EEE-6512: Image Processing and Computer Vision

October 25, 2018

Lecture #9: Segmentation II

Damon L. Woodard, Ph.D.
Dept. of Electrical and Computer Engineering
dwoodard@ece.ufl.edu

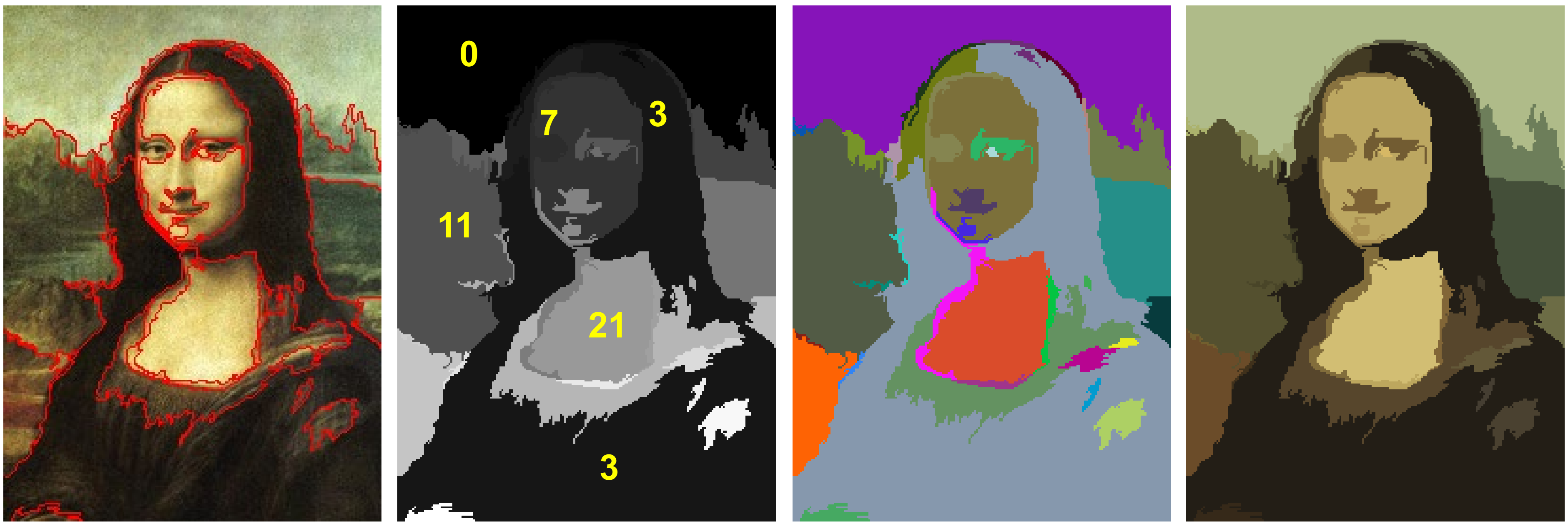
Outline

- Segmentation Background
- Thresholding
- Segmentation by Region Growing and by Region Splitting and Merging
- Clustering with K-Means Algorithm

Background

What is segmentation?

- Segmentation divides an image into groups of pixels
- Pixels are grouped because they share some local property (gray level, color, texture, motion, etc.)



boundaries labels pseudocolors mean colors

(different ways of displaying the output)

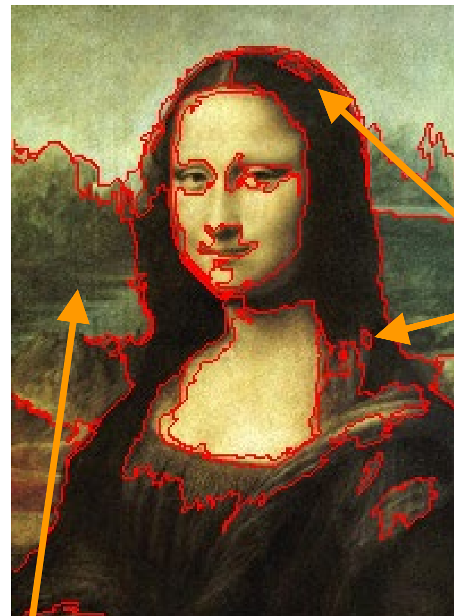
algorithm used: Pedro F. Felzenszwalb and Daniel P. Huttenlocher, Efficient Graph-Based Image Segmentation, IJCV, 59(2), 2004

Segmentation examples

*

* Pictures from Mean Shift: A Robust Approach toward Feature Space Analysis, by D. Comaniciu and P. Meer http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Two errors



oversegmentation (hair should be one group)

undersegmentation (water should be separated from trees)

Simplest Segmentation Problem

Foreground/Background Segmentation

Goal: Separate foreground objects from background.

Common approaches involve finding a **threshold value** which produces a image in which foreground pixels are separated from background pixels.

What we have discussed so far

- **Point and Edge Detection**

- **Main Ideas:**
  - Points represent foreground objects
  - Edges represent boundaries between background and foreground.

