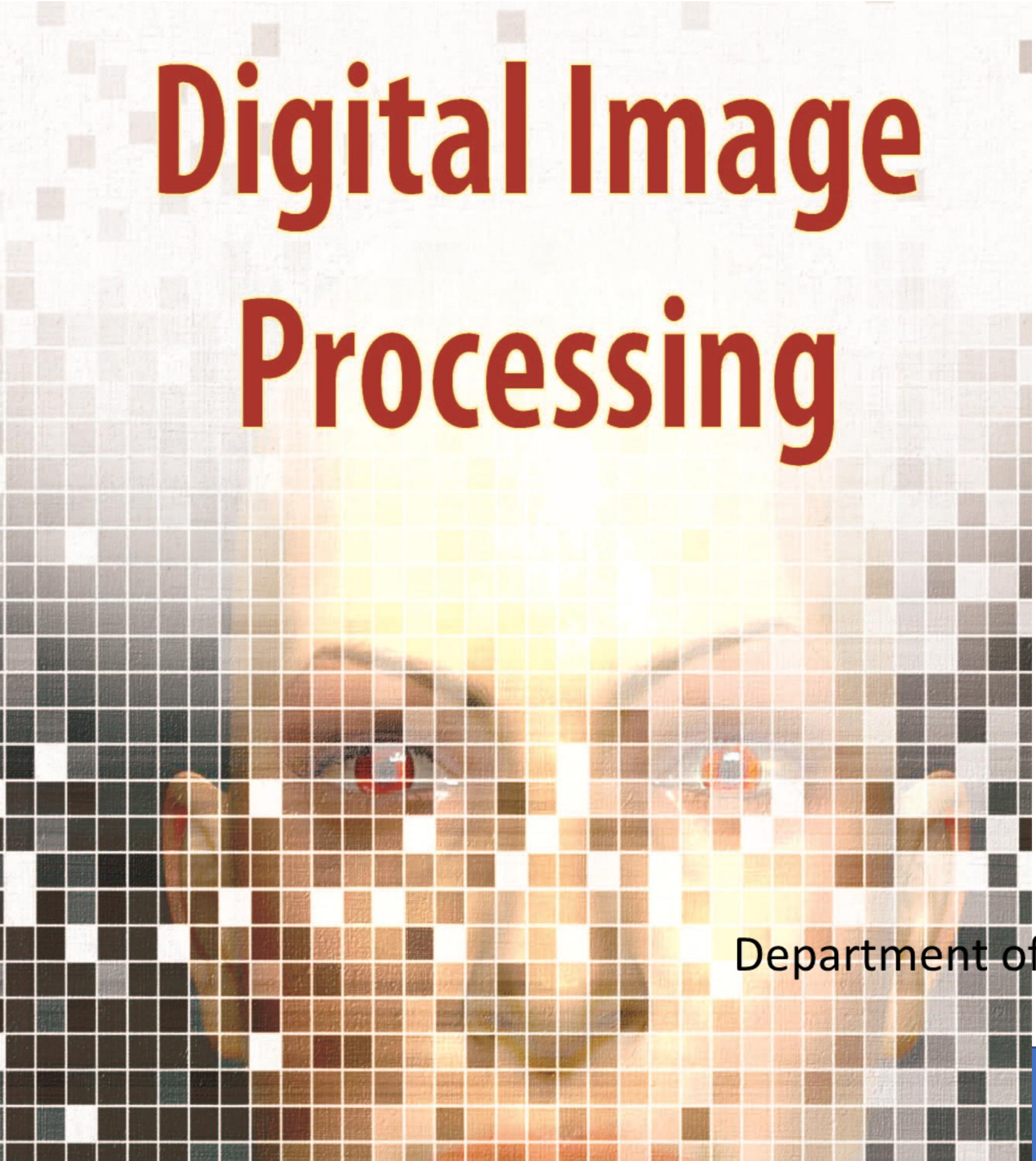


Image Segmentation

Prof. Sheli Sinha Chaudhuri
Department of Electronics and Telecommunication Engineering
Jadavpur University

Chapter 10
Image Segmentation

Contents

Image Segmentation

- Image segmentation divides an image into regions that are connected and have some similarity within the region and some difference between adjacent regions.
- The goal is usually to find individual objects in an image.
- For the most part there are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.
  - Similarity may be due to pixel intensity, color or texture.
  - Differences are sudden changes (discontinuities) in any of these, but especially sudden changes in intensity along a boundary line, which is called an edge.

Detection of Discontinuities

- There are three kinds of discontinuities of intensity: points, lines and edges.
- The most common way to look for discontinuities is to scan a small mask over the image. The mask determines which kind of discontinuity to look for.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

FIGURE 10.1 A general 3×3 mask.

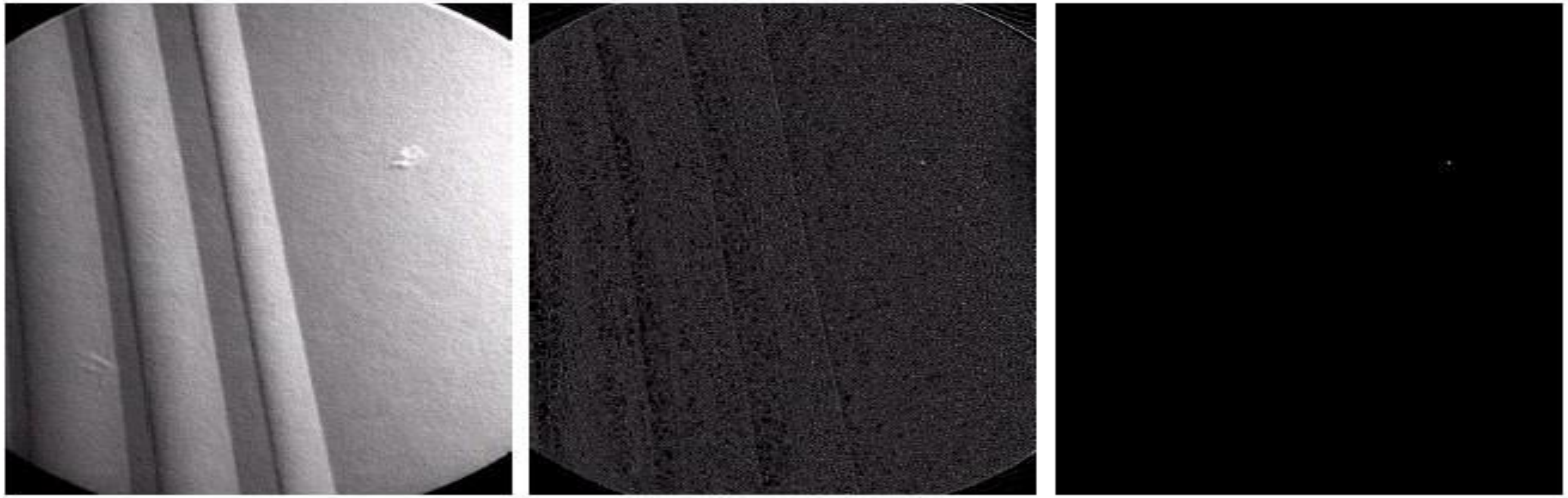
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Detection of Discontinuities: Point Detection

$$R \geq T$$

Where $|T|$: a non-negative threshold

−1	−1	−1
−1	8	−1
−1	−1	−1



a
b c d

FIGURE 10.2 (a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Detection of Discontinuities: Line Detection

- Only slightly more common than point detection is to find a one pixel wide line in an image.
- For digital images the only three point straight lines are only horizontal, vertical, or diagonal (+ or –45°).

FIGURE 10.3 Line masks.

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

−1	−1	2
−1	2	−1
2	−1	−1

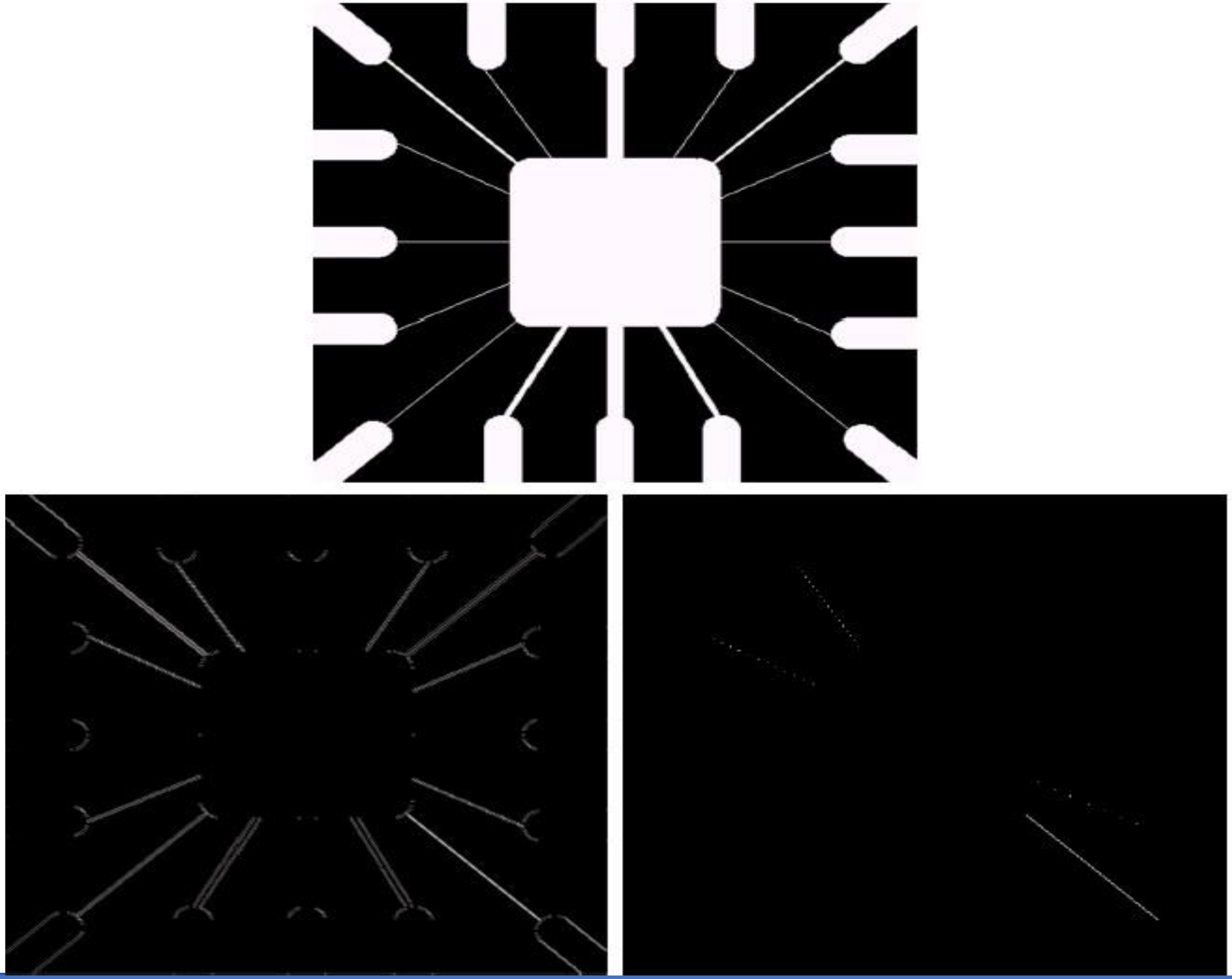
Vertical

−1	2	−1
−1	2	−1
−1	2	−1

−45°

2	−1	−1
−1	2	−1
−1	−1	2

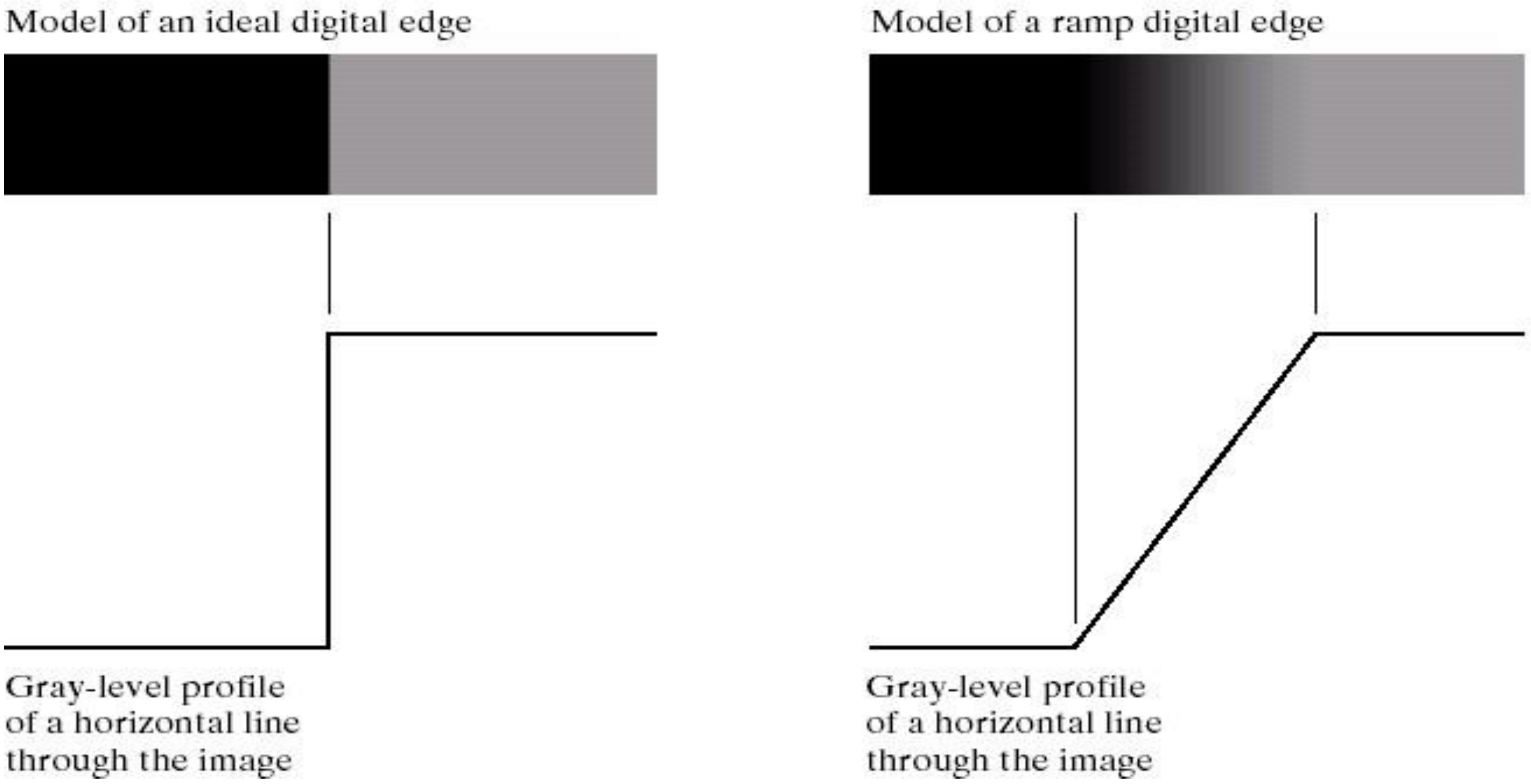
Detection of Discontinuities: Line Detection



a
b c

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

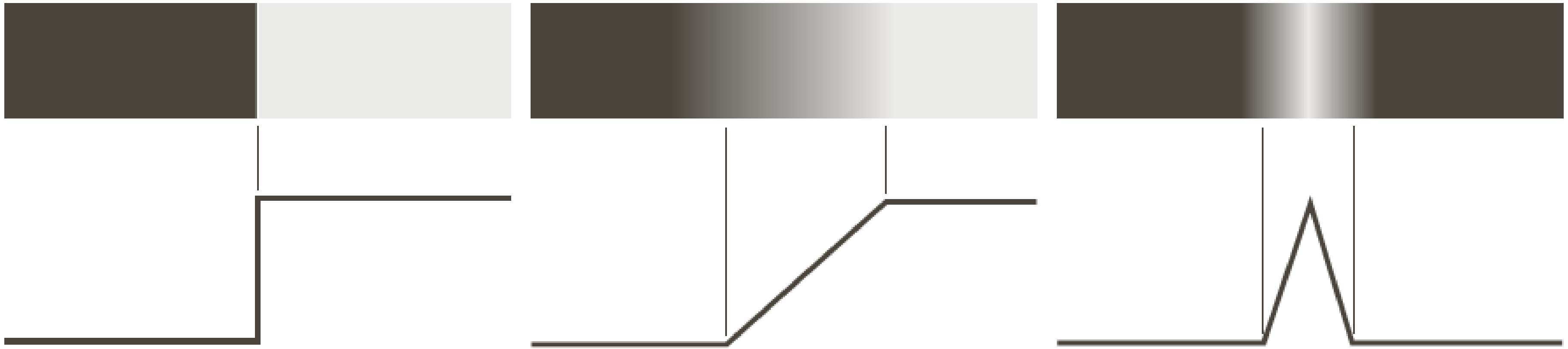
Detection of Discontinuities: Edge Detection



