Lecture # 09

Biomedical Image Segmentation

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

1. $\bigcup_{i=1}^{n} R_i = R$
   - Union must be *complete*; every pixel must be in a region
2. R_i is a connected set, $i = 1, 2, \ldots, n$
   - Points in a region must be connected (8-connected)
3. $R_i \cap R_j = \emptyset \; \forall i, j | i \neq j$
   - Regions must be disjoint
4. $Q(R_i) = \texttt{true}$ for $i = 1, 2, \ldots, n$
   - $Q(R_k)$ is a logical predicate defined over the points in set R_k
5. $Q(R_i \cup R_j) = \texttt{false}$ for any adjacent region R_i and R_j
   - Adjacent regions must be different

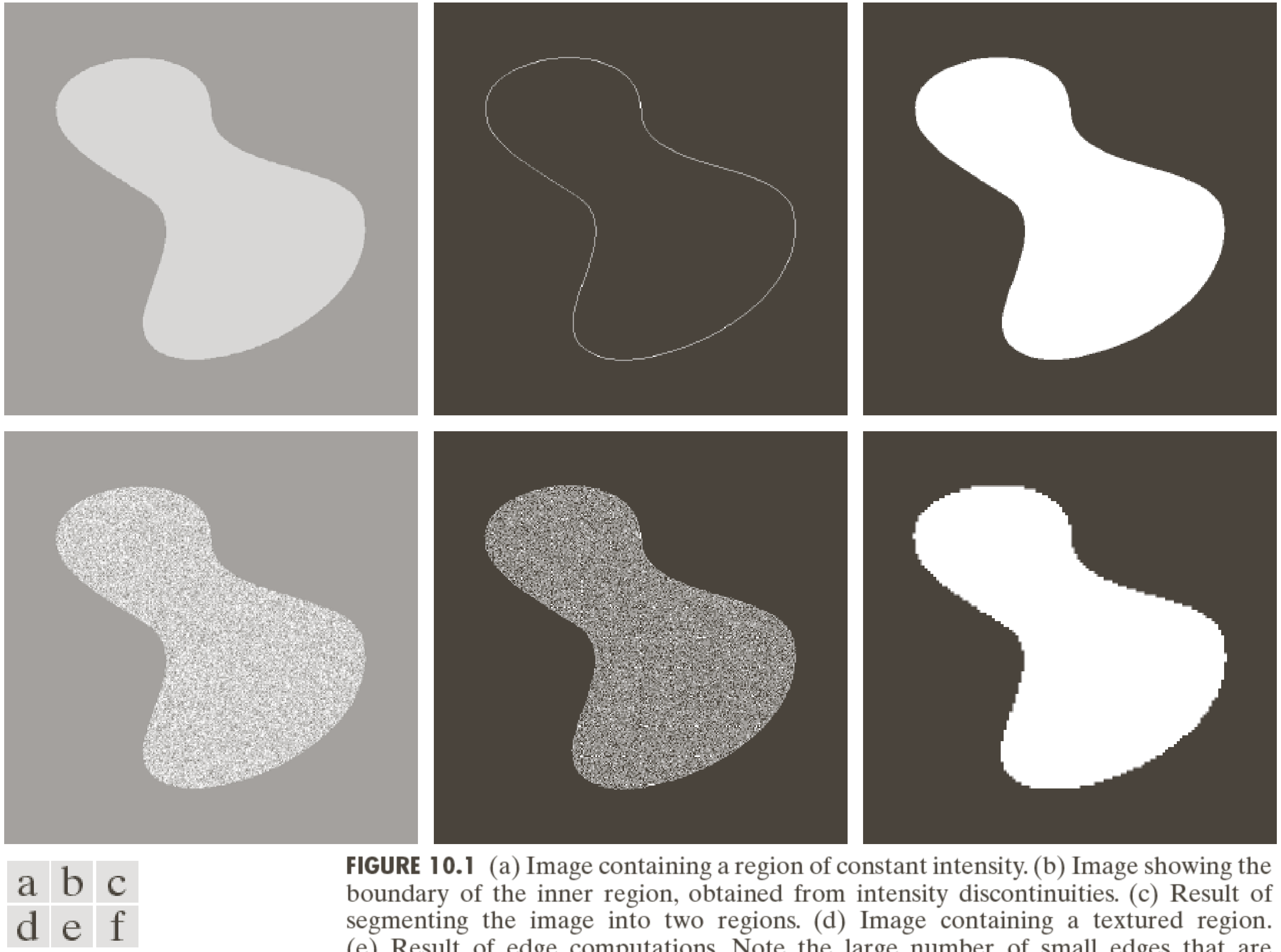


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number 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.

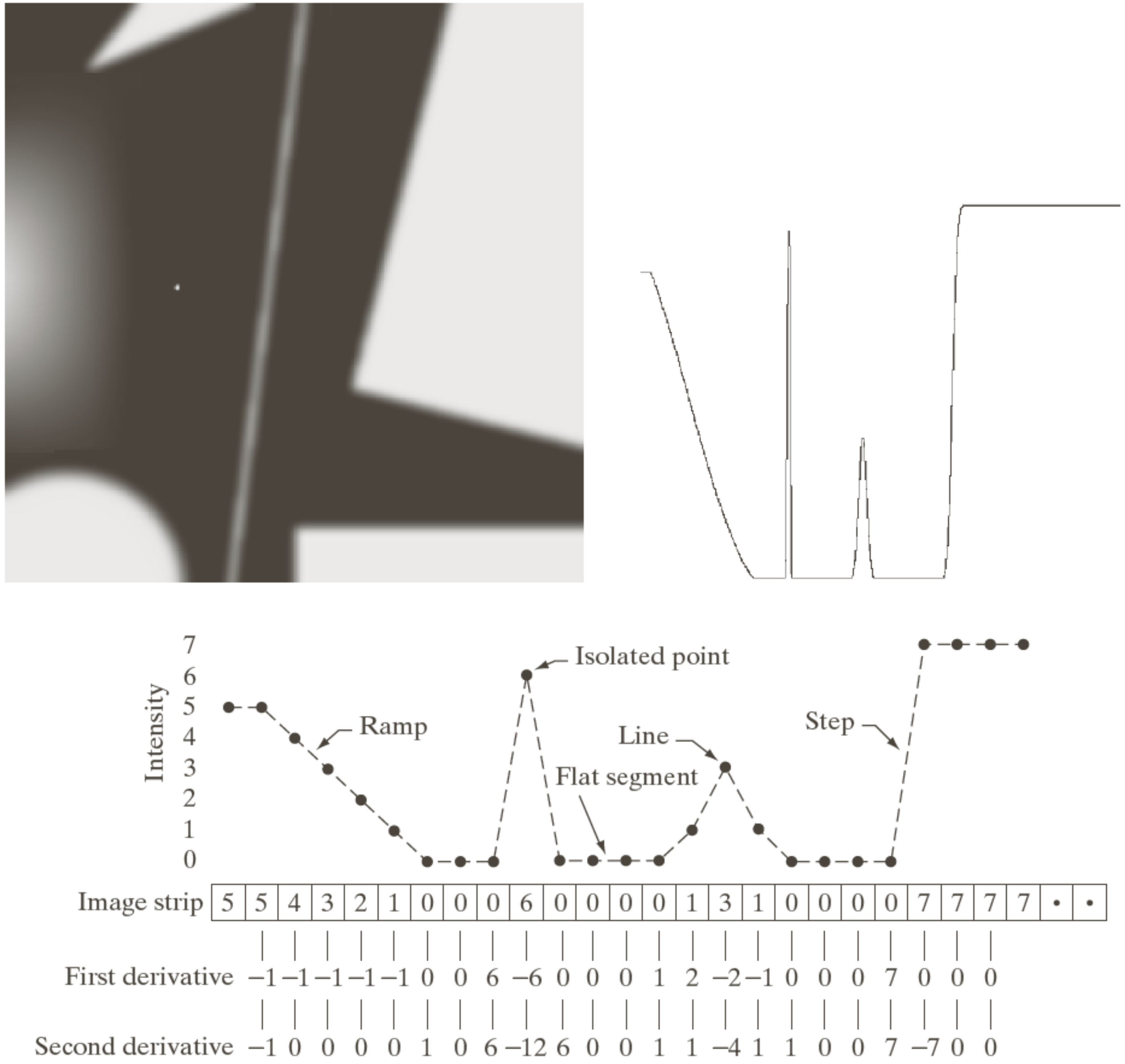
Background

- First-order derivative

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

- Second-order derivative

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



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

- Based on the second derivative using Laplacian: $\nabla^2 f(x,y) = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$

- Partials are given by

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

- Laplacian is given by

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

- Implemented by a 4-neighborhood mask, or can be extended to an 8-neighborhood mask

1	1	1
1	-8	1
1	1	1

Detection of Isolated Points

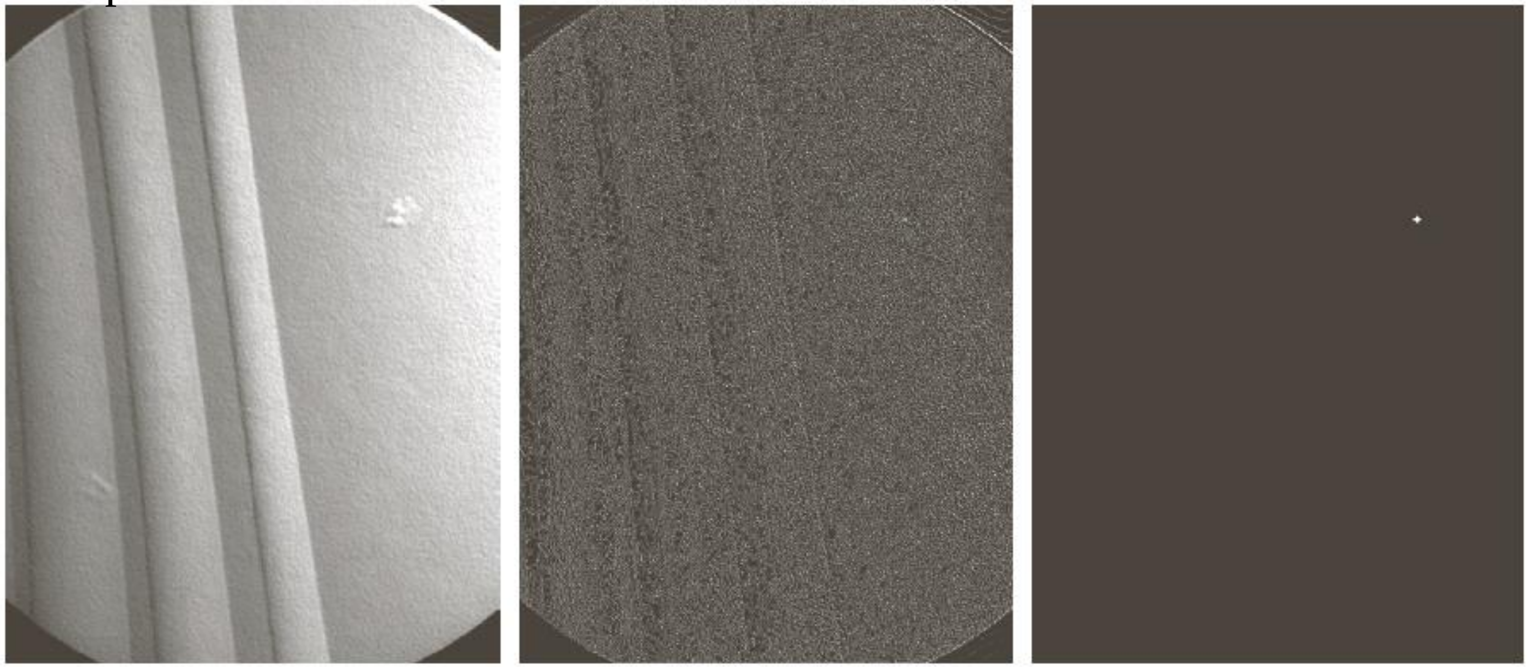
Select a point in the output image if its response to the mask exceeds a specified threshold; deselect others. Eq. 10.2-8:

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

T selected as 90% of highest absolute pixel value in figure. Note that in constant intensity areas, the mask response will be zero.

1	1	1
1	−8	1
1	1	1

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



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

Line Detection

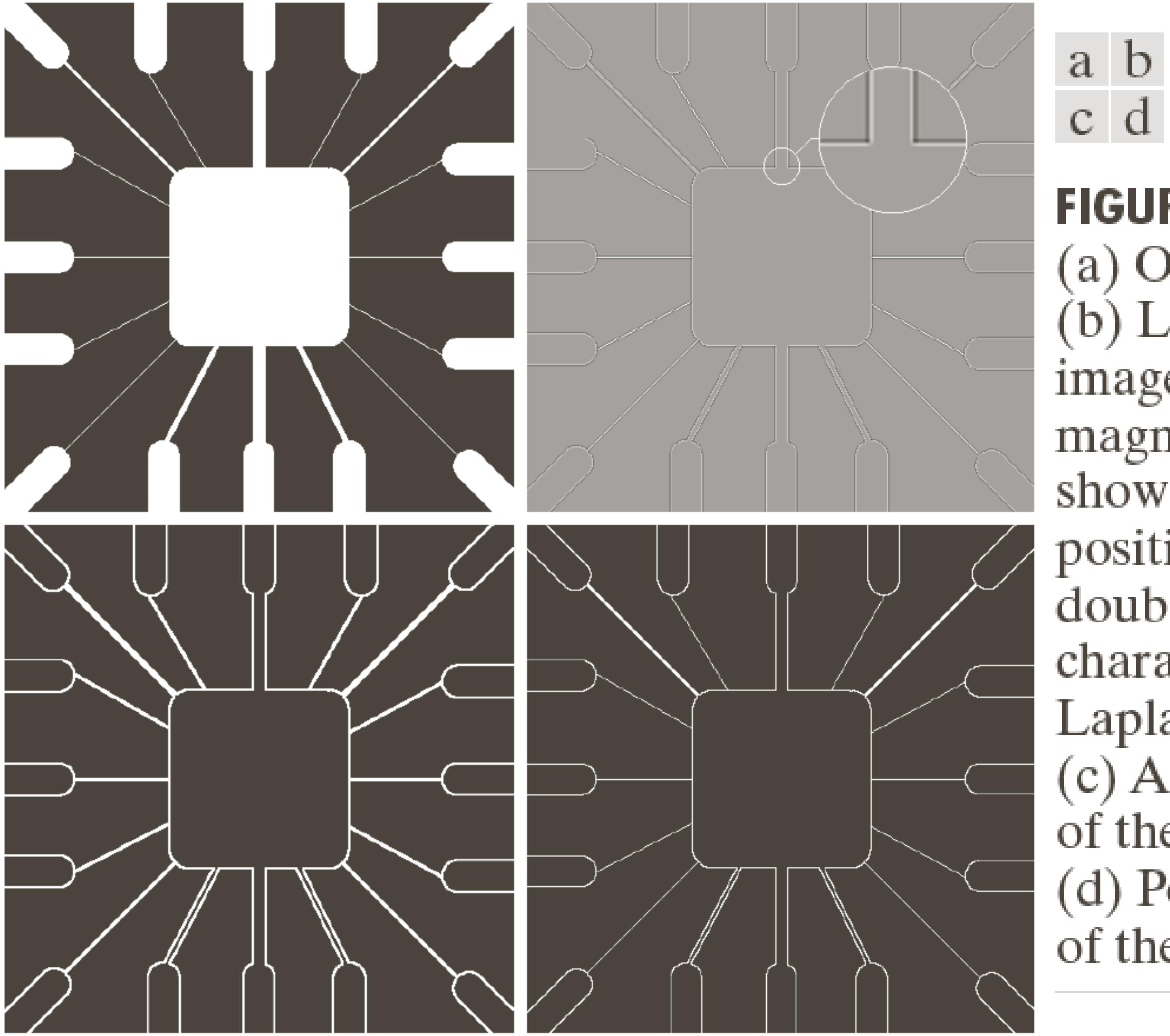


FIGURE 10.5
(a) Original image.
(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.
(c) Absolute value of the Laplacian.
(d) Positive values of the Laplacian.

Detecting Lines in Specified Directions

- 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.

