pixel aggregation***, which starts with a set of "seed" points and from these grow regions by appending to each seed points those **neighboring pixels** that have **similar properties** (such as gray level, texture, color, shape).

3. Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.

Region-Based Segmentation

Example: Region Growing based on 8-connectivity

$f(x, y)$: input image array

$S(x, y)$: seed array containing 1s (seeds) and 0s

$Q(x, y)$: predicate

Region Growing based on 8-connectivity

1. Find all connected components in $S(x, y)$ and erode each connected components to one pixel; label all such pixels found as 1. All other pixels in S are labeled 0.
2. Form an image f_Q such that, at a pair of coordinates (x,y), let $f_Q(x, y) = 1$ if the Q is satisfied otherwise $f_Q(x, y) = 0$.
3. Let g be an image formed by appending to each seed point in S all the 1-value points in f_Q that are 8-connected to that seed point.
4. Label each connencted component in g with a different region label. This is the segmented image obtained by region growing.

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

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

4-connectivity

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	<u>60</u>	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

8-connectivity

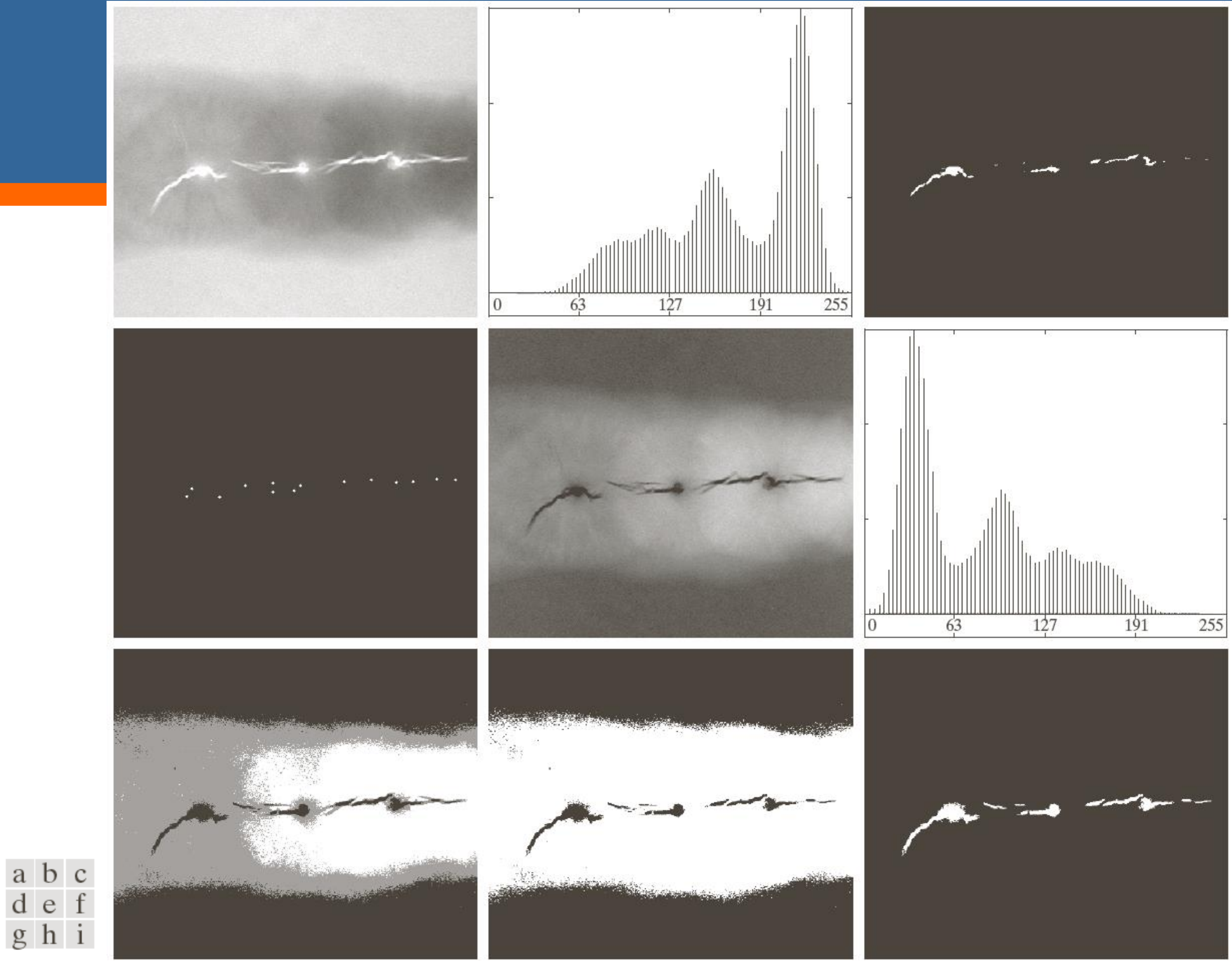
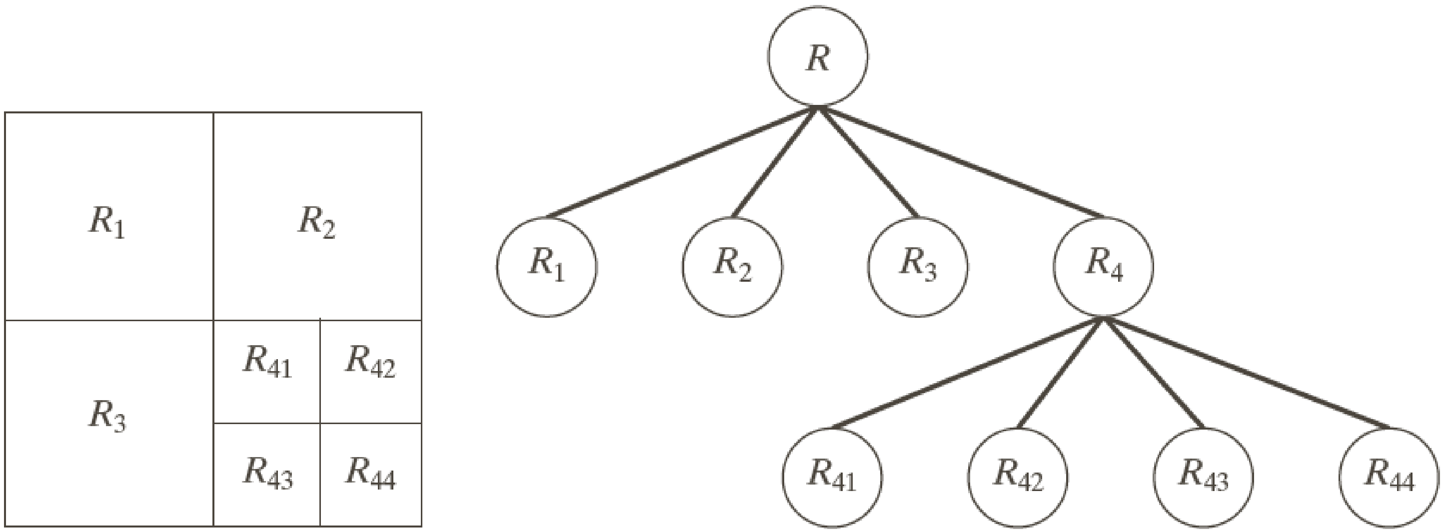


FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference 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

