Image Segmentation

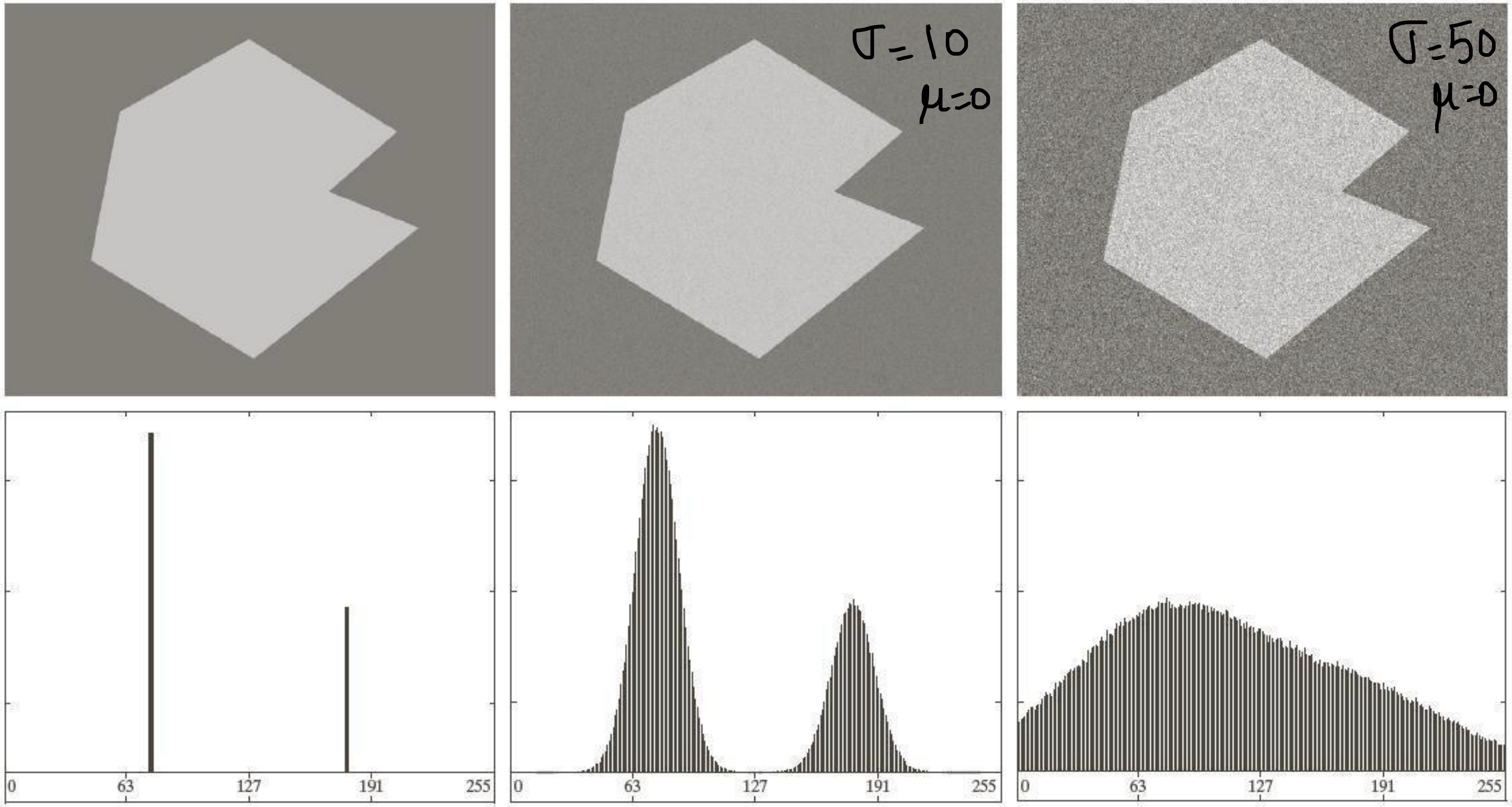
Lecture 4

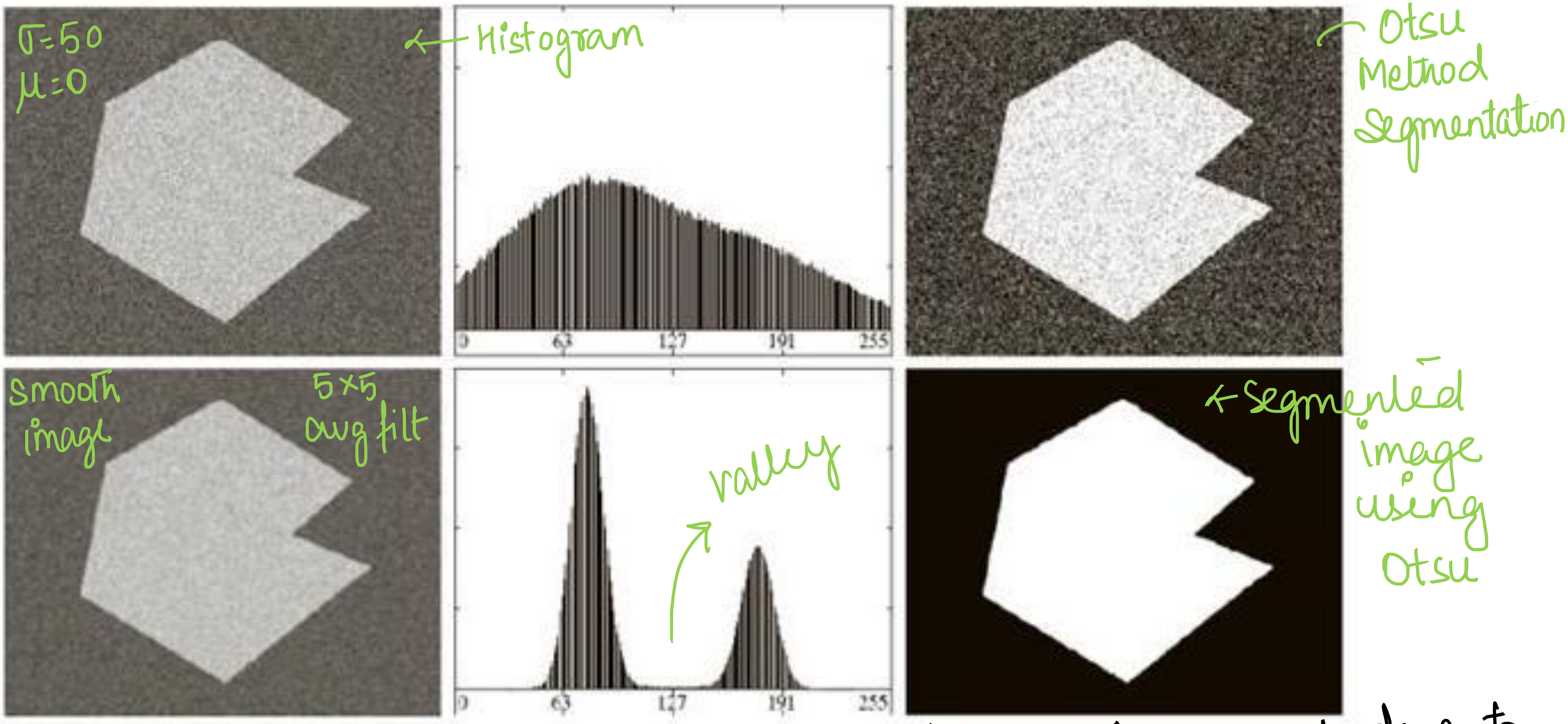
Smoothing to improve Global Thresholding

- Noise creates unsolvable problem in thresholding.
- If noise cannot be removed at source and thresholding is to be done then smoothing can enhance the performance.

no noise image

Histogram





Distortion at edges is caused due to blurring of boundary.

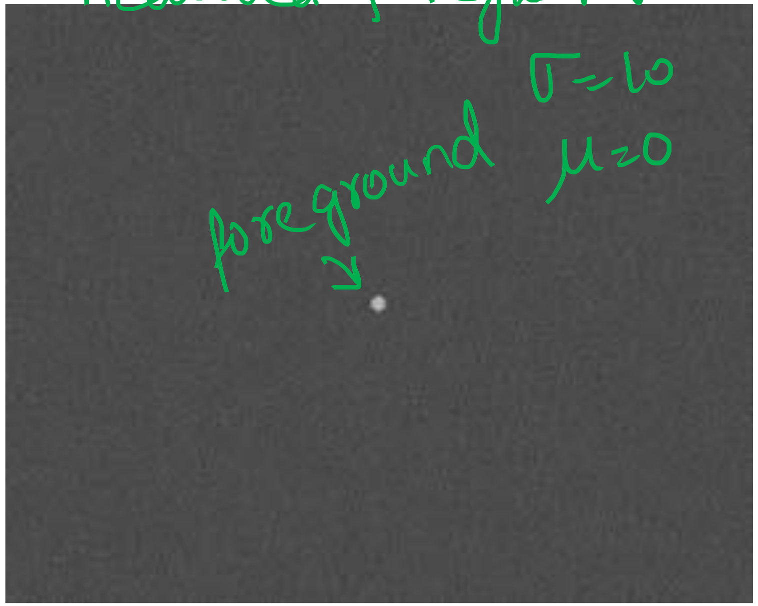
a	b	c
d	e	f

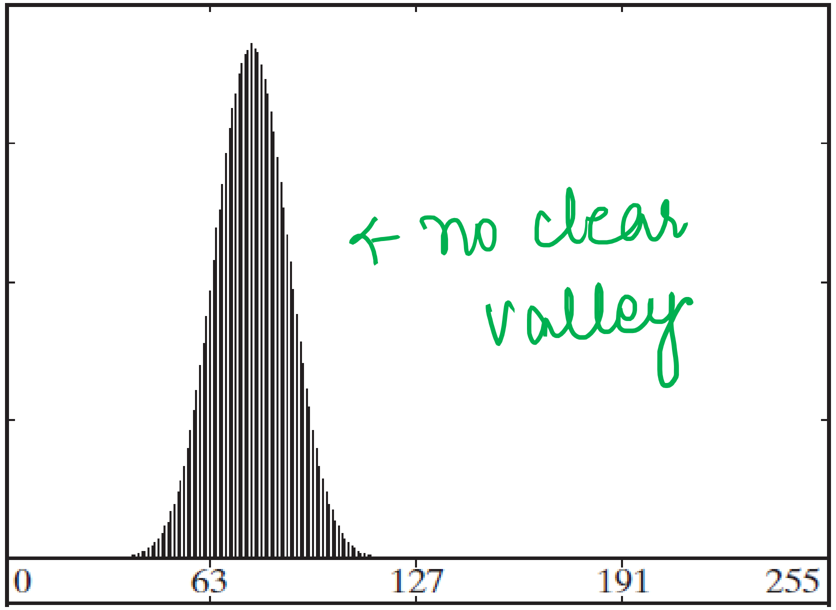
FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using

