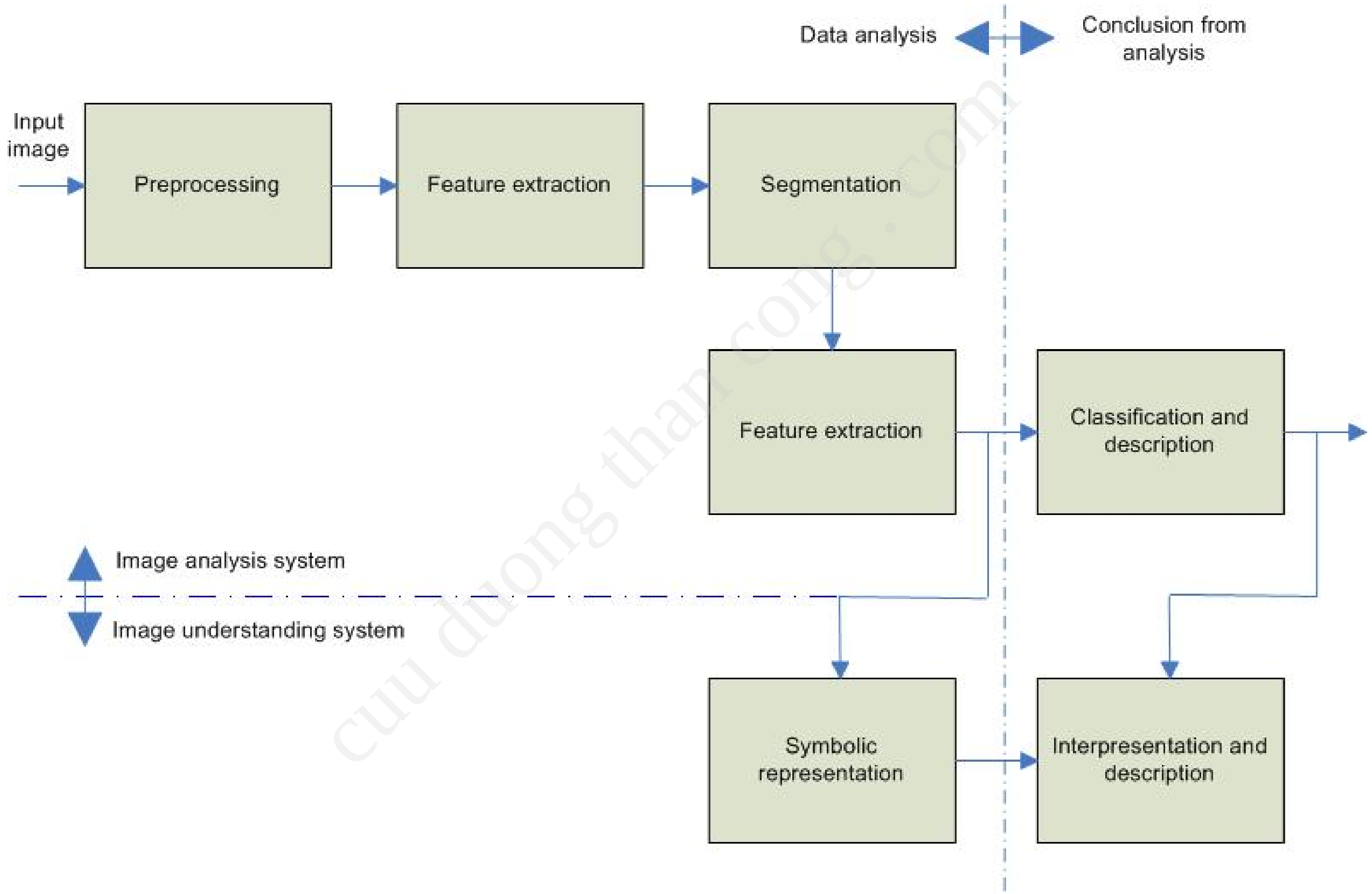
Chapter 5: Image Segmentation

5. Image Analysis



5. Image Segmentation (IS)

- Segmentation is to subdivide an image into its constituent regions or objects.

- Segmentation should stop when the objects of interest in an application have been isolated.

- Segmentation algorithms generally are based on one of 2 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.

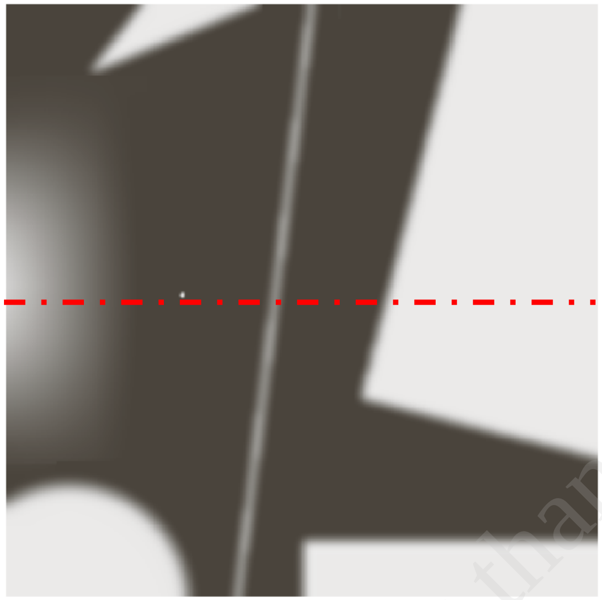
5. Image Segmentation: Detection of Discontinuities (1)

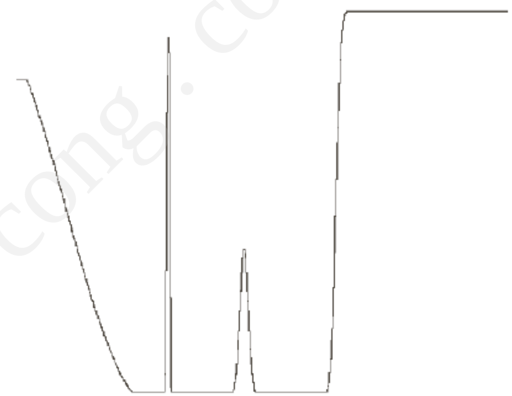
- Detect the three basic types of gray-level discontinuities
  - points, lines, edges.

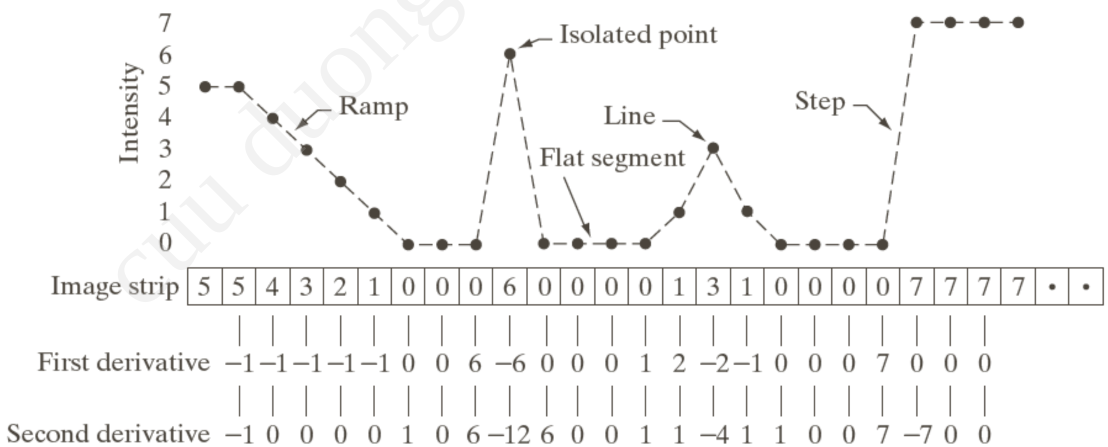
- The common way is to run a mask through the image

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

5. Image Segmentation: Detection of Discontinuities (2)







5. Image Segmentation: Detection of Discontinuities (3)

- **Characteristics of first and second order derivatives**:
  - First-order derivatives generally produce thicker edges in image.
  - Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise.
  - Second-order derivatives produce a double-edge response at ramp and step transition in intensity.
  - The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light.

5. Image Segmentation: Point Detection (1)

- A point has been detected at the location on which the mark is centered if

$$|R| \geq T$$

where

  - T is a nonnegative threshold.
  - R is the sum of products of the coefficients with the gray levels contained in the region encompassed by the mark.

–1	–1	–1
–1	8	–1
–1	–1	–1

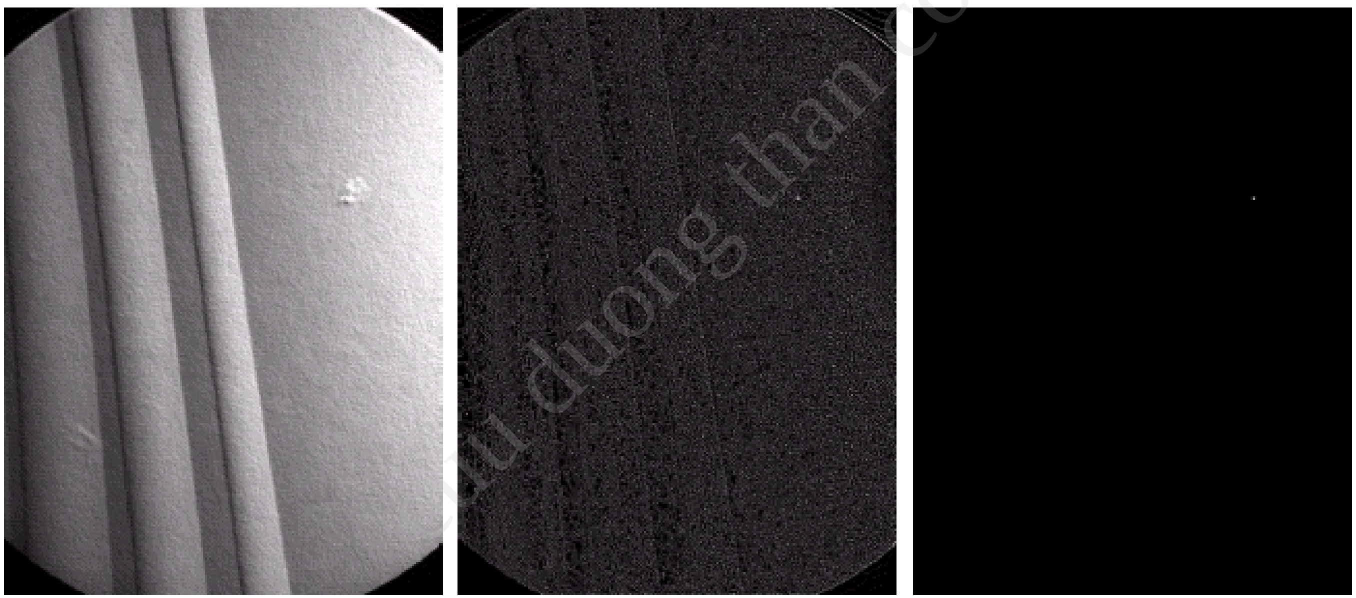
5. Image Segmentation: Point Detection (2)

- Note that the mark is the same as the mask of Laplacian operation.
- The only differences that are considered of interest are those large enough (as determined by *T*) to be considered isolated points.

$$|R| \geq T$$

5. Image Segmentation: Point Detection (3)

–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.)

5. Image Segmentation: Line Detection (1)

- Horizontal mask will result with max response when a line passed through the middle row of the mask with a constant background.
- 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.

−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°

5. Image Segmentation: Line Detection (2)

- Apply every masks on the image
- Let $R1$, $R2$, $R3$, $R4$ denotes the response of the **horizontal**, **+45 degree**, **vertical** and **-45 degree masks**, respectively. If, at a certain point in the image

$$|R_i| > |R_j|,$$

for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.
- 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.

5. Image Segmentation: Line Detection (3)

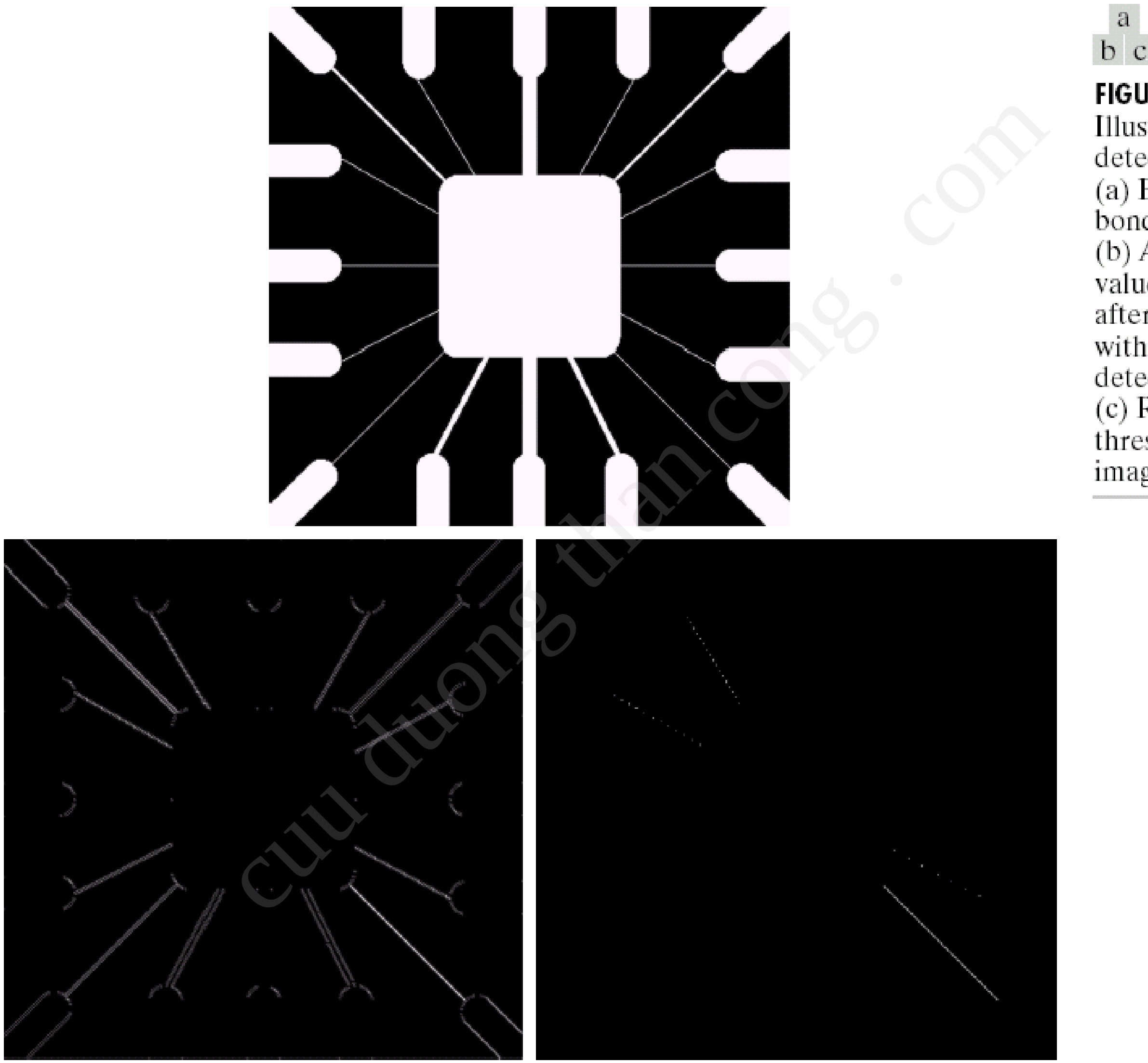


FIGURE 10.4 Illustration of line detection. (a) Binary wire-bond mask. (b) Absolute value of result after processing with $-45°$ line detector. (c) Result of thresholding image (b).

5. Image Segmentation: Edge Detection (1)

- The most common approach for detecting meaningful discontinuities in gray level.
- We discuss approaches for implementing
  - first-order derivative (Gradient operator)
  - second-order derivative (Laplacian operator)

  Here, we will talk only about their properties for 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.

5. Image Segmentation: Edge Detection (2)

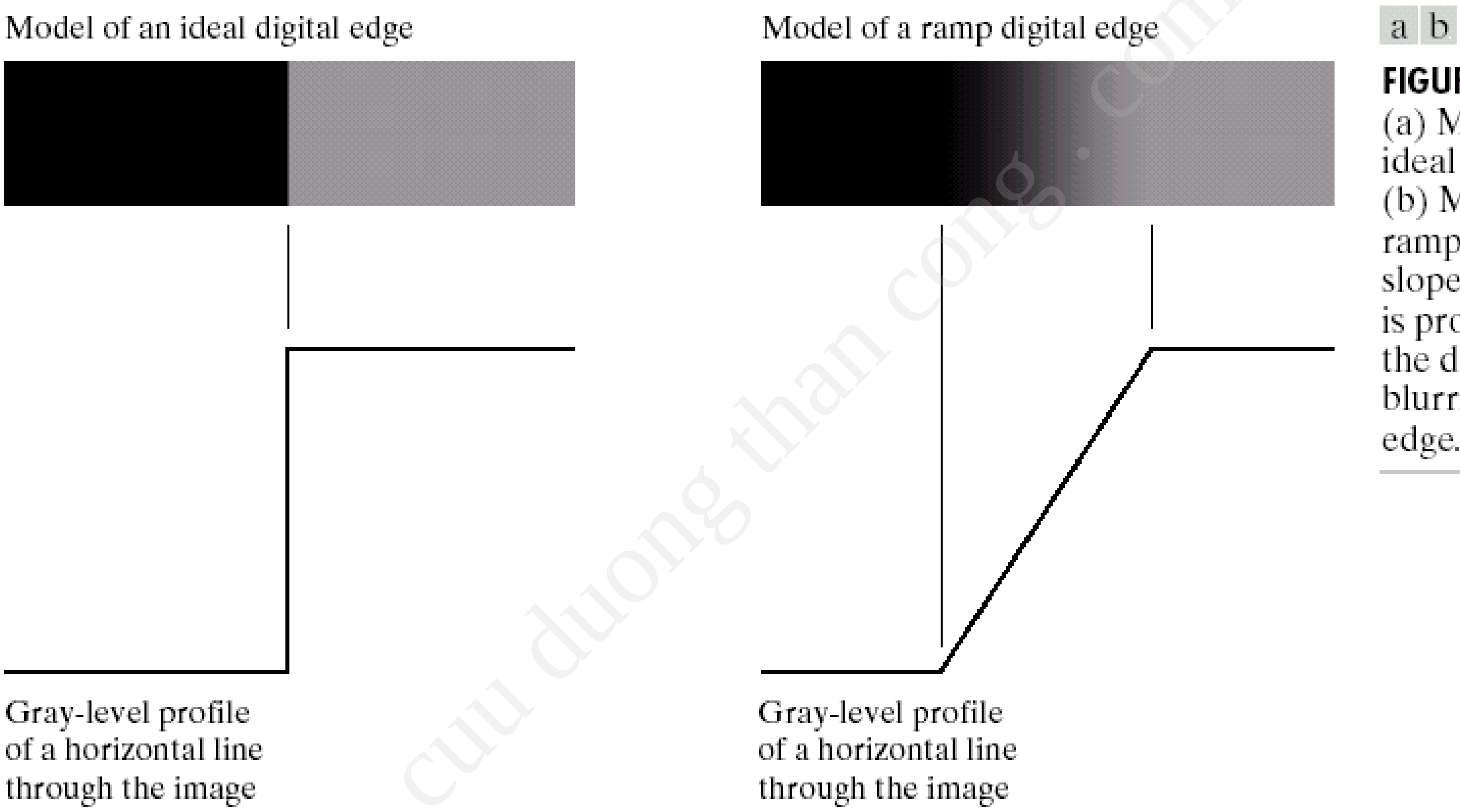
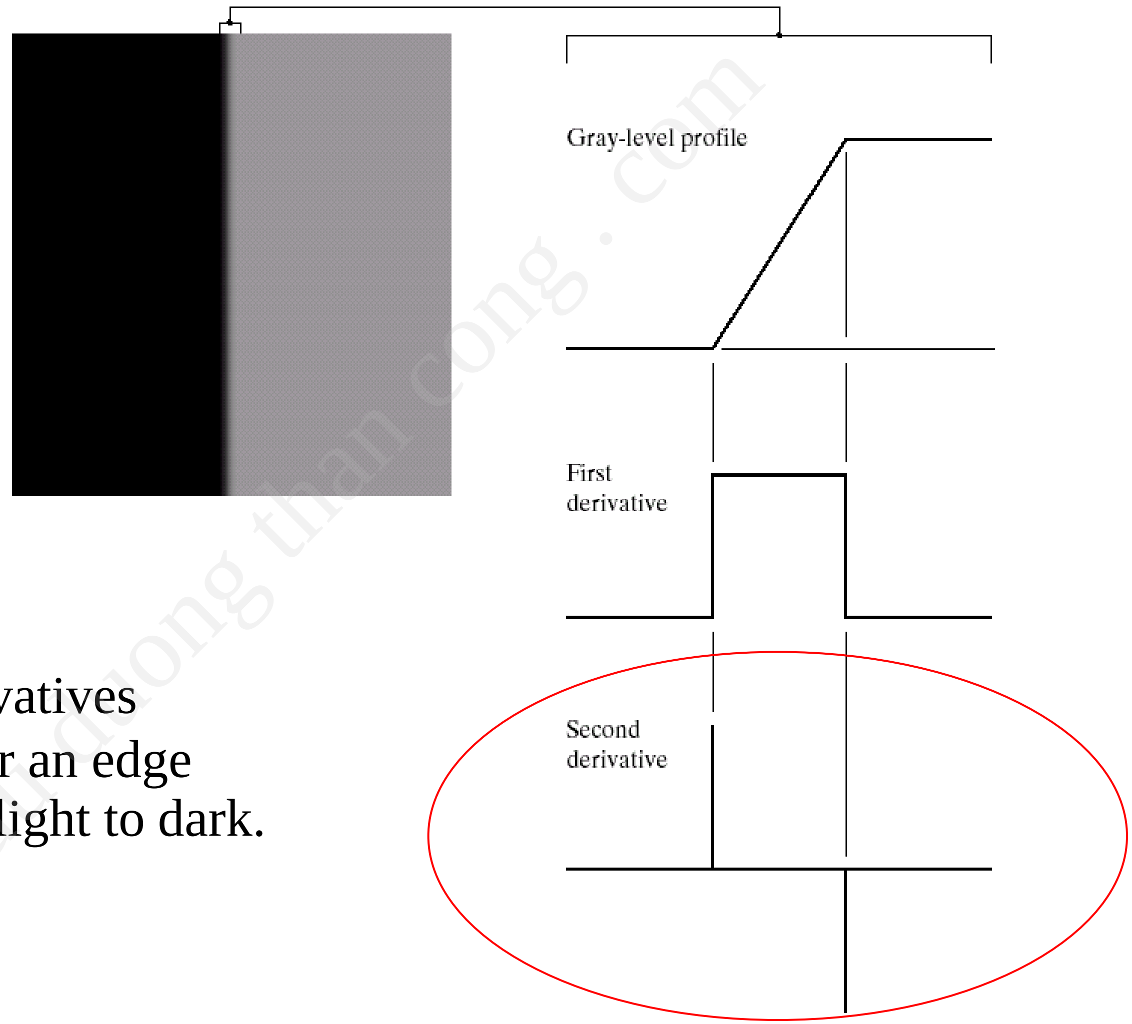


FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

5. Image Segmentation: Edge Detection (3)

- The slope of the ramp is inversely proportional to the degree of blurring in the edge. We no longer have a thin (one pixel thick) path.
- Instead, an edge point now is any point contained in the ramp, and an edge would then be a set of such points that are connected.
- The thickness is determined by the length of the ramp. The length is determined by the slope, which is in turn determined by the degree of blurring.
- **Blurred edges** tend to be **thick** and **sharp edges** tend to be **thin.**

5. Image Segmentation: Edge Detection (4)



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.

The signs of the derivatives would be reversed for an edge that transitions from light to dark.

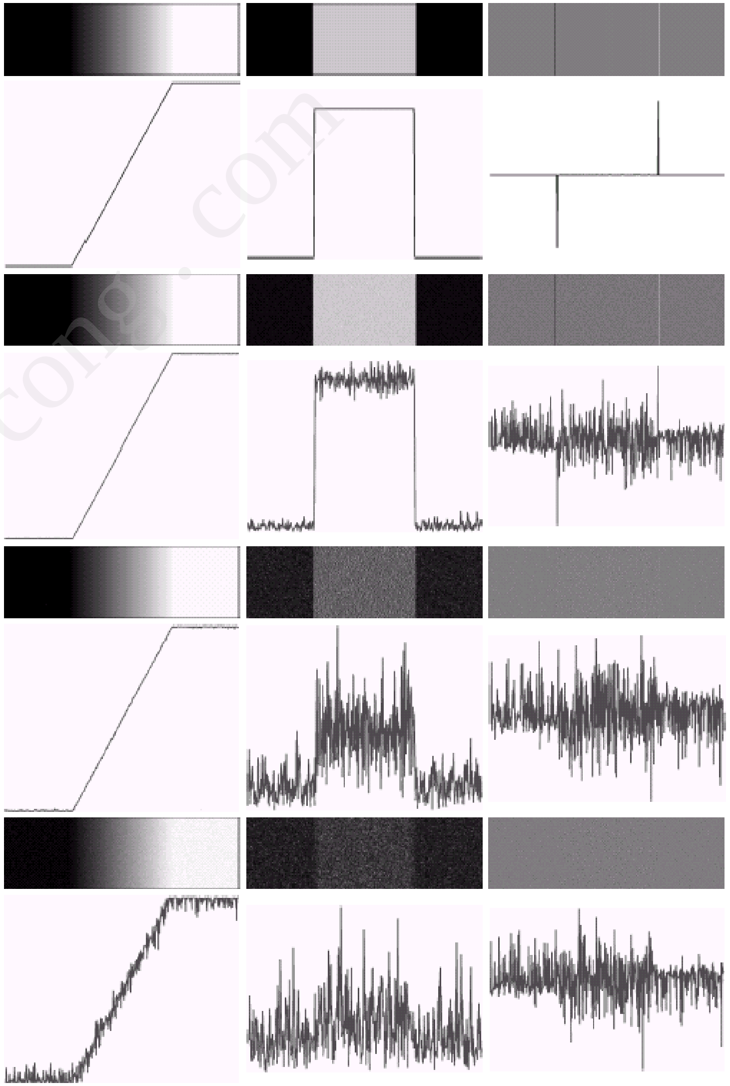
5. Image Segmentation: Edge Detection (5)

- **Second derivatives**:
  – Produces 2 values for every edge in an image (**an undesirable feature !**)
  – 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 property**).