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

2	−1	−1
−1	2	−1
−1	−1	2

Vertical

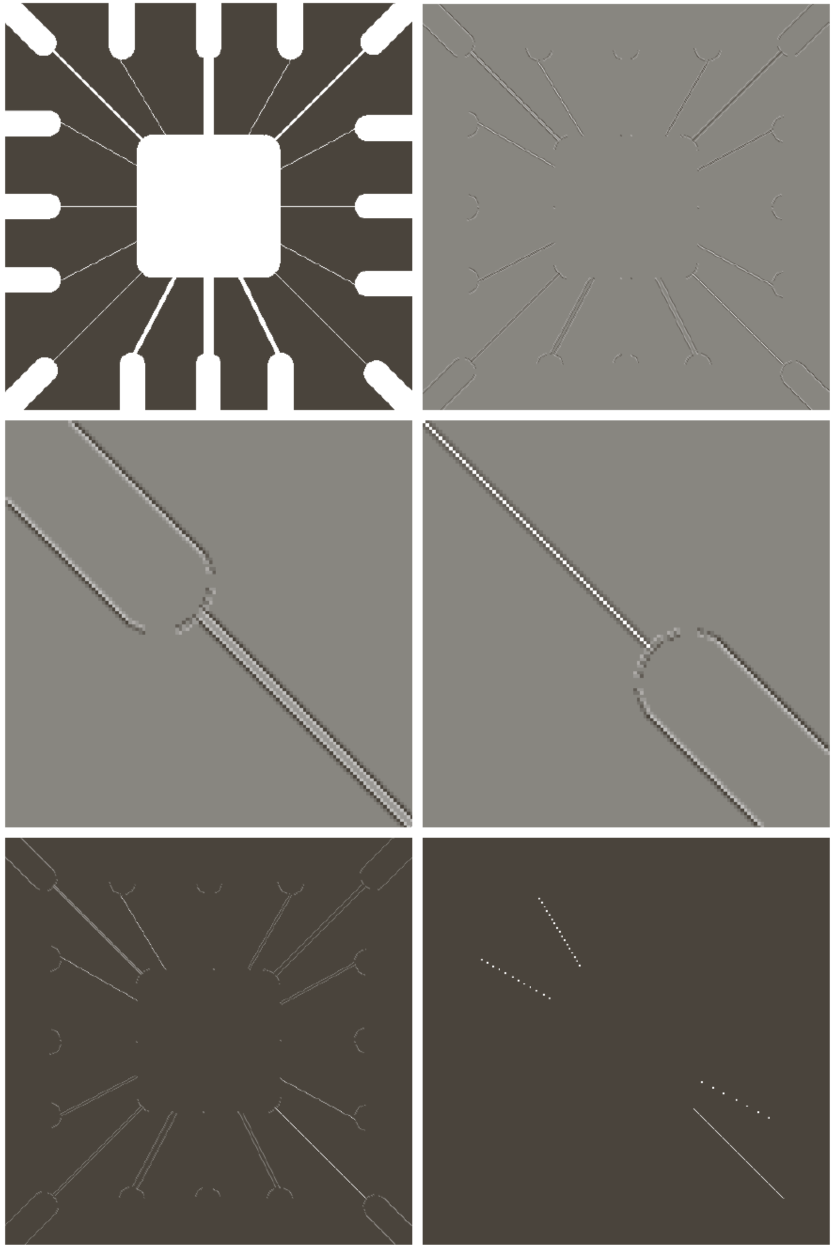
−1	2	−1
−1	2	−1
−1	2	−1

−45°

−1	−1	2
−1	2	−1
2	−1	−1

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

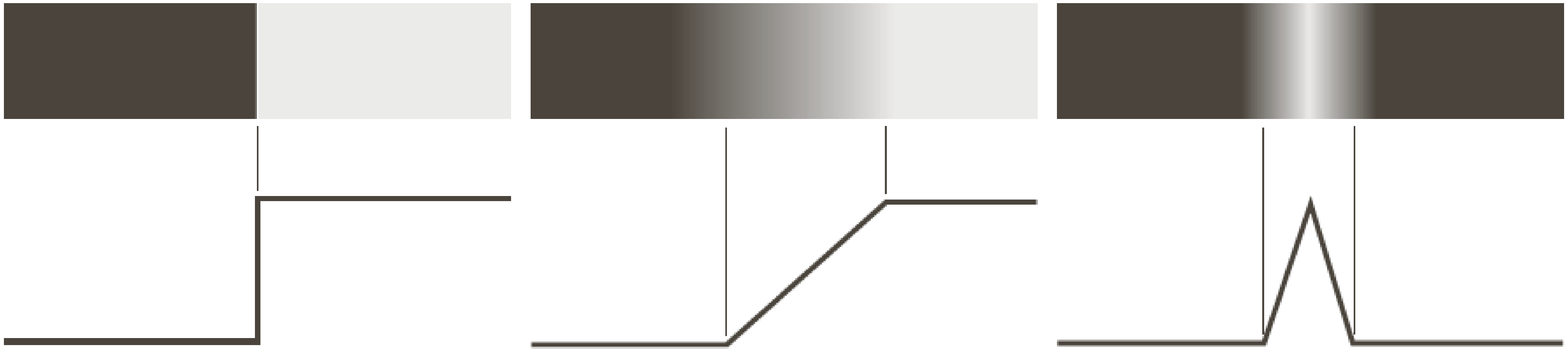
Detecting Lines in Specified Directions



a b
c d
e f

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.)

- 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.

Edge Detection

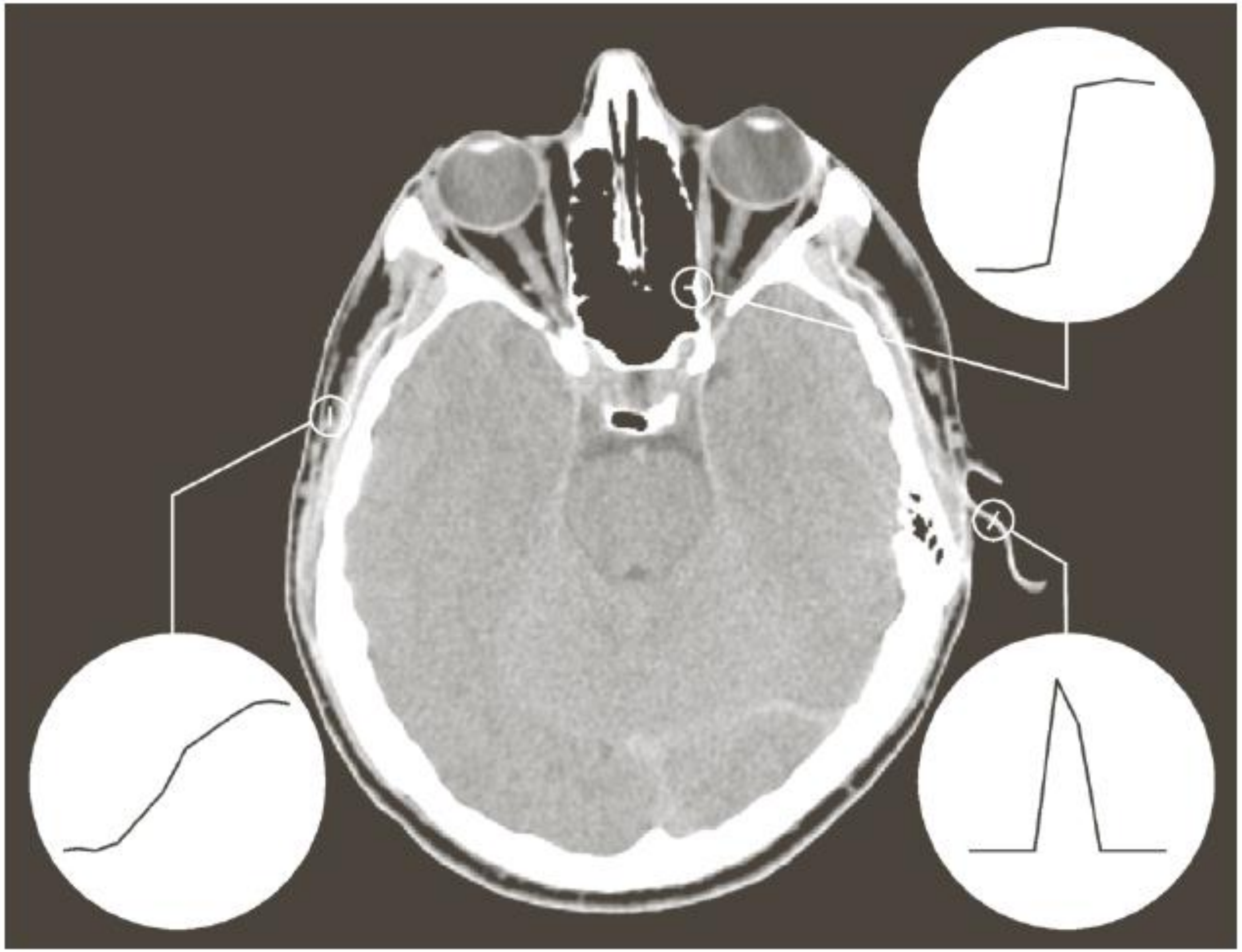
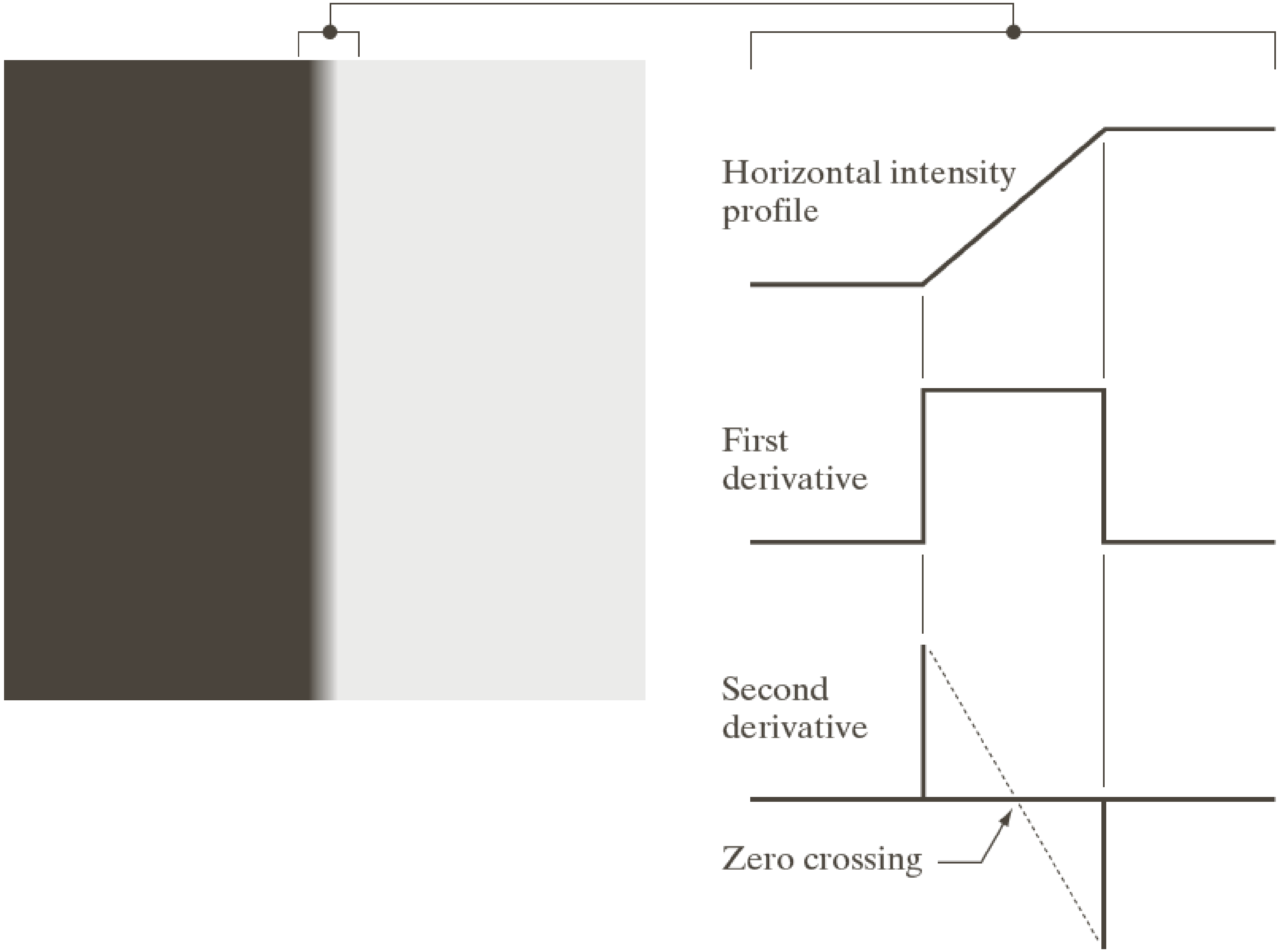


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Edge Detection



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.

Edge Detection

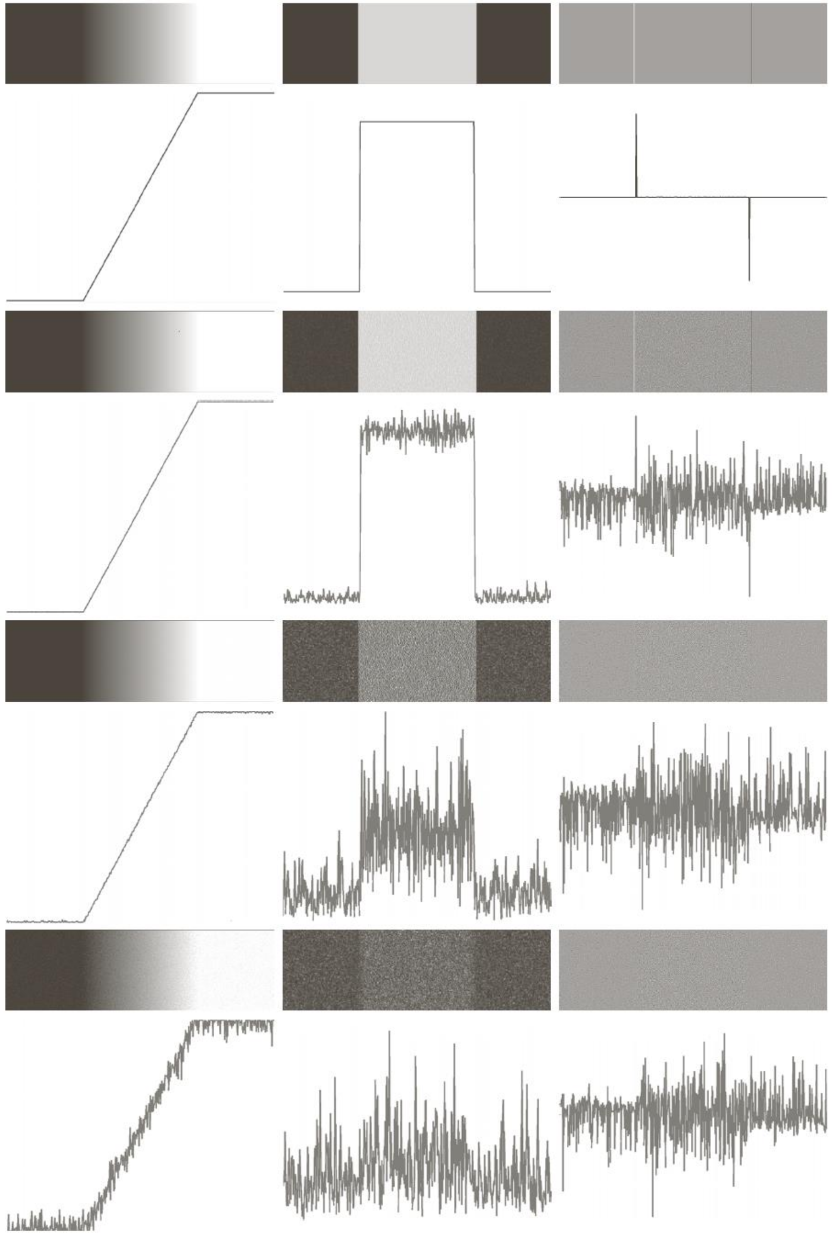


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images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

