48

Image Segmentation fundamentals in Digital Image Processing

❖ Introduction

❑ What is 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.

❖ Image segmentation fundamentals

- Segmentation subdivides an image into its constituent regions or objects, until the objects of interest in an application have been isolated.
- Segmentation partitions image R into subregions R_1, R_2, ..., R_n such that:

✓ $R_1 \cup R_2 \cup \ldots \cup R_n = R$

✓ Each R_i is a connected set, i = 1,2,.....,n

✓ $R_i \cap R_j = \emptyset$ for all i, j where i # j

✓ $Q(R_i)$=TRUE for every i

✓ $Q(R_i \cup R_j)$=FALSE for any two adjacent regions

- Segmentation problem: to partition the image into regions satisfying above conditions.

❖ Two principal approaches:

- ❑ **Edge-based Segmentation**
  - Partition an image based on abrupt changes in intensity(edges).
- ❑ **Region-based Segmentation**
  - Partition an image into regions that are similar according to a set of pre-defined criteria.

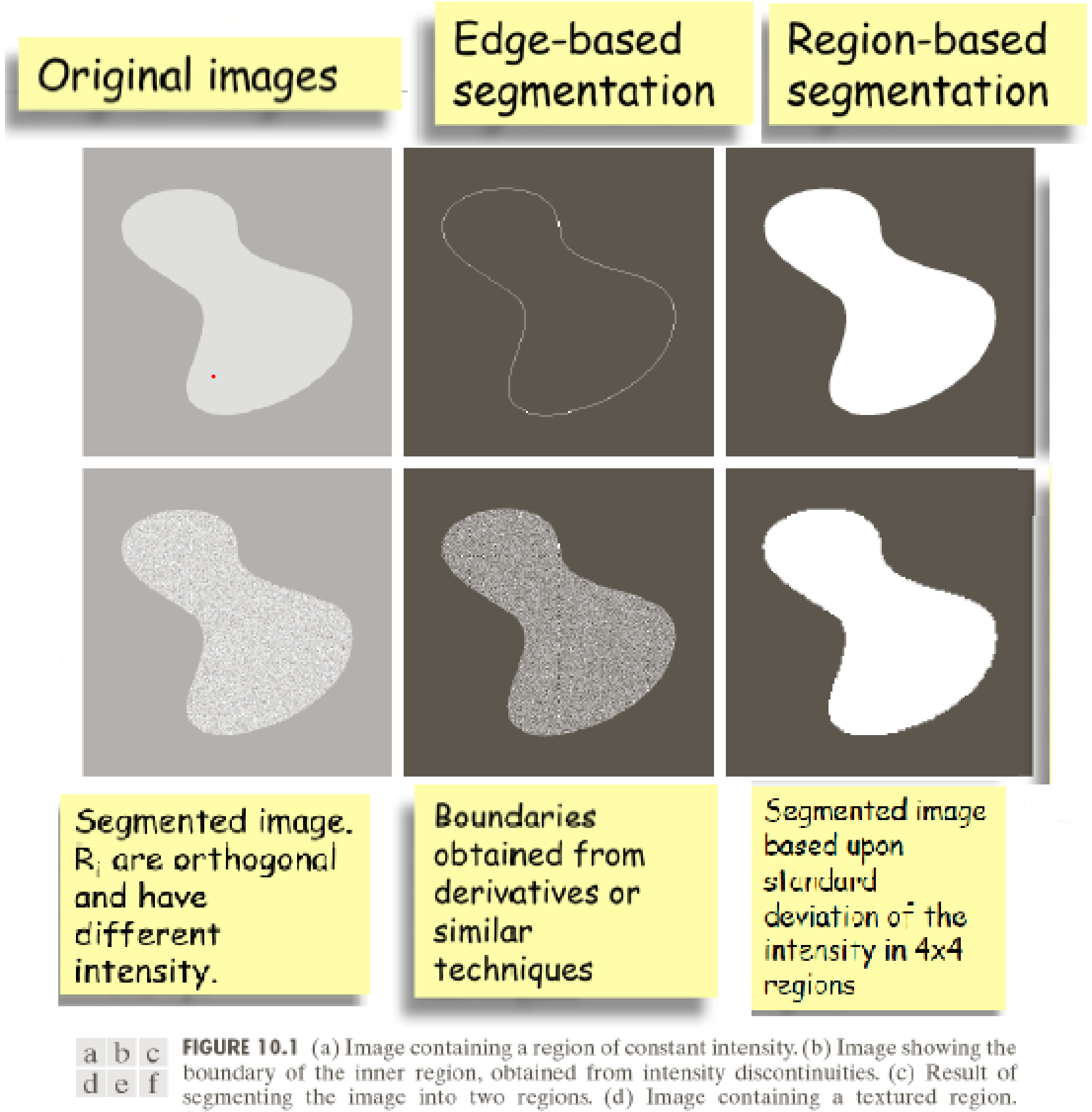


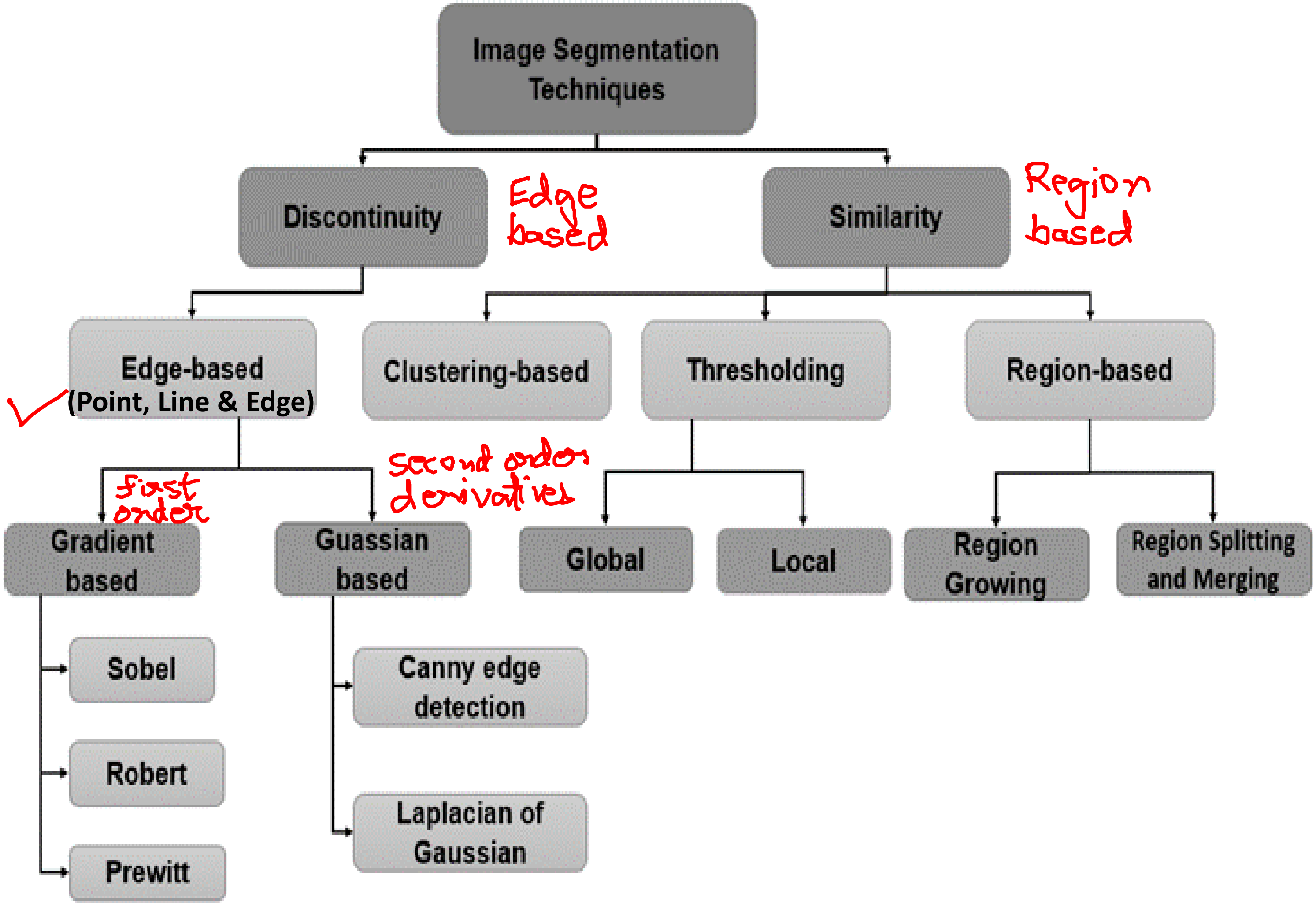
FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number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

Thank You

49

Point, Line and Edge detection in DIP and its implementation in MATLAB

❑ Image Segmentation:



For the most part there are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.

❖ **Discontinuity**

The strategy is to partition an image based on abrupt changes in intensity

➢ Detection of gray level discontinuities:

- Point detection
- Line detection
- Edge detection
  - Gradient operators
  - Gaussian based operators

❖ **Similarity**

The strategy is to partition an image into regions that are similar according to a set of predefined criteria.

❑ Detection of Discontinuities:

- Detect the three basic types of gray-level discontinuities
  – points , lines , edges
- Use the image sharpening techniques
  – The first order derivatives produce thicker edges
  – The second order derivatives(Laplacian operation) have a strong response to fine detail, such as thin lines and isolated points, and noise.

- Can be done by running a mask through the image

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

❑ Point detection:

- Steps for point detection:

1. Apply Laplacian filter to the image to obtain R(x, y).

2. Create binary image by threshold.

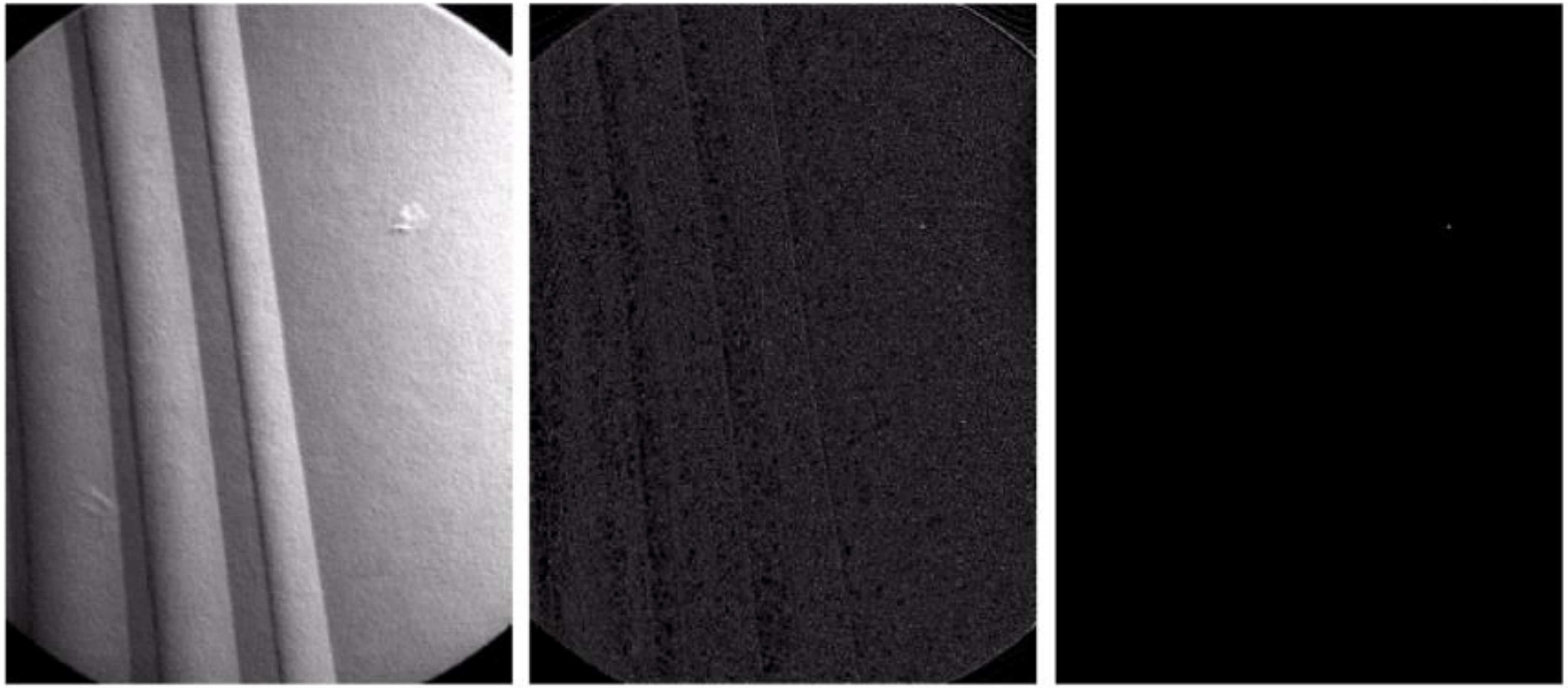
$$g(x,y)=\begin{cases}1, & \text{if } |R(x,y)|\geq T\\ 0, & \text{otherwise}\end{cases}$$

where T is a non-negative threshold.

Laplacian mask

−1	−1	−1
−1	8	−1
−1	−1	−1

−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.)

❑ Line detection:

- A special mask is needed to detect a special type of line.
- Examples:

  - Horizontal mask has high response when a line is passed through the middle row of the mask.

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

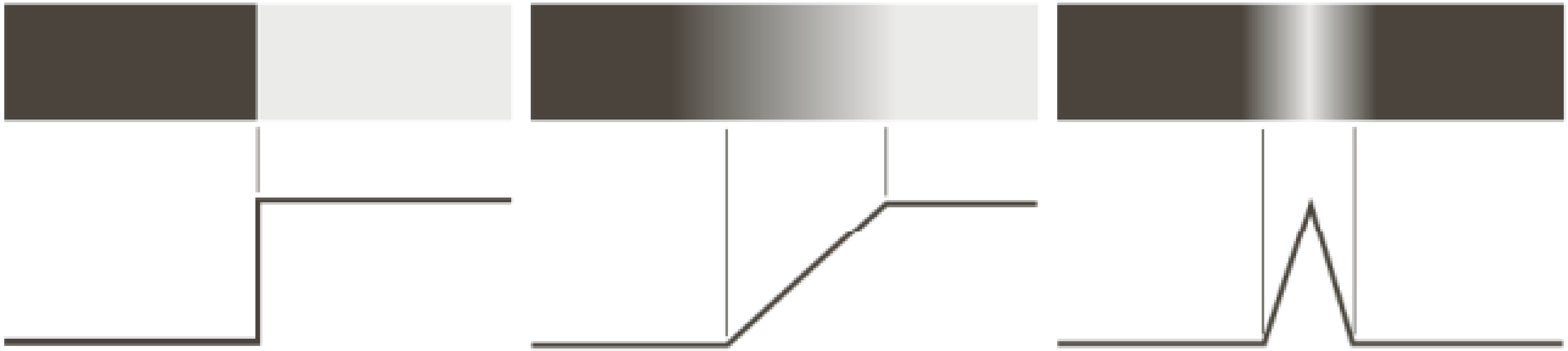
❑ Edge detection:

- Edge detection is the approach for segmenting images based on abrupt changes in intensity. It is used to detect the boundaries or to find size or location of an object in an image.

- What is an edge?
  - an edge is a set of connected pixels that lie on the boundary between two regions.
  - an edge is a "local" concept whereas a region boundary, owing to the way, it is defined, is a more global idea.

❑ Edge models:

- Step edge (Ideal edge), ramp edge (thick edge), and roof edge



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.

- An image may all the three types of edges.