- **Zero-crossing**:
  – Quite useful for locating the centers of thick edges.

5. Image Segmentation: Edge Detection (6)

- **Noise Images**:
  - 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.



5. Image Segmentation: Edge Detection (7)

– Fairly little noise can have such a significant impact on the two key derivatives used for edge detection in images. **Image smoothing** should be serious consideration prior to the use of derivatives in applications where noise is likely to be present.

- **Edge point**: to determine a point as an edge point
  – the transition in grey level associated with the point has to be significantly stronger than the background at that point.
  – use threshold to determine whether a value is "significant" or not.
  – the point's two-dimensional first-order derivative must be greater than a specified threshold.

5. Image Segmentation: Edge Detection (8)

- **Gradient Operator:**

  First derivatives are implemented using the **magnitude of the gradient.**

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

$$\nabla f = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

commonly approx.

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

the magnitude becomes nonlinear

5. Image Segmentation: Edge Detection (9)

- **Gradient direction:**
  - Let $\alpha(x, y)$ represent the direction angle of the vector ∇f at (x, y)

$$\alpha(x, y) = \tan^{-1}(G_y / G_x)$$

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

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

5. Image Segmentation: Edge Detection (10)

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, $|G_x| + |G_y|$.

5. Image Segmentation: Edge Detection (11)

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.

5. Image Segmentation: Edge Detection (12)

❑ **Laplacian operator:**

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

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

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

- **Laplacian of Gaussian (LoG):**

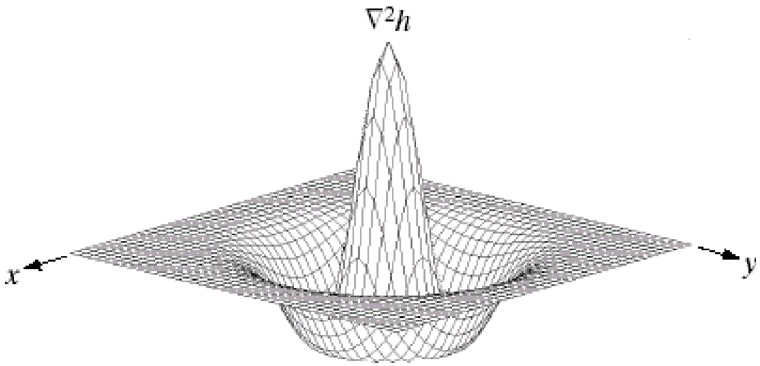
  Laplacian combined with smoothing as a precursor to find edges via zero-crossing.

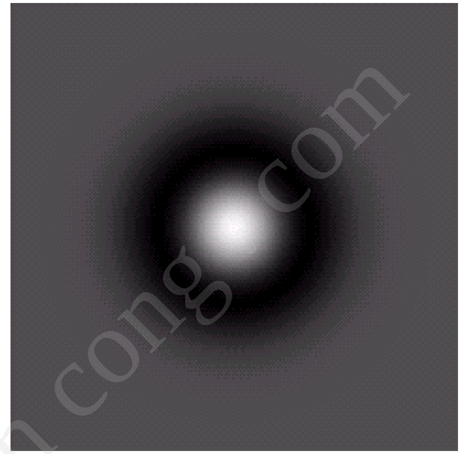
$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

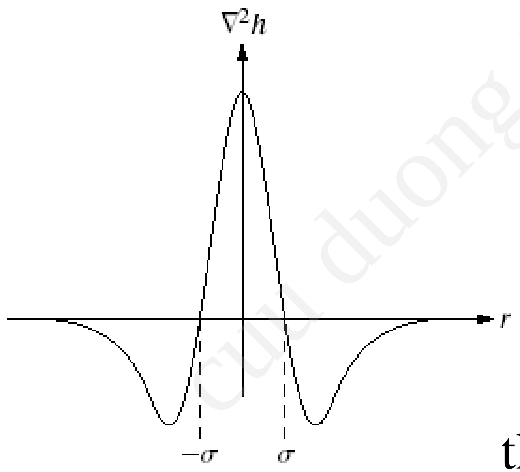
where $r^2 = x^2+y^2$, and σ is the standard deviation

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

5. Image Segmentation: Edge Detection (14)







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

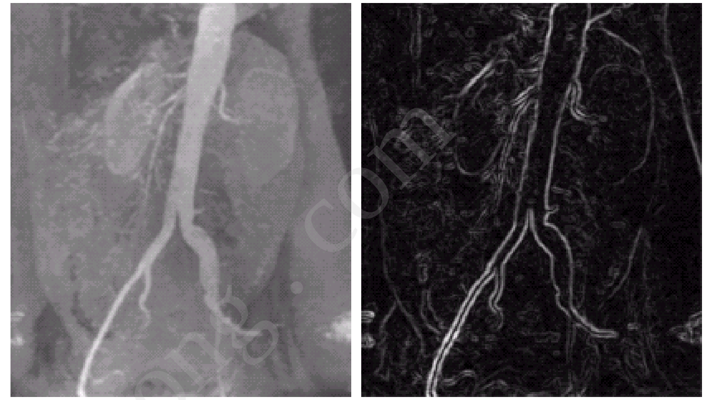
a b
c d

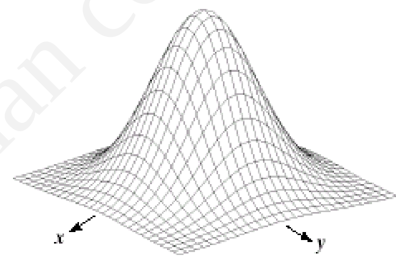
FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5 × 5 mask approximation to the shape of (a).

the coefficient must be sum to zero

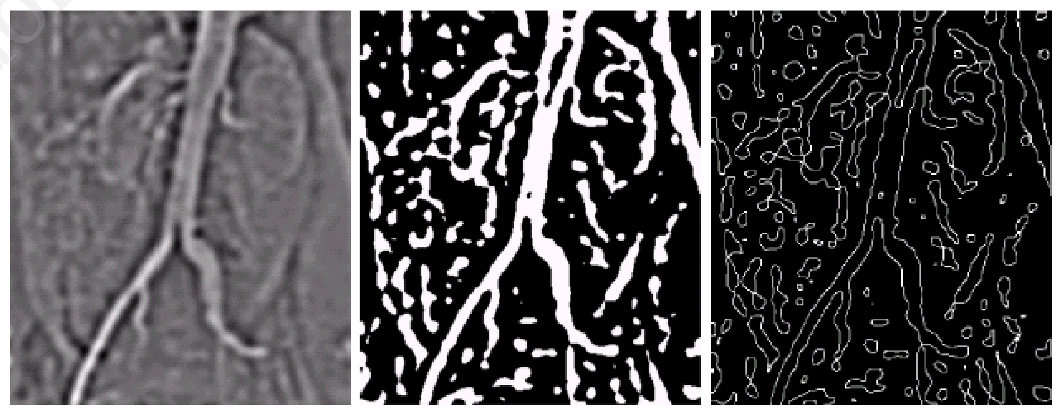
5. Image Segmentation: Edge Detection (15)

a). Original image
b). Sobel Gradient
c). Spatial Gaussian
    smoothing function
d). Laplacian mask
e). LoG
f). Threshold LoG
g). Zero crossing





−1	−1	−1
−1	8	−1
−1	−1	−1



5. Image Segmentation: Edge Detection (16)

- **Zero crossing & LoG:**
  - Approximate the zero crossing from LoG image.
  - to threshold the LoG image by setting all its positive values to white and all negative values to black.
  - the zero crossing occur between positive and negative values of the thresholded LoG.

- **Zero crossing vs. gradient:**
  - Zero crossing produces thinner edges.
  - Noise reduction.
  - Zero crossing creates closed loops (spaghetti effect).
  - Sophisticated computation.
  - Gradient is more frequently used.

5. IS: Edge Linking & Boundary Detection (1)

- Edge detection algorithm are followed by linking procedures to assemble edge pixels into meaningful edges.
- Basic approaches:
  - Local Processing
  - Global Processing via the Hough Transform
  - Global Processing via Graph-Theoretic Techniques

5. IS: Edge Linking & Boundary Detection (2)

- **Local processing**:
  - Analyze the characteristics of pixels in a small neighborhood (e.g, 3×3, 5×5) about every edge pixels (x, y) in an image.
  - All points that are similar according to a set of predefined criteria are linked, forming an edge of pixels that share those criteria.
  - Criteria:

    1. The strength of the response of the gradient operator used to produce the edge pixel: an edge pixel with coordinates (x_0, y_0) in a predefined neighborhood of (x, y) is similar in magnitude to the pixel at (x, y) if

$$|\nabla f(x, y) - \nabla f(x_0, y_0)| \le E$$

5. IS: Edge Linking & Boundary Detection (3)

2. The direction of the gradient vector: an edge pixel with coordinates (x_0, y_0) in a predefined neighborhood of (x, y) is similar in angle to the pixel at (x, y) if

$$|\alpha(x, y) - \alpha(x_0, y_0)| < A$$

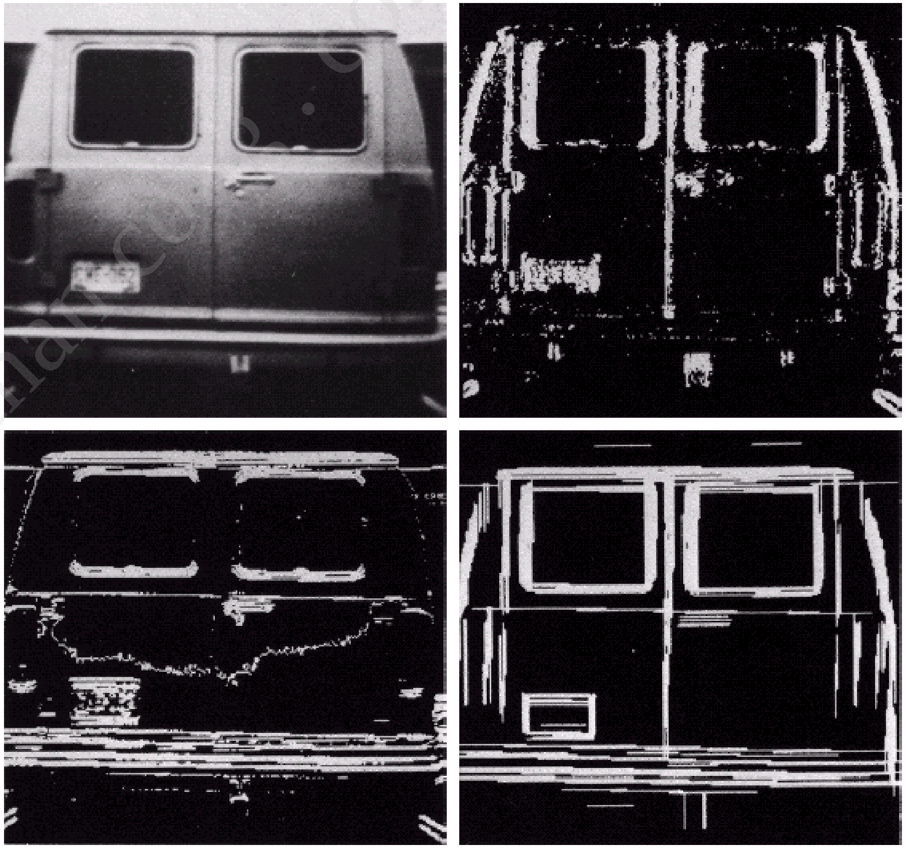
- ✓ A point in the predefined neighborhood of (x, y) is linked to the pixel at (x, y) if **both magnitude and direction criteria are satisfied.**
- ✓ The process is repeated at every location in the image.
- ✓ A record must be kept, simply by assigning a different gray level to each set of linked edge pixels.

5. IS: Edge Linking & Boundary Detection (4)

Example: Find rectangles whose sizes makes them suitable candidates for license plates.

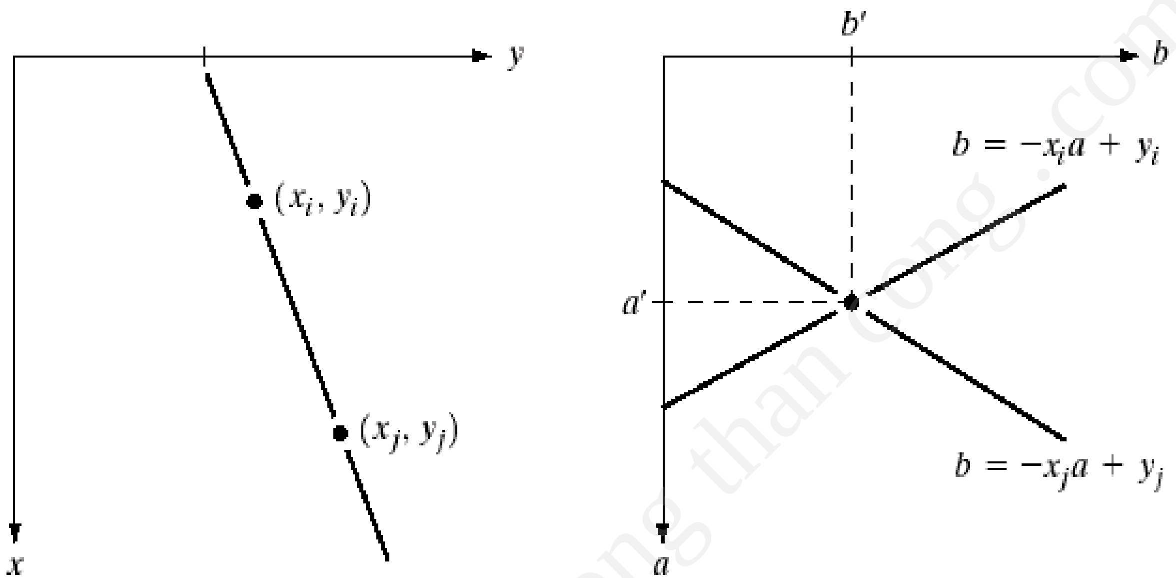
a b
c d

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.)



- Use horizontal & vertical Sobel operators.
- Eliminate isolated short segments.
- **Link conditions:**
  - gradient value > 25
  - gradient direction differs < 15°

❑ Global Processing using Hough Transformation:



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

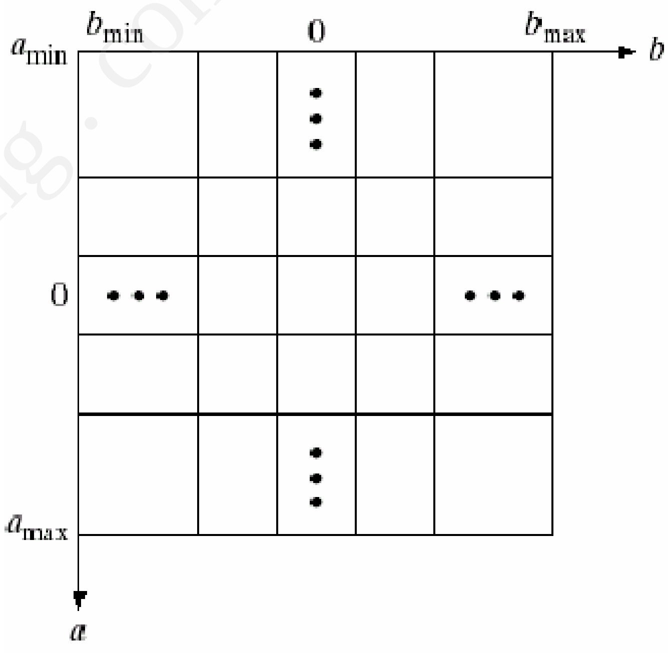
xy-plane — *ab*-plane or parameter space

$y_i = ax_i + b$ ⇒ $b = -ax_i + y_i$

All points (x_i, y_i) contained on the same line must have lines in parameter space that intersect at (a', b').

Accumulator cells:

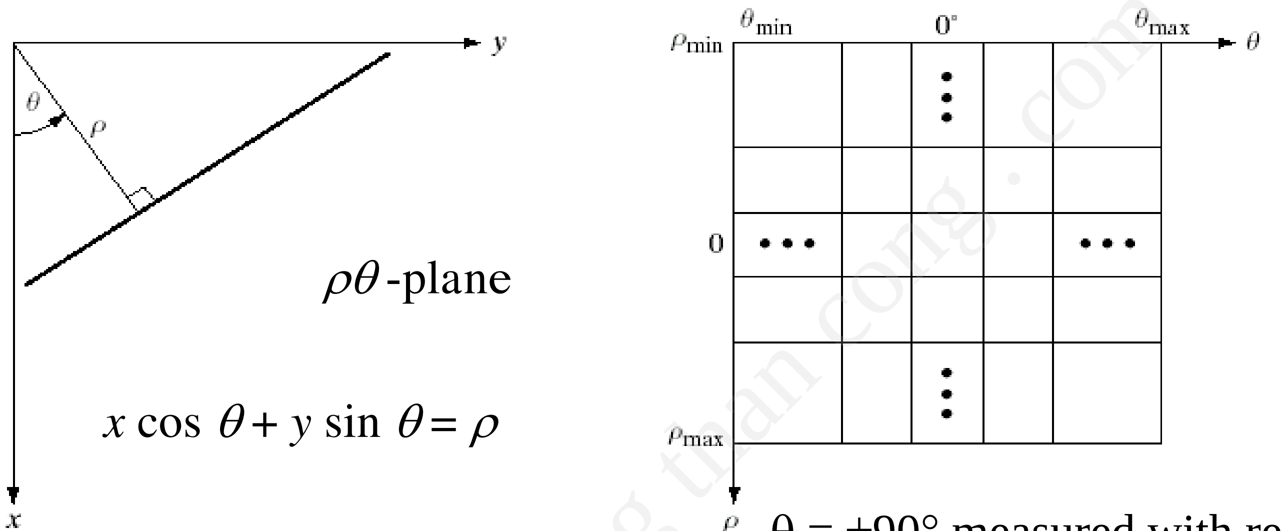
- (a_{max}, a_{min}) and (b_{max}, b_{min}) are the expected ranges of slope and intercept values. All are initialized to zero.
- if a choice of a_p results in solution b_q then we let $A(p,q) = A(p, q) + 1$
- at the end of the procedure, value Q in $A(i, j)$ corresponds to Q points in the xy-plane lying on the line $y = a_i x + b_j$



$b = -ax_i + y_i$

5. IS: Edge Linking & Boundary Detection (7)

- **Hough Transformation (Line): $\rho\theta$ - plane**



a b
FIGURE 10.19
(a) Normal representation of a line.
(b) Subdivision of the $\rho\theta$-plane into cells.

θ = ±90° measured with respect to *x*-axis

- Problem of using equation $y = ax + b$ is that value of a is infinite for a vertical line.
- To avoid the problem, use equation: $x \cos\theta + y \sin\theta = \rho$ to represent a line, instead.
- Vertical line has $\theta = 90°$ with ρ equals to the positive *y*-intercept or $\theta = -90°$ with ρ equals to the negative *y*-intercept.

5. IS: Edge Linking & Boundary Detection (8)

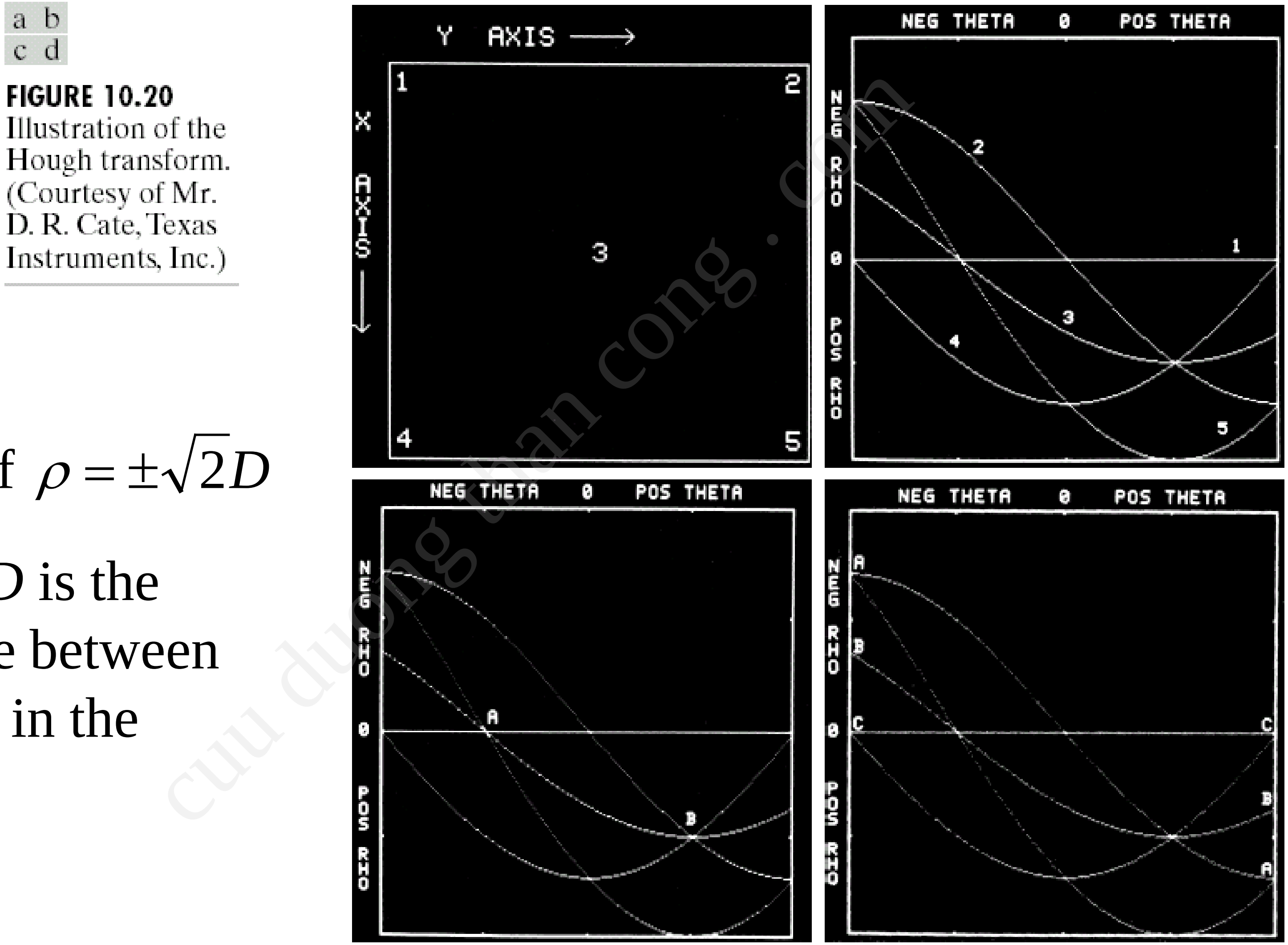


FIGURE 10.20 Illustration of the Hough transform. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

range of $\rho = \pm\sqrt{2}D$

where D is the distance between corners in the image.

5. IS: Edge Linking & Boundary Detection (9)

- **Generalized Hough Transformation: C**an be used for any function of the form:

$$g(v, c) = 0$$

where: v is a vector of coordinates and c is a vector of coefficients.

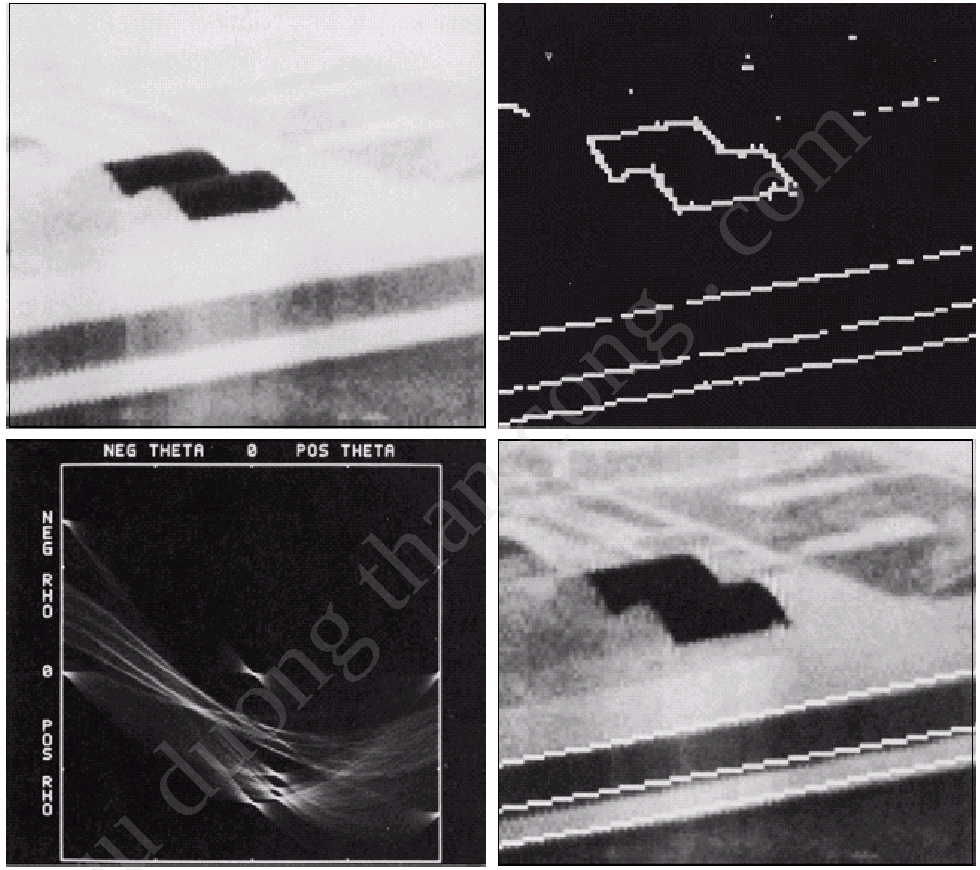
- **Hough Transformation (Circle):**
  - Equation: $(x-c_1)^2 + (y-c_2)^2 = c_3^2$
  - Three parameters (c_1, c_2, c_3)
  - Cube like cells.
  - Accumulators of the form $A(i, j, k)$.
  - Increment c_1 and c_2 , solve of c_3 that satisfies the equation.
  - Update the accumulator corresponding to the cell associated with triplet (c_1, c_2, c_3).