Basic Edge Detection by Using First-Order Derivative

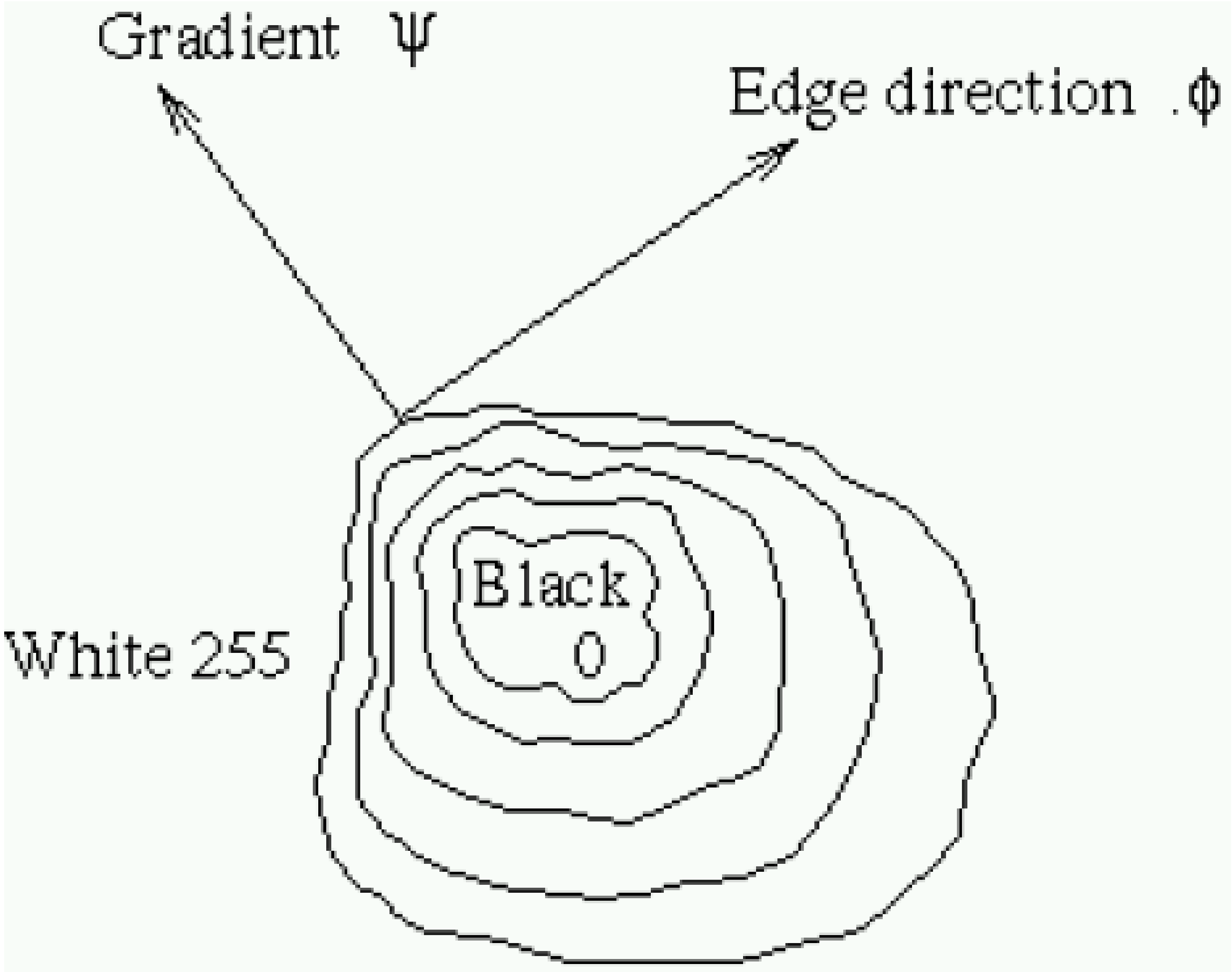
- Find edge strength and direction at location (x, y) in image f
- Use the gradient ∇f, defined as a vector $$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$
- Vector ∇f points in the direction of greatest rate of change of f at (x, y)
- Magnitude of vector $M(x, y)$ gives the rate of change in the direction of the gradient vector $$M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$
- Direction of the ∇f with respect to x axis is given by $$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$
- The direction of the edge: $\varphi = \alpha - 90°$

Basic Edge Detection by Using First-Order Derivative

- Gradient vector may be called *edge normal*: $\nabla f = \mathrm{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$
- If gradient vector is normalized to unit length, the resulting vector may be called *edge unit normal*: ∇f / mag(∇f)
- In practice, sometimes the magnitude is approximated by

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

Basic Edge Detection by Using First-Order Derivative



Basic Edge Detection by Using First-Order Derivative

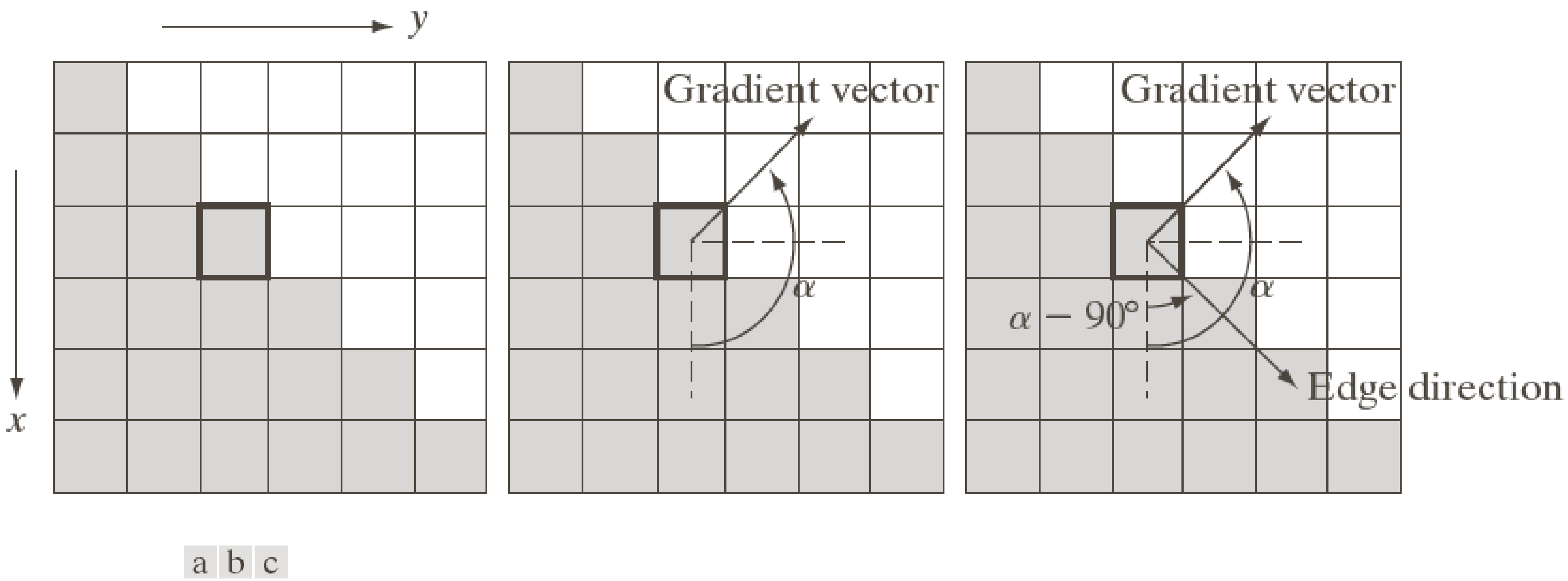


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Basic Edge Detection by Using First-Order Derivative

−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.

Basic Edge Detection by Using First-Order Derivative

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.

Basic Edge Detection by Using First-Order Derivative

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, $|g_x| + |g_y|$.

Basic Edge Detection by Using First-Order Derivative

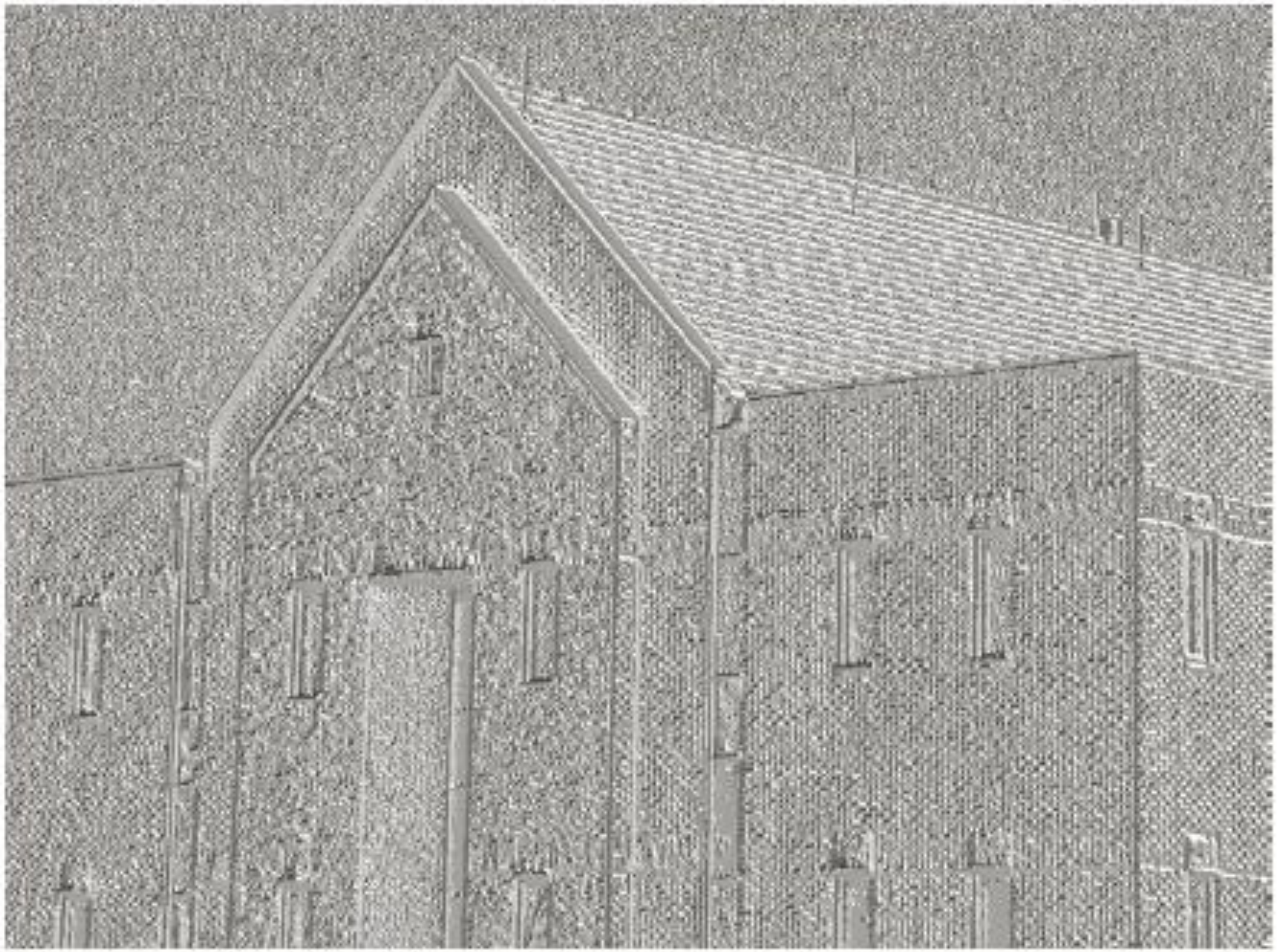


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

Basic Edge Detection by Using First-Order Derivative

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.

Basic Edge Detection by Using First-Order Derivative

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).

Basic Edge Detection by Using First-Order Derivative

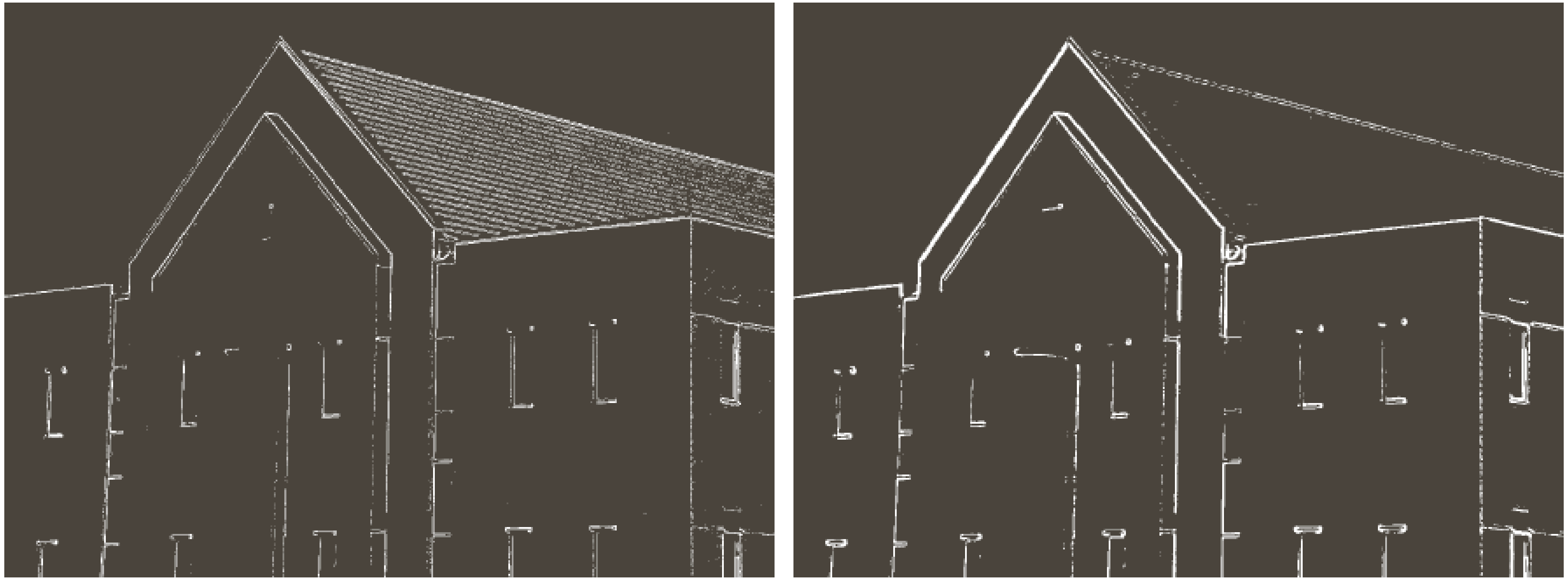
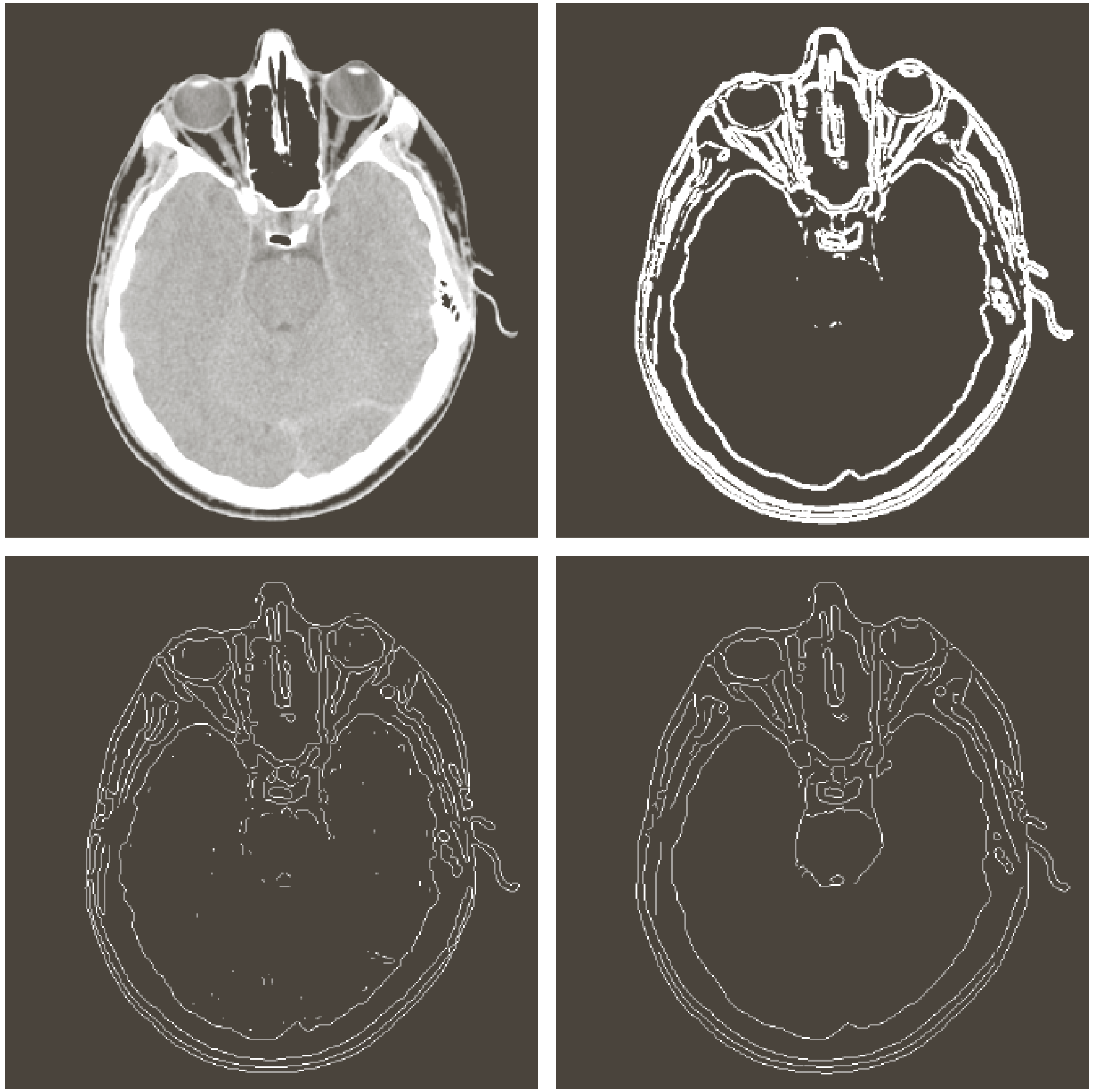


FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using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

The Marr-Hildreth & Canny Edge Detection



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; σ=2; and a mask size of 13x13

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

Thresholding

- The segmented image $g(x,y)$ is given by

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

- If T is constant in the above expression, the thresholding is called *global thresholding*. When the value of T changes over an image, the thresholding is called *variable thresholding*

- Multiple thresholding:

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

Thresholding

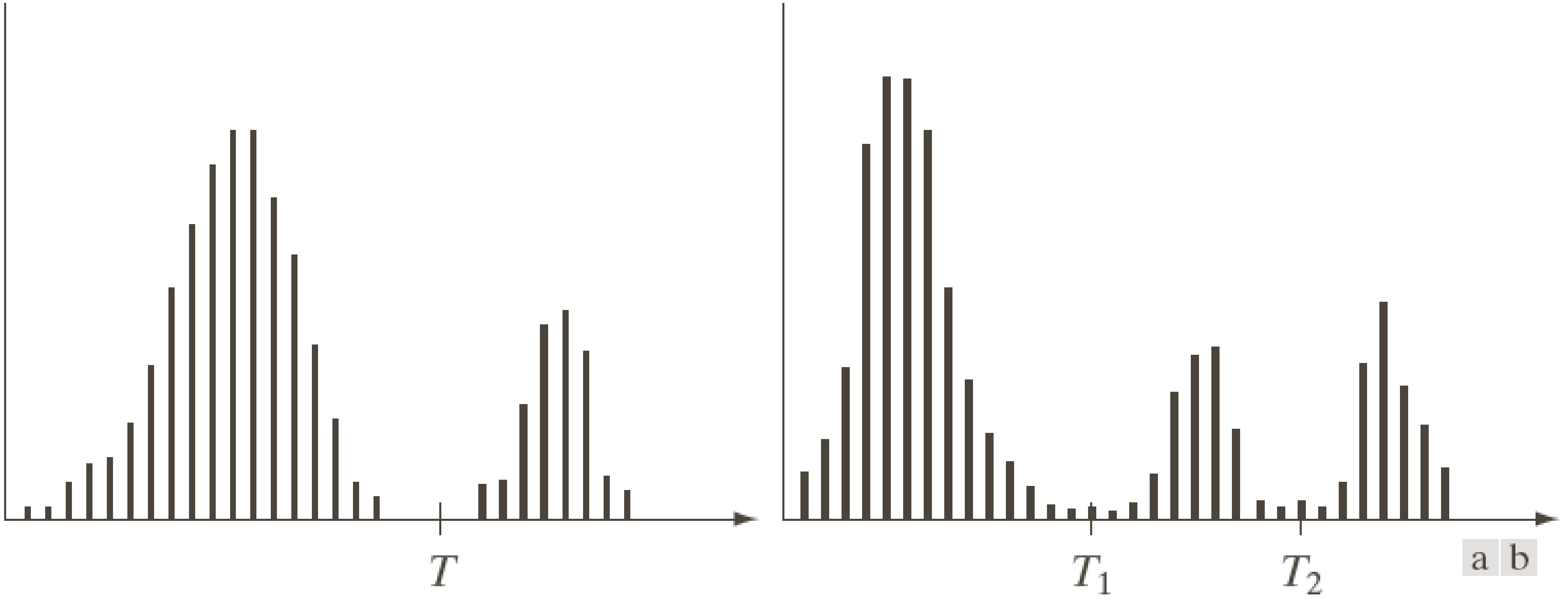


FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

The Role of Noise in Image Thresholding

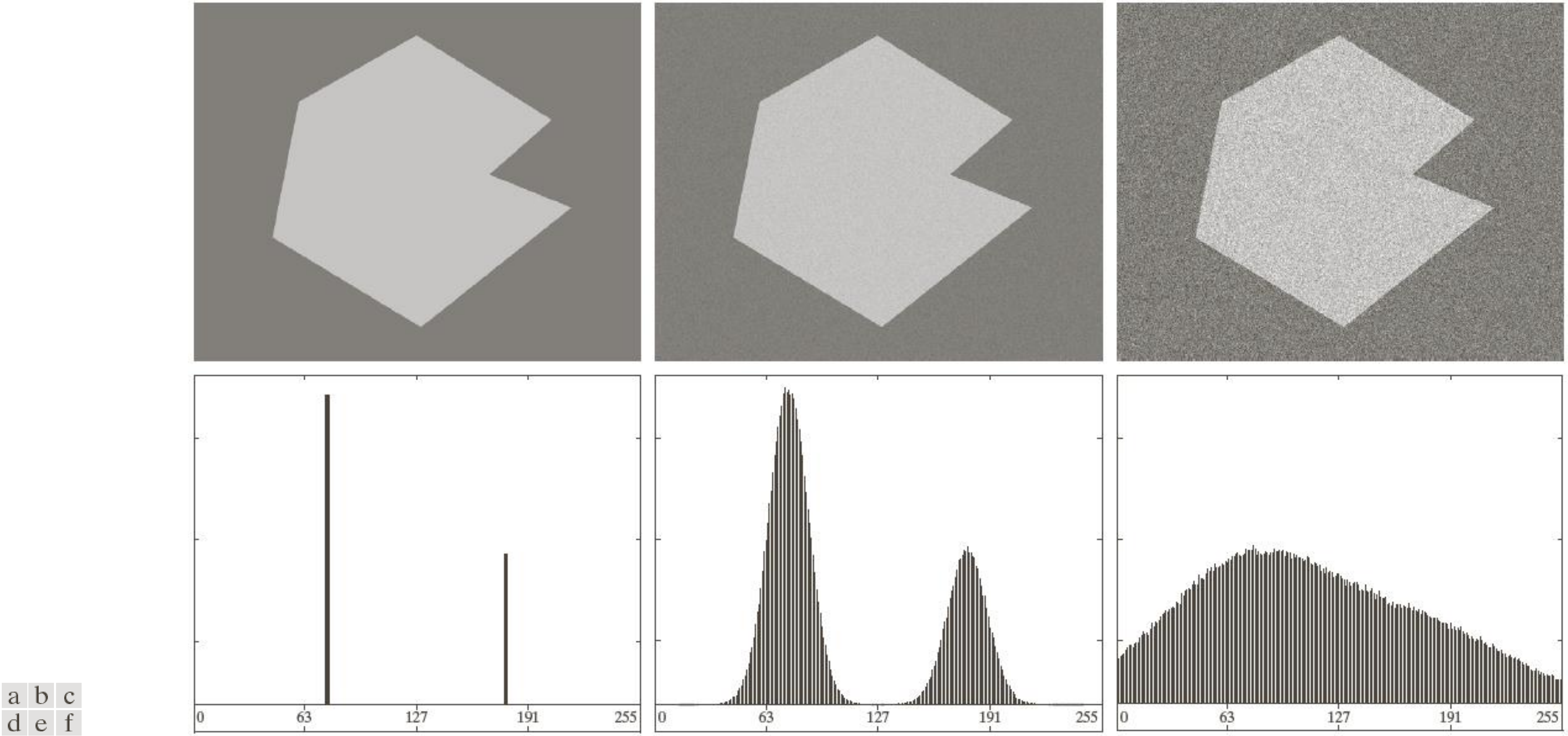


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

The Role of Illumination and Reflectance

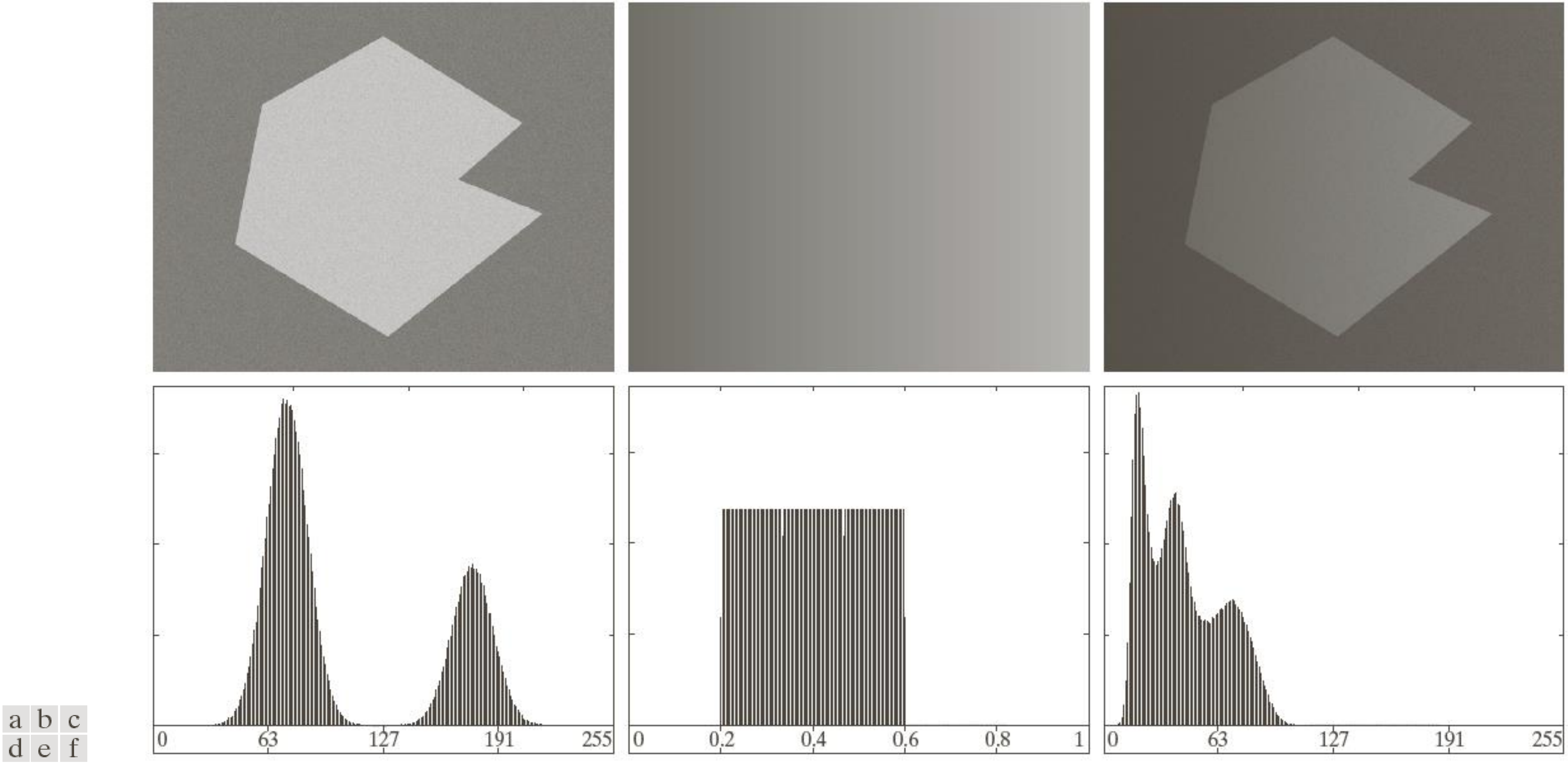


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

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: G_1 consisting of all pixels with intensity values > T and G_2 consisting of pixels with values $\leq T$.
3. Compute the average intensity values m_1 and m_2 for the pixels in G_1 and G_2, respectively.
4. Compute a new threshold value: $T = \frac{1}{2}(m_1 + m_2)$
5. Repeat Steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT.

Basic Global Thresholding

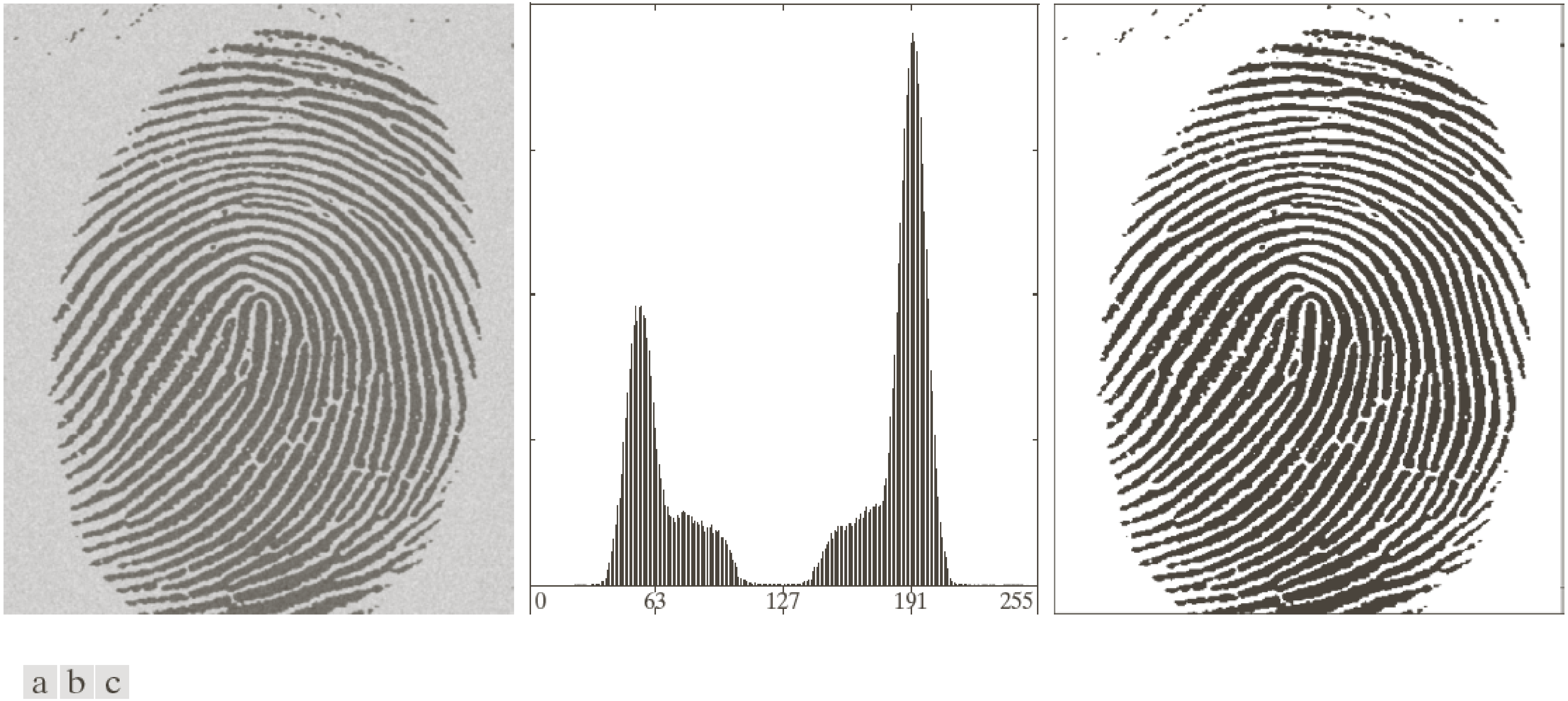


FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

- Principle: maximizing the between-class variance
- Let {0, 1, 2, ..., *L*-1} denote the distinct intensity levels in a digital image of size *M* x *N* pixels, and let n_i denote the number of pixels with intensity *i*.