Effect of Reducing the size of the region:

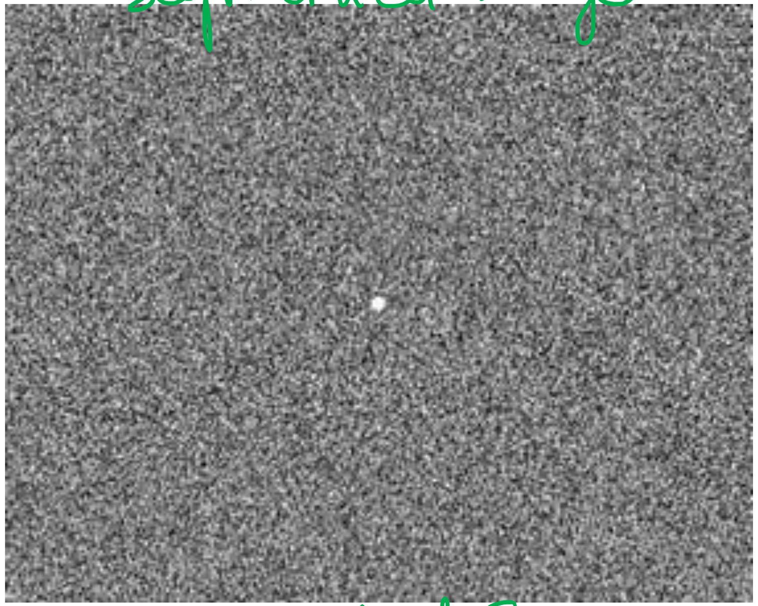
- Reducing relative size of foreground w.r.t background → no clear valley in histogram.
- Smoothing reduces the spread as expected but distribution does not change →not effective.

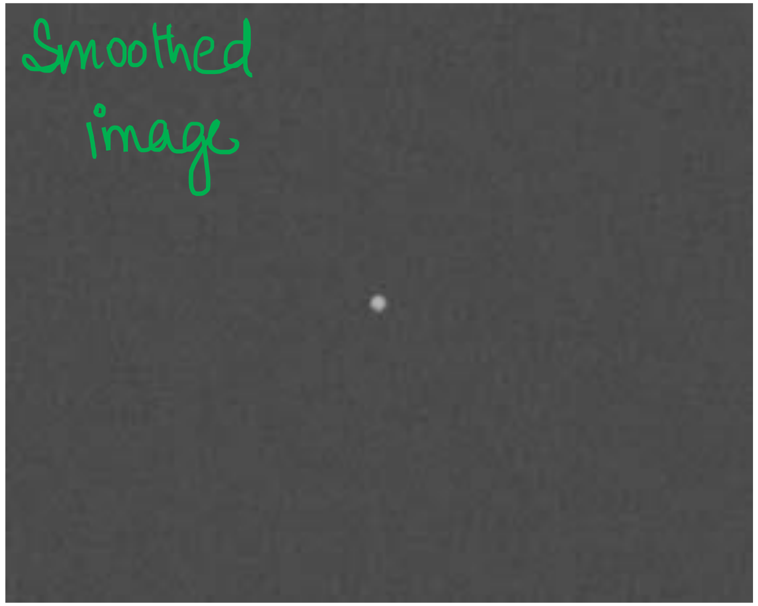
Reduced foreground

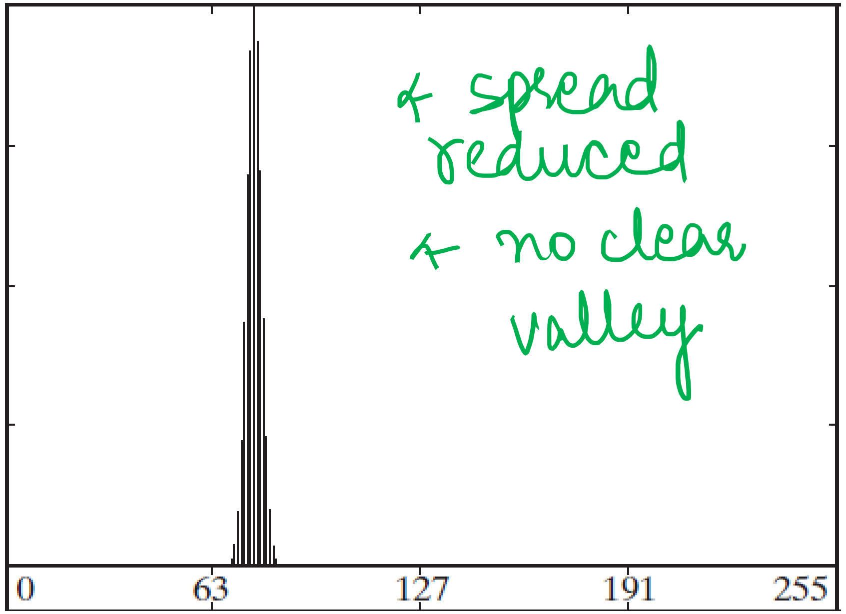




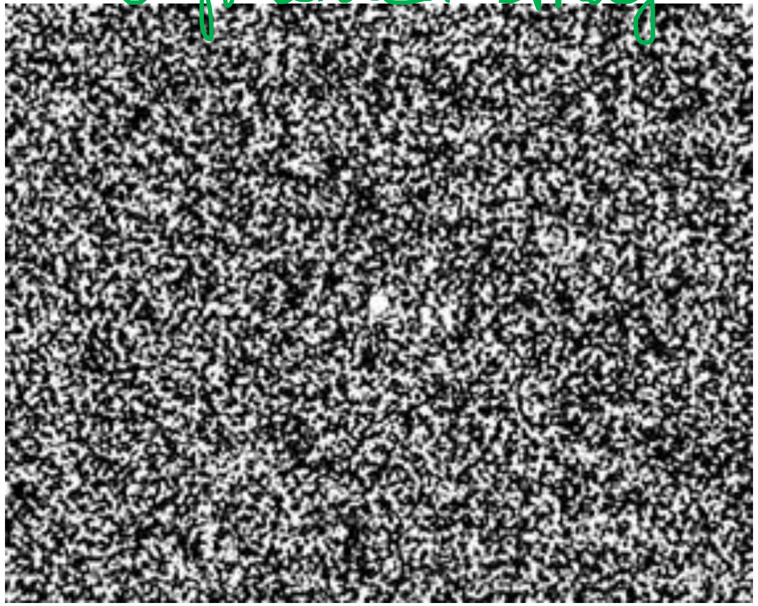
Segmented image







Segmented Image



No proper segmentation

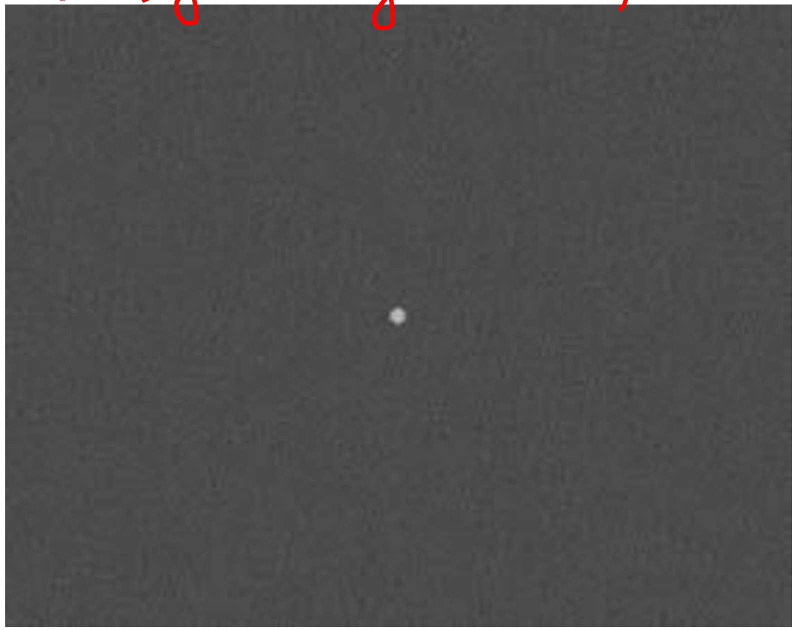
Using Edges to improve thresholding

- Chances of selecting a good threshold are enhanced considerably if the histogram peaks are tall, narrow, symmetric and separated by deep valleys.
- Consider only those pixels which lies on or near to edges between foreground and background.
- Now histogram will be less dependent on sizes of foreground and background.
- Resulting histogram will have peaks of approximately same height
- The pixels that satisfy some measures based on gradient and laplacian has tendency to deepen the valleys in histogram.
- Laplacian is preferred as its computationally more attractive and isotropic.

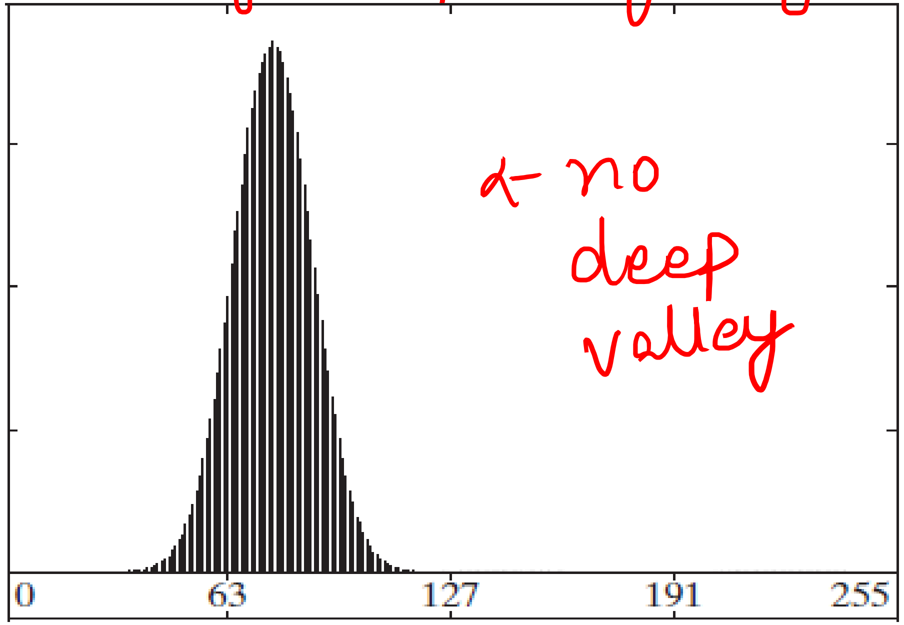
- The preceding discussion is summarized in the following algorithm, where $f(x, y)$ is the input image:

1. Compute an edge image as either the magnitude of the gradient, or absolute value of the Laplacian.
2. Specify a threshold value, T.
3. Threshold the image from Step 1 using the threshold from Step 2 to produce a binary image, $g_T(x, y)$. This image is used as a mask image in the following step to select pixels from $f(x, \mathrm{y})$ corresponding to "strong" edge pixels.
4. Compute a histogram using only the pixels in $f(x, \mathrm{y})$ that correspond to the locations of the 1-valued pixels in $g_T(x, y)$.
5. Use the histogram from Step 4 to segment globally using, for example, Otsu's method.