Thresholding

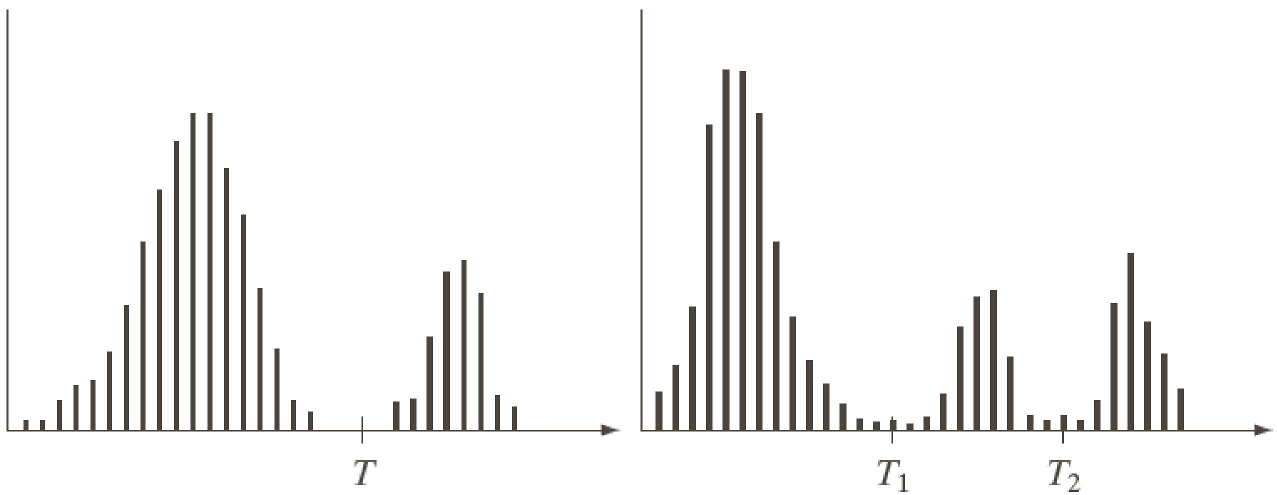
Thresholding

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

T : global thresholding

Multiple thresholding

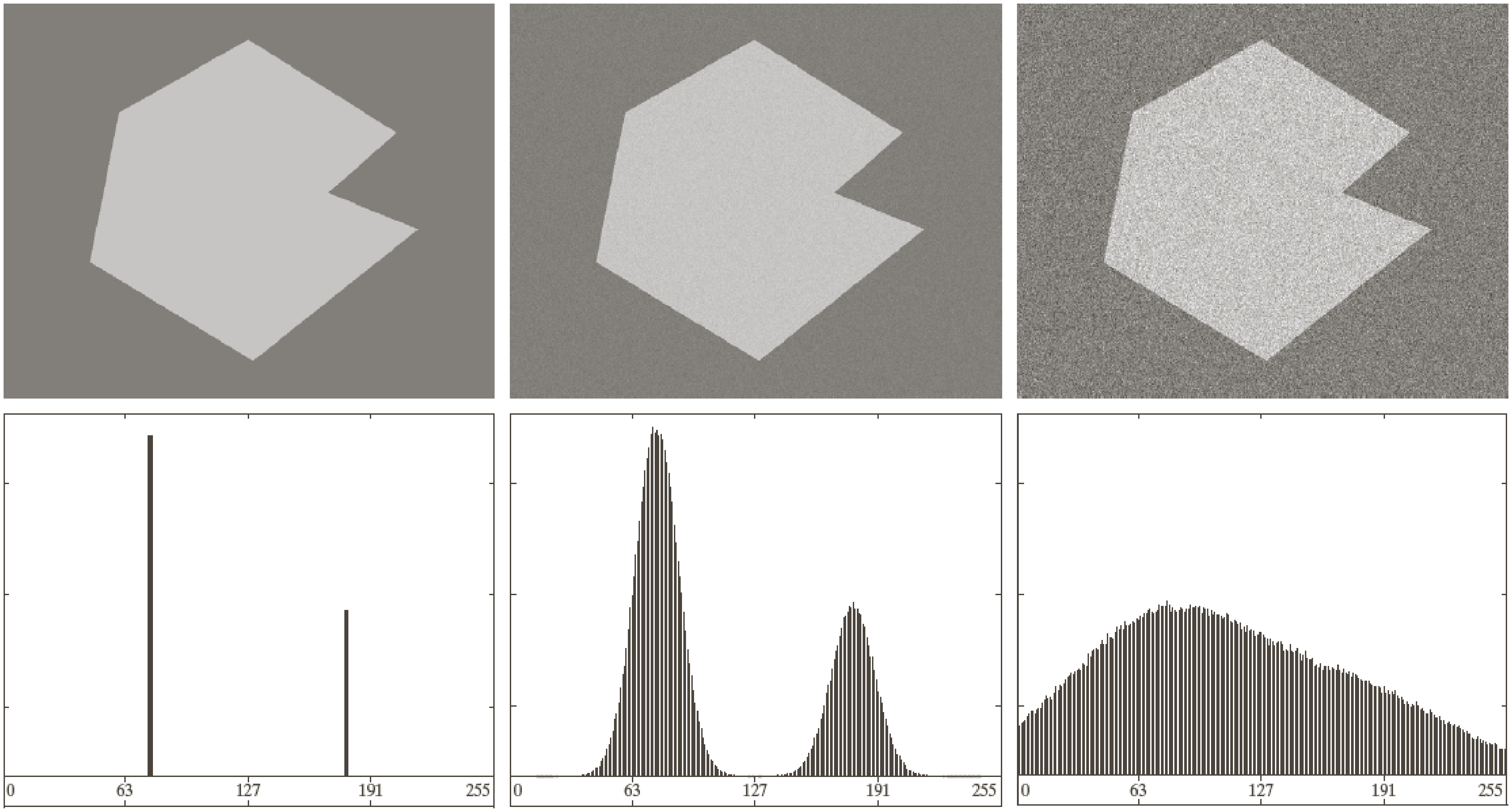
$$g(x,y)=\begin{cases}a & \text{if } f(x,y)>T_2\\ b & \text{if } T_1< f(x,y)\le T_2\\ c & \text{if } f(x,y)\le T_1\end{cases}$$



