CH10

Image Segmentation

Basics of image segmentation

- Input: image => Output: attributes
- Segmentation of nontrivial image is one of the most difficult tasks in image processing

Ex. Separate the sky, clouds, and mountains

Two approaches

- Two basic properties for monochrome image segmentation:
  - **discontinuity**- partition an image based on abrupt changes in gray level, the principal areas are detection of isolated points, detection of lines and edges in an image.
  - **similarity**- thresholding, region growing, and region spliting and merging.

Detection of discontinuity

- Three basic types of gray-level discontinuity
  - Points, lines, and edges
- Method: run a mask through the image

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

3x3 mask

Input: 3x3 region $z_1, z_2, z_3, \ldots, z_9$

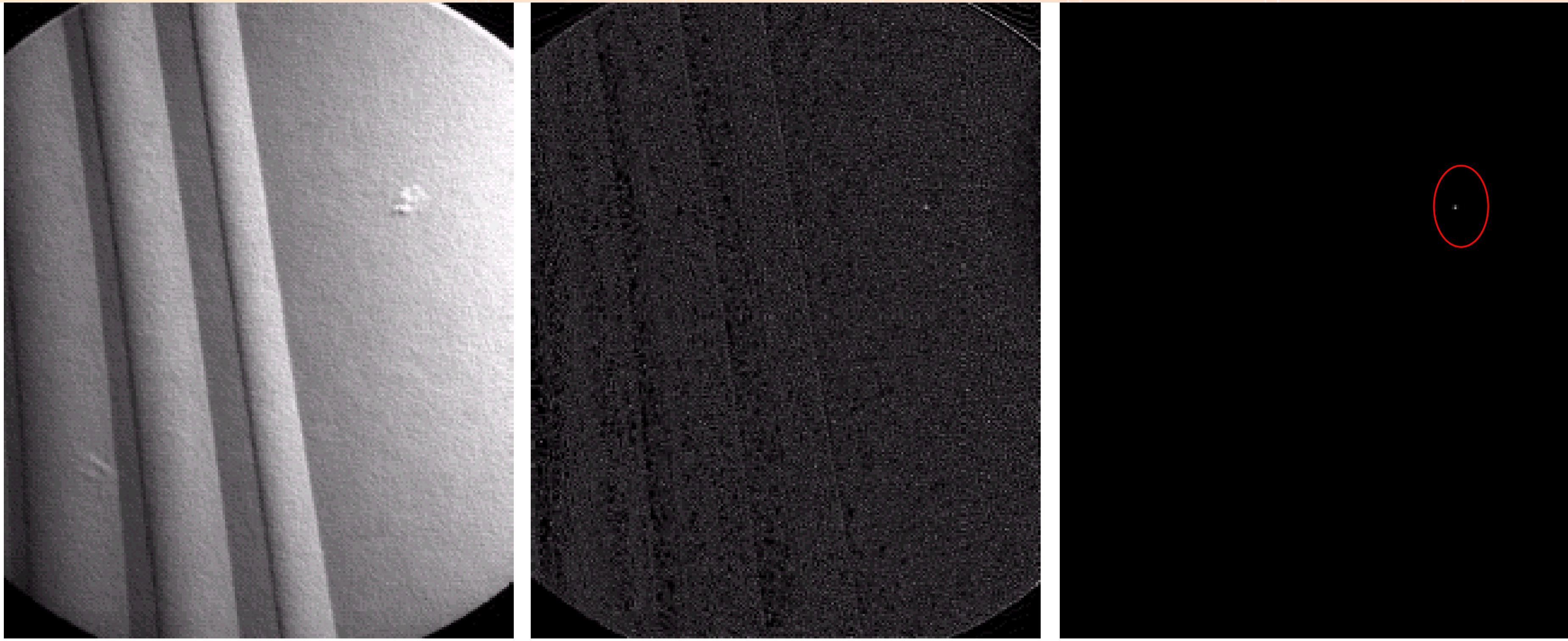
Output: z_5 -> R

$$R = \sum_{i=1}^{9} w_i z_i$$

Point detection

- Point detection mask
  1. Mask operation over whole image
  2. Threshold: $|R| \geq T$
     - The idea is that the gray level of an isolated point will be quite different from the gray level of its neighbors.