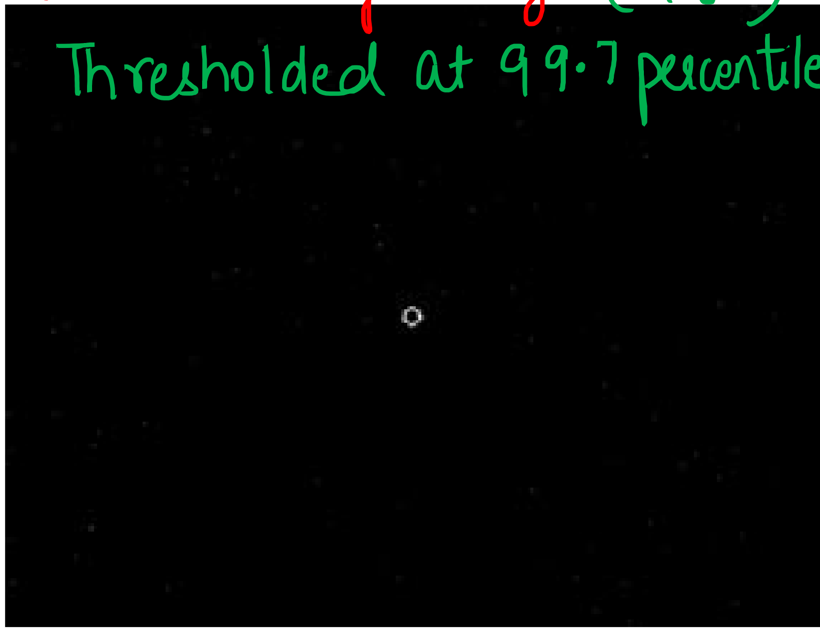
Noisy image $\sigma = 10$ $\mu = 0$

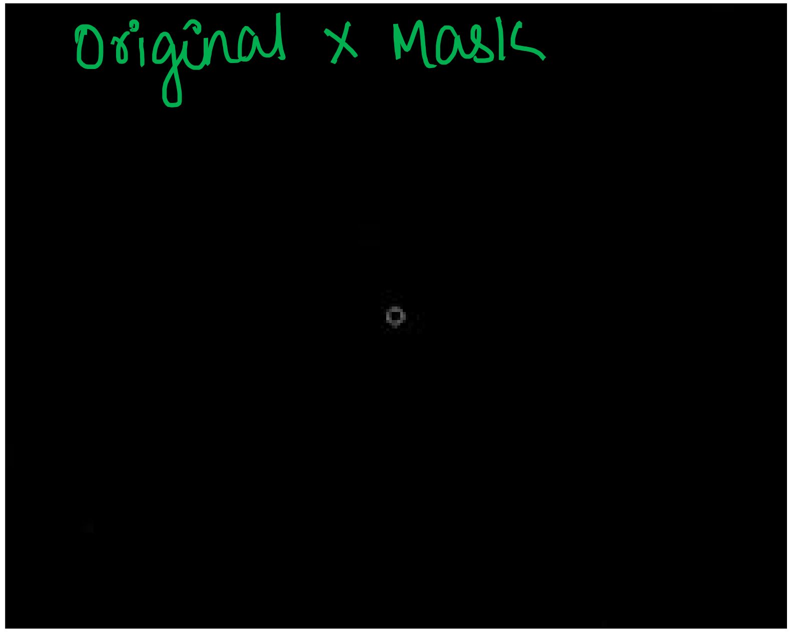


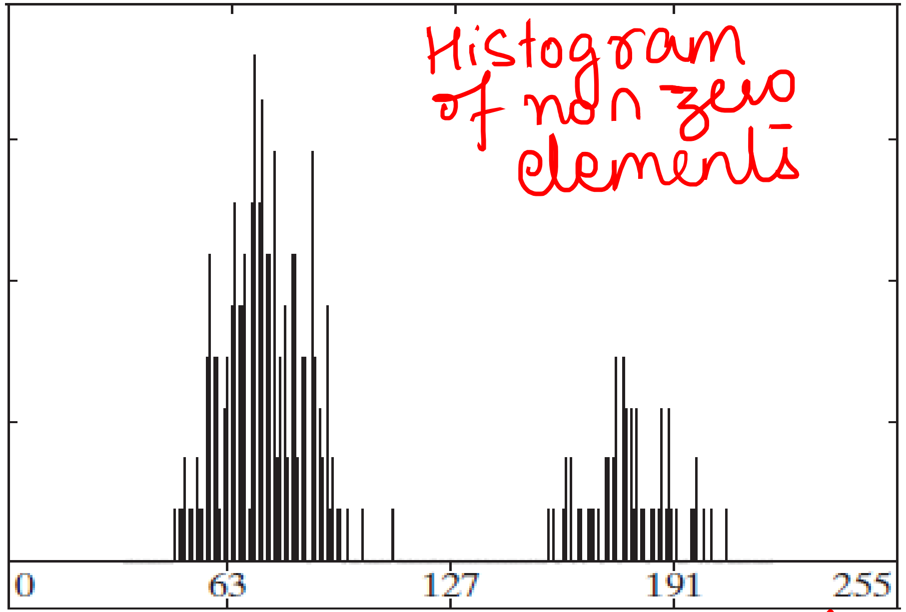
Histogram of noisy image

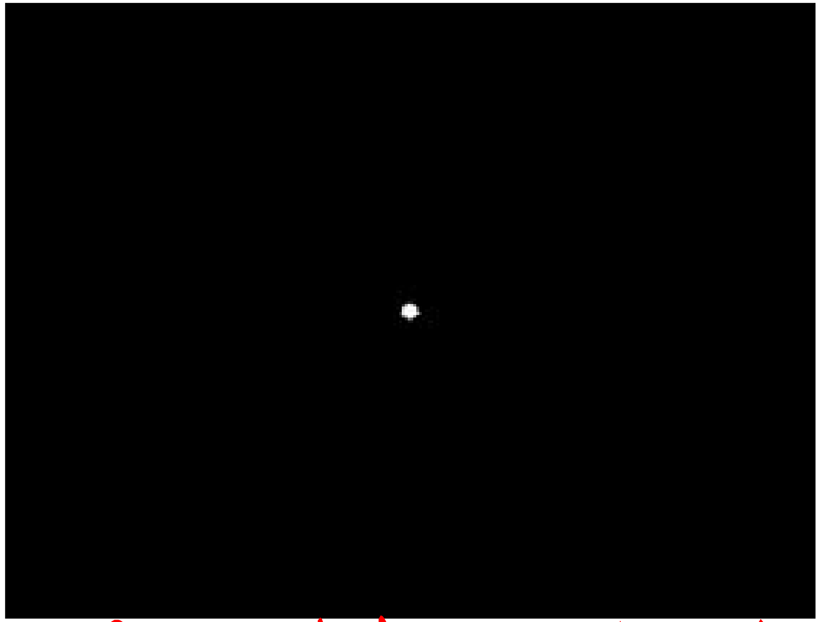


Gradient Mag Image (Mask)