a b

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



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 and Reflectance

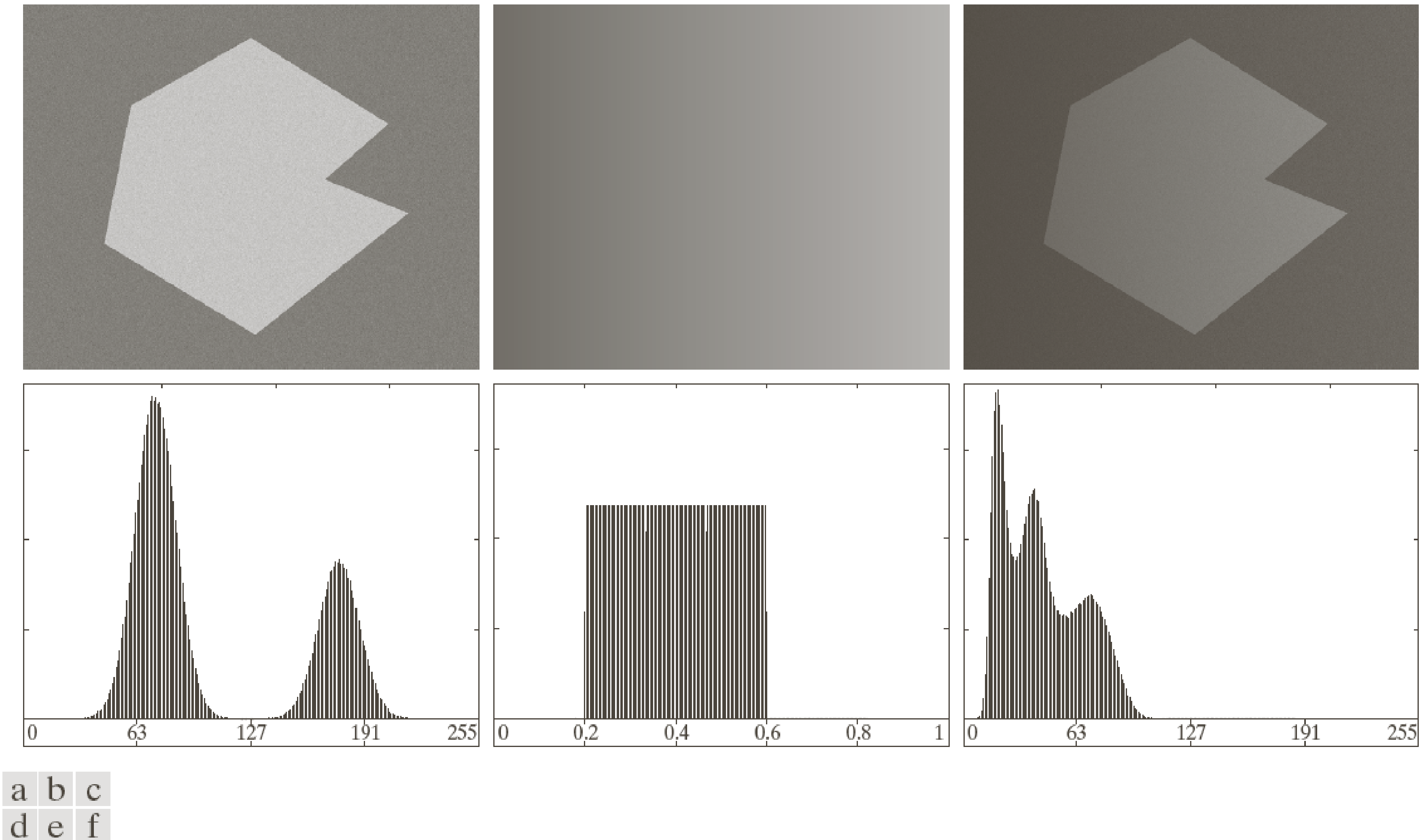


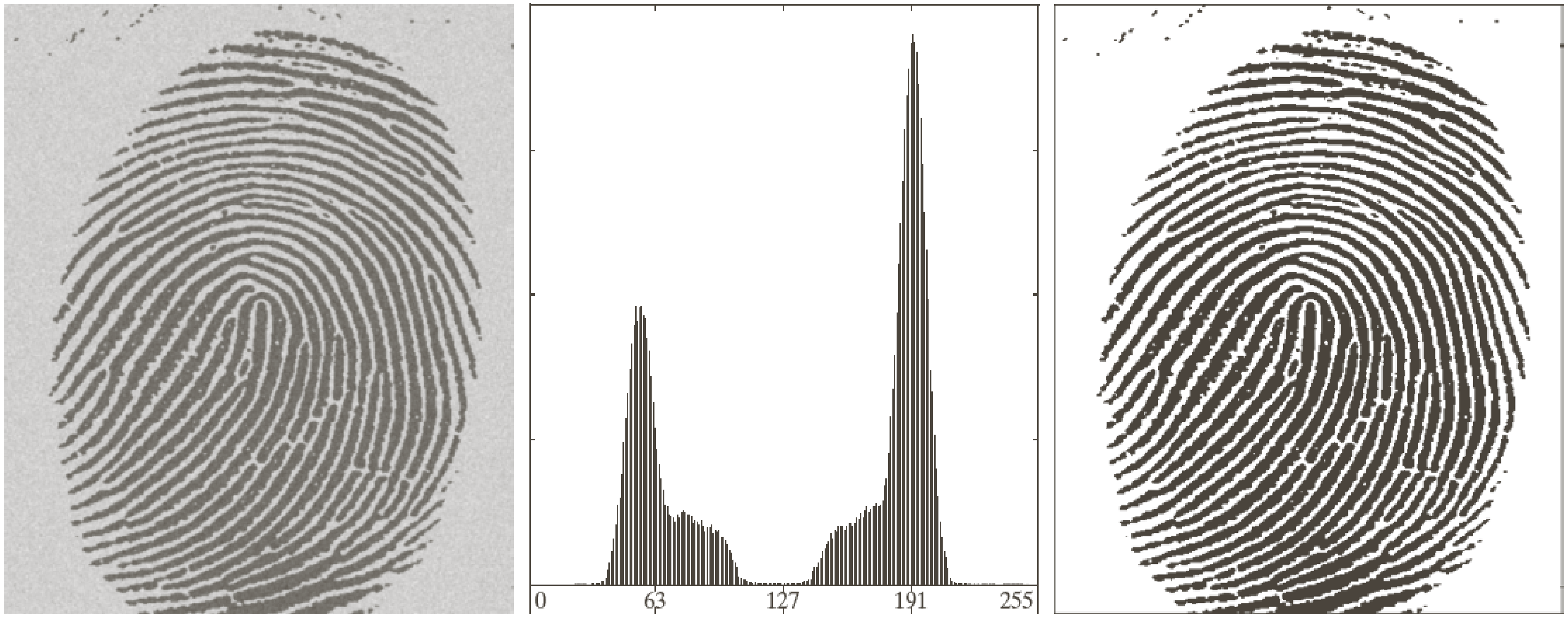
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

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.)

- Algorithm works well with there is a reasonably clear valley between the modes of the histogram related to objects and background.
- Converges in a finite number of steps if the initial choice of T is greater than the minimum but less than the maximum intensity value.

Optimum Global Thresholding Using Otsu's Method

Optimum Global Thresholding Using Otsu's Method

- Basic Idea
  - Threshold giving the best separation between classes in terms of their intensity values would be the best (optimum) threshold.
  - All operations are performed directly on the histogram.
  - Assumes that the histogram (and the image) is bimodal
  - Assumes stationary statistics
  - Assumes uniform illumination (implicitly), so that bimodal brightness behavior arises from object appearance only.
  - Performs an exhaustive search of to determine optimal threshold.

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, $C_1 \rightarrow [0, k],\ C_2 \rightarrow [k+1, L\text{-}1]$

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

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

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

$$= \frac{\left[m_G P_1 - m_1 P_1\right]^2}{P_1(1 - P_1)}$$

$$= \frac{\left[m_G P_1 - m\right]^2}{P_1(1 - P_1)}$$

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

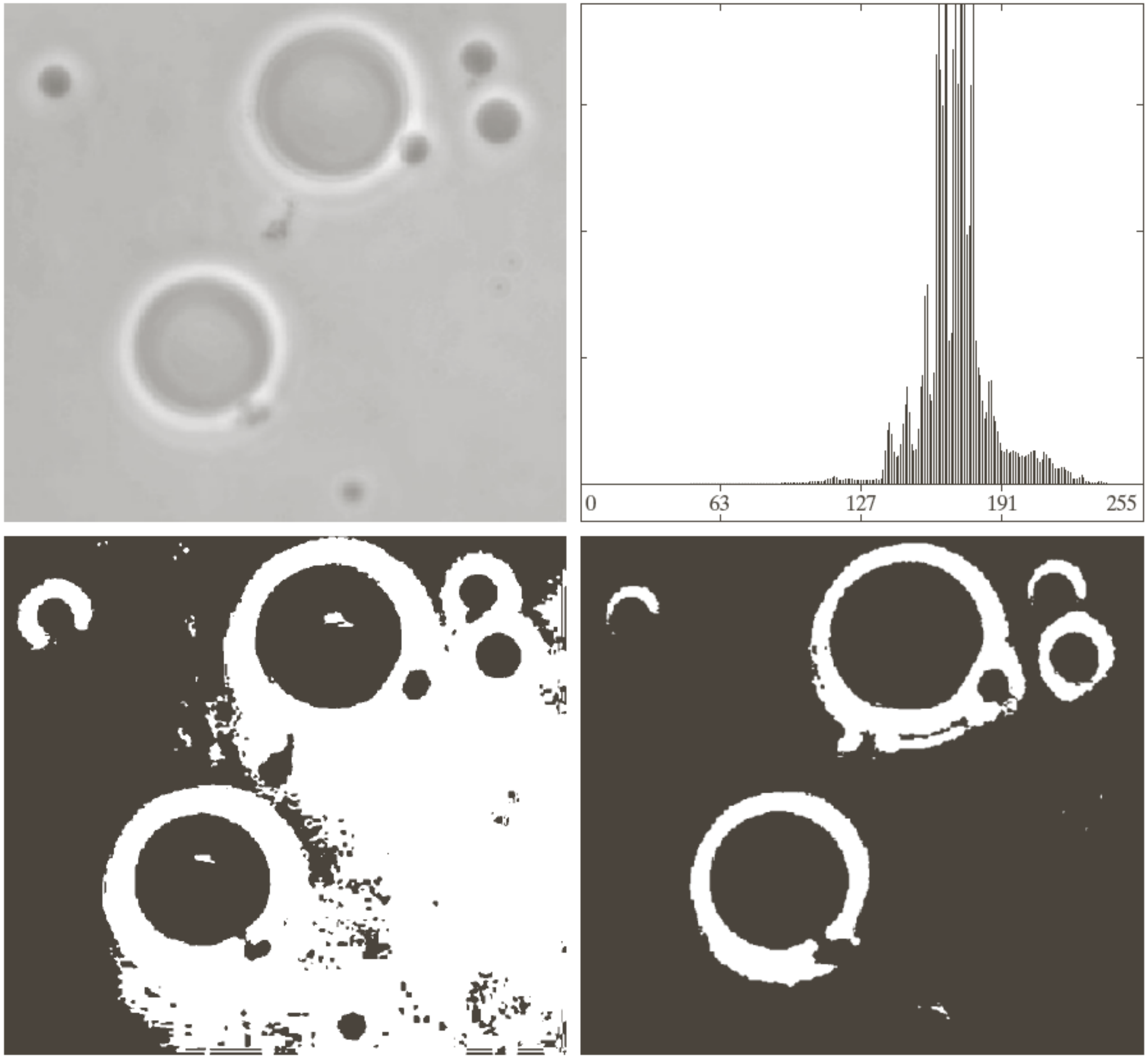
Otsu’s Algorithm: Summary

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.