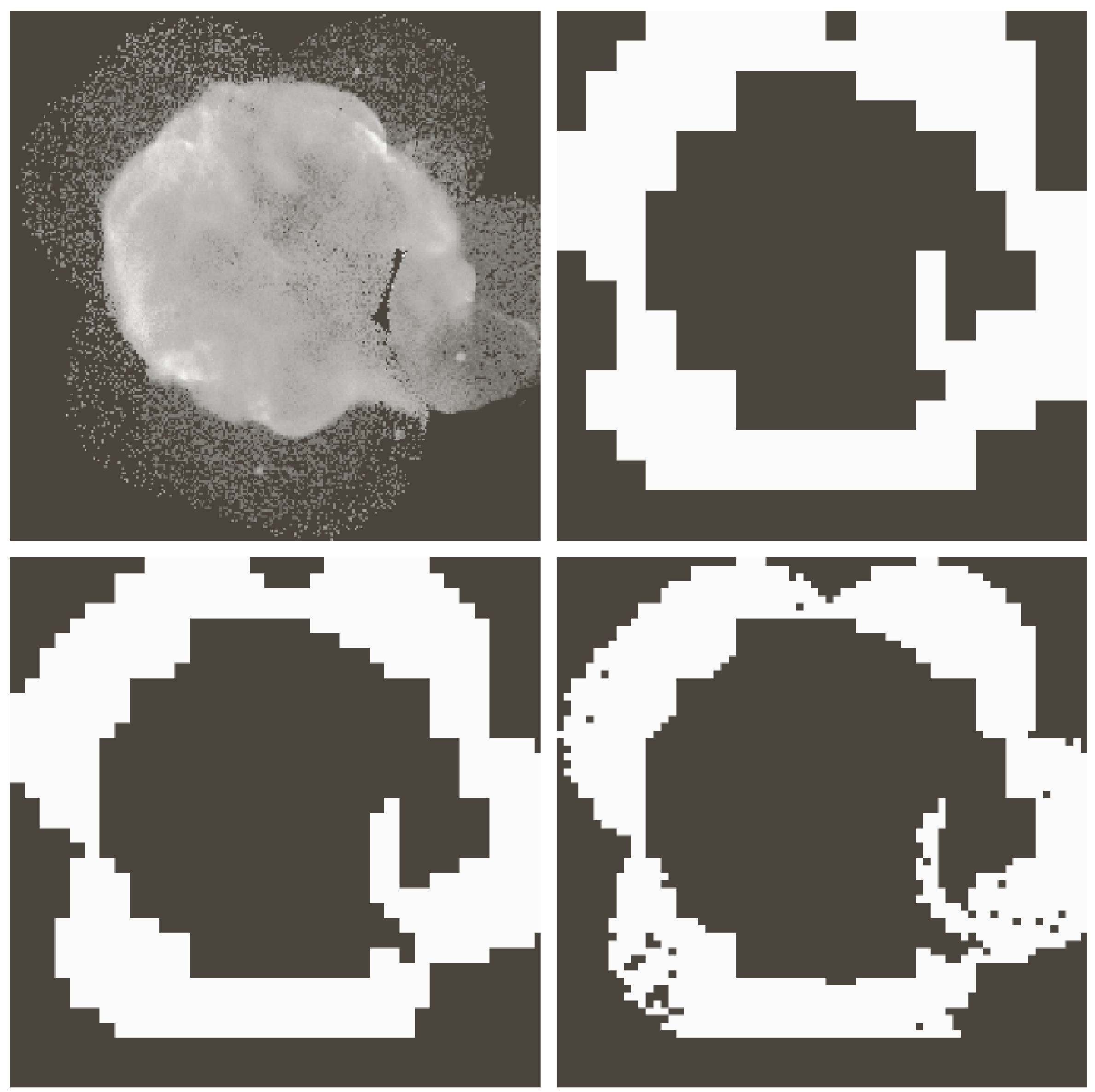
$R:$ entire image R_i: entire image Q: predicate

1. For any region R_i, If $Q(R_i)$ = FALSE,
   we divide the image R_i into quadrants.
2. When no further splitting is possible,
   merge any adjacent regions R_j and R_k
   for which $Q(R_j \cup R_k)$ = TRUE.
3. Stop when no further merging is possible.



a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.



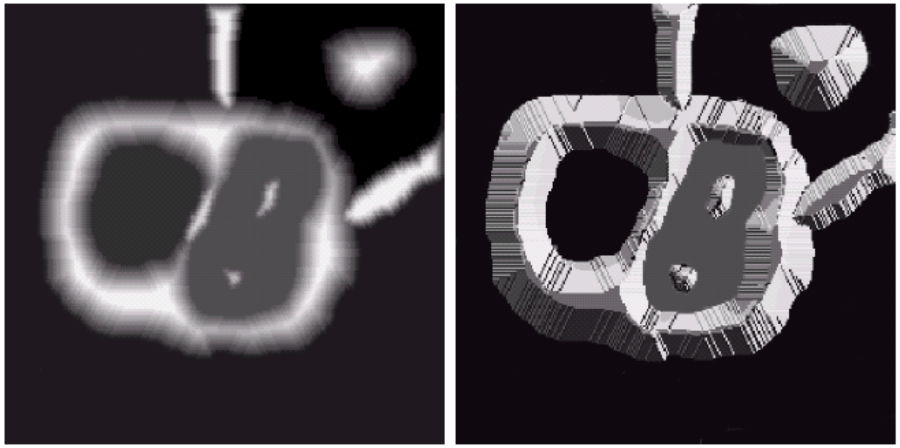
a	b
c	d

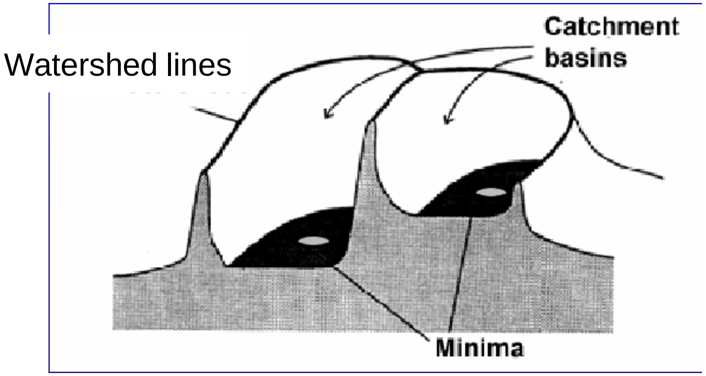
FIGURE 10.53
(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

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

Segmentation Using Morphological Watersheds

- Three types of points in a topographic interpretation:
  - Points belonging to a regional minimum
  - Points at which a drop of water would fall to a single minimum. (→The *catchment basin* or *watershed* of that minimum.)
  - Points at which a drop of water would be equally likely to fall to more than one minimum. (→The *divide lines* or *watershed lines*.)





Segmentation Using Morphological Watersheds: Backgrounds

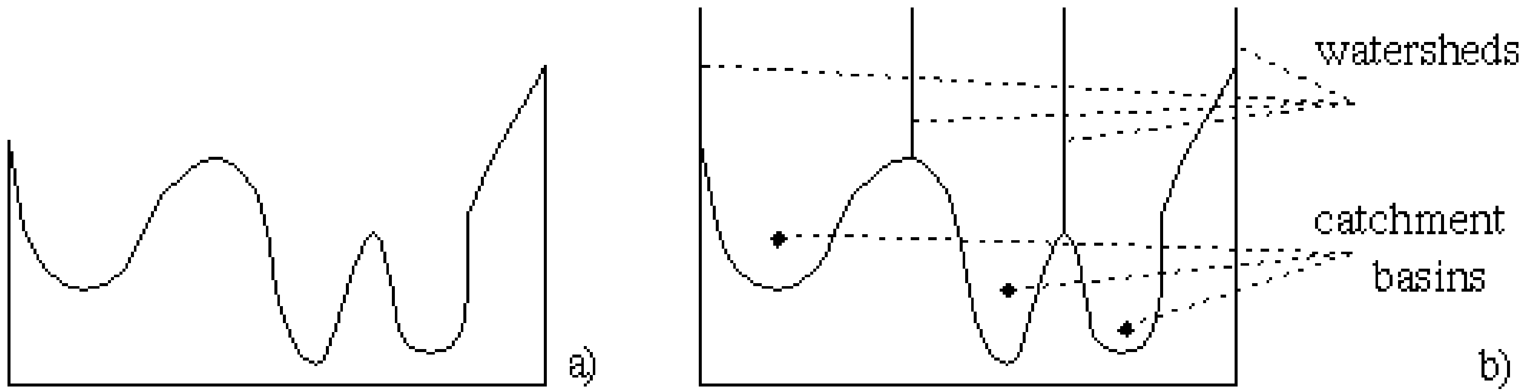
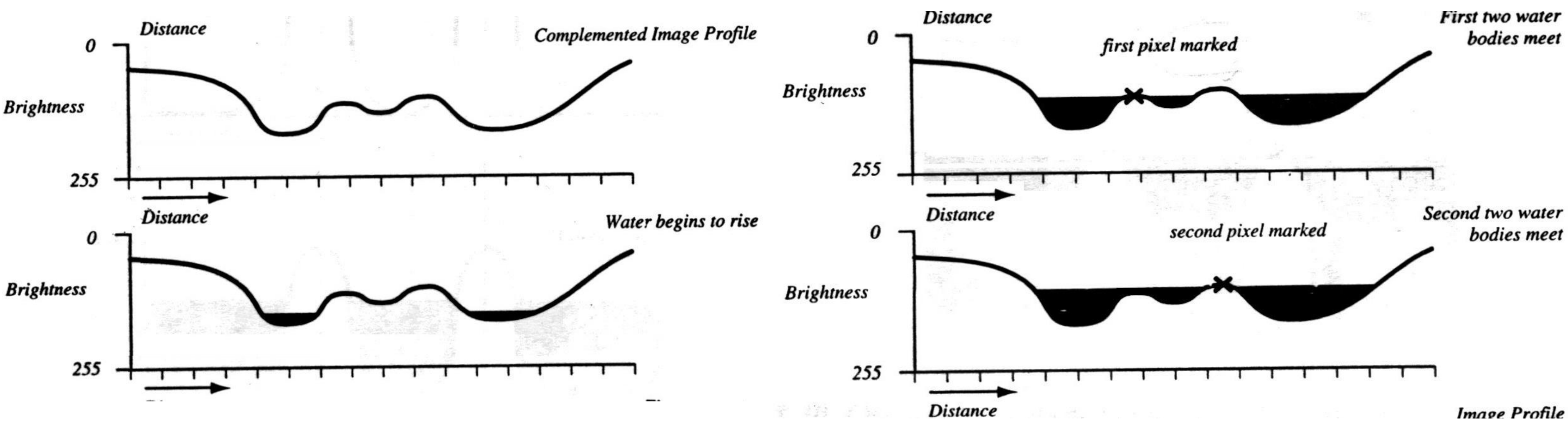
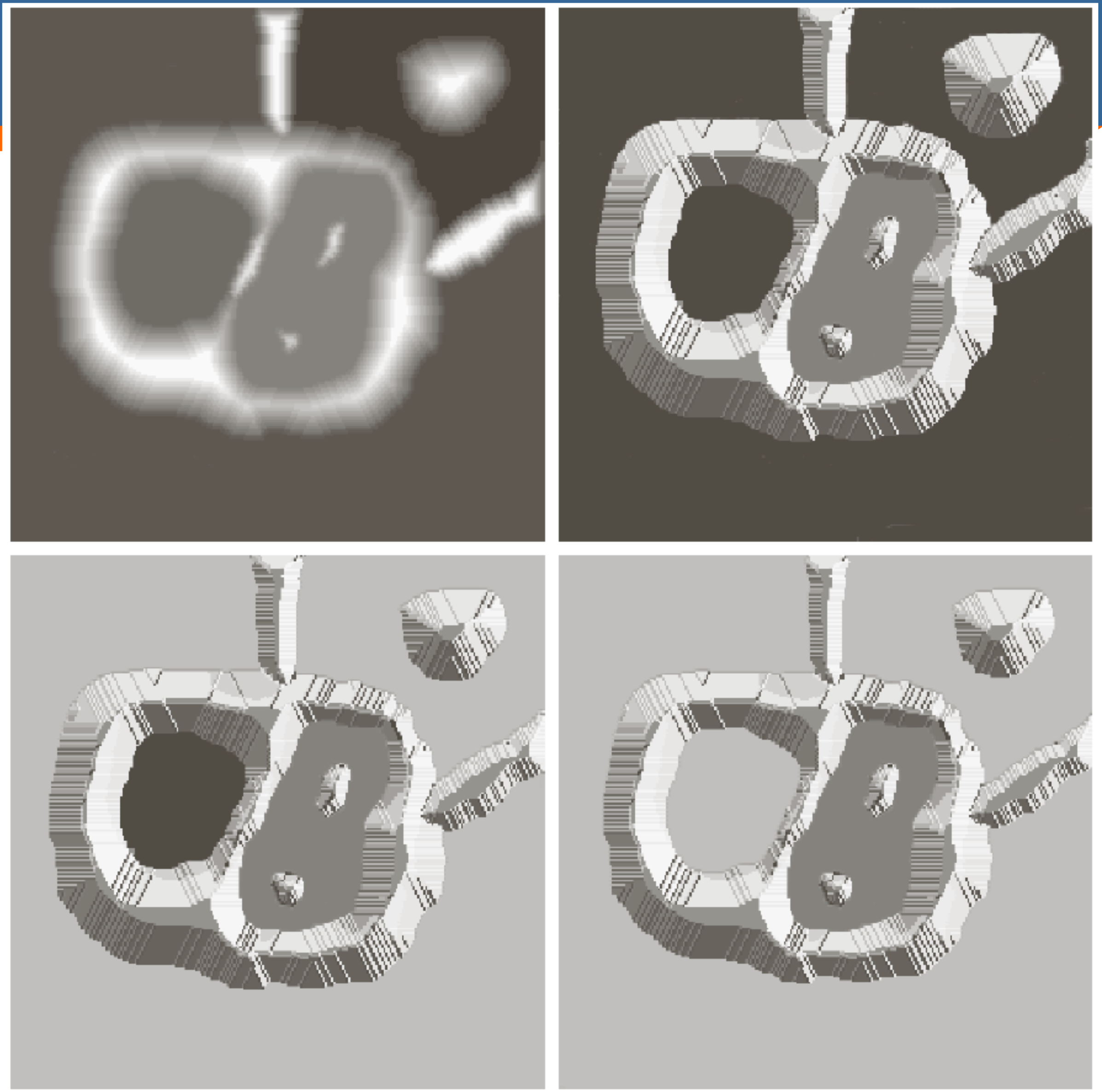