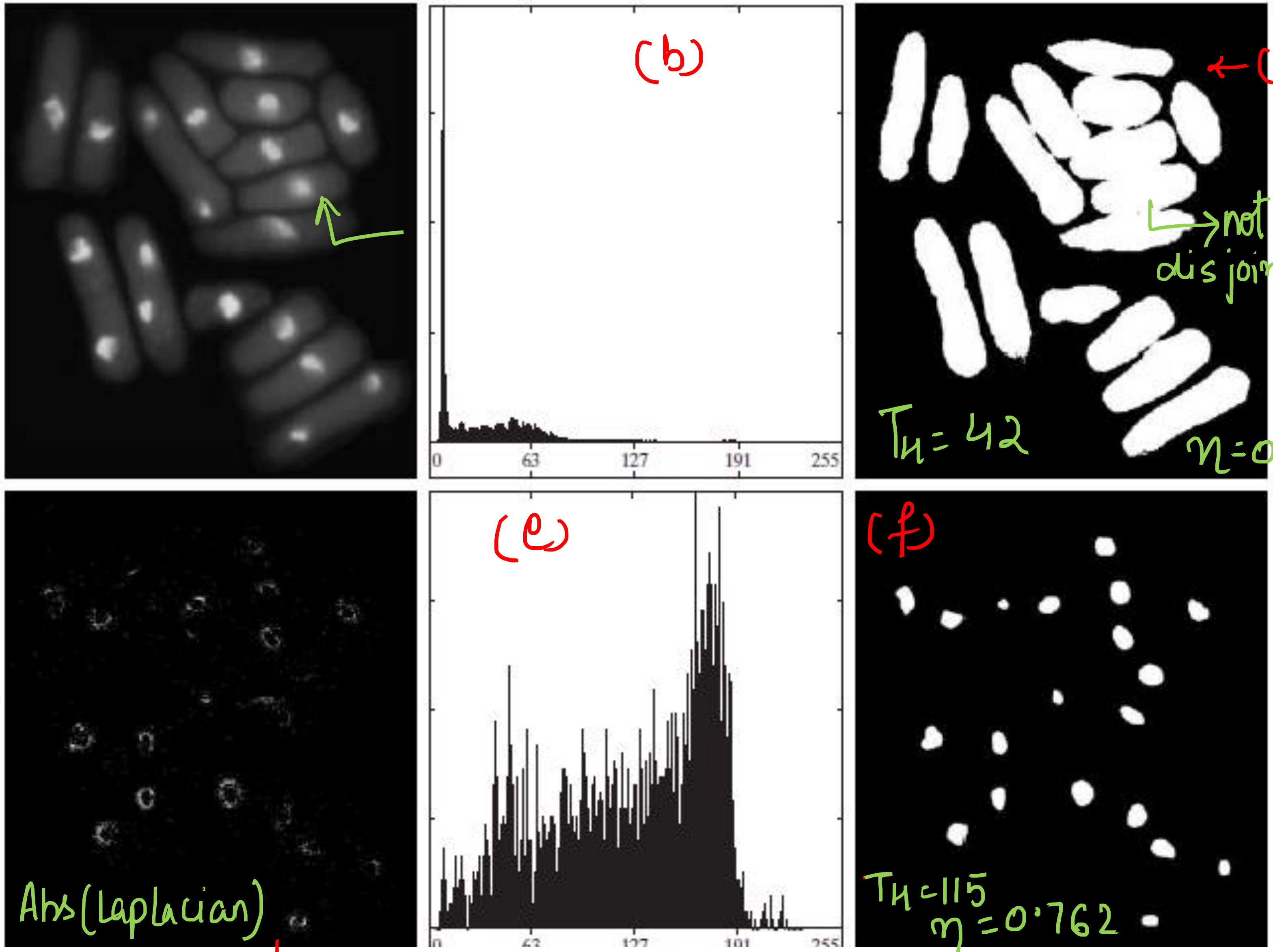








Segmented image (Otsu)



a b c
d e f

→ T = 115 99.5 percentile of fig d

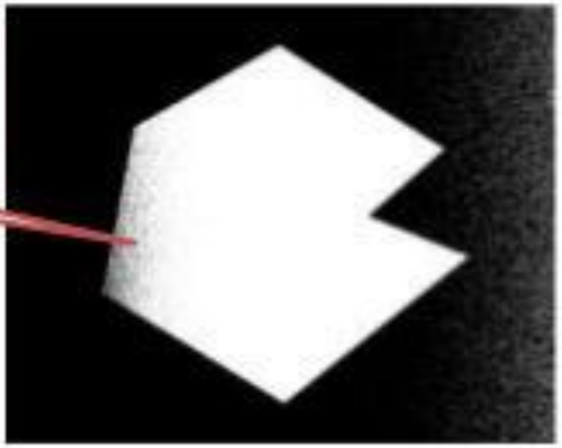
FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram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

FIGURE 10.44 Image in Fig. 10.43(a) segmented using the same procedure as explained in Figs. 10.43(d)–(f), but using a lower value to threshold the absolute Laplacian image.

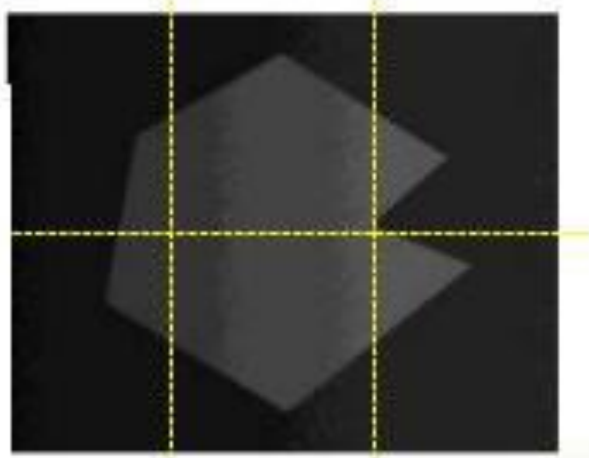
Variable Thresholding

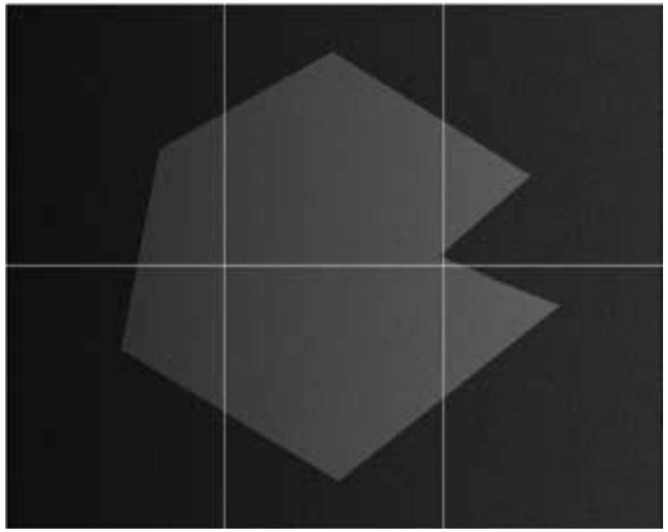
- Factors such as noise and non-uniform illumination play a major role in the performance of a thresholding algorithm.
- Earlier, we saw the use of image smoothing and using edge information.
- However, sometimes these approaches are also ineffective.
- The next level of thresholding complexity involves variable thresholding.

Errors in segmentation



Noisy shaded image	Histogram	Otsu Thresholding
Image sub divide into six sub images		Result of applying Otsu method to each sub image individually





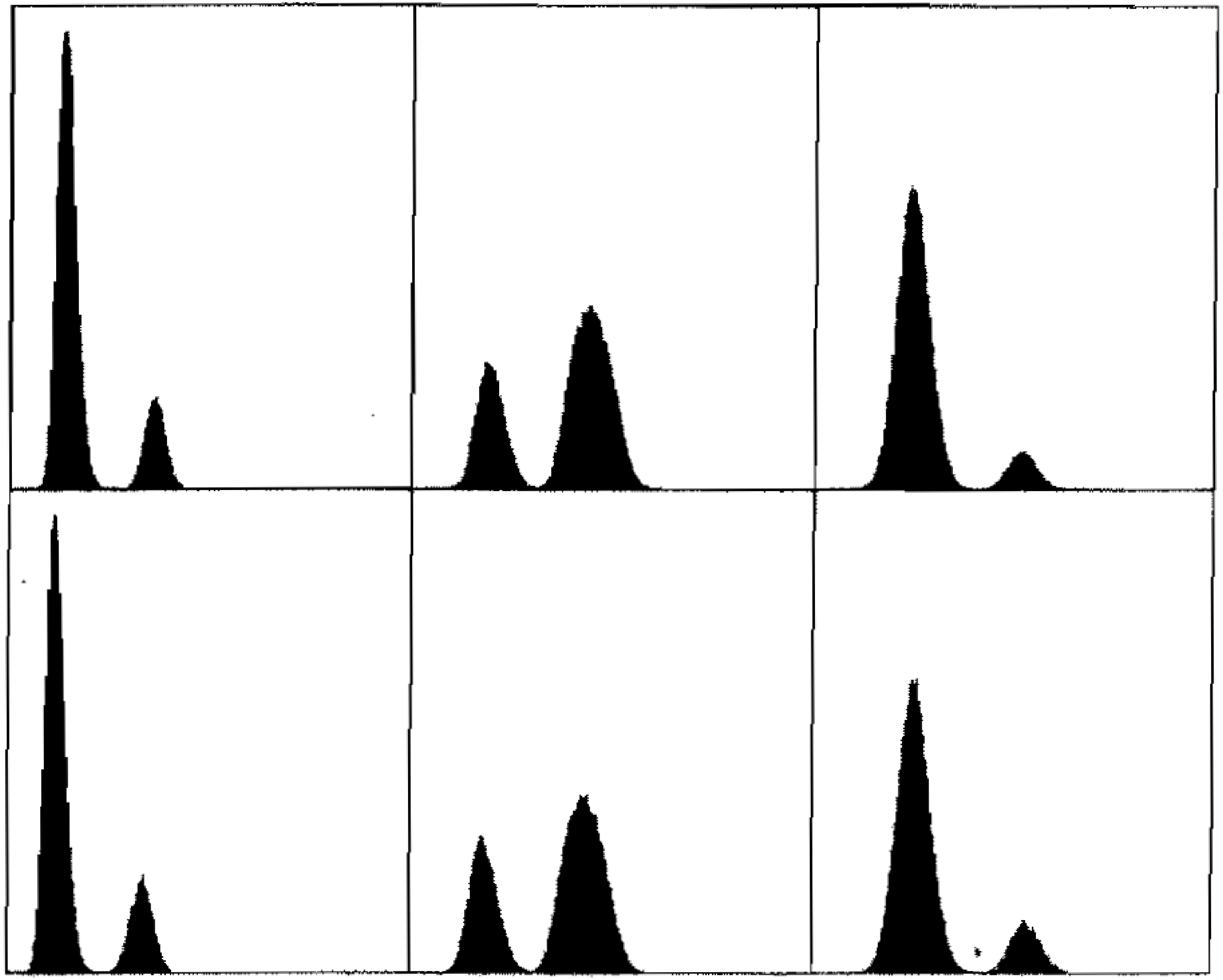


Image Partitioning

- Image partitioning: Subdivide the image in to non overlapping rectangles.
- This approach is used to compensate the non-uniformities in illumination/reflectance.
- The rectangles are chosen small enough so that the illumination of each is approximately uniform.
- Image subdivision generally works well when objects of interest and background occupy regions of reasonably comparable size
- When this is not the case method typically fails because of likelihood of subdivision containing only objects or only background pixels.

Variable Thresholding based on local properties

- Use of standard deviation and mean of local neighbourhood on every point.
- We illustrate the basic approach to local thresholding using the standard deviation and mean of the pixels in a neighborhood of every point in an image.
- These two quantities are quite useful for determining local thresholds because they are descriptors of local contrast and average intensity.
- 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 following are common forms of variable, local thresholds:

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

where a and b are nonnegative constants, and

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

- The segmented image is computed as

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

where $f(x, y)$ is the input image. This equation is evaluated for all pixel locations in the image, and a different threshold is computed at each location (x, y) using the pixels in the neighborhood S_{xy}.