5. IS: Edge Linking & Boundary Detection (10)

- **Edge-linking based on Hough Transformation:**
  1. Compute the gradient of an image and threshold it to obtain a binary image.
  2. Specify subdivisions in the $\rho\theta$ - plane.
  3. Examine the counts of the accumulator cells for high pixel concentrations.
  4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

     Continuity:
     - Based on computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell.
     - A gap at any point is significant if the distance between that point and its closet neighbor exceeds a certain threshold.

5. IS: Edge Linking & Boundary Detection (11)



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.)

<u>Link criteria</u>:

1. The pixels belonged to one of the set of pixels linked according to the highest count.
2. No gaps were longer than 5 pixels.

5. IS: Edge Linking & Boundary Detection (12)

- **Edge linking with Global Processing via Graph-Theoretic Techniques:**
  - Searching graph for low-cost paths that correspond to significant edges.
  - Performs well in the presence of noise.
  - Time consuming process.

- **Edge linking with Global Processing via Graph-Theoretic Techniques: Algorithm**
  - Suppose that a graph $G = (N, A)$ is a finite, none empty set of nodes, N, together with an set A of unordered pairs of distinct/different elements of N (node). Each pair (n_i, n_j) of A is called an arc.
  - A graph in which the arcs are directed, is called a directed graph. If an arc is directed from node n_i to node n_j, the n_j is the successor of the parent node n_i. The process in which one identifies the successor of a node is called expansion of the node.

5. IS: Edge Linking & Boundary Detection (14)

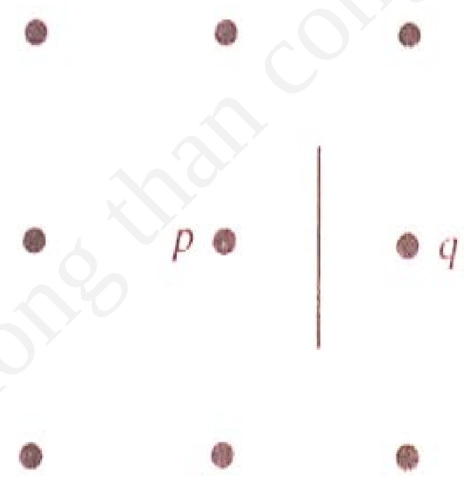
– In each graph levels are defined. The level 0 consists of a start node, while the node on the last level is called the goal node.

– A cost $c(n_i,n_j)$ can be associated with every arc (n_i, n_j). A path is a sequence of nodes $n_1, n_2,\ldots, n_k$, where each node n_i being a successor of node n_i -1. The cost for this graph:

$$c = \sum_{i=2}^{k} c(n_i - 1, n_i)$$

5. IS: Edge Linking & Boundary Detection (15)

– An edge element is the boundary between 2 pixels *p* and *q*, such that *p* and *q* are 4-neighbors. Figure below contains a 3×3 image.



5. IS: Edge Linking & Boundary Detection (16)

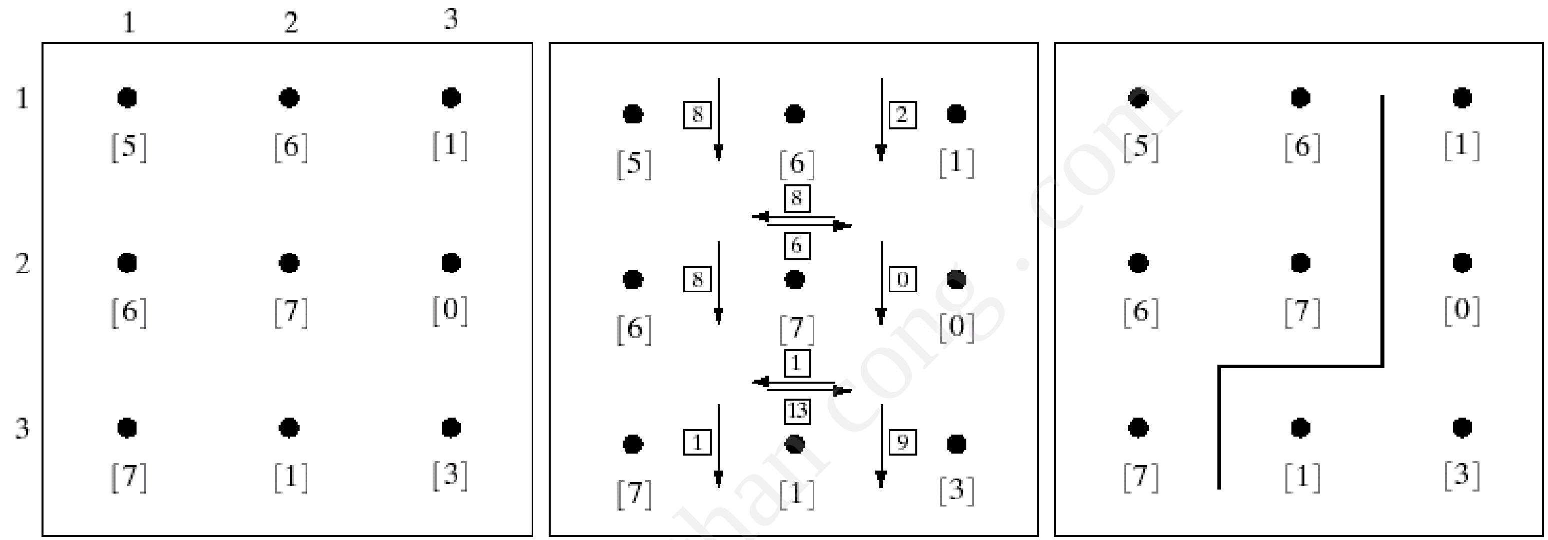
– Each edge element defined by the pixel *p* and *q* has an associated cost:

$$c = H - (f(p) - f(q)),$$

where *H* is the highest pixel value in the image, *f*(*p*) and *f*(*q*) are pixel values for respectively *p* and *q*.

– Edge linking: Minimum cost edge/path (see next slide):

5. IS: Edge Linking & Boundary Detection (17)



a b c

FIGURE 10.23 (a) A 3 × 3 image region. (b) Edge segments and their costs. (c) Edge corresponding to the lowest-cost path in the graph shown in Fig. 10.24.

The number in the parenthesis indicates the pixel values of the pixels indicated as black points. Line number and column number are respectively given along the left and the upper edge.

5. IS: Edge Linking & Boundary Detection (18)

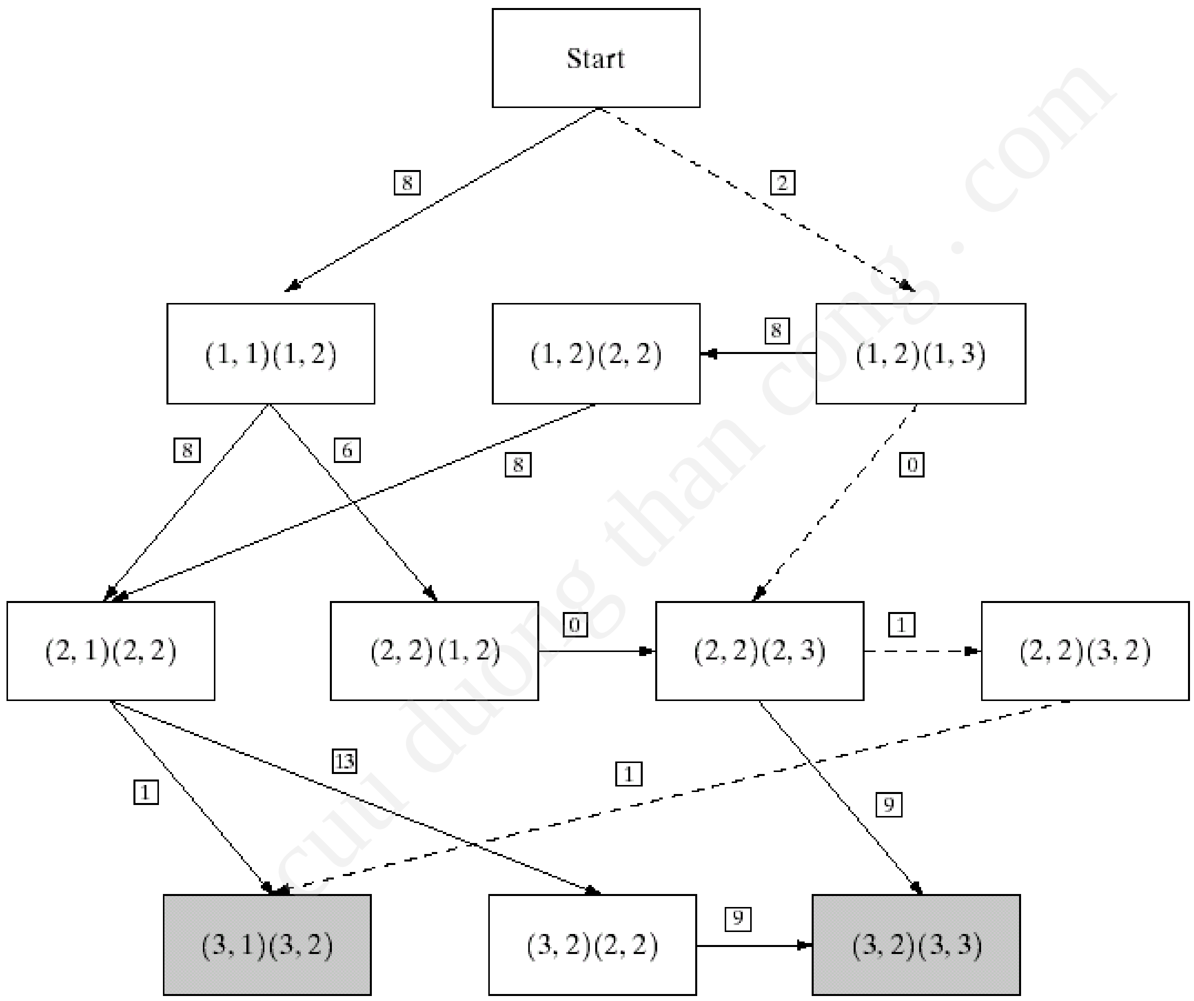


FIGURE 10.24 Graph for the image in Fig. 10.23(a). The lowest-cost path is shown dashed.

5. IS: Thresholding (1)

- **Foundation:**

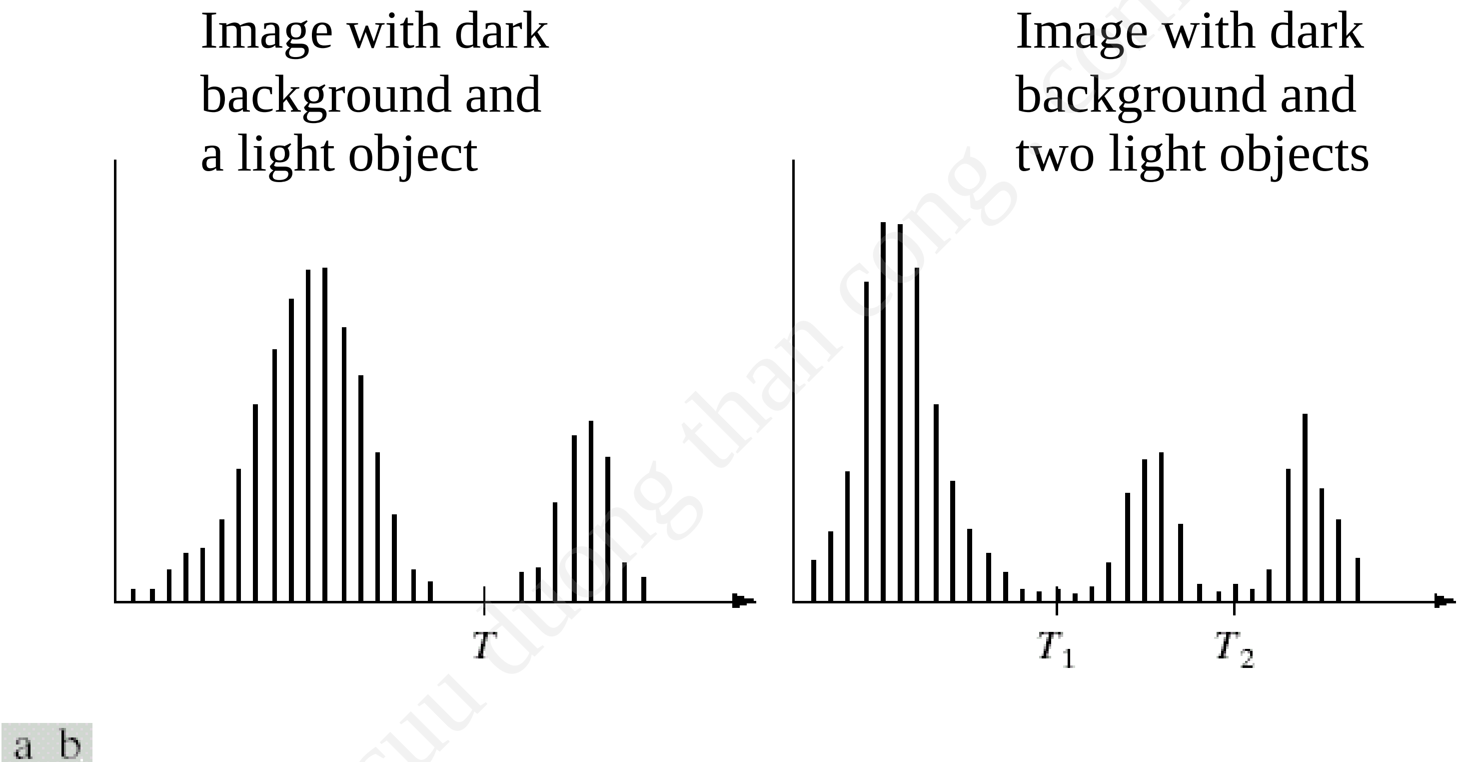


FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

5. IS: Thresholding (2)

- Consider an image *f*(*x*,*y*), the segmented image is given by:

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

T : global thresholding

Multiple thresholding

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

5. IS: Thresholding (3)

- **Role of noise in image thresholding:**

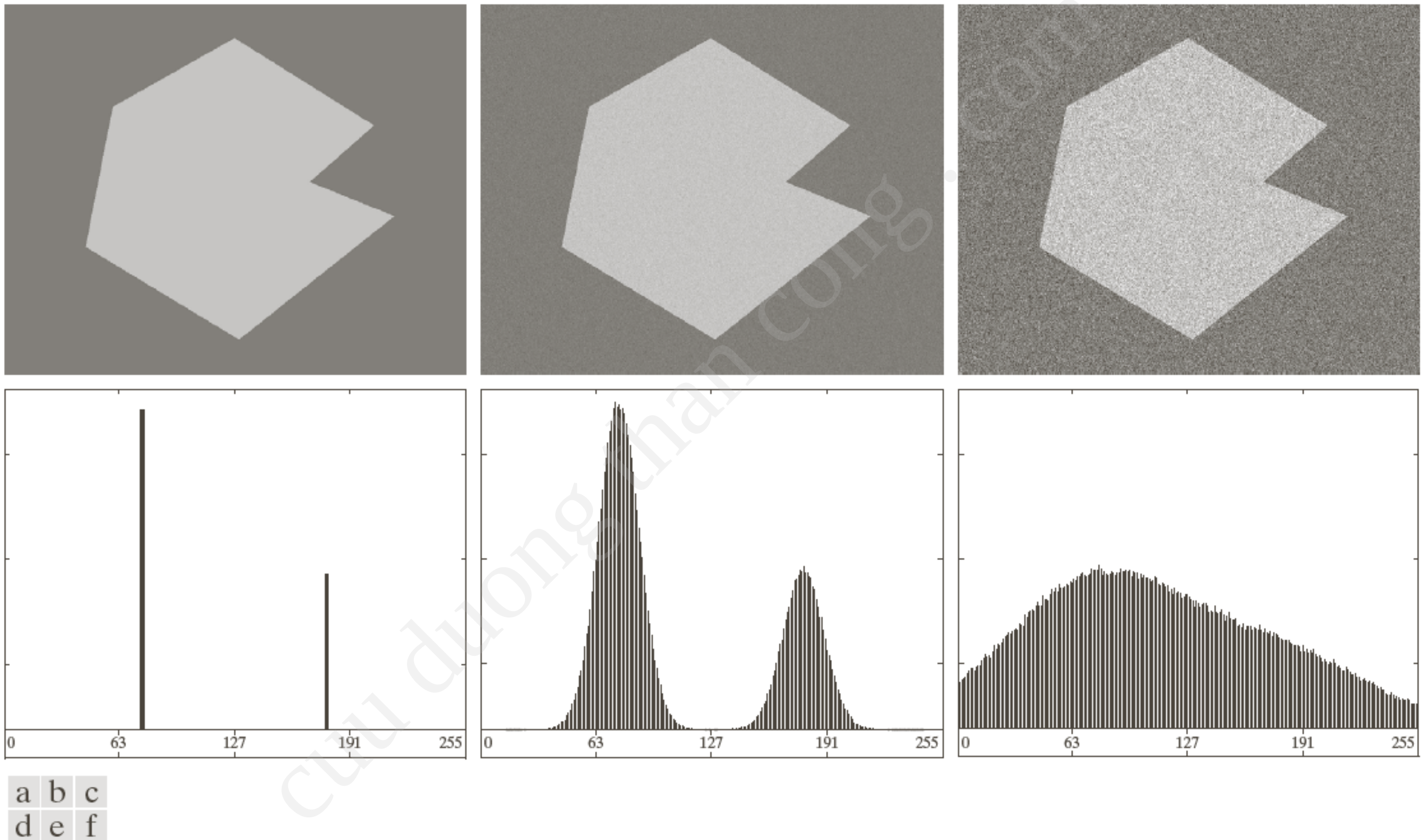


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

- **Role of illumination and reflectance:**

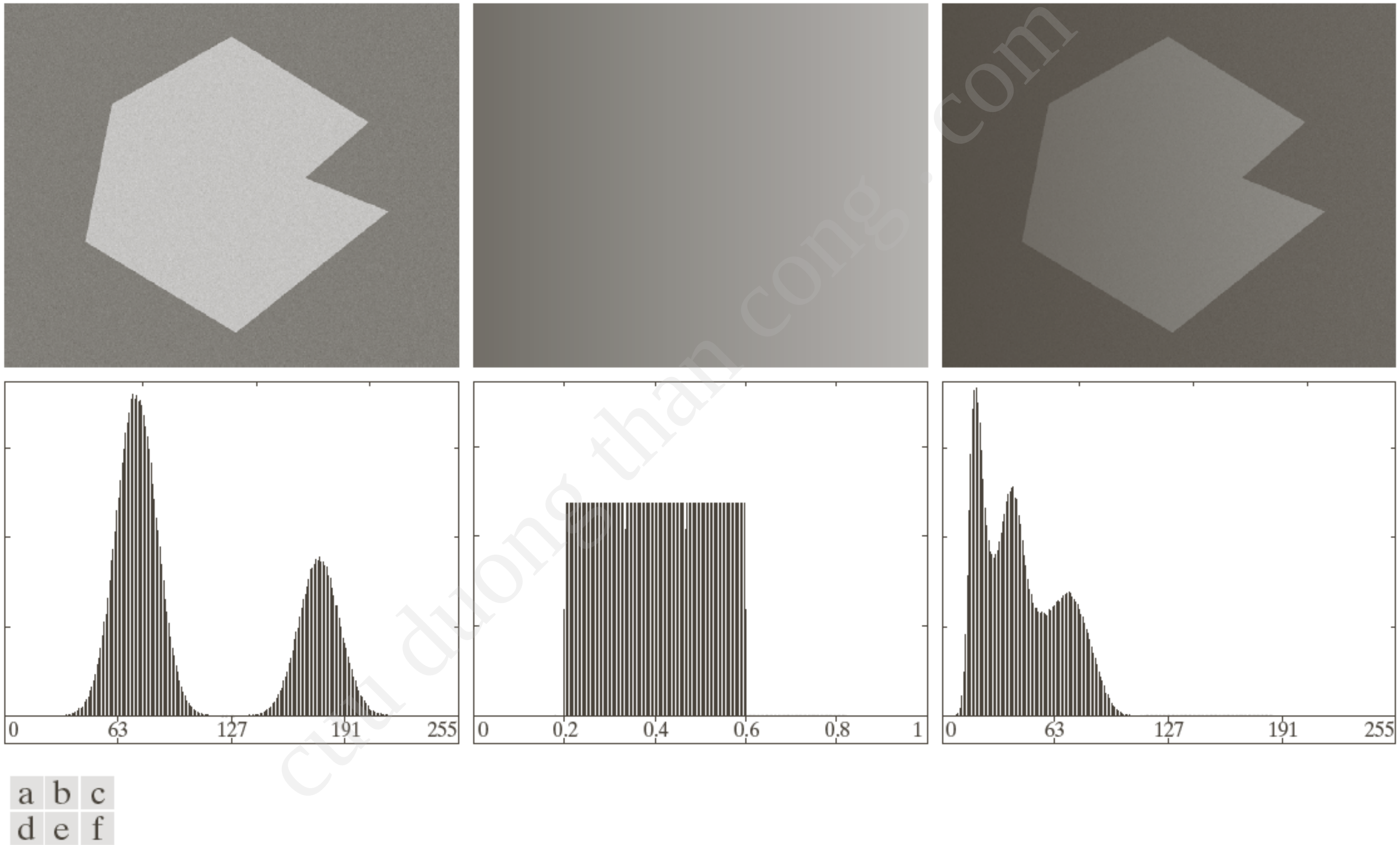
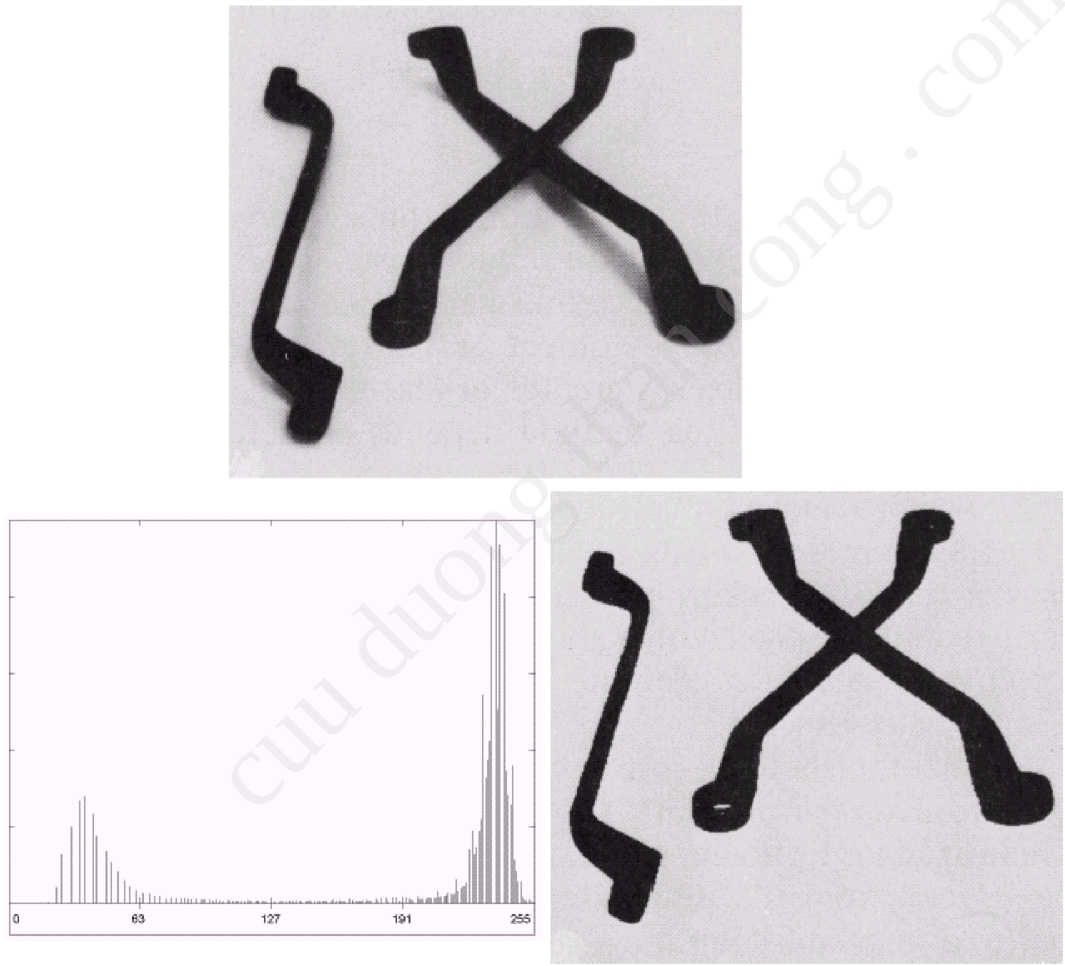


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

5. IS: Thresholding (5)

- **Multilevel thresholding:**
  - A point (x, y) belongs to
    - an **object class** if $T_1 < f(x, y) \le T_2$
    - **another object class** if $f(x, y) > T_2$
    - **background** if $f(x, y) \le T_1$
  - T depends on:
    - only $f(x, y)$: only on gray-level values ⇨ **Global threshold.**
    - both $f(x, y)$ and $p(x, y)$: on gray-level values and its neighbors ⇨ **Local threshold.**

❑ 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.