Figure 5.47 *One-dimensional example of watershed segmentation. (a) Gray level profile of image data. (b) Watershed segmentation – local minima of gray level (altitude) yield catchment basins, local maxima define the watershed lines.*

Watershed Segmentation: Example

- The objective is to find watershed lines.
- The idea is simple:
  - Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate.
  - When rising water in distinct catchment basins is about to merge, a dam is built to prevent merging. These dam boundaries correspond to the watershed lines.





a	b
c	d

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

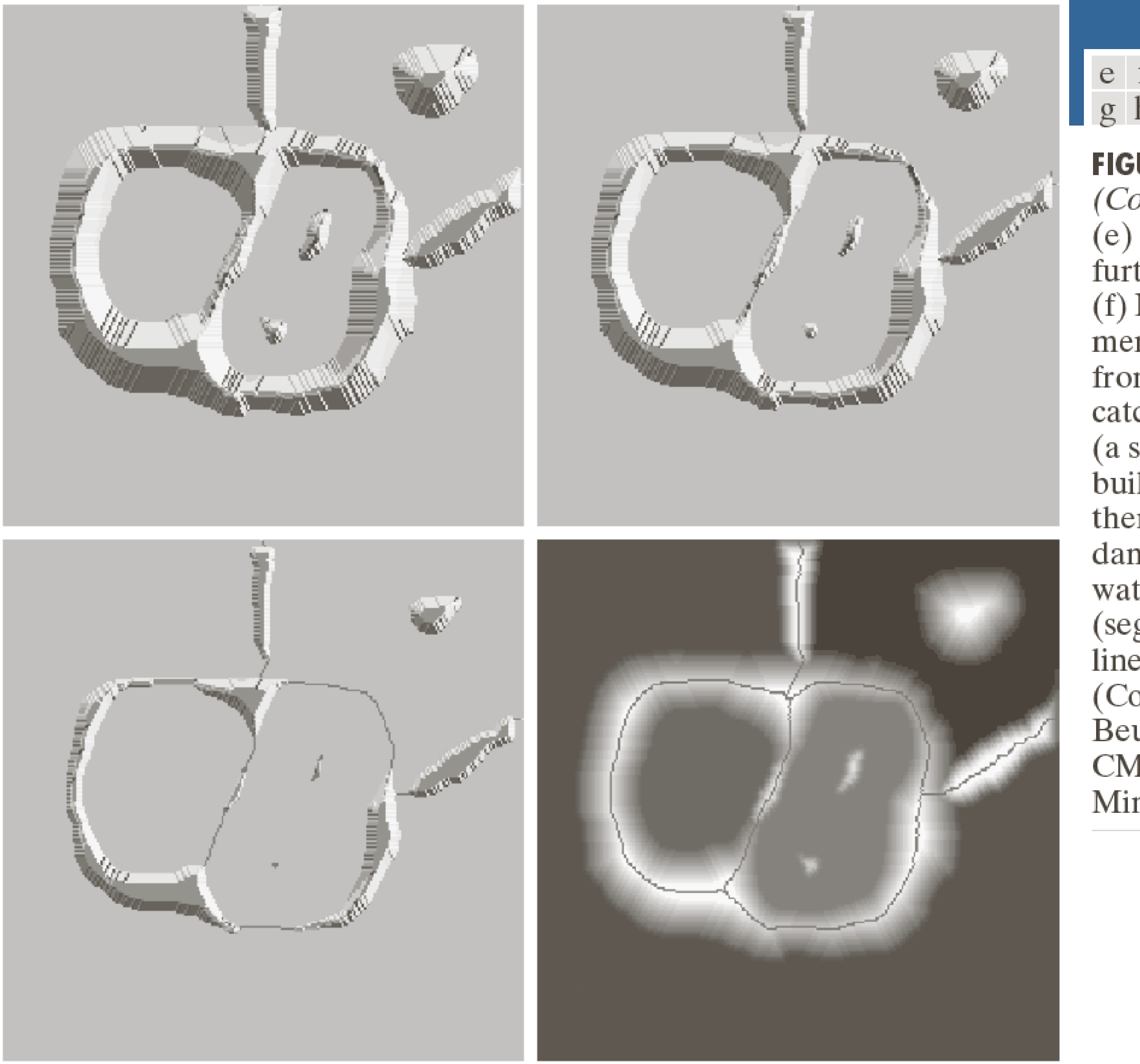


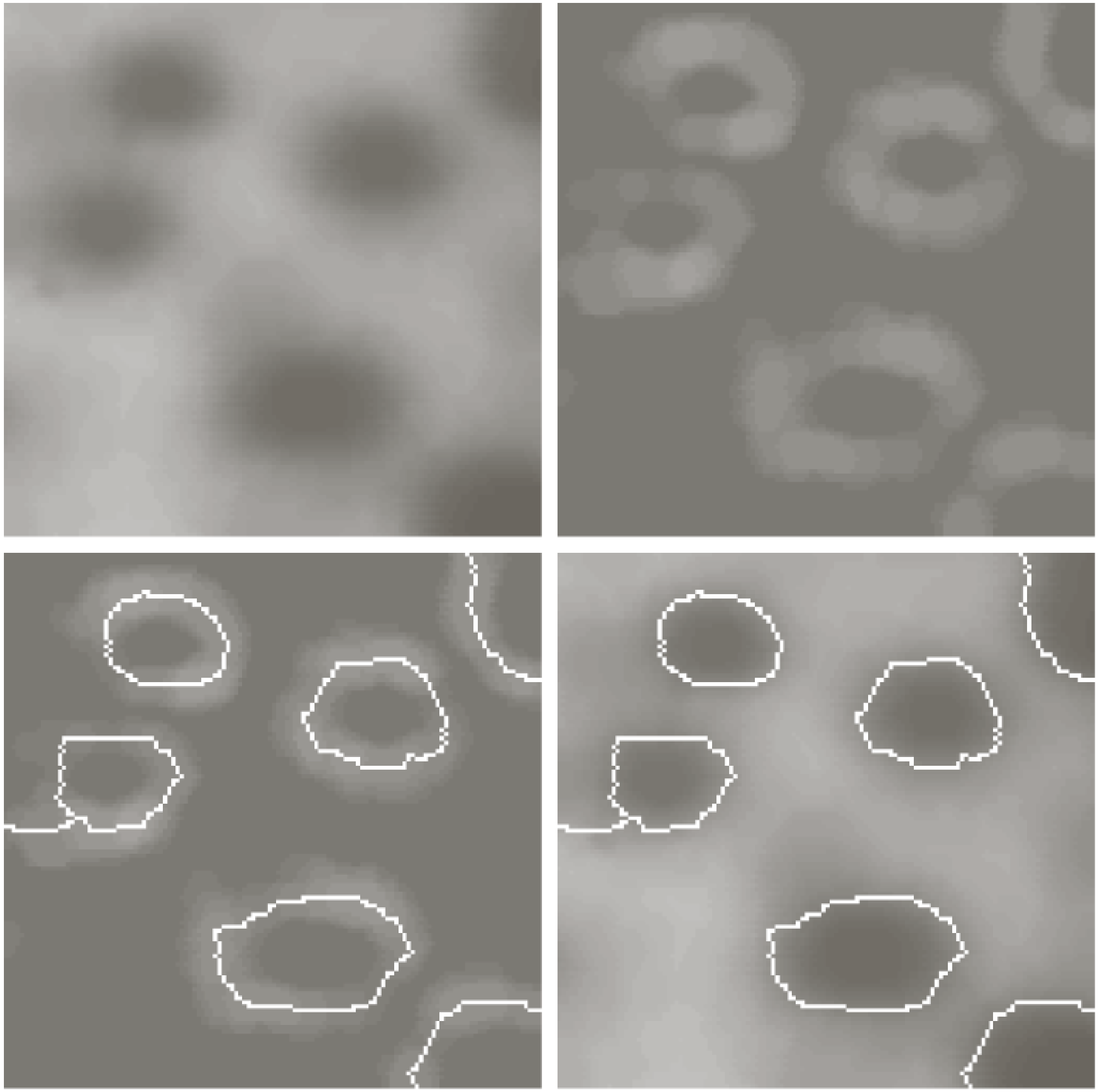
FIGURE 10.54 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed Segmentation Algorithm

