CSC447: Digital Image Processing

Chapter 10: Image Segmentation

Prof. Dr. Mostafa Gadal-Haqq M. Mostafa
Computer Science Department
Faculty of Computer & Information Sciences
AIN SHAMS UNIVERSITY

Image Segmentation

- Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image
- Typically the first step in any automated computer vision application

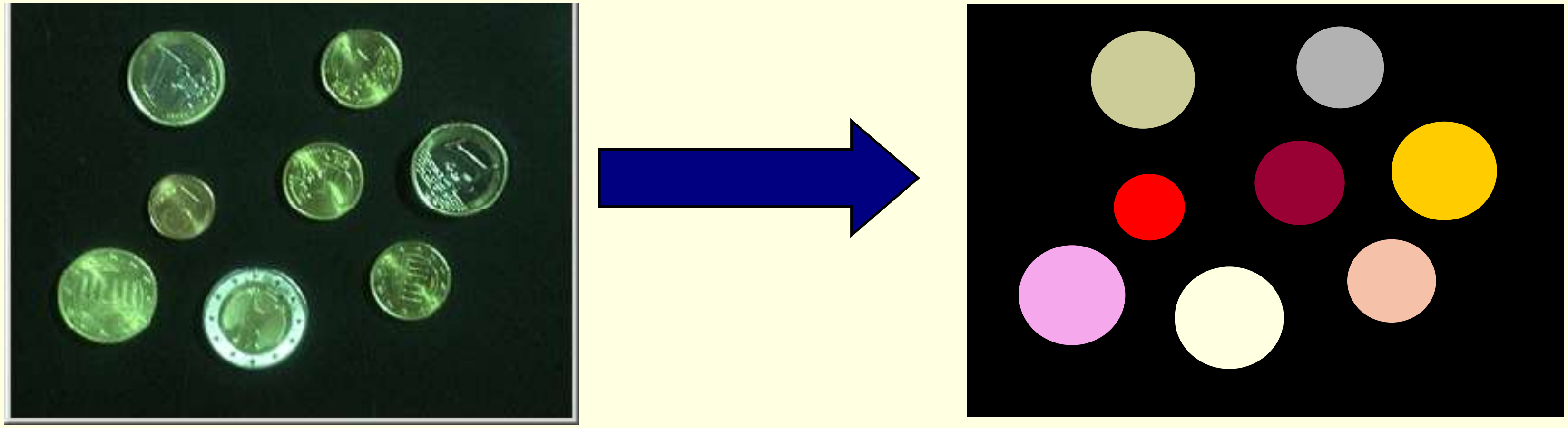


Image Segmentation

- Segmentation algorithms generally are based on one of two basis properties of intensity values
  - Discontinuity: to partition an image based on abrupt changes in intensity (such as edges)
  - Similarity: to partition an image into regions that are similar according to a set of predefined criteria.

Image Segmentation

- Image Segmentation

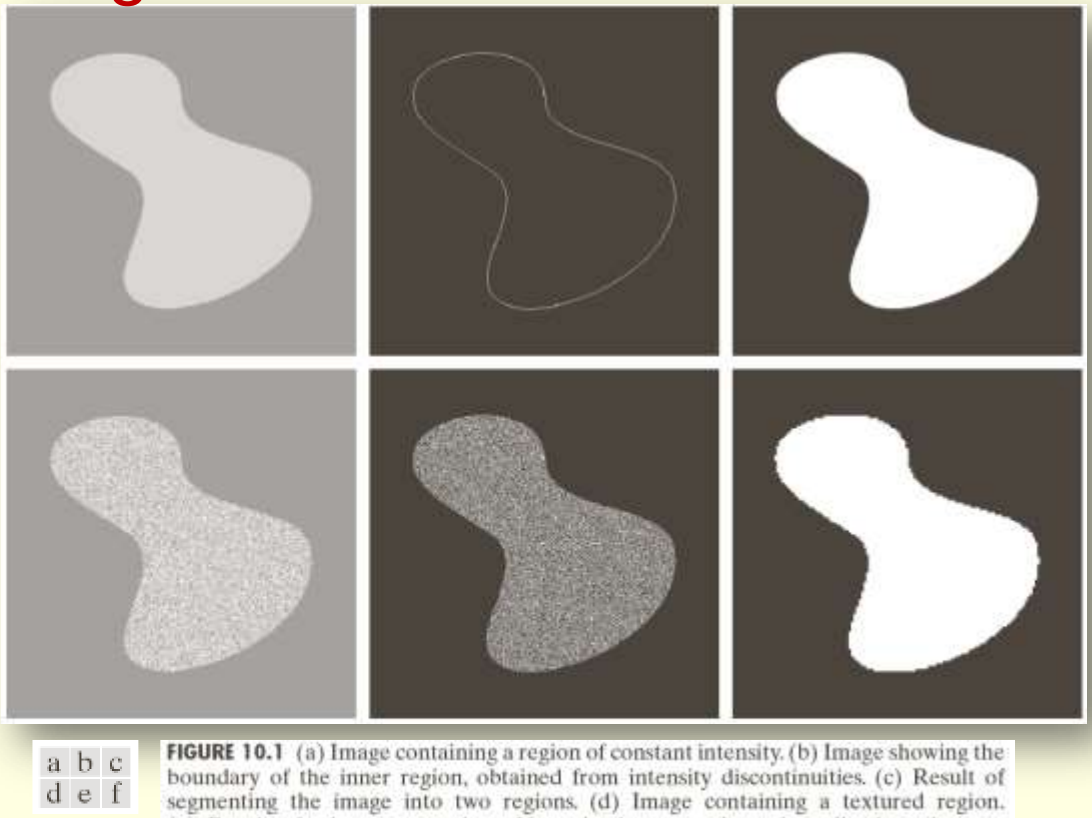


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number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

Image Segmentation

- Image Segmentation

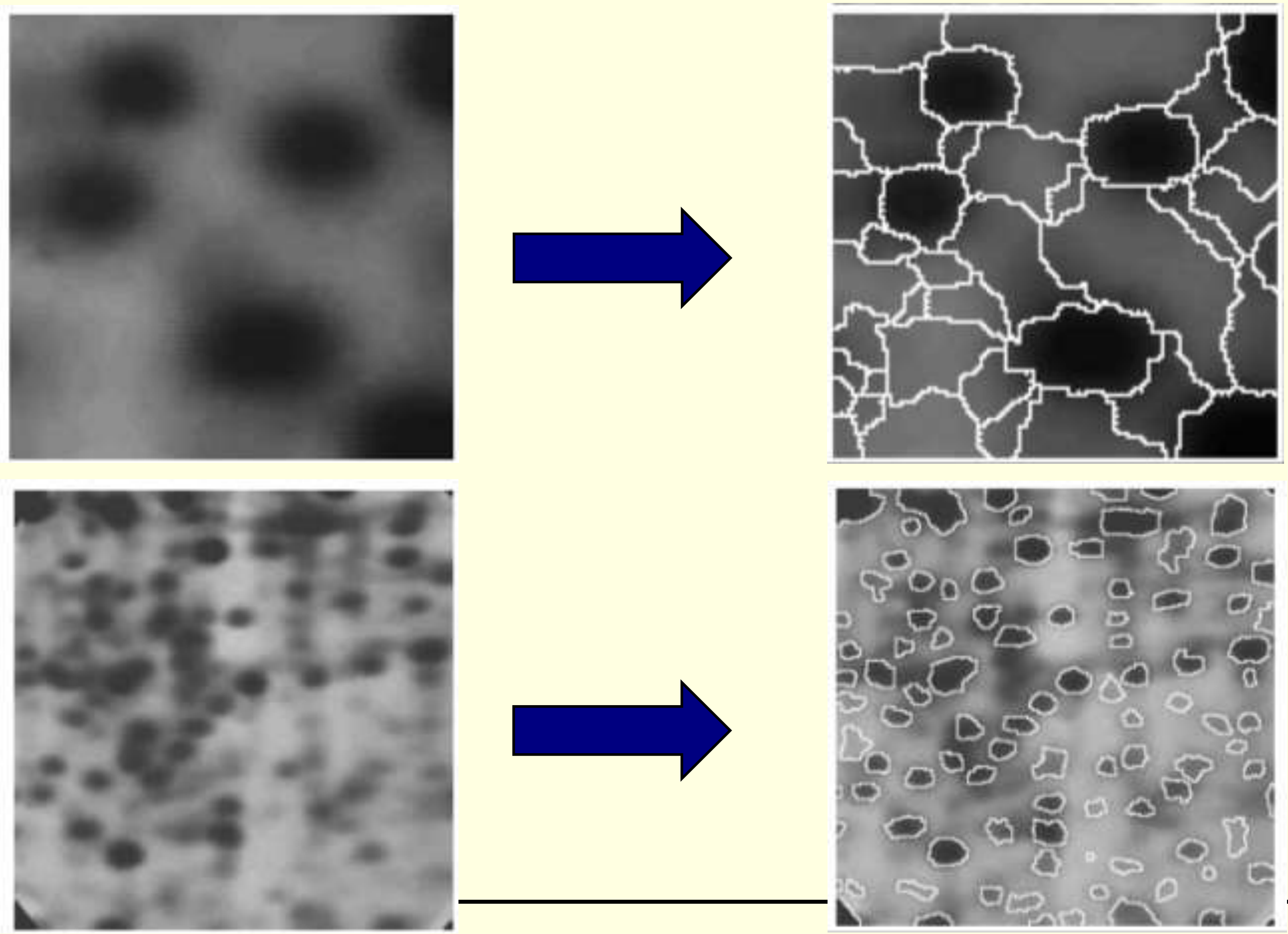


Image Segmentation

- Detection of discontinuities:
  - There are three basic types of gray-level discontinuities:
    - points , lines , edges
  - the common way is to run a mask through the image

Point Detection:

- Note that the mark is the same as the mask of Laplacian Operation (in chapter 3)

- The only differences that are considered of interest are those large enough (as determined by T) to be considered isolated points.

$$|R| > T$$

Point Detection:

1	1	1
1	−8	1
1	1	1

a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Line Detection

−1	−1	−1
2	2	2
−1	−1	−1

R_1 Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

R_2 +45°

−1	2	−1
−1	2	−1
−1	2	−1

R_3 Vertical

−1	−1	2
−1	2	−1
2	−1	−1

R_4 −45°

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

- Horizontal mask will result with max response when a line passed through the middle row of the mask with a constant background.
- the similar idea is used with other masks.
- note: the preferred direction of each mask is weighted with a larger coefficient (i.e.,2) than other possible directions.