Significant power (with a modest increase in computation) can be added to local thresholding by using predicates based on the parameters computed in the neighborhoods of (x, y):

$$g(x, y) = \begin{cases} 1 & \text{if } Q(\text{local parameters}) \text{ is true} \\ 0 & \text{if } Q(\text{local parameters}) \text{ is false} \end{cases} \tag{10.3-36}$$

where Q is a *predicate* based on parameters computed using the pixels in neighborhood S_{xy}. For example, consider the following predicate, $Q(\sigma_{xy}, m_{xy})$, based on the local mean and standard deviation:

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{true} & \text{if } f(x, y) > a\sigma_{xy} \text{ AND } f(x, y) > bm_{xy} \\ \text{false} & \text{otherwise} \end{cases} \tag{10.3-37}$$

$Q = \text{if } (f(x,y) > a\sigma_{xy} \text{ AND } f(x,y) > b m_G)$

$a = 30$

$b = 1.5$

a b
c d

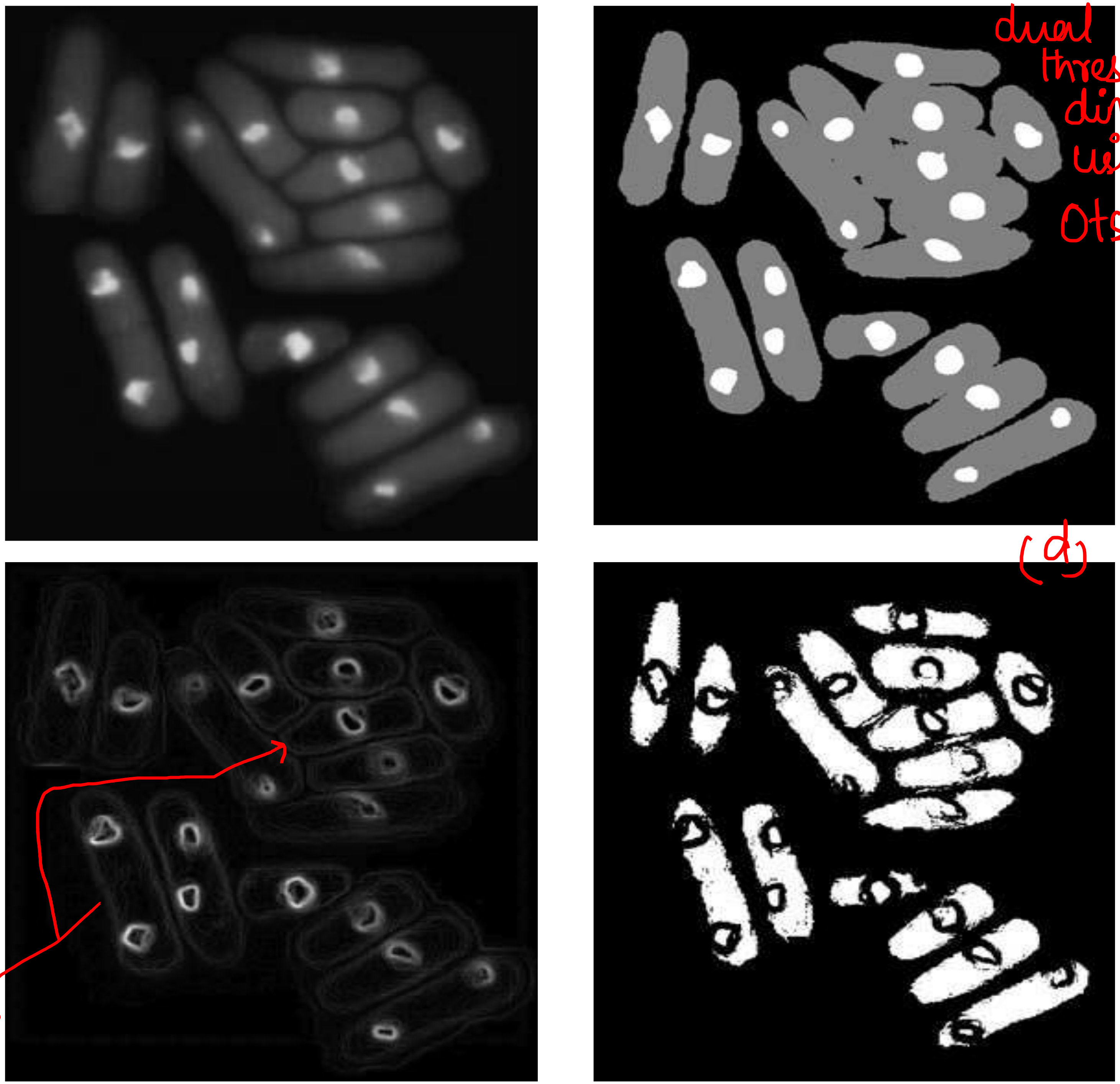


FIGURE 10.48
(a) Image from Fig. 10.43.
(b) Image segmented using the dual thresholding approach discussed in Section 10.3.6.
(c) Image of local standard deviations.
(d) Result obtained using local thresholding. using MG

Yeast cells image

3 predominant intensity levels

dual thresholding using Otsu

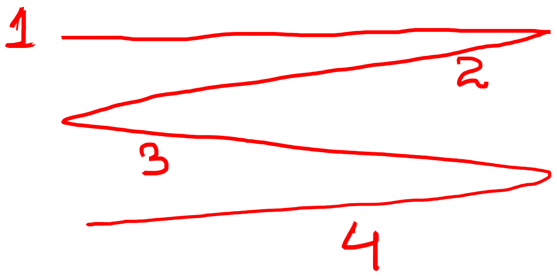
(d)

σ_{xy} for 3×3 neighbourhood

correct boundaries

Moving Average

- A special case of local thresholding method.
- Method is based on computing a moving average along scan lines of an image.
- Scanning is carried out line by line in a zigzag pattern to reduce illumination bias.
- Useful in applications like document processing where speed is the fundamental requirement.



- 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

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

- where n denotes the number of points used in computing the average and $m(1) = z_1/n$.
- The algorithm is initialized only once, not at every row. Because a moving average is computed for every point in the image.
- Segmentation is implemented using the usual equation with $T_{xy} = bm_{xy}$ where b is constant and m_{xy} is the moving average at point (x, y) in the input image.

$$m[k] = \frac{z[k]}{3} + \frac{z[k-1]}{3} + \frac{z[k-2]}{3}$$

$$m[k] = \underbrace{\frac{z[k]}{n} + \frac{z[k-1]}{n} + \frac{z[k-2]}{n} + \cdots \frac{z[k-(n-3)]}{n} + \frac{z[k-(n-2)]}{n}} + \frac{z[k-(n-1)]}{n}$$

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

$$m[k+1] = \frac{z[k+1]}{n} + \underbrace{\frac{\overset{(k+1)-1}{z[k+1-1]}}{n} + \frac{\overset{(k+1)-2}{z[k+1-2]}}{n} + \cdots + \frac{z[k+1-(n-2)]}{n} + \frac{\overset{(k+1)-(n-1)}{z[k+1-(n-1)]}}{n}}_{z(k)/n \quad z(k-1)/n \quad + \cdots \quad z(k-(n-2))/n}$$

$m(k+1) = m(k) + \frac{1}{n}(z_{k+1} - z_{k-n})$

$$m[k+1] = \frac{z[k+1]}{n} + \underbrace{m[k] - \frac{z[k-(n-1)]}{n}}$$

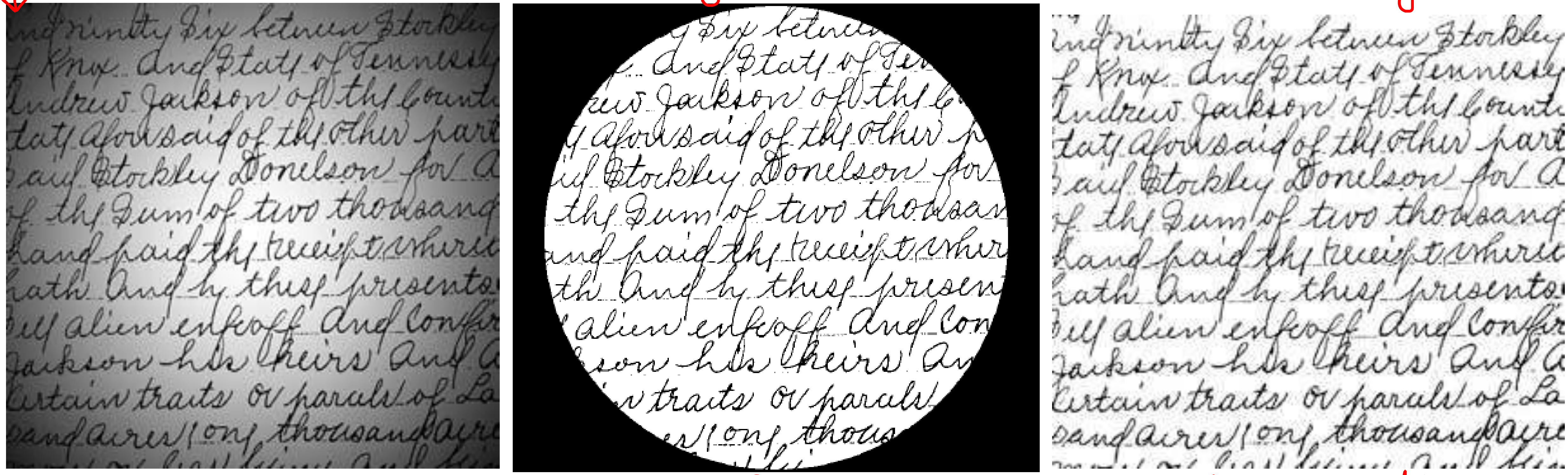
$$m[1] = \frac{z[1]}{n} + \frac{z[1-1]}{n} + \frac{z[1-2]}{n} + \cdots + \frac{z[1-(n-1)]}{n}$$