- Start with all pixels with the lowest possible value.
  - These form the basis for initial watersheds
- For each intensity level k:
  - For each group of pixels of intensity k
    1. If adjacent to exactly one existing region, add these pixels to that region
    2. Else if adjacent to more than one existing regions, mark as boundary
    3. Else start a new region

Watershed Segmentation: Examples

Watershed algorithm is often used on the gradient image instead of the original image.



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.)

Watershed Segmentation: Examples

a b

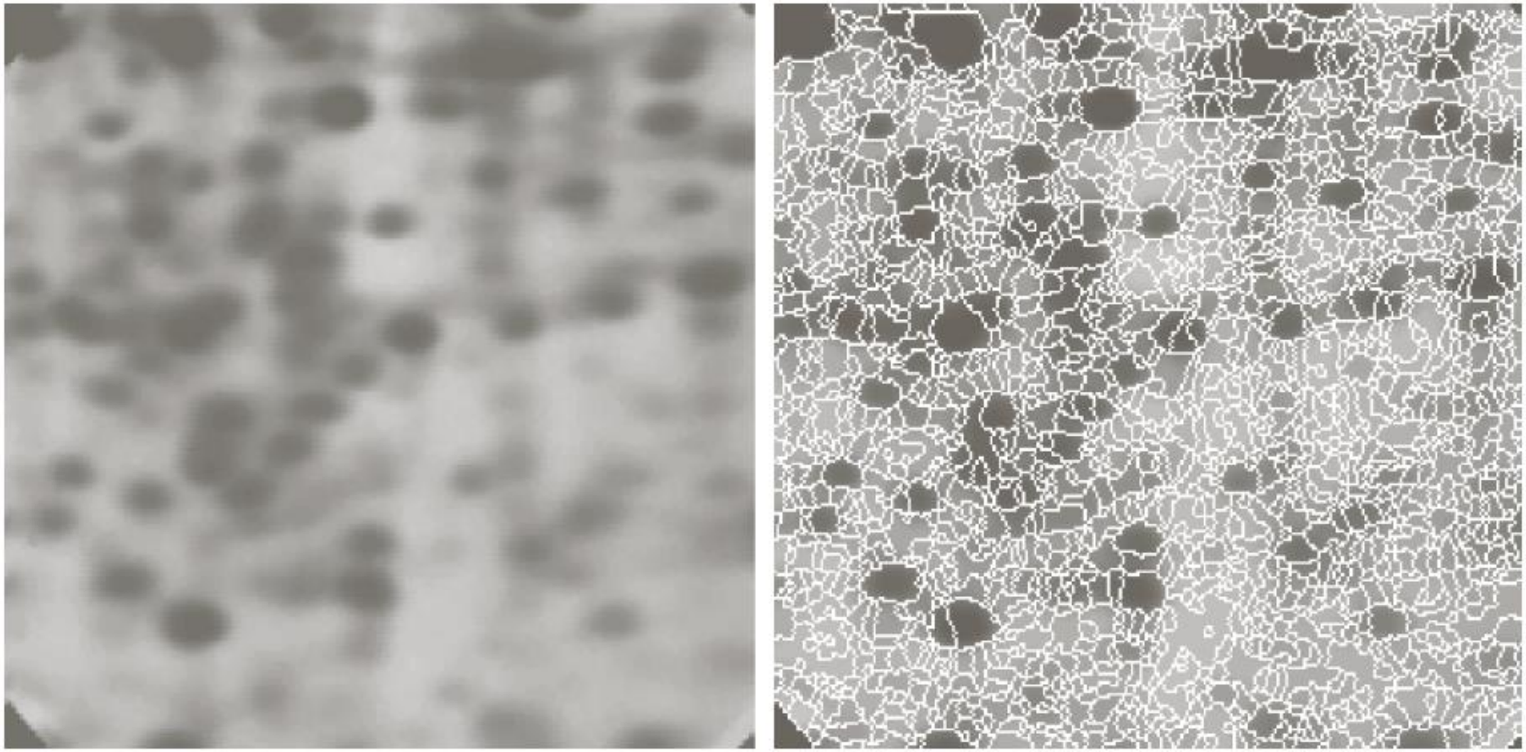


FIGURE 10.57 (a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

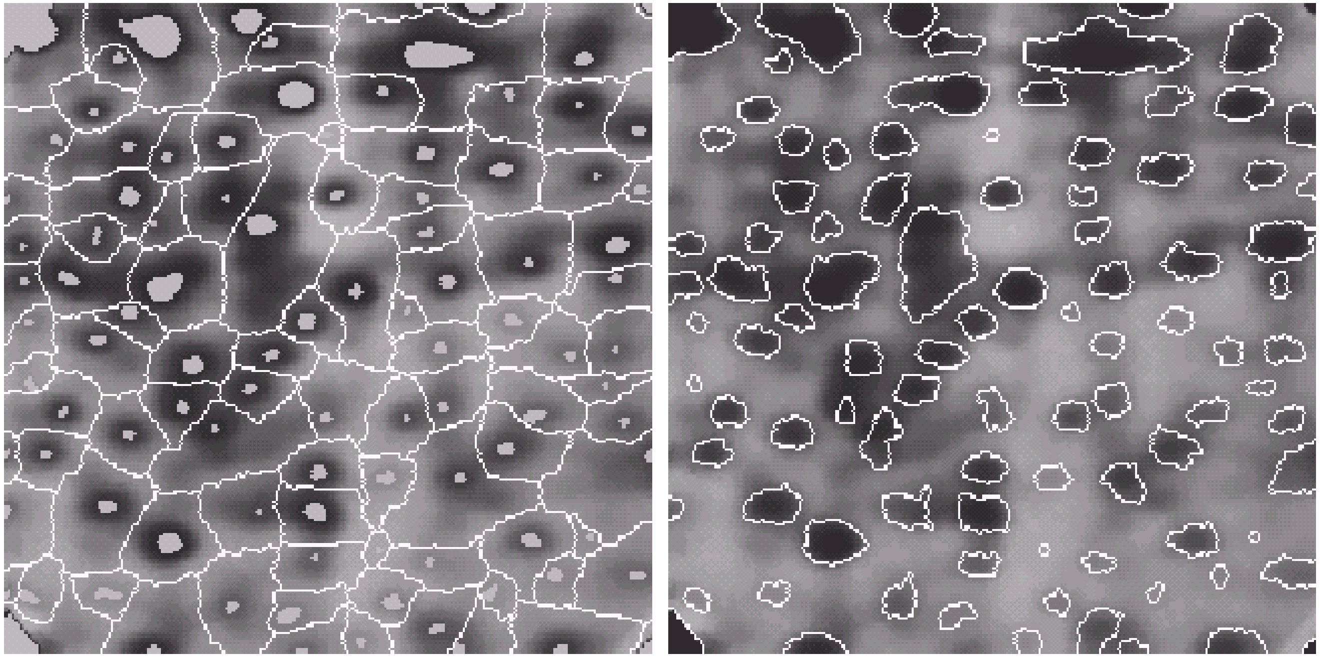
Due to noise and other local irregularities of the gradient, over-segmentation might occur.