Line Detection

- Apply every masks on the image
- let R_1, R_2, R_3, R_4 denotes the response of the horizontal, +45 degree, vertical and -45 degree masks, respectively.
- if, at a certain point in the image

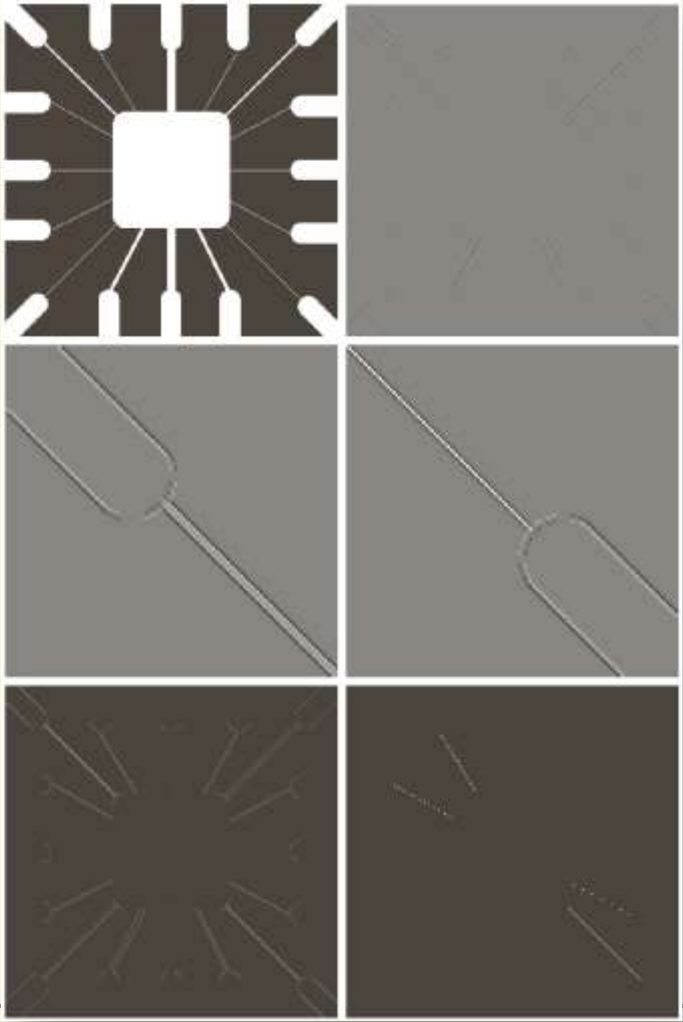
$$|R_i| > |R_j|, \text{ for all } j \neq i,$$

- that point is said to be more likely associated with a line in the direction of mask i.

Line Detection

- Alternatively, if we are interested in detecting all lines in an image in the direction defined by a given mask, we simply run the mask through the image and threshold the absolute value of the result.
- The points that are left are the strongest responses, which, for lines one pixel thick, correspond closest to the direction defined by the mask.

Line Detection



a b
c d
e f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the $+45°$ line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Edge Detection Approach

- Segmentation by finding pixels on a region boundary.
- Edges found by looking at neighboring pixels.
- Region boundary formed by measuring gray value differences between neighboring pixels

Edge Detection

- 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 Detection

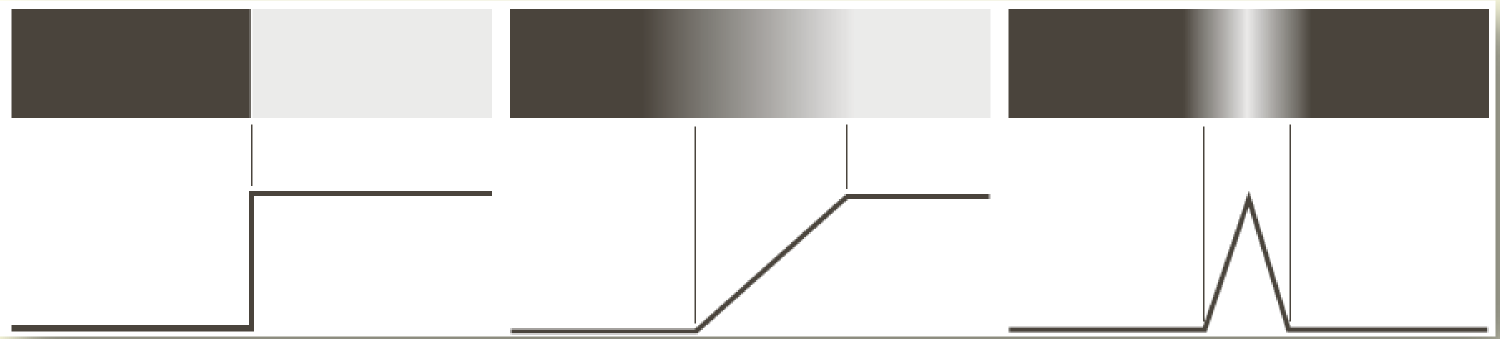


FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

Edge Detection

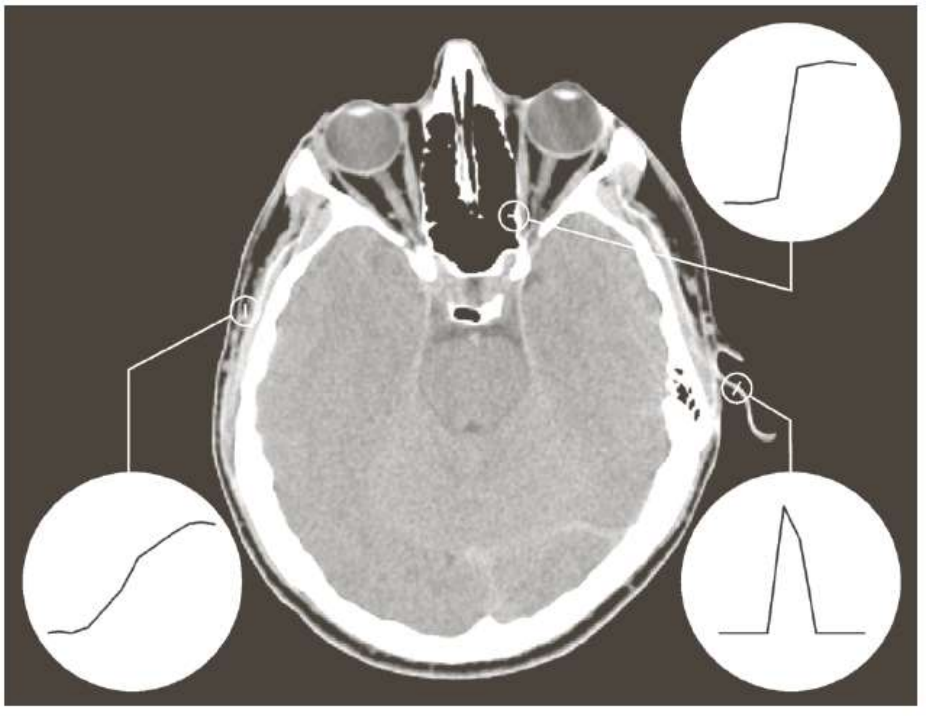
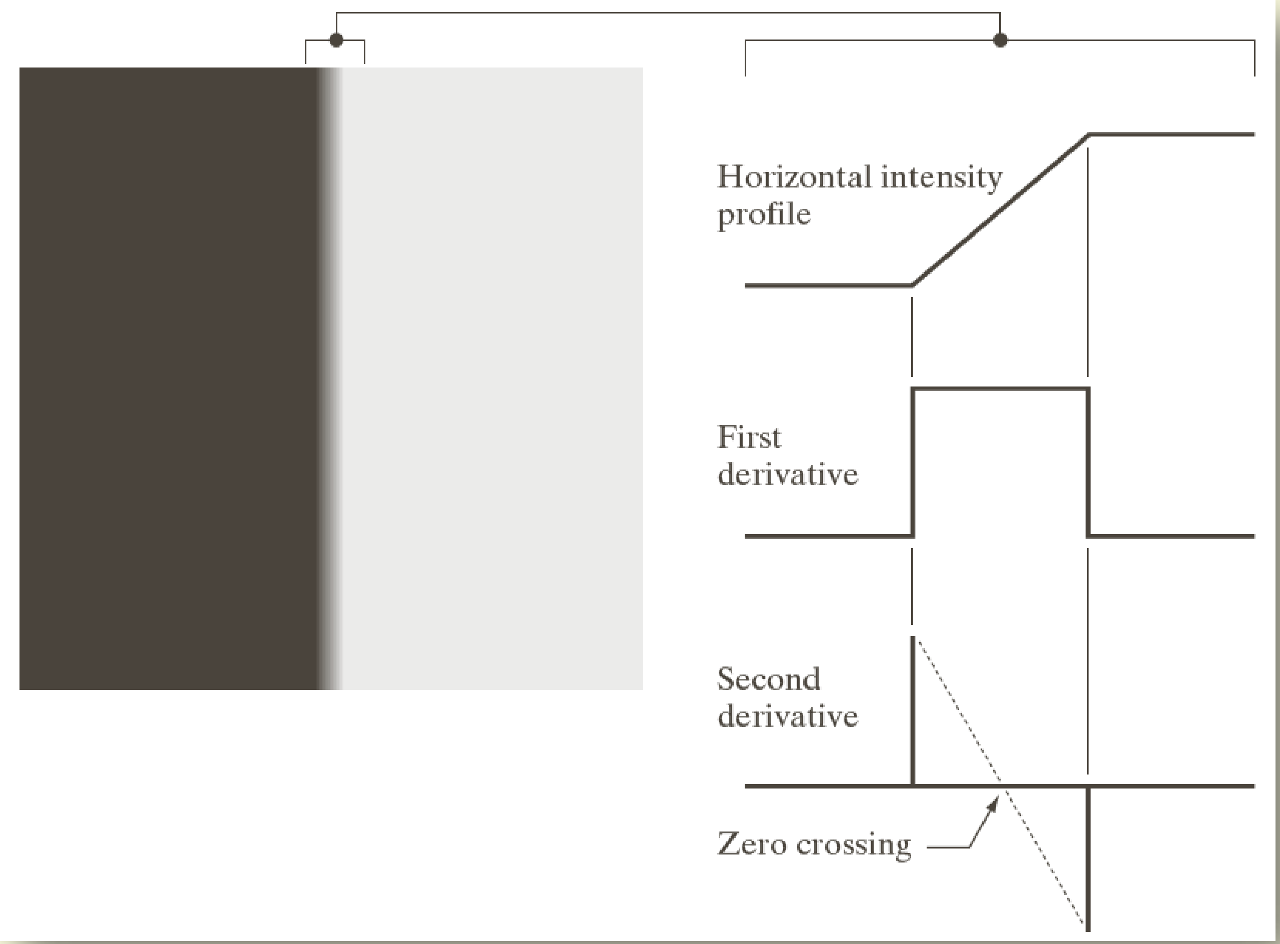