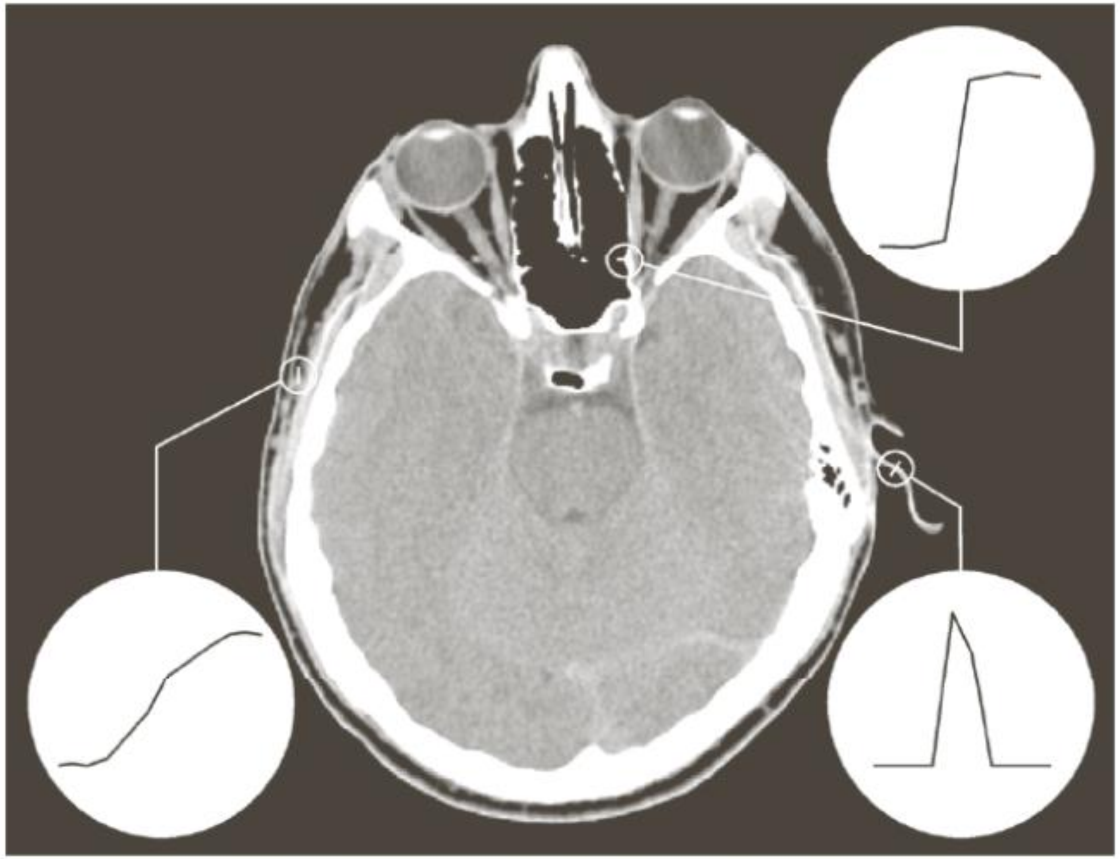
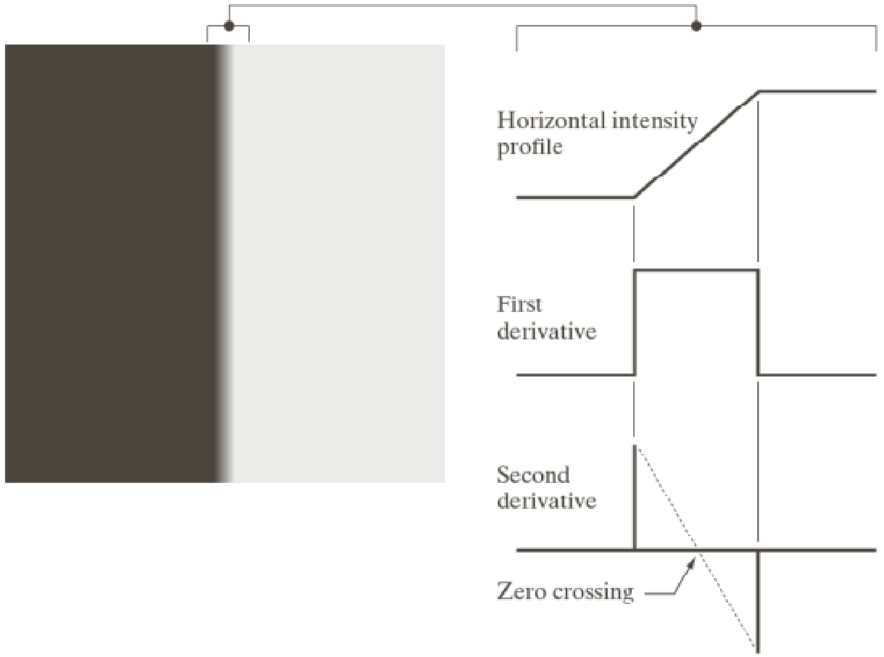


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

First and Second order derivatives response at the edge:



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.

1. The magnitude of the first derivative can be used to detect the presence of an edge at a point.
2. Sign of second derivative indicates which side of the edge the pixel is on.
3. Zero crossings of the second derivative can be used to locate the centers of thick edges.

Second derivative produces two values for every edge in an image. An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. Zero-crossing point is the center of thick edges.

❑ Steps in edge detection

1. Image smoothing for noise reduction.
2. Detection of edge points (this extracts from an image all points that are potential candidates to become edge points).
3. Edge localization (this selects from candidate edge points only the points that are true members of the set of points comprising an edge).

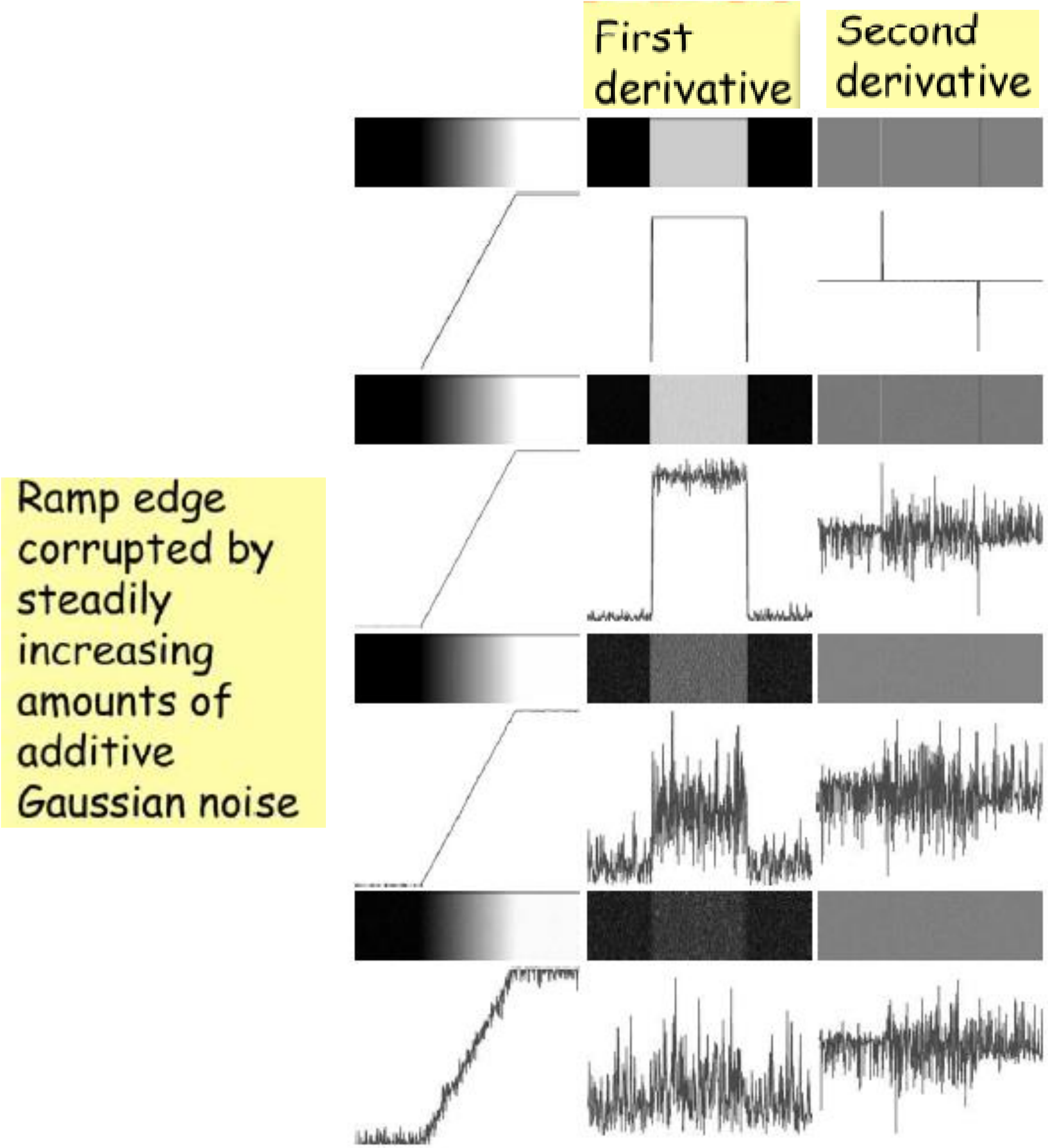


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images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

❑ Image Gradient (1^{st} order derivatives)

- Gradient is a vector

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

- The magnitude of the gradient

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

- The direction of the gradient vector

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

- The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point

❑ Gradient (1st order derivatives) Operators and Masks

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

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

Roberts cross-gradient operators

$$g_x = \frac{\partial f(x,y)}{\partial x} = z_9 - z_5$$

$$g_y = \frac{\partial f(x,y)}{\partial y} = z_8 - z_6$$

Prewitt operators

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

Sobel operators

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

a b

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

(a)

−1
1

(b)

−1	1

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.

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

Prewitt and Sobel masks for Diagonal edges

Modified Prewitt and Sobel masks for detecting diagonal edges

Prewitt

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Sobel

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

-45°   +45°

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

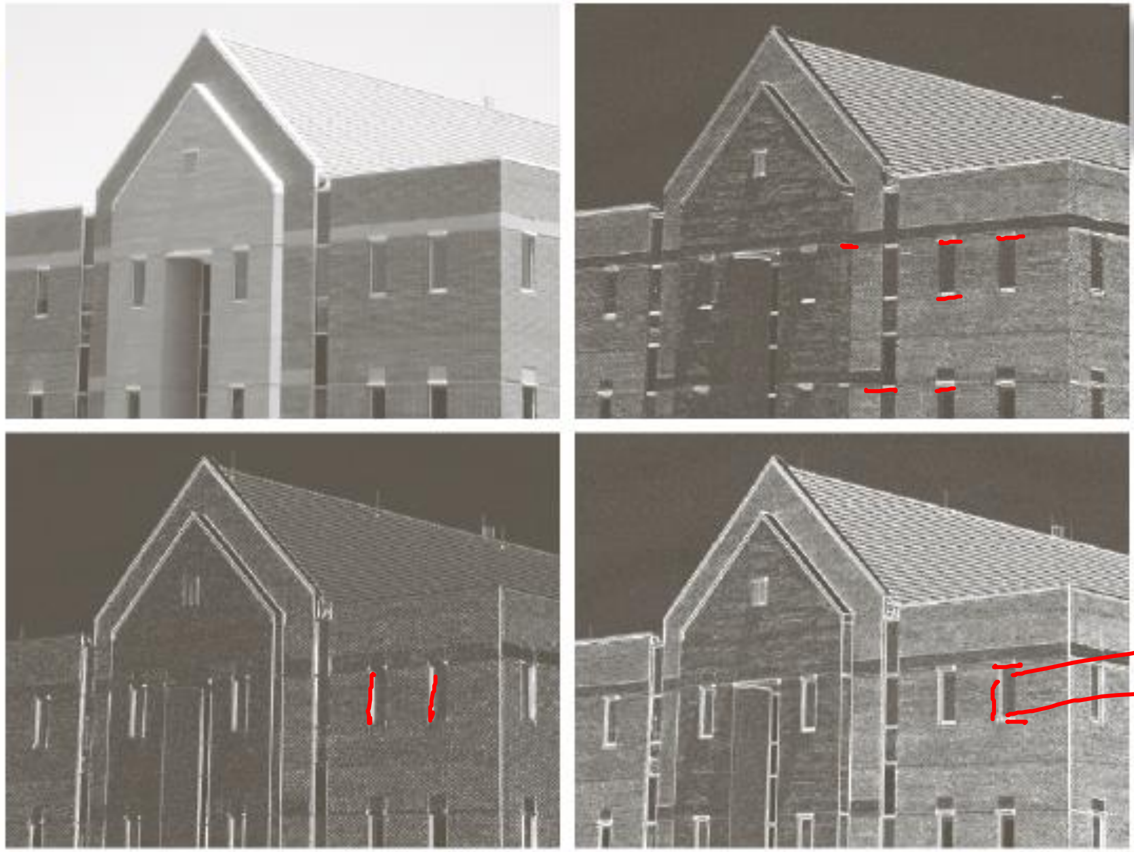
The Prewitt and Sobel are the most commonly used gradient masks

The Sobel has better noise characteristics than the Prewitt

Example of use of Gradient

1200x1600 original image

$|G_x|$ using 3x3 Sobel



$|G_y|$ using 3x3 Sobel

$|G_x|+|G_y|$ gradient approximation

a b
c d

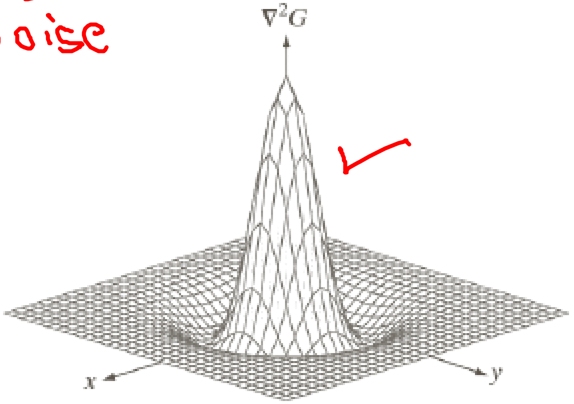
FIGURE 10.16 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

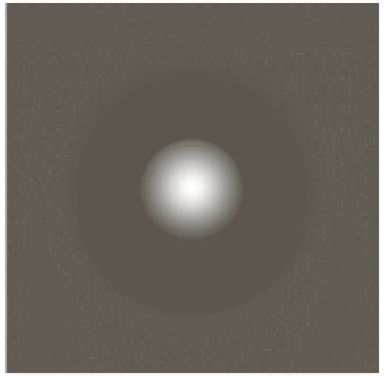
breaks

❖ Laplacian (Second order derivative) for edge detection

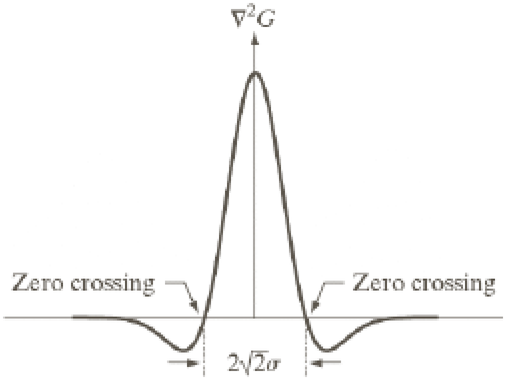
Laplacian $\rightarrow$ sharp edges detection
Gaussian $\rightarrow$ reduce noise

The LoG is sometimes called the Mexican hat operator





The Laplacian is NEVER used directly because of its strong noise sensitivity



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

a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

Combining the Laplacian with a Gaussian gives the LoG

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

Edge supports six different edge-finding methods:

The Sobel	Finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
The Prewitt	Finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
The Roberts	Finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
The Laplacian of Gaussian	Finds edges by looking for zero crossings after filtering I with a Laplacian of Gaussian filter.
The zero-cross	Finds edges by looking for zero crossings after filtering I with a filter you specify.
The Canny	Finds edges by looking for local maxima of the gradient of I. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be "fooled" by noise, and more likely to detect true weak edges.

❖ Conclusions for Edge detections:

- Gradient based classical operators (first derivatives) are simple in implementation and does fast computations.
- Robert operator is simplest but it is most sensitive to noise.
- The response of Sobel and Prewitt operators are almost similar, but still Sobel produces better output due to more resistance against noise than Prewitt.
- For noisy images, Laplacian of Gaussian and Canny operators (Second derivatives) must be used.