a b

FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

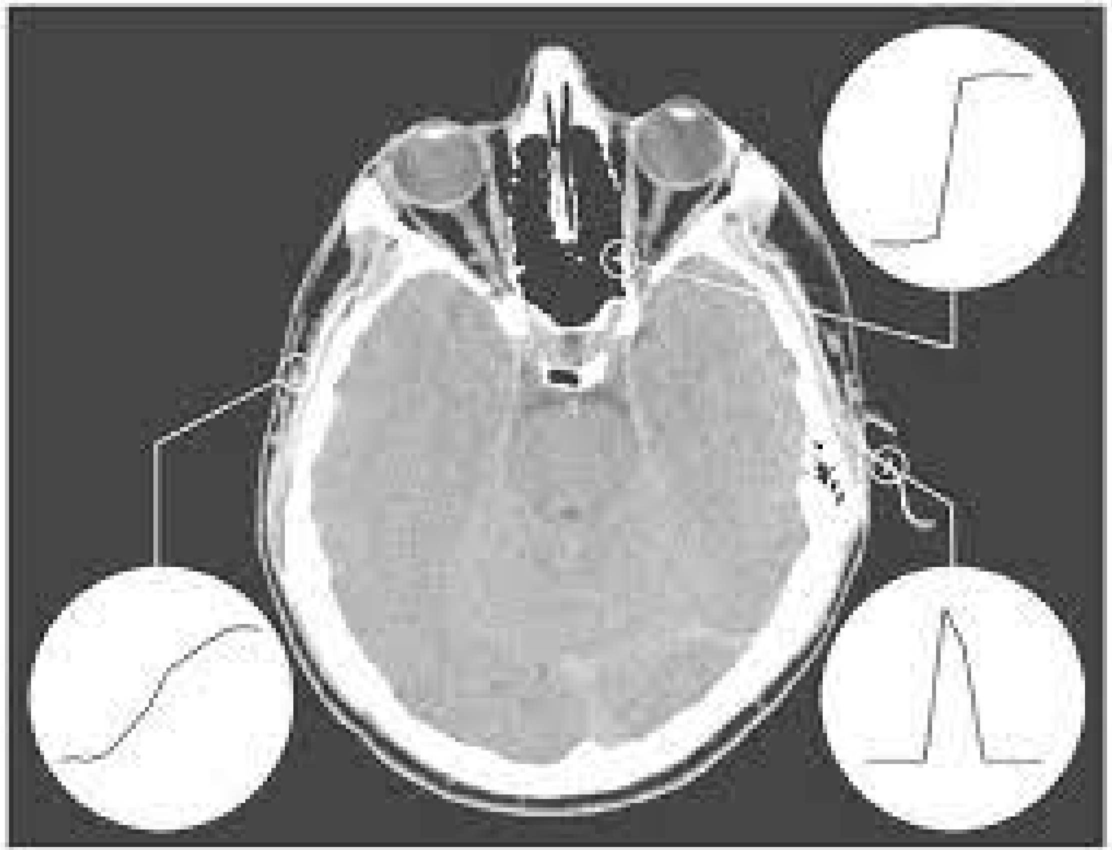
Detection of Discontinuities: Edge Detection



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.

Detection of Discontinuities: Edge Detection

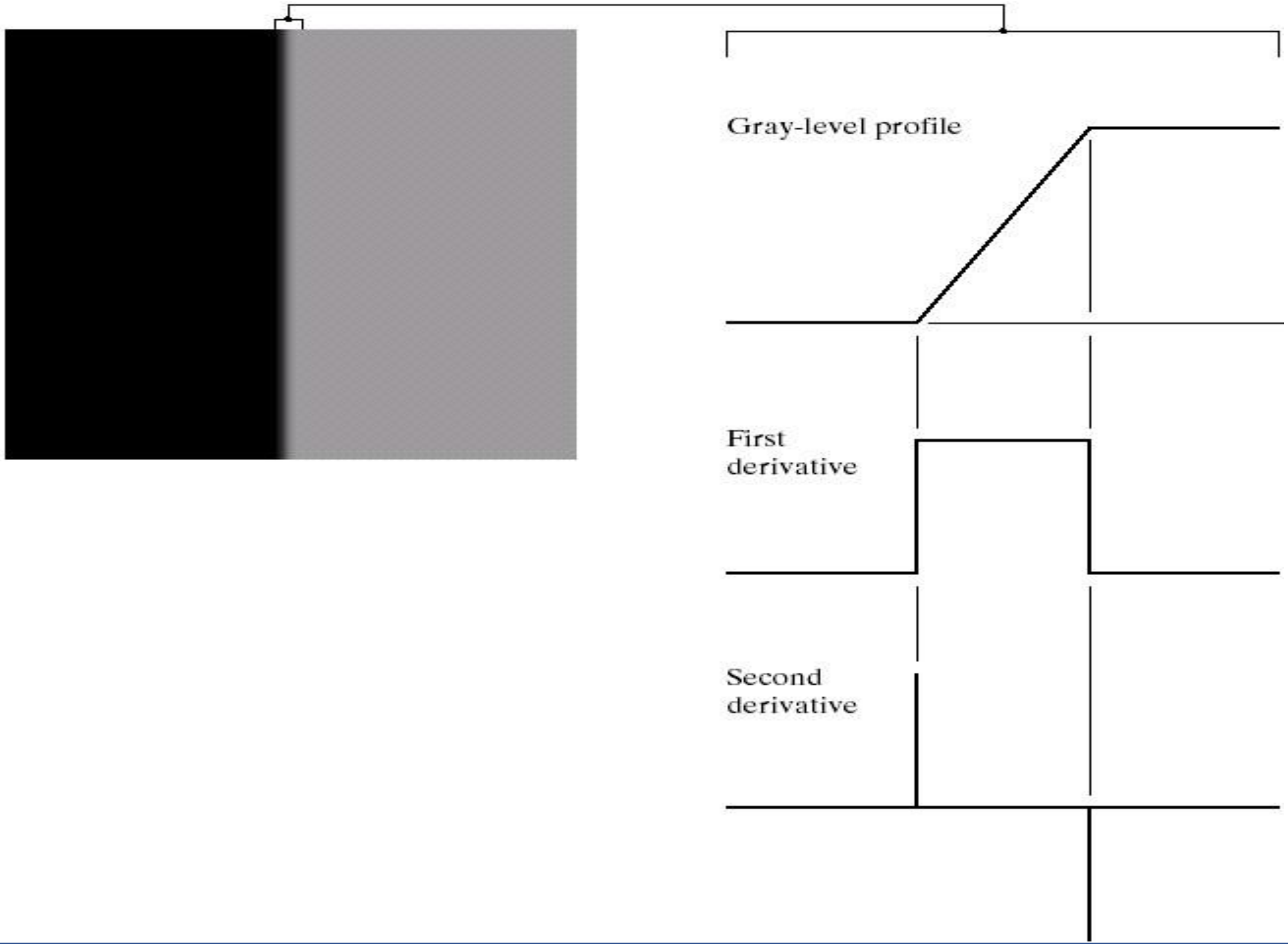


A 1508x1970 image showing actual ramp (bottom left), step (top right) and roof edge profiles.
The profile are from dark to light, in the areas indicated by short lines shown in the small circles.
The ramp and srep profiles span 9 pixels and 2 pixels.
(Image by Dr. David R Pickens, Vanderbilt University)

Detection of Discontinuities Edge Detection

a b

FIGURE 10.6
(a) Two regions separated by a vertical edge.
(b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.



Detection of Discontinuities Edge Detection

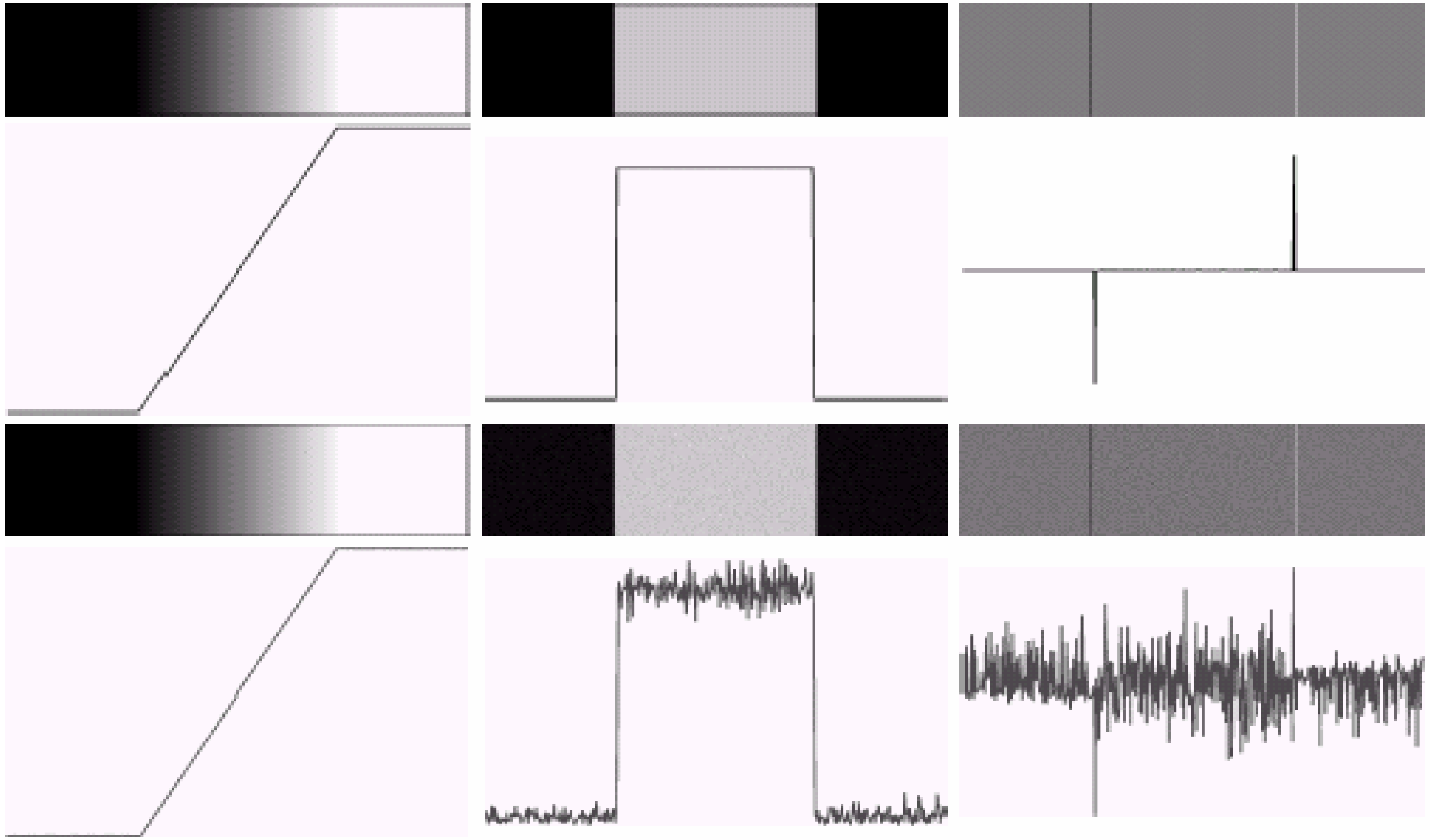


FIGURE 10.7 First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0,$ and 10.0, respectively. Second column: first-derivative images and gray-level profiles. Third column: second-derivative images and gray-level profiles.

a
b
c
d

Detection of Discontinuities Edge Detection

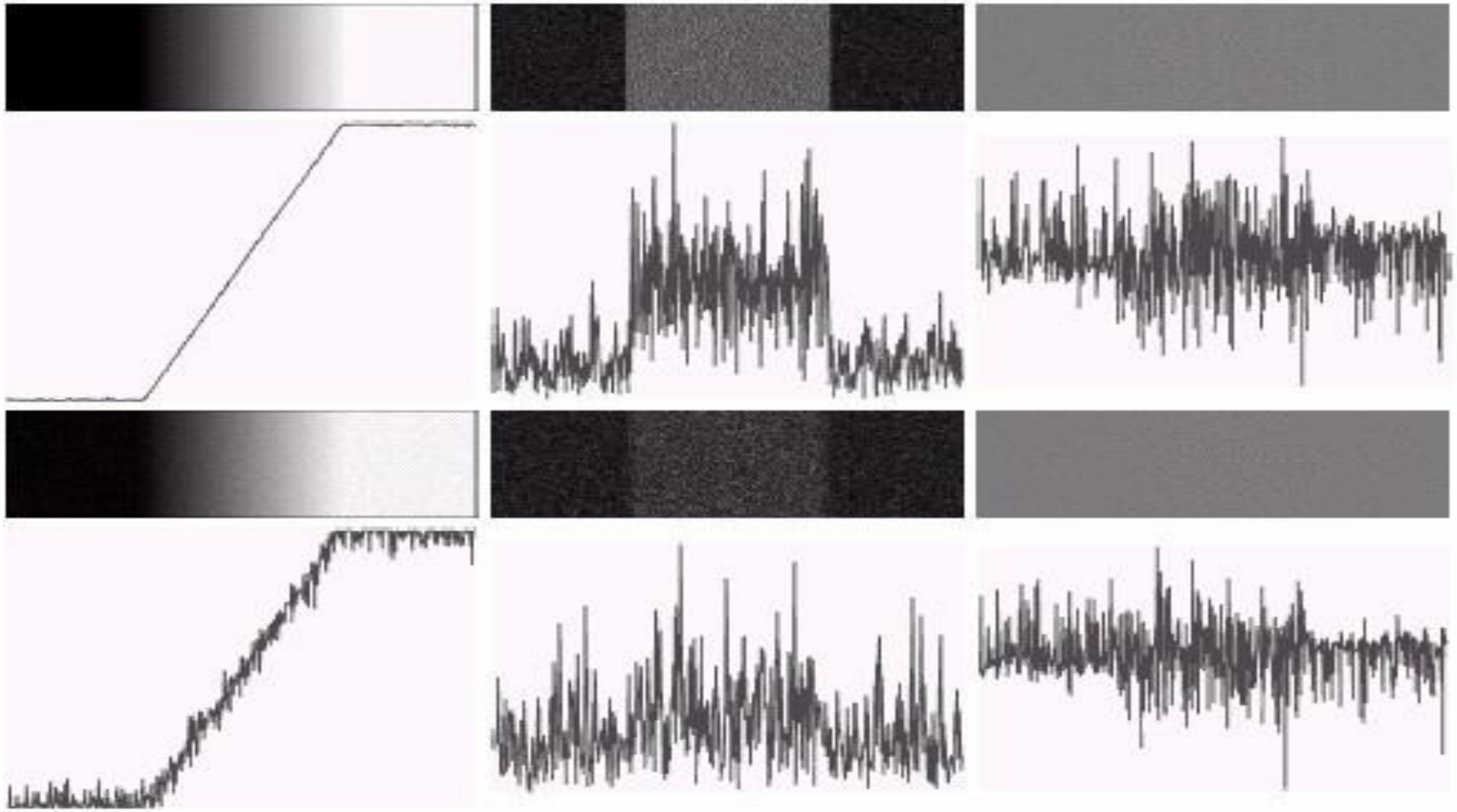


FIGURE 10.7 First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0,$ and 10.0, respectively. Second column: first-derivative images and gray-level profiles. Third column: second-derivative images and gray-level profiles.

a
b
c
d

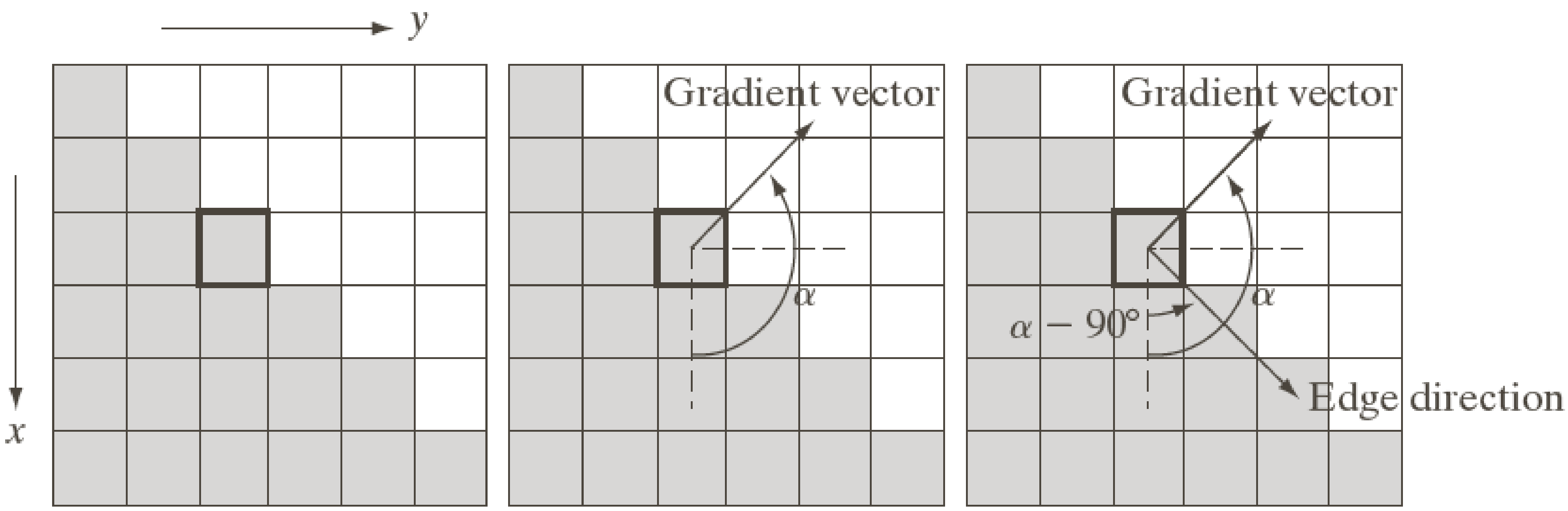
Detection of Discontinuities Gradient Operators

- First-order derivatives:
  - The gradient of an image *f*(*x*,*y*) at location (*x*,*y*) is defined as the vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

  - The magnitude of this vector: $\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$
  - The direction of this vector: $\alpha(x, y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$
  - It points in the direction of the greatest rate of change of *f* at location (x,y)

Detection of Discontinuities Gradient Operators



a b c

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.