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

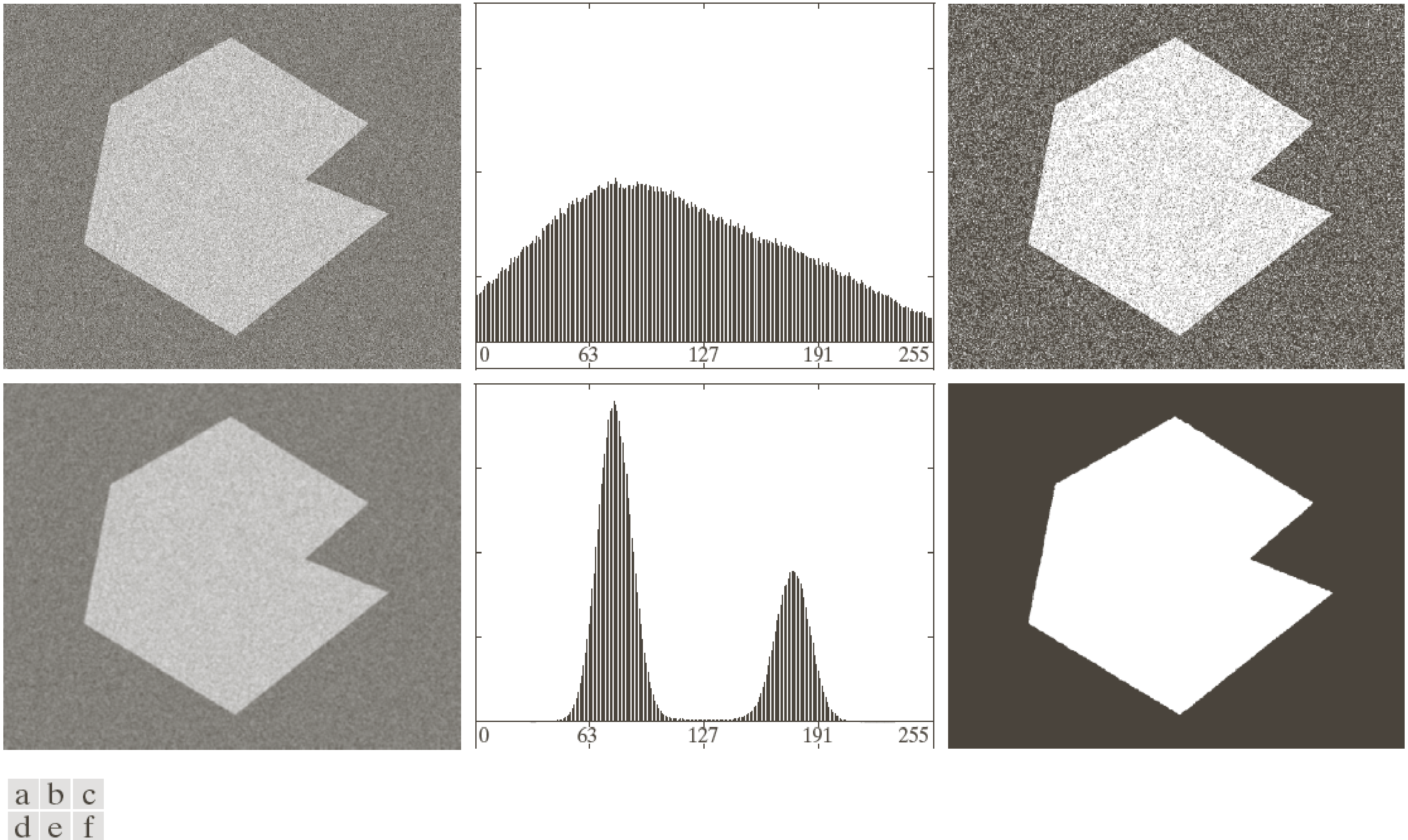


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Otsu’s Method

What happens if the size of the foreground is significantly reduced in the image?

What can be done to address this situation?