5. IS: Thresholding (7)

- **Basic global thresholding: Summary**

  Based on visual inspection of histogram:

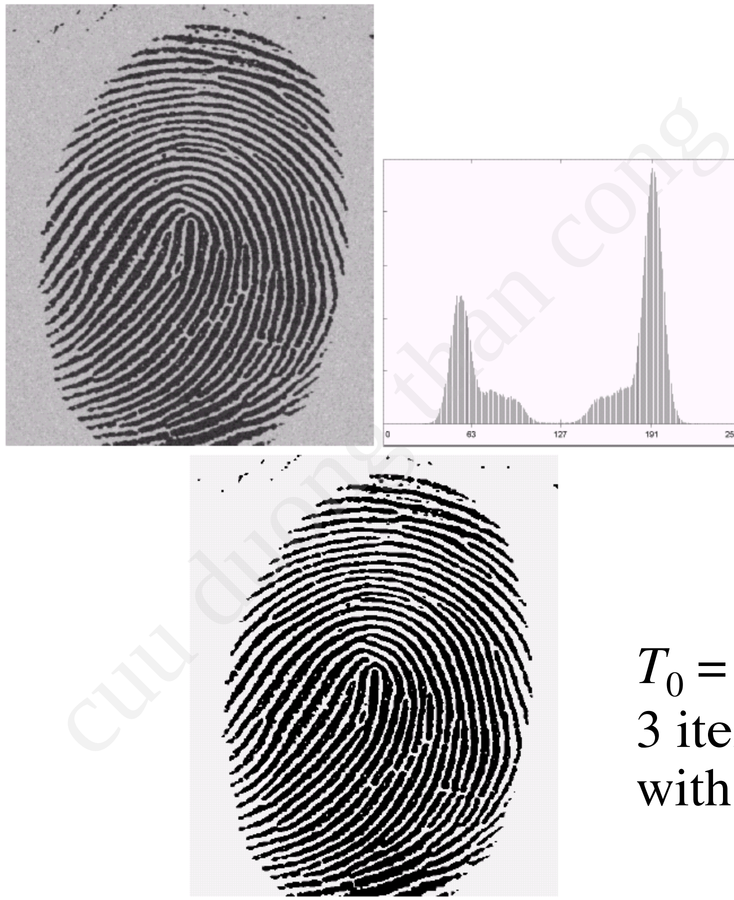
  1. Select an initial estimate for T.
  2. Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values > T and G_2 consisting of pixels with gray level values ≤ T.
  3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions G_1 and G_2.
  4. Compute a new threshold value:

  $$T = 0.5\,(\mu_1 + \mu_2)$$

  5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.

5. IS: Thresholding (8)

Example: Clear valley of the histogram and the effective of the segmentation between object and background.



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.)

$T_0 = 0$
3 iterations
with result $T = 125$

5. IS: Thresholding (9)

- **<u>Optimum global thresholding using Otsu's method</u>**:

Principle: Maximizing the **between-class variance**.

Let $\{0, 1, 2, \ldots, L\text{-}1\}$ denote the L distinct intensity levels in a digital image of size $M \times N$ pixels, and let n_i denote the number of pixels with intensity *i*.

$$p_i = n_i / MN \quad \text{and} \quad \sum_{i=0}^{L-1} p_i = 1$$

k is a threshold value, $C_1 \to [0, k]$, $C_2 \to [k+1, L\text{-}1]$

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

5. IS: Thresholding (10)

The mean intensity value of the pixels assigned to class C_1 is

$$m_1(k) = \sum_{i=0}^{k} iP(i/C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$$

The mean intensity value of the pixels assigned to class C_2 is

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i/C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

$$P_1 m_1 + P_2 m_2 = m_G \quad \text{(Global mean value)}$$

5. IS: Thresholding (11)

Between - class variance, σ_B^2 is defined as

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

$$= P_1P_2(m_1 - m_2)^2$$

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

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

5. IS: Thresholding (12)

The optimum threshold is the value, k*, that maximizes

$$\sigma_B^2(k^*), \qquad \sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$

Then, the **segmented image** is:

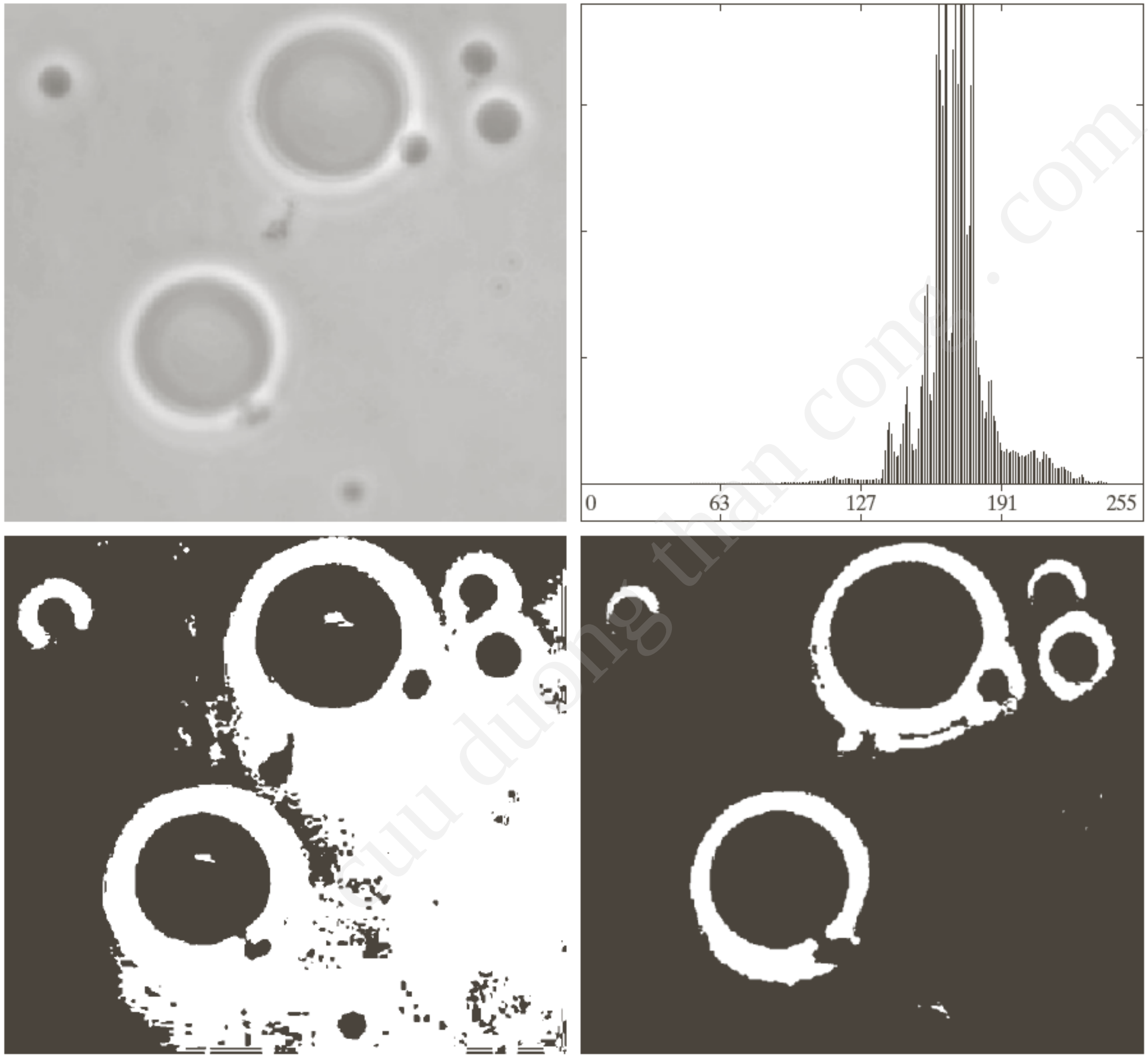
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > k^* \\ 0 & \text{if } f(x,y) \le k^* \end{cases}$$

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

Otsu's Algorithm: Summary

1. Compute the normalized histogram of the input image. Denote the components of the histogram by p_i, $i = 0, 1, \ldots, L\text{-}1$.
2. Compute the cumulative sums, $P_1(k)$, for $k = 0, 1, \ldots, L$.
3. Compute the cumulative means, $m(k)$, for $k = 0, 1, \ldots, L\text{-}1$.
4. Compute the global intensity mean, m_G.
5. Compute the between-class variance, for $k = 0, 1, \ldots, L\text{-}1$.
6. Obtain the Otsu's threshold, k^*.
7. Obtain the separability measure.

5. IS: Thresholding (14)



a b
c d

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

❑ Using image smoothing to improve global thresholding:

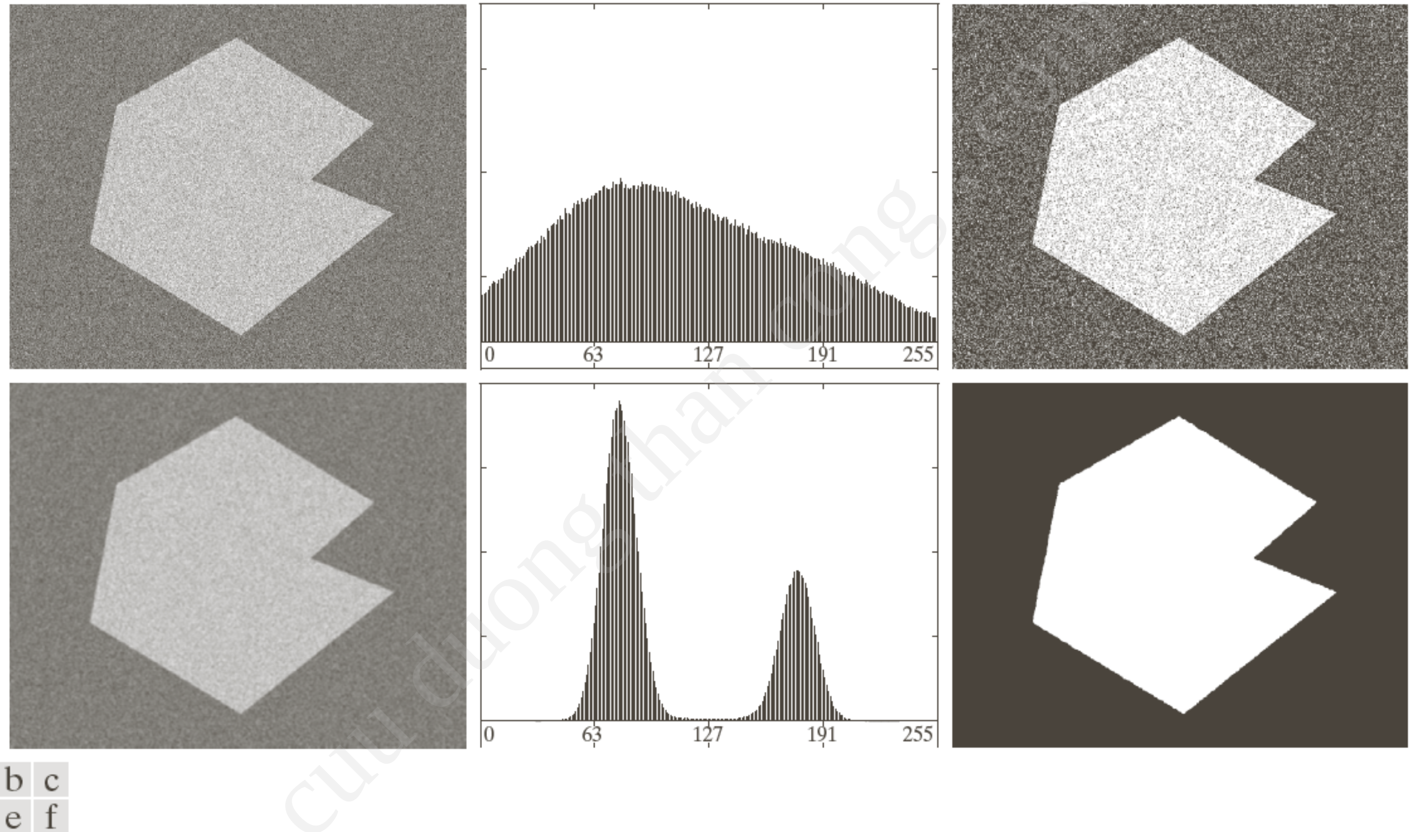


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

5. IS: Thresholding (16)

❑ **<u>Using edges to improve global thresholding:</u>**

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

5. IS: Thresholding (17)

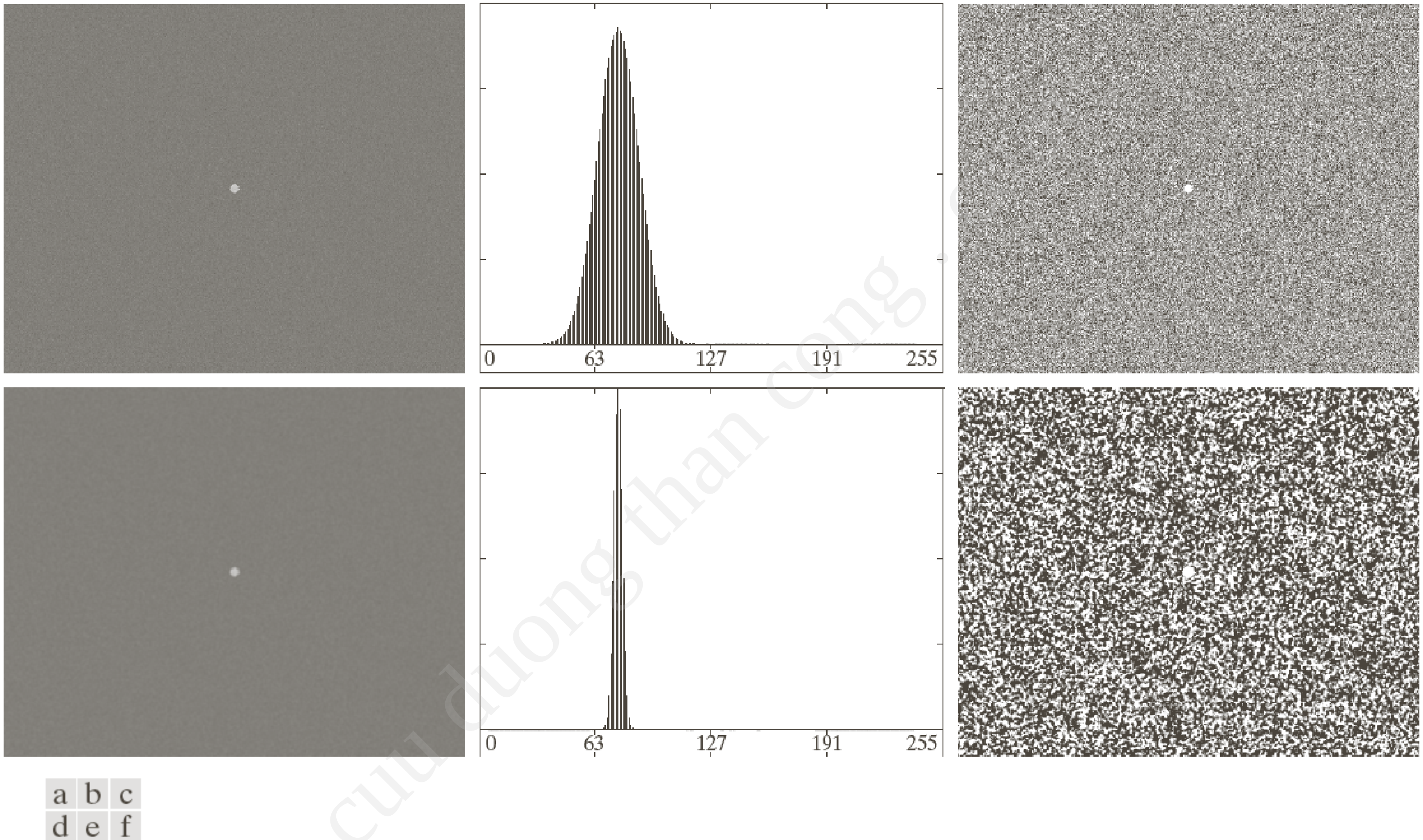
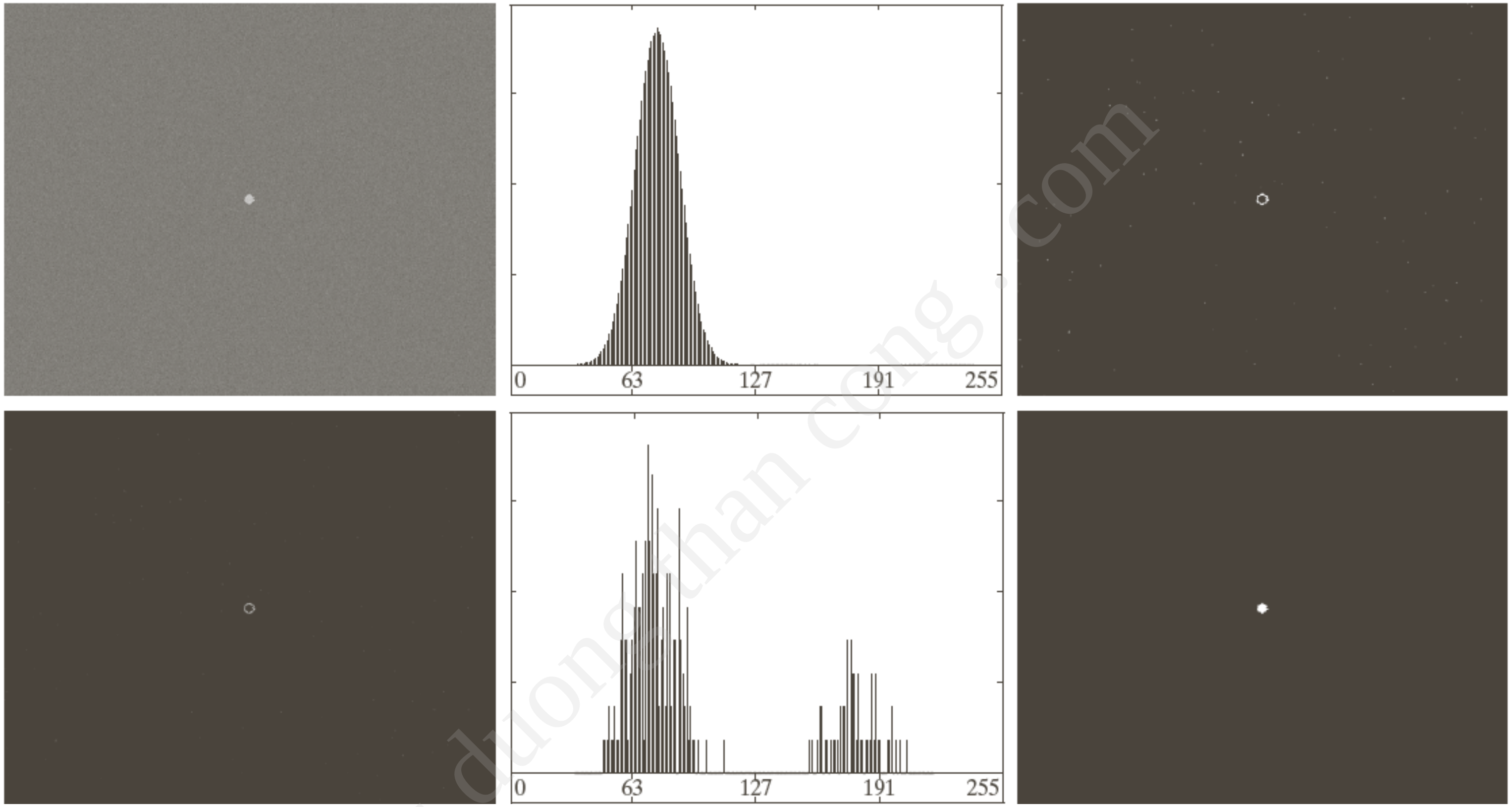


FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

5. IS: Thresholding (18)



a b c
d e f

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

5. IS: Thresholding (19)

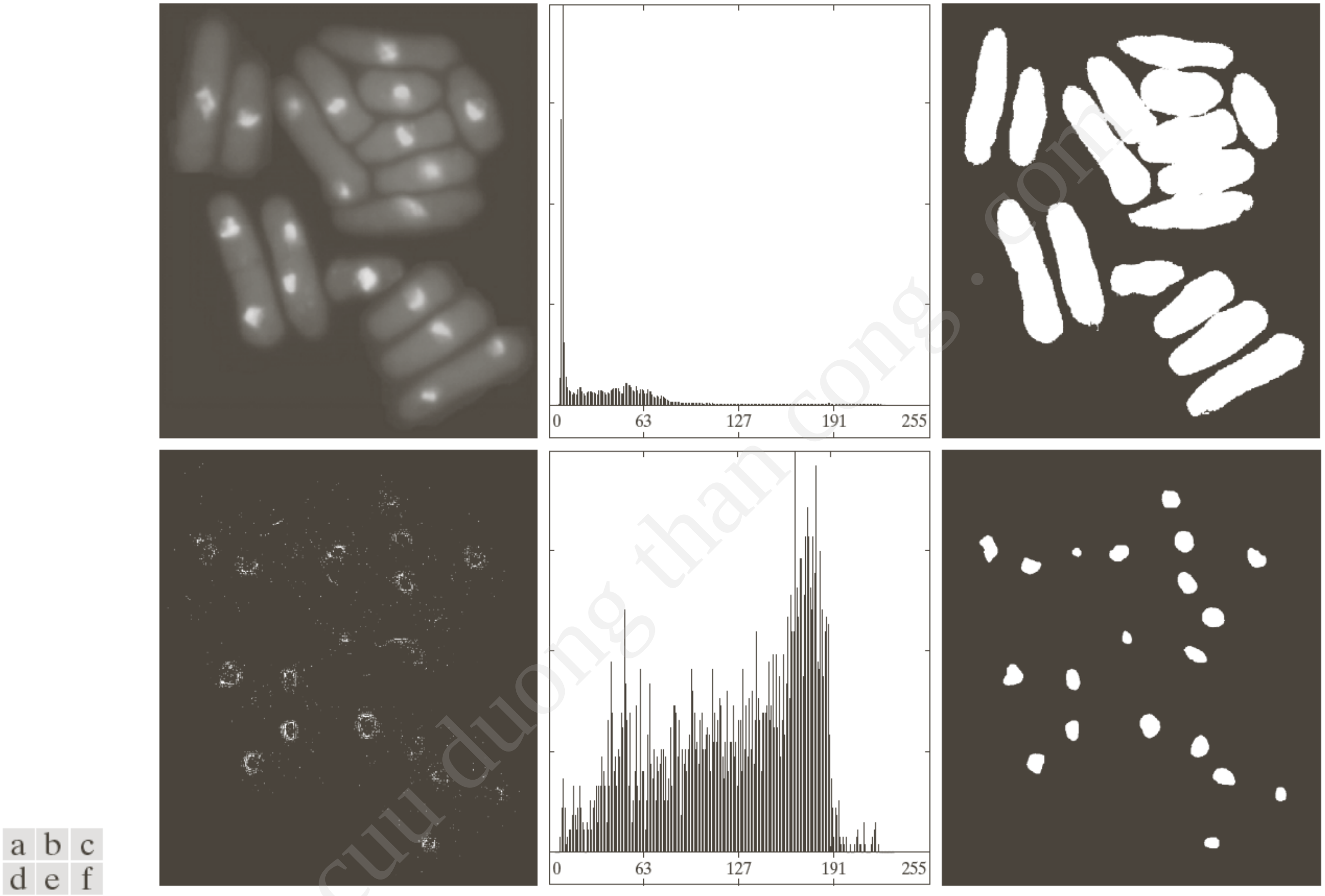


FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram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

5. IS: Thresholding (20)

❑ **Multiple thresholds:**

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

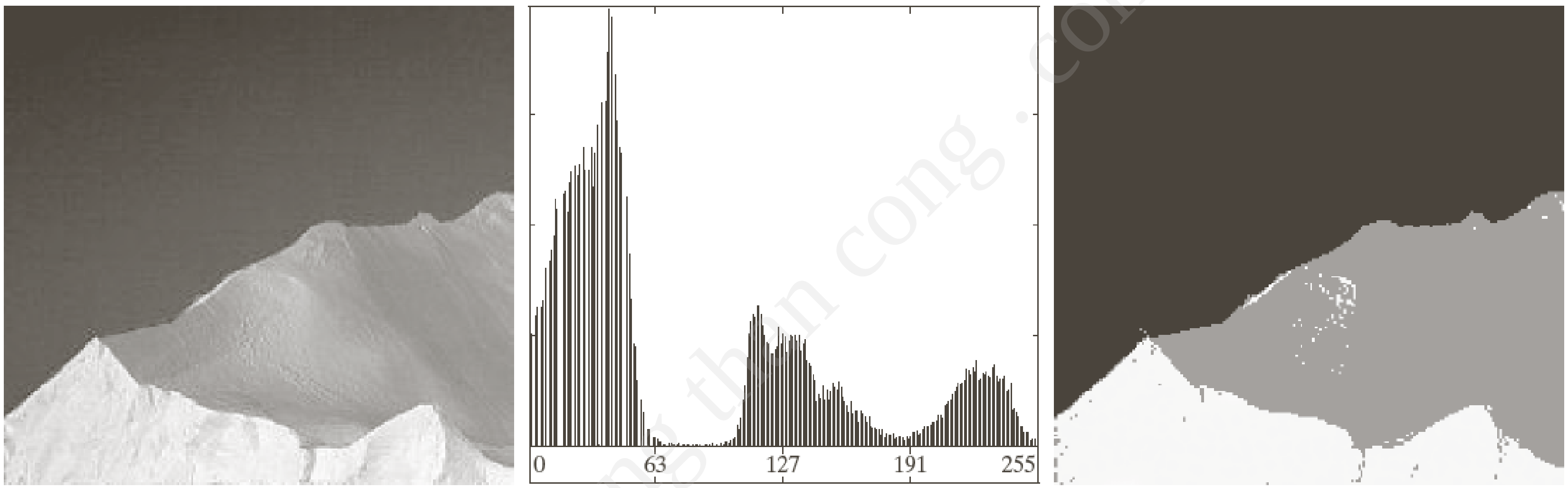
$$\sigma_B^2 = \sum_{k=1}^{K} P_k \left(m_k - m_G\right)^2$$