$$m[1] = \frac{z[1]}{n} + 0 + 0 + \cdots + 0$$

spot intensity pattern

segmented image using otsu method

using moving averages

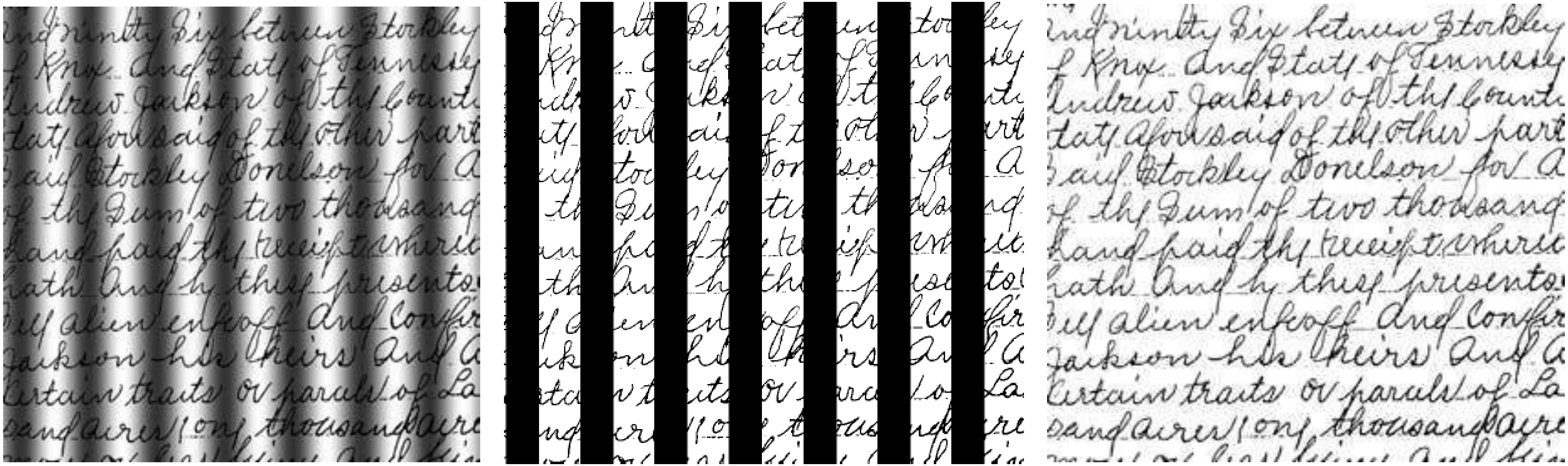


* let n equal 5 times the average stroke width

$n = 20 \quad b = 0.5$

a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using 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.

moving average works best when objects of interest are small wrt image size (hand written text)

Edge Linking and Boundary Detection

Edge Linking

- Even after hysteresis thresholding, the detected pixels do not completely characterize edges completely due to occlusions, non-uniform illumination and noise. Edge linking may be:
  - –Local: requiring knowledge of edge points in a small neighbourhood.
  - –Regional: requiring knowledge of edge points on the boundary of a region.
  - –Global: the Hough transform, involving the entire edge image.

Edge Linking by Local Processing

- A simple algorithm:

1. Compute the gradient magnitude and angle arrays $M(x, y)$ and $\alpha(x, y)$ of the input image $f(x, y)$.
2. Let S_{xy} denote the neighborhood of pixel (x, y).
3. A pixel (s, t) in S_{xy} is linked to (x, y) if:

$$|M(s,t) - M(x,y)| \leq E \qquad |\alpha(s,t) - \alpha(x,y)| \leq A$$

- Computationally expensive as all neighbours of every pixel should be examined.

A faster algorithm:

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:

$$g(x,y)=\begin{cases}1 & M(x,y)\geq T_M \text{ and } \alpha(x,y)\in[A-T_A, A+T_A]\\ 0 & \text{otherwise}\end{cases}$$

A is angle directi[on]
$\pm T_A$ gives acceptable band

3. Scan the rows of $g(x,y)$ and fill (set to 1) all gaps (zeros) that do not exceed a specified length K.
4. To detect gaps in any other direction $A=\theta$, rotate $g(x,y)$ by θ and apply the horizontal scanning.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

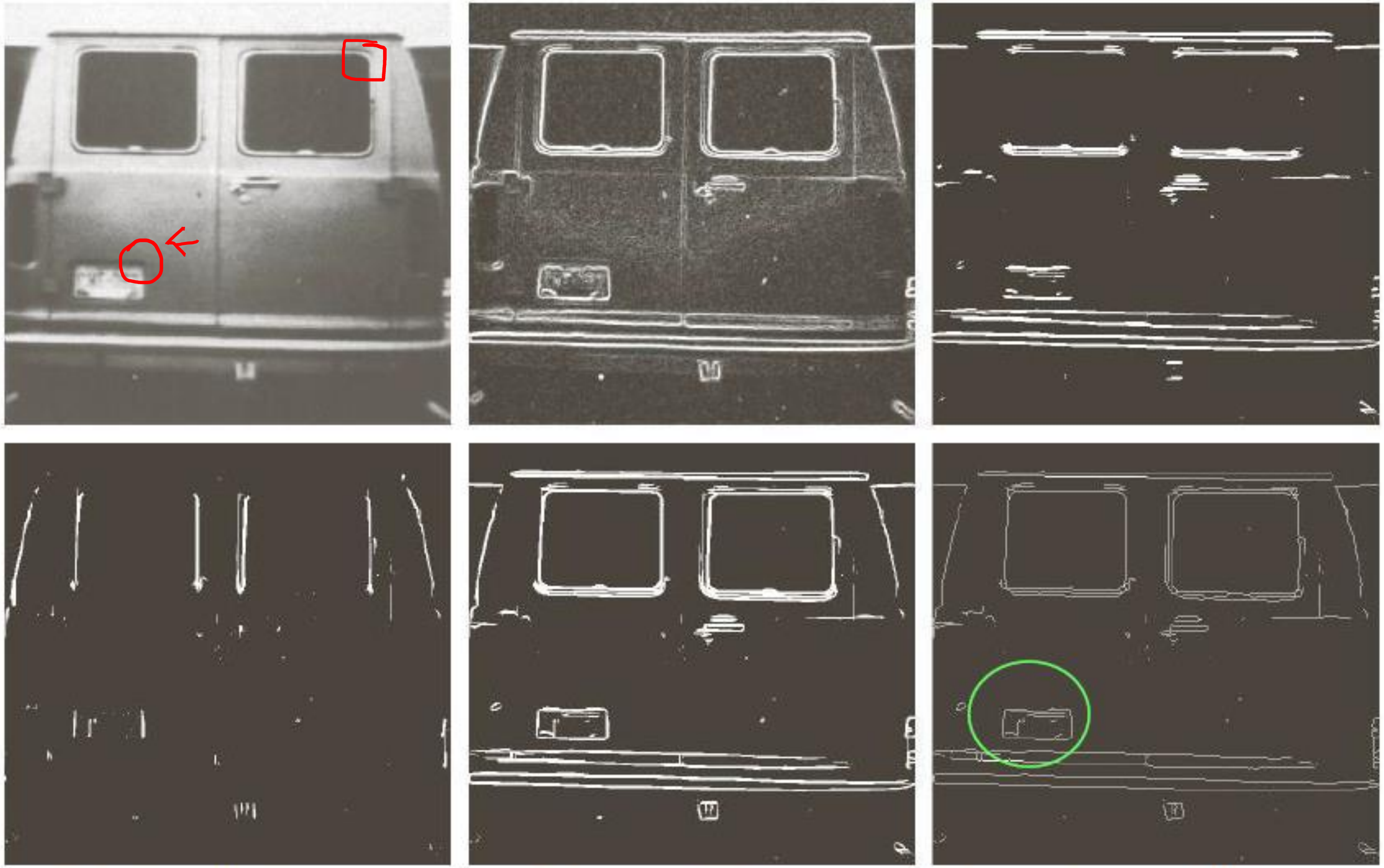
First time
$K=3$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

2nd time

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Image   Gradient magnitude   Horizontal linking



Vertical linking   Logical OR   Morphological thinning

$T_M = 30\%$ of max gradient value.

$A = 90^\circ$ $T_A = 45^\circ$

filling all gaps of 25 or fewer pixels

We may detect the licence plate from the ratio width/length (2:1 in the USA)

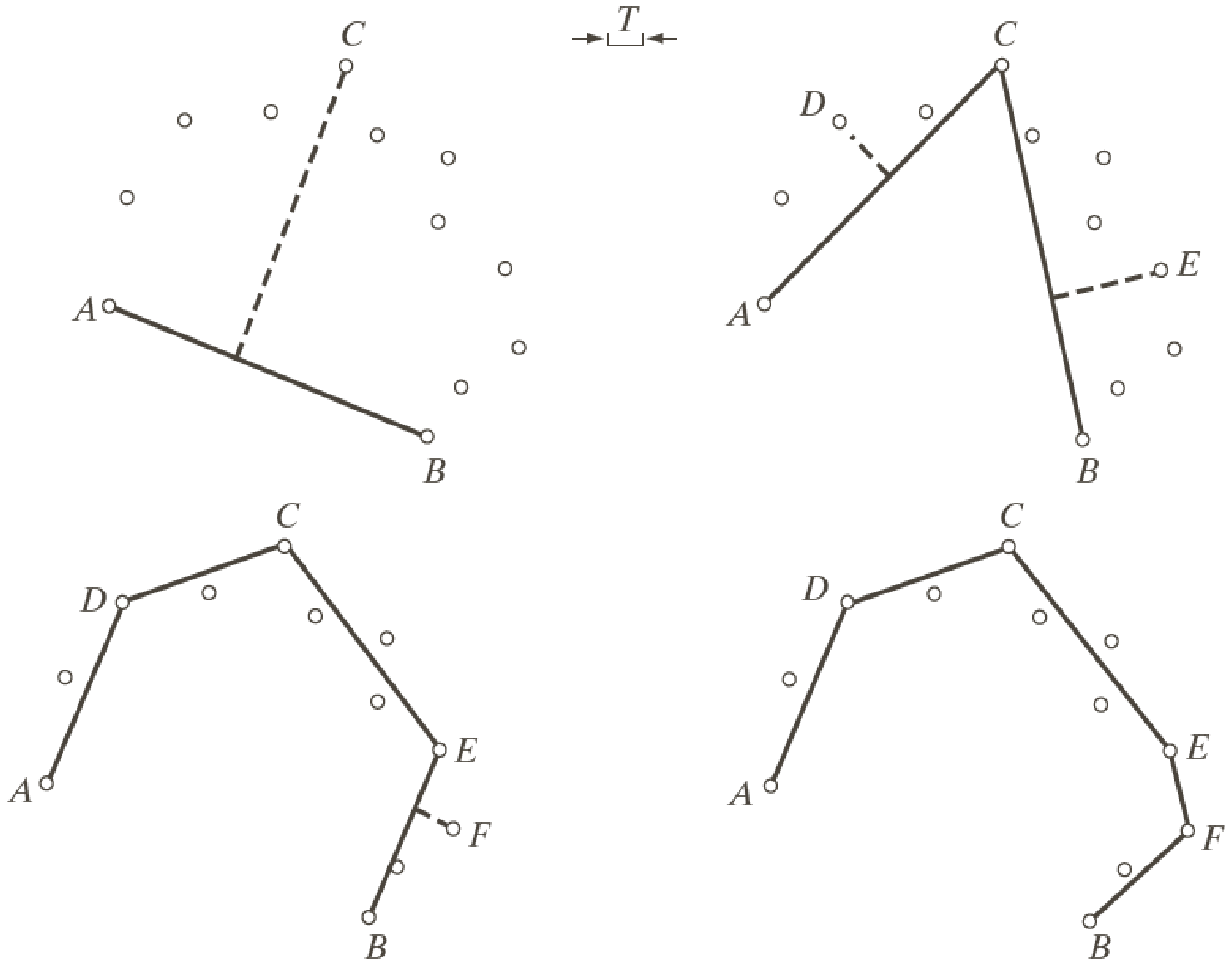
Edge Linking by Regional Processing

- Often, the location of regions of interest is known and pixel membership to regions is available.
- Approximation of the region boundary by fitting a polygon. Polygons are attractive because:

    – They capture the essential shape.