-1	-1	-1
-1	8	-1
-1	-1	-1



渦輪葉片 X 光 → point mask → thresholding

Line detection

1. Line detection 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°

2. Which orientation?

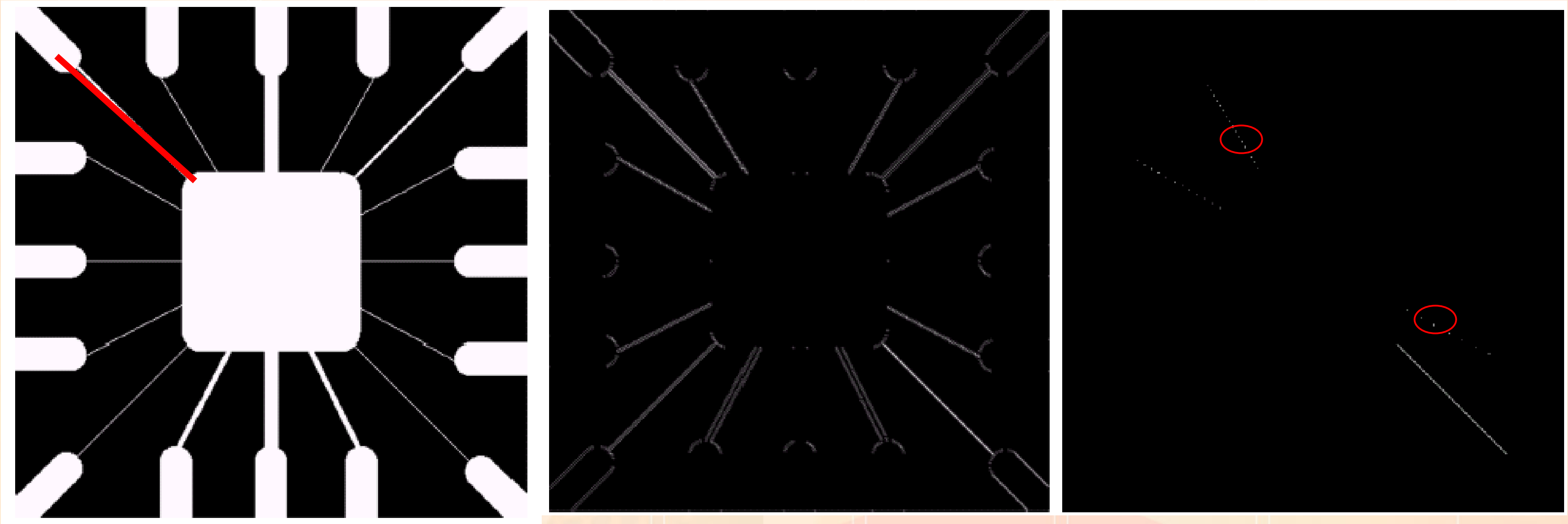
A thin line:

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{matrix} \rightarrow$$

Hori.	45°	Veri.	-45°
$R_1=8$	$R_2=2$	$R_3=2$	$R_4=2$

Thresholding: $|R_i| \geq |R_j| \geq T$

Example: single-line detection



Input: binary image

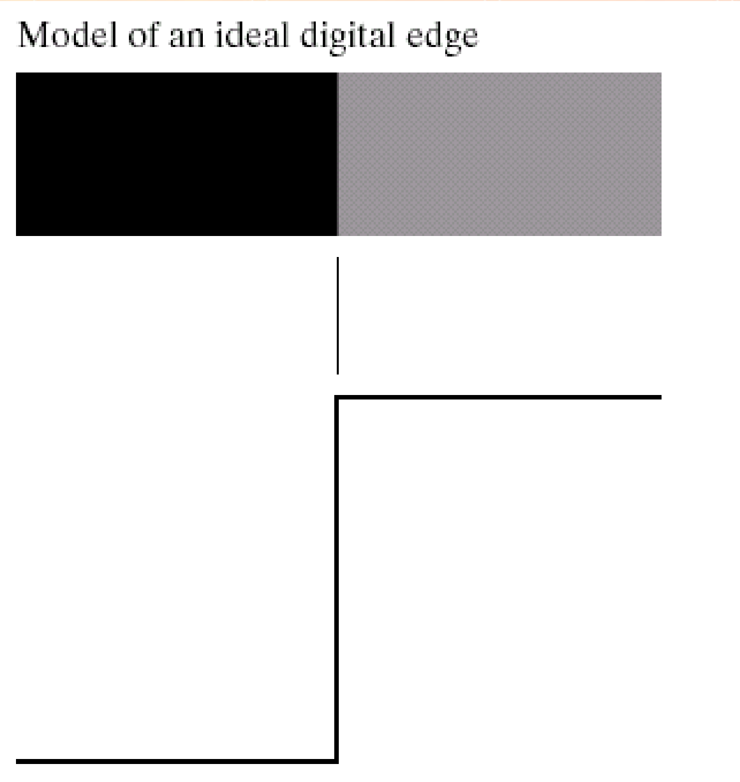
absolute value after -45^{o} mask

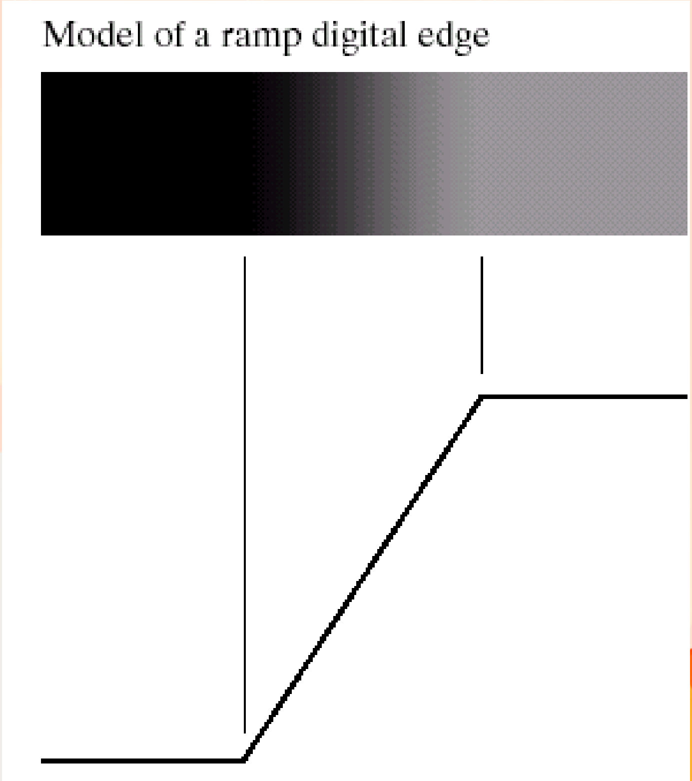
$|R| \geq T$

1. Detect line that are one pixel thick
2. Isolated "line points"

Definition of edges?

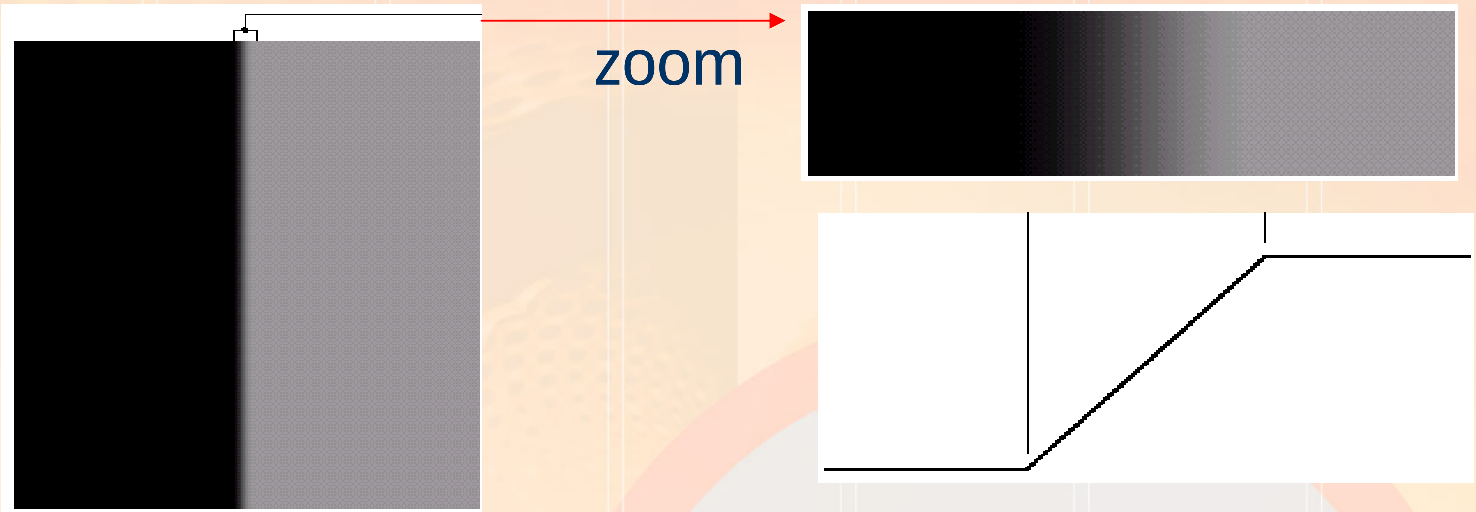
- Intuition of edge: set of connected pixels that line one the boundary between two regions
- Model of digital edge