Detection of Discontinuities Gradient Operators

Roberts cross-gradient operators

−1	0		0	−1
0	1		1	0

Roberts

Prewitt operators

−1	−1	−1		−1	0	1
0	0	0		−1	0	1
1	1	1		−1	0	1

Prewitt

Sobel operators

−1	−2	−1		−1	0	1
0	0	0		−2	0	2
1	2	1		−1	0	1

Sobel

Detection of Discontinuities Gradient Operators

Prewitt masks for detecting diagonal edges

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

Sobel masks for detecting diagonal edges

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a	b
c	d

FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.

Detection of Discontinuities Gradient Operators: Example

a b
c d

FIGURE 10.10 (a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

$$\nabla f \approx |G_x| + |G_y|$$

Detection of Discontinuities Gradient Operators: Example

a b
c d

FIGURE 10.11 Same sequence as in Fig. 10.10, but with the original image smoothed with a 5×5 averaging filter.

Detection of Discontinuities Gradient Operators: Example

a b

FIGURE 10.12 Diagonal edge detection. (a) Result of using the mask in Fig. 10.9(c). (b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Detection of Discontinuities Gradient Operators: Example

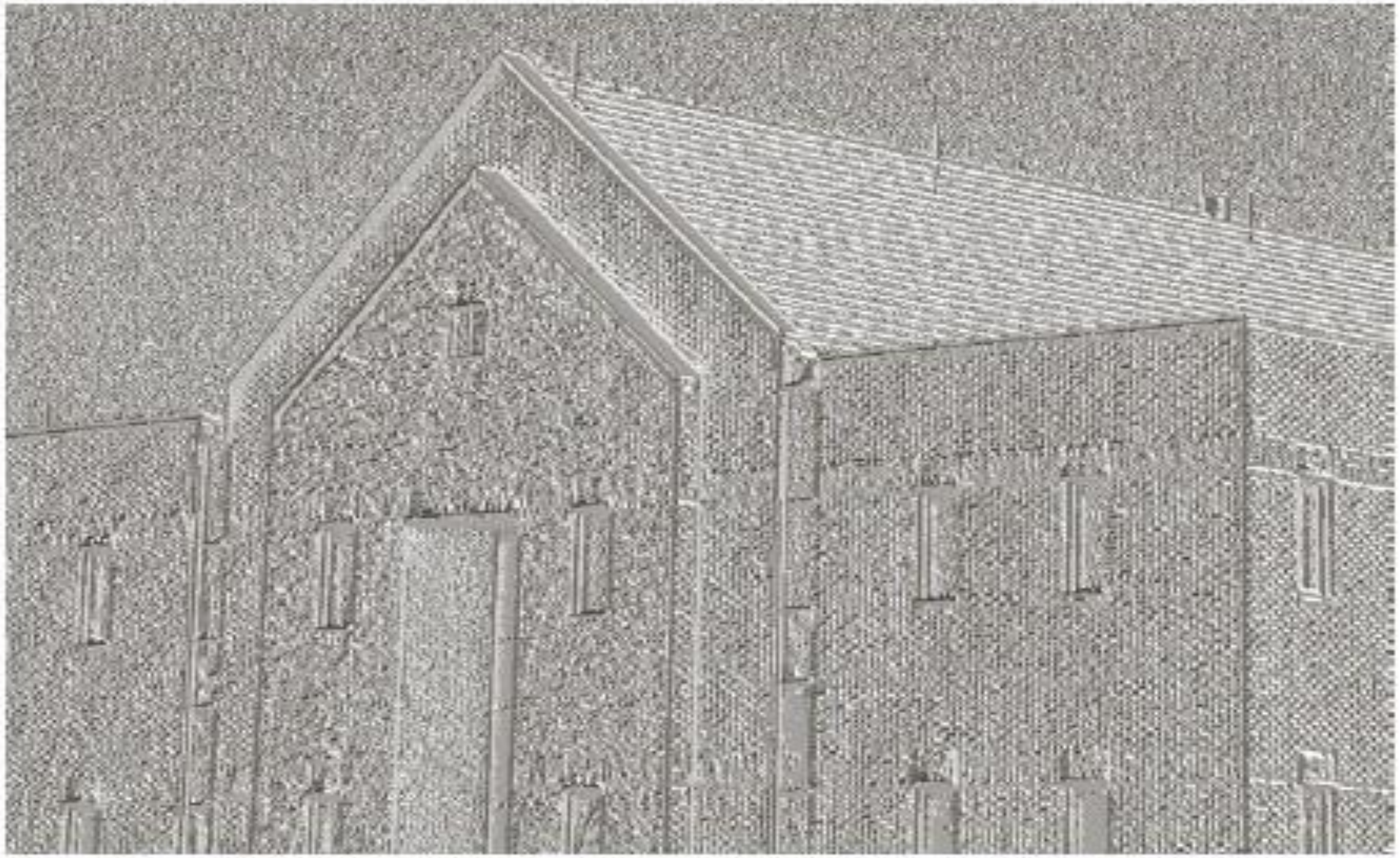


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

Detection of Discontinuities Gradient Operators

Second-order derivatives: (The Laplacian)

- – The Laplacian of an 2D function $f(x,y)$ is defined as

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

– Two forms in practice:

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

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

More Advanced Techniques for Edge Detection

Marr-Hildreth edge detector

Marr and Hildreth argued that

1) Intensity changes are dependent of image scale and so their detection requires the use of operators different sizes and
2) That a sudden intensity change will give rise to a peak or trough in the first derivative or, equivalently, to zero crossing in the second derivative.

Two salient feature:

1)It should be differential operator capable of computing first or second order derivatives at every point in an image
2)It should be capable of being tuned to act at any desired scale, so that large operator can be used to detect blurry edges and small operators to detect sharply focused fine detail

More Advanced Techniques for Edge Detection

- Consider the function:

A Gaussian function

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}} \quad \text{where} \quad r^2 = x^2 + y^2$$

and σ : the standard deviation

- The Laplacian of *h* is

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

The Laplacian of a Gaussian (LoG)

- The Laplacian of a Gaussian sometimes is called the Mexican hat function. It also can be computed by smoothing the image with the Gaussian smoothing mask, followed by application of the Laplacian mask.

More Advanced Techniques for Edge Detection

- Marr-Hildreth edge detector
- Marr and Hildreth argued that the most satisfactory operator fulfilling these conditions is the filter $\nabla^2 G$ where, ∇^2 is the Laplacian operator, and G is the 2-D Gaussian function

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

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

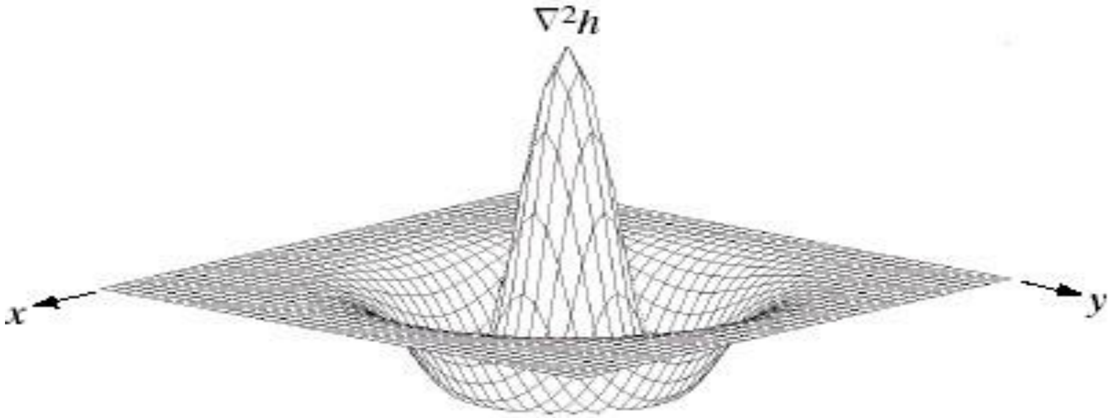
$$\nabla^2 G(x,y) = \frac{\partial}{\partial x}\left(\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right) + \frac{\partial}{\partial y}\left(\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right)$$

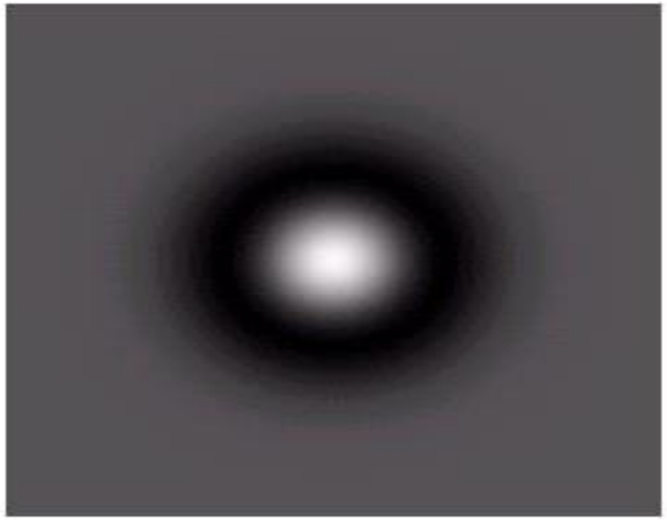
$$\nabla^2 G(x,y) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} + \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

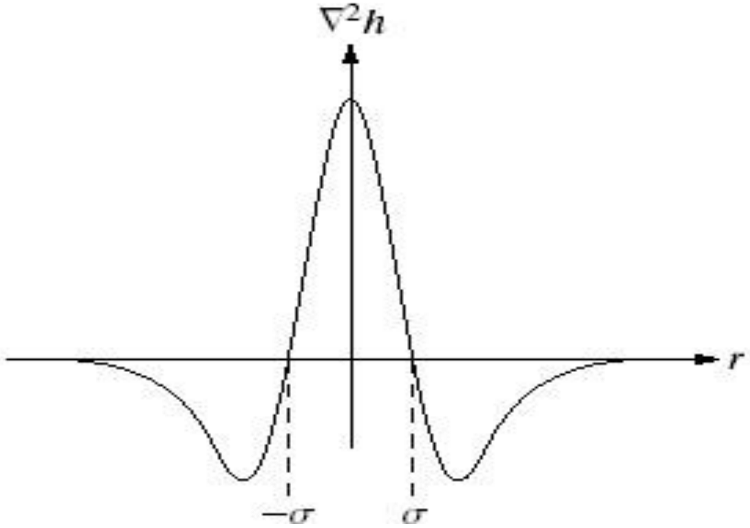
$$\nabla^2 G(x,y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This expression is called Laplacian of a Gaussion (LoG)

More Advanced Techniques for Edge Detection





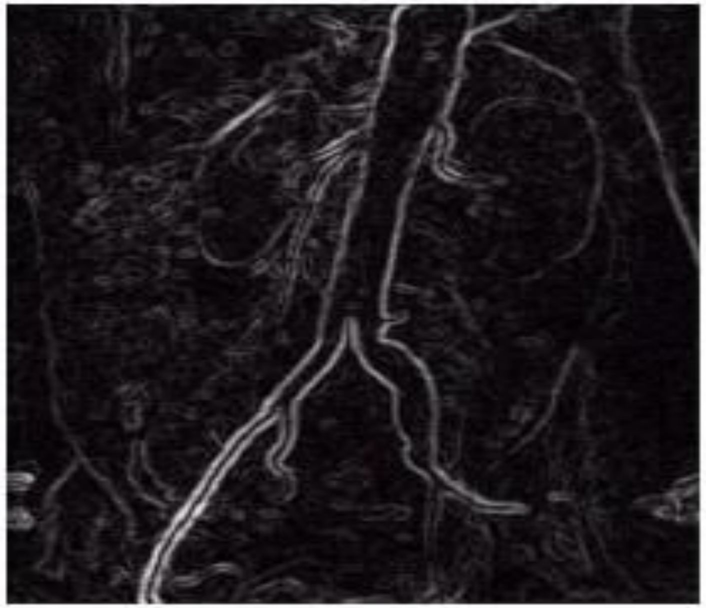


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

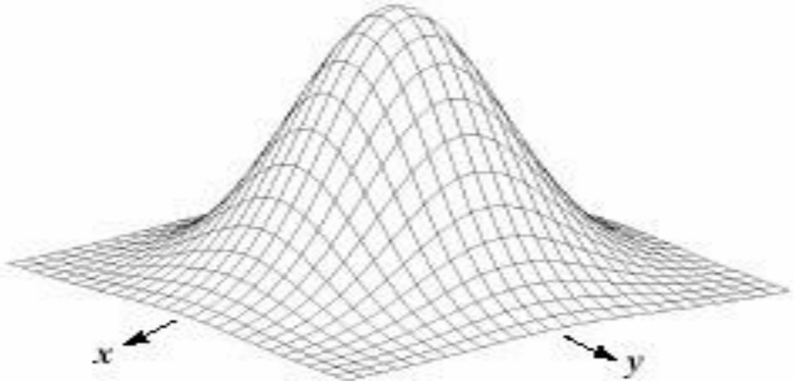
a b
c d

FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5×5 mask approximation to the shape of (a).