    – They keep the representation simple.

Basic Mechanism for Polygon Fitting



–Given the end points *A* and *B*, compute the straight line *AB*.

–Compute the perpendicular distance from all other points to this line.

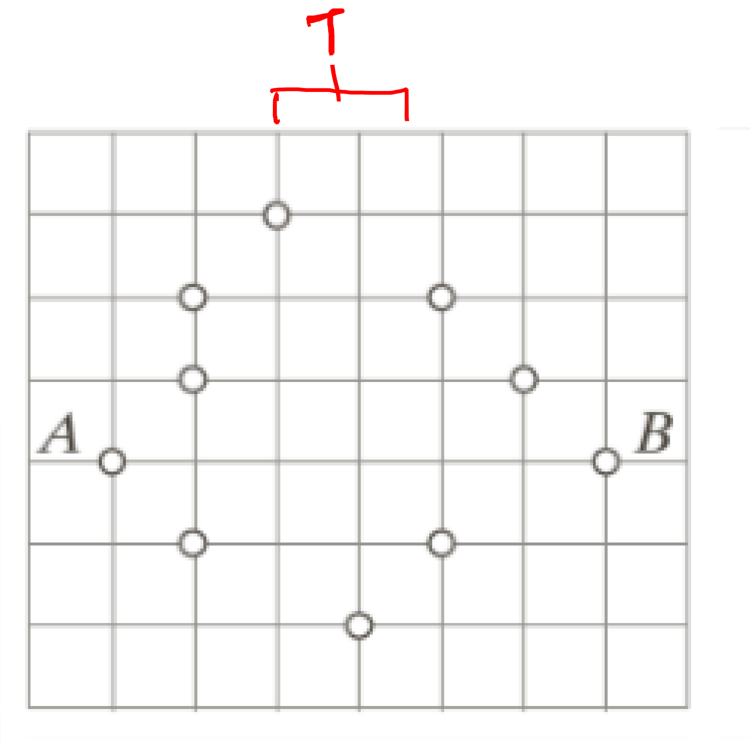
–If this distance exceeds a threshold, the corresponding point *C* having the maximum distance from *AB* is declared a vertex.

–Compute lines *AC* and *CB* and continue.

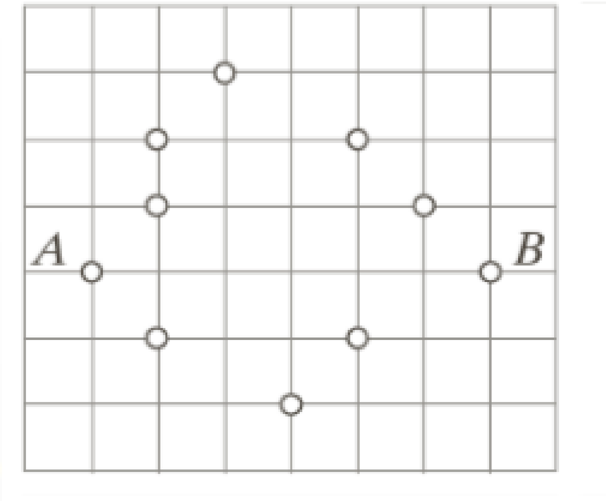
- Requirements
  - –Two starting points must be specified (e.g. rightmost and leftmost points).
  - –The points must be ordered (e.g. clockwise).
- Variations of the algorithm handle both open and closed curves.
- If this is not provided, it may be determined by distance criteria:
  - –Uniform separation between points indicate a closed curve.
  - –A relatively large distance between consecutive points with respect to the distances between other points indicate an open curve.

Algorithm:

1. Let P be the sequence of ordered, distinct, 1-valued points of a binary image. Specify two starting points, A and B.
2. Specify a threshold, T, and two empty stacks, OPEN and ClOSED.



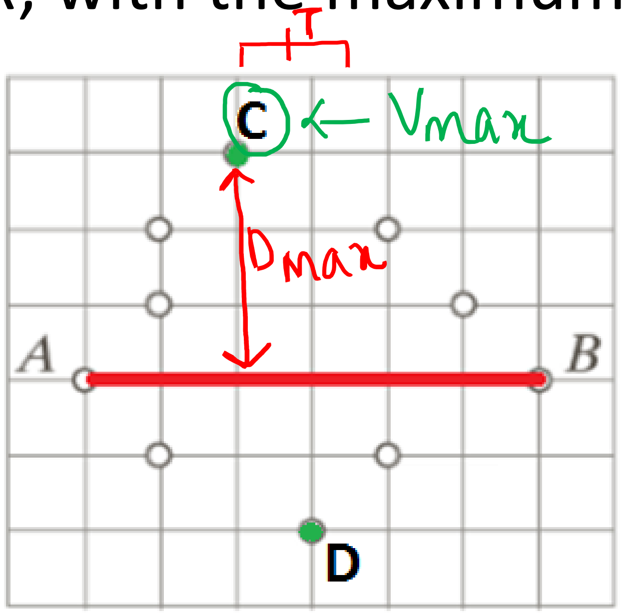
3. If the points in P correspond to a closed curve, put A into OPEN and put B into OPEN and CLOSED. If the points correspond to an open curve, put A into OPEN and B into CLOSED.



CLOSED	OPEN
B	B,A

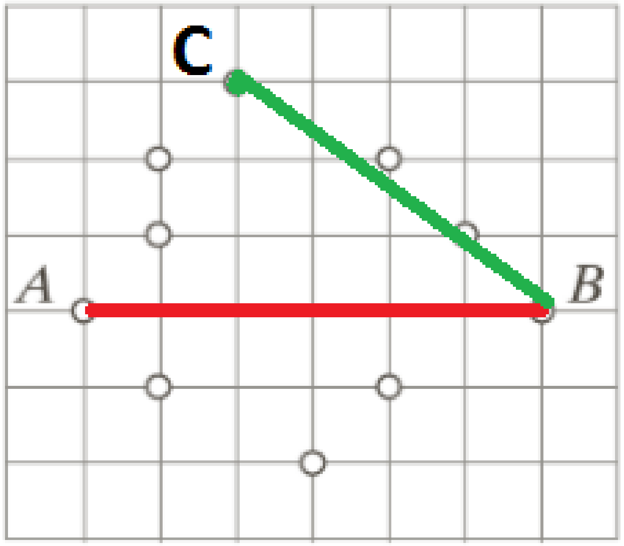
4. Compute the parameters of the line passing from the last vertex in CLOSED to the last vertex in OPEN (LINE: BA)

5. Compute the distances from the line in Step 4 to all the points in P whose sequence places them between the vertices from Step 4. Select the point, *Vmax*, with the maximum distance, *Dmax*



6. *If Dmax> T, place Vmax at the end of the OPEN stack as a new vertex. Go to step 4.*

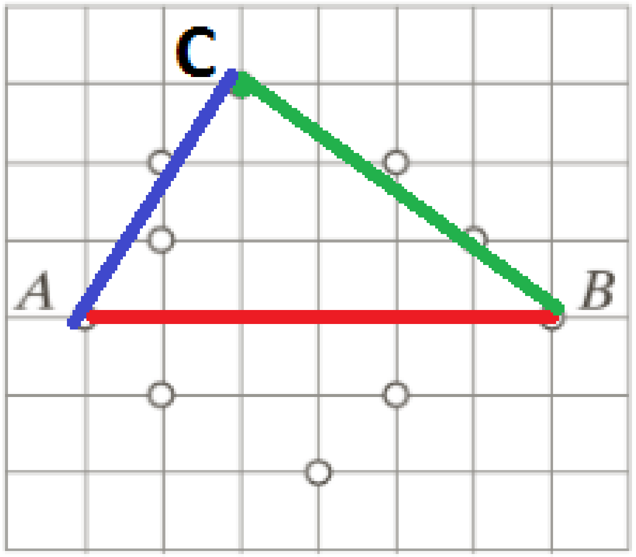
CLOSED	OPEN	CURVE SEGMENT SELECTED IN STEP :4
B	B,A,C	BC



- STEP: 5,6 Dmax does not exceed threshold.

7. Else, remove the last vertex from OPEN and insert it as the last vertex of CLOSED.
8. If OPEN is not empty, go to step 4.

CLOSED	OPEN	CURVE SEGMENT SELECTED IN STEP :4
B,C	B,A	CA

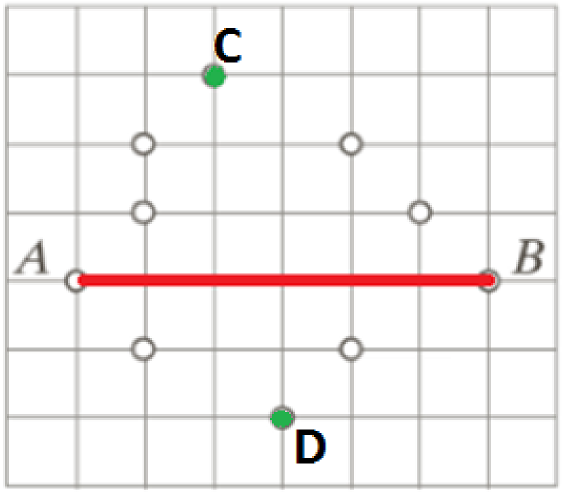


- STEP: 5,6 Dmax does not exceed threshold.

7. Else, remove the last vertex from OPEN and insert it as the last vertex of CLOSED.
8. If OPEN is not empty, go to step 4.