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

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

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

5. IS: Thresholding (21)



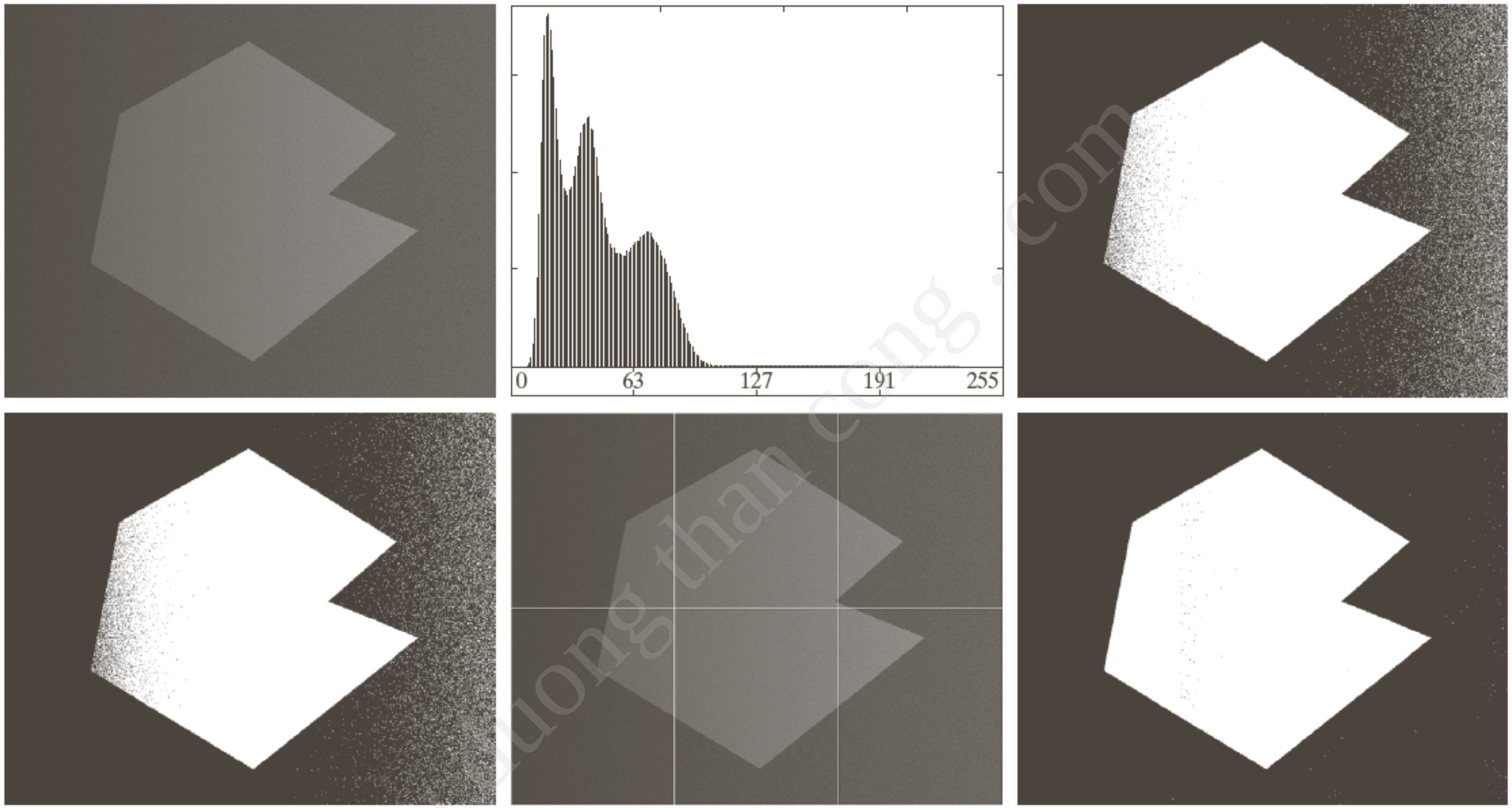
a b c

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

5. IS: Thresholding (22)

- **<u>Variable thresholding: Image partitioning (adaptive thresholding):</u>**
  - Subdivide original image into small areas.
  - Utilize a different threshold to segment each subimages.
  - Since the threshold used for each pixel depends on the location of the pixel in terms of the subimages, this type of thresholding is adaptive.

5. IS: Thresholding (23)



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.

5. IS: Thresholding (24)

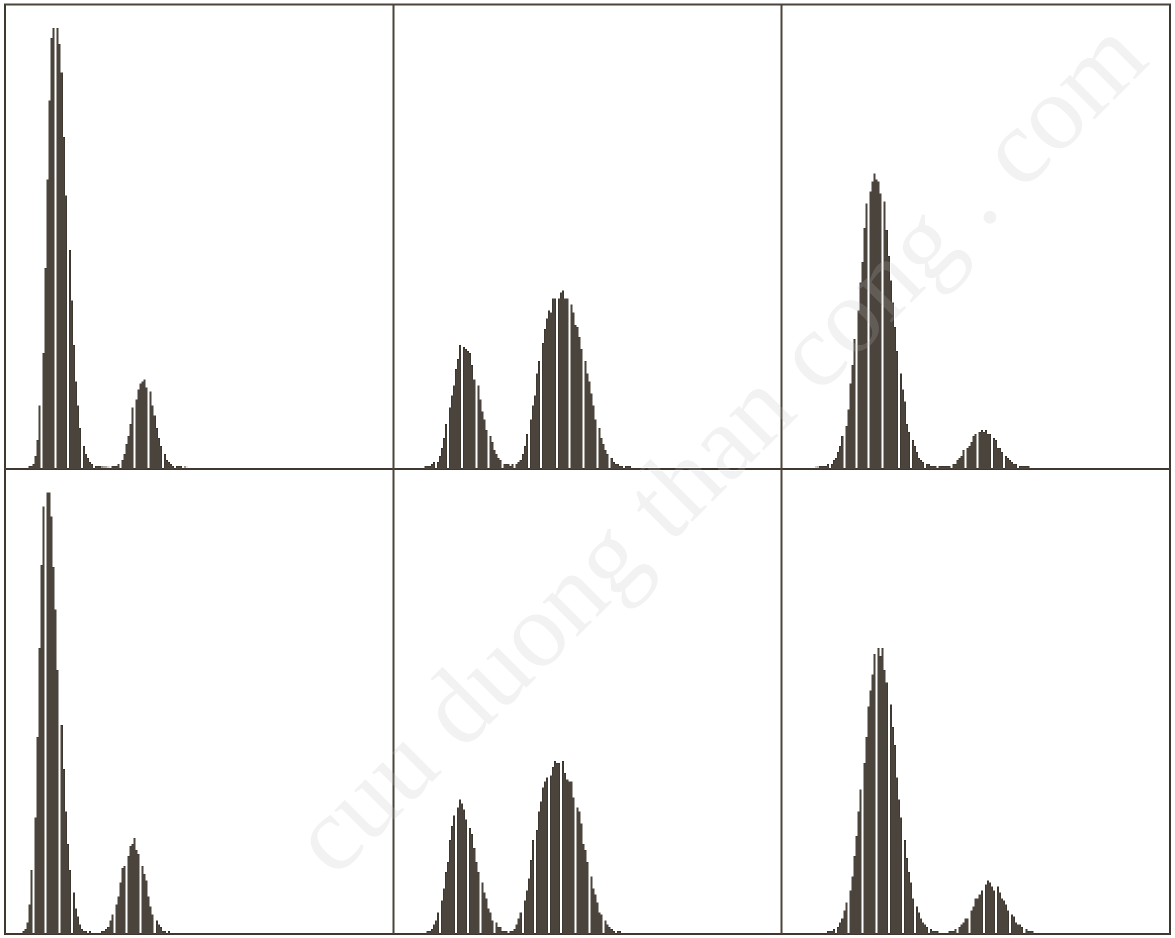


FIGURE 10.47
Histograms of the six subimages in Fig. 10.46(e).

5. IS: Thresholding (25)

- **<u>Variable thresholding based on local image properties</u>**:

  Let σ_{xy} and m_{xy} denote the standard deviation and mean value of the set of pixels contained in a neighborhood S_{xy}, centered at coordinates (x, y) in an image. The local thresholds,

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

  If the background is nearly constant,

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

  Then,

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

5. IS: Thresholding (26)

A modified thresholding:

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

E.g.,

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

5. IS: Thresholding (27)

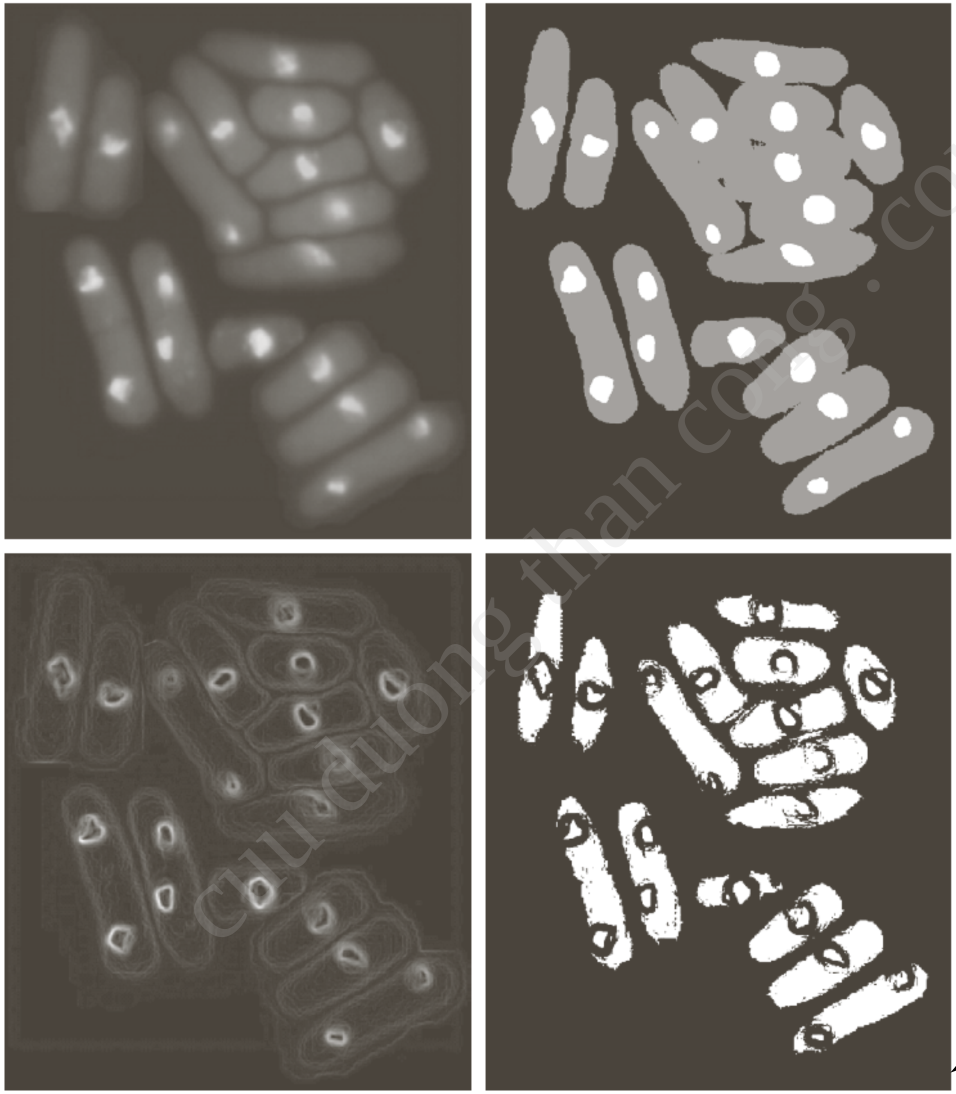


FIGURE 10.48 (a) Image from Fig. 10.43. (b) Image segmented using the dual thresholding approach discussed in Section 10.3.6. (c) Image of local standard deviations. (d) Result obtained using local thresholding.

a = 30
b = 1.5
$m_{xy} = m_G$

5. IS: Thresholding (28)

❑ **<u>Variable thresholding using moving averages:</u>**

– Thresholding based on moving averages works well when the objects are small with respect to the image size.

– Quite useful in document processing.

– The scanning (moving) typically is carried out line by line in zigzag pattern to reduce illumination bias.

5. IS: Thresholding (29)

Let z_{k+1} denote the intensity of the point encountered in the scanning sequence at step $k+1$. The moving average (mean intensity) at this new point is given by

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

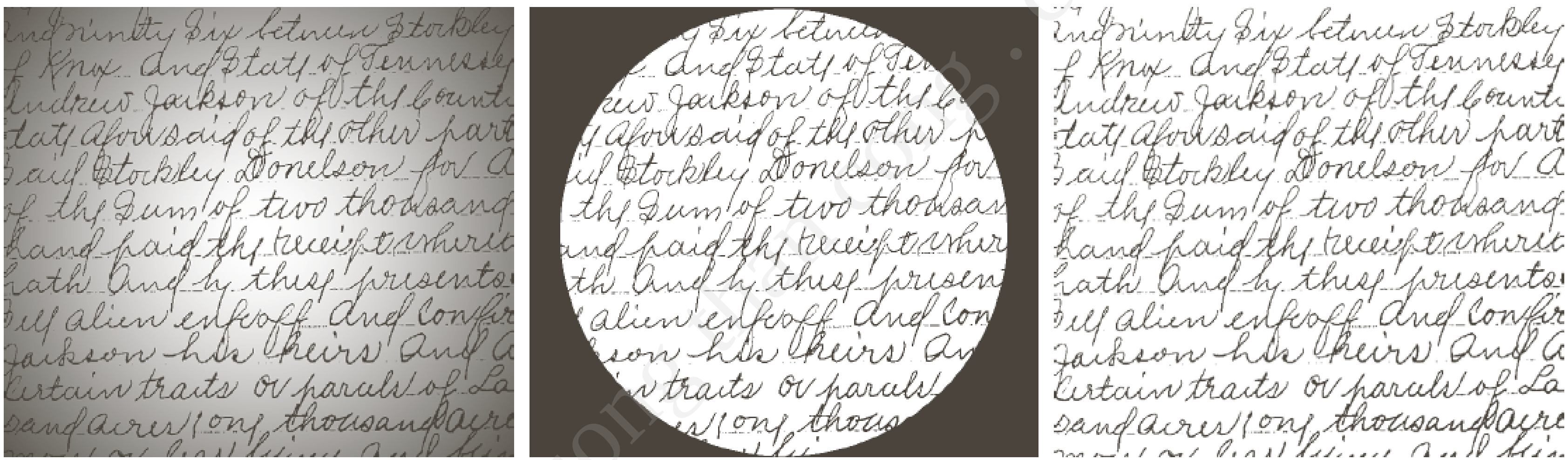
where n denotes the number of points used in computing the average and $m(1) = z_1 / n$, the border of the image were padded with n-1 zeros.

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

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

5. IS: Thresholding (30)

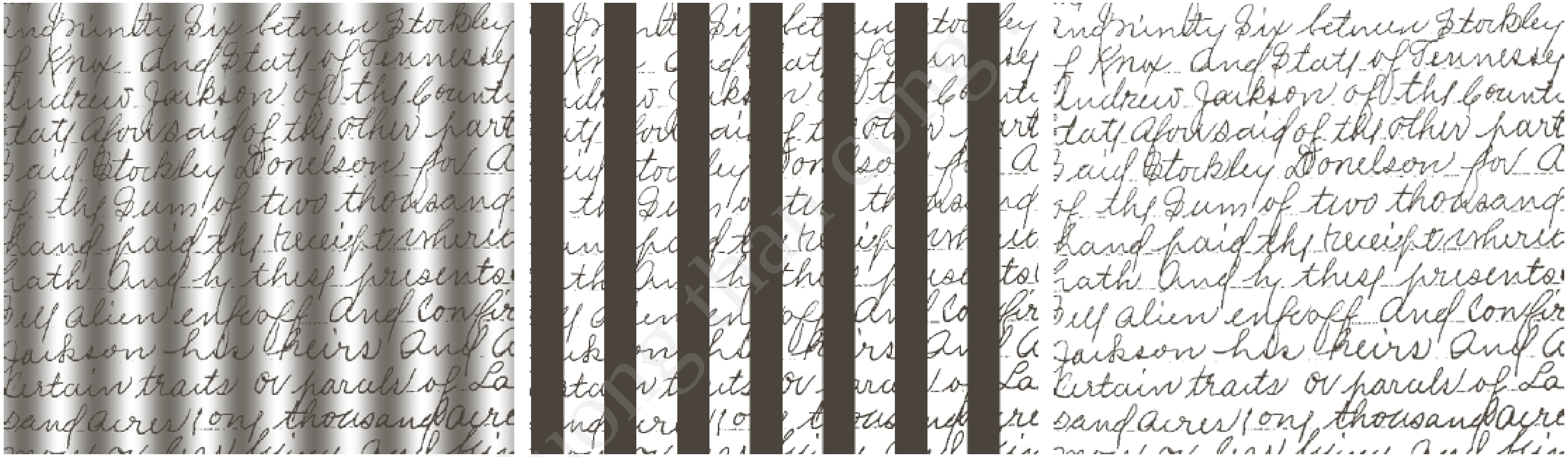
$n = 20$
$b = 0.5$



a b c

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

5. IS: Thresholding (31)



a b c

FIGURE 10.50 (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

5. IS: Region-Based Segmentation (1)

- **<u>Region Growing</u>:**
  - Region growing is a procedure that groups pixels or subregions into larger regions.
  - The simplest of these approaches is **pixel aggregation**, which starts with a set of "**seed**" points and from these grows regions by appending to each seed points those **neighboring pixels** that have **similar properties** (such as gray level, texture, color, shape).
  - Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.

5. IS: Region-Based Segmentation (2)

- **Basic Formulation**:

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

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

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

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

$$(e) \quad P(R_i \cup R_j) = FALSE \quad for \; i \neq j$$

$P(R_i)$ is a logical predicate property defined over the points in set R_i

e.g. $P(R_i)$ = TRUE if all pixel in R_i have the same gray level.

5. IS: Region-Based Segmentation (3)

Example: Region growing based on 8-connectivity

$f(x, y)$: input image array

$S(x, y)$: seed array containing 1s (seeds) and 0s

$Q(x, y)$: predicate

5. IS: Region-Based Segmentation (4)

1. Find all connected components in $S(x, y)$ and erode each connected components to one pixel; label all such pixels found as 1. All other pixels in S are labeled 0.
2. Form an image f_Q such that, at a pair of coordinates (x,y), let $f_Q(x, y) = 1$ if the Q is satisfied otherwise $f_Q(x, y) = 0$.
3. Let g be an image formed by appending to each seed point in S all the 1-value points in f_Q that are 8-connected to that seed point.
4. Label each connencted component in g with a different region label. This is the segmented image obtained by region growing.

5. IS: Region-Based Segmentation (5)

$$Q = \begin{cases} \text{TRUE} & \text{if the absolute difference of the intensities} \\ & \text{between the seed and the pixel at } (x, y) \text{ is } \le T \\ \text{FALSE} & \text{otherwise} \end{cases}$$

5. IS: Region-Based Segmentation (6)

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	<u>60</u>	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

5. IS: Region-Based Segmentation (7)

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

4-connectivity

5. IS: Region-Based Segmentation (8)

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

8-connectivity

5. IS: Region-Based Segmentation (9)

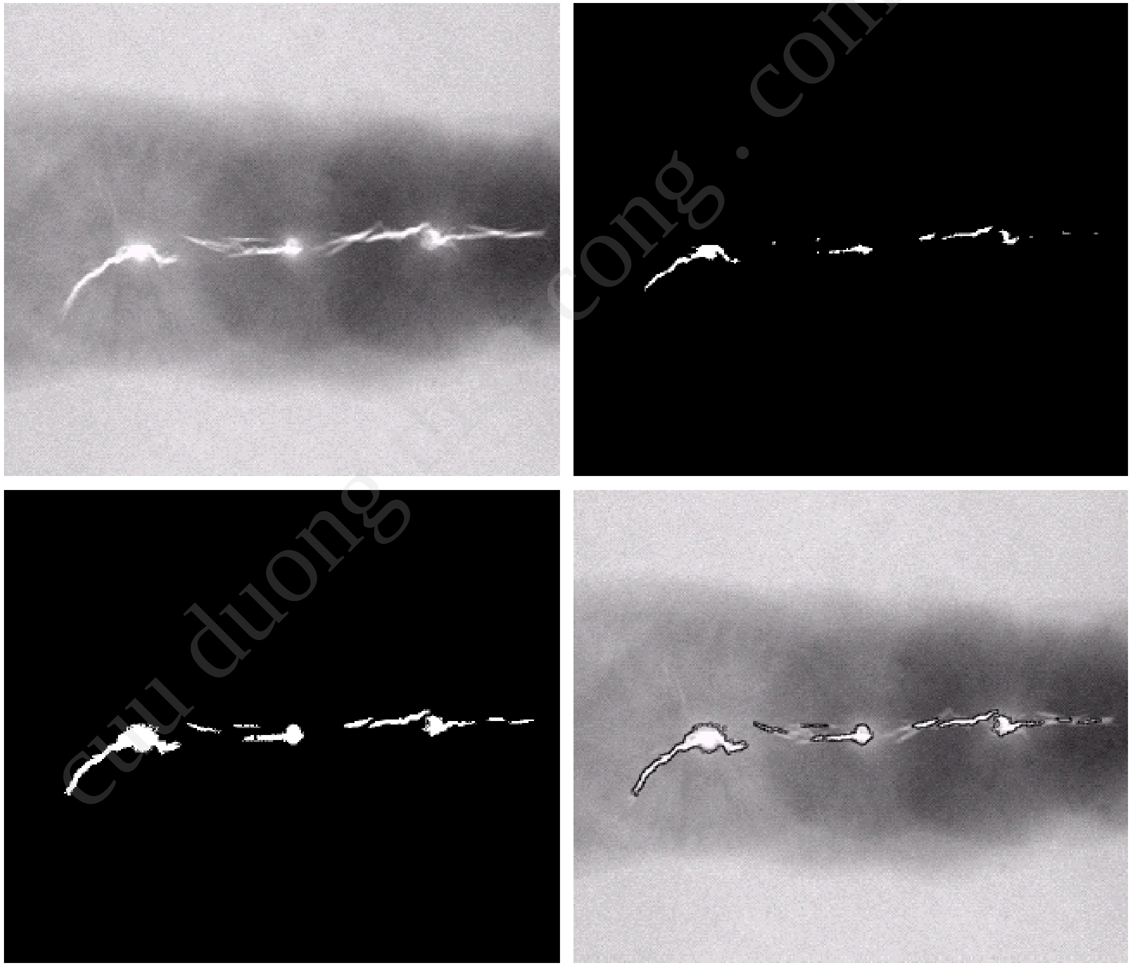
Example of Region-Based Segmentation:

Criteria:

1. The absolute gray-level difference between any pixel and the seed has to be less than 65 (based on the histogram next 2 slides).
2. The pixel has to be 8-connected to at least one pixel in that region (if more, the regions are merged).

5. IS: Region-Based Segmentation (10)

Select all seed points with gray level of 255



a b
c d

FIGURE 10.40 (a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).

Boundaries (in black)

5. IS: Region-Based Segmentation (11)

Histogram of Fig. 10.40(a) used to find the criteria of the difference gray-level between each pixels and the seeds.