Detection of Discontinuities Gradient Operators: Example



Sobel gradient



Gaussian smooth function

–1	–1	–1
–1	8	–1
–1	–1	–1

Laplacian mask

Detection of Discontinuities Gradient Operators: Example

a b
c d
e f g

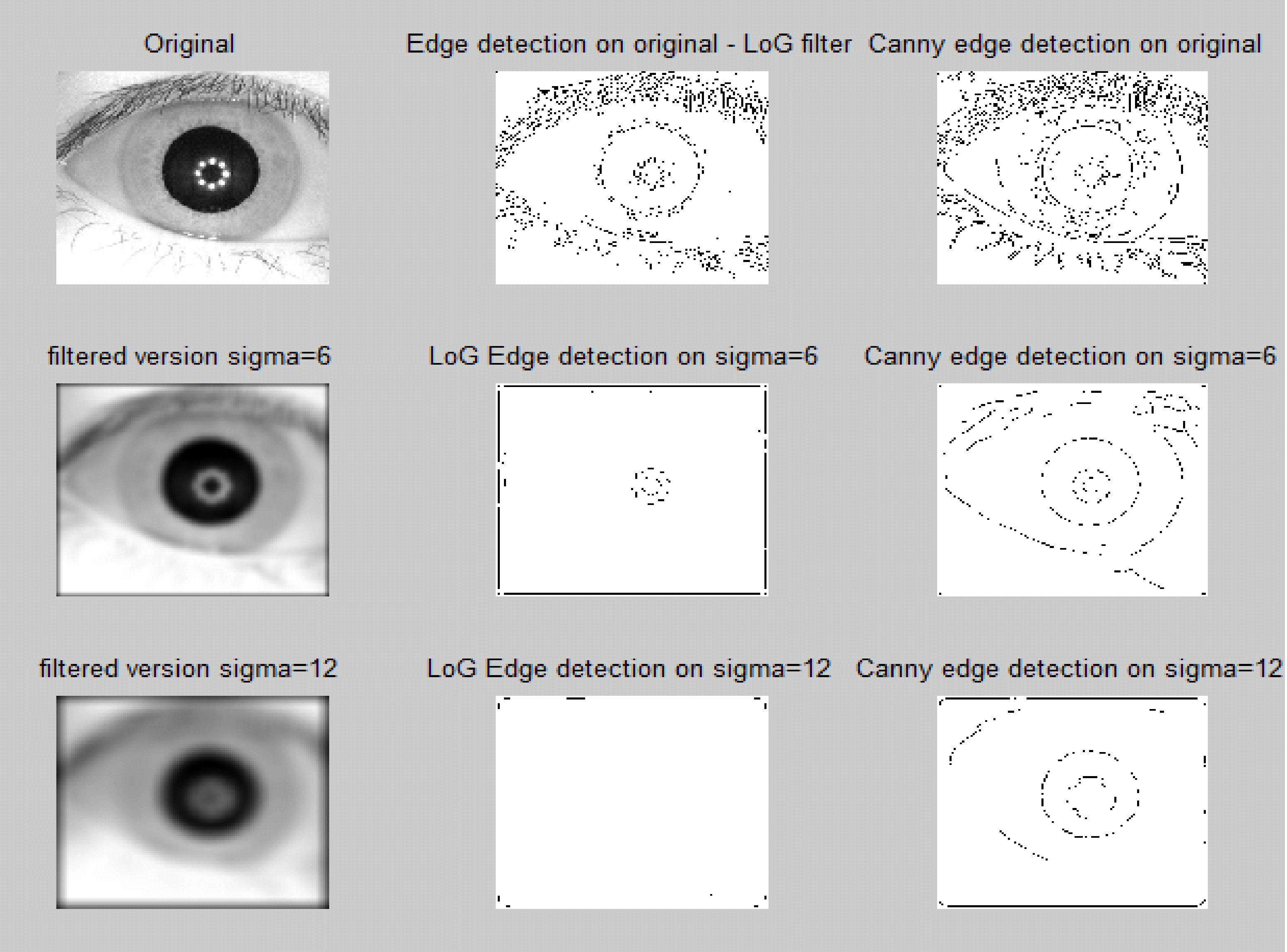
FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Detection of Discontinuities Gradient Operators: Example

a b

FIGURE 10.23
(a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1.
(b) Profiles obtained using a ratio of 1.6:1.

Edge Detection by Canny and LoG



Hough Transforms

Hough Transforms takes the images created by the edge detection operators

Most of the time, the edge map generated by the edge detection algorithms is disconnected

HT can be used to connect the disjointed edge points

It is used to fit the points as plane curves

Plane curves are lines, circles, and parabolas

The line equation is $y = mx + c$

However, the problem is that there are infinite line passing through one points

Therefore, an edge point in an x-y plane is transformed to a c-m plane

Now equation of line is $c = (-x)m + y$

Hough Transforms

All the edge points in the x-y plane need to be fitted
All the points are transformed in c-m plane
The objective is to find the intersection point
A common intersection point indicates that the edges points which are part of the line
If A and B are the points connected by a line in the spatial domain, then they will be intersecting lines in the Hough Space (c-m plane)
To check whether they are intersecting lines, the c-m plane is partitioned as accumulator lines
To find this, it can be assumed that the c-m plane can be partitioned as an accumulator array
For every edge point (x,y), the corresponding accumulator element is incremented in the accumulator array
At the end of this process, the accumulator values are checked
Significance is that this point gives us the line equation of the (x,y) space

Hough Transforms

Hough Transform steps:

1)Load the image

2)Find the edges of the image using any edge detector

3)Quantize the parameter space P

4)Repeat the following for all the pixels of the image:

if the pixel is an edge pixel, then

(a) $c = (-x)m + y$ *or calculate* ρ

(b) $P(c,m) = P(c,m) + 1$ *or increment position in P*

5)Show the Hough Space

6)Find the local maxima in the parameter space

7)Draw the line using the local maxima

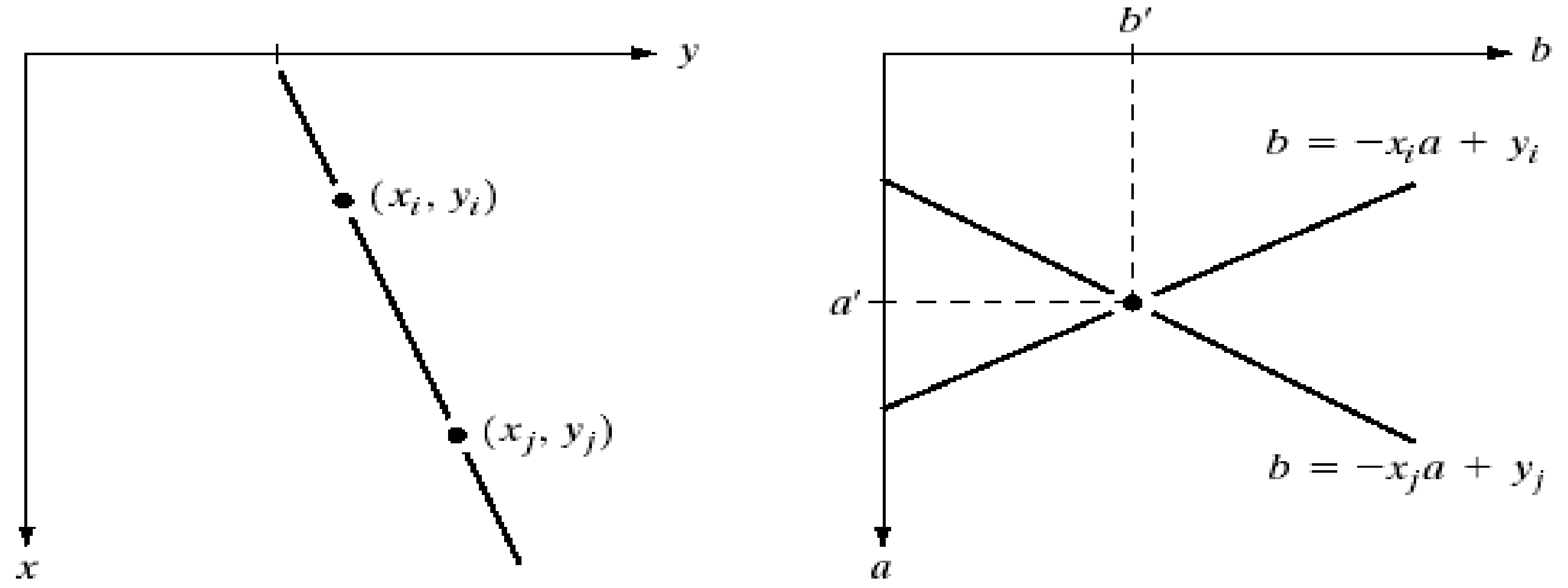
The major problem with this algorithm is that it does not work for vertical lines, as they have a slope of infinity

Convert line into polar coordinates $\rho = x\cos\Theta + y\sin\Theta$, where Θ is the angle between the line and x-axis, and ρ is the diameter

Edge Linking and Boundary Detection
Global Processing via the Hough Transform

- Hough transform: a way of finding edge points in an image that lie along a straight line.
- Example: *xy*-plane v.s. *ab*-plane (parameter space)

$$y_i = ax_i + b$$

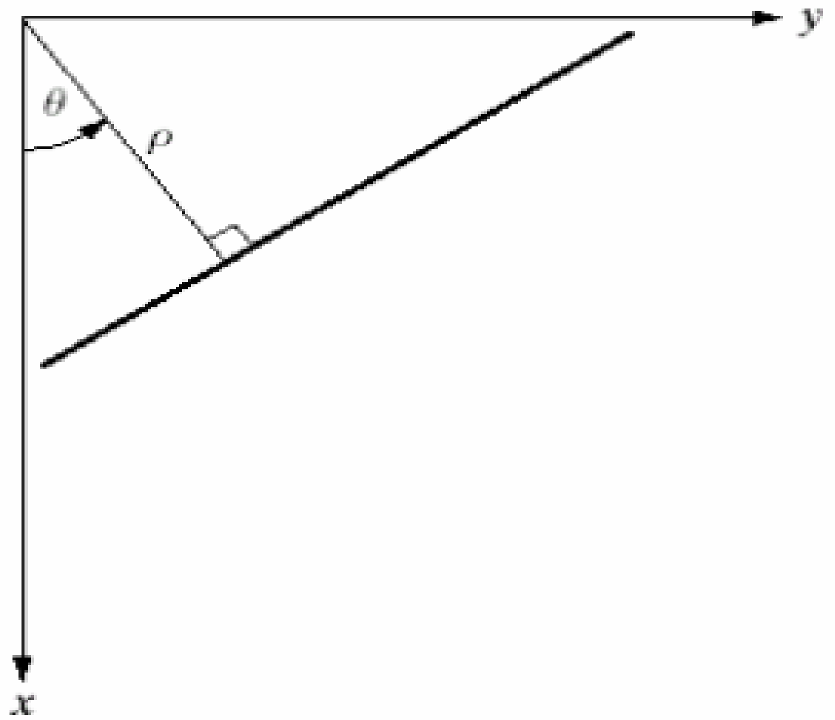


a b

FIGURE 10.17
(a) *xy*-plane.
(b) Parameter space.

Edge Linking and Boundary Detection
Global Processing via the Hough Transform

- The Hough transform consists of finding all pairs of values of θ and ρ which satisfy the equations that pass through (x,y).
- These are accumulated in what is basically a 2-dimensional histogram.
- When plotted these pairs of θ and ρ will look like a sine wave. The process is repeated for all appropriate (x,y) locations.



Edge Linking and Boundary Detection
Hough Transform Example

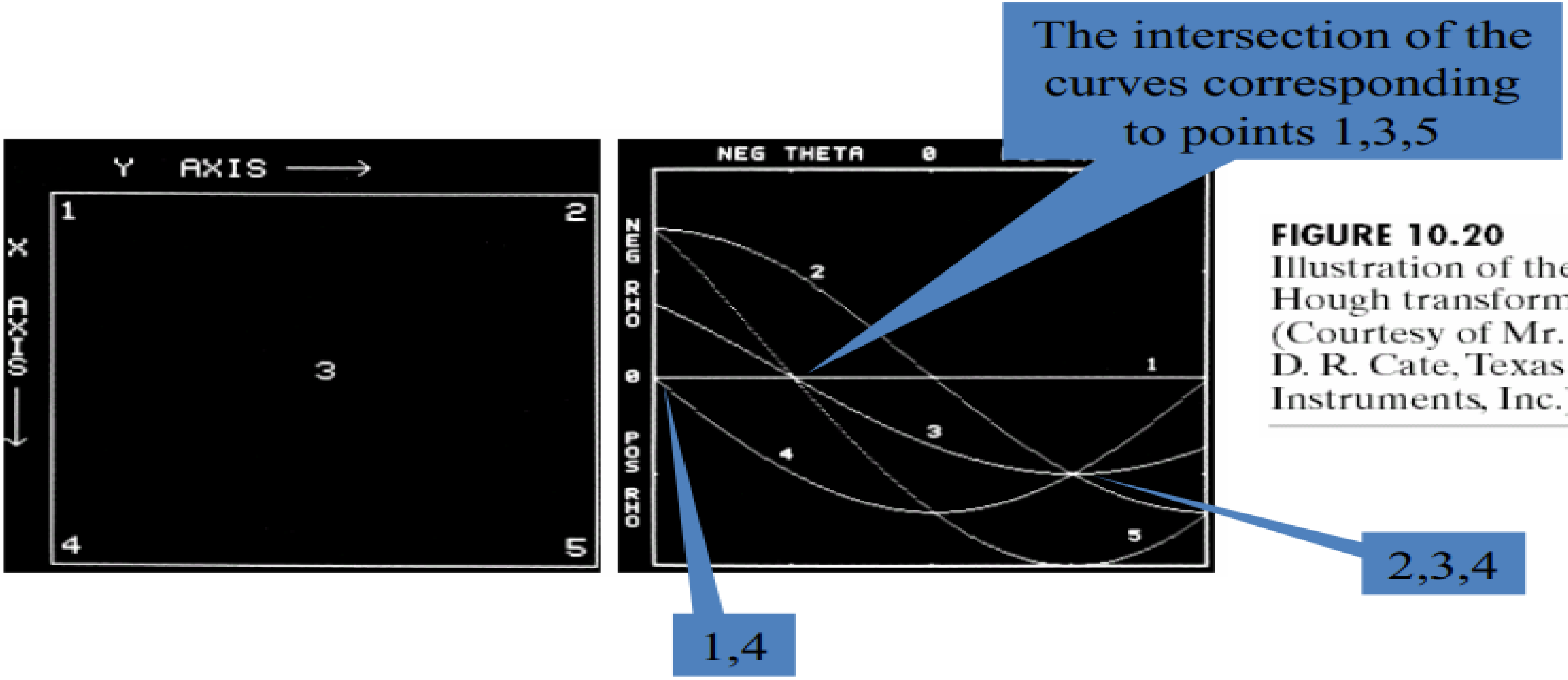
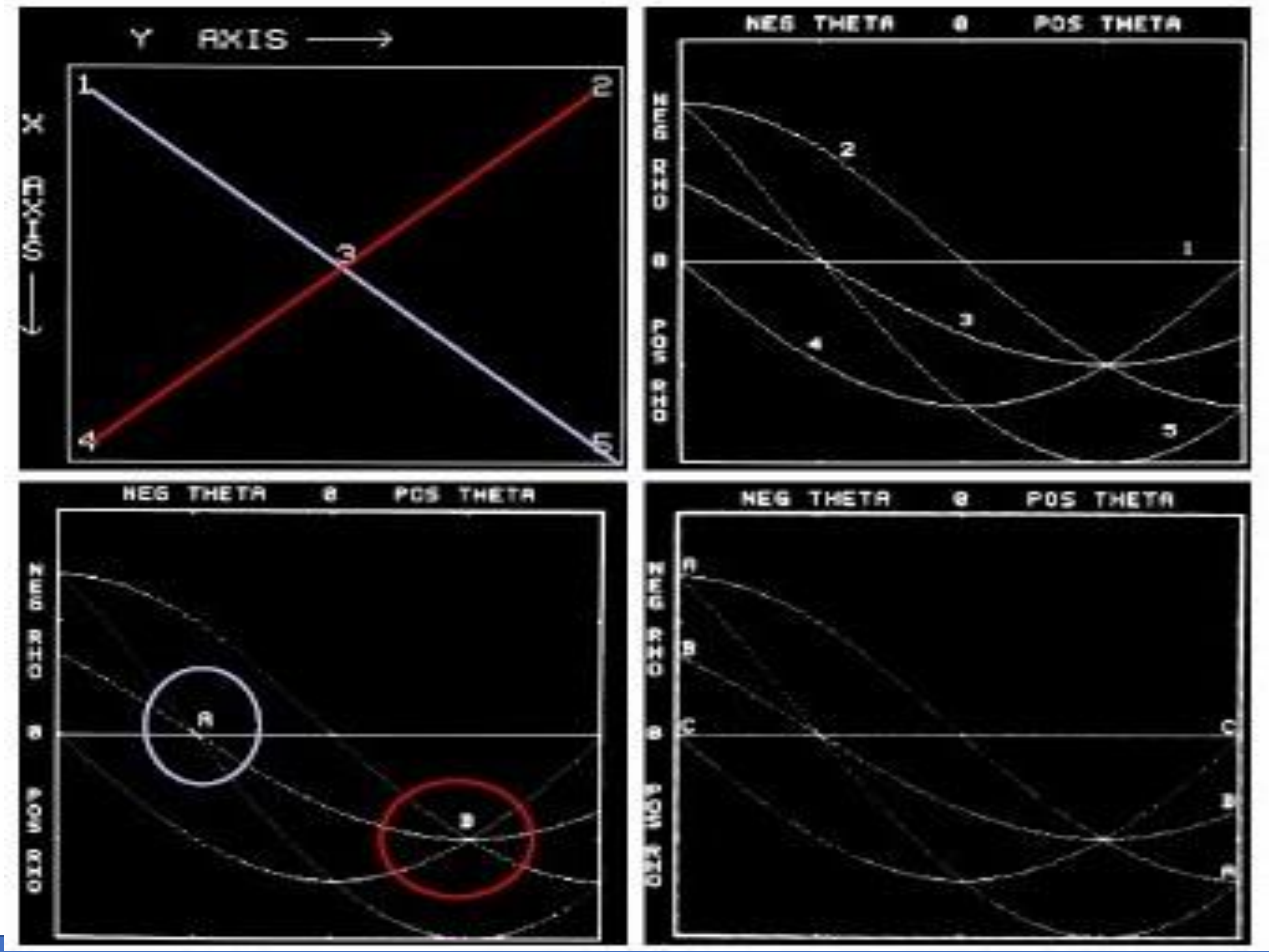
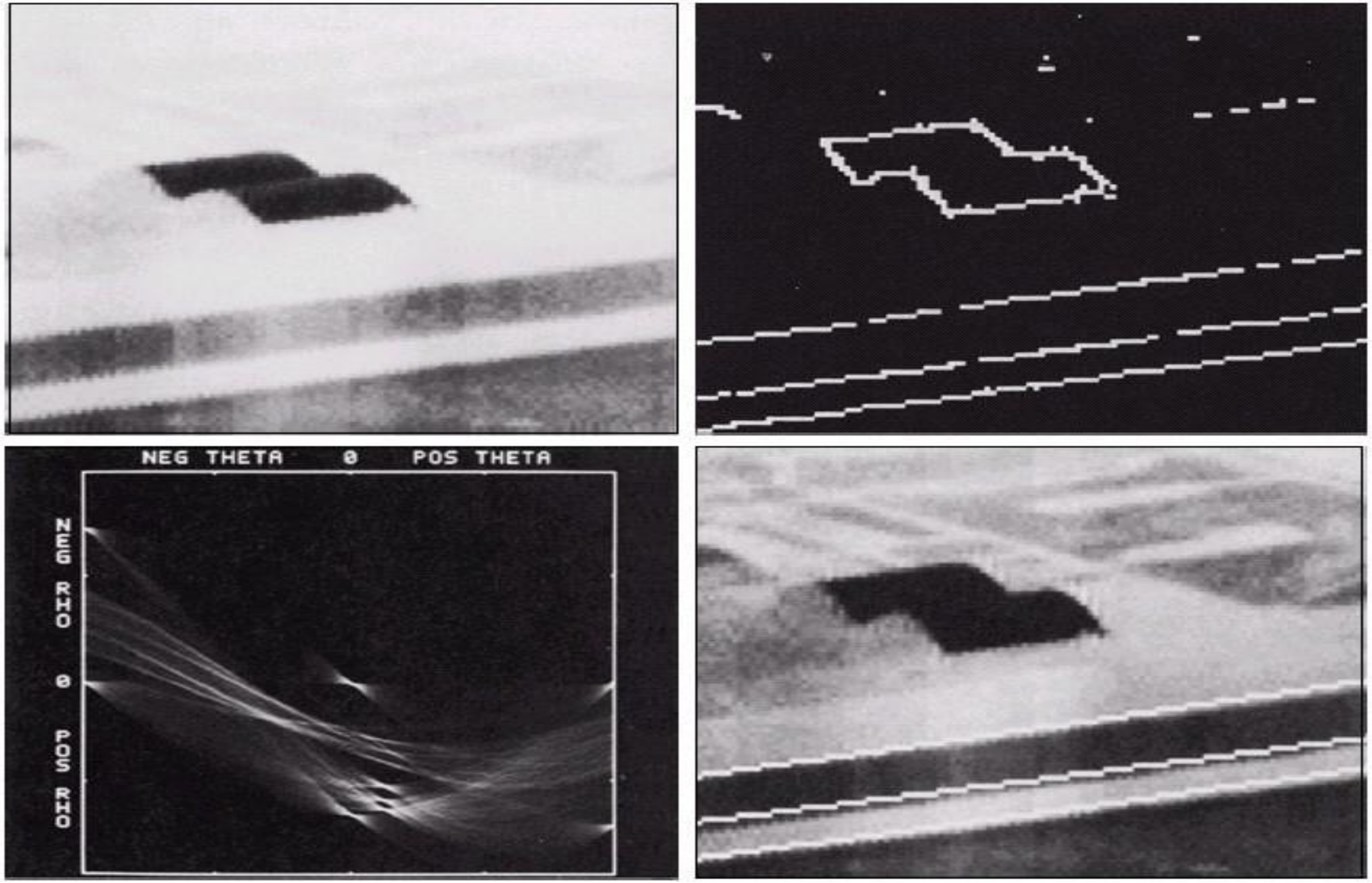