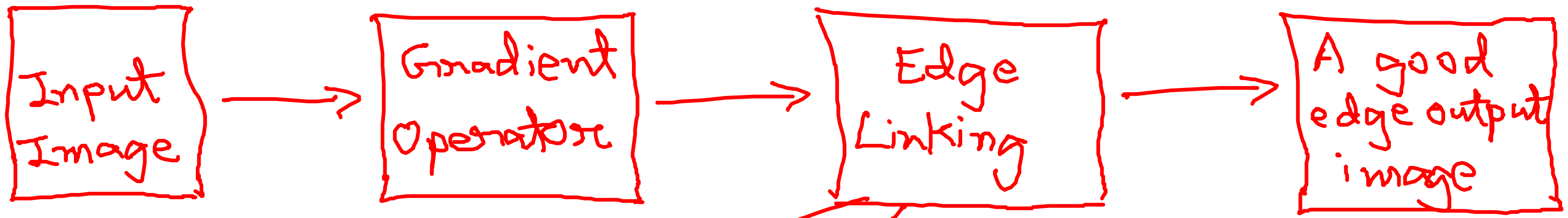
Thank You

50

Edge Linking and Boundary Detection, Hough Transform and its implementation in MATLAB

❖ Edge Linking

- An edge detection algorithm (Roberts, Sobel, Prewitt, LoG etc.) enhance the edges. When implemented, there are normally breaks in lines. Due to this reason, these are generally followed by linking procedures to assemble edge pixels into meaningful edges.



- There are two basic approaches for edge linking:

(1) Local Processing. (This is a simplest approach for linking pixels in a small neighborhood)

(2) Global Processing via the Hough Transform. (Here, we attempt to link edge pixels that lie on specified curves. The Hough transform is designed to detect lines, using the parametric representation of a line.)

❖ Local Processing

- Analyze the characteristics of pixels in a small neighborhood S_{xy} (say, 3x3, 5x5) about every edge pixels (x, y) in an image that have undergone edge detection.

Edge to be linked

- All points that share some common properties are linked together. These are:

(x,y) (s,t)

a) Strength/magnitude of the gradient.

b) Direction of the gradient.

- The magnitude of the gradient

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

- The direction of the gradient vector

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

✓ pixels (s,t) and (x,y) are similar and linked if,

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

where E is a positive threshold

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

where A is a positive angle threshold

- **Algorithm steps:**

1. compute $\nabla f, M(x,y), \alpha(x,y)$

2. Form a binary image $g(x,y) = \begin{cases} 1, & M(x,y) > T_M \text{ and } \alpha(x,y) = A \pm T_A \\ 0, & \text{otherwise} \end{cases}$

3. Scan the rows of g and fill all gaps ineach tow that do not exceed a specified length K

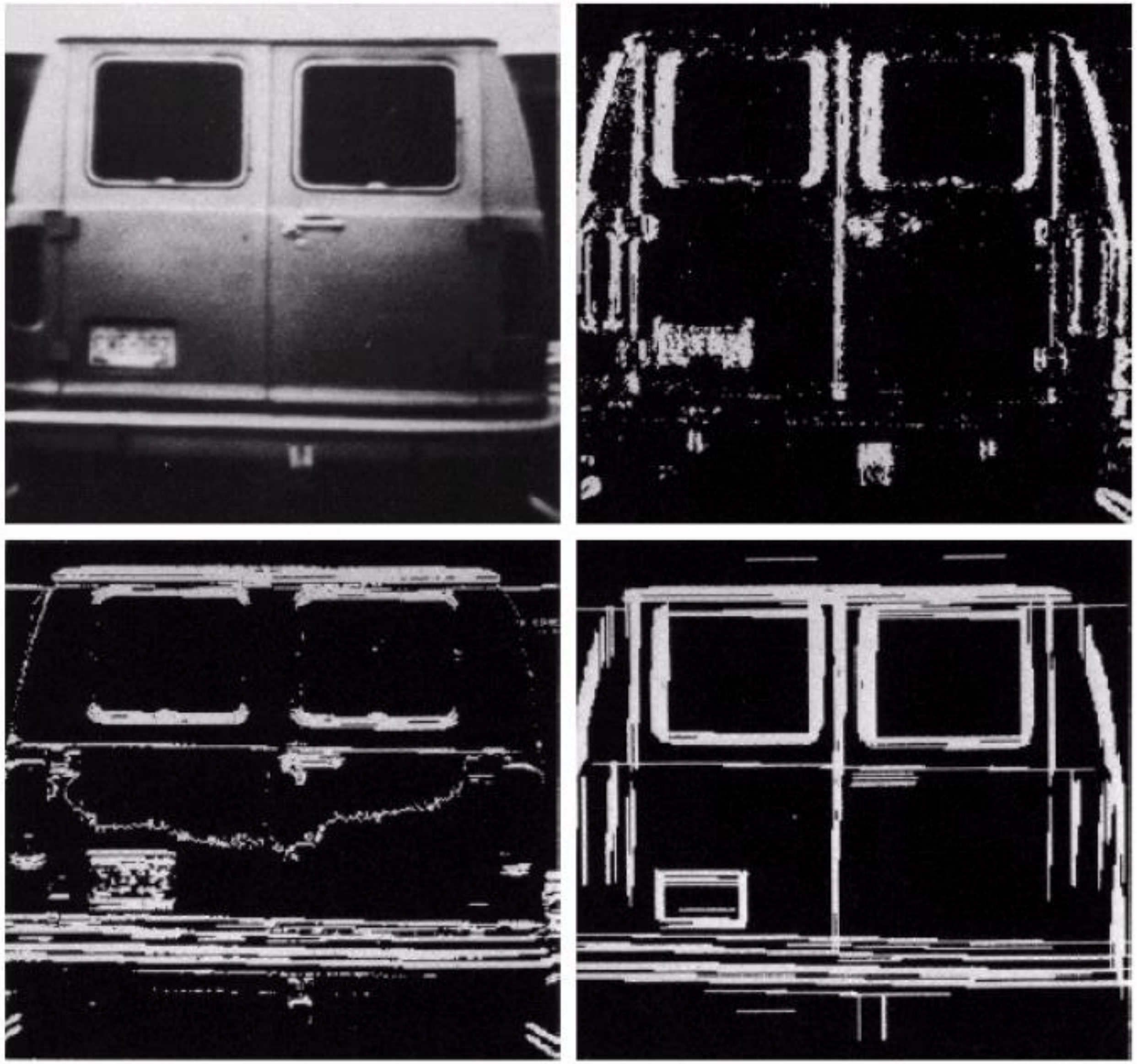
4. Detect the gap in any direction θ, rotate g by this angle and apply the horizontal scanning procedure in Step 3. Rotate the result by - θ

- This local processing is expensive. A record has to be kept of all linked points by, for example, assigning a different label to every set of linked points.

❖ Example of local processing:

a	b
c	d

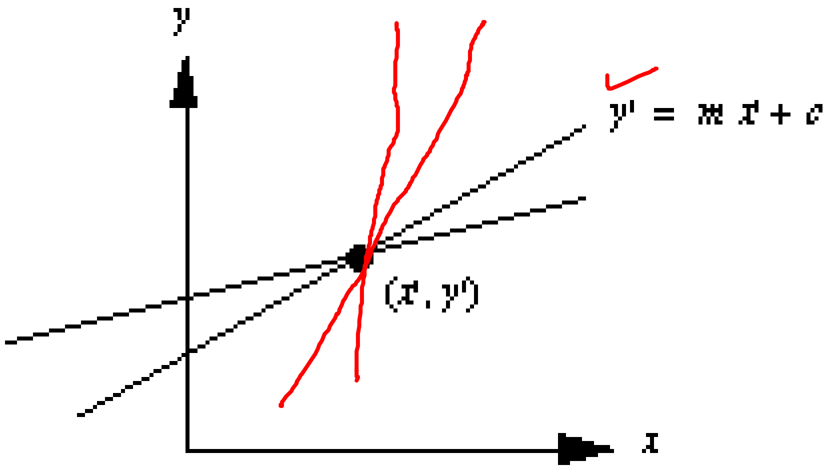
FIGURE 10.16
(a) Input image.
(b) G_y component of the gradient.
(c) G_x component of the gradient.
(d) Result of edge linking. (Courtesy of Perceptics Corporation.)



❖ Hough Transform (Global Processing)

- One powerful global method for detecting edges is called the ***Hough transform***.
- Let us suppose that we are looking for straight lines in an image.
- If we take a point (x',y') in the image, all lines which pass through that pixel have the form,

$$y' = mx' + c$$ for varying values of m and c.



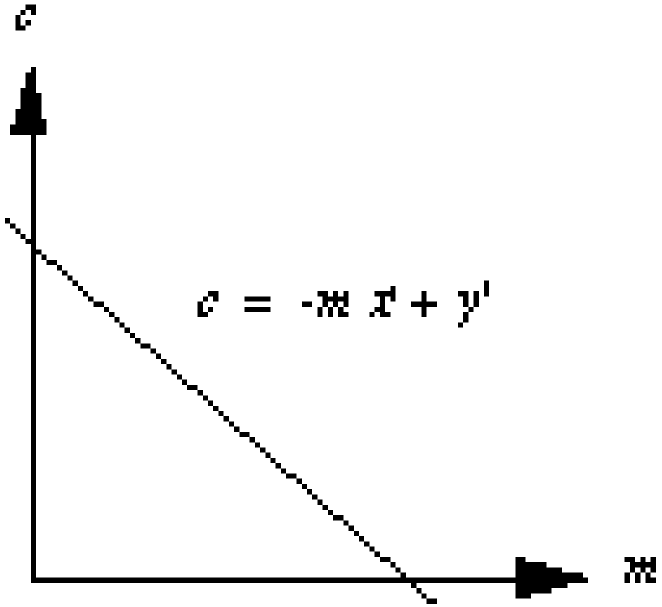
- However, this equation can also be written as,

$$c = -x'm + y'$$

where we now consider x' and y' to be constants, and m and c as varying.

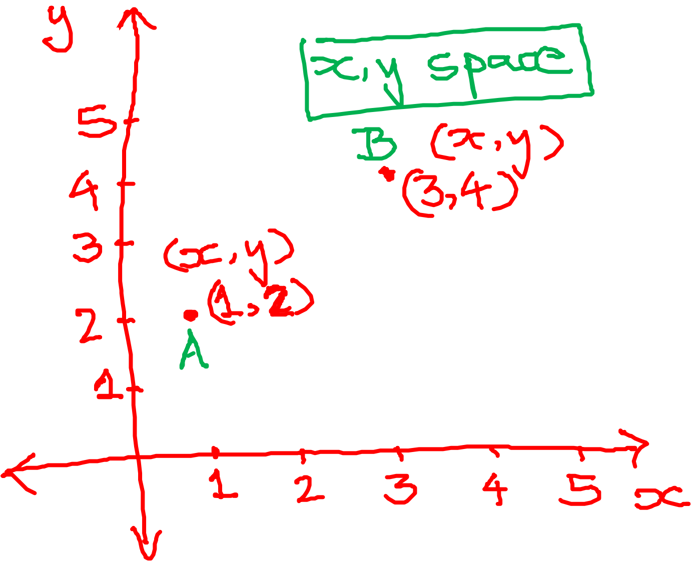
❖ Hough Transform (Global Processing)

- This is a straight line on a graph of c against m as shown in Fig. below.



- Each different ***line*** through the point (x',y') corresponds to one of the ***points*** on the line in (m,c) space.

m,c space/Hough space



$(x,y) = (1,2)$

$y = mx + c \Rightarrow c = -mx + y$

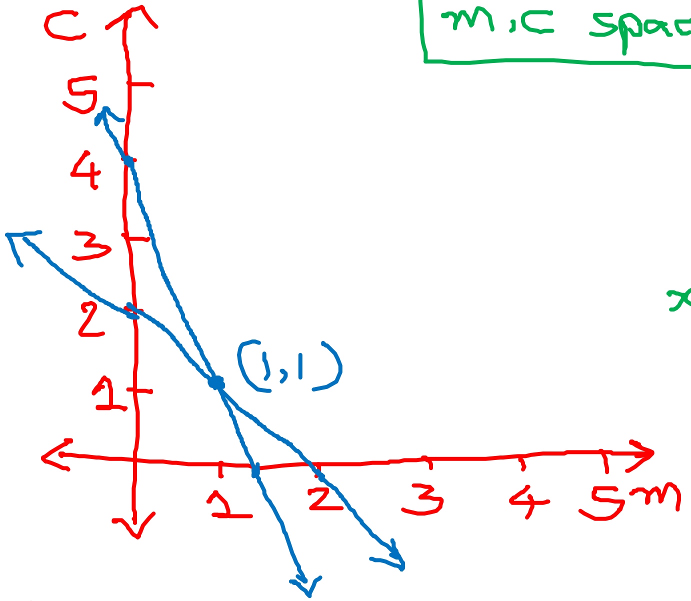
$\therefore c = -m(1) + 2$

$\therefore c = -m + 2$

If $c = 0, m = 2$

$m = 0, c = 2$

$\therefore (m,c) = (2,2)$



$(x,y) = (3,4)$

$y = mx + c \Rightarrow c = -mx + y$

$\therefore c = -m3 + 4$

$\therefore c = -3m + 4$

If $c = 0$, $m = \frac{4}{3} = 1.33$

$m = 0, \; c = 4$

$\therefore (m,c) = (1.33, 4)$

$(m,c) = (1,1)$

$x,y = 1,2$ $\quad y = mx + c$

$y = 1 \cdot 1 + 1 = 2$ ✓

$x,y = 3,4$

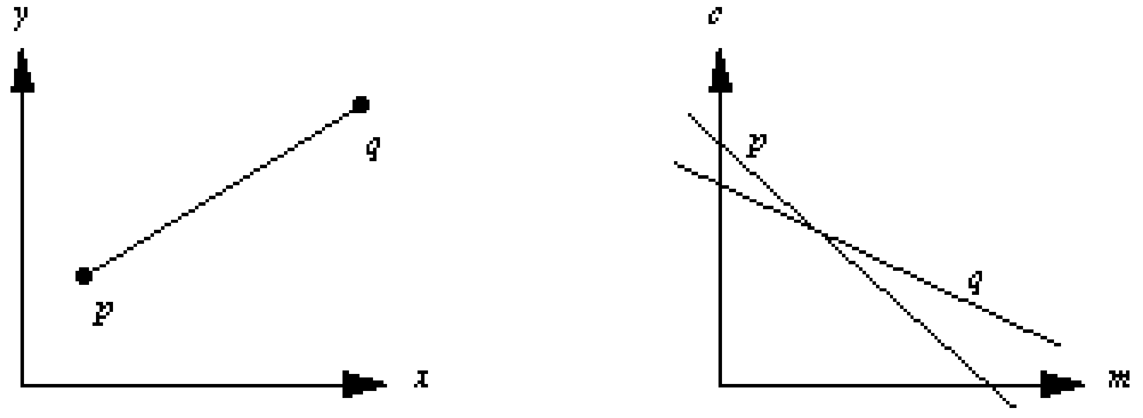
$y = mx + c$

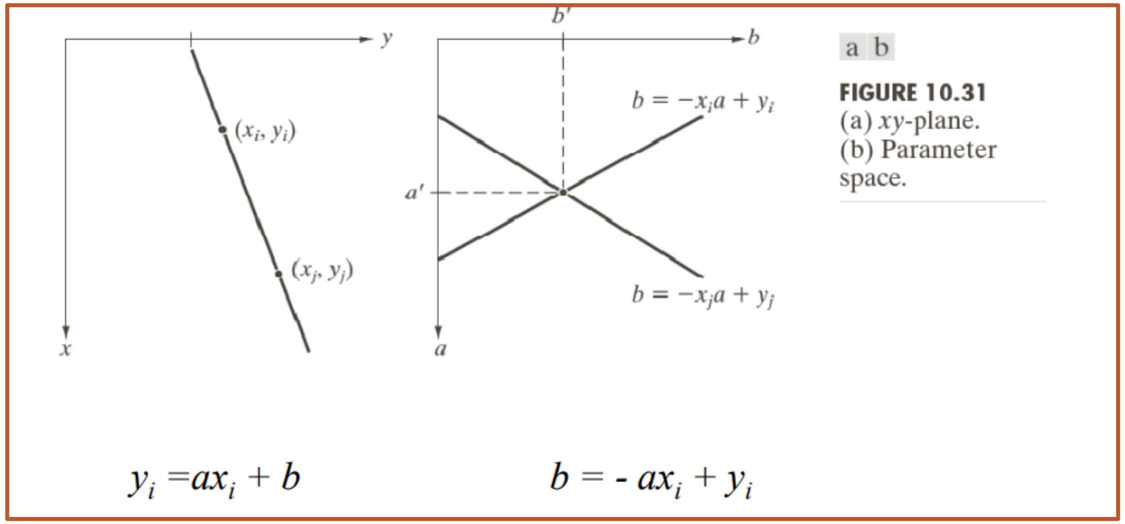
$y = 1 \cdot 3 + 1 = 4$ ✓

If points A and B are two points connected by a line in the spatial domain (x,y) space, they will be intersecting lines in the Hough space.