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

- k is a threshold value, $C_1 \rightarrow [0,k]$, $C_2 \rightarrow [k+1, L-1]$

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

- 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

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

$P_1 m_1 + P_2 m_2 = m_G$ (Global mean value)

Optimum Global Thresholding Using Otsu's Method

- Between-class variance, σ^2_B is defined as

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

$$\begin{aligned}\sigma_B^2 &= P_1 P_2 (m_1 - m_2)^2 \\ &= \frac{(m_G P_1 - m)^2}{P_1(1 - P_1)}\end{aligned}$$

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

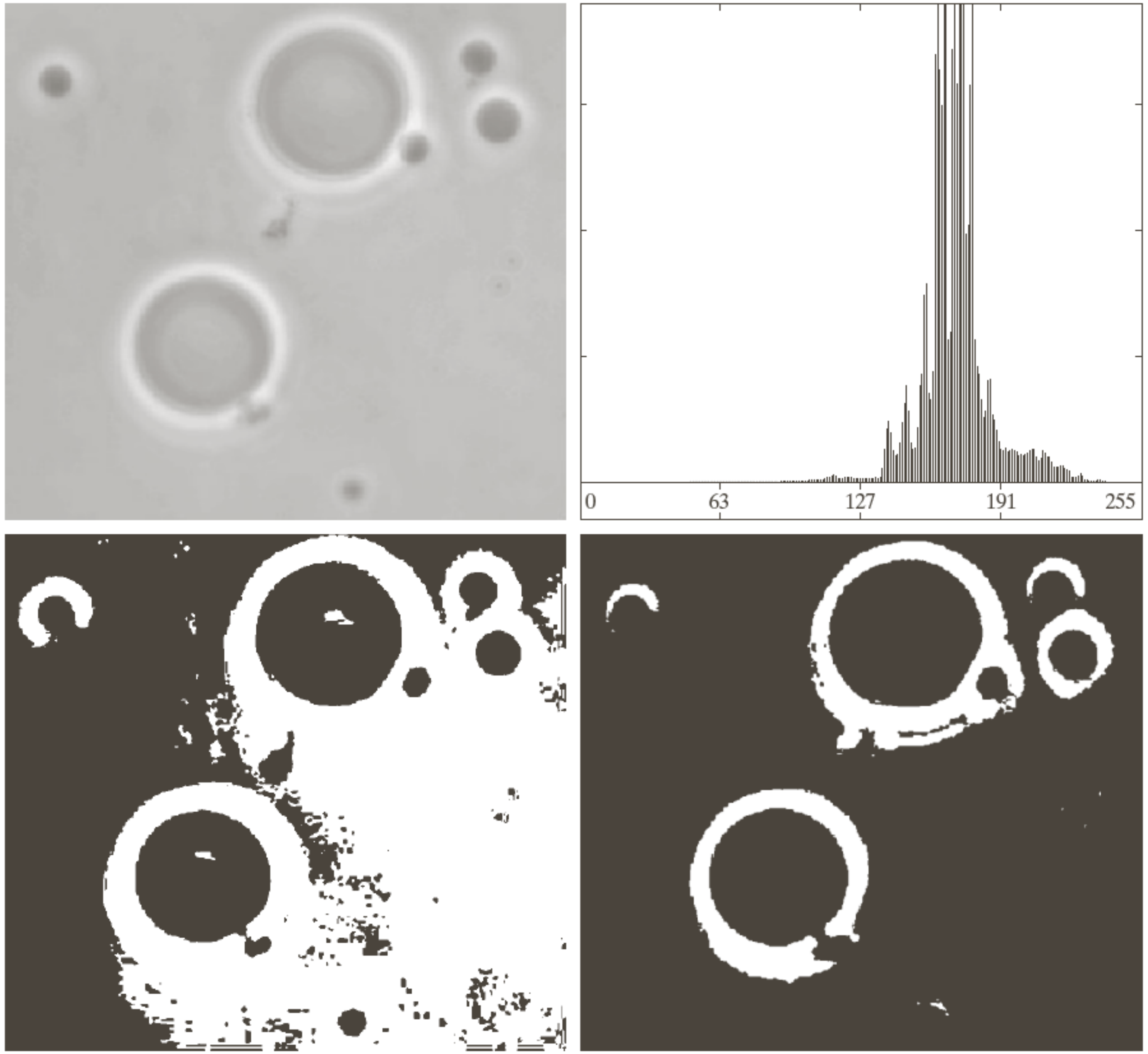
$$\sigma_B^2(k^*), \qquad \sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$ and $$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > k^* \\ 0 & \text{if } f(x,y) \le k^* \end{cases}$$

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

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, $P1(k)$, 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.
6. Obtain the Otsu’s threshold, k*.
7. Obtain the separability measure.

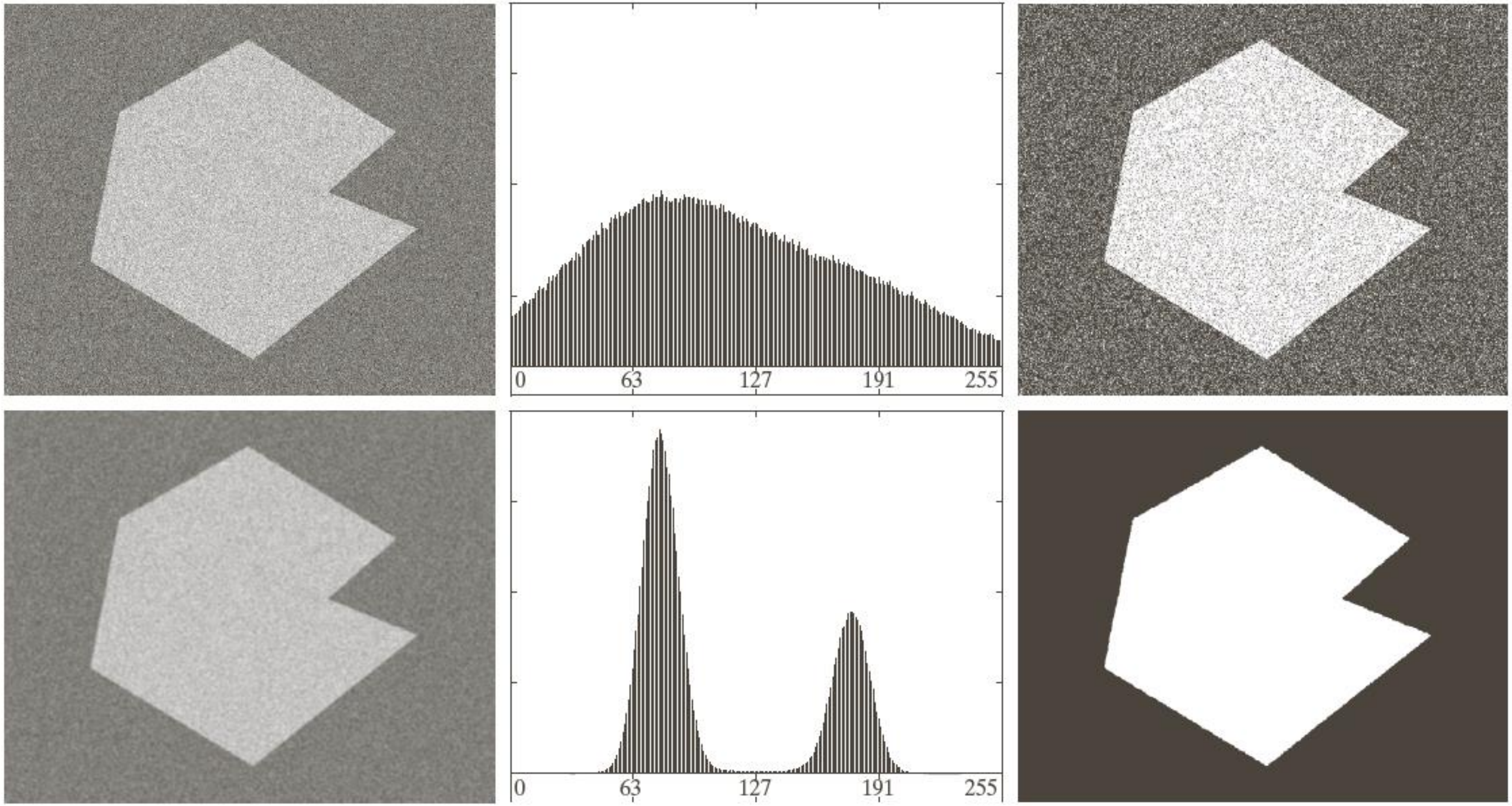
Otsu's Algorithm



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.)

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

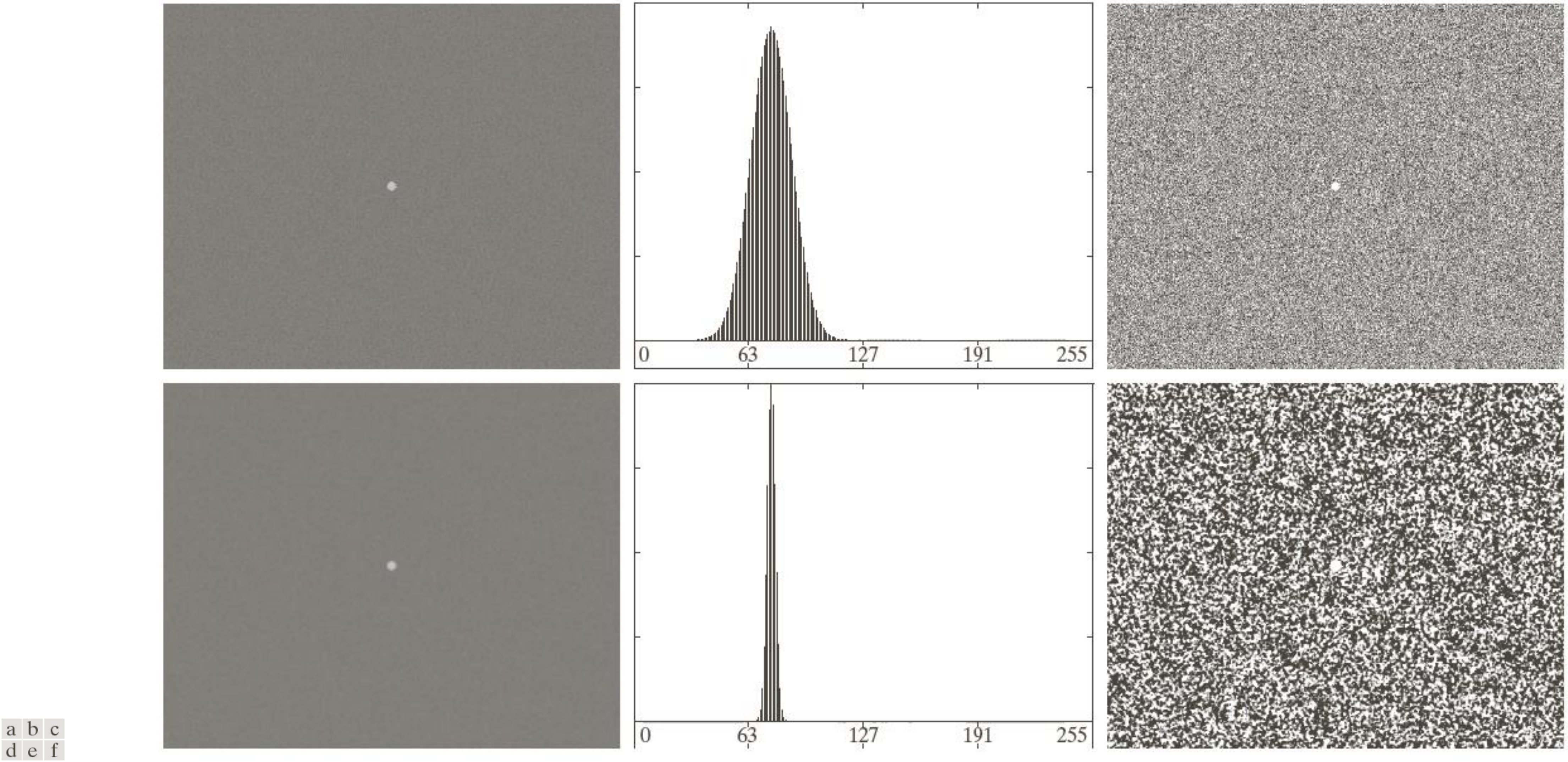


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

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

Using Edges to Improve Global Thresholding

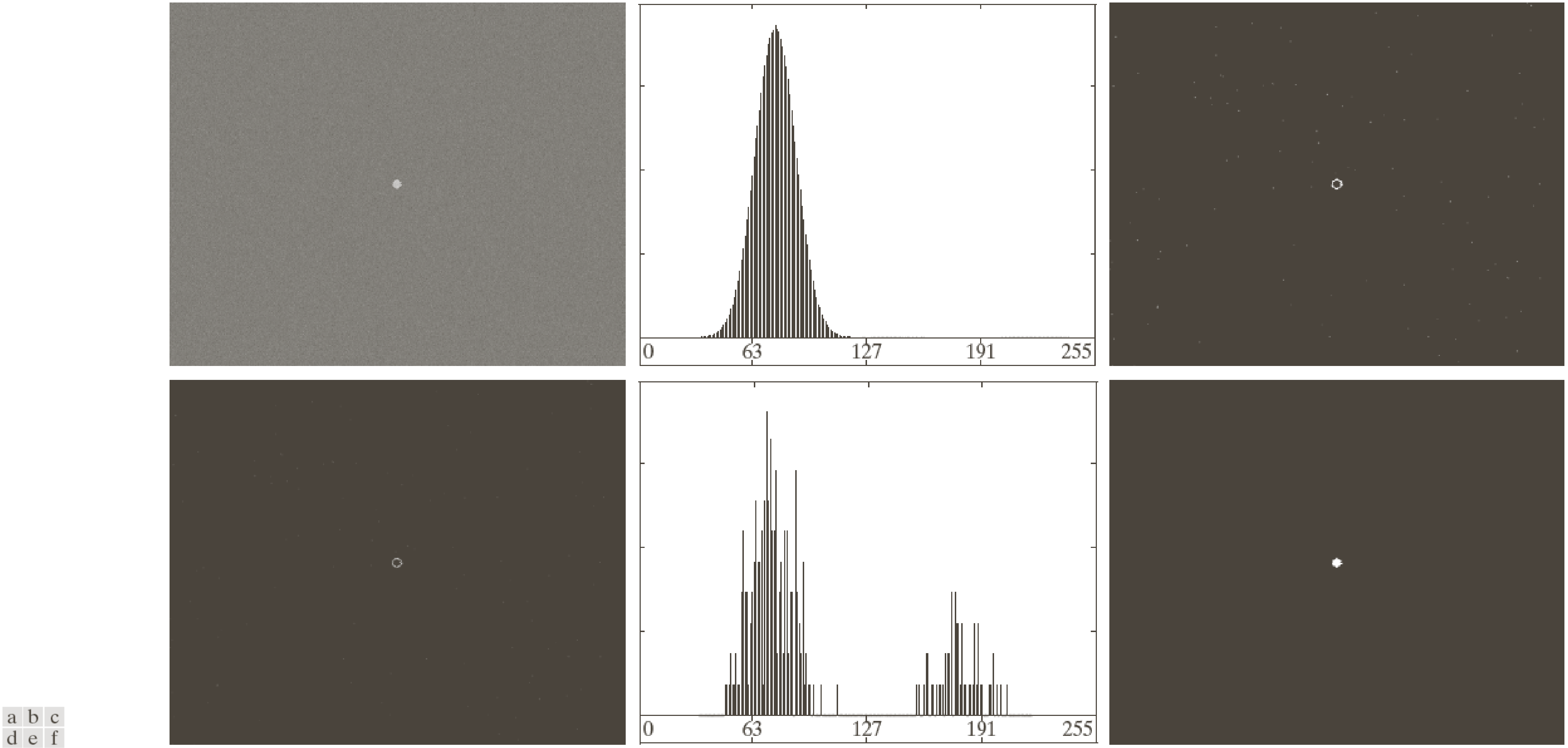
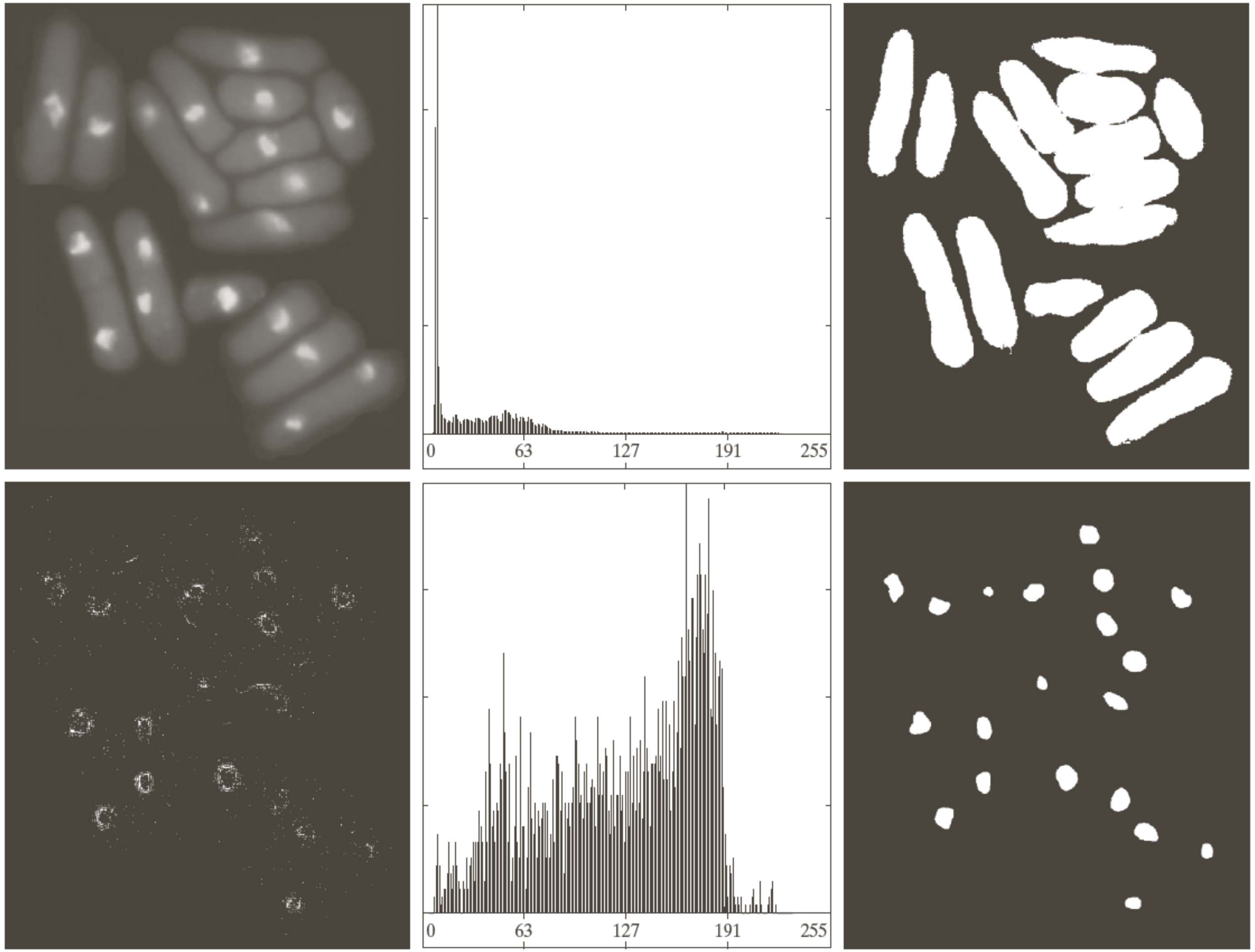


FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based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

Using Edges to Improve Global Thresholding



a b c
d e f

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

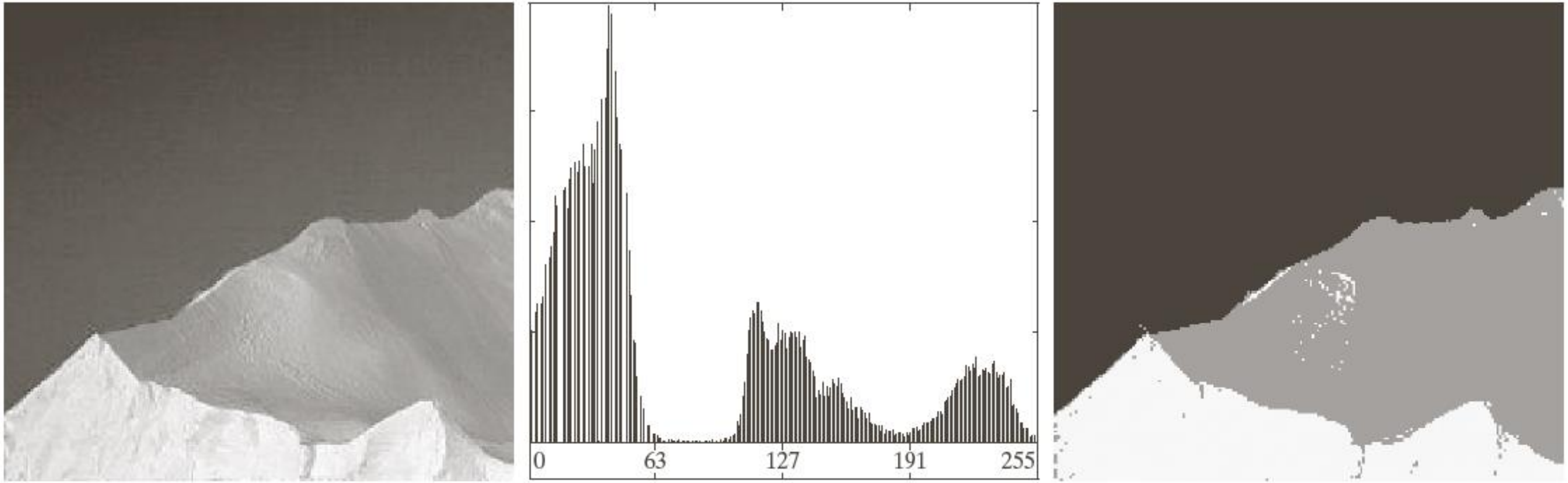
- In the case of K classes, C_1, C_2, ..., C_K , the between-class variance is

$$\sigma_B^2 = \sum_{k=1}^{K} P_k \left(m_k - m_G\right)^2 \quad \text{where} \quad P_k = \sum_{i \in C_k} p_i \quad \text{and} \quad m_k = \frac{1}{P_k} \sum_{i \in C_k} i p_i$$

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

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

Multiple Thresholds



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.

Variable Thresholding: Image Partitioning

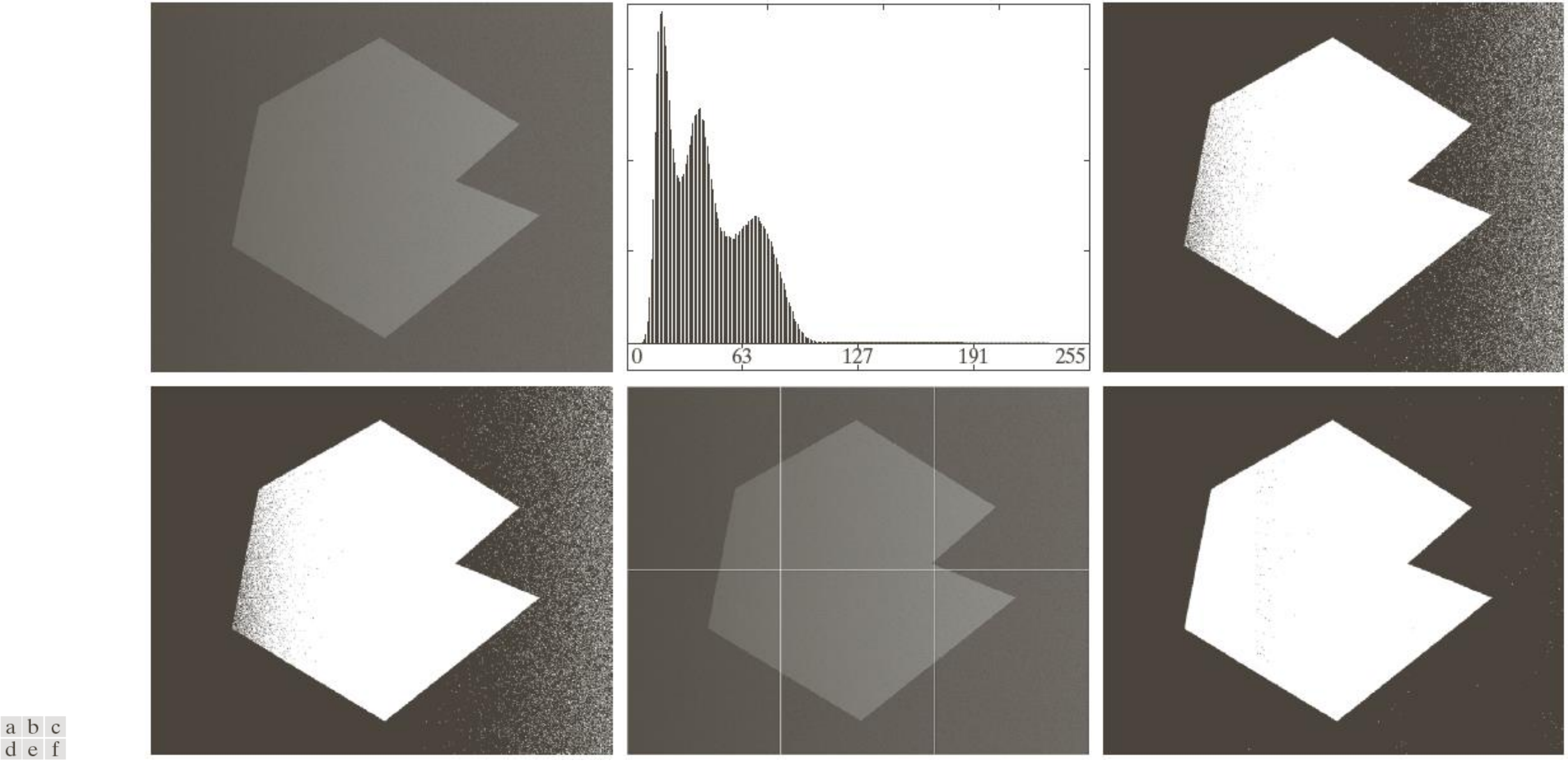


FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

Variable Thresholding: Image Partitioning

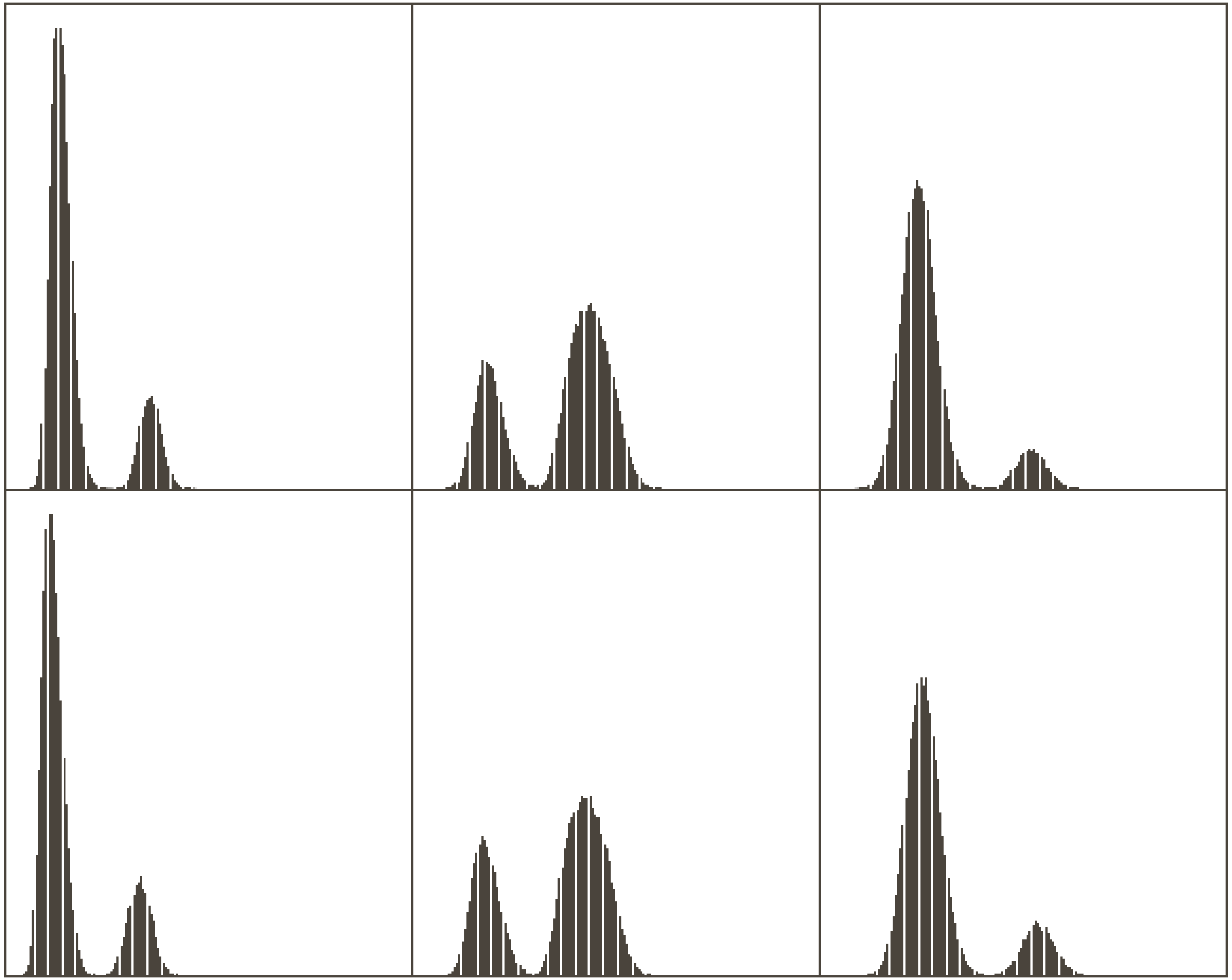


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

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,

$$T_{xy} = a\,\sigma_{xy} + b\,m_{xy}$$

- If the background is nearly constant,

$$T_{xy} = a\,\sigma_{xy} + b\,m$$

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

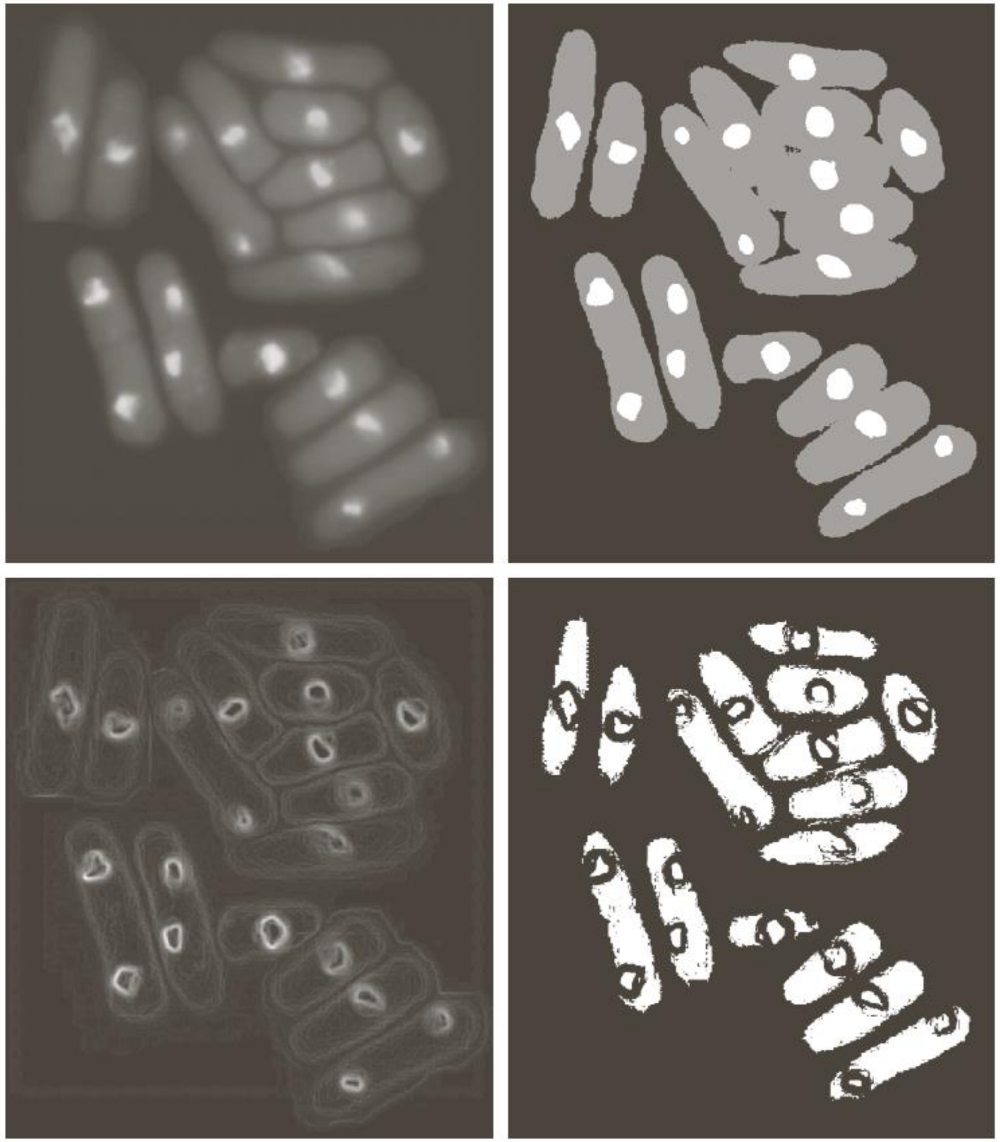
- A modified thresholding

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

e.g.,

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

Variable Thresholding Based on Local Image Properties



a b
c d

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

- Let z_{k+1} denote the intensity of the point encountered in the scanning sequence at step 1.
- The moving average (mean intensity) at this new point is given by

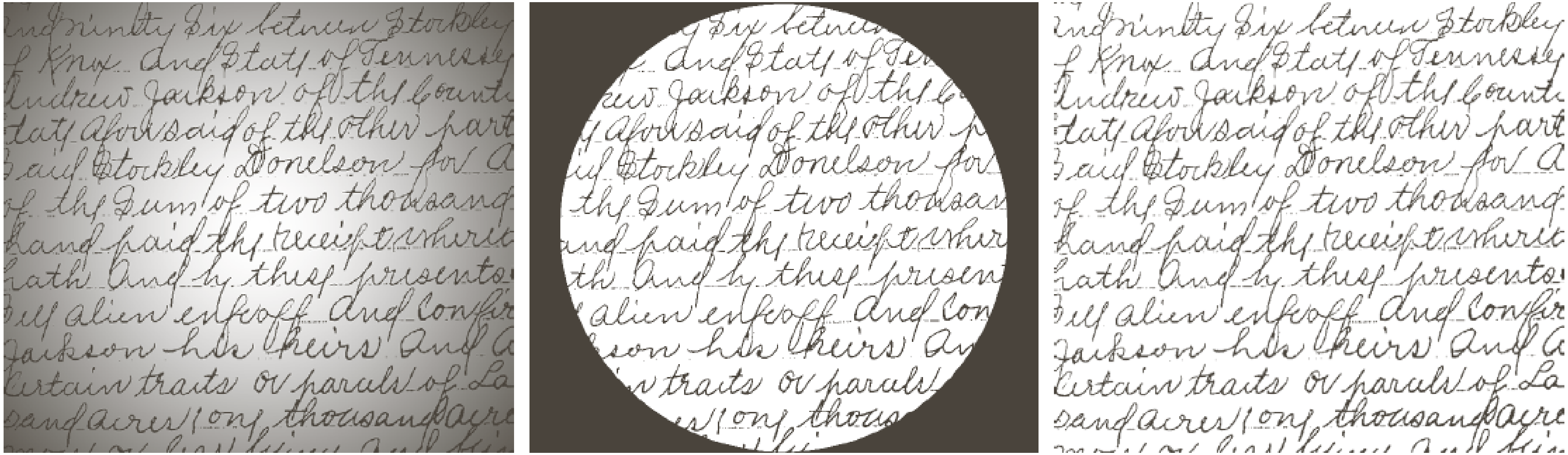
$$m(k+1) = \frac{1}{n} \sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n}(z_{k+1} - z_k)$$

where n denotes the number of points used in computing the average and $m(1) = z_1/n$, the border of the image were padded with n -1 zeros.