Watershed Segmentation: Examples

A solution is to limit the number of regional minima. Use markers to specify the only allowed regional minima.

Watershed Segmentation: Examples

A solution is to limit the number of regional minima. Use markers to specify the only allowed regional minima. (For example, gray-level values might be used as a marker.)



a b

FIGURE 10.48 (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.)

K-means Clustering

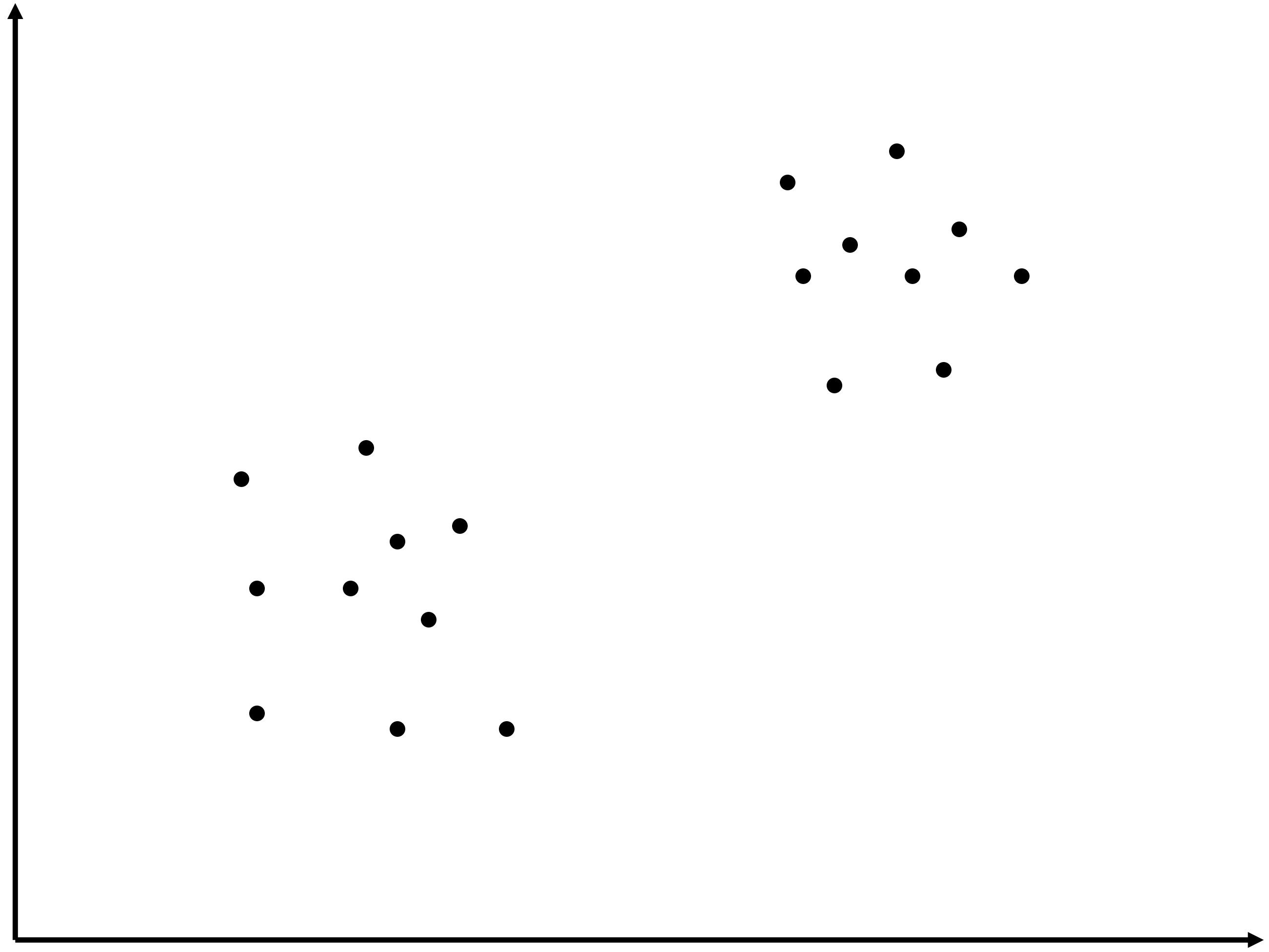
- Partition the data points into K clusters randomly. Find the centroids of each cluster.

- For each data point:
  - Calculate the distance from the data point to each cluster.
  - Assign the data point to the closest cluster.

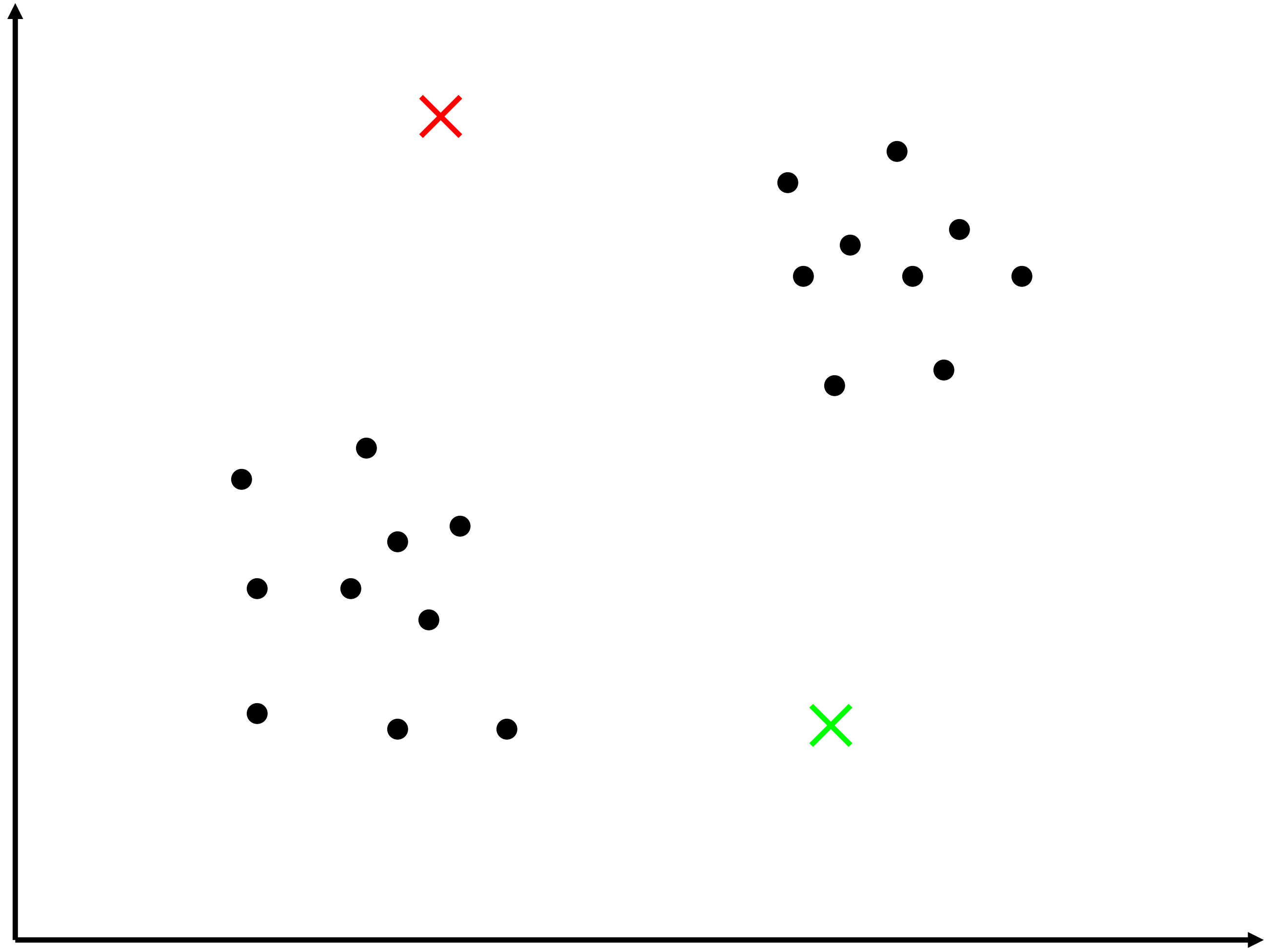
- Recompute the centroid of each cluster.

- Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

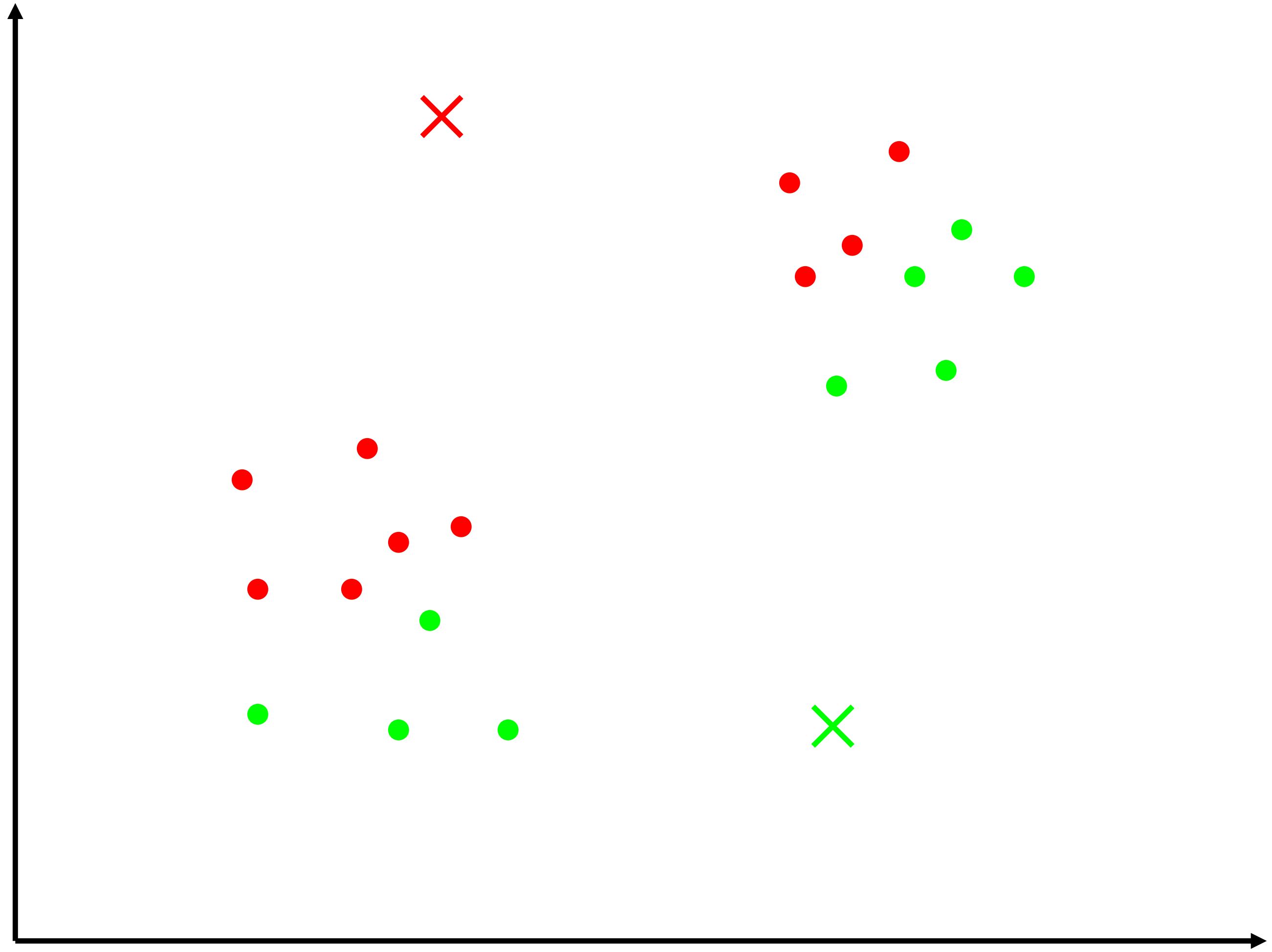
K-Means Clustering



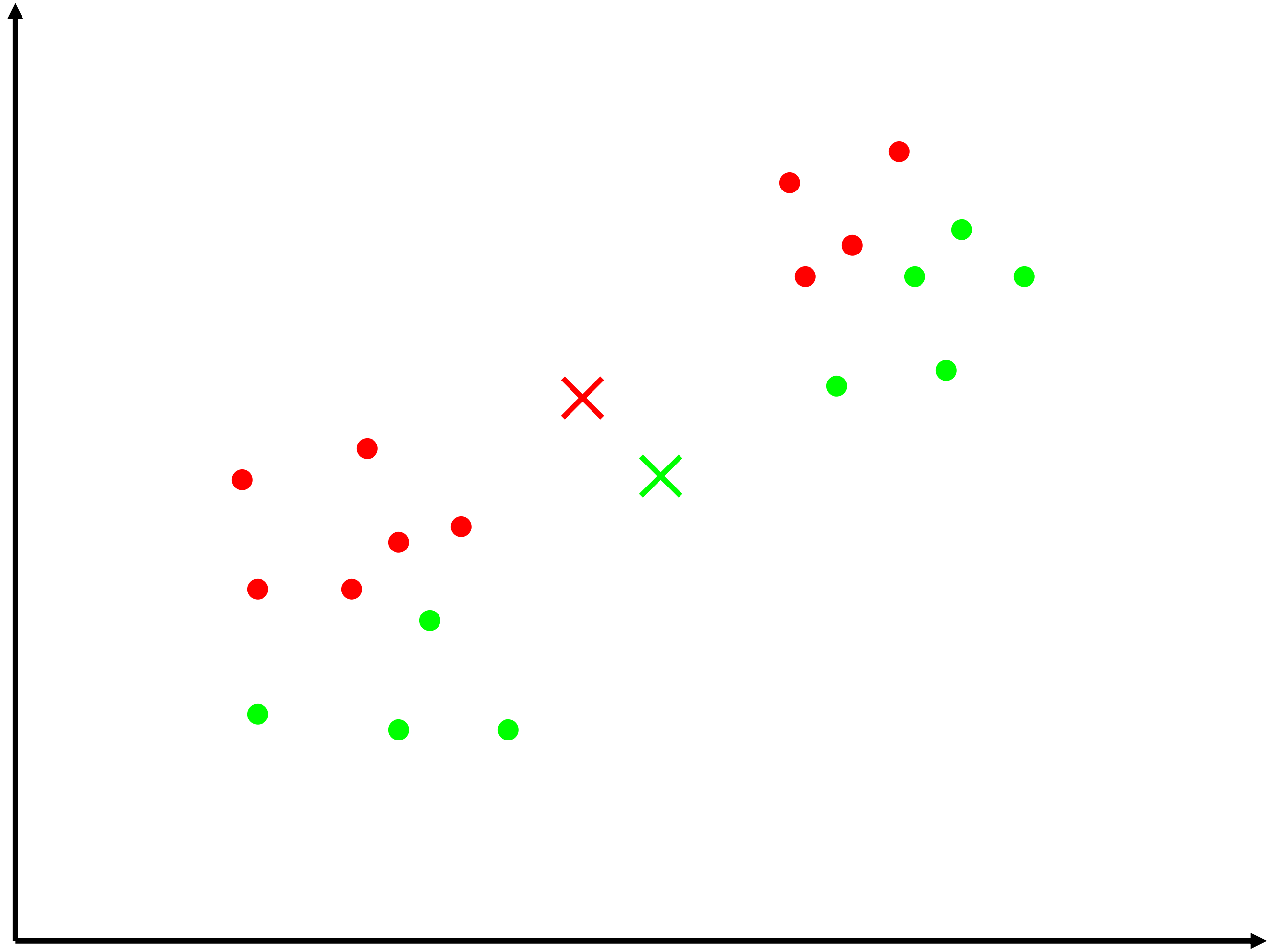
K-Means Clustering



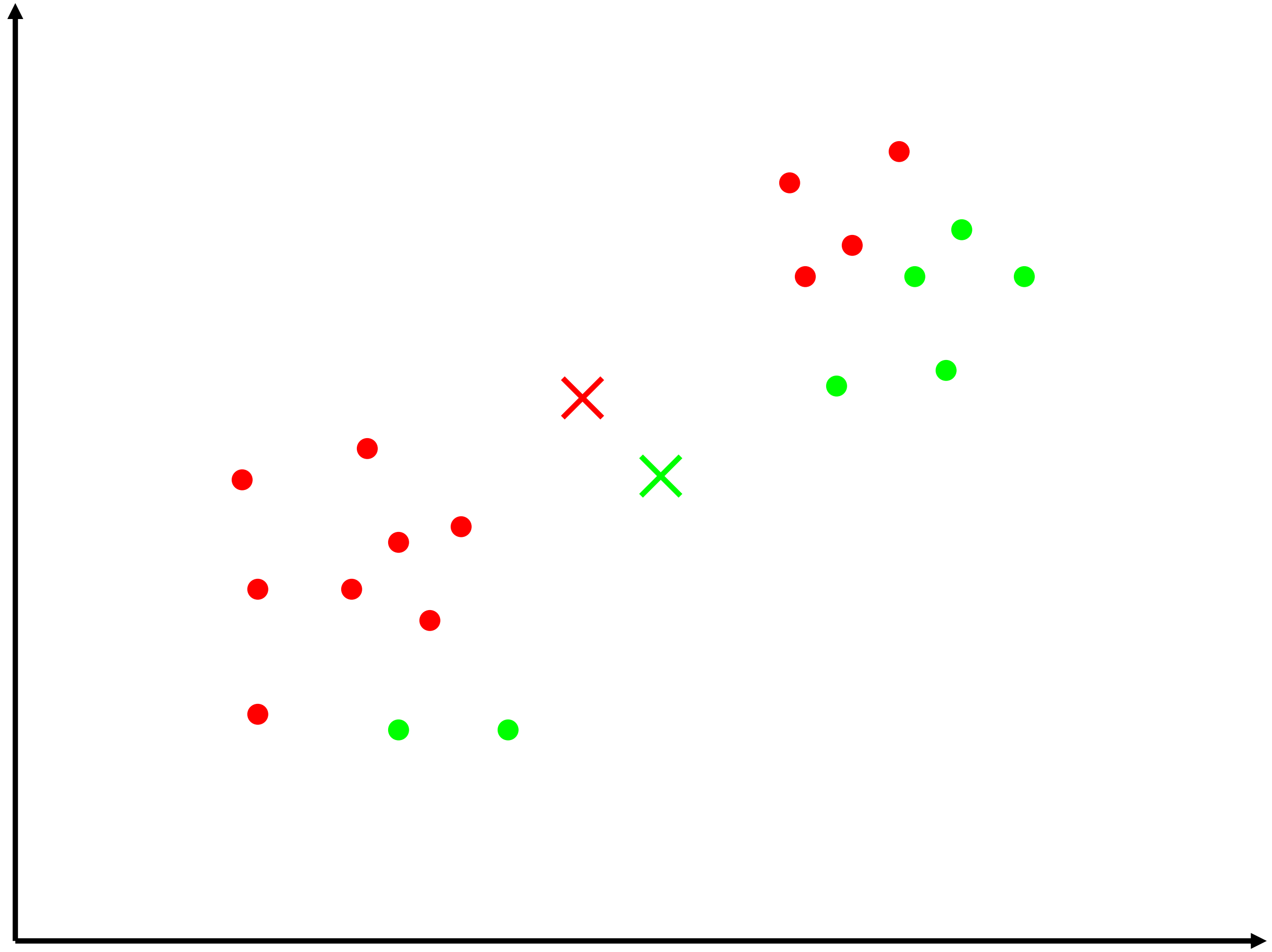