❖ Hough Transform (Global Processing)

- Now consider two pixels *p* and *q* in (*x*,*y*) space which lie on the same line.
- For each pixel, all of the possible lines through it are represented by a single line in (*m*,*c*) space.





- Thus the single line in (*x*,*y*) space which goes through both pixels lies on the intersection of the two lines representing *p* and *q* in (*m*,*c*) space, as shown in figure. If p and q are the points connected by a line in the spatial domain, then they will be intersecting lines in the Hough Space (m-c plane).

❖ Hough Transform (Global Processing)

- Taking this one step further, all pixels which lie on the same line in (x,y) space are represented by lines which all pass through a single point in (m,c) space.
- The single point through which they all pass gives the values of m and c in the equation of the line $y=mx+c$.

To detect straight lines in an image, we do:

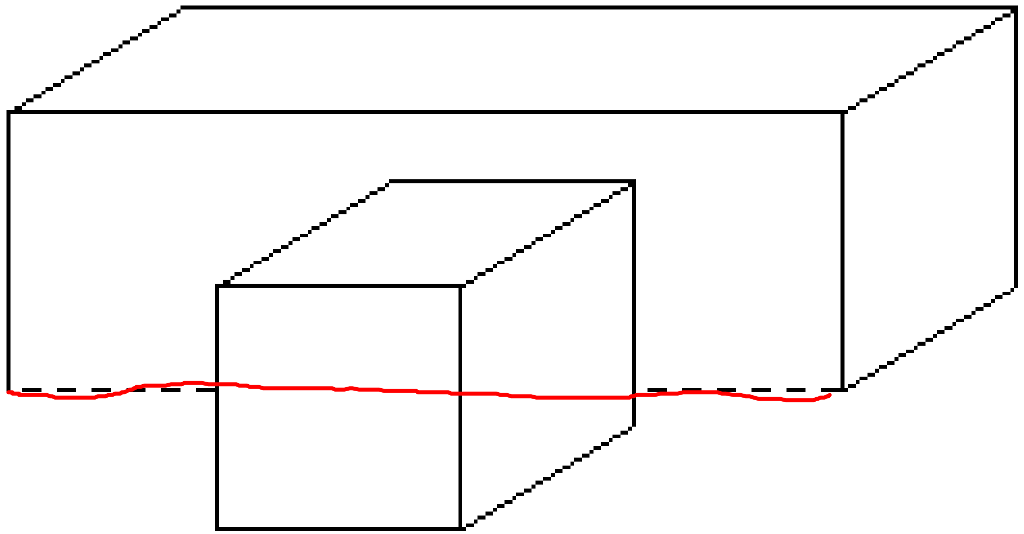
1. Quantize (m,c) space into a two-dimensional array A for appropriate steps of m and c.
2. Initialize all elements of $A(m,c)$ to zero.
3. For each pixel (x',y') which lies on some edge in the image, we add 1 to all elements of $A(m,c)$ whose indices m and c satisfy $y'=mx'+c$.
4. Search for elements of $A(m,c)$ which have large values. Each one found corresponds to a line in the original image.

- One useful property of the Hough transform is that the pixels which lie on the line need not all be contiguous.

❖ Hough Transform (Global Processing)

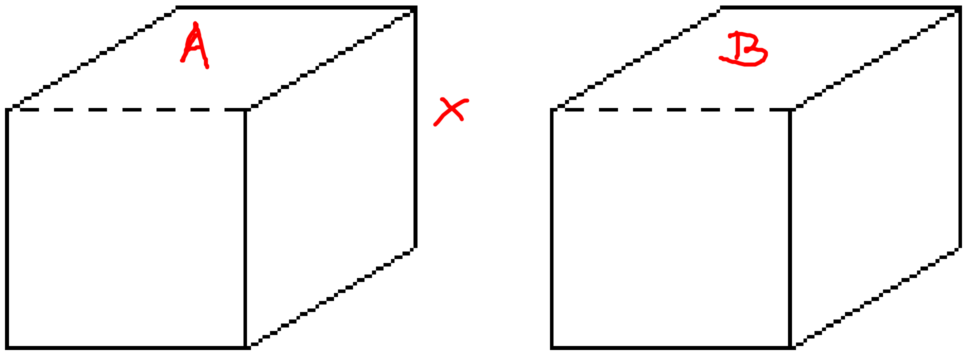
- For example, all of the pixels lying on the two dotted lines in below Fig. will be recognized as lying on the same straight line.

- This can be very useful when trying to detect lines with short breaks in them due to noise, or when objects are partially obstructed as shown.



❖ Hough Transform (Global Processing)

- On the other hand, it can also give misleading results when objects happen to be aligned by chance, as shown by the two dotted lines in Fig. below,



- Indeed, this clearly shows that one disadvantage of the Hough transform method is that it gives an ***infinite line*** as expressed by the pair of m and c values, rather than a finite ***line segment*** with two well-defined endpoints.
- One practical detail is that the $y = mx + c$ form for representing a straight line breaks down for vertical lines, when m becomes infinite.

$(\rho, \theta) \rightarrow$ polar coordinates

$x\cos\theta + y\sin\theta = \rho$

❖ Hough Transform (Global Processing)

- To avoid this problem, it is better to use the alternative formulation given earlier,

$$x\cos\theta + y\sin\theta = r,$$

as a means of describing straight lines.

- Note, however, that a point in (x,y) space is now represented by a curve in (r, θ) space rather than a straight line. Otherwise, the method is unchanged.
- The Hough transform can be used to detect other shapes in an image as well as straight lines.

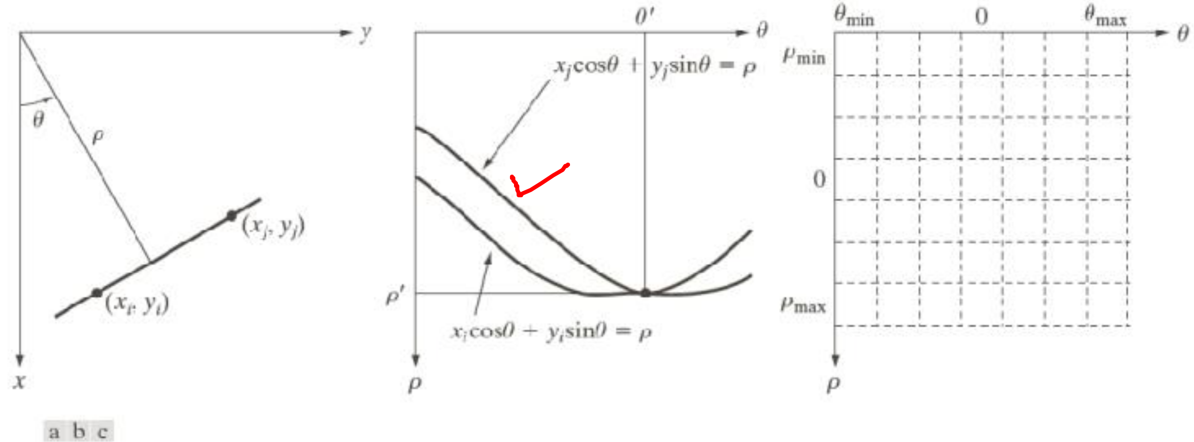


FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.

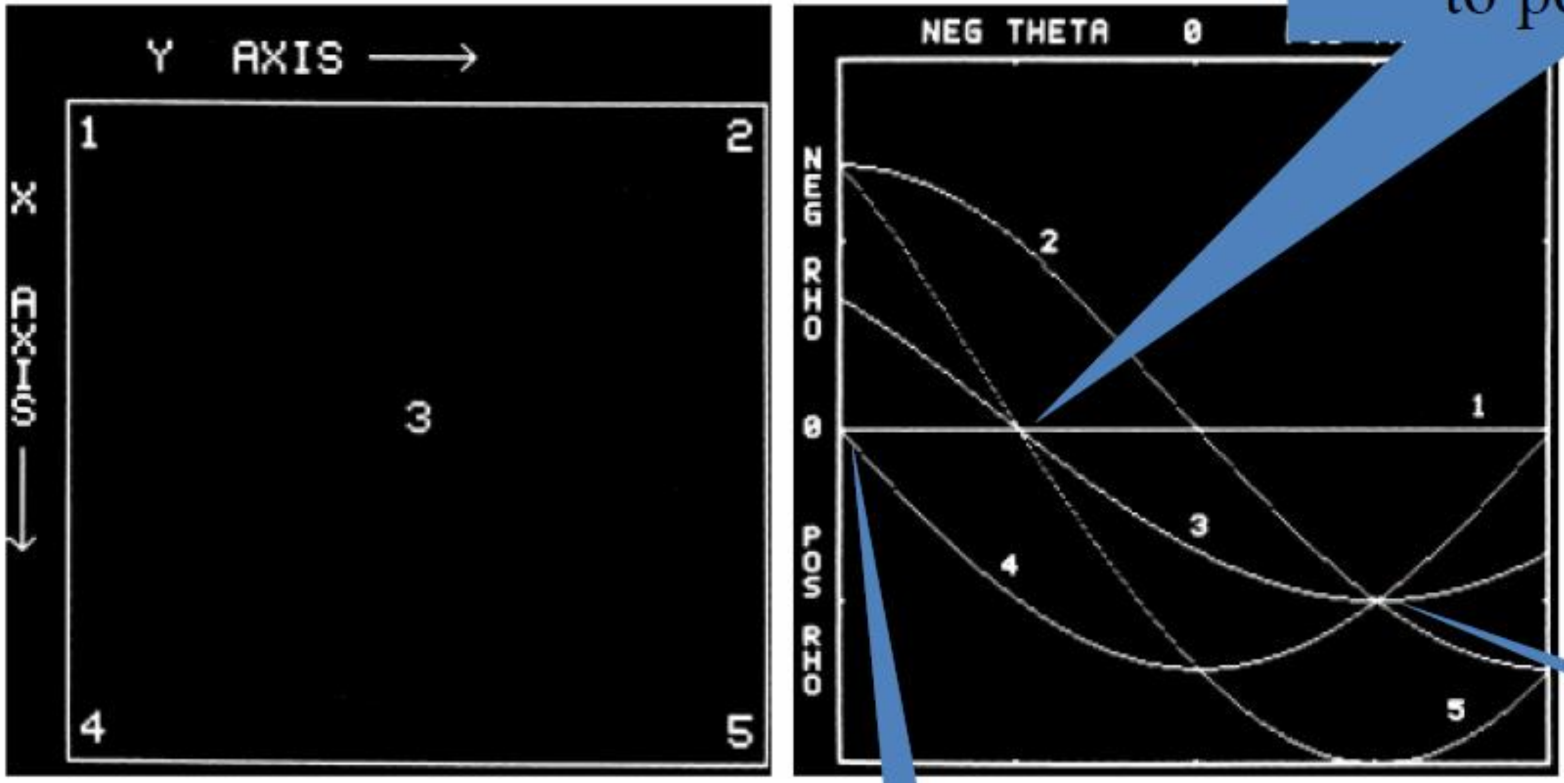


FIGURE 10.20
Illustration of the Hough transform. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

❖ Hough Transform (Global Processing)

- We attempt to link edge pixels that lie on specified curves. The Hough transform is designed to detect lines, using the parametric representation of a line.
- Hough Transforms takes the images created by the edge detection operators.
- Hough Transform 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 point.
- Therefore, an edge point in an x-y plane is transformed to a m-c plane.
- Now equation of line is $c = (-x)m + y$

❖ Hough Transform (Global Processing)

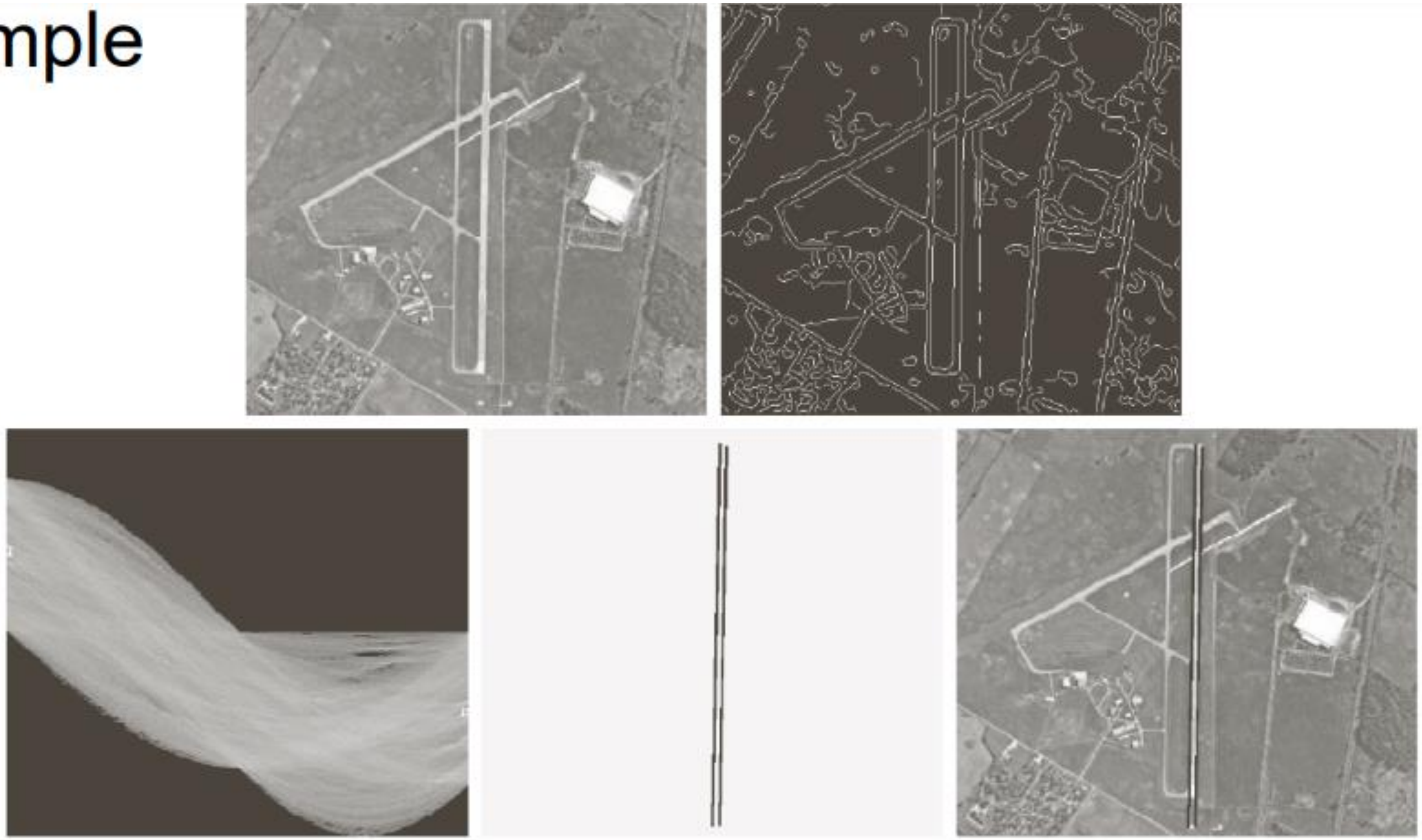
- All the edge points in the x-y plane need to be fitted.
- All the points are transformed in m-c plane.
- The objective is to find the intersection point.
- A common intersection point indicates that the edges points which are part of the line.
- If A and B are the points connected by a line in the spatial domain, then they will be intersecting lines in the Hough Space (m-c plane).
- To check whether they are intersecting lines, the m-c 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 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.
- Solution is, 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.