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Edge Detection

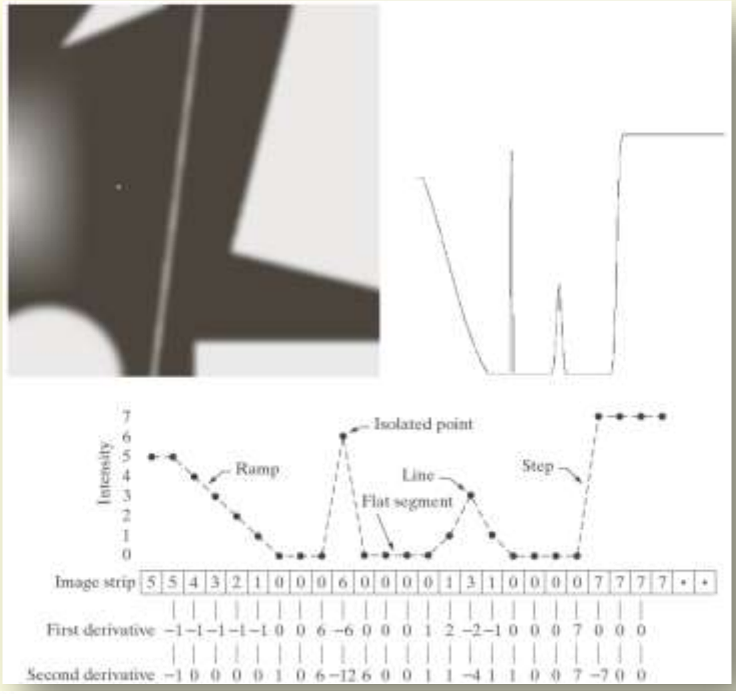


a b

FIGURE 10.10
(a) Two regions of constant intensity separated by an ideal vertical ramp edge.
(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Edge Detection

- Detection of discontinuities: Image Derivatives



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

Edge Detection

- 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.

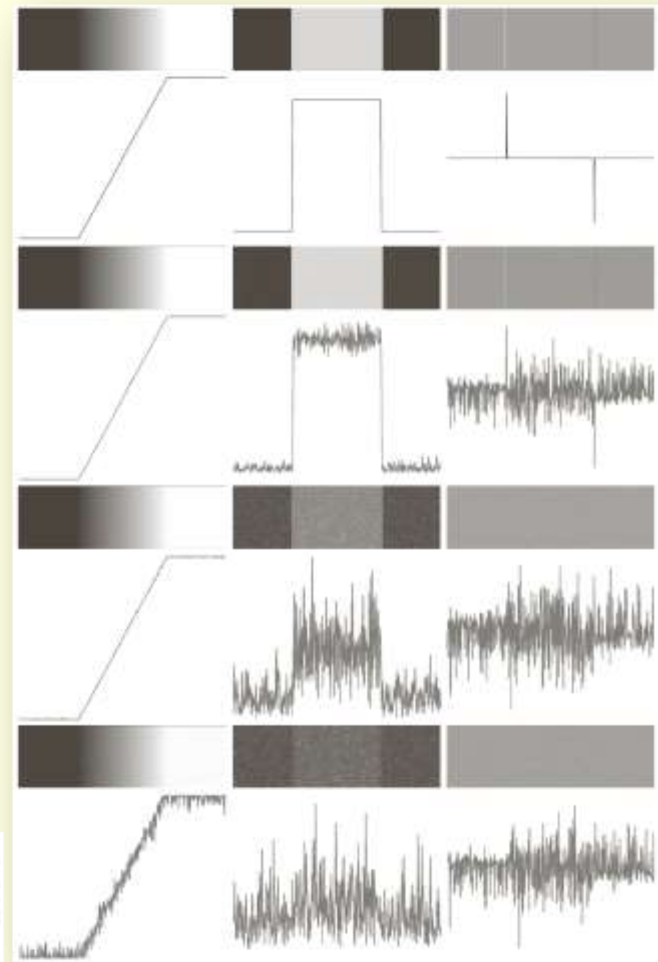


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images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

Edge Detection

- Gradient Operator

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

$$|\nabla f| = \left[G_x^2 + G_y^2\right]^{1/2}$$

$$\text{for } 3\times 3 \text{ mask}$$

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

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		−1	0	1
0	0	0		−1	0	1
1	1	1		−1	0	1

Prewitt

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

Sobel

a	
b	c
d	e
f	g

FIGURE 10.14 A 3 × 3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.

Edge Detection

- Prewitt and Sobel Operators

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a	b
c	d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

Edge Detection

a b
c d

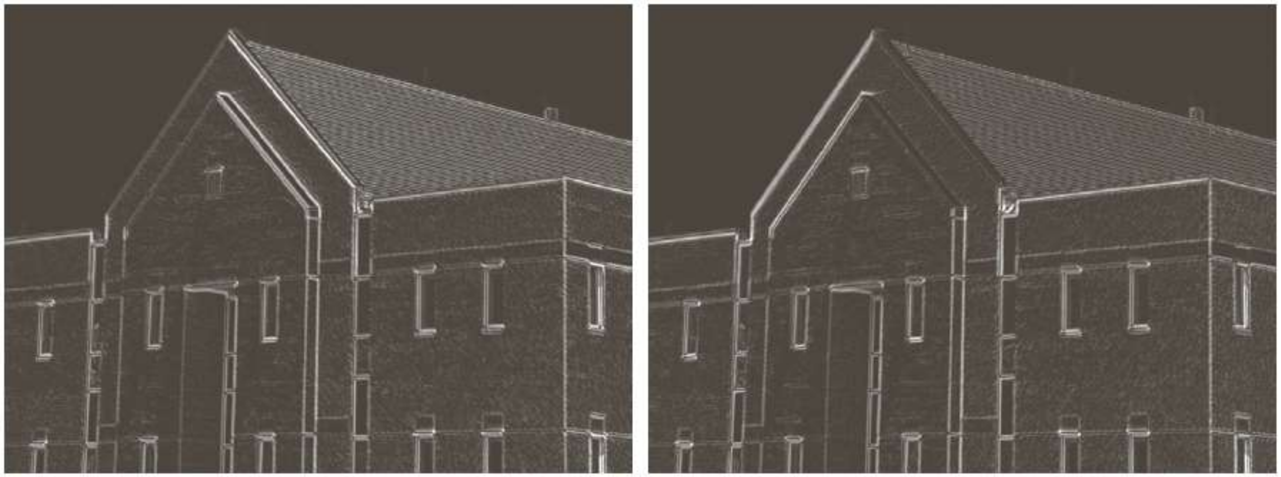
FIGURE 10.16 (a) Original image of size 834×1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The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

Edge Detection

a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.