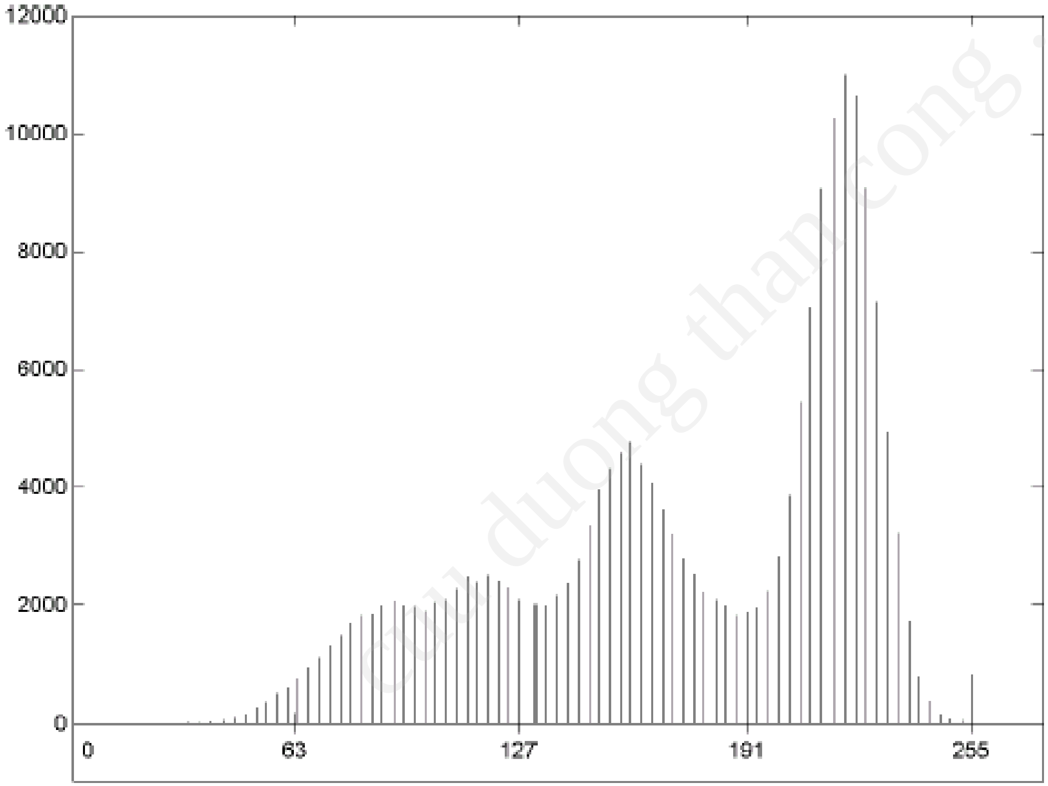


FIGURE 10.41
Histogram of Fig. 10.40(a).

5. IS: Region-Based Segmentation (12)

❑ Region Splitting and Merging: Quadtree

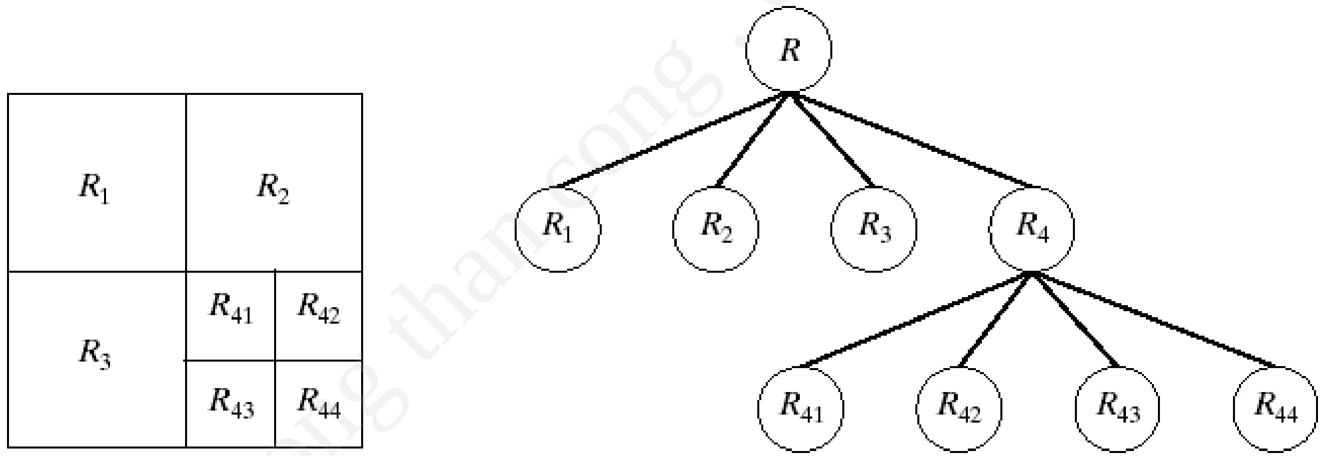
1. Split into 4 disjoint quadrants any region R_i for which $P(R_i)$ = FALSE.
2. Merge any adjacent region R_j and R_k for which $P(R_i \cup R_k)$ = TRUE.
3. Stop when no further merging or splitting is possible.

$$P = \begin{cases} \text{TRUE} & \text{if } \sigma > a \text{ and } 0 < m < b \\ \text{FALSE} & \text{otherwise} \end{cases}$$

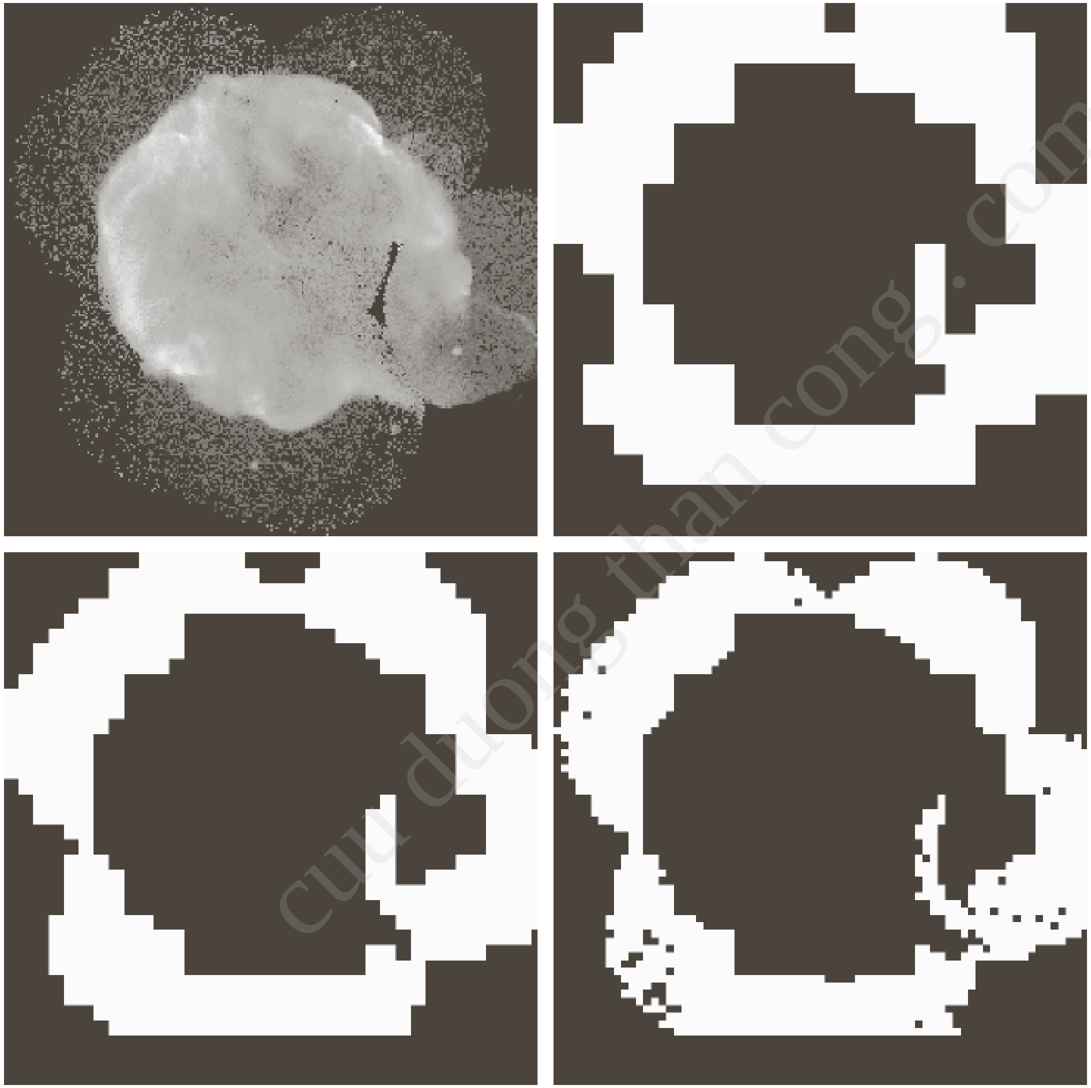
5. IS: Region-Based Segmentation (13)

a b

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



5. IS: Region-Based Segmentation (14)



a b
c d

FIGURE 10.53
(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

5. IS: Region-Based Segmentation (15)

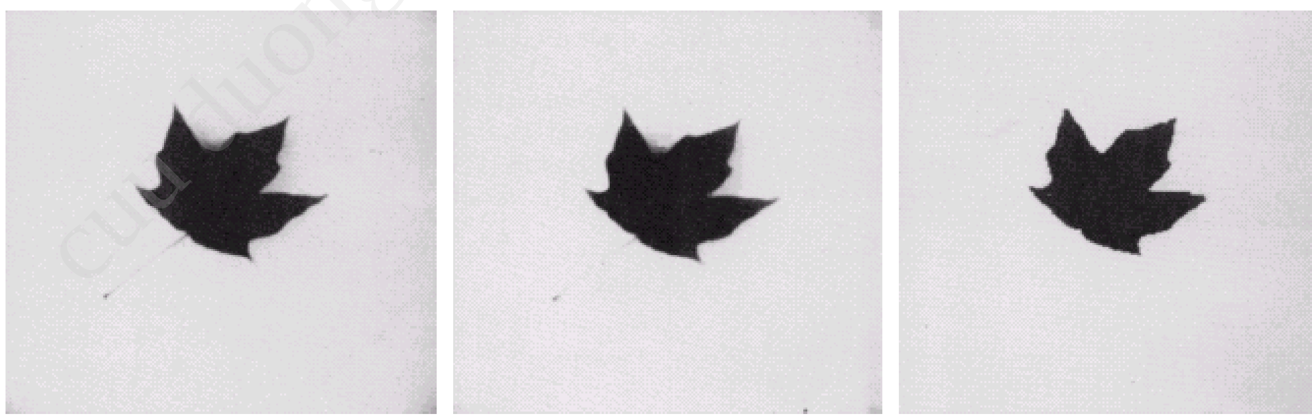
Example of Region splitting and merging using Quadtree:

$P(R_i)$ = TRUE if at least 80% of the pixels in R_i have the property $|z_j - m_i| \le 2\sigma_i$,
where:

z_j is the gray level of the j^{th} pixel in R_i
m_i is the mean gray level of that region
σ_i is the standard deviation of the gray levels in R_i

a b c

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



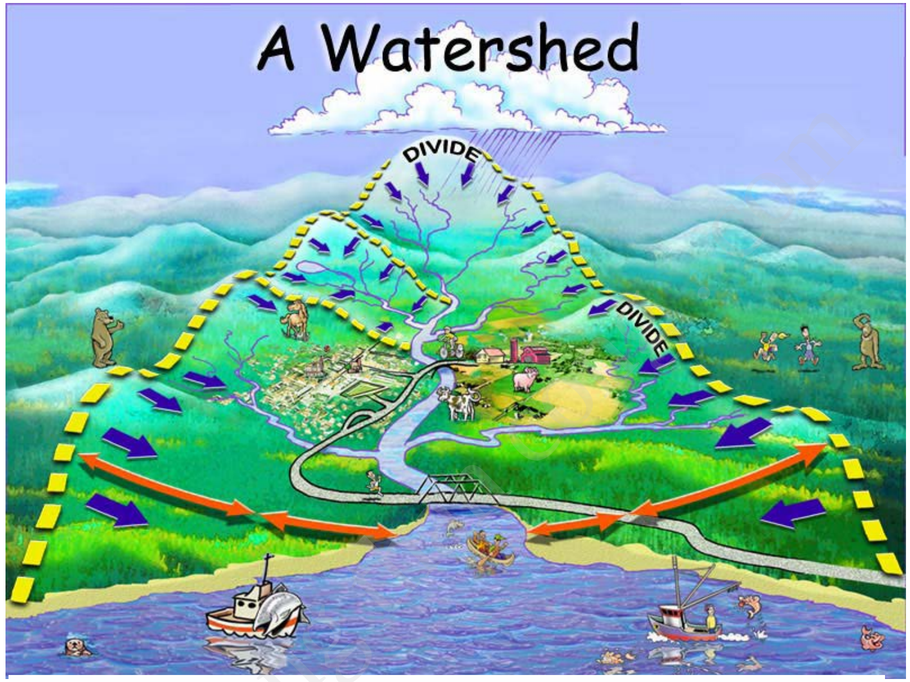
5. IS: Segmentation Using Morphological Watersheds (1)

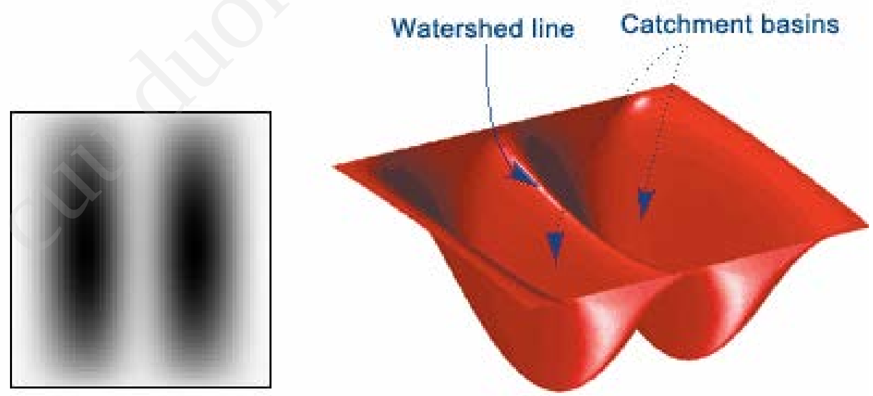
Based on geography:
– Ridge (watershed) divides areas drained by different river systems.
– Catchment basin drains into a river or reservoir.

Applied to an image:
– Visualize image in 3D.
– Two spatial coordinates (x,y).
– Versus image intensity (z).

Watershed transform:
– Find catchment basins and watershed lines.
– Using gray scale image.
– Catchment basin $\rightarrow$ object of interest.

5. IS: Segmentation Using Morphological Watersheds (2)

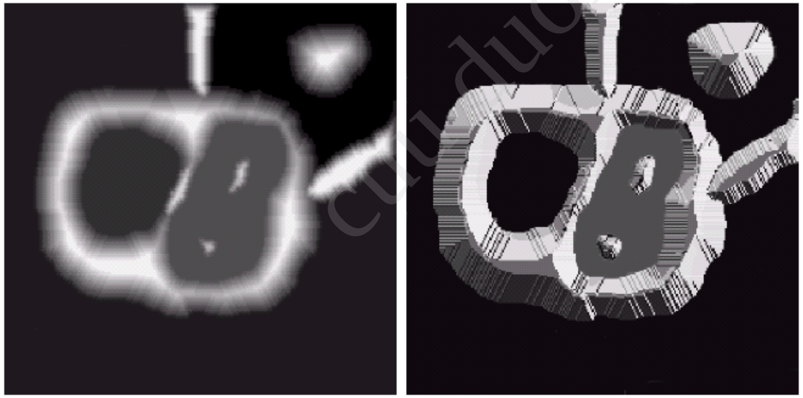


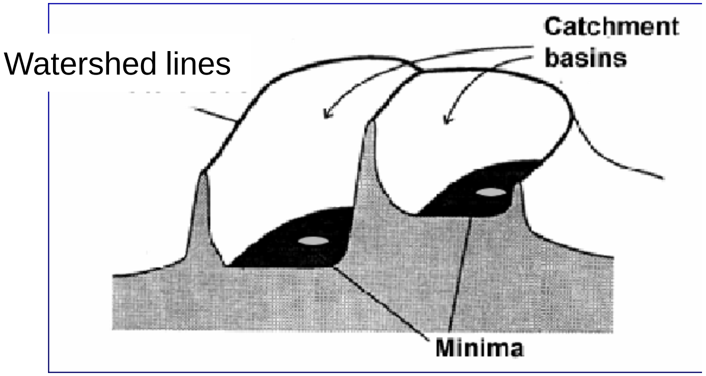


5. IS: Segmentation Using Morphological Watersheds (3)

Three types of points in a topographic interpretation:

- Points belonging to a regional minimum.
- Points at which a drop of water would fall to a single minimum (**catchment basin** or **watershed** of that minimum).
- Points at which a drop of water would be equally likely to fall to more than one minimum (**divide lines** or **watershed lines**).





5. IS: Segmentation Using Morphological Watersheds (4)

❑ Background:

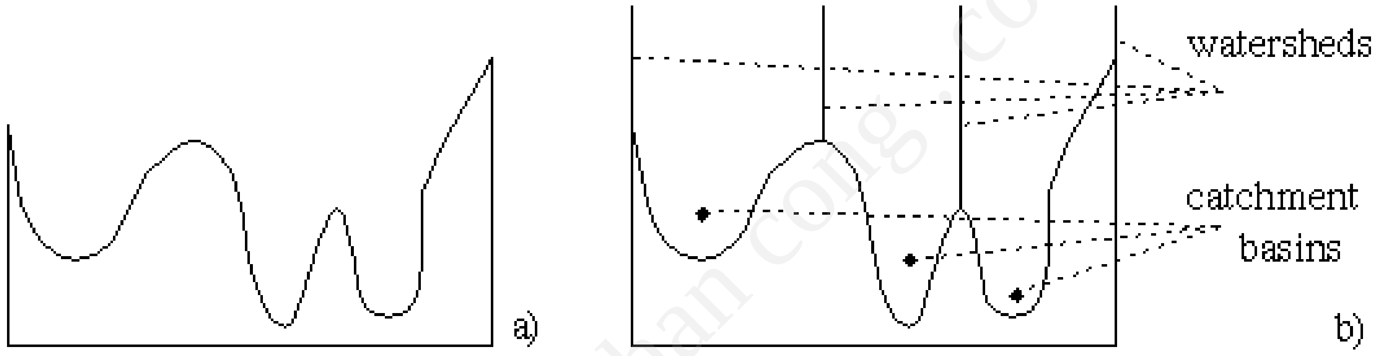
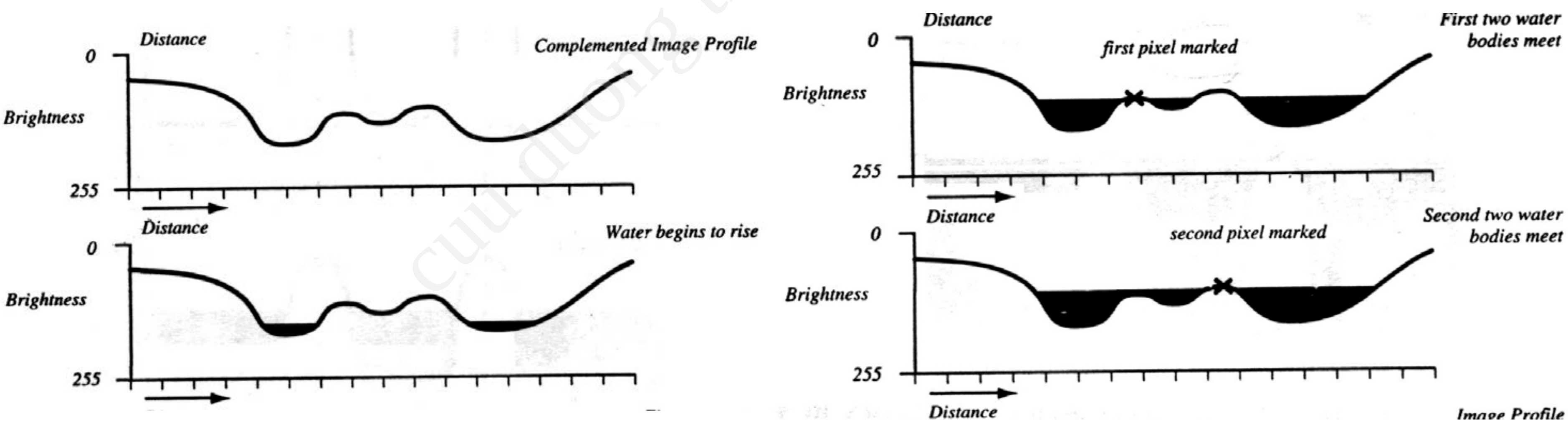


Figure 5.47 *One-dimensional example of watershed segmentation. (a) Gray level profile of image data. (b) Watershed segmentation – local minima of gray level (altitude) yield catchment basins, local maxima define the watershed lines.*

5. IS: Segmentation Using Morphological Watersheds (5)

Objective is to find watershed lines.

- Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate.
- When rising water in distinct catchment basins is about the merge, a dam is built to prevent merging. These dam boundaries correspond to the watershed lines.



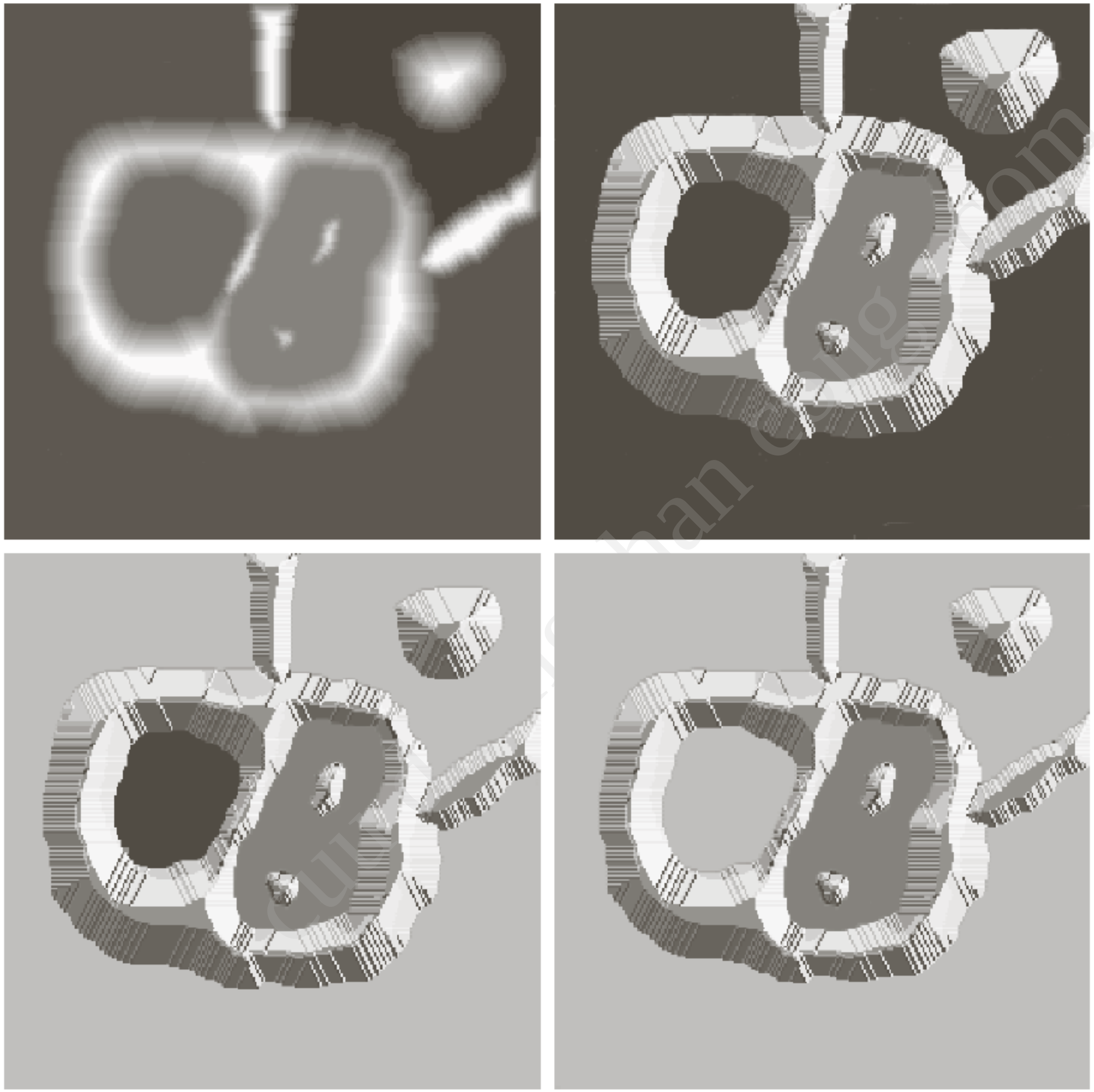
5. IS: Segmentation Using Morphological Watersheds (6)

Basic algorithm:

1. Find the regional minima in image
2. Flood the entire topography
   - Start from regional minima
   - Use a uniform rate
3. Build dam to prevent merging
   - Water rises in distinct watershed
   - Done when close to merging distinct watersheds
4. Stop flooding when only the tops of the dams are visible above the water line
5. Dam boundaries correspond watershed divide lines.

Note: Often done on gradient of image

5. IS: Segmentation Using Morphological Watersheds (7)



a b
c d

FIGURE 10.54
(a) Original image.
(b) Topographic view. (c)–(d) Two stages of flooding.