Example



a b
c d e

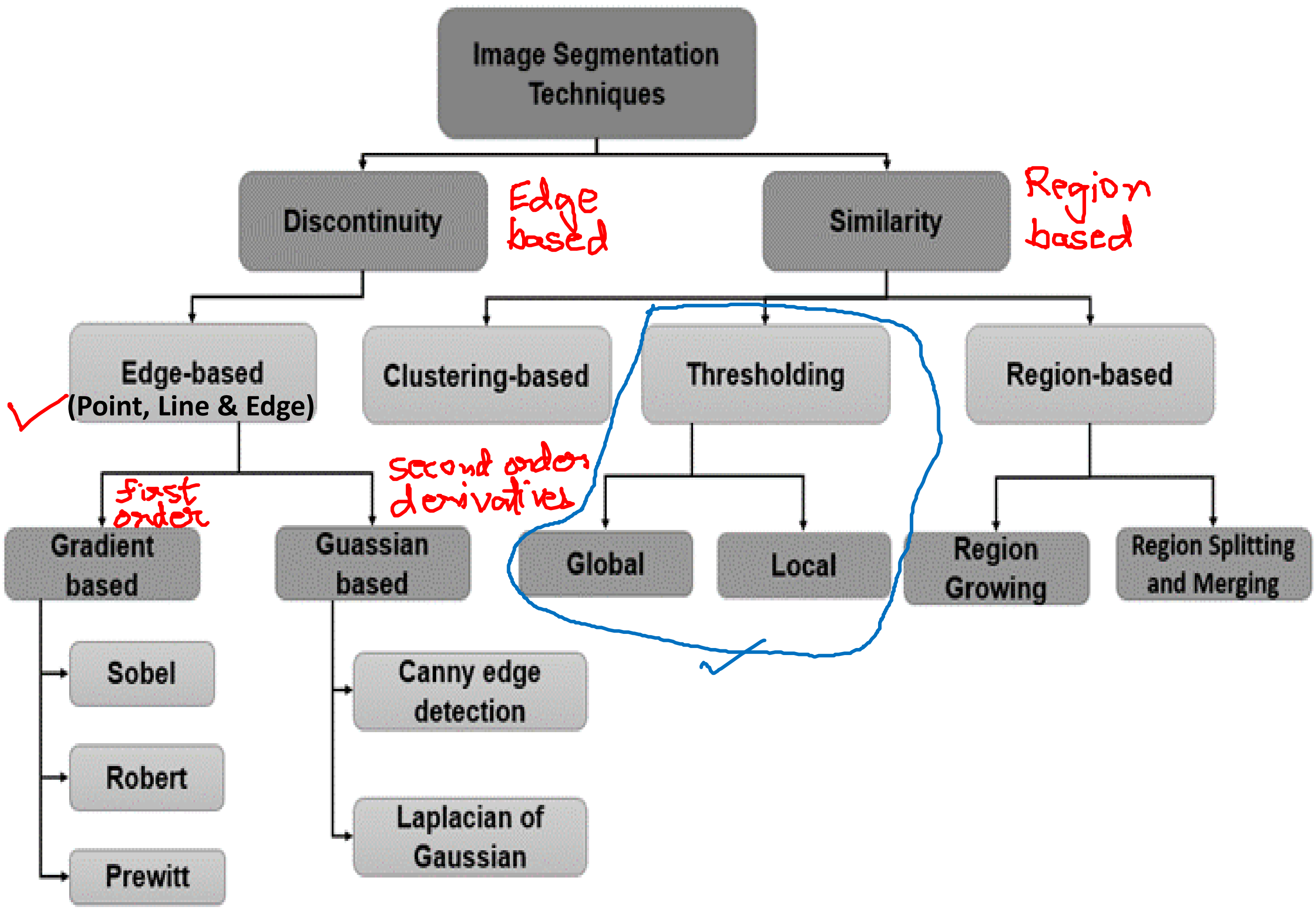
FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed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

Thank You

51

Thresholding : Global and Adaptive(local) in DIP and its implementation in MATLAB

❑ Image Segmentation:



For the most part there are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.

❖ **Discontinuity**

The strategy is to partition an image based on abrupt changes in intensity

➢ Detection of gray level discontinuities:

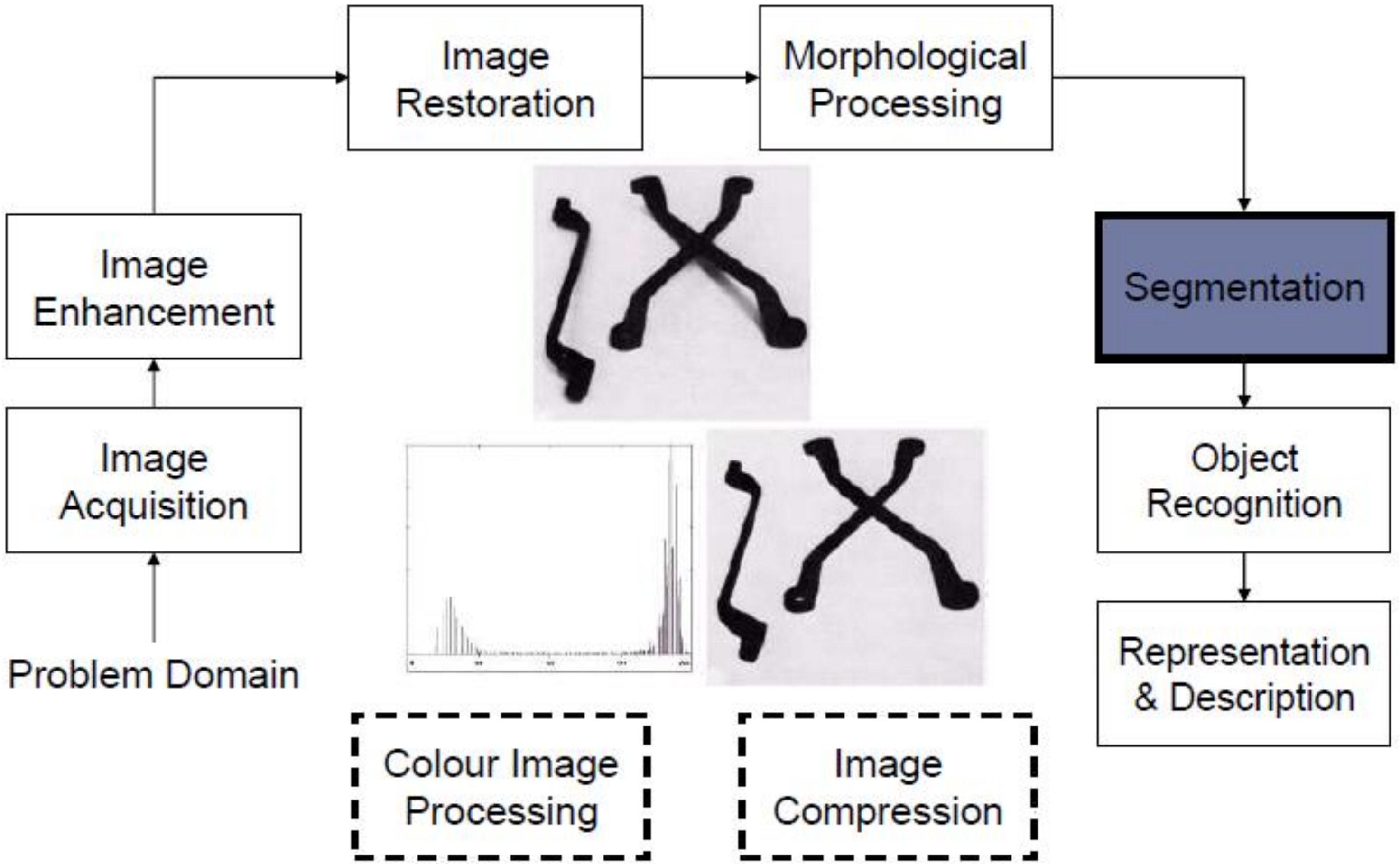
- Point detection
- Line detection
- Edge detection
    - Gradient operators
    - Gaussian based operators

❖ **Similarity**

The strategy is to partition an image into regions that are similar according to a set of predefined criteria.

Image Segmentation

Here the computer tries to separate objects from the image.



- The spatial domain processes can be denoted by the expression, g(x, y) = T[f(x, y)], where f(x, y) is the input image, g(x, y) is the output image, and T is an operator on f defined over a neighborhood of point (x, y).

- The smallest possible neighborhood is of size 1 x 1. Here, g depends only on the value of f at a single point (x, y) and T becomes s = T(r), where, for simplicity in notation, s and r are variables denoting, respectively, the intensity of g and f at any point (x, y).

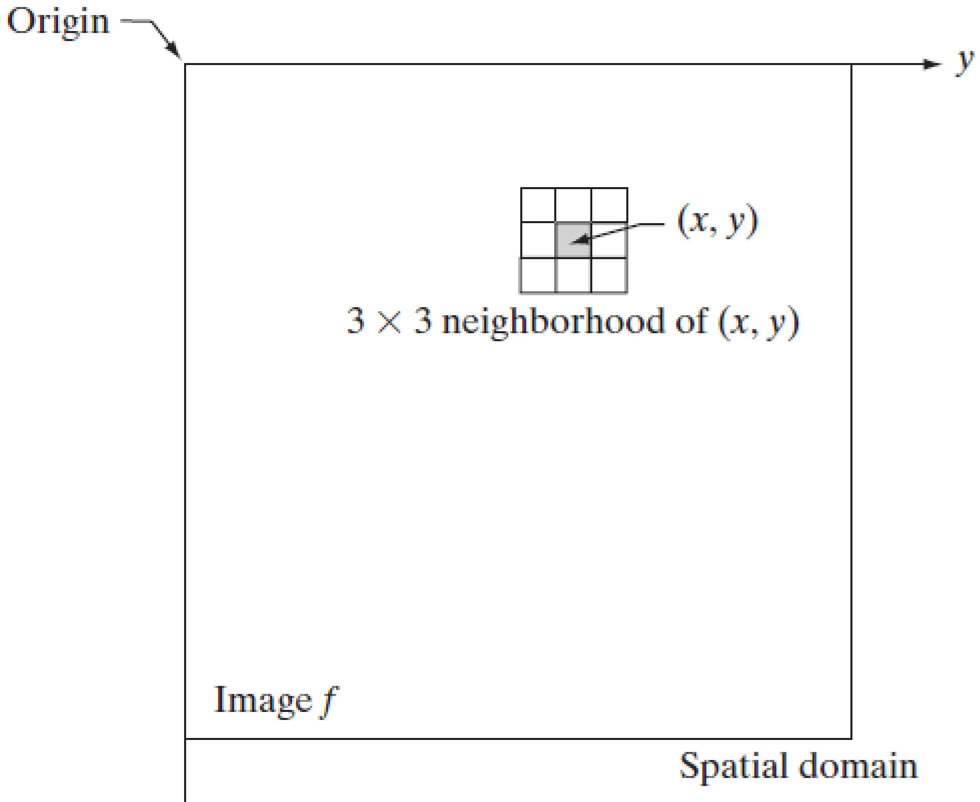
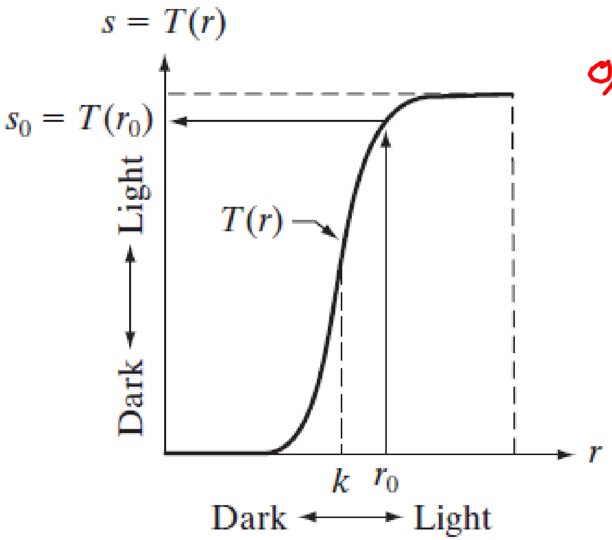
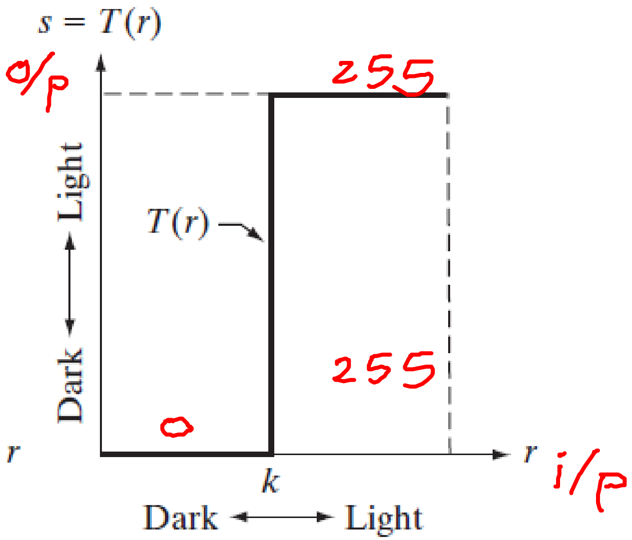


Image with a 3 X 3 spatial mask



Contrast stretching
T function



Thresholding
T function

❖ Segmentation by thresholding

- Thresholding is the simplest segmentation method. Thresholding is used to produce regions of similarity within the given image, based on some threshold criteria T. Hence it partitions/segments an image into different objects.
- The pixels are partitioned depending on their intensity value.