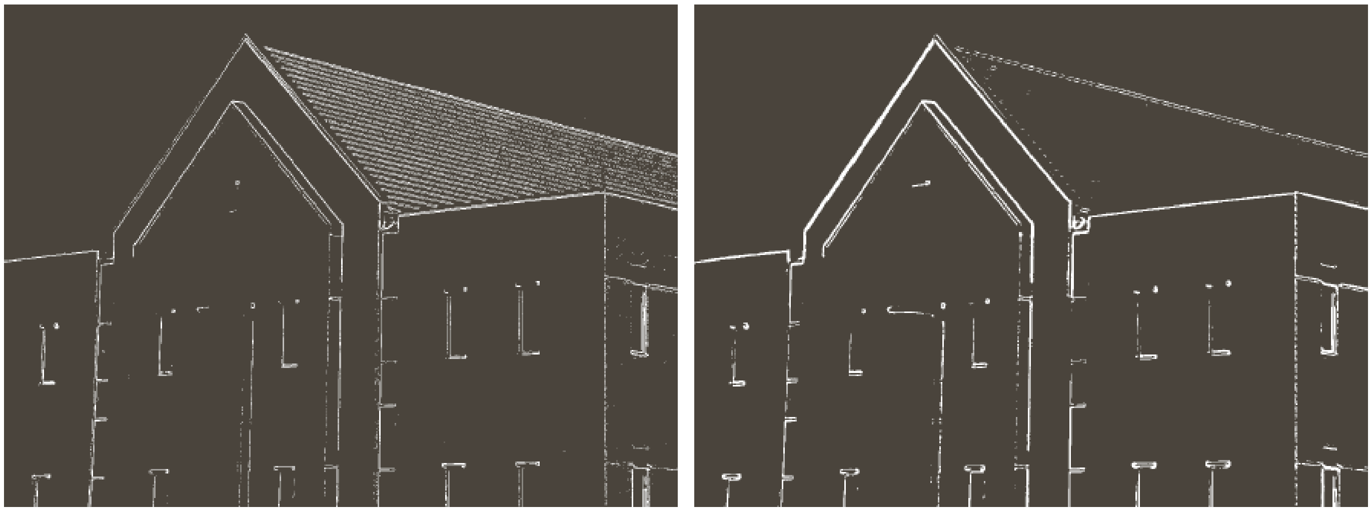
Edge Detection



a b

FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

Edge Detection



a b

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

Edge Detection

- The Laplacian

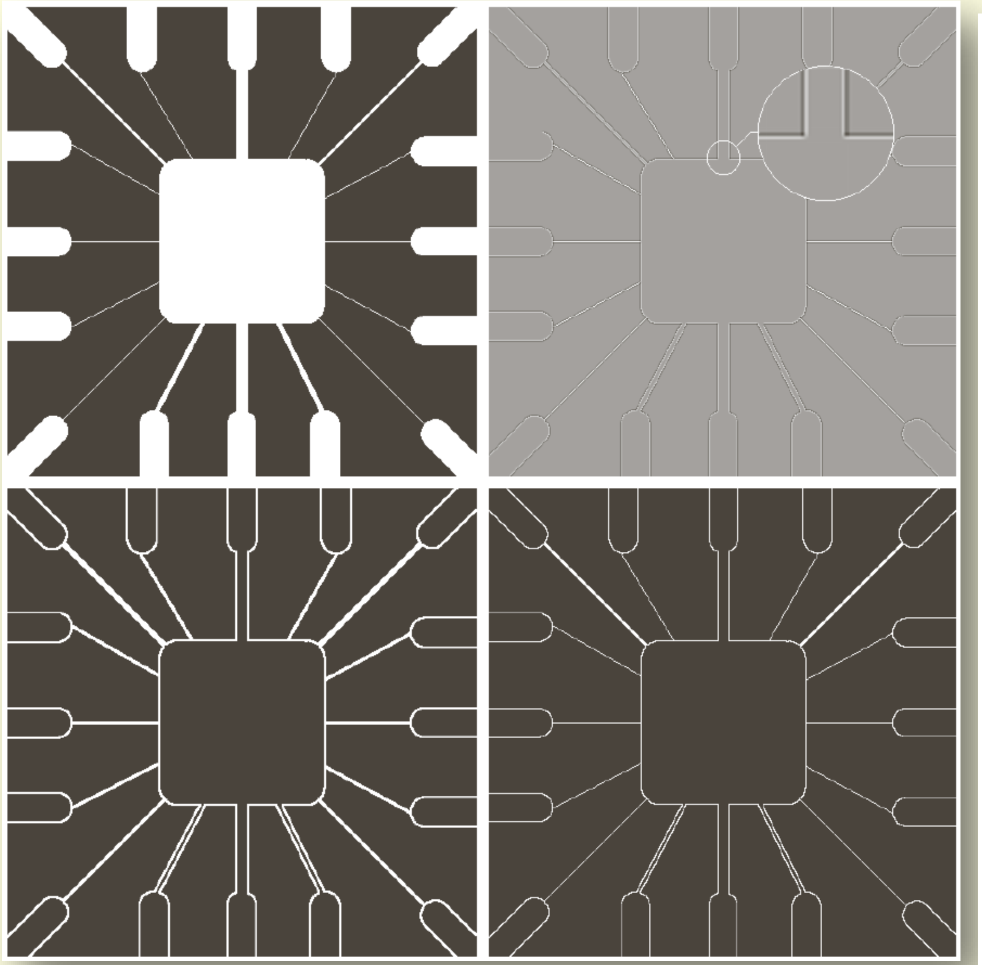
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

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

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

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

Edge Detection

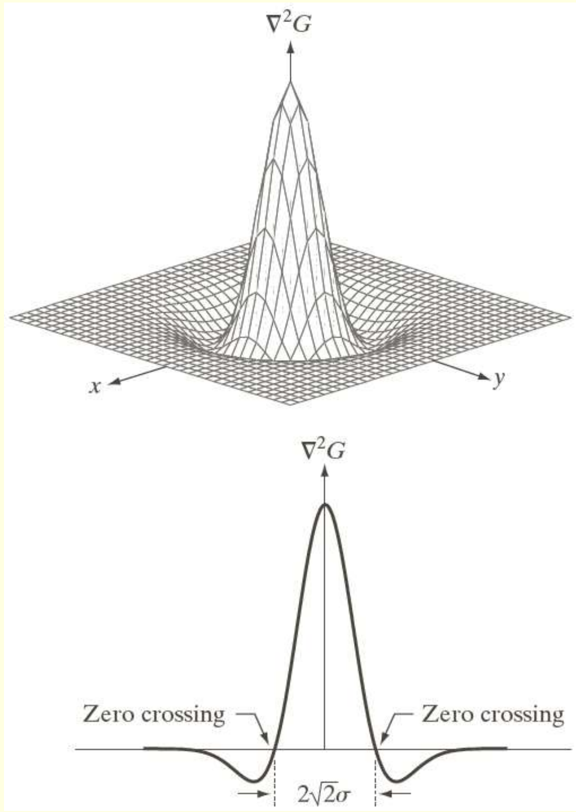


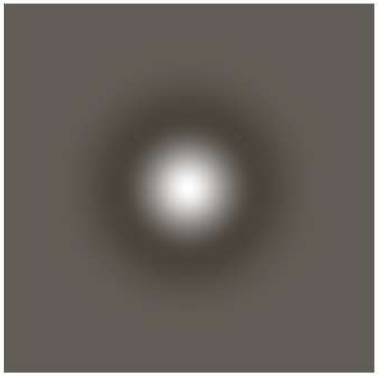
a b
c d

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

Edge Detection

- The Laplacian of Gaussian (LoG)





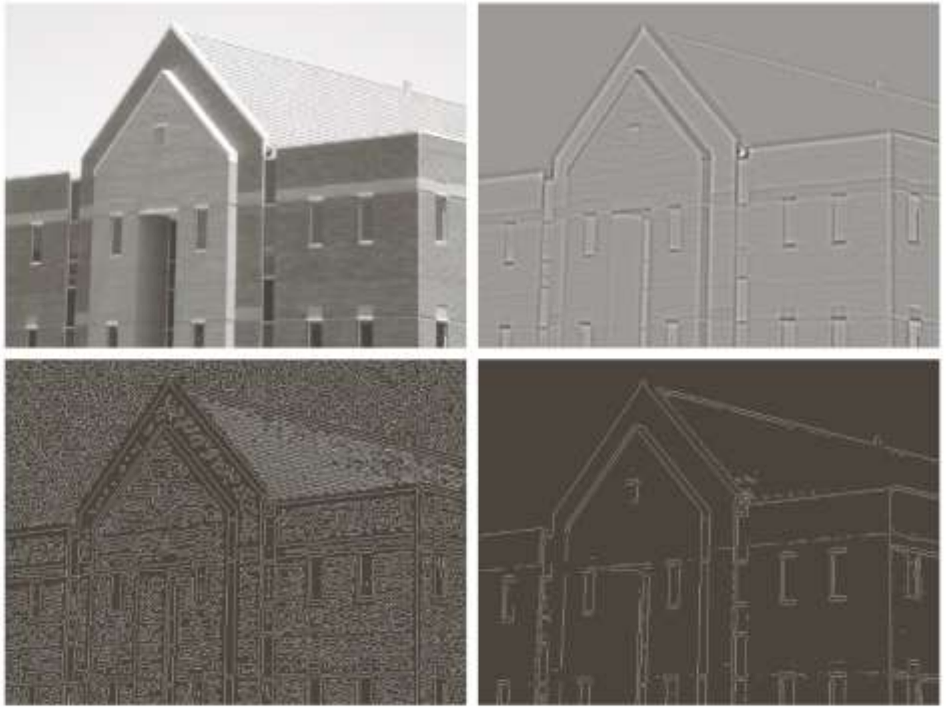
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

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.

Edge Detection

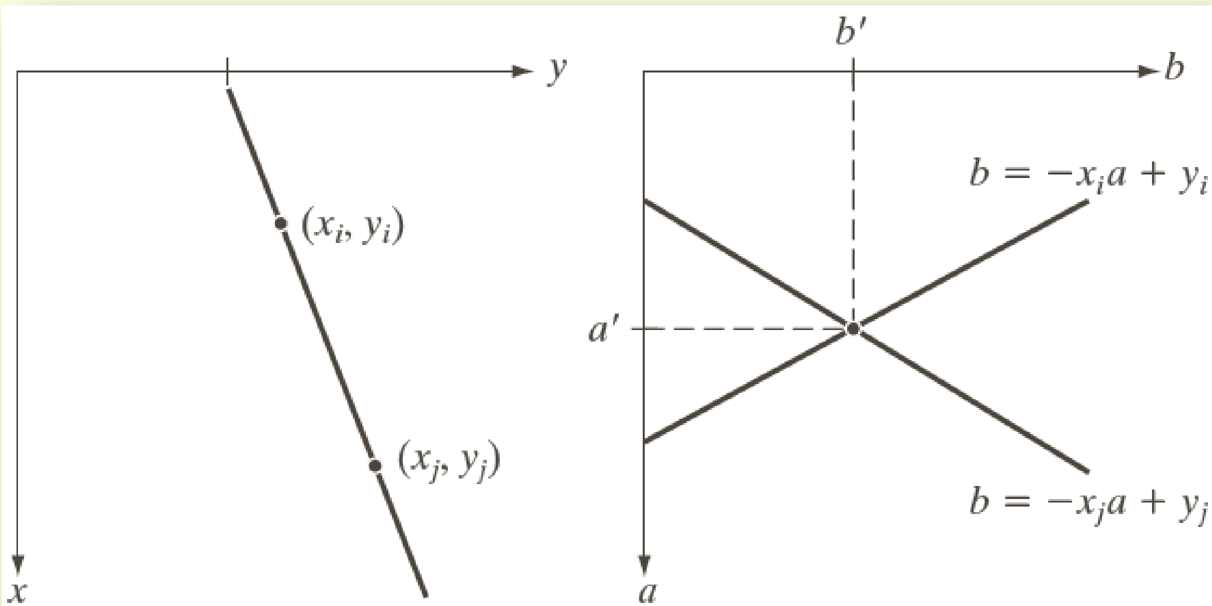
- The Laplacian of Gaussian (LoG)