5. IS: Segmentation Using Morphological Watersheds (8)

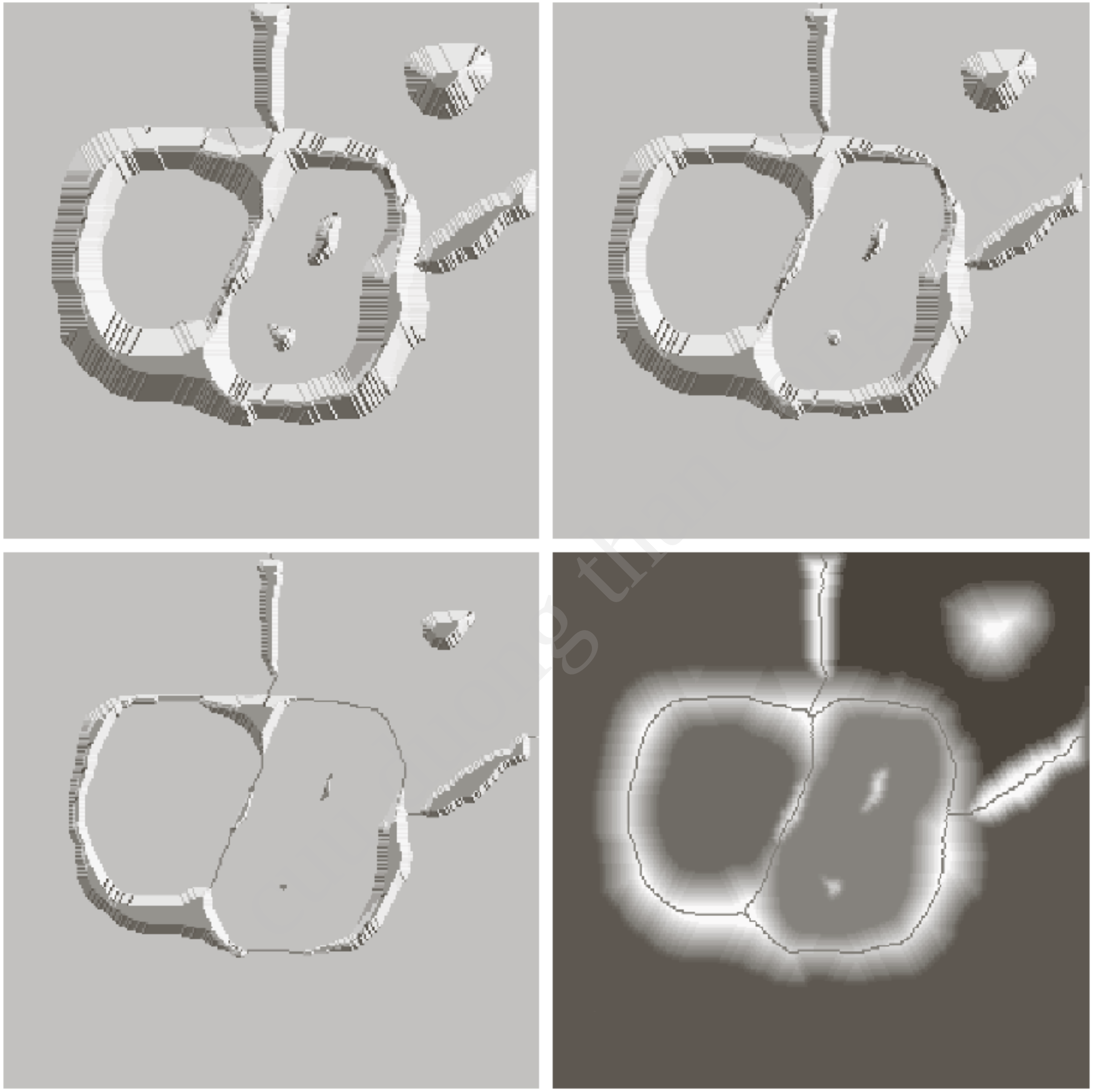


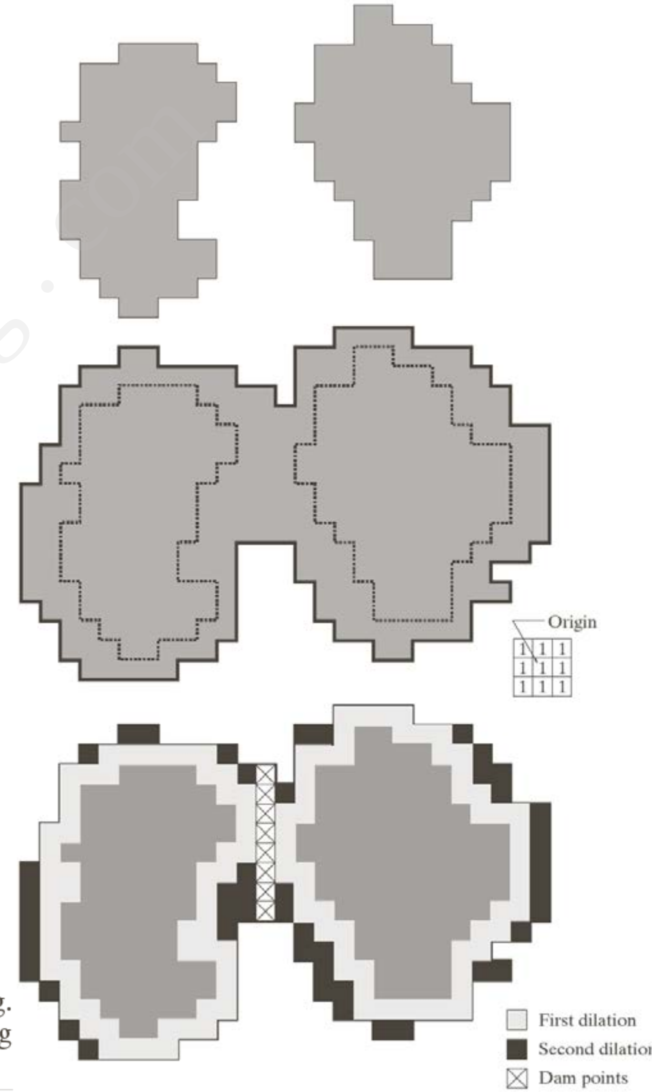
FIGURE 10.54 *(Continued)* (e) Result of further flooding. (f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

5. IS: Segmentation Using Morphological Watersheds (9)

- **Dam construction**:
  - M_i: coordinates of a regional minima point.
  - $C_{n\text{-}1}(M_i)$:
    - – coordinates of a point in the catchment basin.
    - – associated with M_i at stage n-1.
  - $C[n\text{-}1]$: union of $C_{n\text{-}1}(M_1)$ and $C_{n\text{-}1}(M_2)$
    - – Two components merge at the flooding step n.
    - – Denote this connected component as q.
  - Dilate each region by structure element
    1. Constrained to q.
    2. Cannot be performed on points causing sets being dilated to merge.

5. IS: Segmentation Using Morphological Watersheds (10)

- First dilation pass (right)
  - – Condition 1 satisfied
  - – Condition 2 satisfied
- Second dilation pass (right)
  - – Condition 1 failed
  - – Condition 2 satisfied



a
b c
d

FIGURE 10.55 (a) Two partially flooded catchment basins at stage $n-1$ of flooding. (b) Flooding at stage n, showing that water has spilled between basins. (c) Structuring element used for dilation. (d) Result of dilation and dam construction.

5. IS: Segmentation Using Morphological Watersheds (11)

- **Watershed segmentation algorithm:**
  - Start with all pixels with the lowest possible value.
  - – These form the basis for initial watersheds.
  - For each intensity level *k*:
  - – For each group of pixels of intensity *k*
    1. If adjacent to exactly one existing region, add these pixels to that region.
    2. Else if adjacent to more than one existing regions, mark as boundary.
    3. Else start a new region.

5. IS: Segmentation Using Morphological Watersheds (12)

Algorithm Definitions:

- Let $M_1, M_2, \ldots, M_R$ be sets:
  - Denoting the coordinates of the points.
  - In the regional minima of an image $g(x,y)$.
- Let $C(M_i)$ be a set:
  - Denoting the coordinates of points in catchment basin.
  - Associated with regional minimum M_i
- Let $T[n]$ represent the set of coordinates (s, t) for
  - $g(s,t) < n$
  - $T[n]=\{(s,t)|g(s,t) < n\}$
- Topology flooded in integer flood increments
  - From n = min+1 to n = max+1
  - min is the minimum value of $g(x,y)$
  - max is the maximum value of $g(x,y)$

5. IS: Segmentation Using Morphological Watersheds (13)

- $C_n(M_i)$ denote:
  – set of coordinates of points in the catchment basin.
  – Associated with M_i that are flooded at stage n.
- $C_n(M_i)$ can be viewed as:
  – A binary image given by $C_n(M_i) = C(M_i) \cap T[n]$
  – $C_n(M_i) = 1$ at location (x,y) if
    - $(x,y) \in C(Mi)$
    - $(x,y) \in T[n]$
  – Otherwise $C_n(M_i) = 0$
- C[n]: the union of the flooded catchment basins portion at stage n

$$C[n] = \bigcup_{i=1}^{R} C_n(M_i)$$

5. IS: Segmentation Using Morphological Watersheds (14)

- Then C[max+1] is the union of all catchment basins as:

$$C[\max+1]=\bigcup_{i=1}^{R} C(M_i)$$

- $C[n\text{-}1]$ is subset of $C[n]$ and $C[n]$ is subset of $T[n]$
- Initialize: $C[\min+1]=T[\min+1]$
- $Q[n]$: set of connected components in $T[n]$

5. IS: Segmentation Using Morphological Watersheds (15)

- Recursive algorithm obtaining C[n] from C[n-1] as follows:
- Let $p = q \cap C[n-1]$
- For each connected component $q \in Q[n]$

1. p is **empty**
   – i.e. a new minimum is encountered.
   – Add q into $C[n-1]$ to form $C[n]$
2. p contains **one** connected component
   – q lies within catchment basin of some regional minimum.
   – Add q into $C[n-1]$ to form $C[n]$
3. p contains **more than one** component
   – Flooding would cause water level in catchment basins to merge.
   – Dam must built within q by dilating of p.

5. IS: Segmentation Using Morphological Watersheds (16)

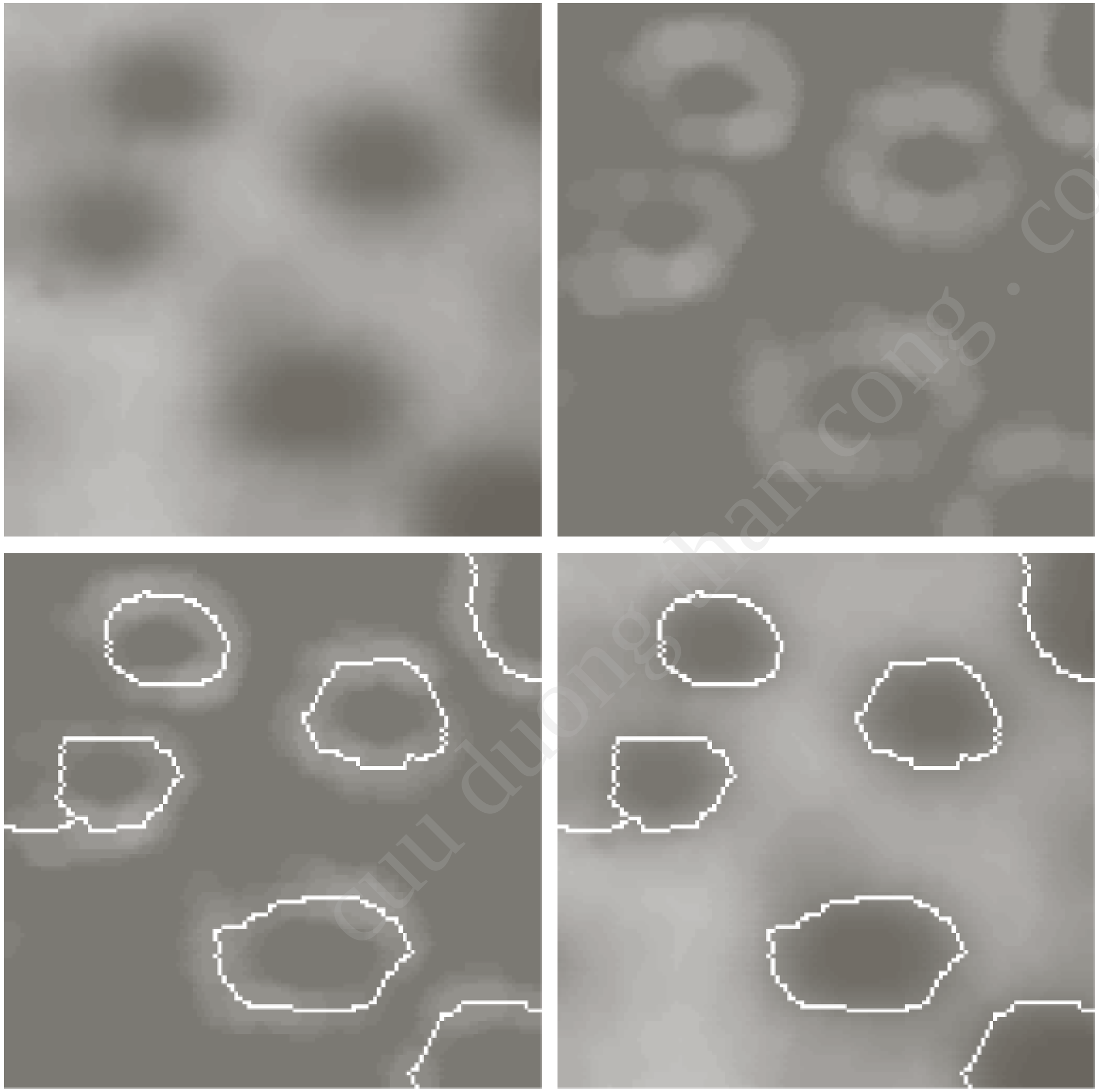
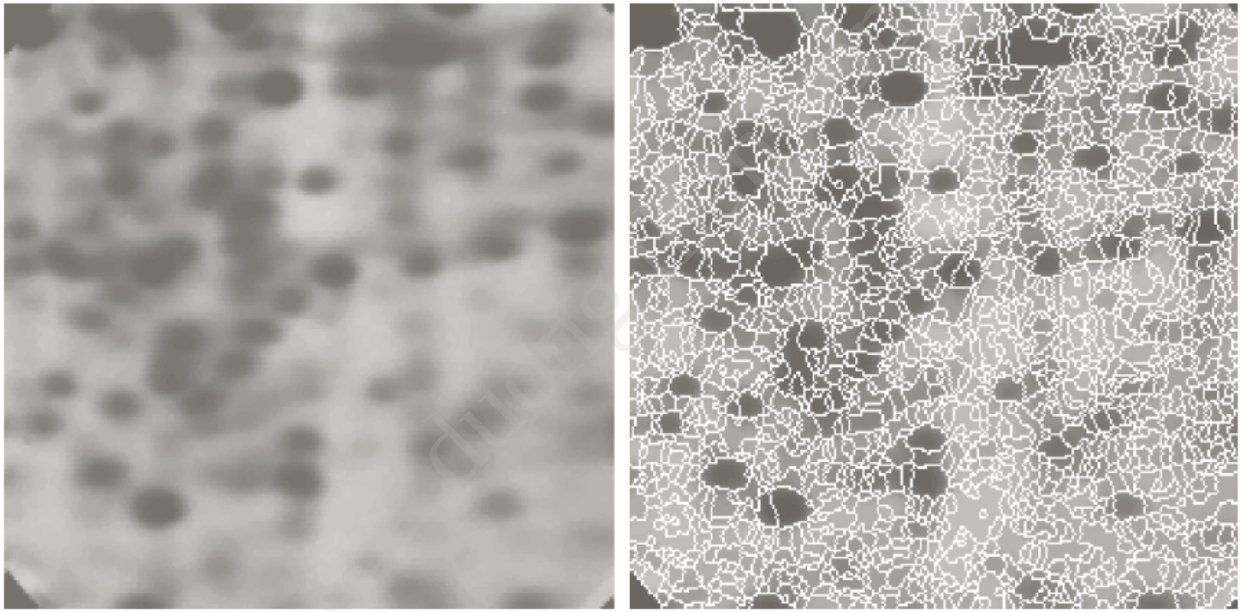


FIGURE 10.56
(a) Image of blobs.
(b) Image gradient.
(c) Watershed lines.
(d) Watershed lines superimposed on original image.
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

Watershed algorithm is often used on the gradient image instead of the original image.

❑ The use of marker

Due to noise and other local irregularities of the gradient, over-segmentation might occur.

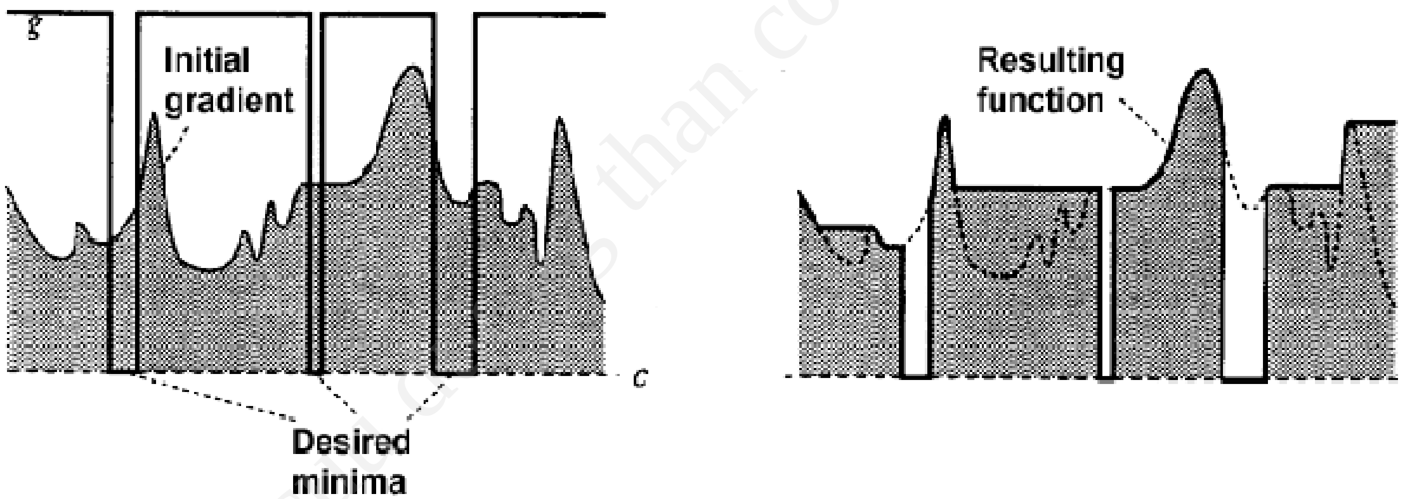


a b

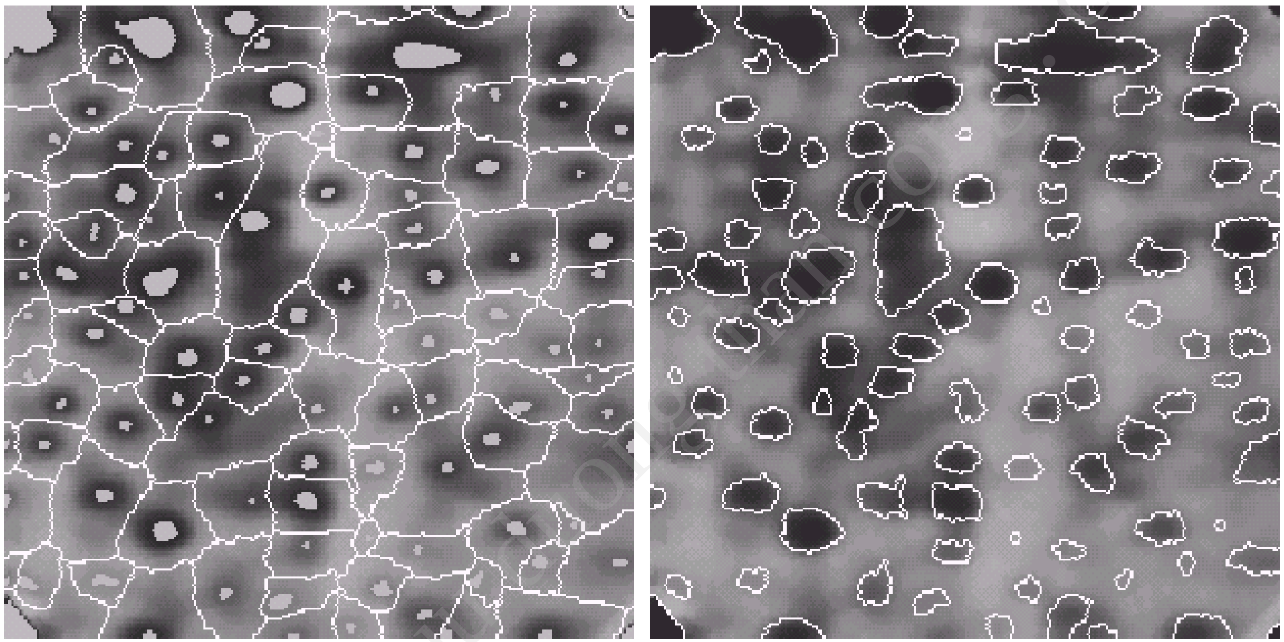
FIGURE 10.57 (a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

5. IS: Segmentation Using Morphological Watersheds (18)

A solution is to limit the number of regional minima. Use **markers** to specify the only allowed regional minima (for example, gray-level values might be used as a marker).



5. IS: Segmentation Using Morphological Watersheds (19)



a b

FIGURE 10.48 (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

5. IS: Segmentation Using Clustering Method (1)

❑ <u>K-means clustering:</u>

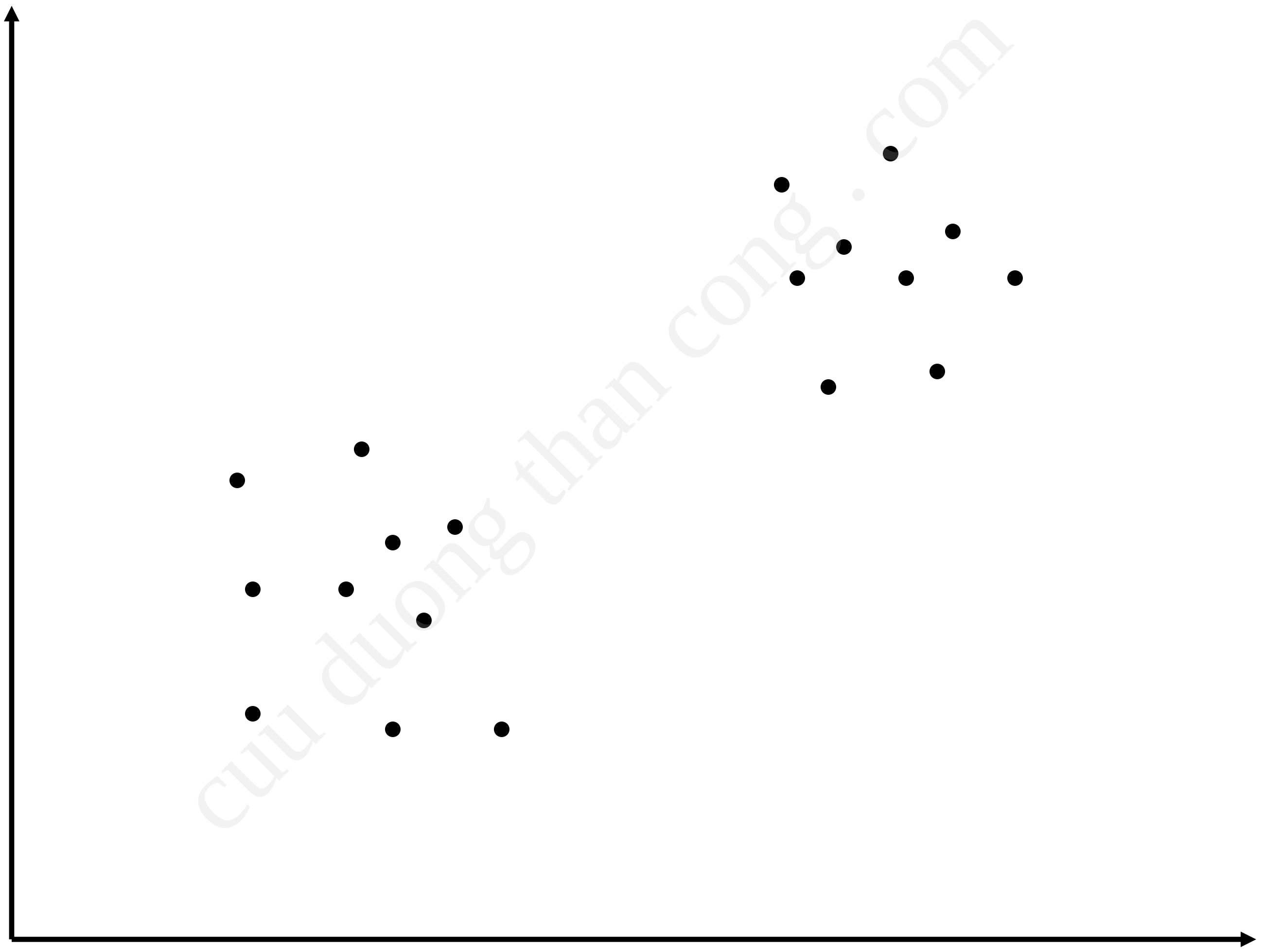
1. Partition the data points into K clusters randomly. Find the centroids of each cluster.

2. For each data point:
   - Calculate the distance from the data point to each cluster (i.e. to the centroid of the cluster).
   - Assign the data point to the closest cluster.