FIGURE 10.20 Illustration of the Hough transform. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)


Y AXIS
X AXIS
1
2
3
4
5
NEG THETA 0 POS THETA
NEG RHO
0
POS RHO
A
B
C

Edge Linking and Boundary Detection Hough Transform Example



a	b
c	d

FIGURE 10.21 (a) Infrared image. (b) Thresholded gradient image. (c) Hough transform. (d) Linked pixels. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Thresholding

- Assumption: the range of intensity levels covered by objects of interest is different from the background.

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

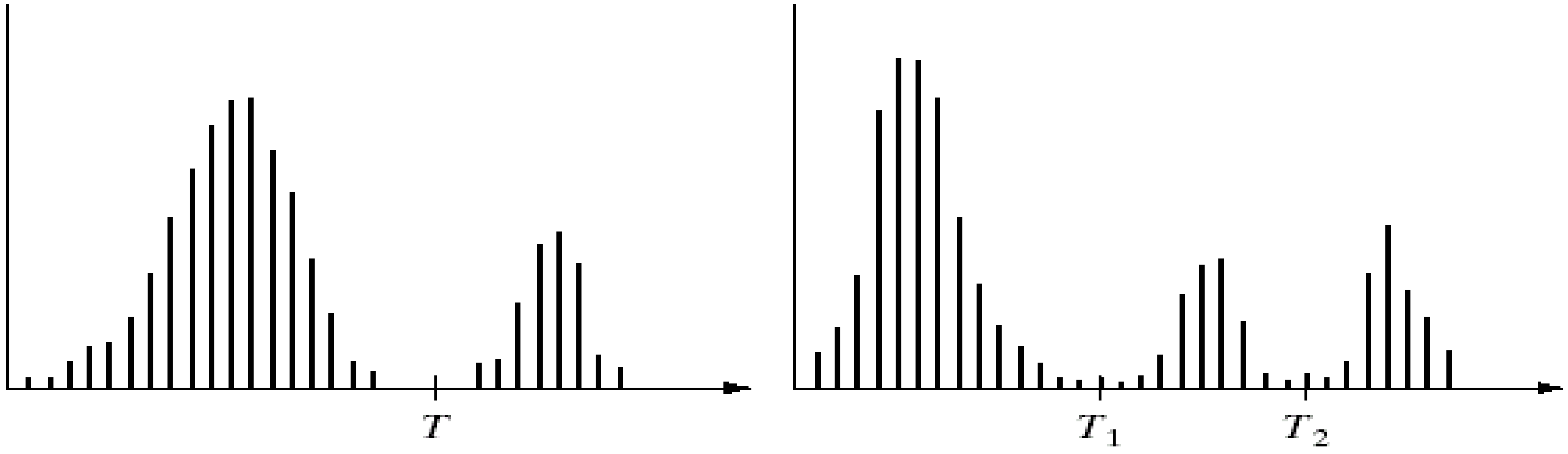
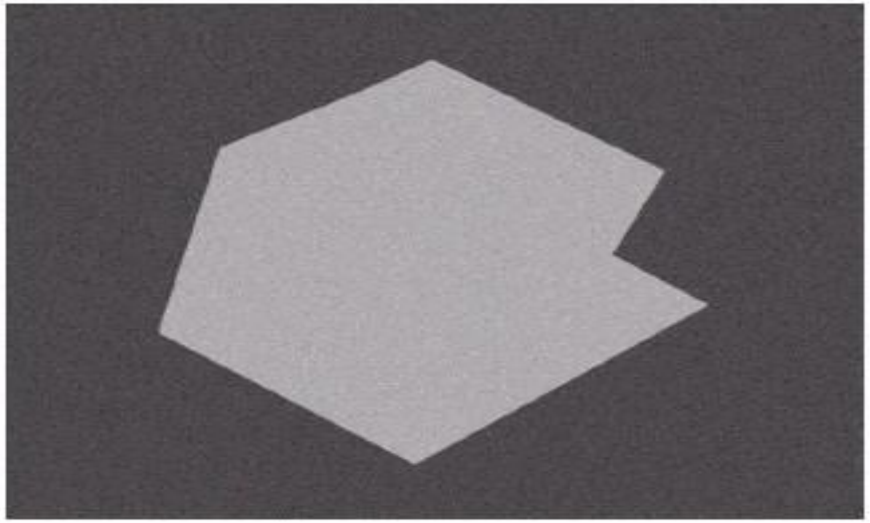


FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

Single threshold Multiple threshold

Thresholding
The Role of Illumination



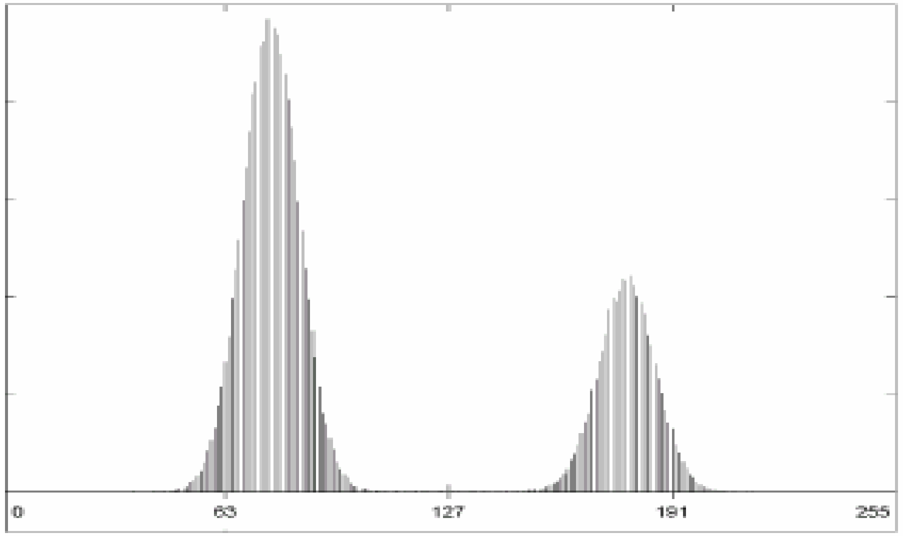
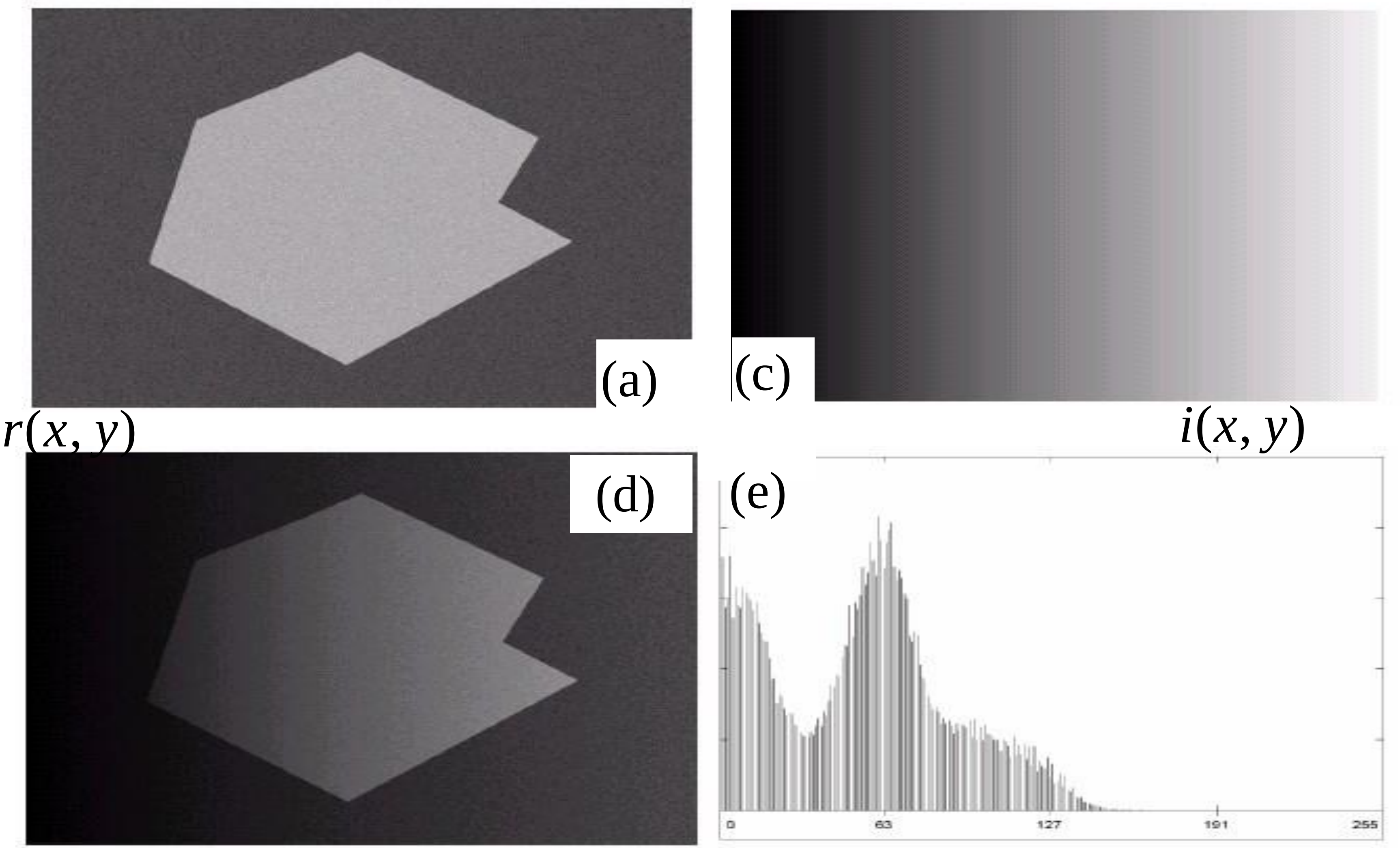


FIGURE 10.27
(a) Computer generated reflectance function.
(b) Histogram of reflectance function.