a b
c d

FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

The Hough Transform

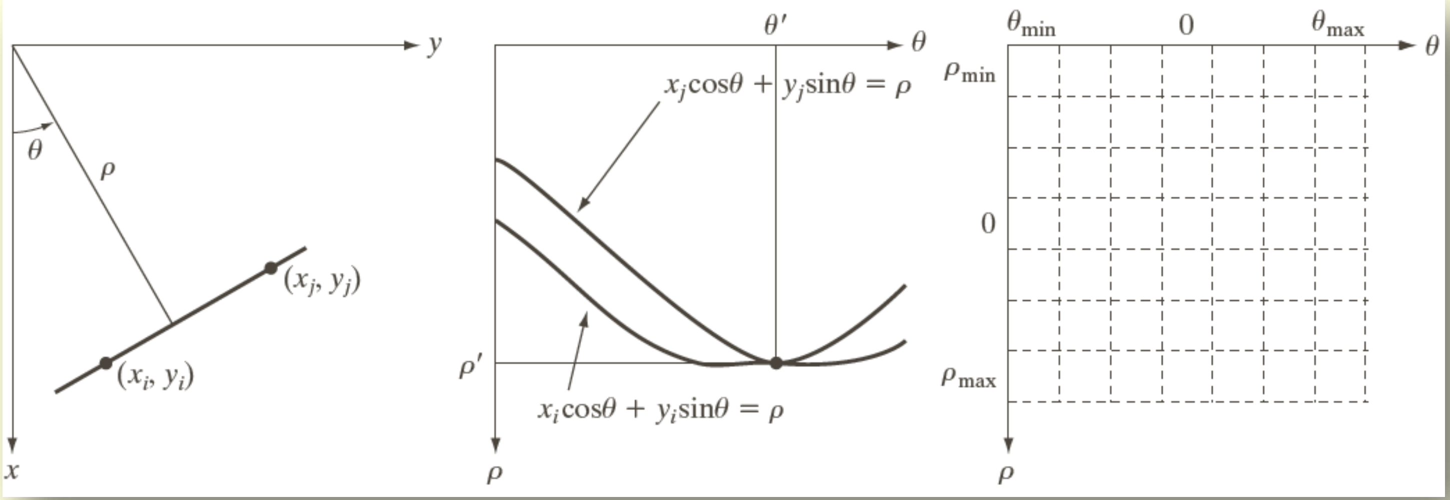


a b

FIGURE 10.31 (a) *xy*-plane. (b) Parameter space.

The Hough Transform

- Global processing: The Hough Transform

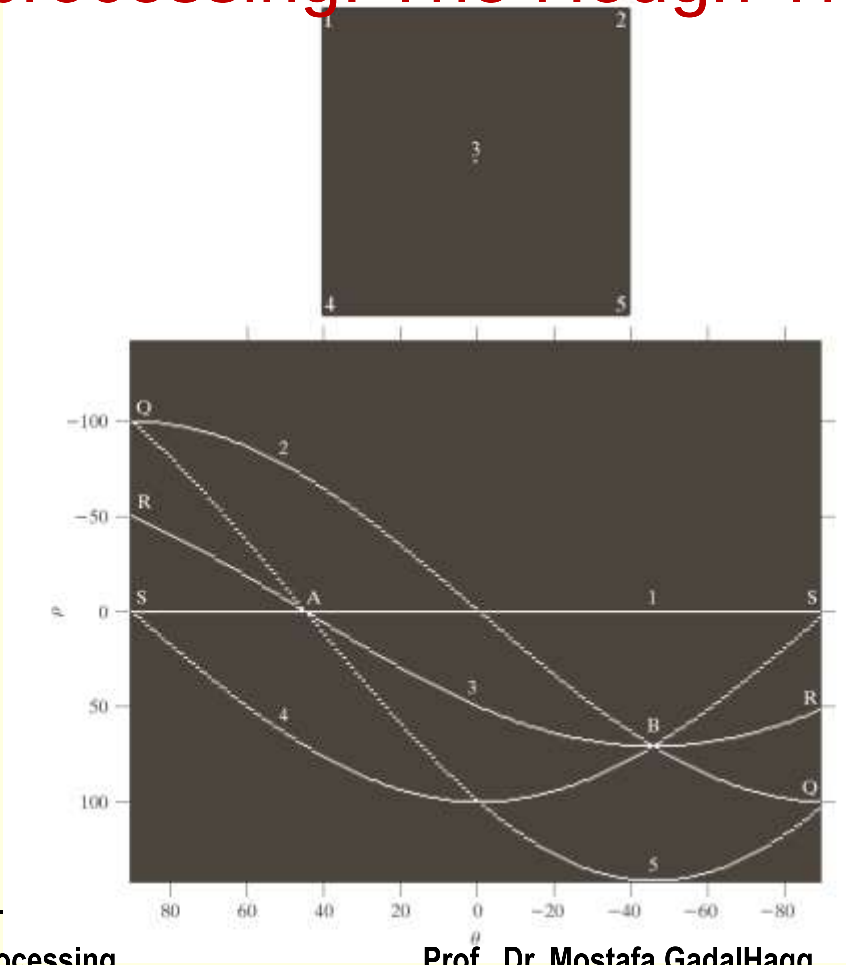


a b c

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.

The Hough Transform

- Global processing: The Hough Transform



a
b

FIGURE 10.33 (a) Image of size 101 × 101 pixels, containing five points. (b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

The Hough Transform

- Global processing: The Hough Transform

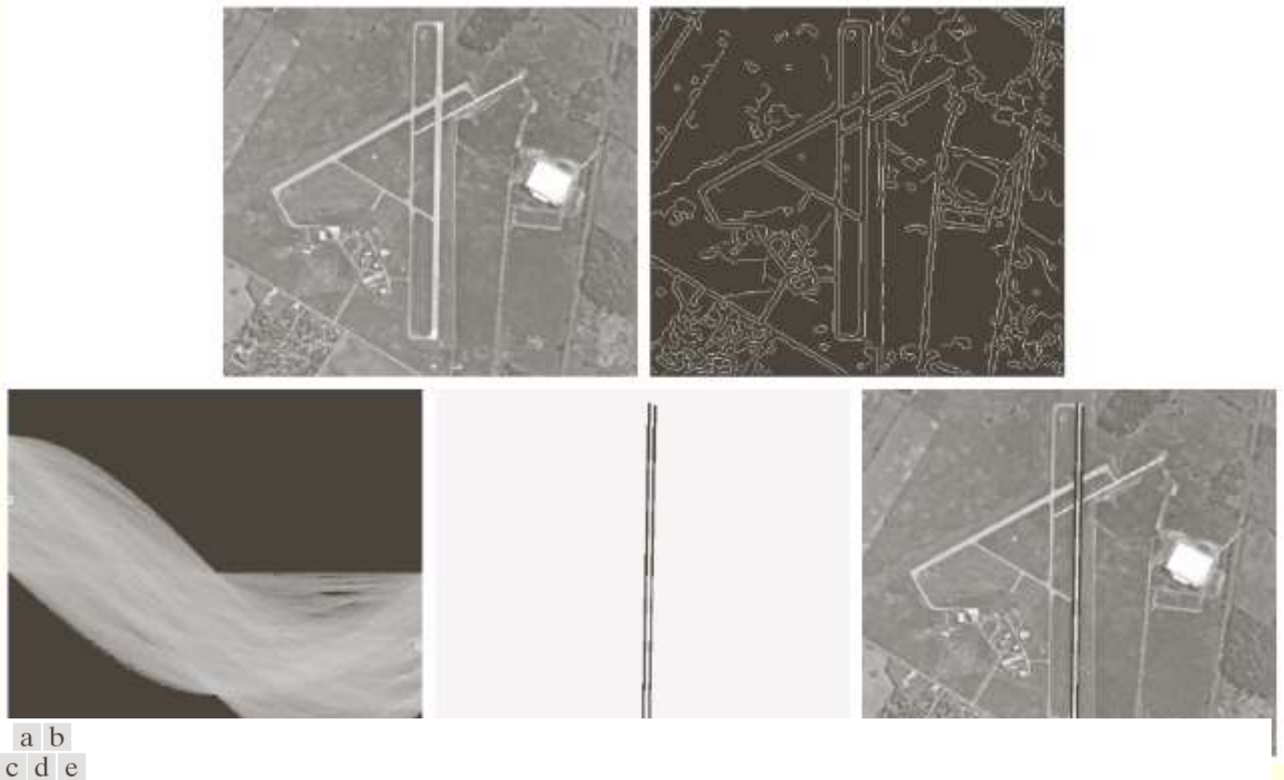


FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed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

Region-Based Segmentation

What is a Region?

- Basic definition :- A group of connected pixels with similar properties.

- Important in interpreting an image because they may correspond to objects in a scene.
  - For that an image must be partitioned into regions that correspond to objects or parts of an object.