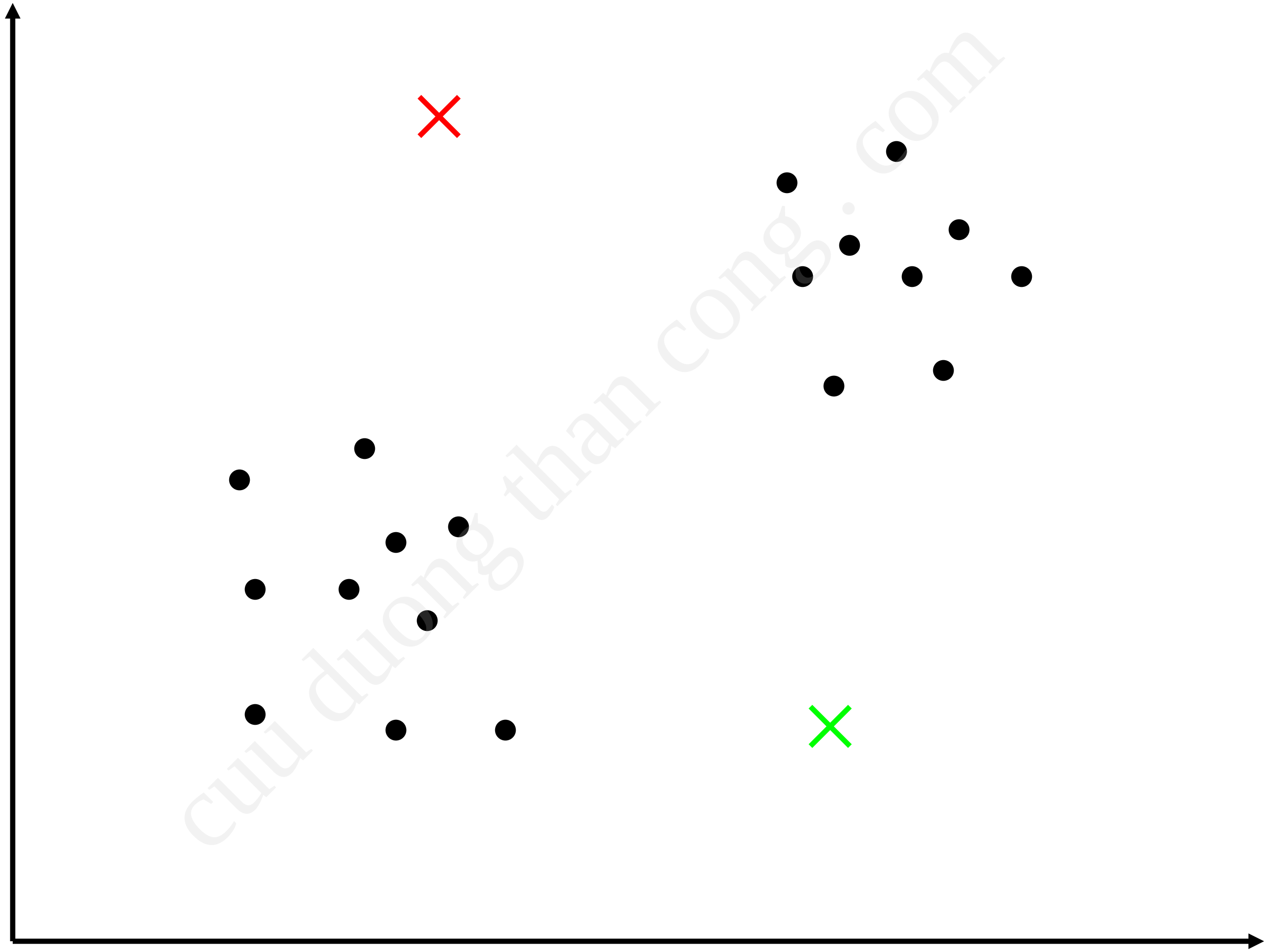
3. Recompute the centroid of each cluster.

4. Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

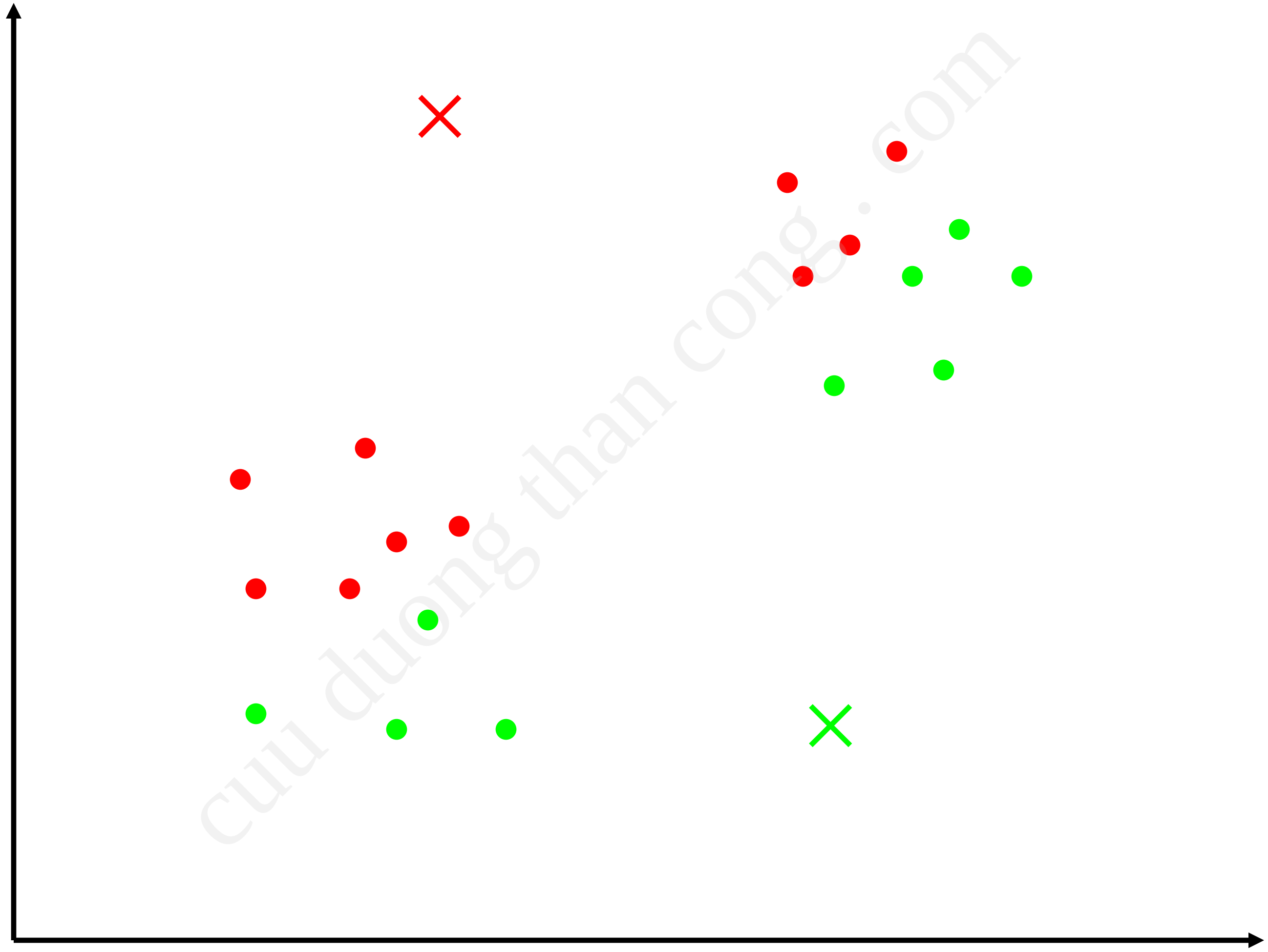
5. IS: Segmentation Using Clustering Method (2)



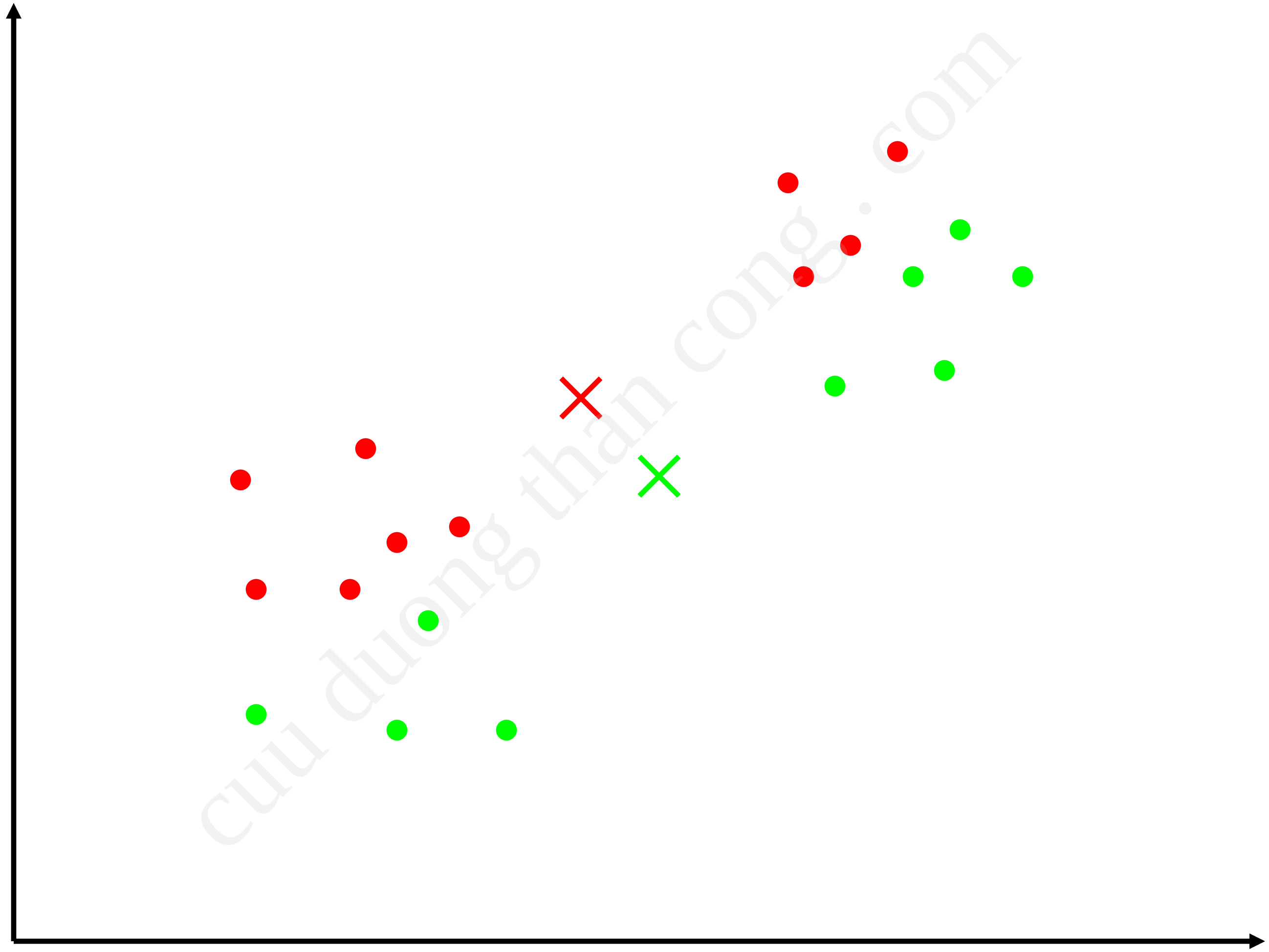
5. IS: Segmentation Using Clustering Method (3)



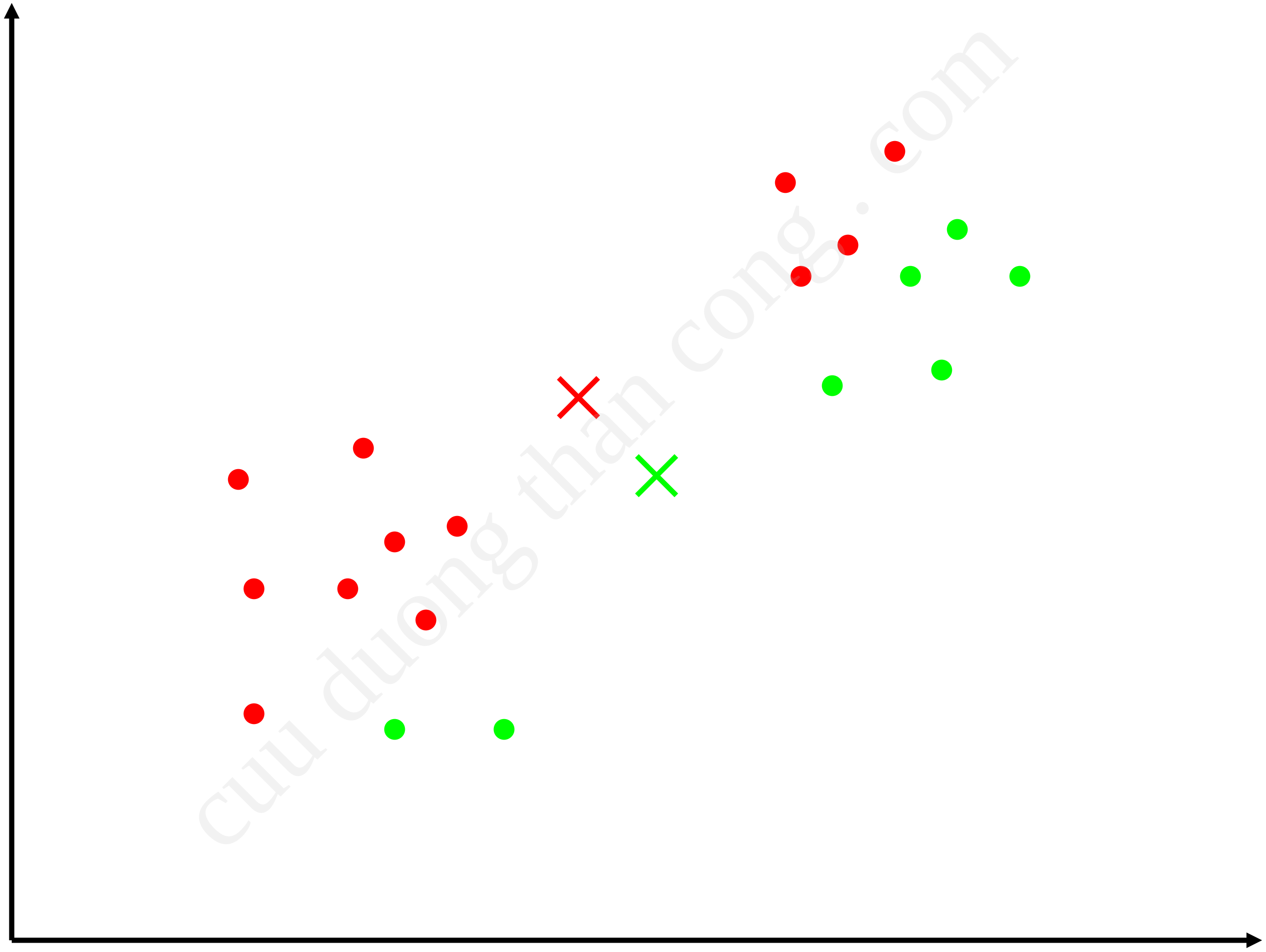
5. IS: Segmentation Using Clustering Method (3)



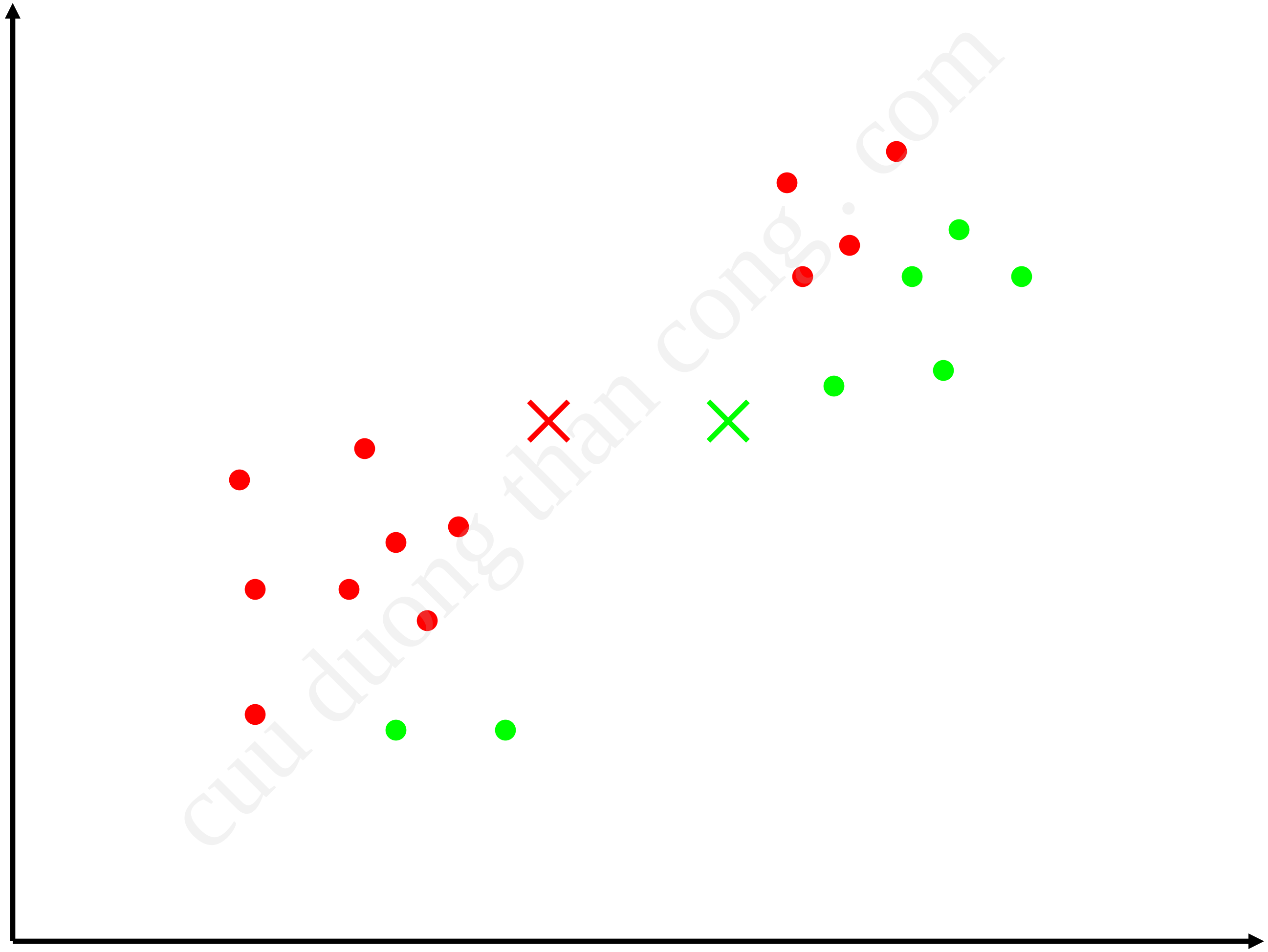
5. IS: Segmentation Using Clustering Method (4)



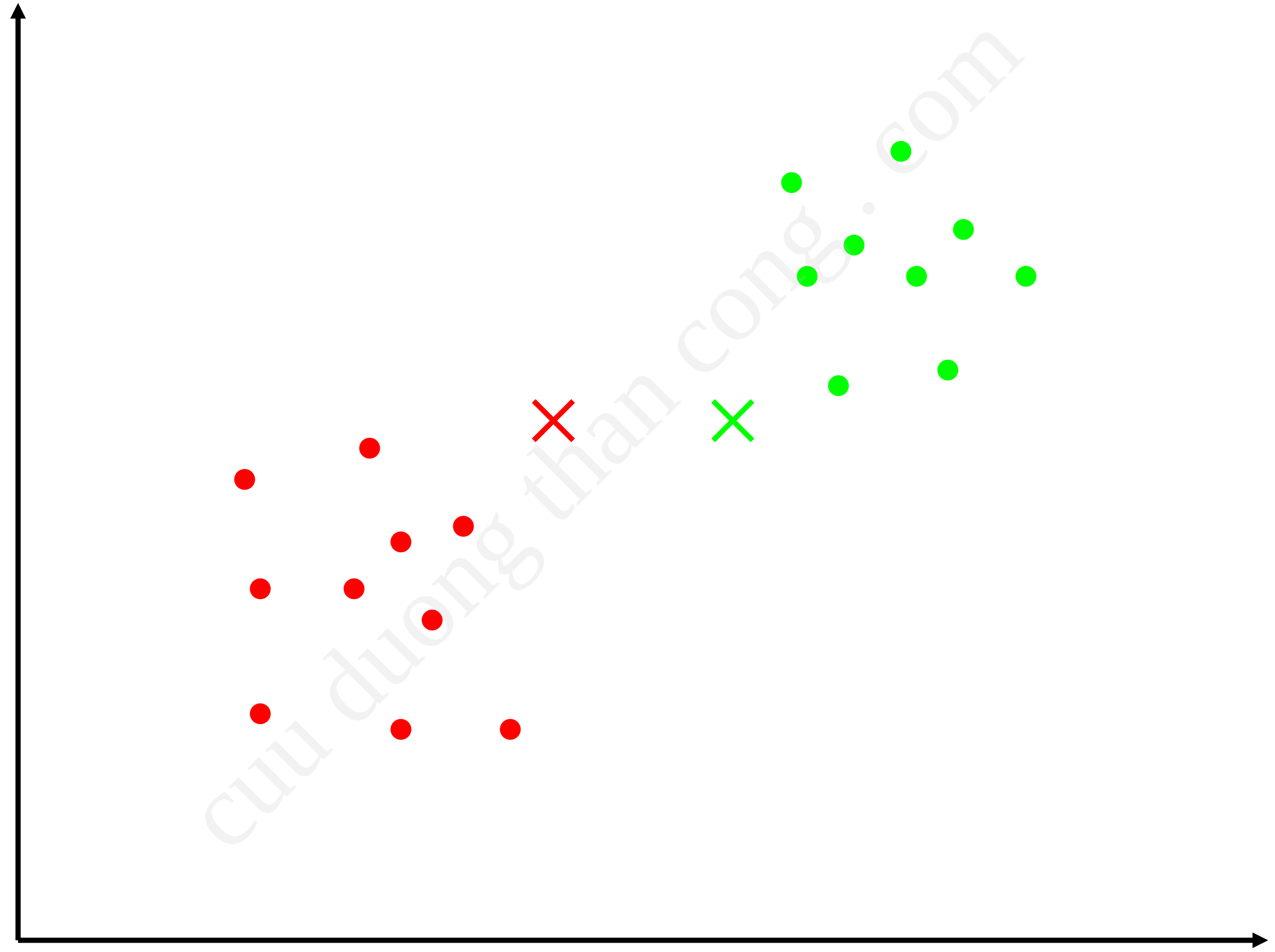
5. IS: Segmentation Using Clustering Method (5)



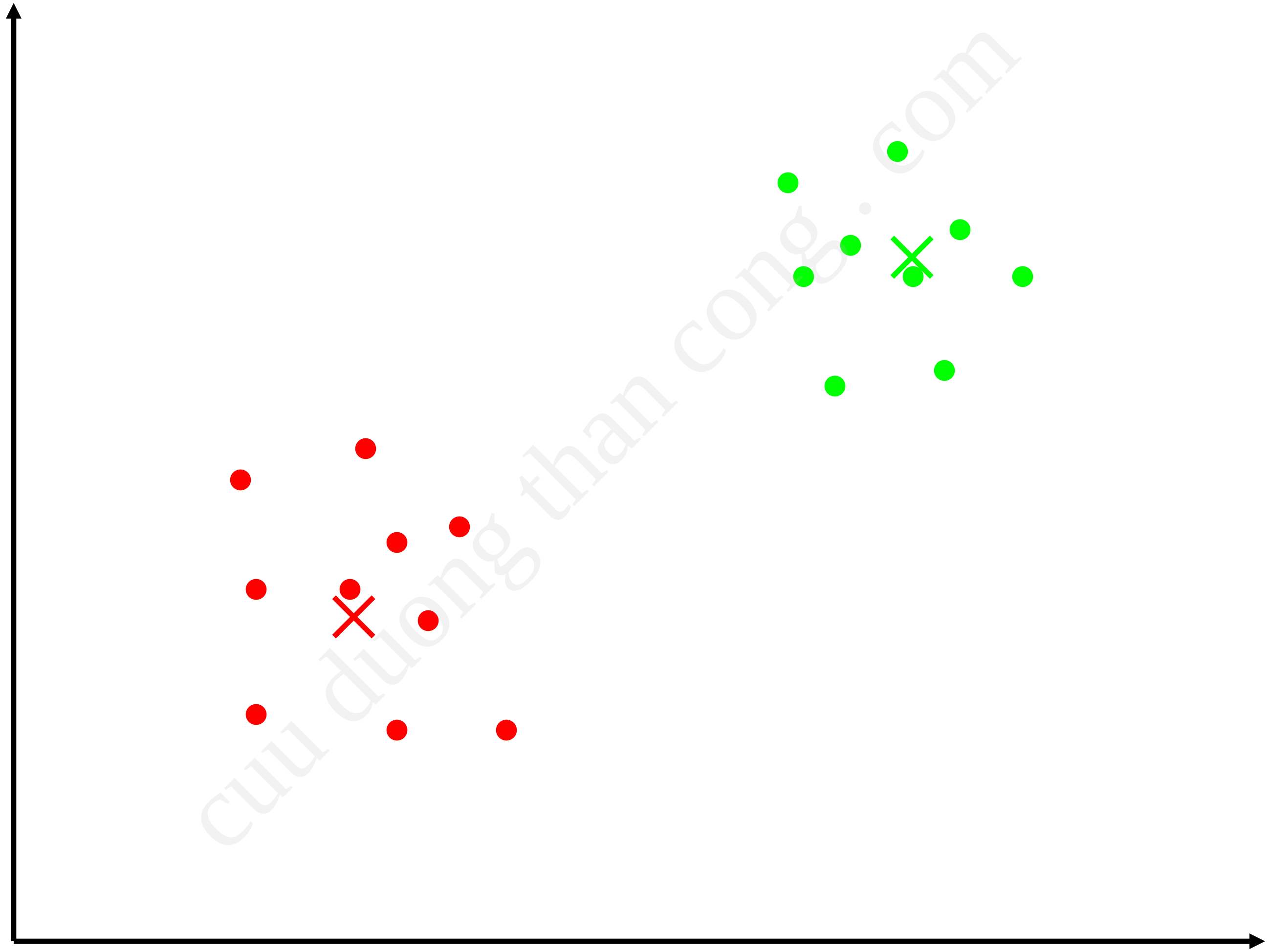
5. IS: Segmentation Using Clustering Method (6)



5. IS: Segmentation Using Clustering Method (7)



5. IS: Segmentation Using Clustering Method (8)



5. IS: Segmentation Using Clustering Method (9)

Example: Clustering