Using Edges to Improve Global Thresholding

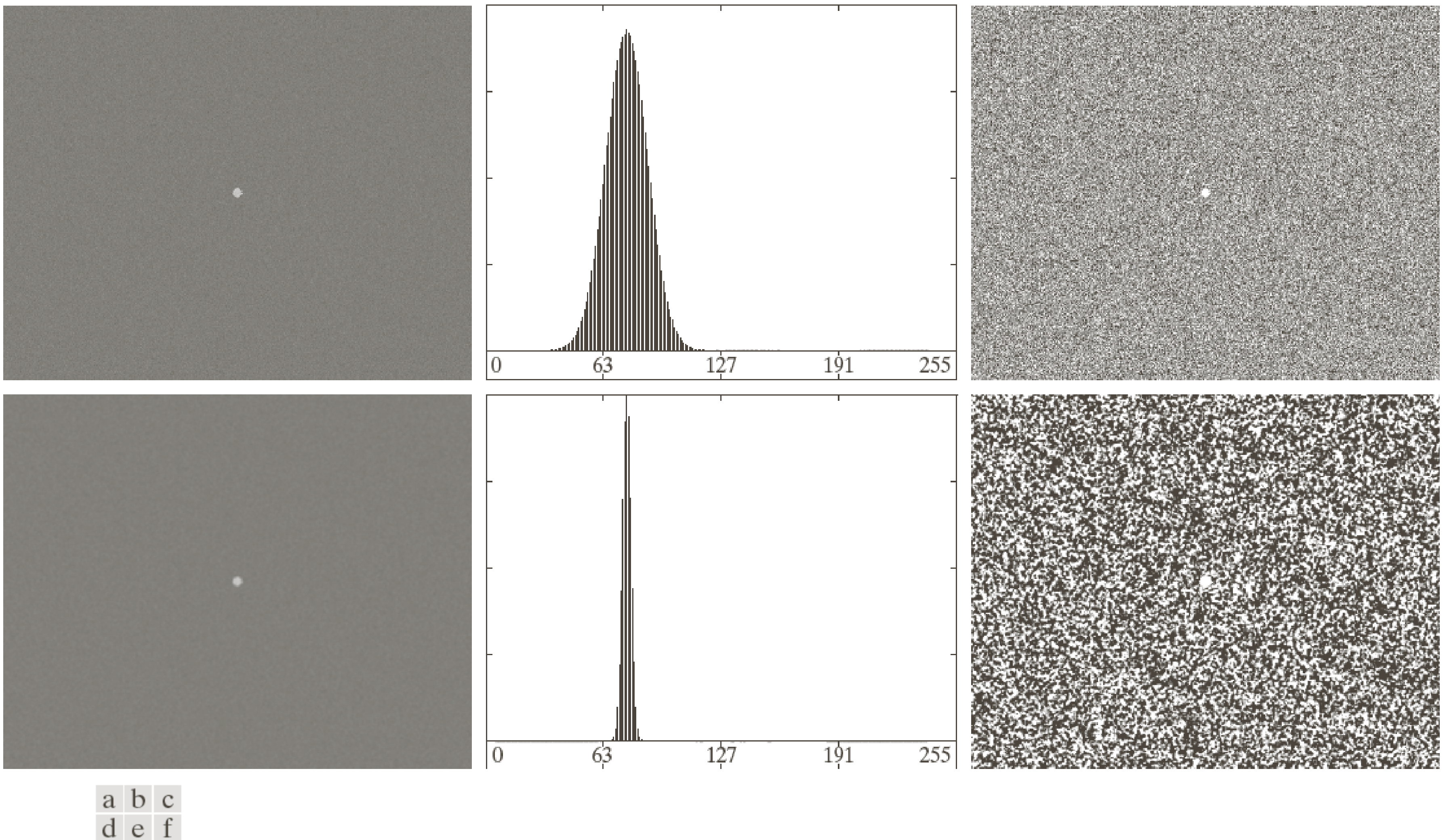


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

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

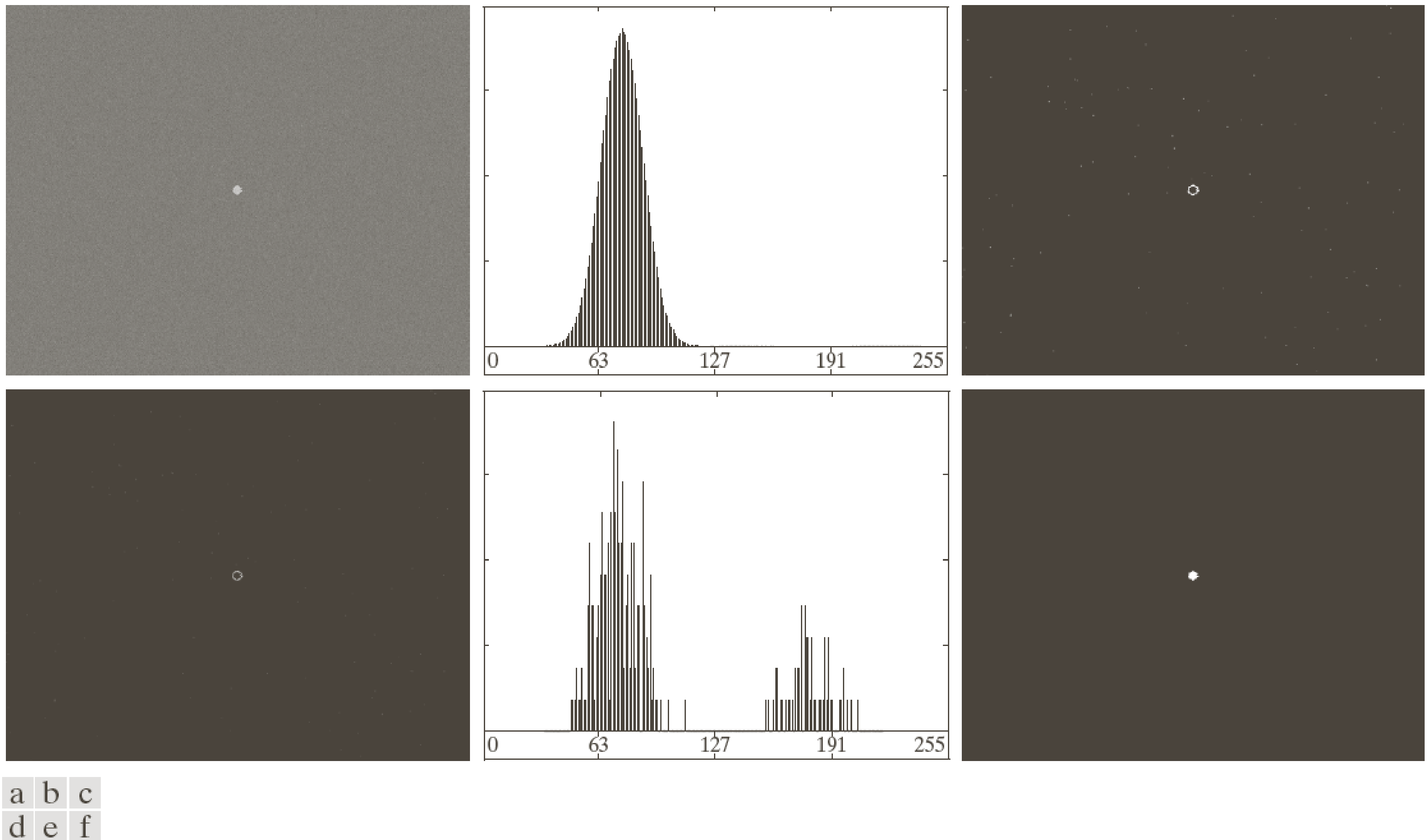


FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based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

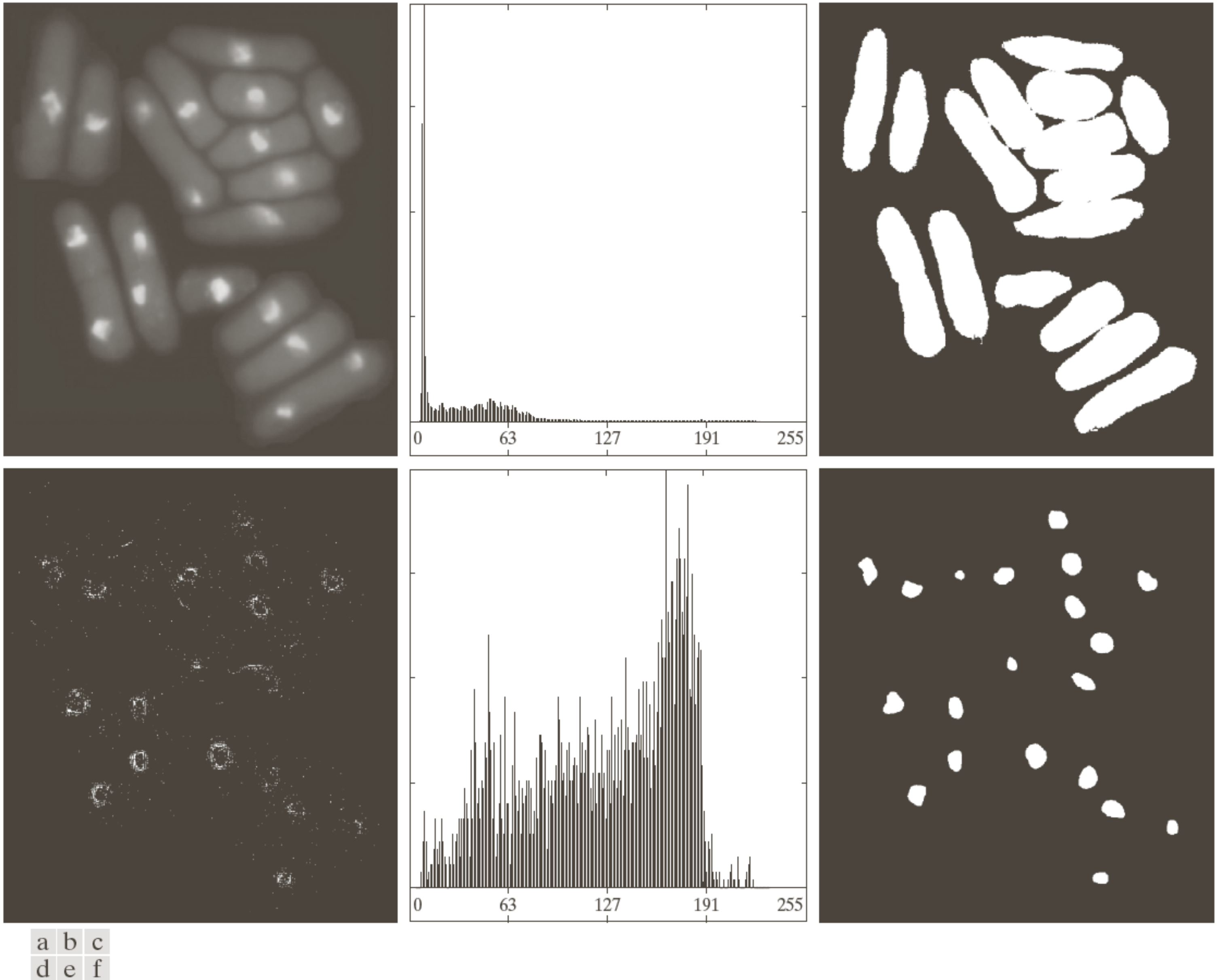


FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram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

Multiple Thresholds

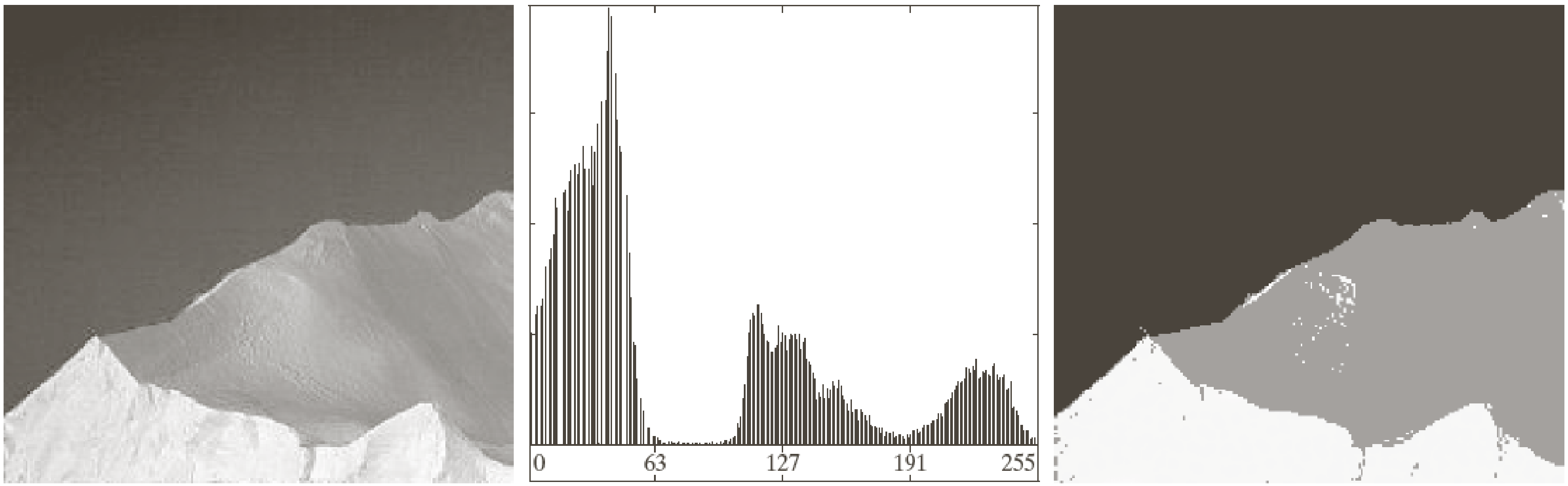
In the case of K classes, $C_1, C_2, ..., C_K$, the between-class variance is