- If m_{xy} is the moving average at point (x,y), the threshold at the point T_{xy} is given by cm_{xy} where c is a positive scalar

Variable Thresholding Using Moving Averages

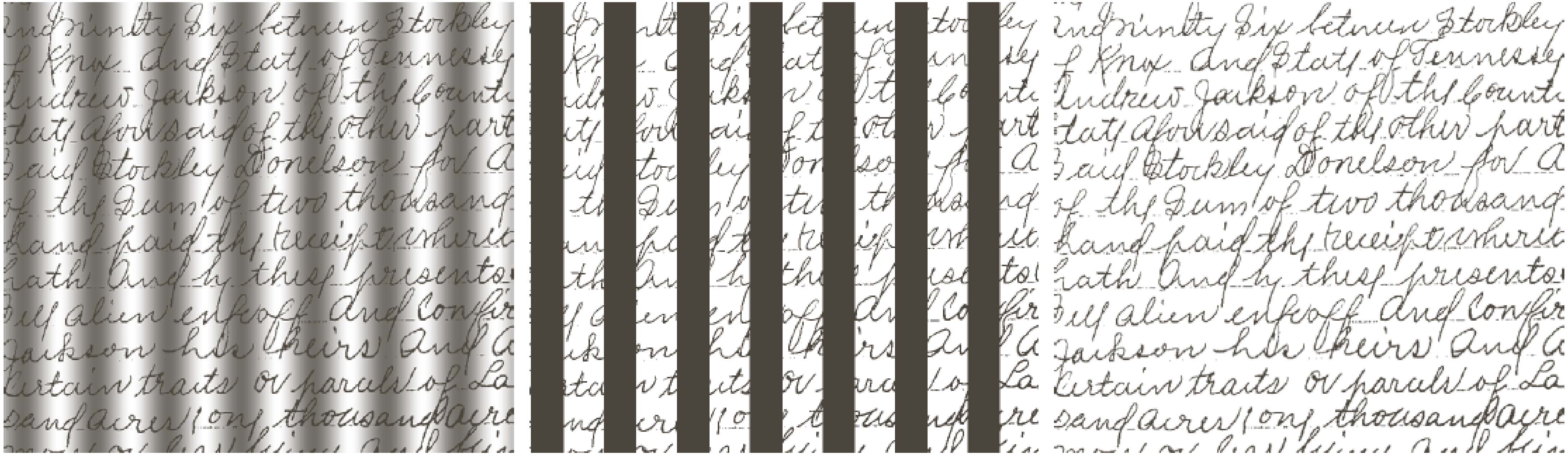
N = 20
b=0.5



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.

Variable 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 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 grows 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.

k-means Clustering

- Partition a set of observations Q into k clusters
- Each observation assigned to the cluster with the nearest mean
- Each mean is called the *prototype* of its cluster

A set of vector observations $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_\mathbf{Q}\}$

- Each vector is of the form $\mathbf{z} = [z_{1,}; z_2, \ldots, z_n]^T$
- Each component of vector $\mathbf{z}$ represents a numerical pixel attribute: grayscale value or a 3D vector for RGB images

k-means Clustering

- Objective of k-means clustering is to partition the set Q into k ($k \leq Q$) disjoint cluster sets $C = \{C_1, C_2, \ldots, C_k\}$ so that the following criterion for optimality is satisfied

$$\underset{C}{\arg\min} \left(\sum_{i=1}^{k} \sum_{\mathbf{z} \in C_i} ||\mathbf{z} - \mathbf{m}_i||^2 \right)$$

m_i is the mean vector or centroid of cluster C_i

- Find the sets $C = \{C_1, C_2, \ldots, C_k\}$ such that the sum of distances from each point in a set to the centroid of that set is minimized

* NP-hard problem with no practical solution

Algorithm for *k*-means

1. Initialize: Specify an initial set of means $\mathbf{m}_i(1)$; $i = 1, 2, \ldots, k$

2. Assign samples to clusters: Each sample assigned to cluster with closest centroid

$$\mathbf{z}_q \rightarrow C_i \text{ if } \|\mathbf{z}_q - \mathbf{m}_i\|^2 < \|\mathbf{z}_q - \mathbf{m}_j\|^2 \qquad j = 1, 2, \ldots, k \ (j \neq i);\ q = 1, 2, \ldots, Q$$

3. Update cluster centroid

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{z} \in C_i} \mathbf{z} \qquad i = 1, 2, \ldots, k$$

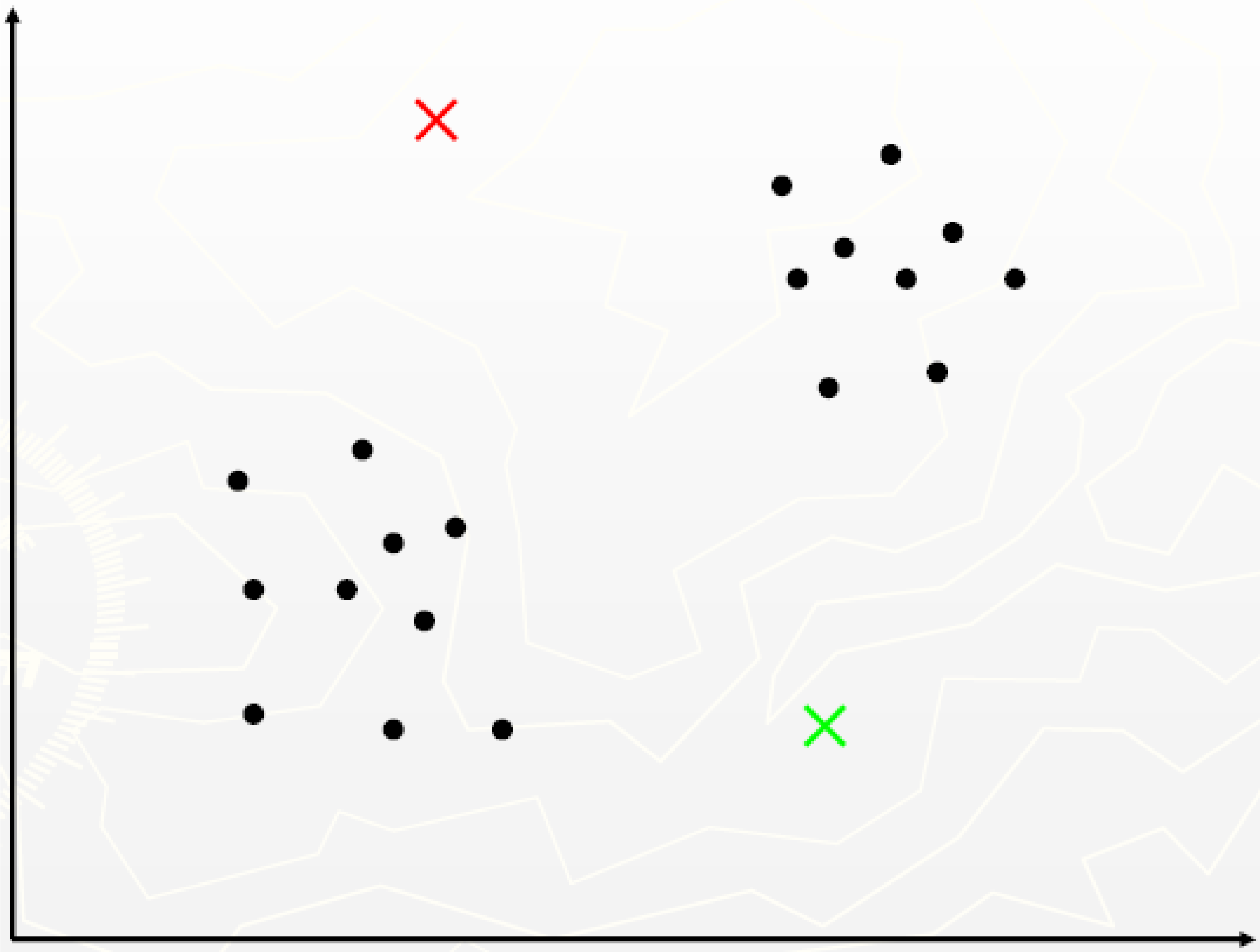
4. Test for completion
   - Compute Euclidean norms of differences between the centroid in the current and previous steps
   - Compute residual error E as the sum of k norms
   - If $E \leq T$ stop else go to step 2 (T is a prespecified nonnegative threshold)

Algorithm for *k*-means

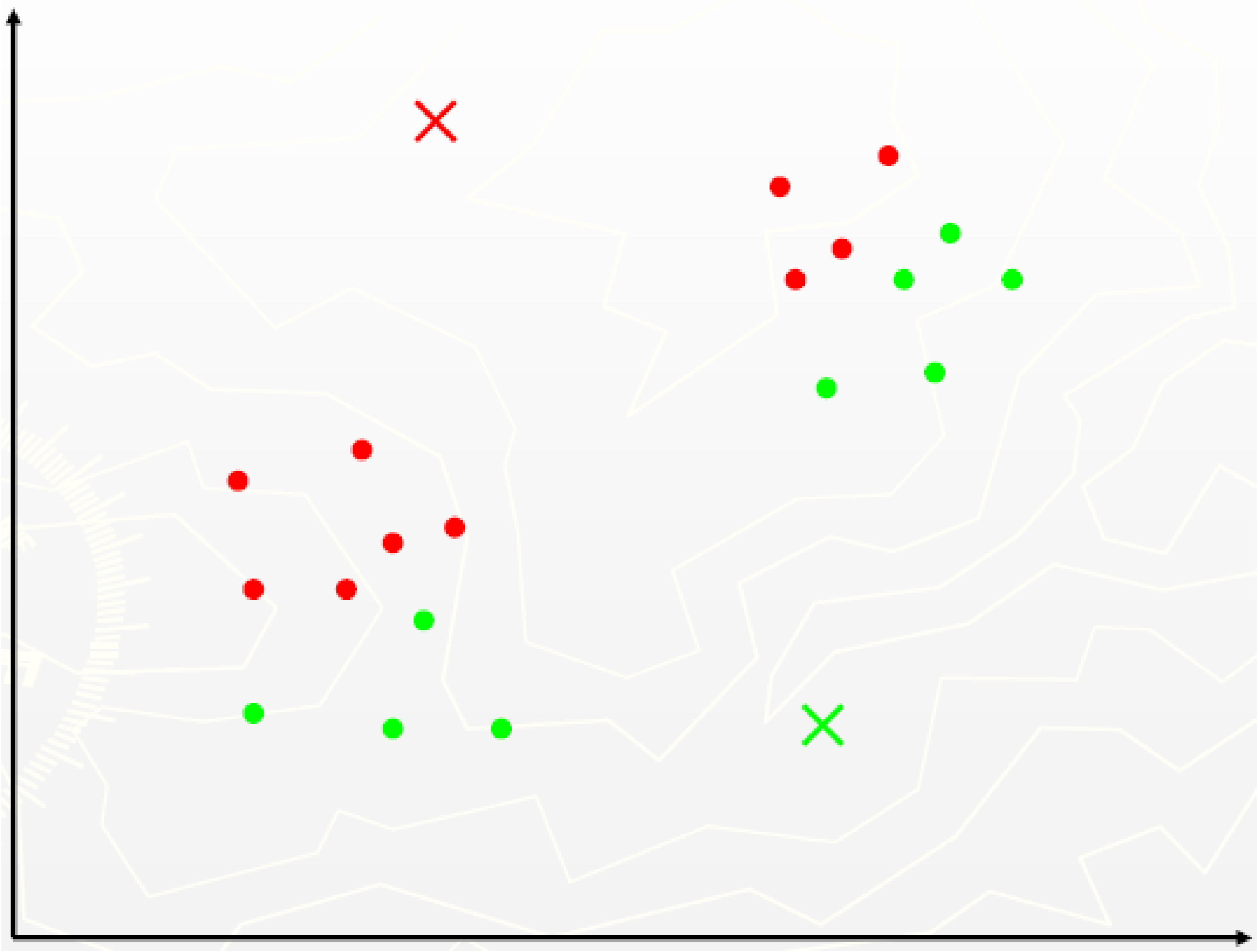
- Algorithm converges in a finite number of iterations when $T = 0$
- Not guaranteed to yield global minima
- Result depends on initial values chosen for $\mathbf{m}_i$
- Choice of k gives you the number of segmented regions