Thresholding
The Role of Illumination

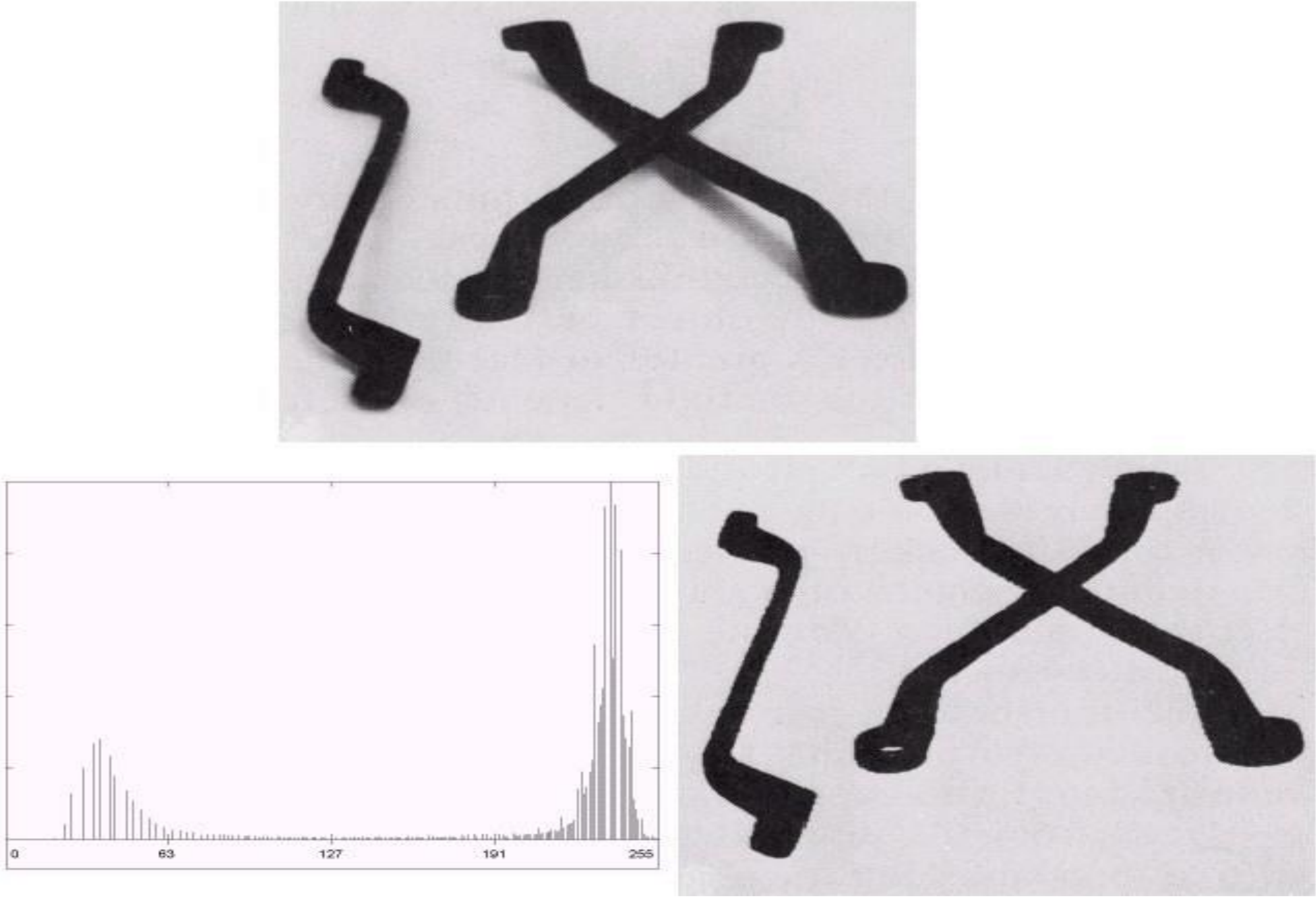


$f(x, y) = i(x, y)r(x,y)$

	a	
	b	c
	d	e

FIGURE 10.27 (a) Computer generated reflectance function. (b) Histogram of reflectance function. (c) Computer generated illumination function. (d) Product of (a) and (c). (e) Histogram of product image.

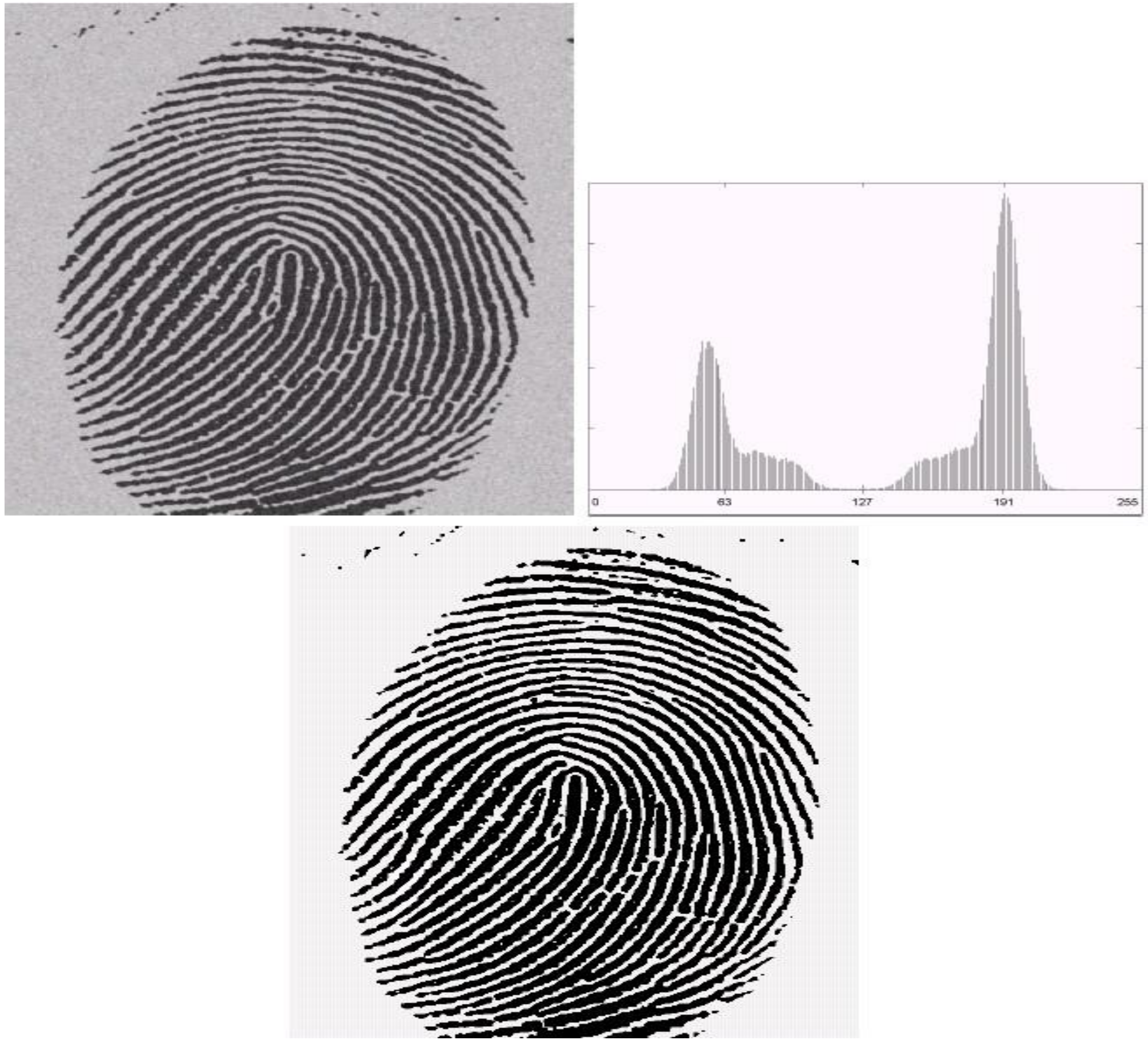
Thresholding:Basic Global Thresholding



a
b c

FIGURE 10.28 (a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

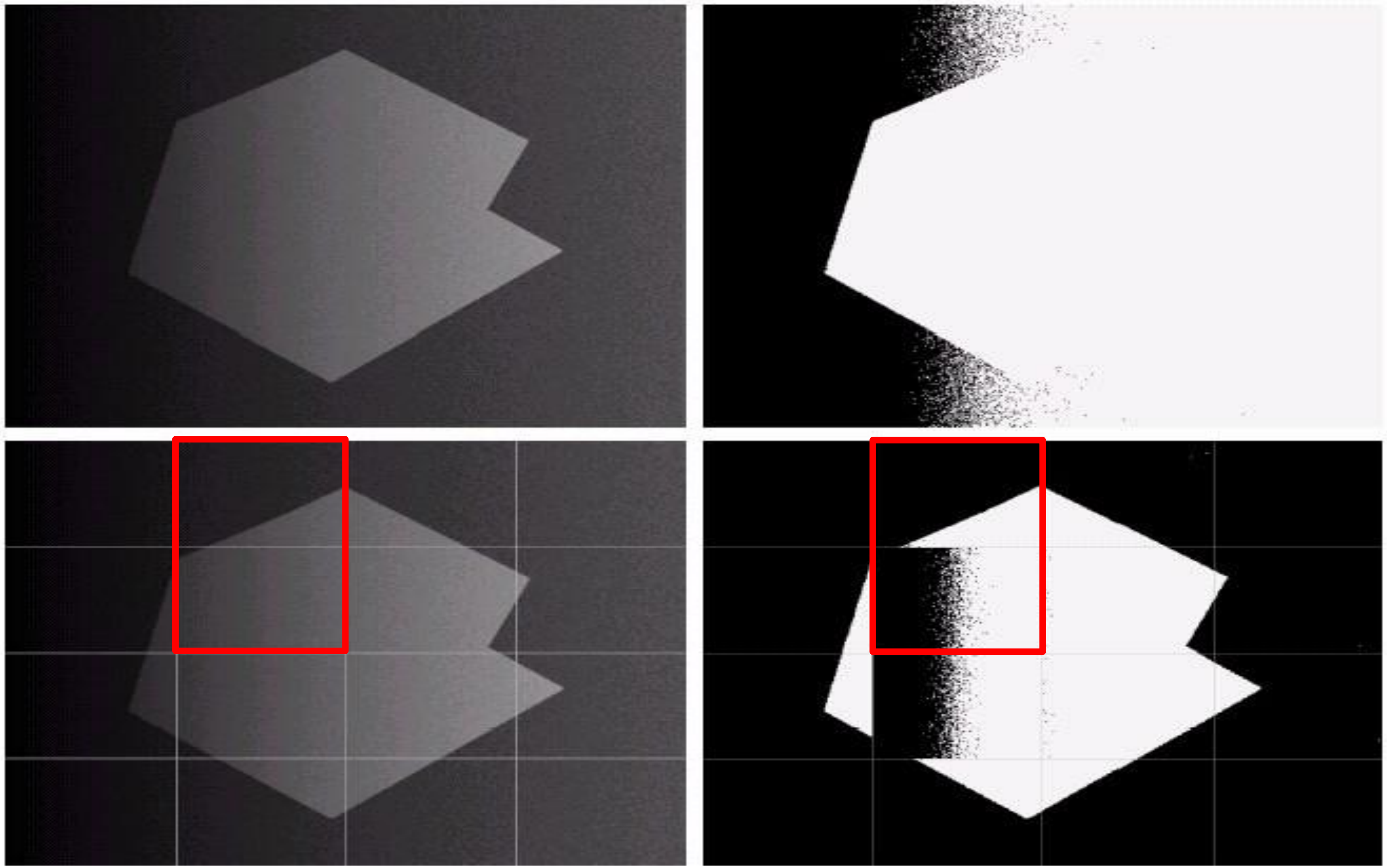
Basic Global Thresholding



a b
c

FIGURE 10.29 (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

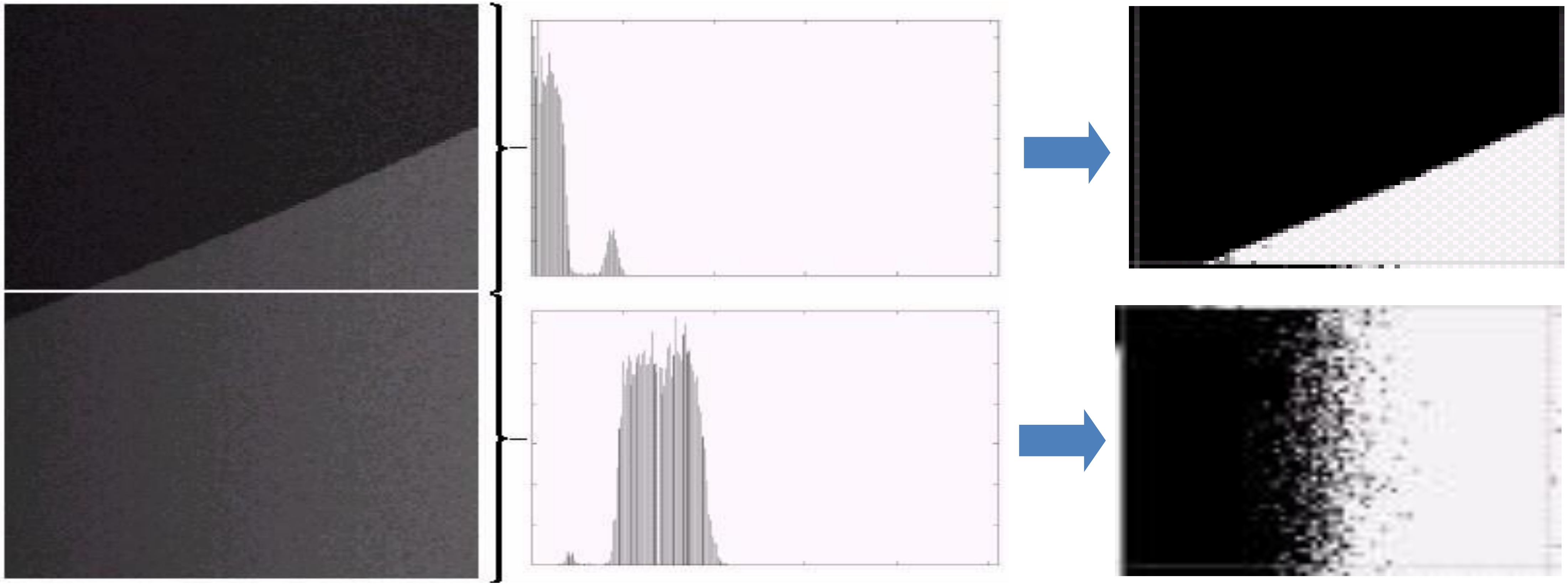
Basic Adaptive Thresholding



a b
c d

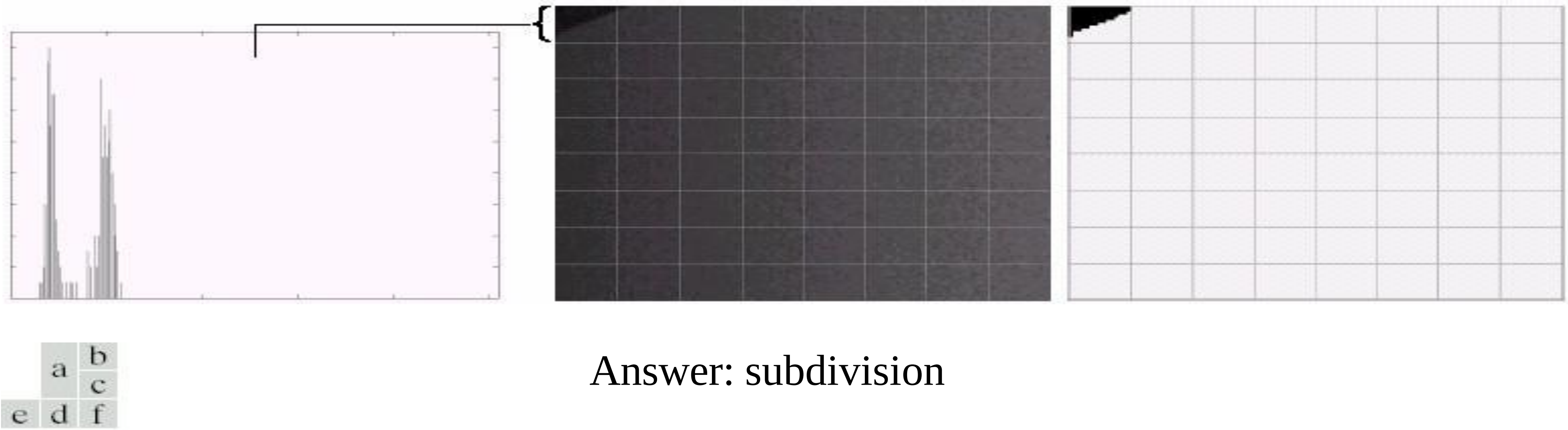
FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

Thresholding
Basic Adaptive Thresholding



How to solve this problem?

Thresholding
Basic Adaptive Thresholding



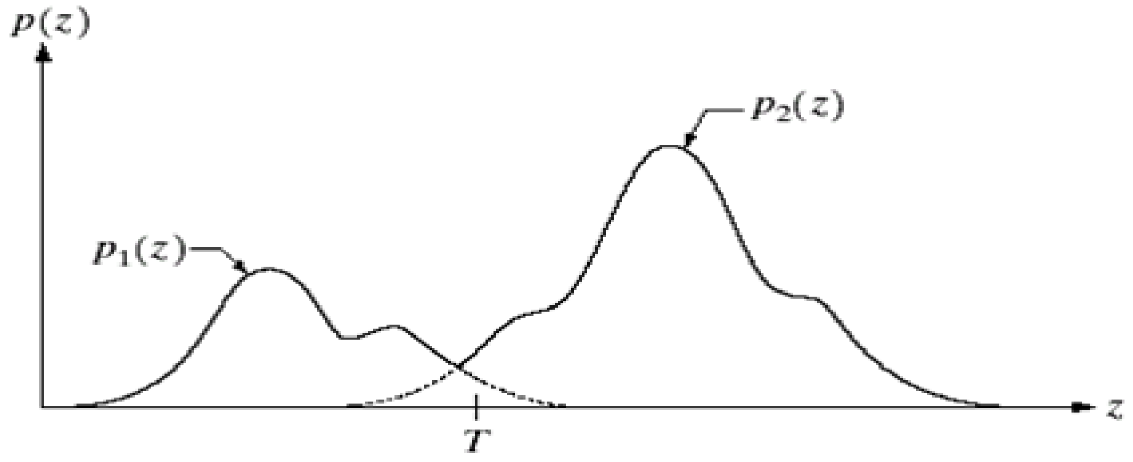
Answer: subdivision

FIGURE 10.31 (a) Properly and improperly segmented subimages from Fig. 10.30. (b)–(c) Corresponding histograms. (d) Further subdivision of the improperly segmented subimage. (e) Histogram of small subimage at top, left. (f) Result of adaptively segmenting (d).