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

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

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

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



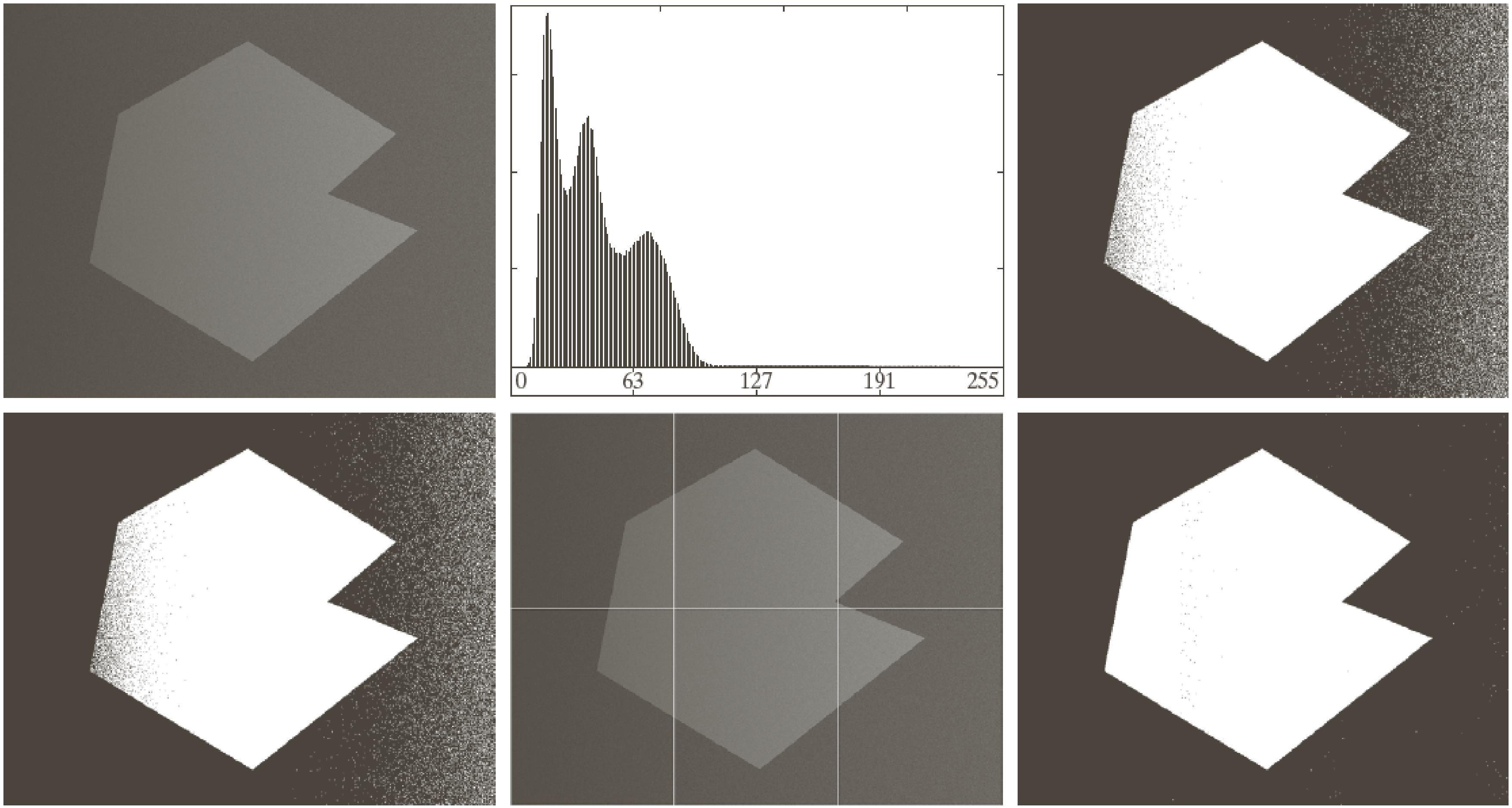
a b c

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

What about non-uniform illumination?

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.