Region-Based vs. Edge-Based

Region-Based	Edge-Based
Closed boundaries	Boundaries formed not necessarily closed
Multi-spectral images improve segmentation	No significant improvement for multi-spectral images
Computation based on similarity	Computation based on difference

Image Thresholding

- What is thresholding?
- Simple thresholding
- Adaptive thresholding

Thresholding – A Key Aspect

- Most algorithms involve establishing a threshold level of certain parameter.
- Correct thresholding leads to better segmentation.
- Using samples of image intensity available, appropriate threshold should be set automatically in a robust algorithm i.e. no hard-wiring of gray values

Automatic Thresholding

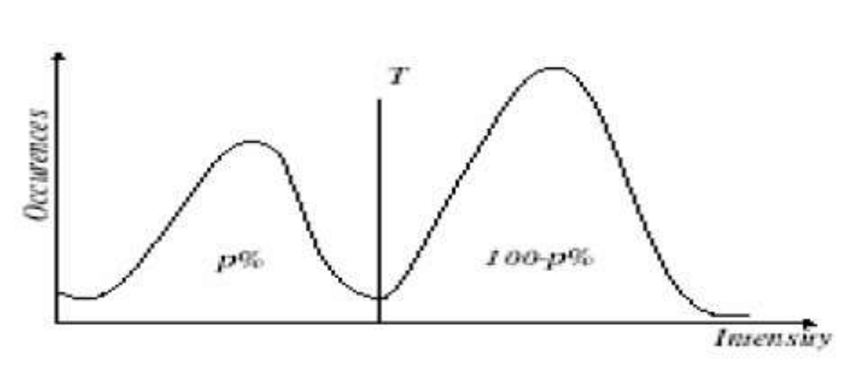
- Use of one or more of the following:-
  1. Intensity characteristics of objects
  2. Sizes of objects
  3. Fractions of image occupied by objects
  4. Number of different types of objects
- Size and probability of occurrence – most popular
- Intensity distributions estimate by histogram computation.

Automatic Thresholding Methods

- Some automatic thresholding schemes:
  1. P-tile method
  2. Iterative threshold selection
  3. Adaptive thresholding

Thresholding Methods

- P-tile Method:- If object occupies P% of image pixels then set a threshold T such that P% of pixels have intensity below T.



- Iterative Thresholding:-

  Successively refines an approx. threshold to get a new value which partitions the image better.

$$T = \frac{1}{2}(\mu 1 + \mu 2)$$

P-Tile Thresholding

- Thresholding is usually the first step in any segmentation approach
- Single value thresholding can be given mathematically as follows:

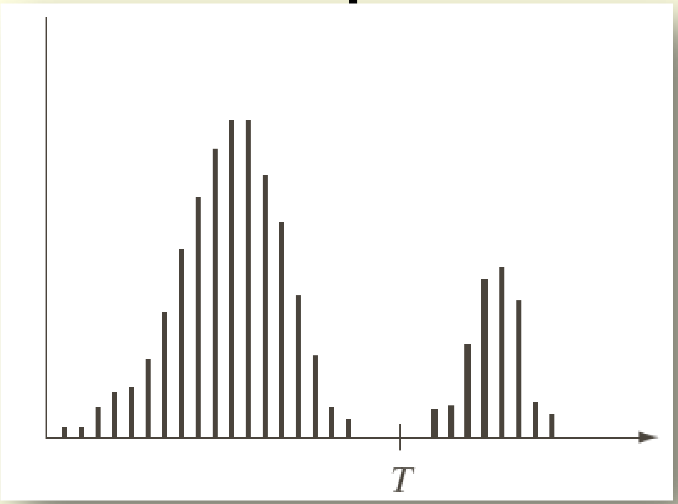
$$g(x,y) = \begin{cases} 1 & \text{if} \quad f(x,y) > T \\ 0 & \text{if} \quad f(x,y) \le T \end{cases}$$

P-Tile Thresholding

- Basic global thresholding:
  - Based on the histogram of an image Partition the image histogram using a single global threshold

P-Tile Thresholding

- Basic global thresholding:
  - The success of this technique very strongly depends on how well the histogram can be partitioned



Iterative P-Tile Thresholding

- The Basic global thresholding:
  1. Select an initial estimate for T (typically the average grey level in the image)
  2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels $\leq$ T
  3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

Iterative P-Tile Thresholding

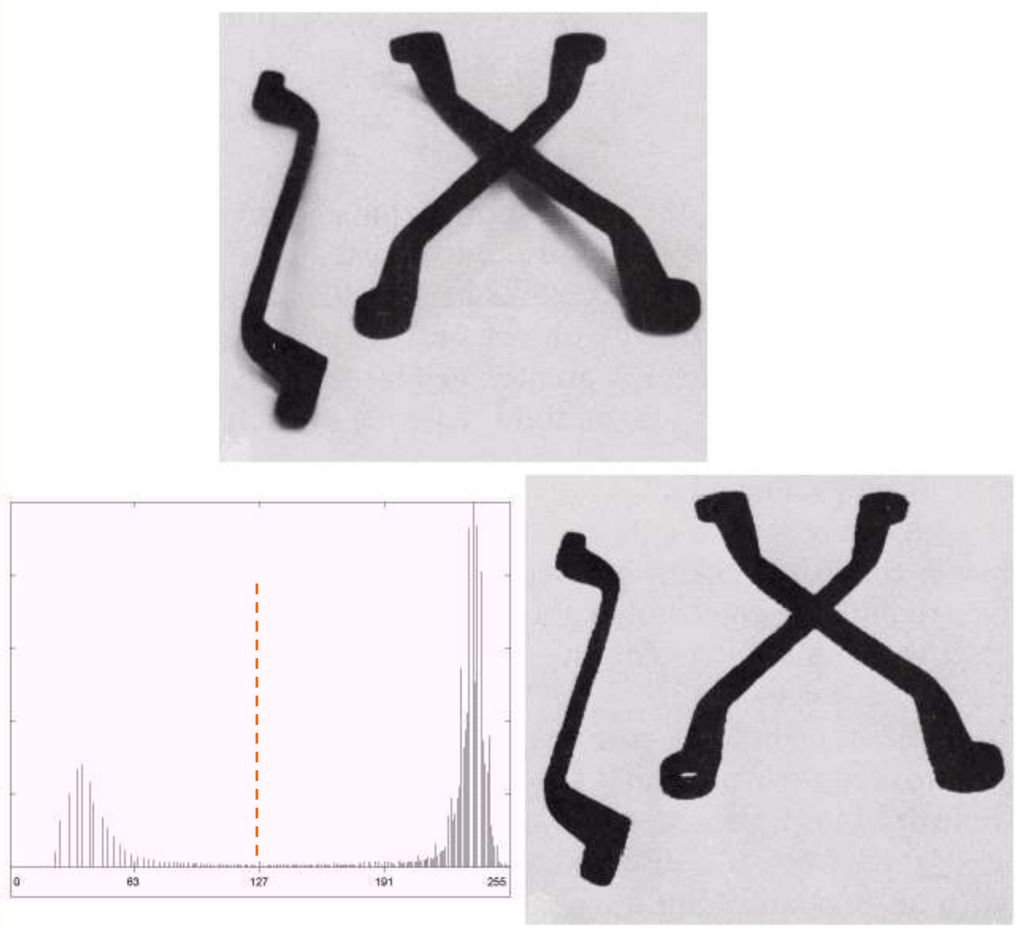
- The Basic global thresholding:
  4. Compute a new threshold value:

$$T = \frac{\mu_1 + \mu_2}{2}$$

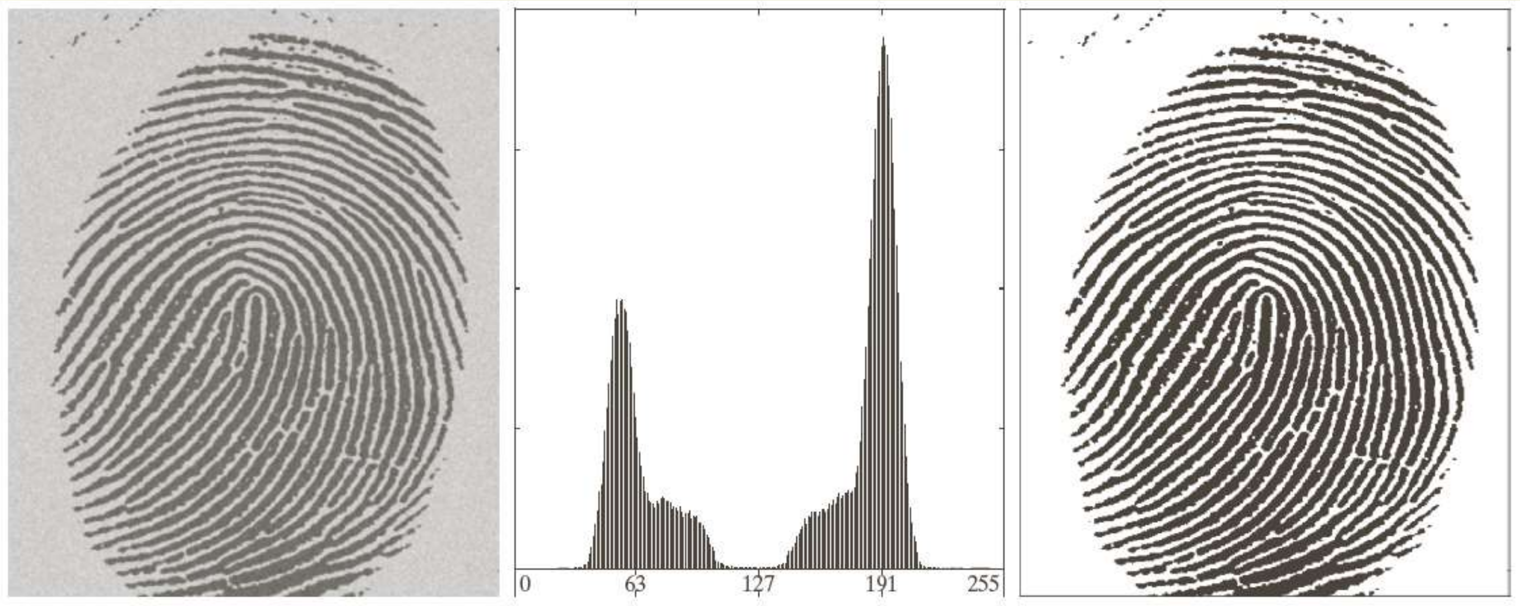
  5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞

This algorithm works very well for finding thresholds when the histogram is suitable.