K-Means Clustering



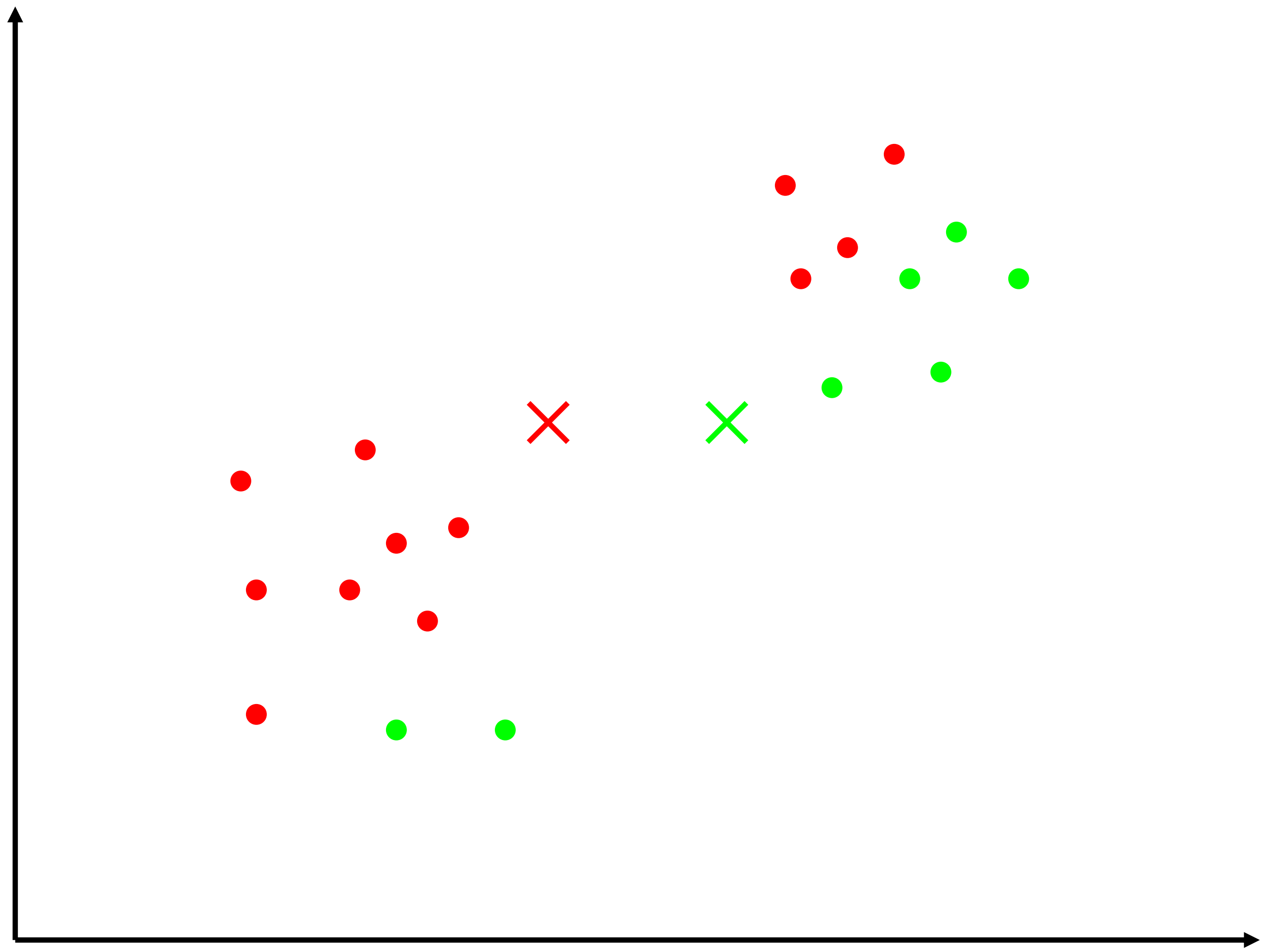
K-Means Clustering



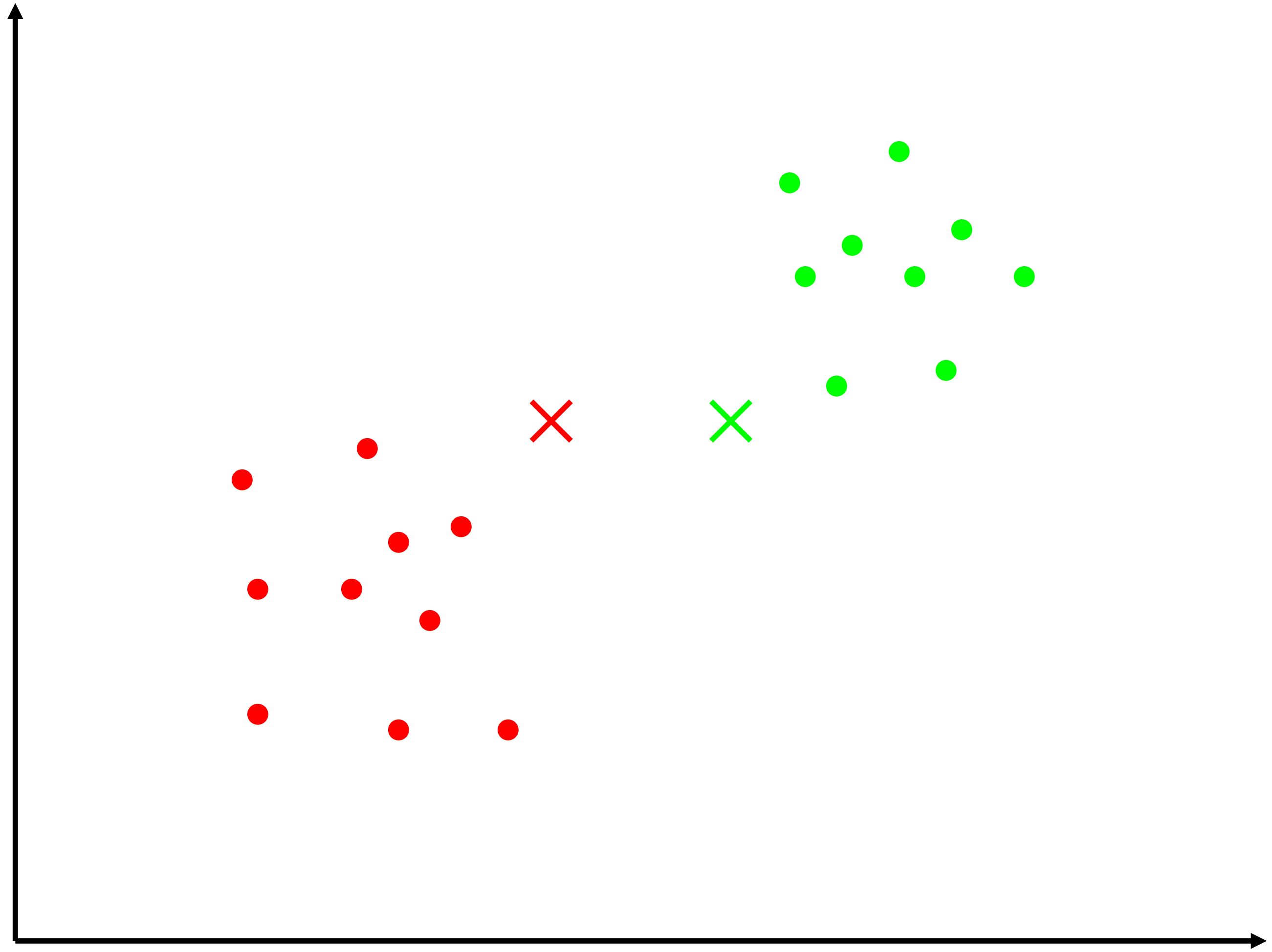
K-Means Clustering



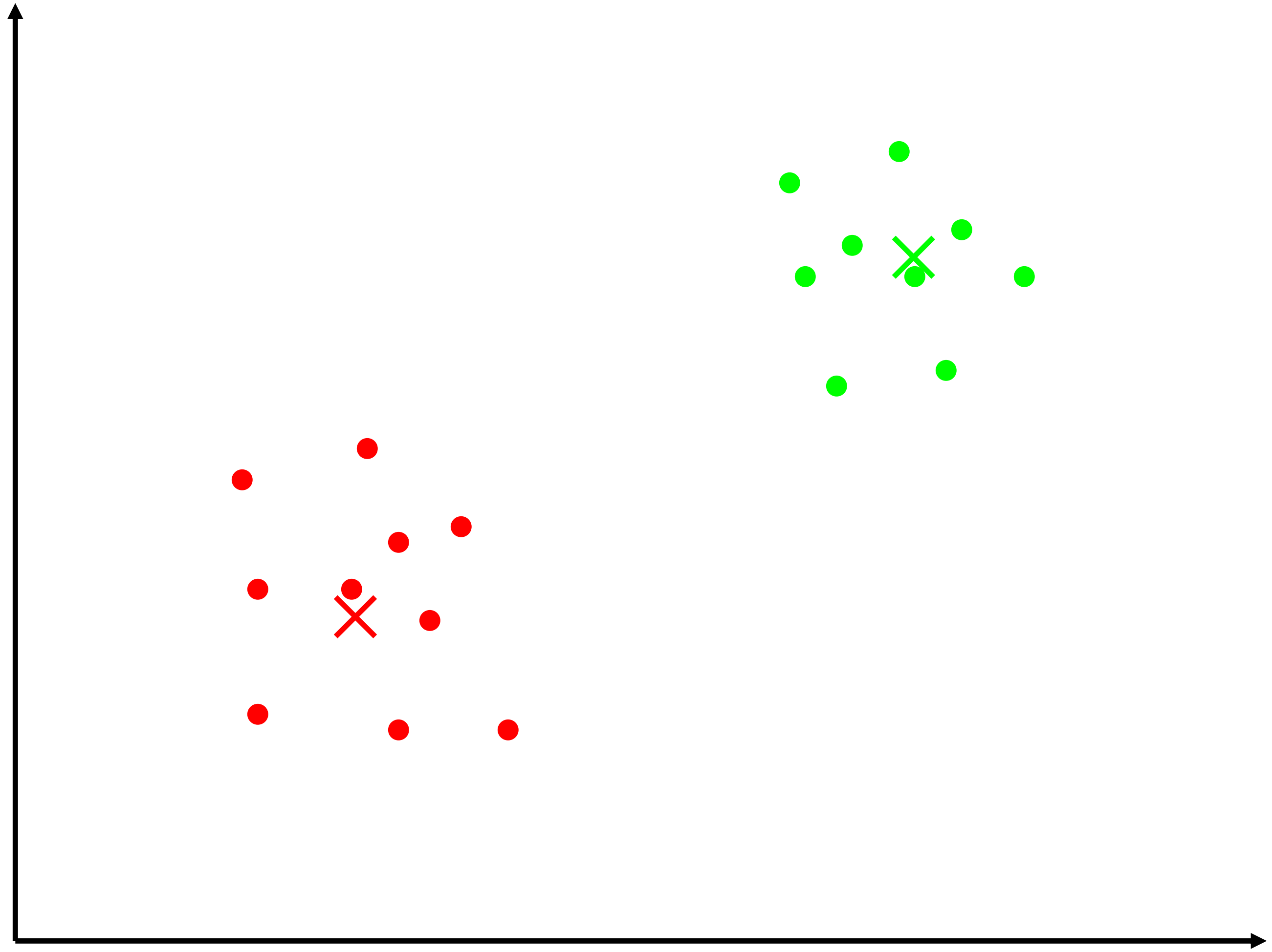
K-Means Clustering