k-means Clustering

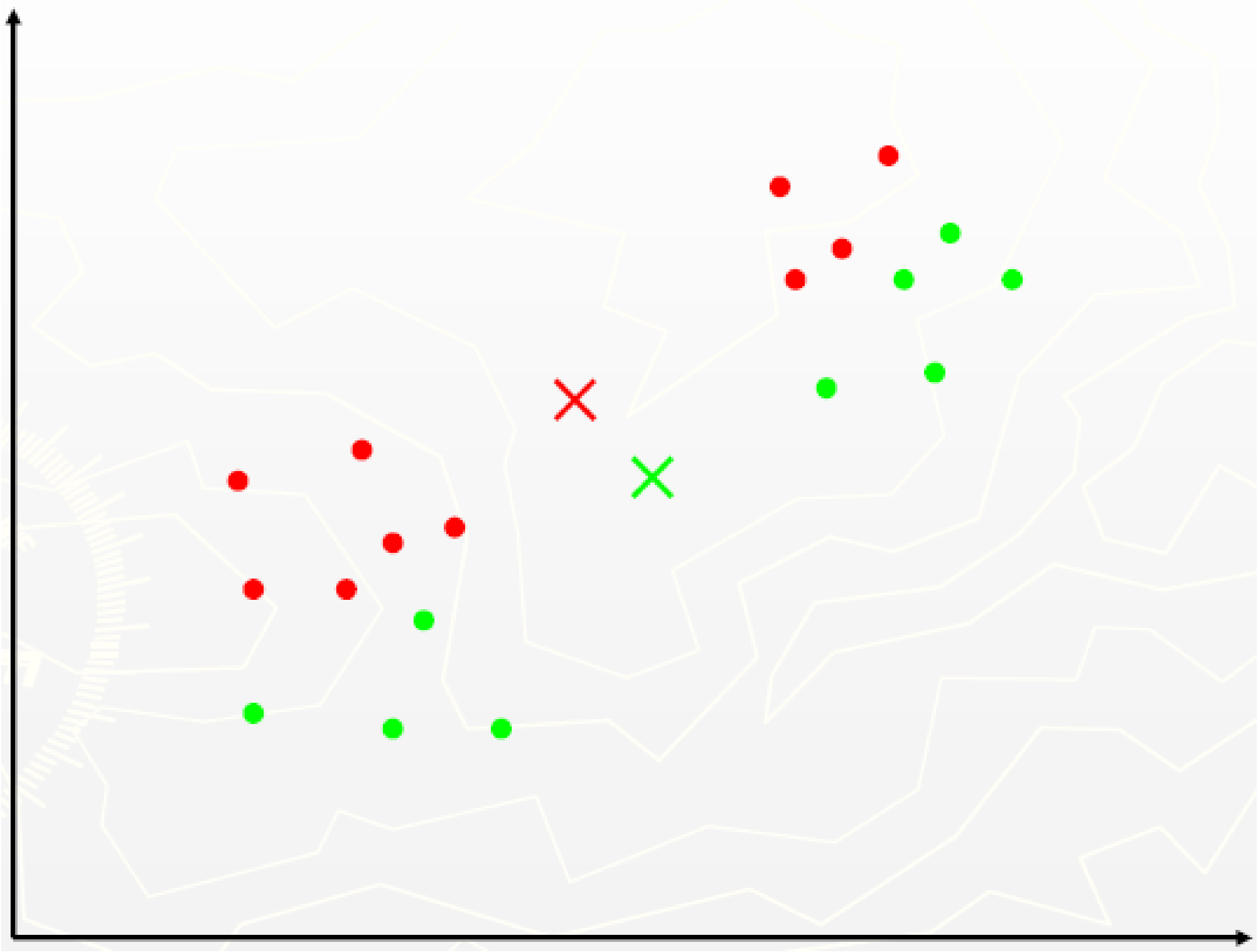
k-means Clustering



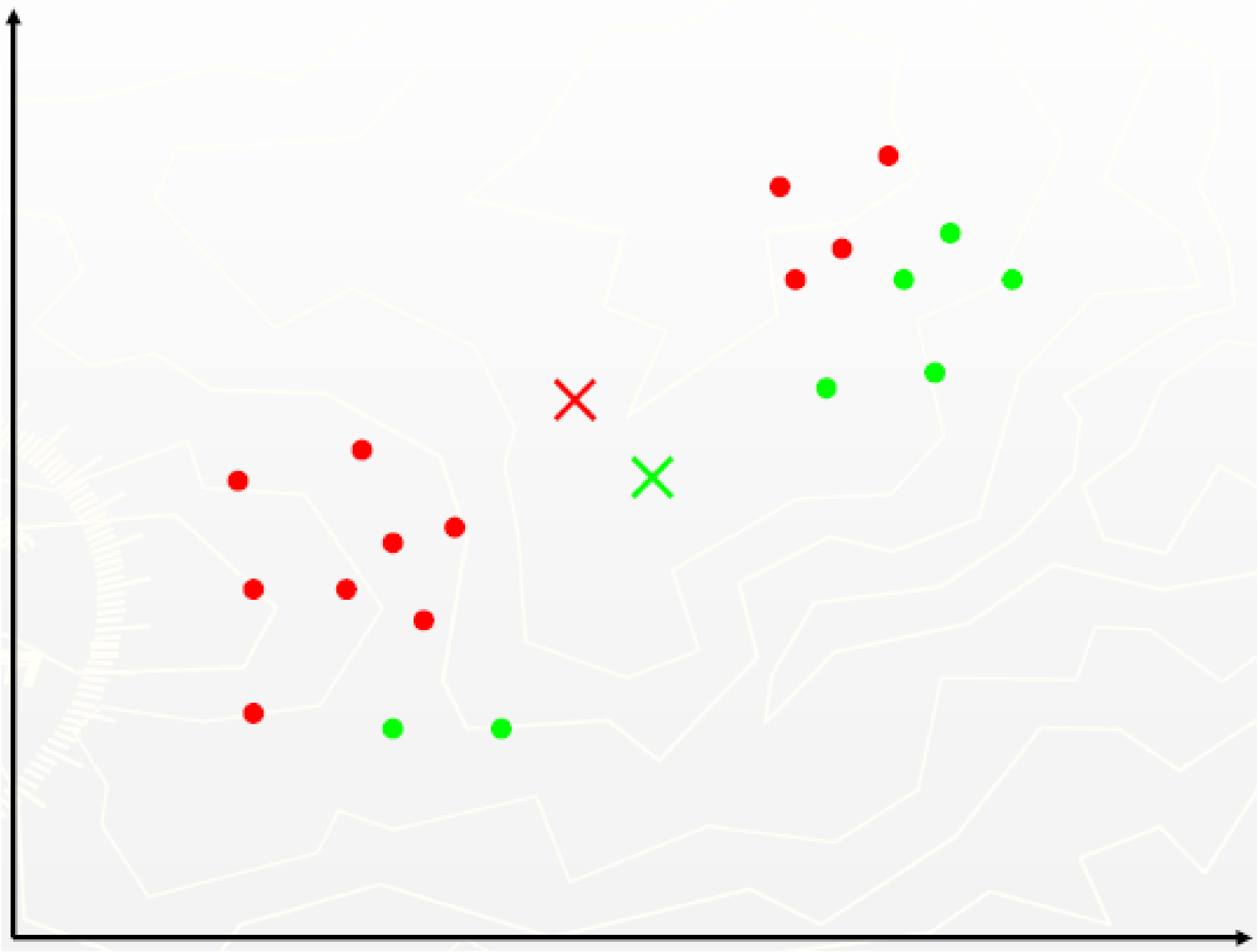
k-means Clustering



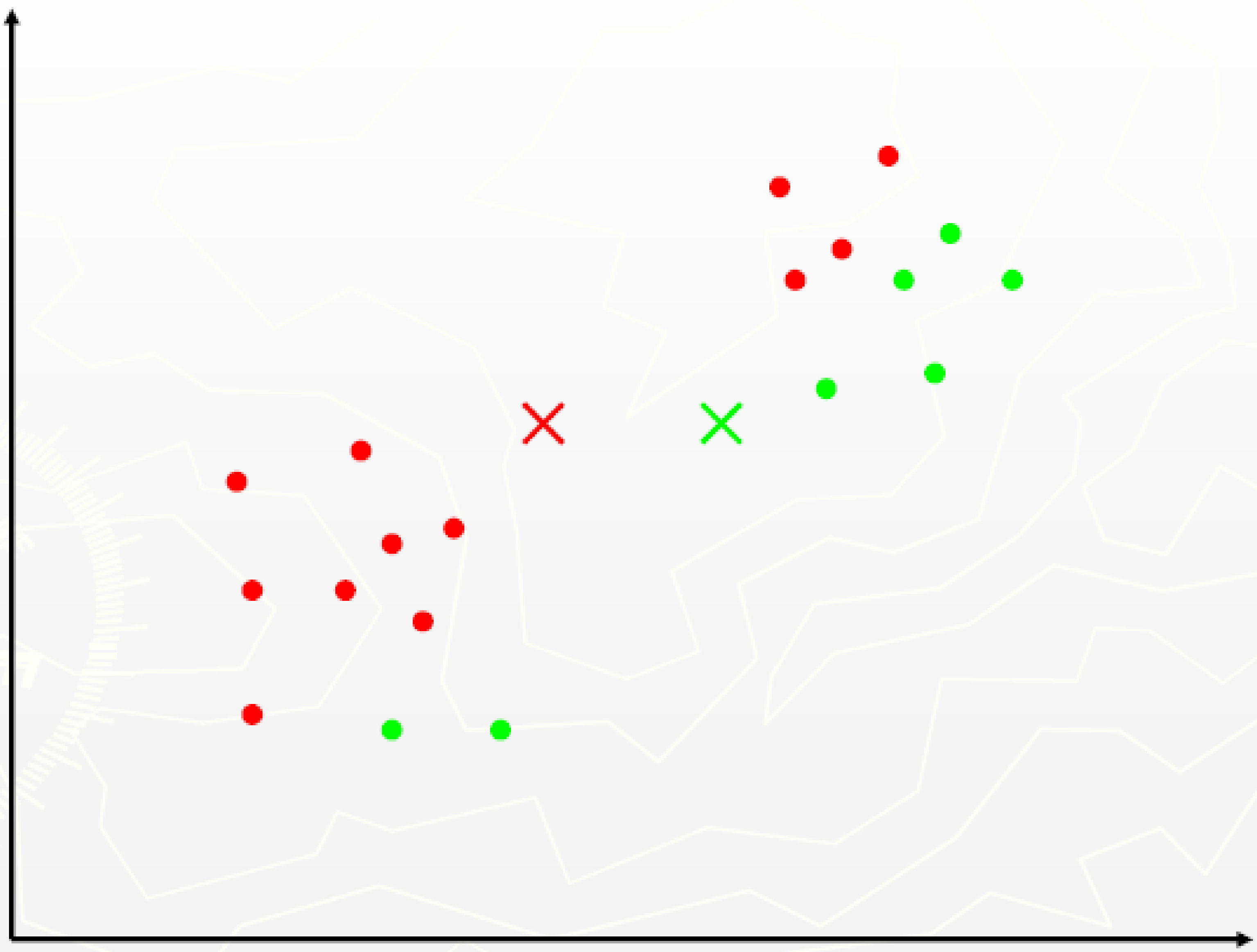
k-means Clustering



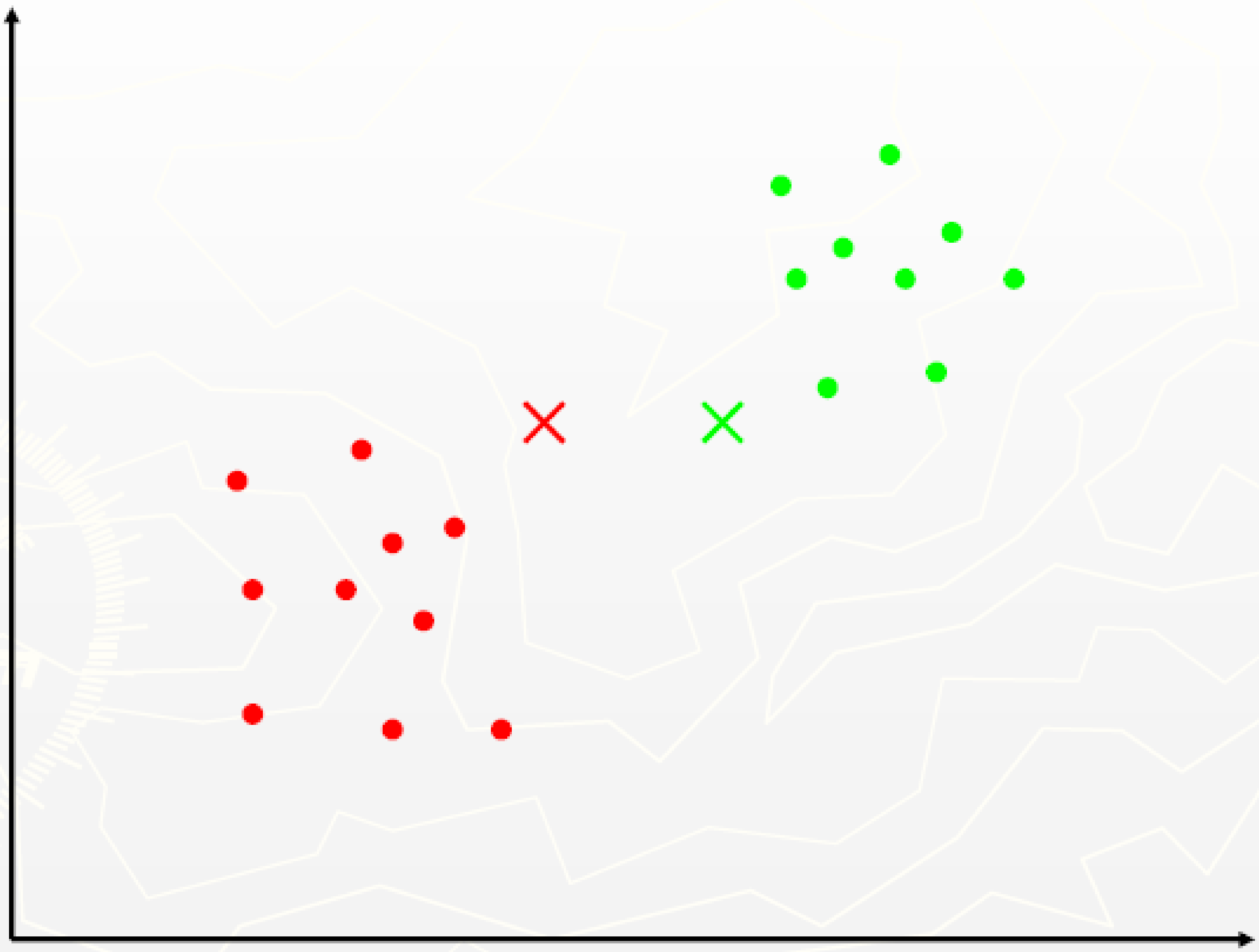
k-means Clustering



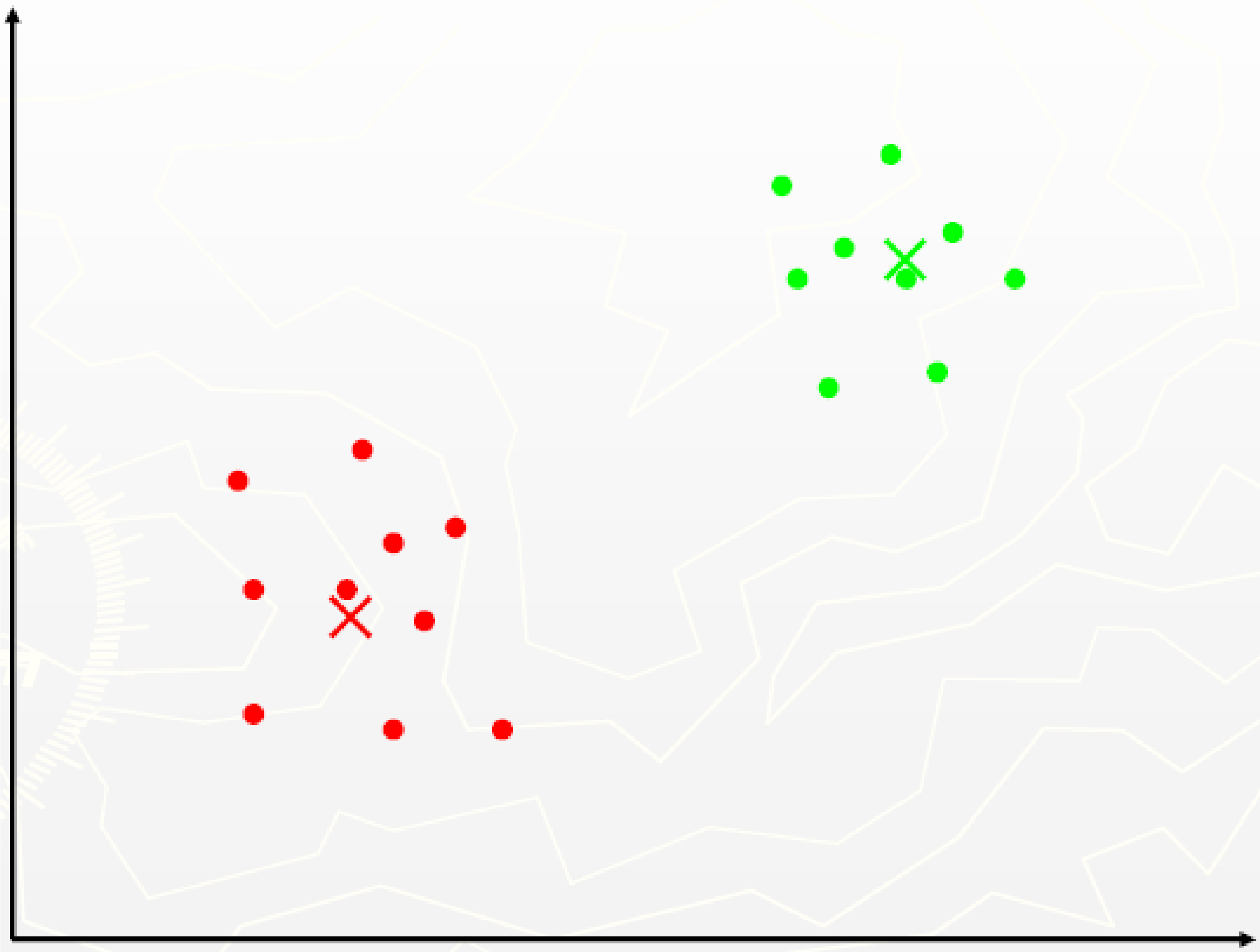
k-means Clustering



k-means Clustering



k-means Clustering



k-means Clustering - Example