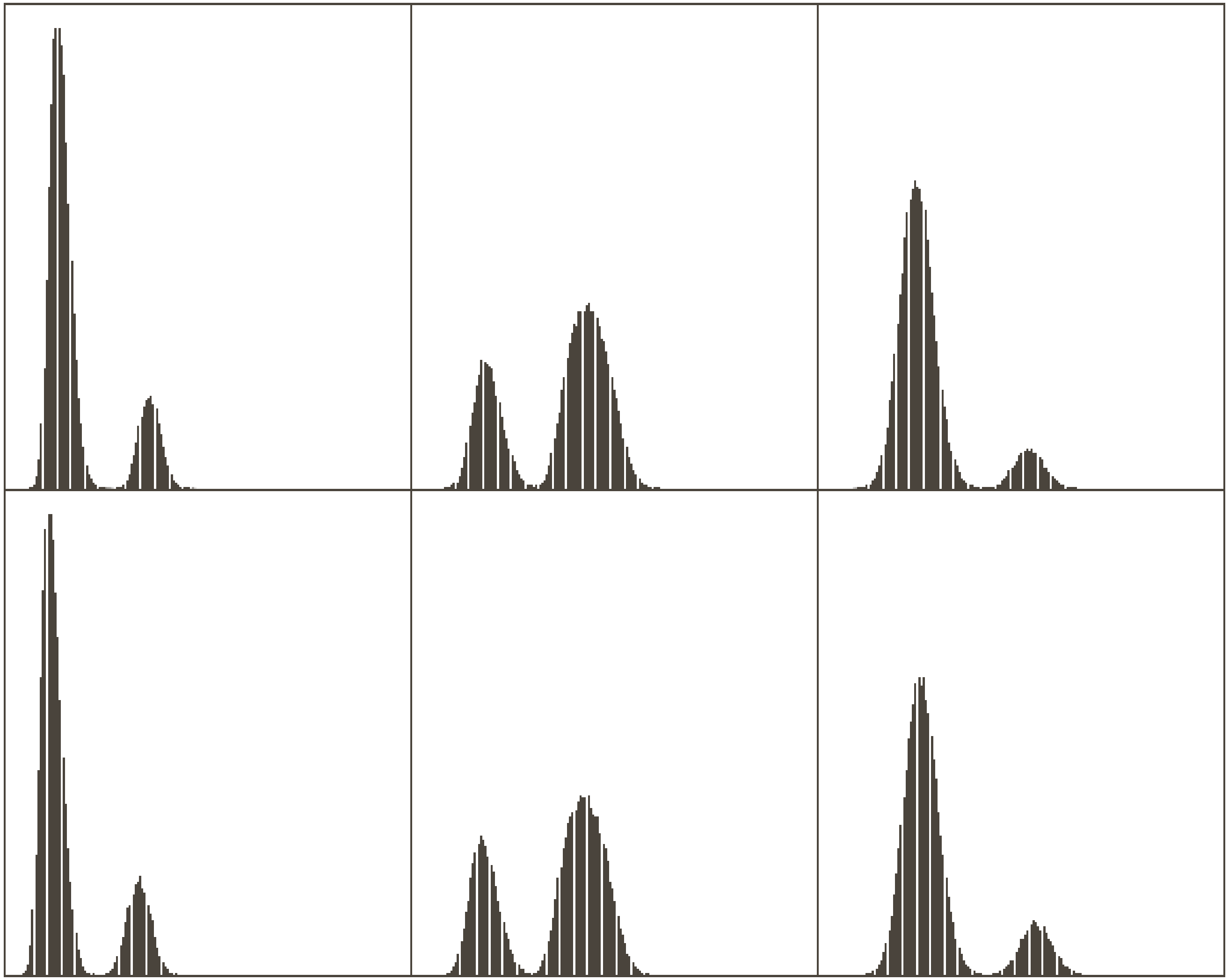


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

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,

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

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

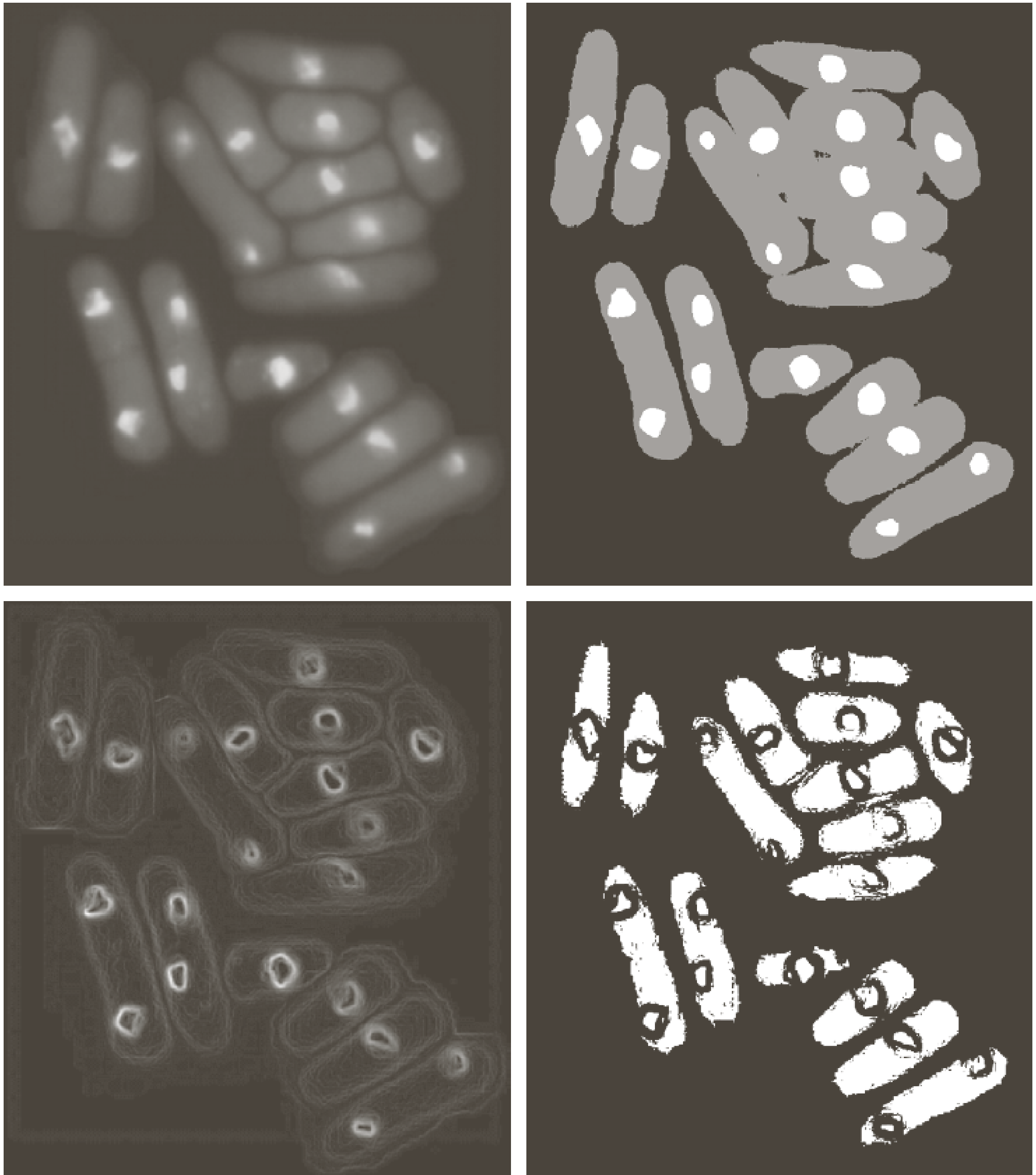
Variable Thresholding Based on Local Image Properties

A modified thresholding

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

$e.g.,$

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



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

Variable Thresholding Using Moving Averages

Let z_{k+1} denote the intensity of the point encountered in the scanning sequence at step $k+1$. 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.

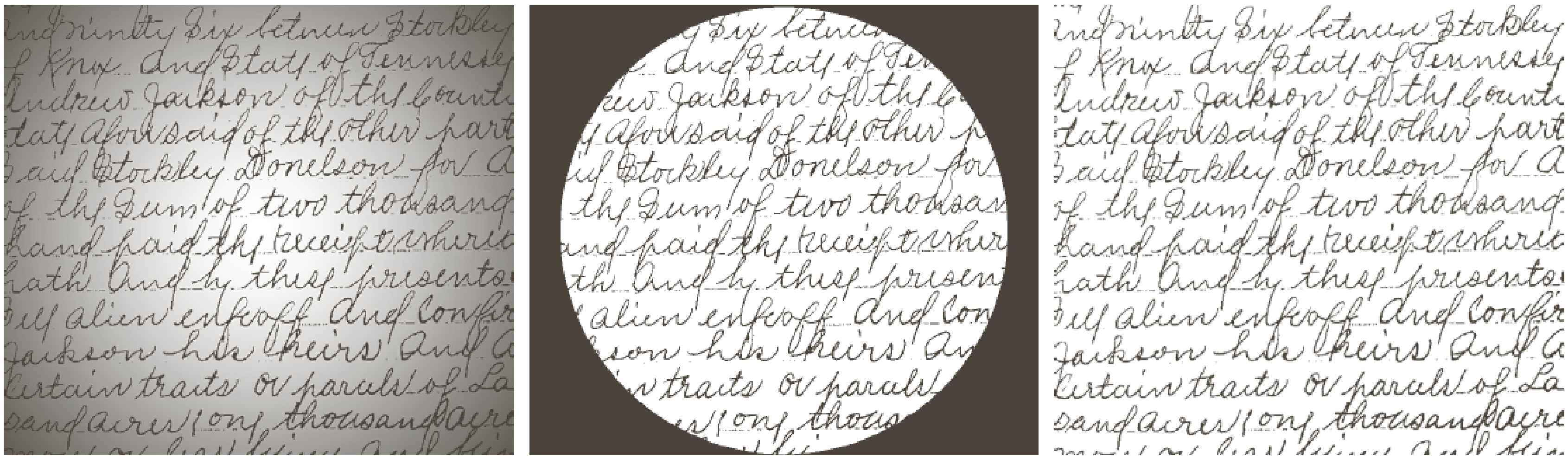
Variable Thresholding Using Moving Averages

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

$$T_{xy} = bm_{xy}$$

m_{xy} is the moving average and b is a positive constant.

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.

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.

Segmentation by Region Growing and by Region Splitting and Merging

Region Growing

Region growing: each pixel is initialized as a separate region, and adjacent regions that look similar are successively merged.

For a grayscale image, the mean and variance are given by:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} I(\mathbf{p}_i)$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (I(\mathbf{p}_i) - \mu)^2$$

Region Growing (cont'd)

The dissimilarity of a pixel **q** is measured by its distance from the mean, relative to the standard deviation:

$$d(\mathbf{q}; I, \mu, \sigma) = |I(\mathbf{q}) - \mu|/\sigma$$

Figure 10.23 Left: An RGB image. Right: Regions found by region growing, pseudocolored.

P. Chockalingam, N. Pradeep, and S.T. Birchfield, Adaptive fragments-based tracking of non-rigid objects using level sets. In Proceedings of the International Conference on Computer Vision, Oct. 2009.

Splitting and Merging

Let us define the segmentation of an image I as a set of regions $R_1, \ldots, R_n$ such that every pixel is in exactly one region:

$$I = \bigcup_{i=1}^{n} \mathcal{R}_i \qquad \text{(covers entire image)}$$

$$\mathcal{R}_i \cap \mathcal{R}_j = \emptyset \quad \text{for all} \quad i \neq j \qquad \text{(non-overlapping)}$$

In the classic view of the problem of segmentation, a predicate $h(R_i)$ measures the **homogeneity** of a region.

Splitting and Merging (cont'd)

Merging algorithms begin with each pixel as a separate region, then recursively merge adjacent regions whenever they are similar to each other.