K-Means Clustering

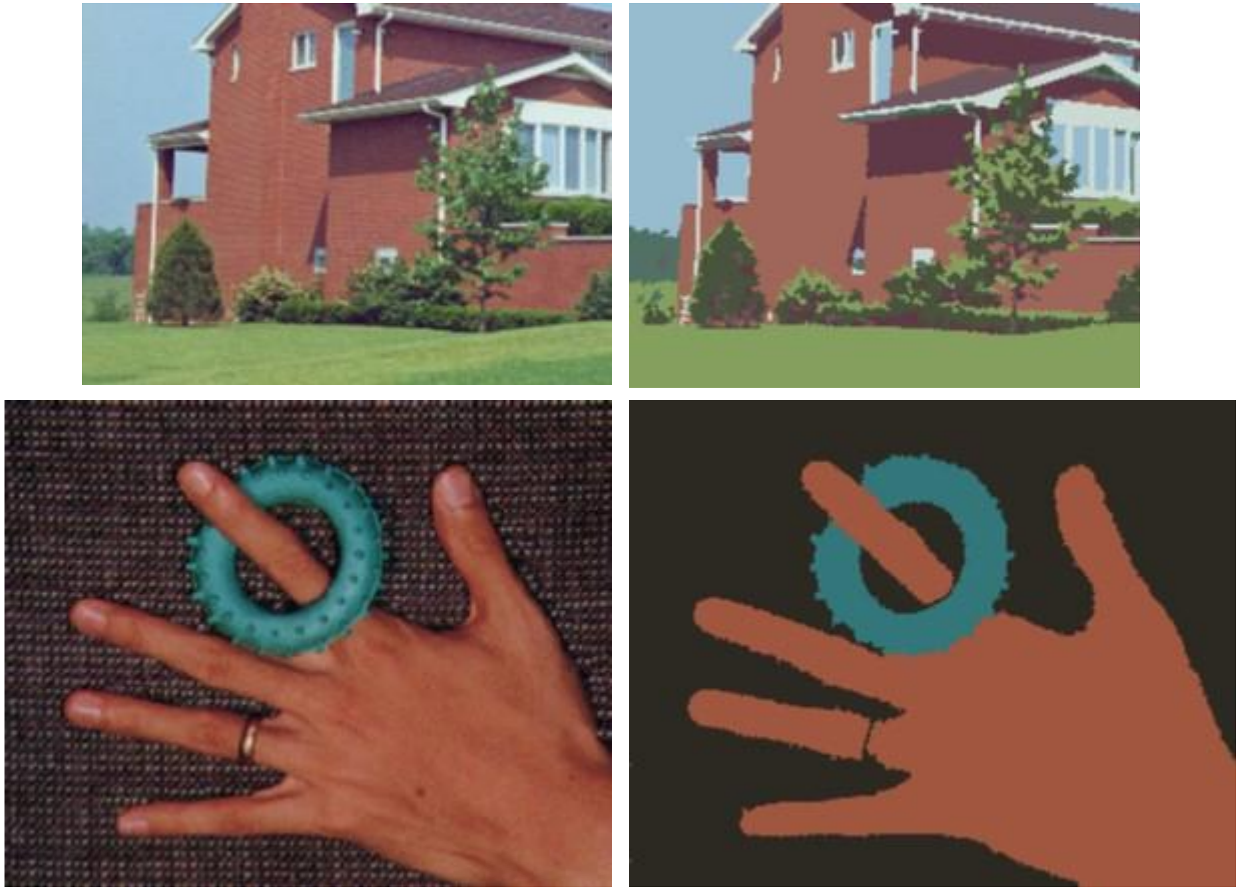


K-Means Clustering



Clustering

- Example



D. Comaniciu and P. Meer, *Robust Analysis of Feature Spaces: Color Image Segmentation*, 1997.

Thank You