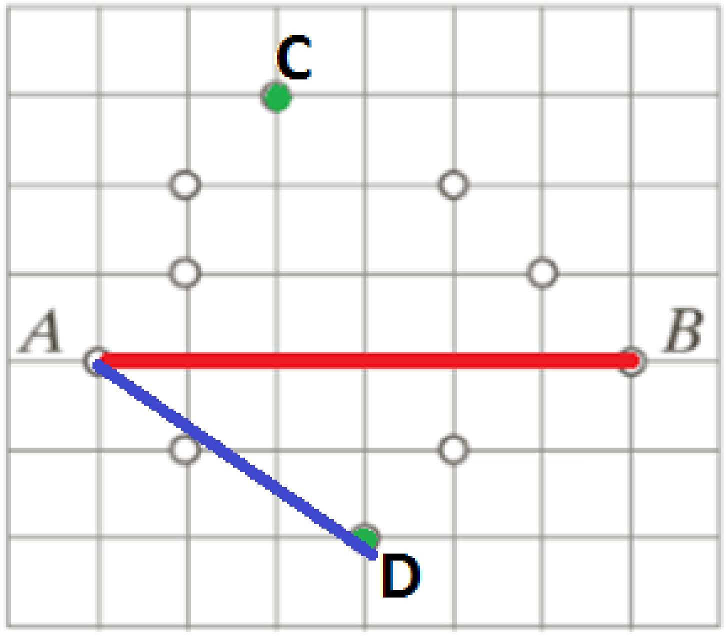
CLOSED	OPEN	CURVE SEGMENT SELECTED IN STEP :4
B,C,A	B	AB

5. Compute the distances from the line in Step 4 to all the points in P whose sequence places them between the vertices from Step 4. Select the point, Vmax, with the maximum distance, Dmax



6. If Dmax> T, place Vmax at the end of the OPEN stack as a new vertex. Go to step 4.

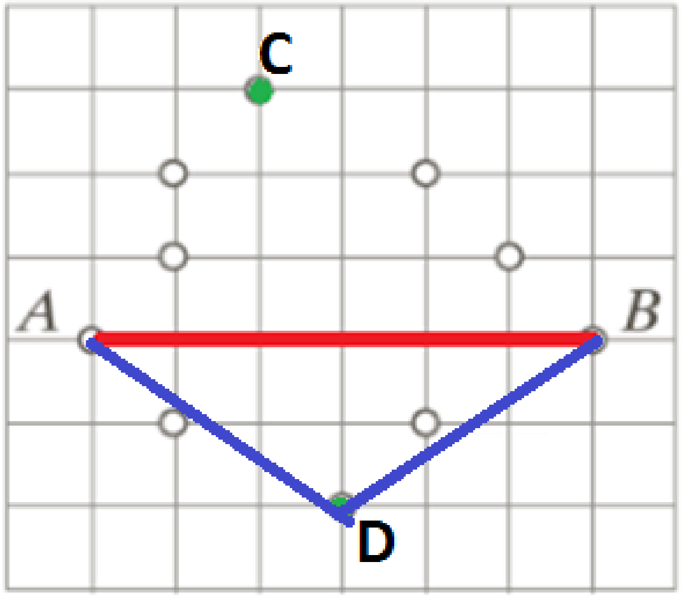
CLOSED	OPEN	CURVE SEGMENT SELECTED IN STEP :4
B,C,A	B,D	AD



- STEP: 5,6 Dmax does not exceed threshold.

7. Else, remove the last vertex from OPEN and insert it as the last vertex of CLOSED.
8. If OPEN is not empty, go to step 4.

CLOSED	OPEN	CURVE SEGMENT SELECTED IN STEP :4
B,C,A,D	B	BD× DB

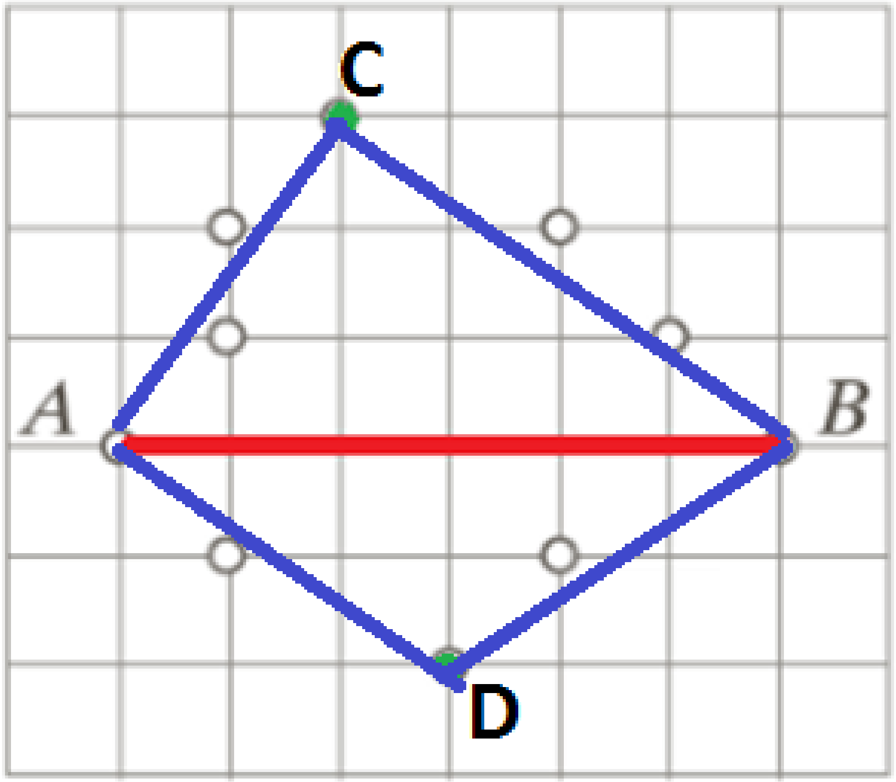


- STEP: 5,6 Dmax does not exceed threshold.

7. Else, remove the last vertex from OPEN and insert it as the last vertex of CLOSED.
8. If OPEN is not empty, go to step 4.

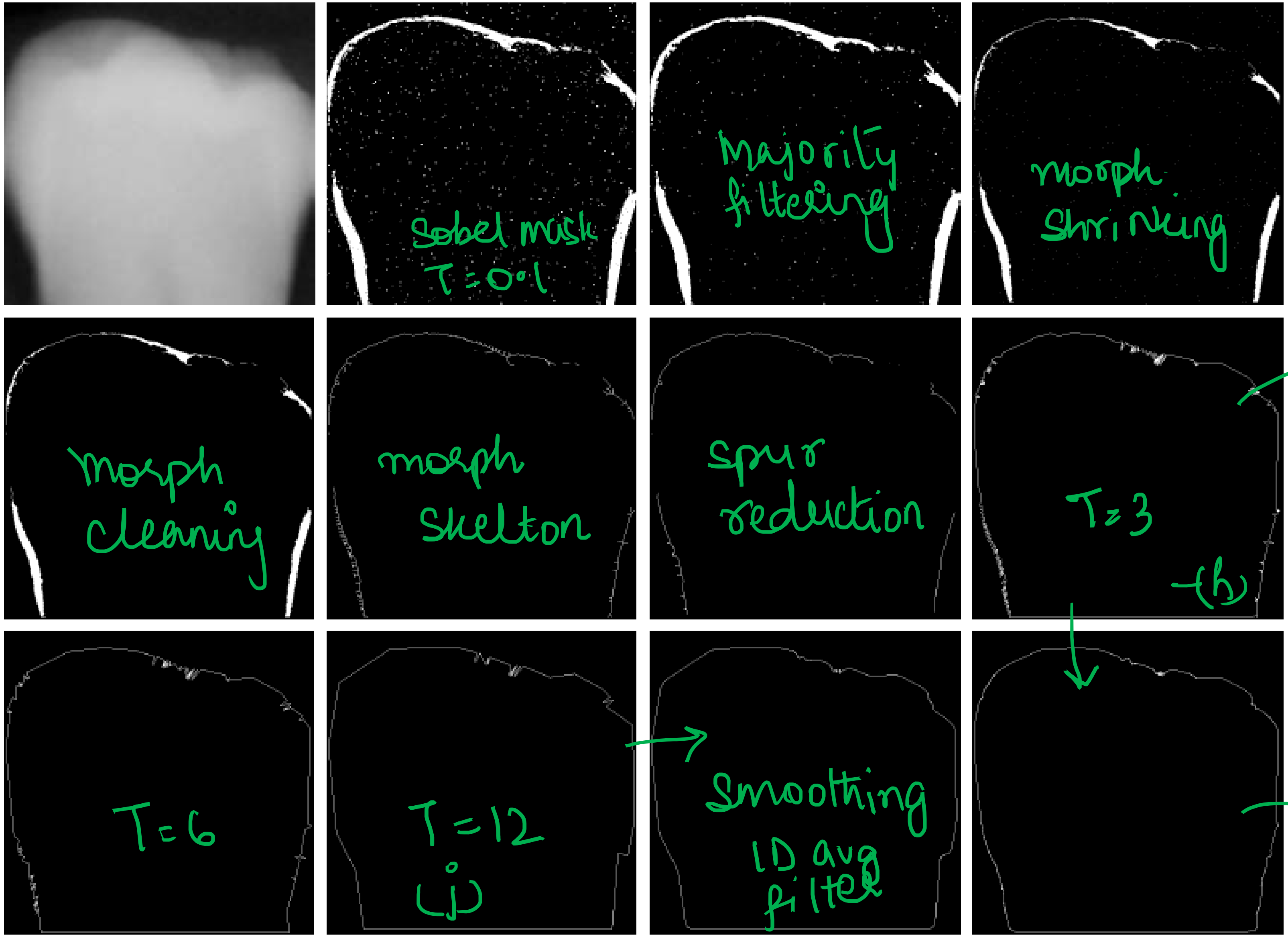
CLOSED	OPEN	CURVE SEGMENT SELECTED IN STEP :4
B,C,A,D,B	-	-

9. Else, exit. The vertices in CLOSED are the vertices of the polygonal fit to the points in P.


C
A
B
D

CLOSED	OPEN	Curve segment processed	Vertex generated
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—
B, C, A, D	B	(DB)	—
B, C, A, D, B	Empty	—	—

TABLE 10.1
Step-by-step details of the mechanics in Example 10.11.



a	b	c	d
e	f	g	h
i	j	k	l

polygon fitting

Regional processing for edge linking is used in combination with other methods in a chain of processing.

preserves boundary features.

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.