Optimal Global and Adaptive Thresholding

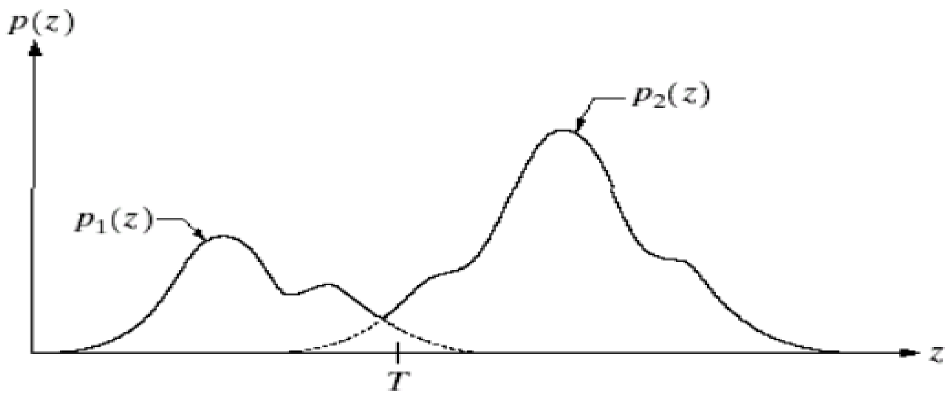
- This method treats pixel values as probability density functions.
- The goal of this method is to minimize the probability of misclassiyfing pixels as either object or background.
- There are two kinds of error:
  - mislabeling an object pixel as background, and
  - mislabeling a background pixel as object.

FIGURE 10.32 Gray-level probability density functions of two regions in an image.



- Method for estimating thresholds that produce the minimum average segmentation error.
- Let an image contains only two principal gray regions.
- Let z represents the gray-level values.
- These can be viewed as random quantities, and the histogram may be considered an estimate of their probability density function (PDF), p(z).

FIGURE 10.32 Gray-level probability density functions of two regions in an image.



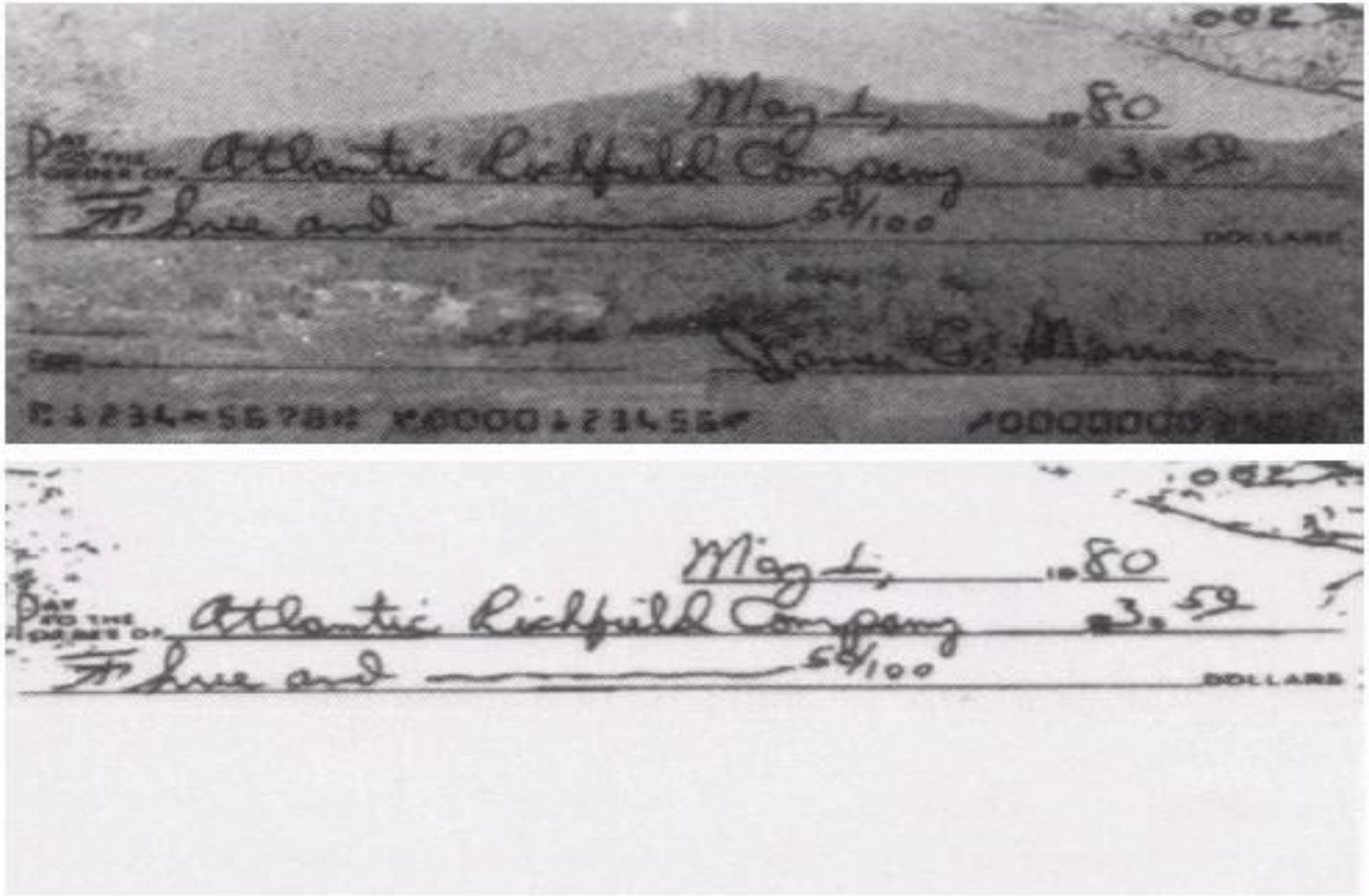
$$p(z) = P_1 p_1(z) + P_2 p_2(z) \qquad P_1 + P_2 = 1$$

P1: probability that a random pixel with value z is an object pixel.
P2: probability that a random pixel Is a background pixel

Thresholding
Use of Boundary Characteristics

a
b

FIGURE 10.37
(a) Original image. (b) Image segmented by local thresholding. (Courtesy of IBM Corporation.)

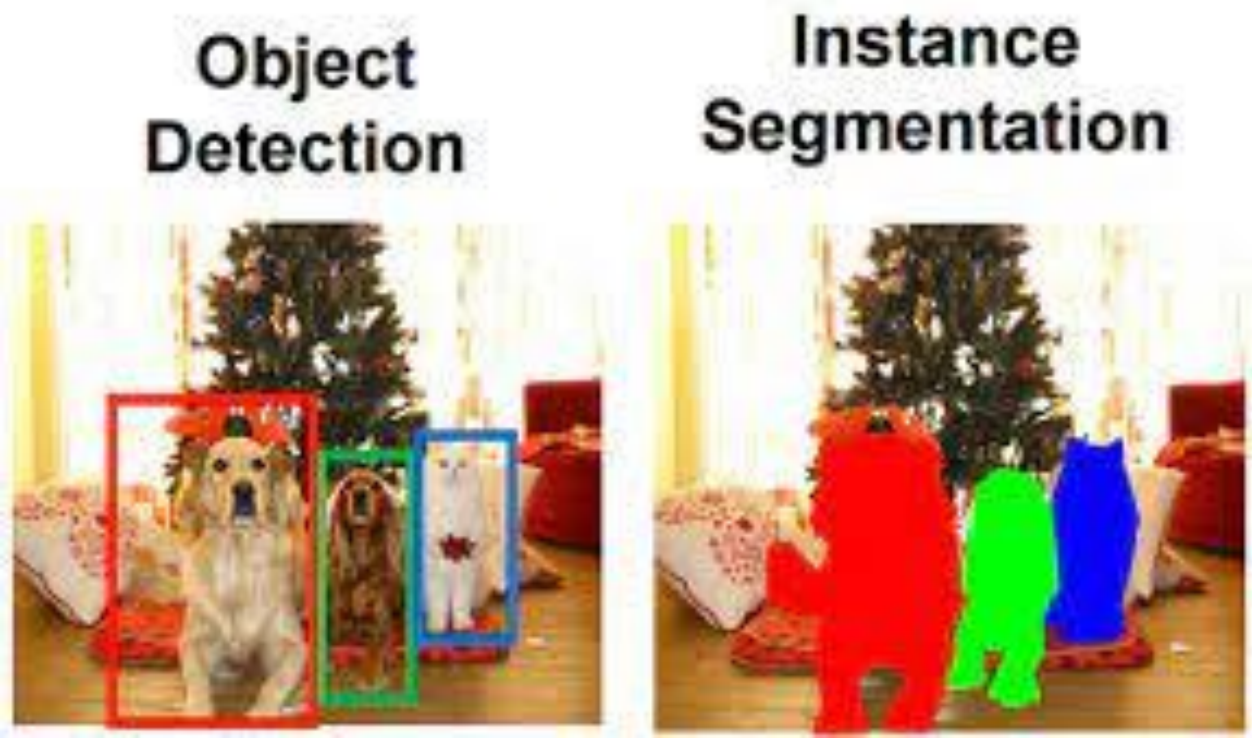


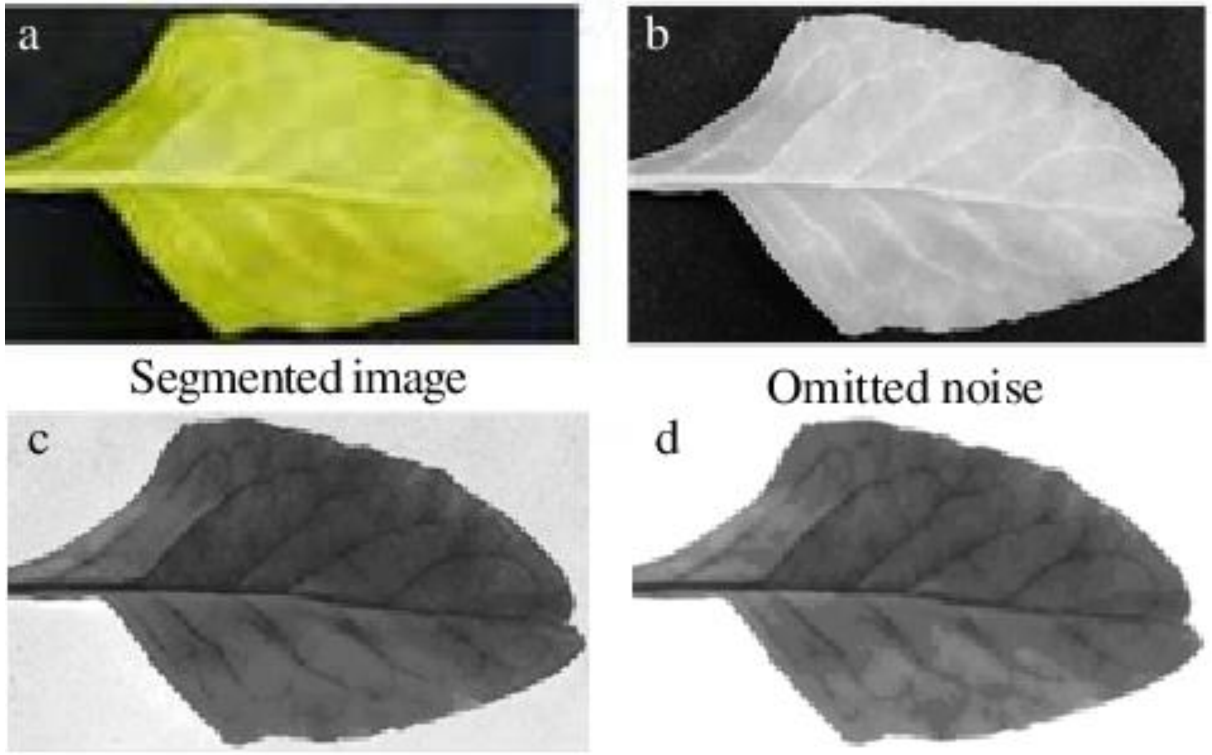
Thresholding
Thresholds Based on Several Variables

Color image

a b c

FIGURE 10.39 (a) Original color image shown as a monochrome picture. (b) Segmentation of pixels with colors close to facial tones. (c) Segmentation of red components.


Object
Detection
Instance
Segmentation


a
b
Segmented image
Omitted noise
c
d

Region-Based Segmentation

- Edges and thresholds sometimes do not give good results for segmentation.
- Region-based segmentation is based on the connectivity of similar pixels in a region.
  - Each region must be uniform.
  - Connectivity of the pixels within the region is very important.
- There are two main approaches to region-based segmentation: region growing and region splitting.

Region-Based Segmentation
Basic Formulation

- Let R represent the entire image region.
- Segmentation is a process that partitions R into subregions, $R_1, R_2, \ldots, R_n$, such that

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

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

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

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

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

where $P(R_k)$: a logical predicate defined over the points in set R_k

For example: $P(R_k)$=TRUE if all pixels in R_k have the same gray level.

Region Growing

- Thresholding still produces isolated image

- Region growing algorithms works on **principle of similarity**

- **It states that a region is coherent if all the pixels of that region are homogeneous with respect to some characteristics such as colour, intensity, texture, or other statistical properties**

- Thus idea is to pick a pixel inside a region of interest as a starting point (also known as a **seed point**) and allowing it to grow

- **Seed point is compared with its neighbours, and if the properties match , they are merged together**

- This process is repeated till the regions converge to an extent that no further merging is possible

Region Growing Algorithm

- It is a process of grouping the pixels or subregions to get a bigger region present in an image

- **Selection of the initial seed:** Initial seed that represent the ROI should be given typically by the user. Can be chosen automatically. The seeds can be either single or multiple

- **Seed growing criteria:** Similarity criterion denotes the minimum difference in the grey levels or the average of the set of pixels. Thus, the initial seed 'grows' by adding the neighbours if they share the same properties as the initial seed

- **Terminate process:** If further growing is not possible then terminate region growing process

Region Growing Algorithm

- Consider image shown in figure:

1	0	7	8	7
0	1	8	9	8
0	0	7	9	8
0	1	8	8	9
1	2	8	8	9

- Assume seed point indicated by underlines. Let the seed pixels 1 and 9 represent the regions C and D, respectively

- Subtract pixel from seed value

- If the difference is less than or equal to 4 (i.e. T=4), merge the pixel with that region. Otherwise, merge the pixel with the other region.

Split and Merge Algorithm

- Region growing algorithm is slow
- So seed point can be extended to a seed region
- Instead of a single pixel, a node of a Regional adjacency graph (RAG) a region itself is now considered as a starting point.
- The split process can be stated as follows:

1)Segment the image into regions R1, R2,….Rn using a set of thresholds 2)Create RAG. Use a similarity measure and formulate a homogeneity test

3) The homogeneity test is designed based on the similarity criteria such as intensity or any image statistics

4)Repeat step 3 until no further region exits that requires merging

Split and Merge Algorithm

•

2	2	2	2	2
2	8	8	8	1
2	8	8	8	1
2	1	1	1	1
2	1	1	1	1

2	2	2	2	2
2	8	8	8	1
2	8	8	8	1
2	1	1	1	1
2	1	1	1	1

8	8	8
8	8	8

Region-Based Segmentation Region Growing

- Fig. 10.41 shows the histogram of Fig. 10.40 (a). It is difficult to segment the defects by thresholding methods. (Applying region growing methods are better in this case.)