P-Tile Thresholding



P-Tile Thresholding

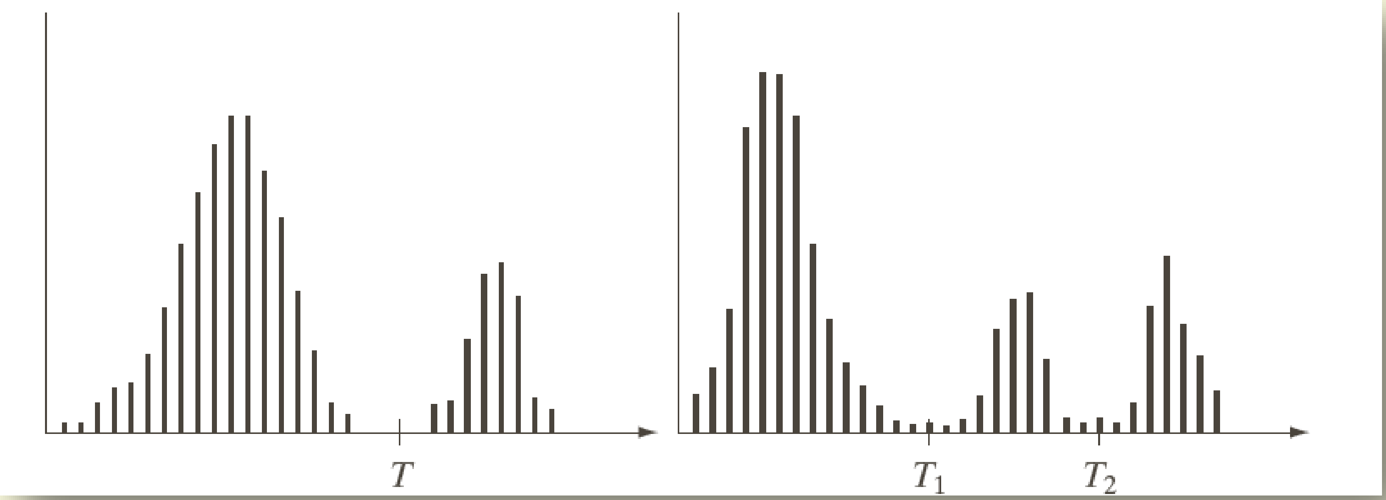


a b c

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

P-Tile Thresholding

- Limitation of P-Tile thresholding:

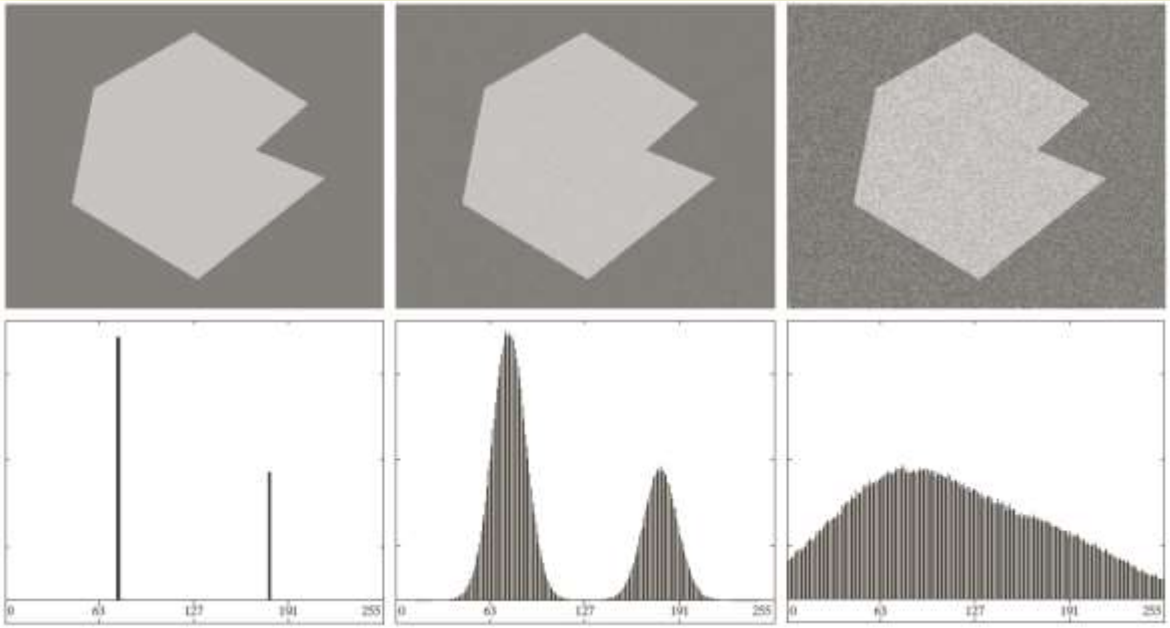


a b

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

P-Tile Thresholding

- Limitation of P-Tile thresholding:

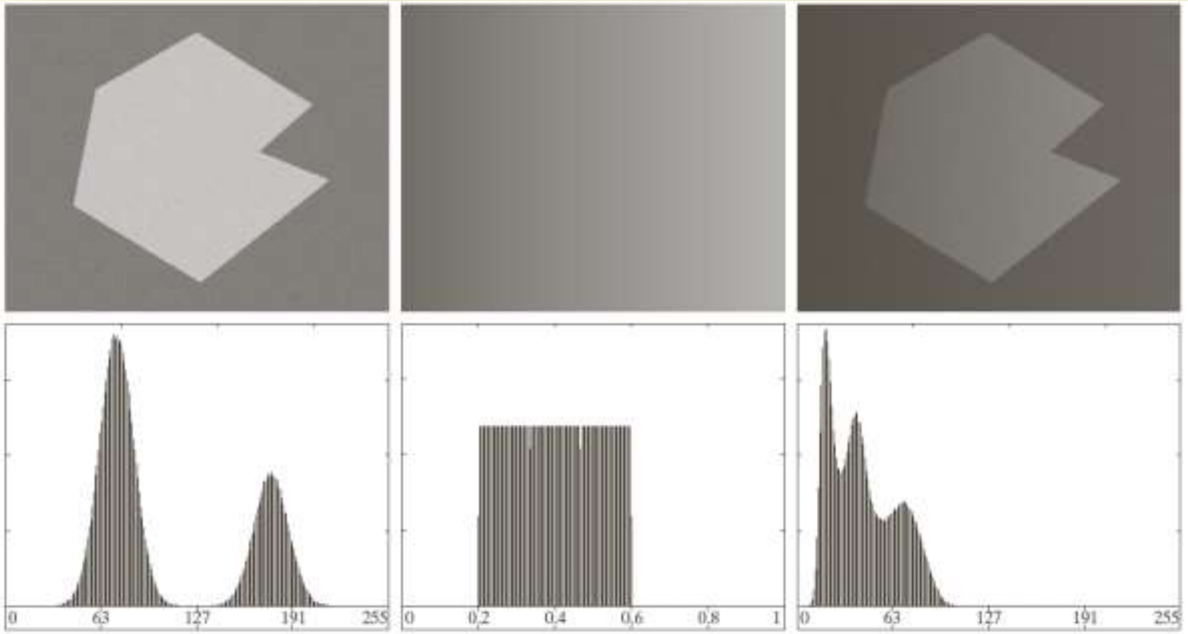


a b c
d e f

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

P-Tile Thresholding

- Limitation of P-Tile thresholding:



a	b	c
d	e	f

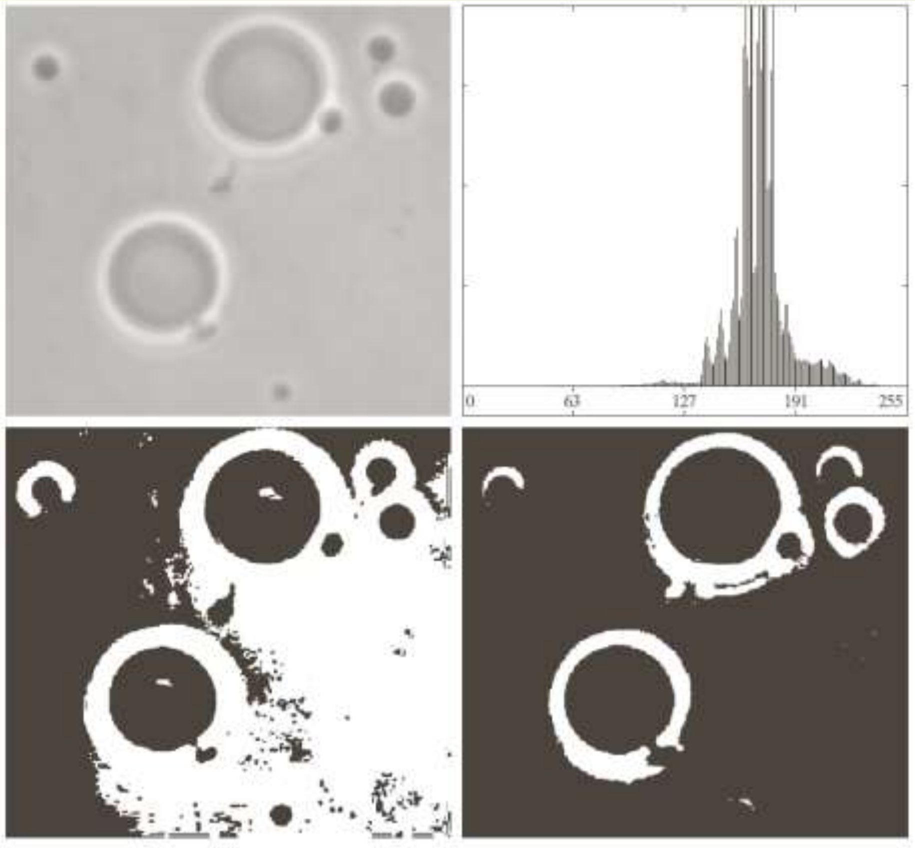
FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Adaptive Thresholding

- Adaptive Thresholding is used in scenes with uneven illumination where same threshold value not usable throughout complete image.
- In such case, look at small regions in the image and obtain thresholds for individual sub-images. Final segmentation is the union of the regions of sub-images.