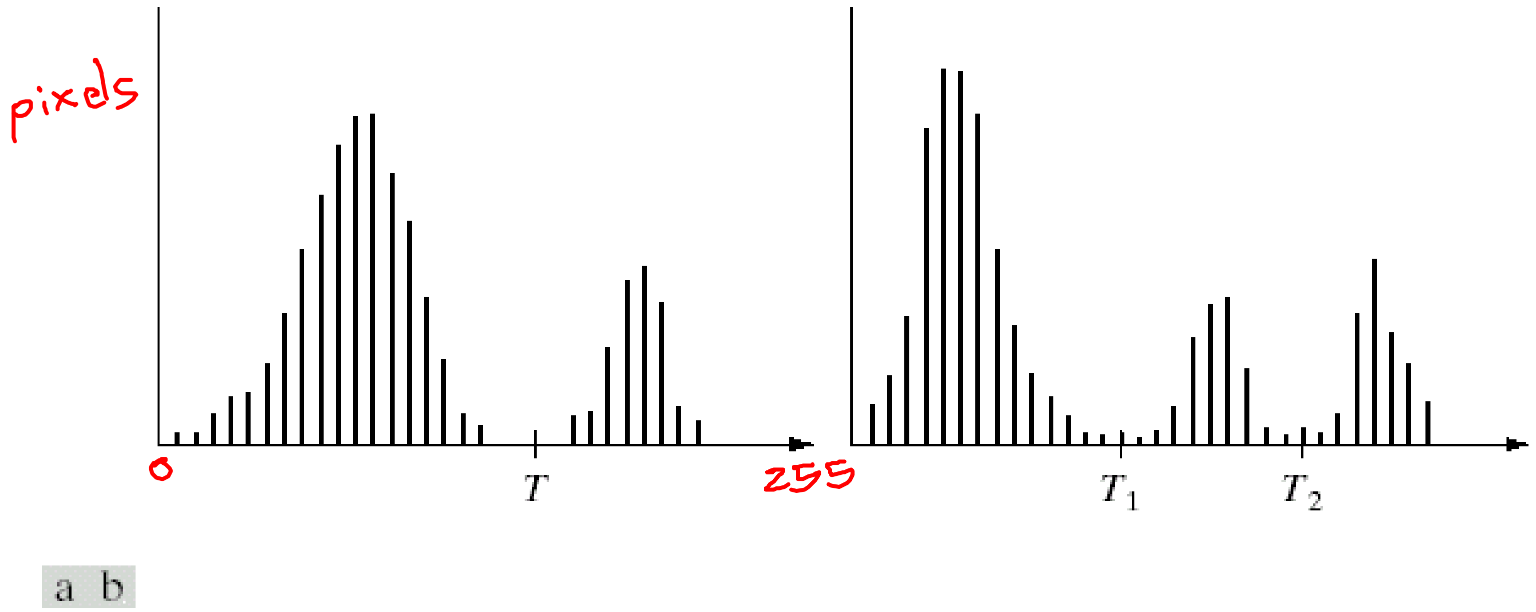
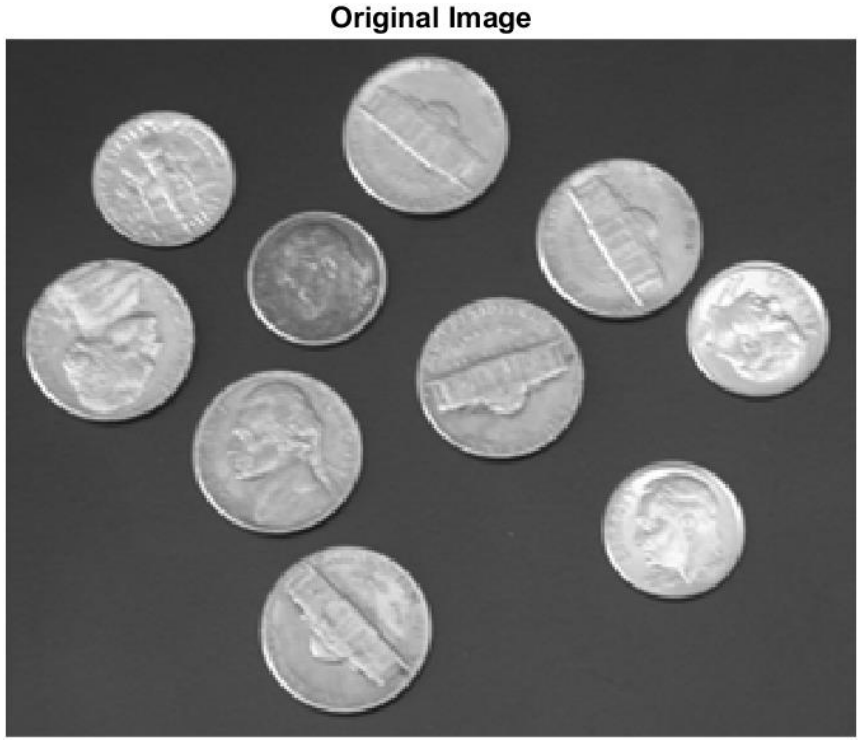
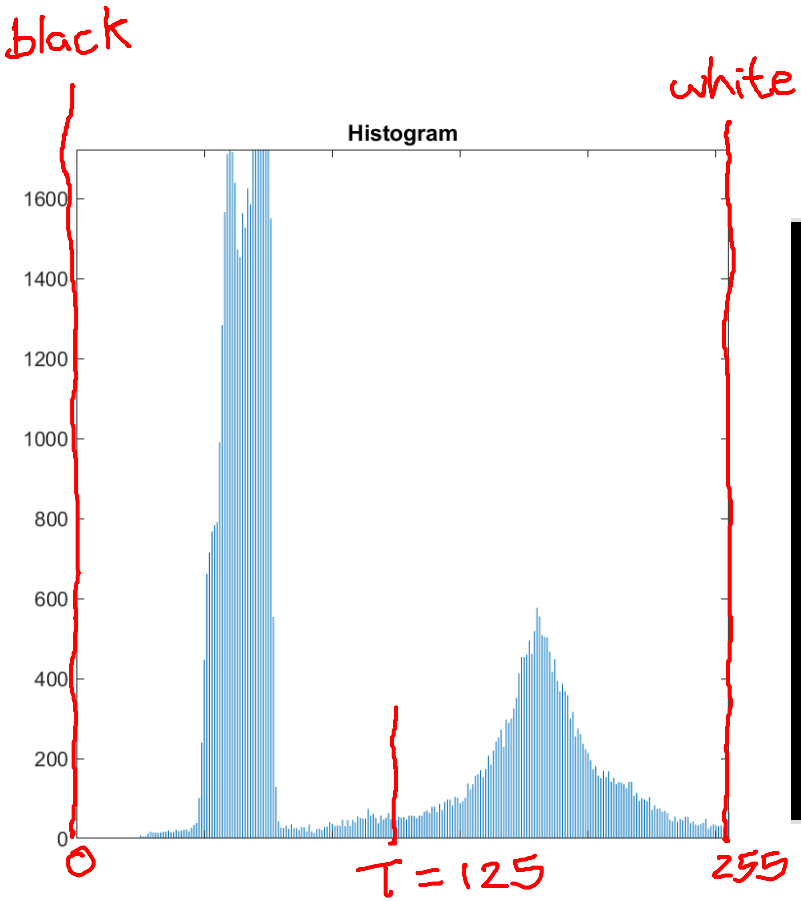


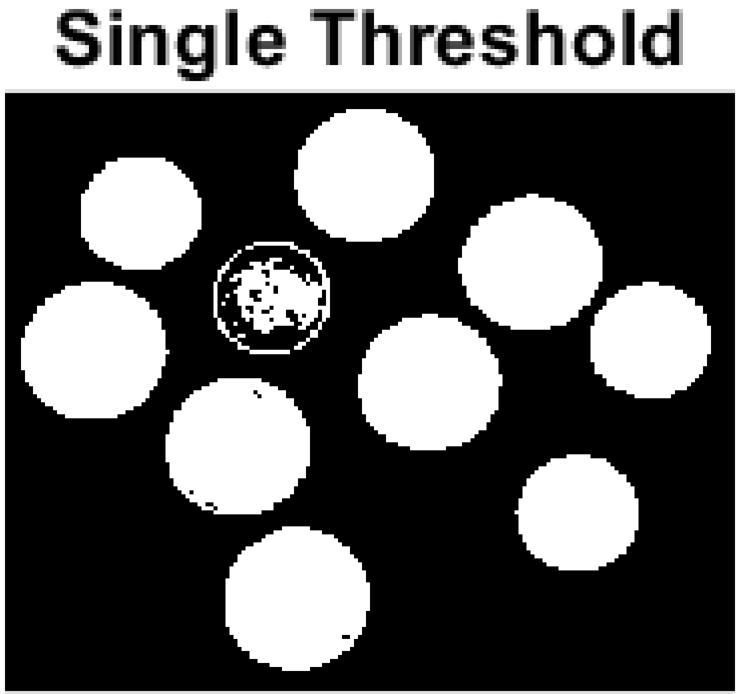
FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

- **Single thresholding:**

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

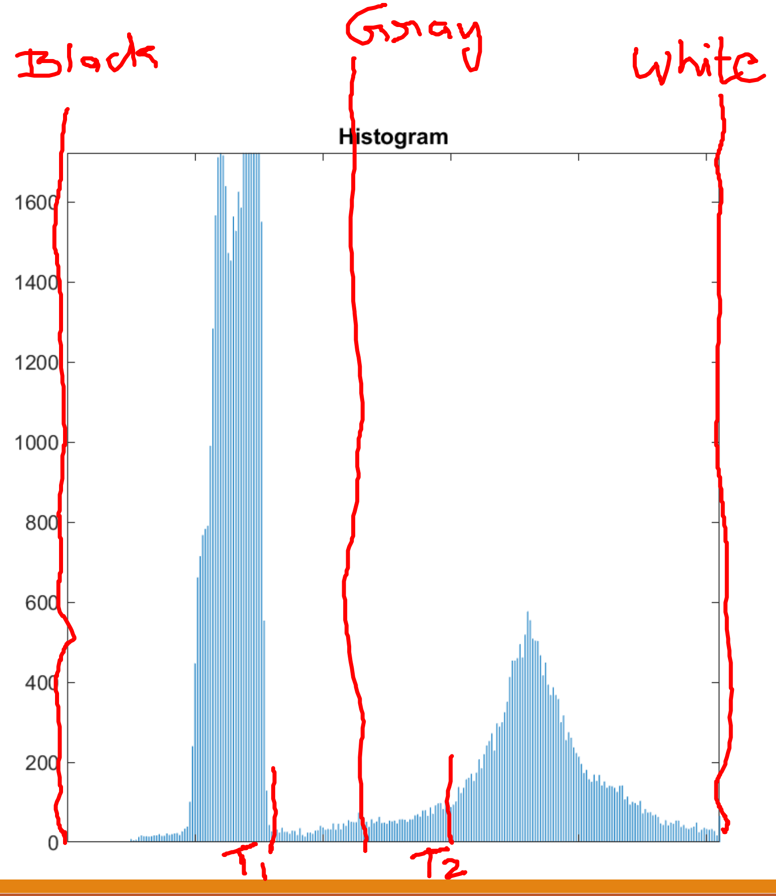


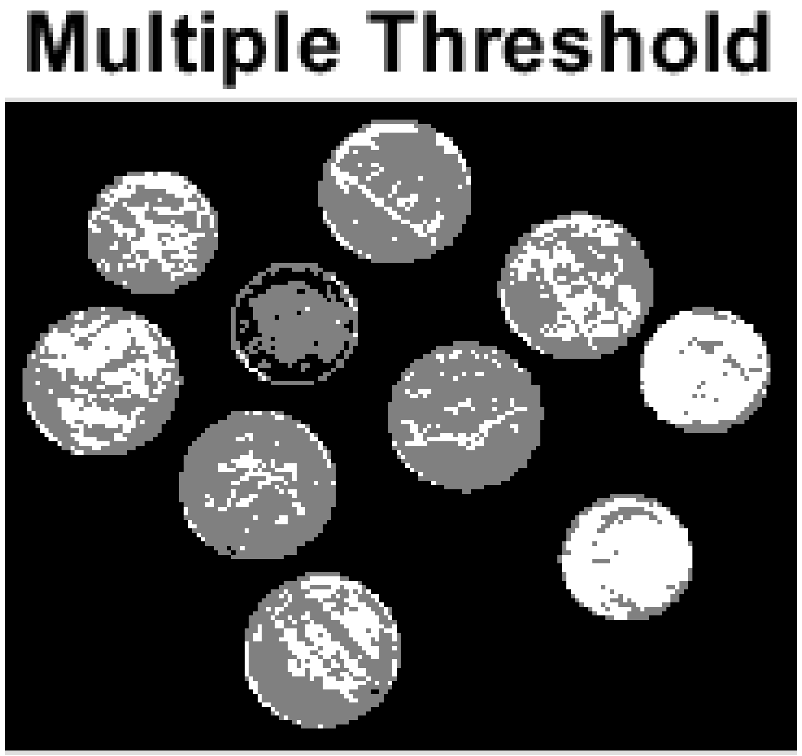




- **Multiple thresholding:**

$$
g(x, y) = \begin{cases} a, & \text{if } f(x, y) > T2 \\ b, & \text{if } T1 < f(x, y) \leq T2 \\ c, & \text{if } f(x, y) \leq T1 \end{cases}
$$





❖ Thresholding types:

- Thresholding operation can be thought of as an operation, such that, $T = T[x, y, p(x, y), f(x, y)]$, where $f(x, y)$ is the gray level of input pixel at (x, y) and p(x, y) denotes some local property of this point (x, y), e.g. the average level of a neighborhood centered on (x, y).
- If the thresholding operation depends only on the gray scale value, it is called global thresholding. If the neighborhood property is also taken into account, it is called local thresholding. If T depends on pixel coordinates also, T is called dynamic/adaptive thresholding. Thresholding is called adaptive when a different threshold is used for different regions in the image.

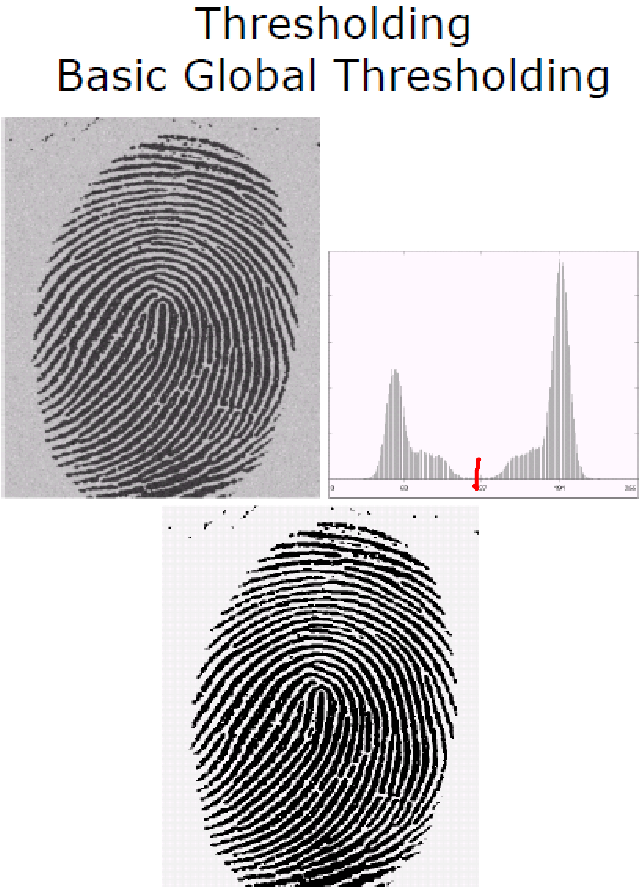
❖ How to select T ? OR How to choose Threshold value T ?

- Assuming that the background and the object occupy comparable areas in the image, a good initial value of T is the average gray level of the image.

1. Select an initial estimate for T.
2. Segment the image into two group of pixels G_1 and G_2 using T.
3. Compute the average gray level values of G_1 and G_2, say the values obtained are μ_1 and μ_2.
4. Compute a new threshold value as $T_{new}=(\mu_1+\mu_2)/2$,
5. Compare if $|T_{new}-T|>z_0$ (predefined threshold z_0) then $T=T_{new}$ and go to step 2, else stop

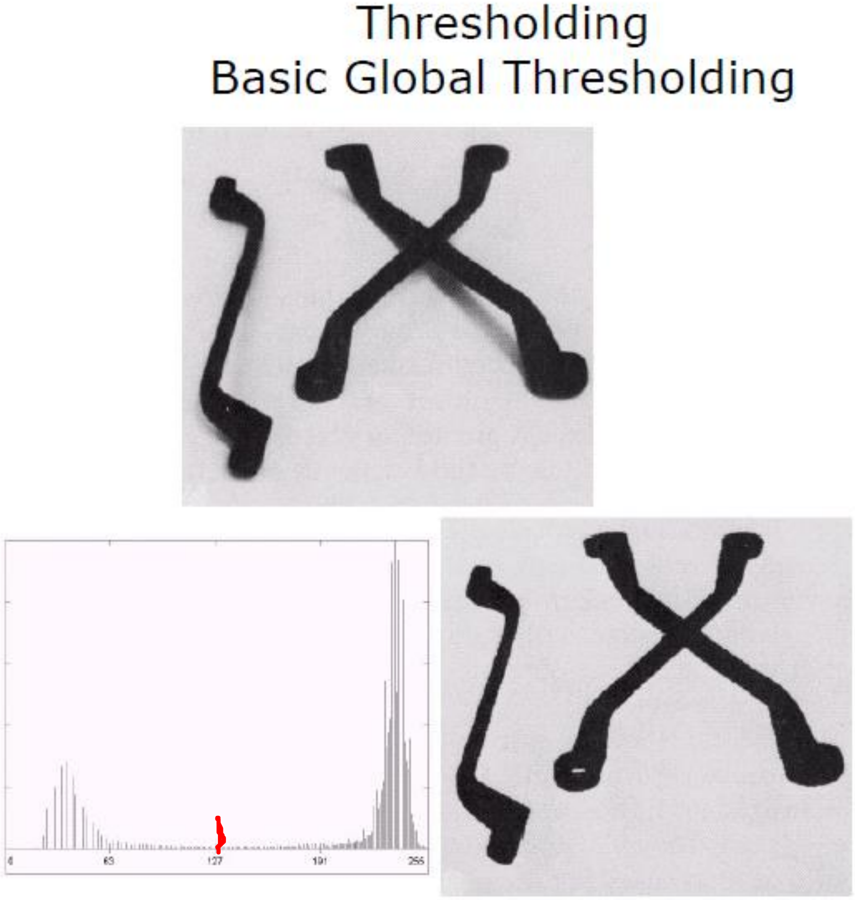
❖ Basic Global Thresholding

- Global thresholding having a single threshold (T) is used when the objects are easily differentiated from each other.



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.)



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.