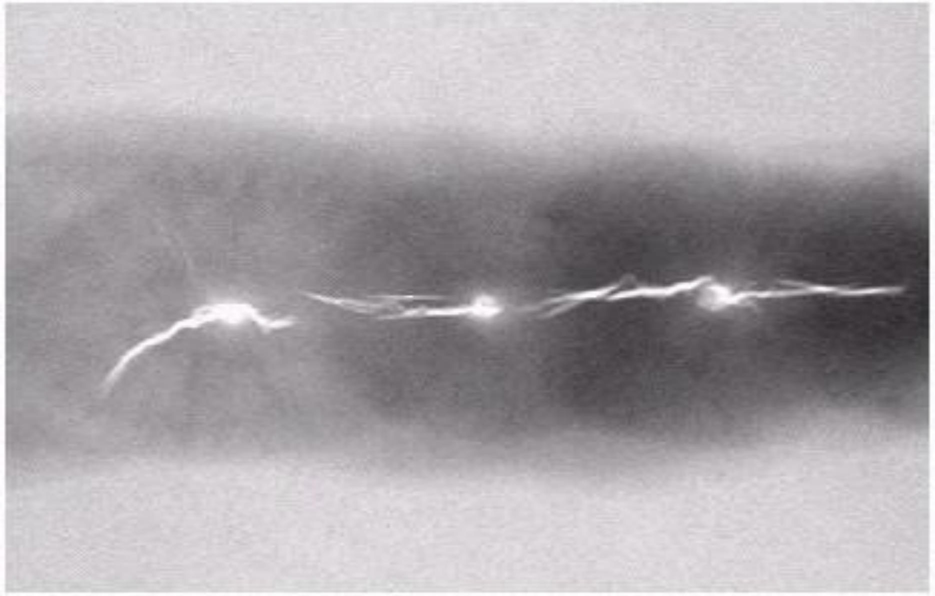


Figure 10.40(a)

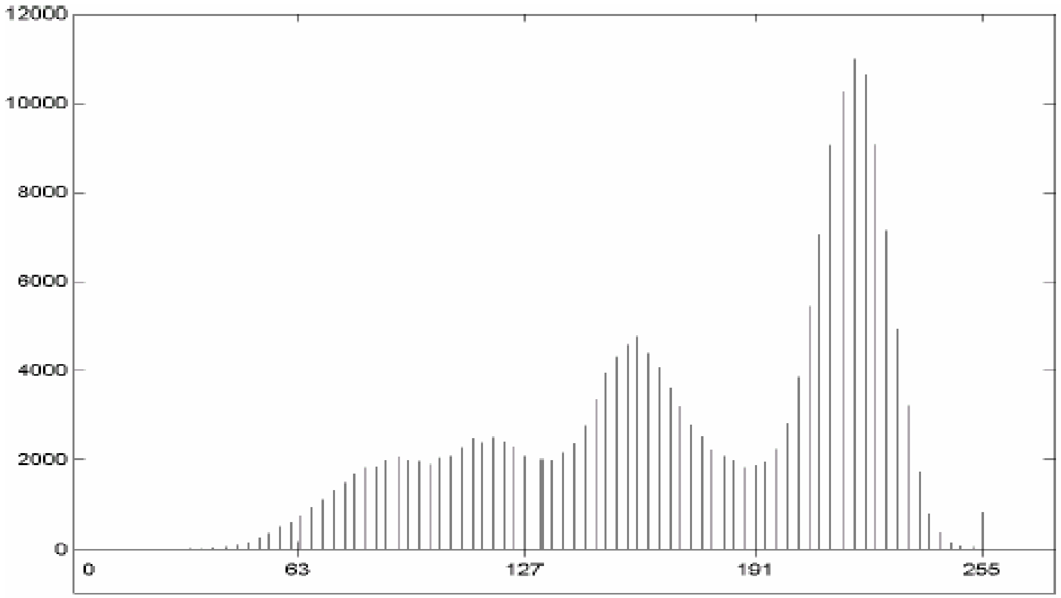


Figure 10.41

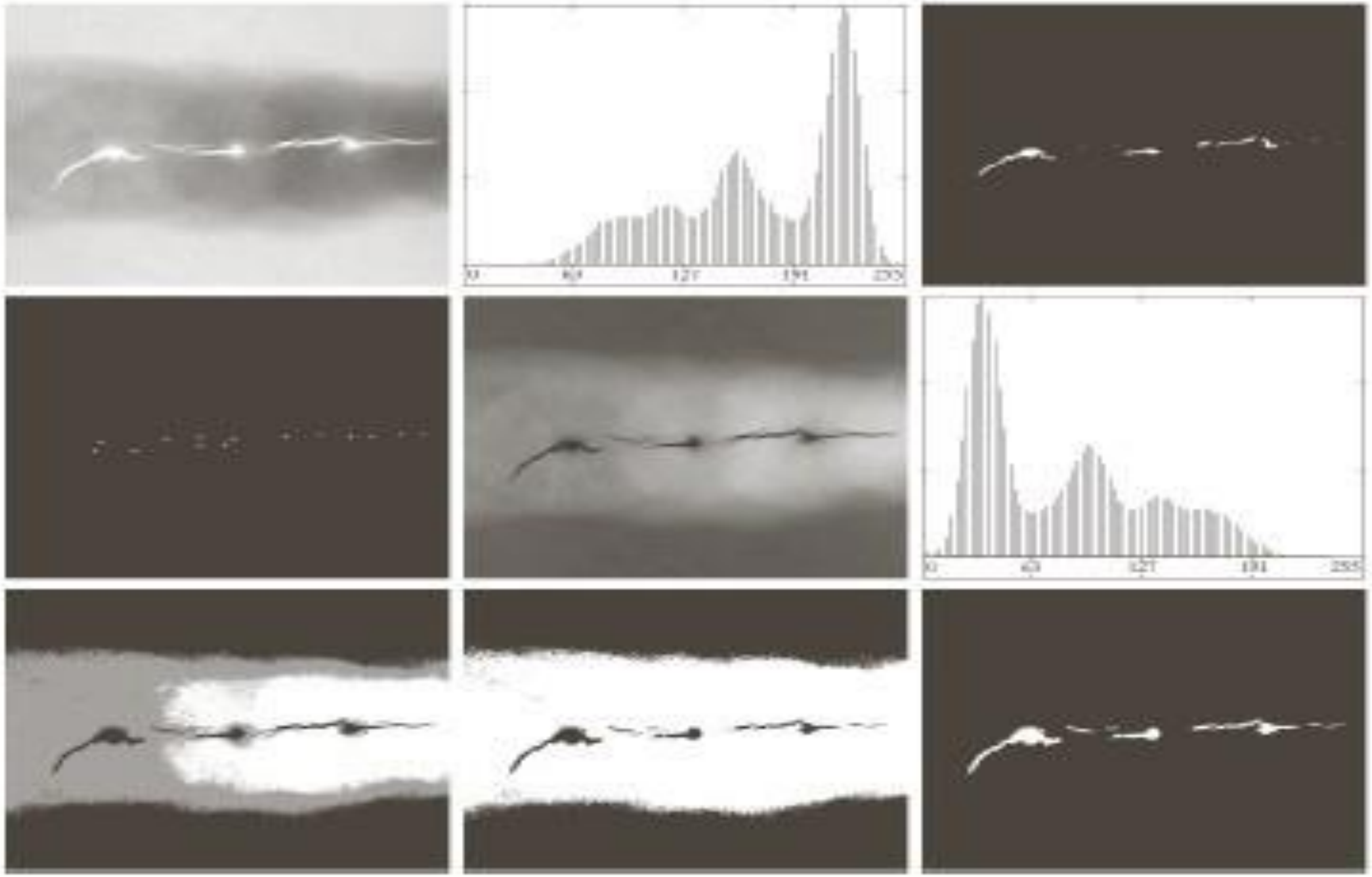


FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (e). (g) Difference image thresholded using dual thresholds. (h) Difference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

Split and Merge using Quadtree

- Entire image is assumed as a single region. Then the homogeneity test is applied. If the conditions are not met, then the regions are split into four quadrants.

- This process is repeated for each quadrant until all the regions meet the required homogeneity criteria. If the regions are too small, then the division process is stopped.

- 1) Split and continue the subdivision process until some stopping criteria is fulfilled. The stopping criteria often occur at a stage where no further splitting is possible.
- 2) Merge adjacent regions if the regions share any common criteria. Stop the process when no further merging is possible

Region-Based Segmentation
Region Splitting and Merging

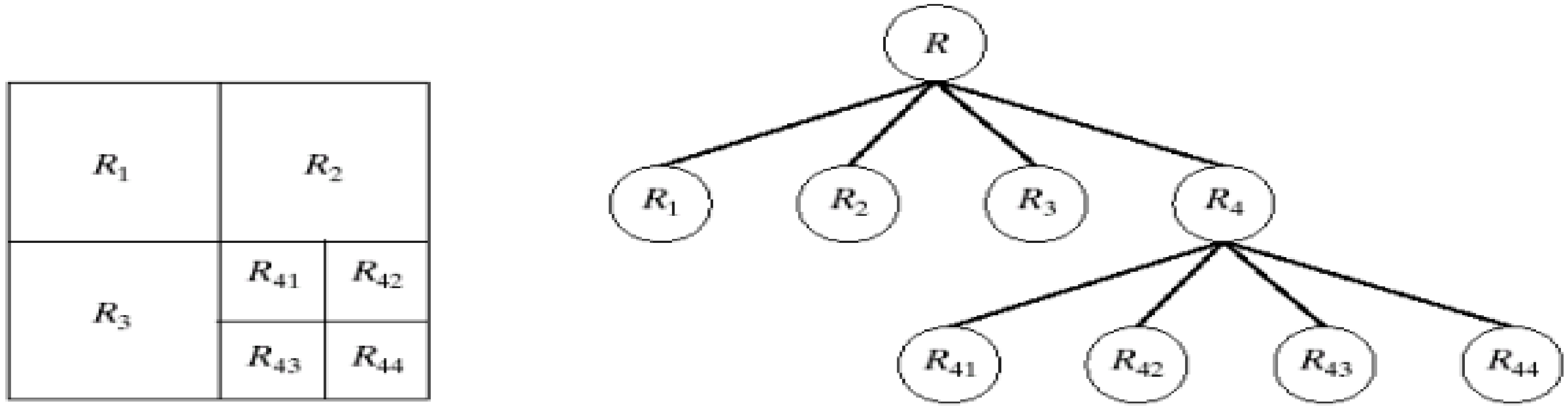
- Region splitting is the opposite of region growing.
  - First there is a large region (possible the entire image).
  - Then a predicate (measurement) is used to determine if the region is uniform.
  - If not, then the method requires that the region be split into two regions.
  - Then each of these two regions is independently tested by the predicate (measurement).
  - This procedure continues until all resulting regions are uniform.

Region-Based Segmentation
Region Splitting

- The main problem with region splitting is determining where to split a region.
- One method to divide a region is to use a quadtree structure.
- Quadtree: a tree in which nodes have exactly four descendants.

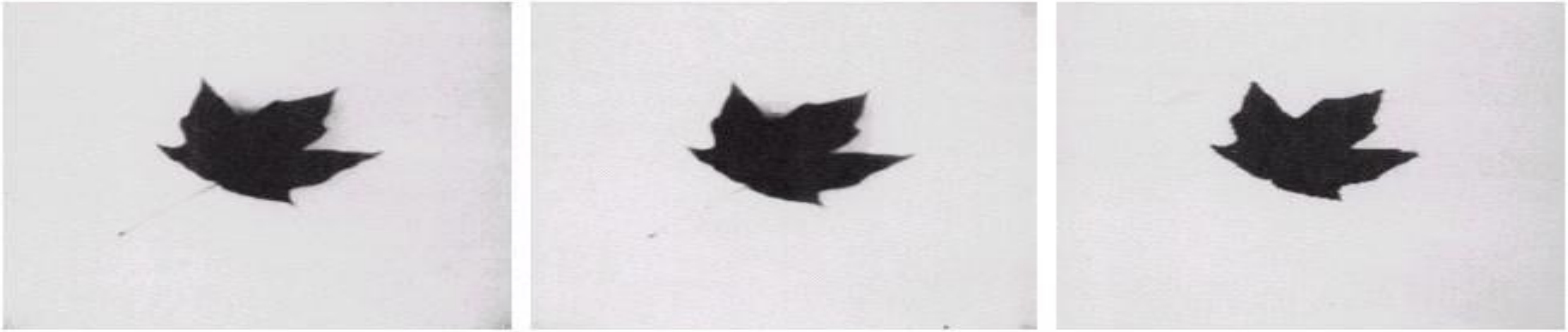
a b

FIGURE 10.42
(a) Partitioned image.
(b) Corresponding quadtree.



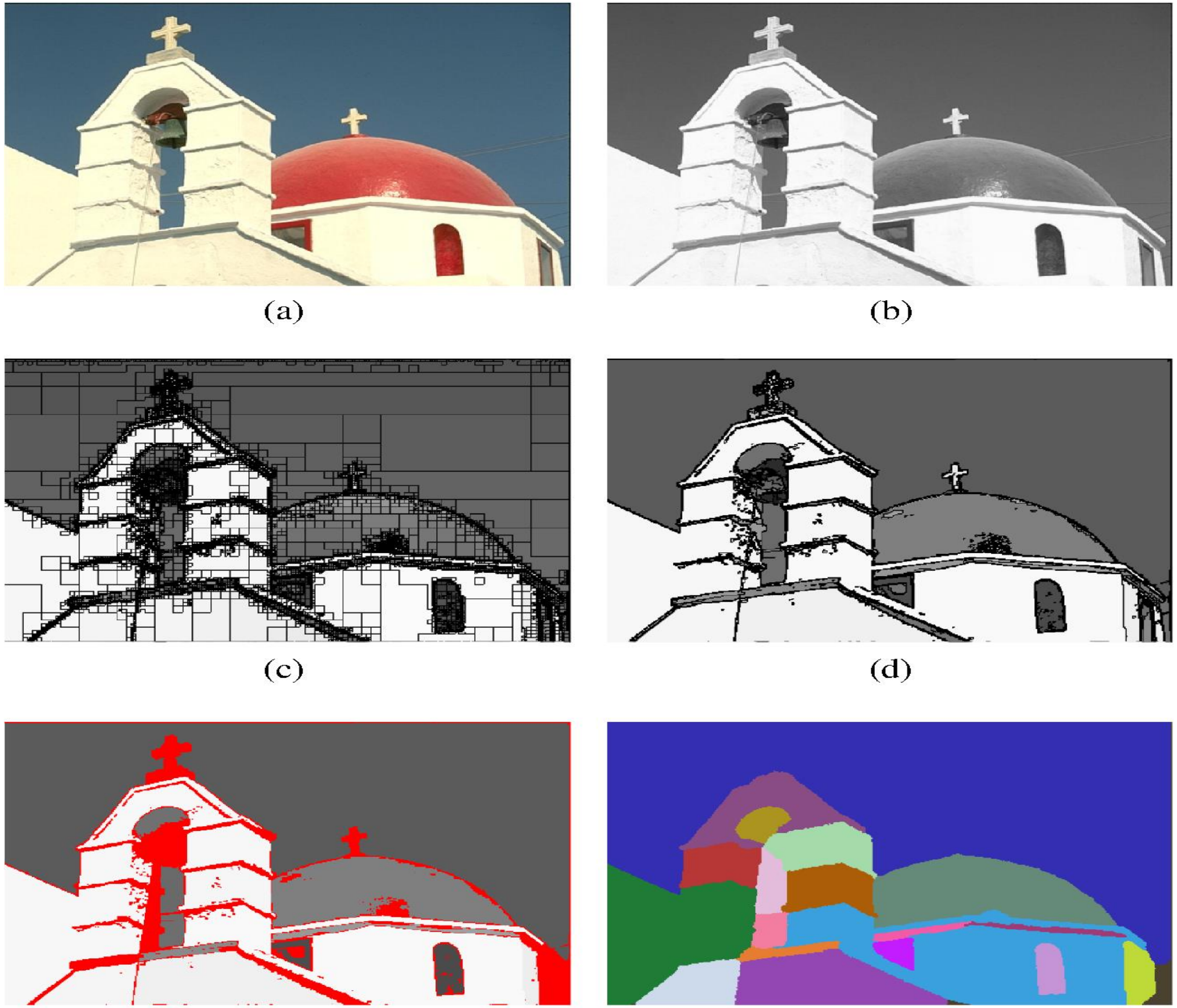
Region-Based Segmentation
Region Splitting and Merging

- The split and merge procedure:
  - Split into four disjoint quadrants any region R_i for which $P(R_i)$ = FALSE.
  - Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k)$ = TRUE. (the quadtree structure may not be preserved)
  - Stop when no further merging or splitting is possible.



a b c

FIGURE 10.43 (a) Original image. (b) Result of split and merge procedure. (c) Result of thresholding (a).



(a) (b) (c) (d) (e) (f)

The End