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding

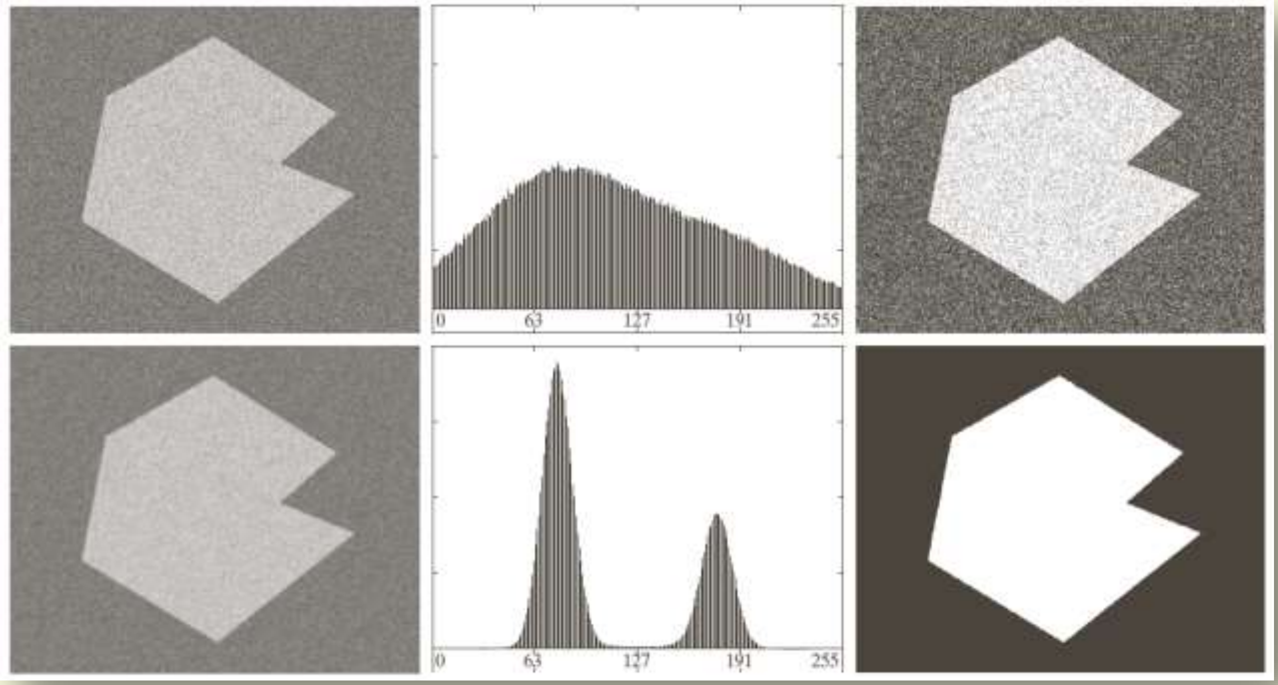


a b
c d

FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding

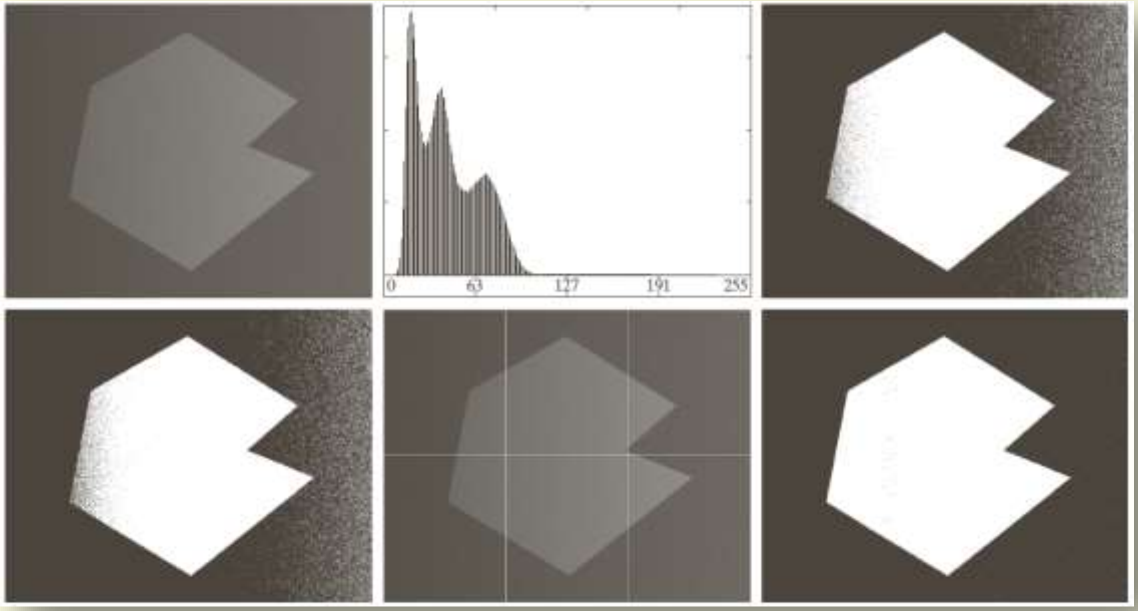


a b c
d e f

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

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding



a b c
d e f

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

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding

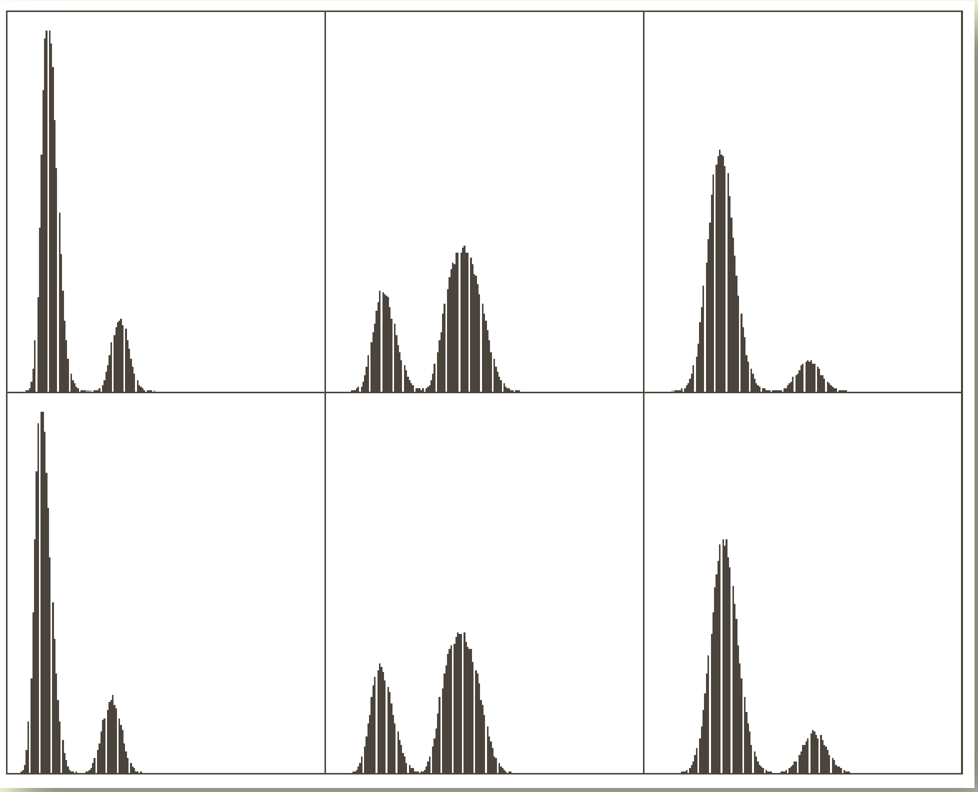
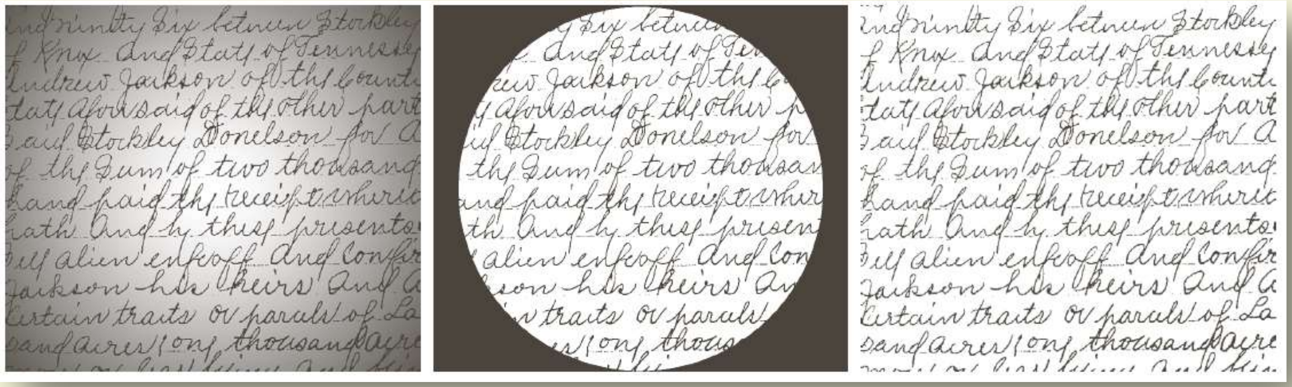


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

Adaptive Thresholding

- Thresholding – Basic Adaptive Thresholding



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

Summary

- Segmentation is the most essential step in most scene analysis and automatic pictorial pattern recognition problems.

- Choice of the technique depends on the peculiar characteristics of individual problem in hand.