❖ The Role of Illumination

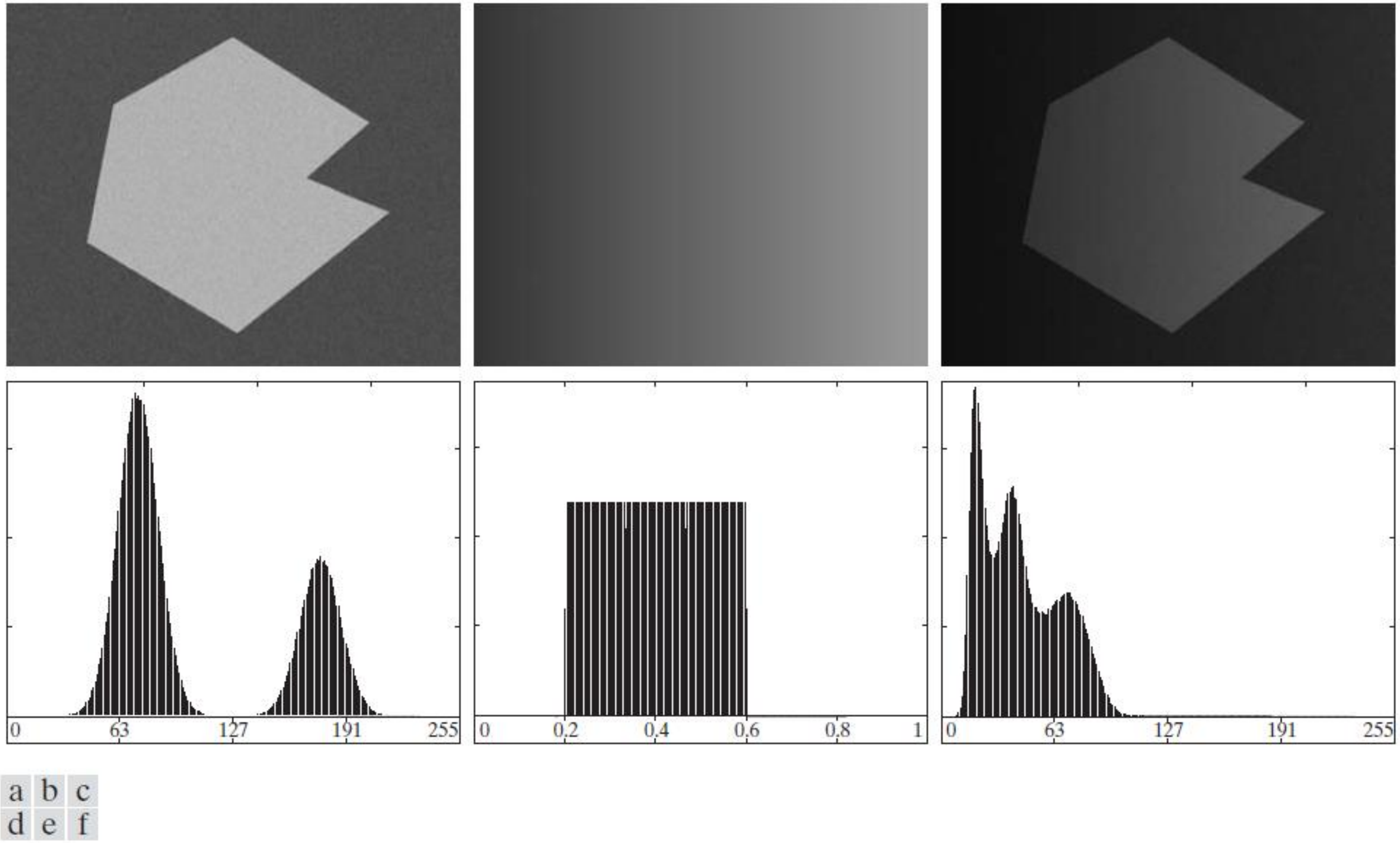


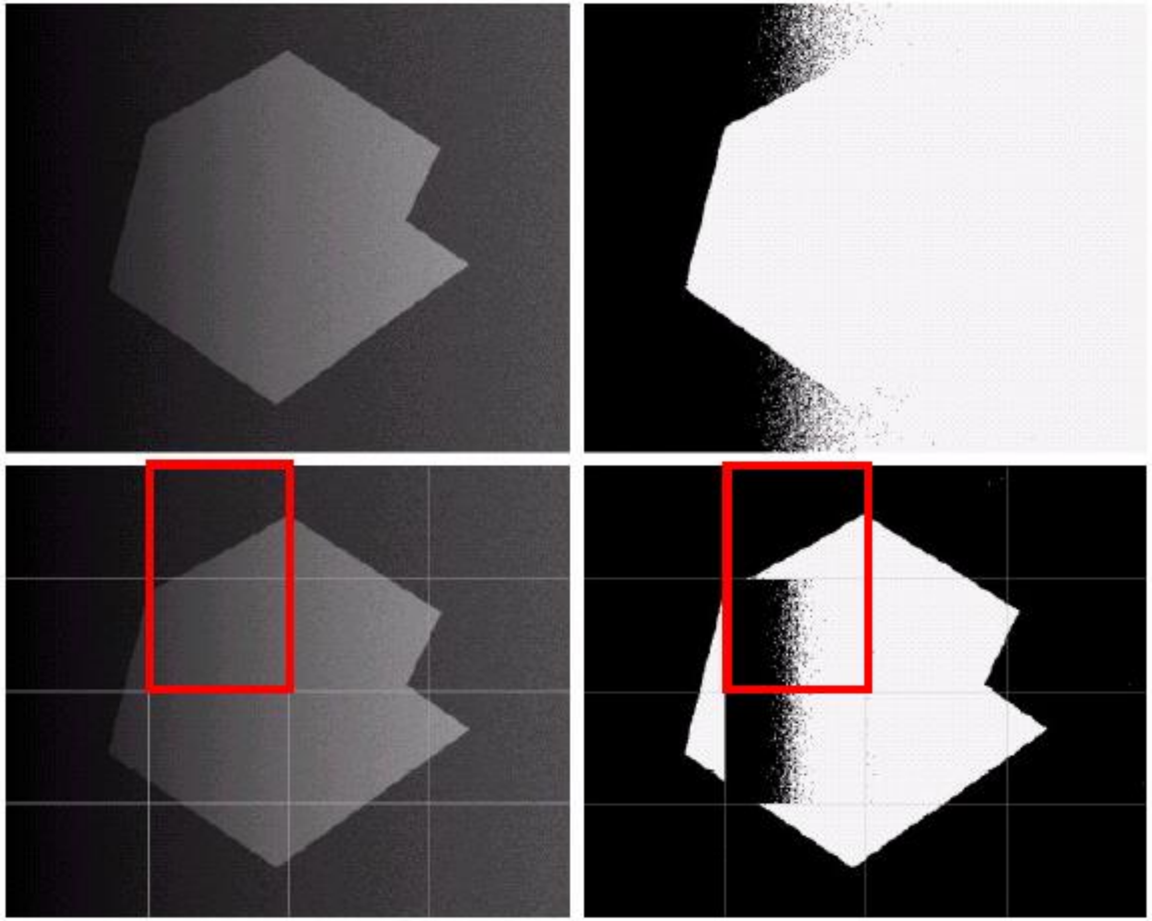
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

❖ Adaptive (Local) thresholding

Thresholding
Basic Adaptive Thresholding

a b
c d

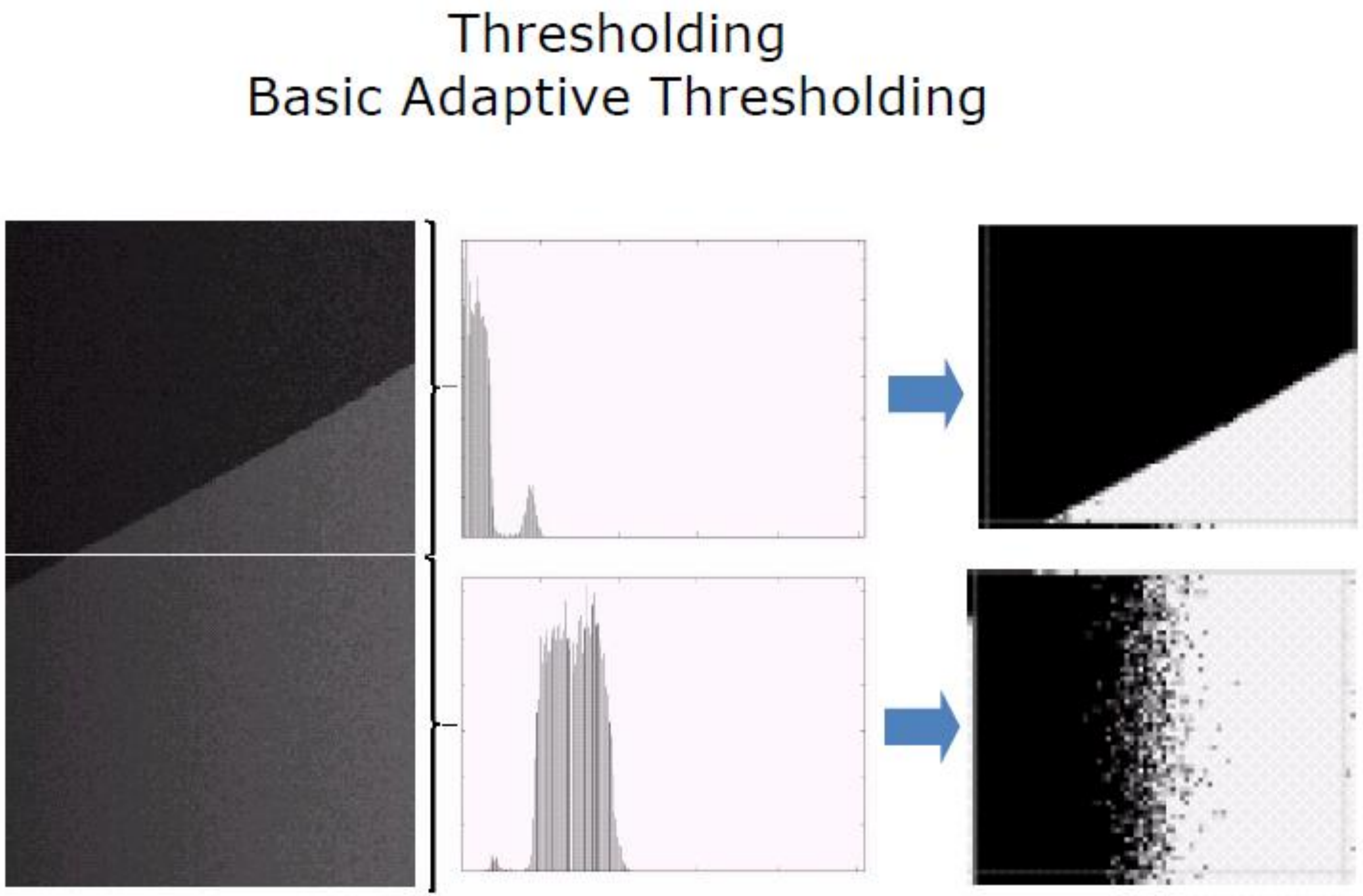
FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



- Global thresholding often fails in the case of uneven illumination.

- The solution is to divide the image into sub-images, and determine T for each sub-image. This method is known as adaptive(local) thresholding.

❖ Adaptive (Local) thresholding

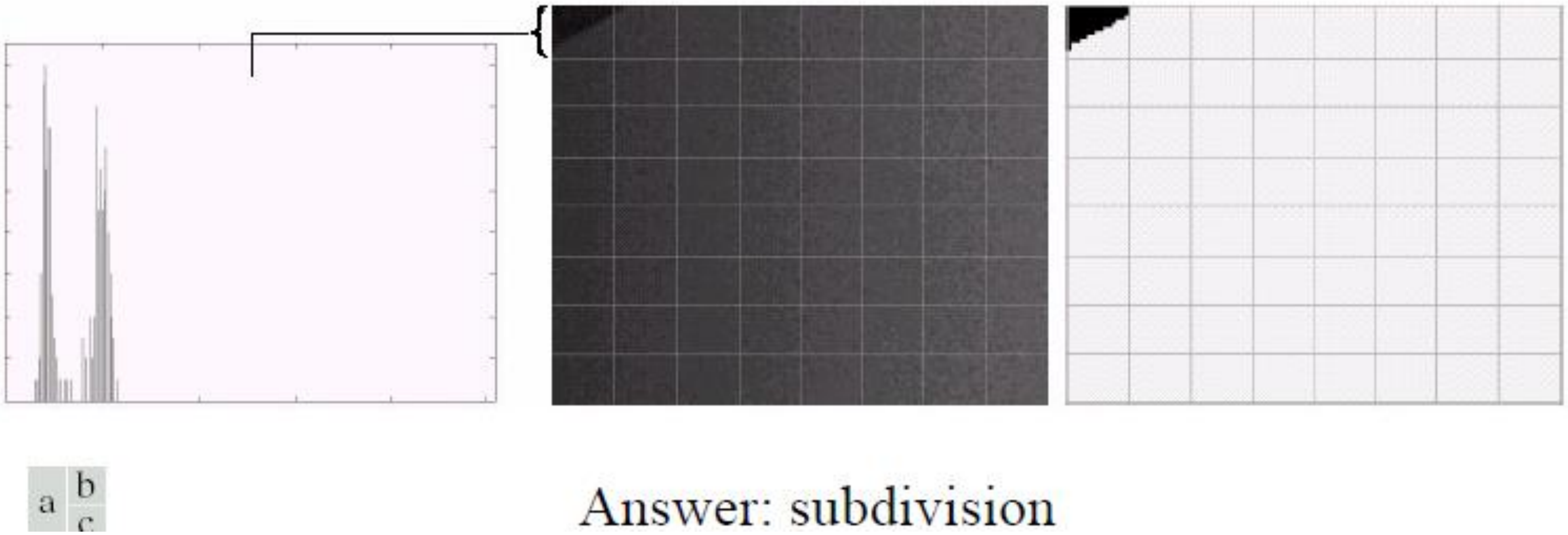


How to solve this problem?

❖ Basic Adaptive Thresholding:

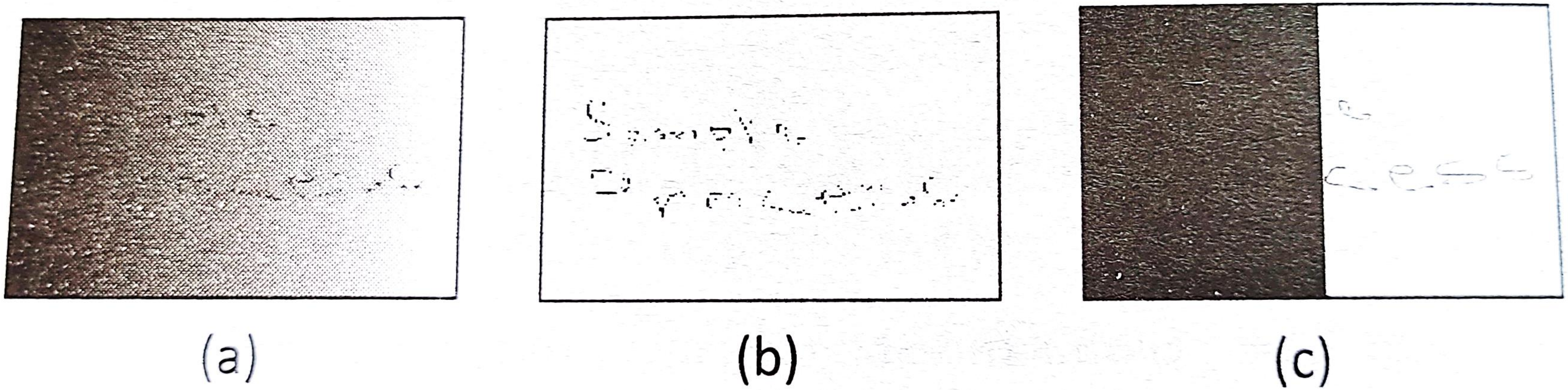
- Subdivide original image into small areas.
- Utilize a different threshold to segment each sub-image.
- As the threshold used for each pixel depends on the location of the pixel in terms of the sub-images, this type of thresholding is adaptive(local).

Thresholding
Basic Adaptive Thresholding



Answer: subdivision

- Further subdivision can improve the quality of adaptive thresholding.
- But this comes at the cost of additional computational complexity.