Splitting algorithms begin with the entire image as a single region, then recursively split regions whenever they are found to be nonhomogeneous.

$$h(\mathcal{R}_i) = \text{TRUE for all } i \qquad \text{(each region is homogeneous)}$$

$$h(\mathcal{R}_i \cup \mathcal{R}_j) = \text{FALSE for all adjacent } \mathcal{R}_i, \mathcal{R}_j, i \neq j \qquad \text{(adjacent regions are different)}$$

Splitting and Merging (cont'd)

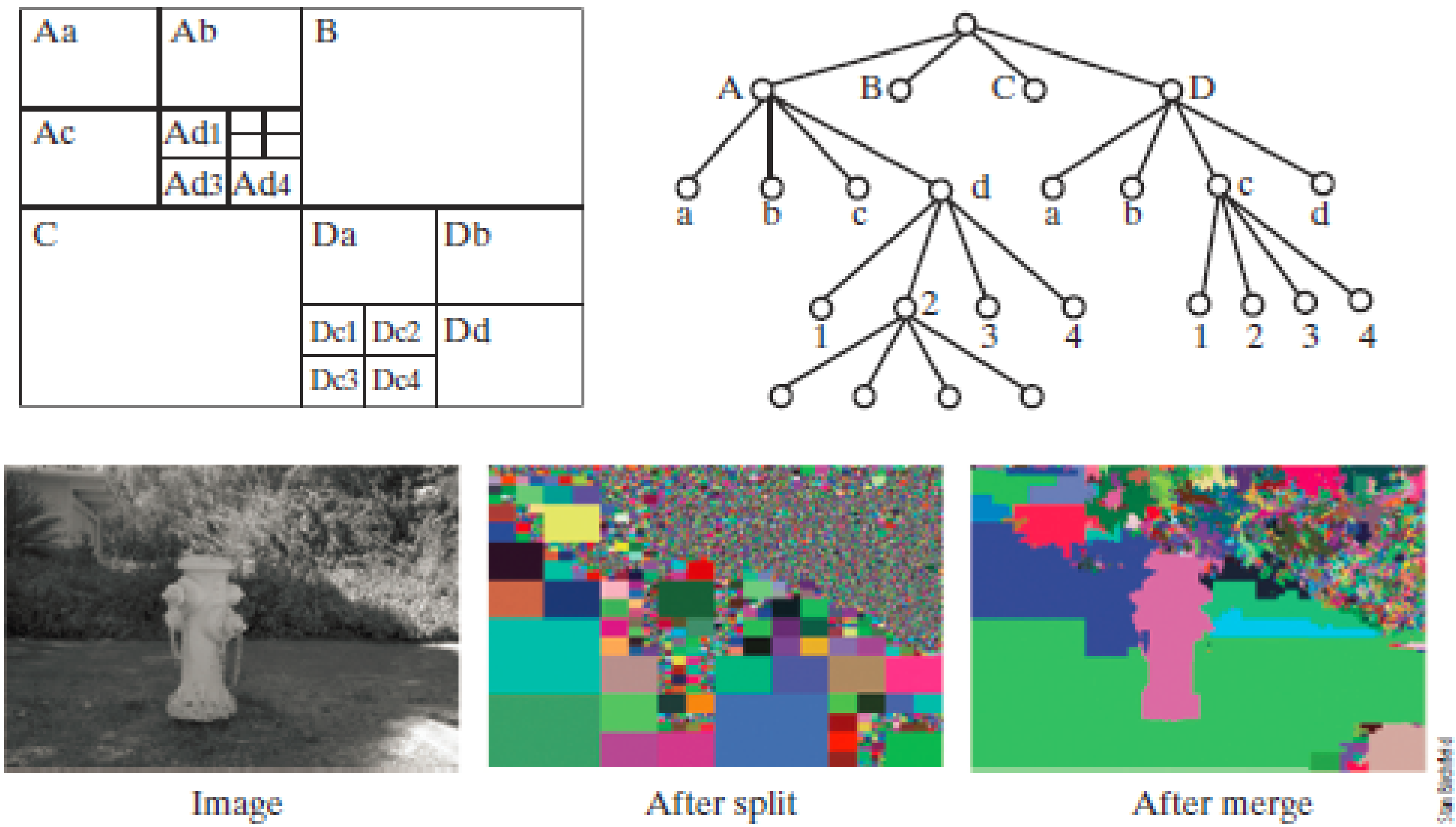


Figure 10.22 Top: The quad-tree data structure used in splitting. Bottom: The split-and-merge algorithm applied to a grayscale image. The algorithm is able to find the fire hydrant and most of the ground, though it oversegments the textured background.

Clustering

Clustering

- There are K clusters $C_1, \ldots, C_K$ with means $m_1, \ldots, m_K$.
- The **least-squares error** is defined as

$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} || x_i - m_k ||^2 \quad .$$

- Out of all possible partitions into K clusters, choose the one that minimizes D.

K-Means Clustering

Form K-means clusters from a set of n-dimensional vectors

1. Set ic (iteration count) to 1

2. Choose randomly a set of K means $m_1(1), \ldots, m_K(1)$.

3. For each vector x_i compute $D(x_i, m_k(ic))$, $k=1,\ldots K$ and assign x_i to the cluster C_j with nearest mean.

4. Increment ic by 1, update the means to get $m_1(ic),\ldots,m_K(ic)$.

5. Repeat steps 3 and 4 until $C_k(ic) = C_k(ic+1)$ for all k.

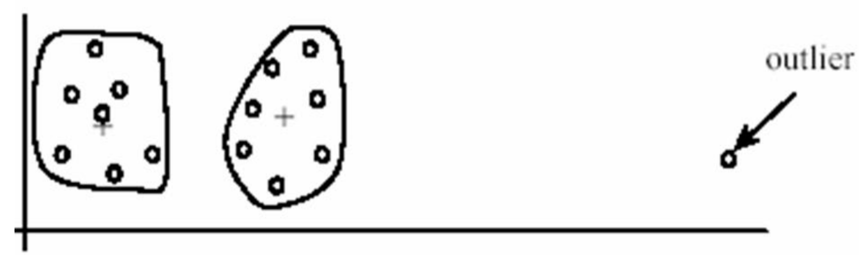
K-means Variants

- Different ways to initialize the means
- Different stopping criteria
- Dynamic methods for determining the right number of clusters (K) for a given image

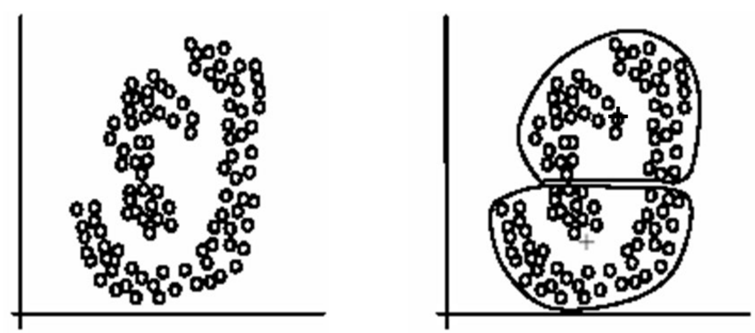
K-Means Algorithm

- Pros
  - Simple and Fast
  - Converges to a local minimum of the error function

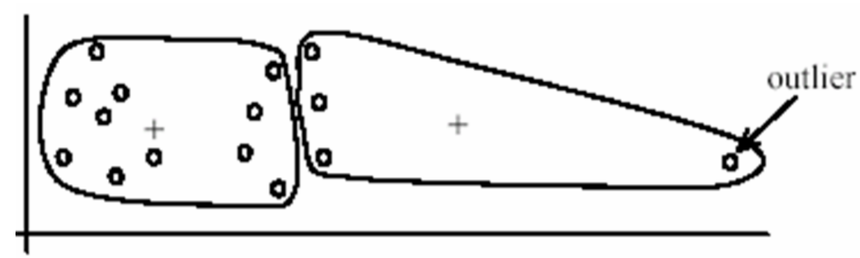
- Cons
  - Need to pick K
  - Sensitive to initialization
  - Only finds "spherical" clusters
  - Sensitive to outliers



(B): Ideal clusters



(A): Two natural clusters (B): *k*-means clusters



Summary

- Segmentation is important problem in image processing and computer vision
- The use of edges to separate foreground from background
- Thresholding methods
  - Global, local, and adaptive methods
- Region Growing and Region Splitting/Merging
- Segmentation my clustering (K-means)

Questions?

Next Topic: Chapter 12 Feature Extraction

Slide Credits

Images taken from Digital Image Processing by Gonzalez and Woods Text.

Material taken from Jen-Chang Liu lecture slides

Material taken from Stanley Birchfield lecture slides