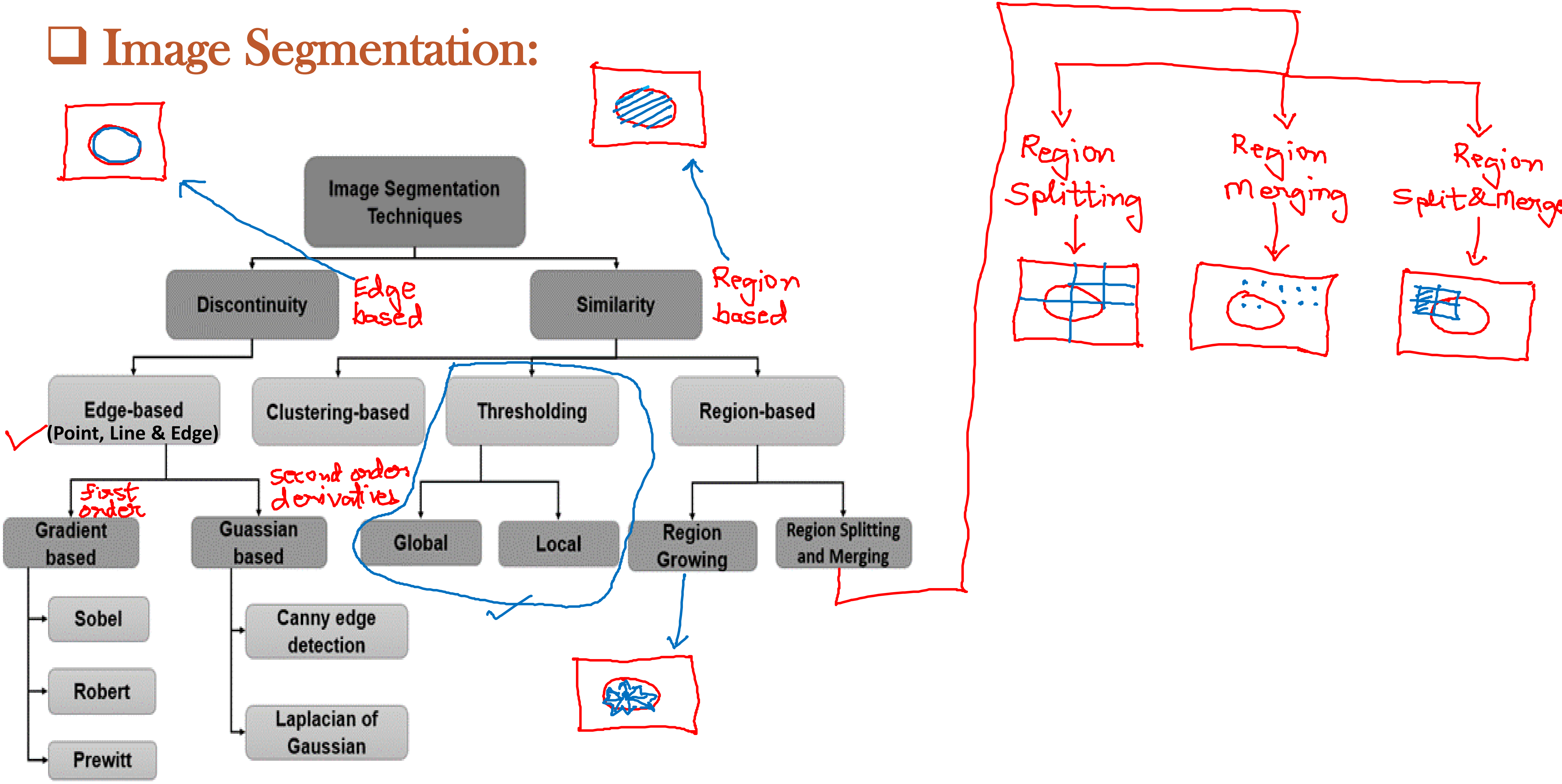
Comparison of thresholding algorithms (a) Original image (b) Result of adaptive algorithm (c) Result of global thresholding algorithm

Thank You

52

Region-Based Segmentation in DIP and its implementation in MATLAB

❑ Image Segmentation:



❖ Why to use Region based segmentation ?

- 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.)

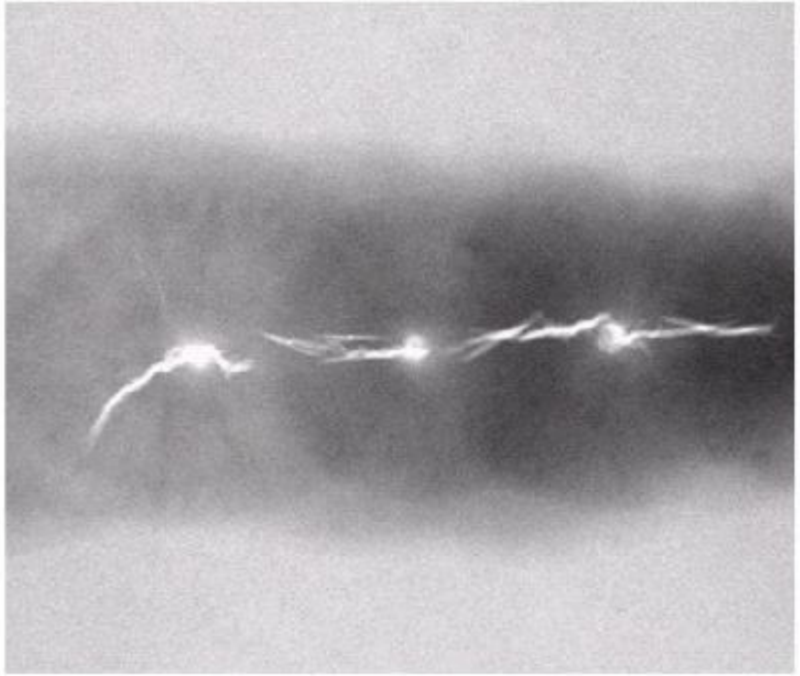


Figure 10.40(a)

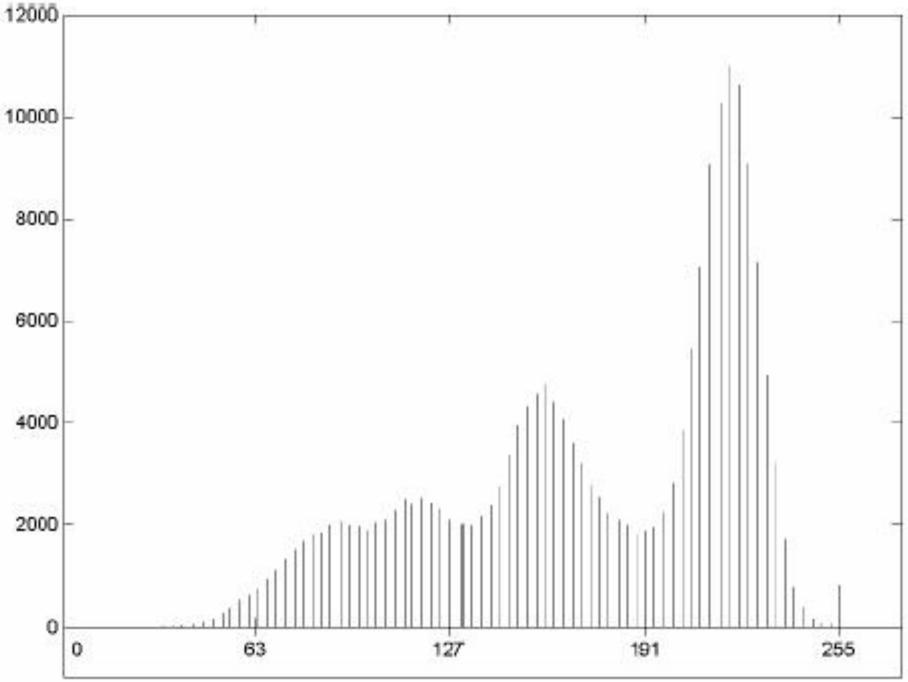
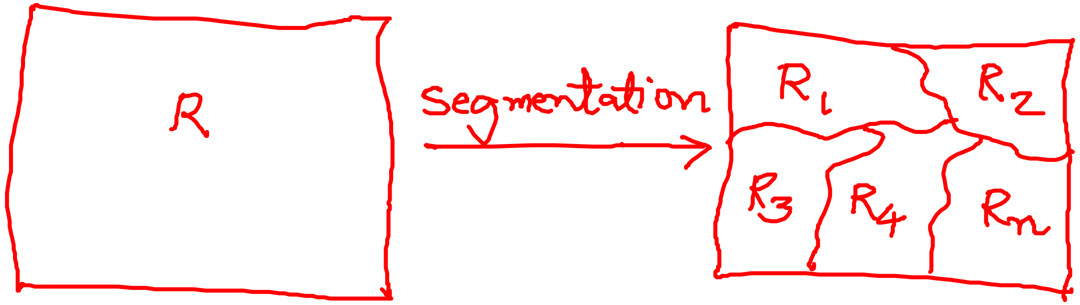


Figure 10.41

❖ 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

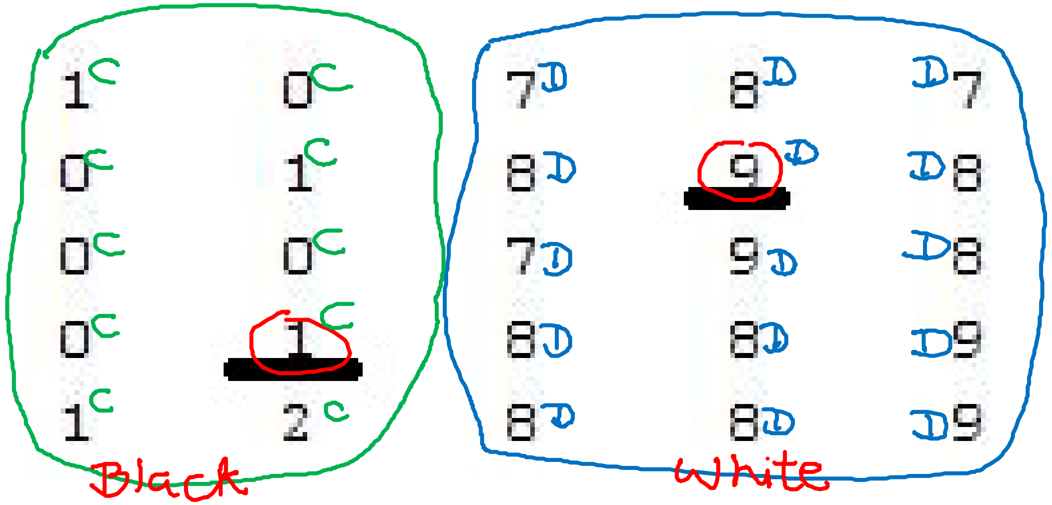
- Edges and thresholds sometimes do not give good results for segmentation. Thresholding still produces isolated image.
- Region growing algorithms works on **principle of similarity**.
- It states that a region is coherent if all the pixels of that region are homogeneous with respect to some characteristics 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 Example

- Consider image shown in figure:



- 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.

Abs |seed value − pixel value| ≤ Threshold

- 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.

❖ Region Splitting

- Entire image is assumed as a single region. Then the homogeneity(similarity) test is applied, where pixels that are similar are grouped together. If the conditions are not met, then the regions are split into four quadrants, else leave the region as it is.

- 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.

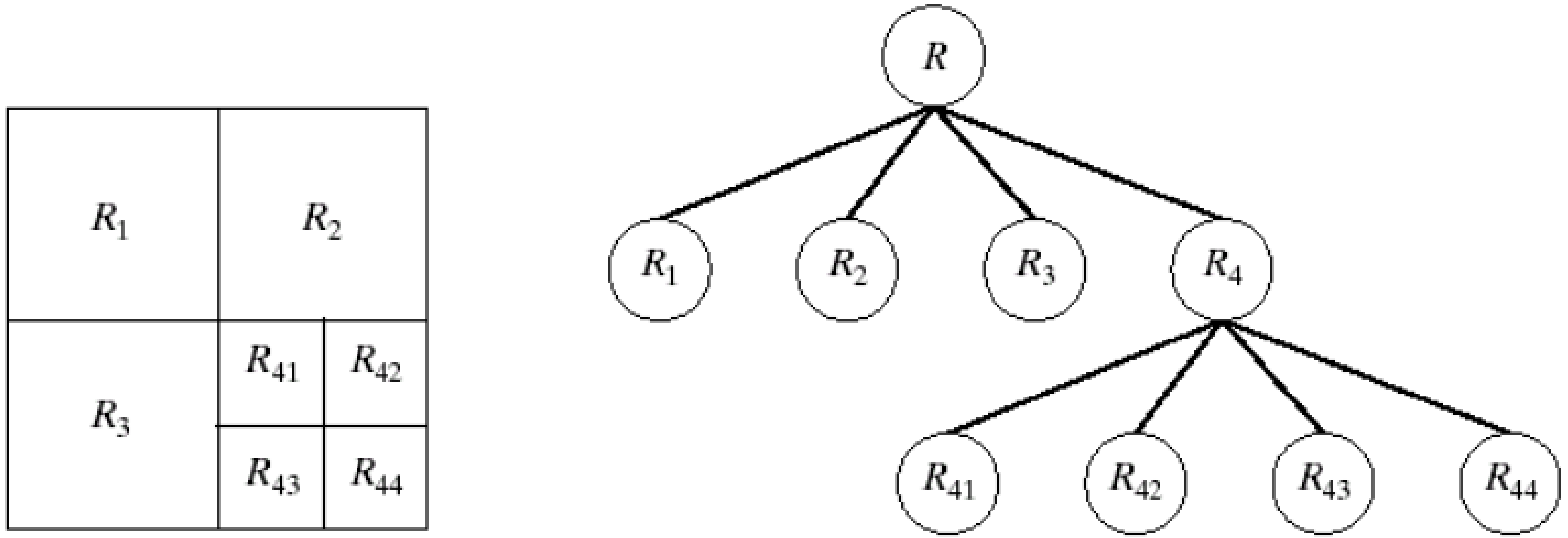
- This process is repeated for each quadrant until all the regions meet the required homogeneity criteria. If the regions are too small, then the division process is stopped.

❖ Region Splitting

- To explain this in terms of graph theory, we call each region a node.
- This technique has a convenient representation in the form of a quadtree structure.
- Quadtree: a tree in which nodes have exactly four descendants.

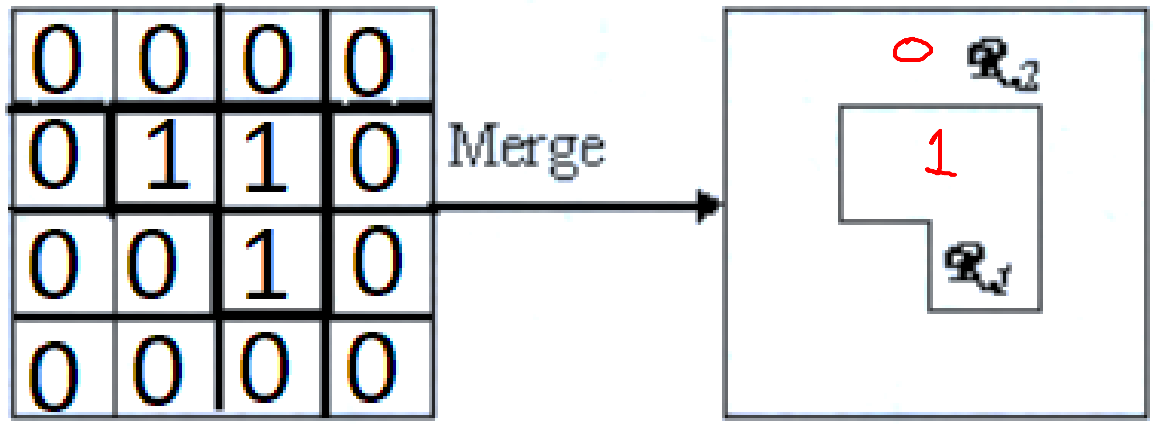
a b

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



❖ Region Merging

- Region merging is opposite to region splitting.
- Here we start from the pixel level and consider each of them as a homogeneous region.
- At any level of merging, we check if the four adjacent regions satisfy the homogeneity property. If yes, they are merged to form a bigger region, otherwise the regions are left as they are.
- This is repeated until no further region exits that requires merging.



Region Splitting and Merging

- Splitting or merging might not produce good results when applied separately. Better results can be obtained by interleaving merge and split operations.
- The split and merge procedure is as follows:
- First there is a large region (possible the entire image).

a) Split into four disjoint quadrants any region R_i for which $P(R_i)$ = FALSE.

b) Merge any adjacent regions R_j and R_k for which $P(R_j \text{ U } R_k)$ = TRUE. (the quadtree structure may not be preserved).

c) Stop when no further merging or splitting is possible.

❖ Region Splitting and Merging Example

T = 3

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

Thank You