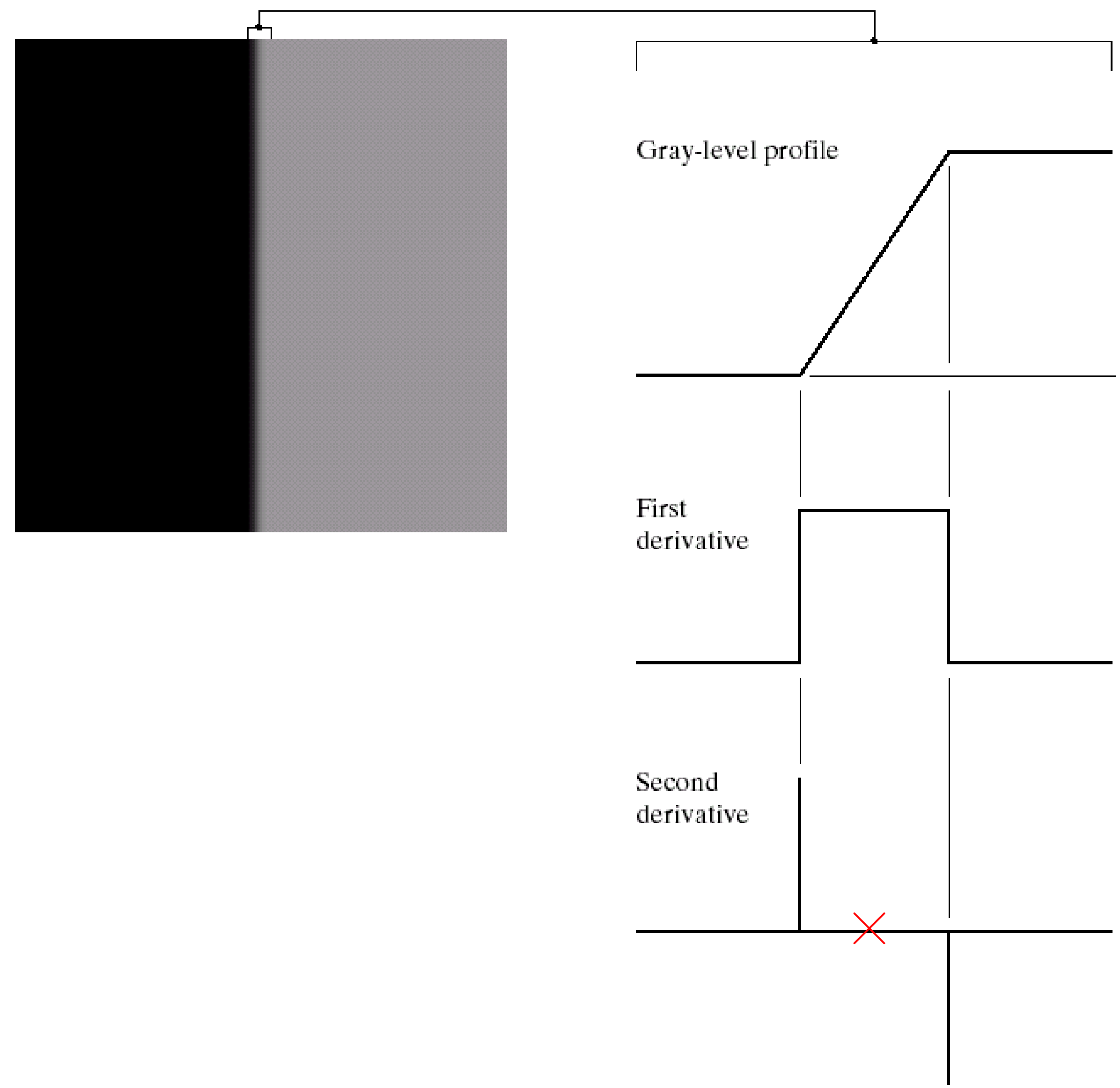
Two factors affects acquired edges

- Blur 模糊化
  - Sampling rate, illumination condition



- Noise 雜訊

Method of edge detection: derivatives



Edge detection method

Magnitude of 1^{st} derivatives $\geq T$

1. Sign determines Dark or light region

2. Zero-crossing

Methods: 1st derivatives

- 1st derivatives

2-D gradient

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

Magnitude = $\left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right]^{0.5}$

$$\approx \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

Direction = $\tan^{-1}\left(\dfrac{\partial f / \partial y}{\partial f / \partial x}\right)$

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

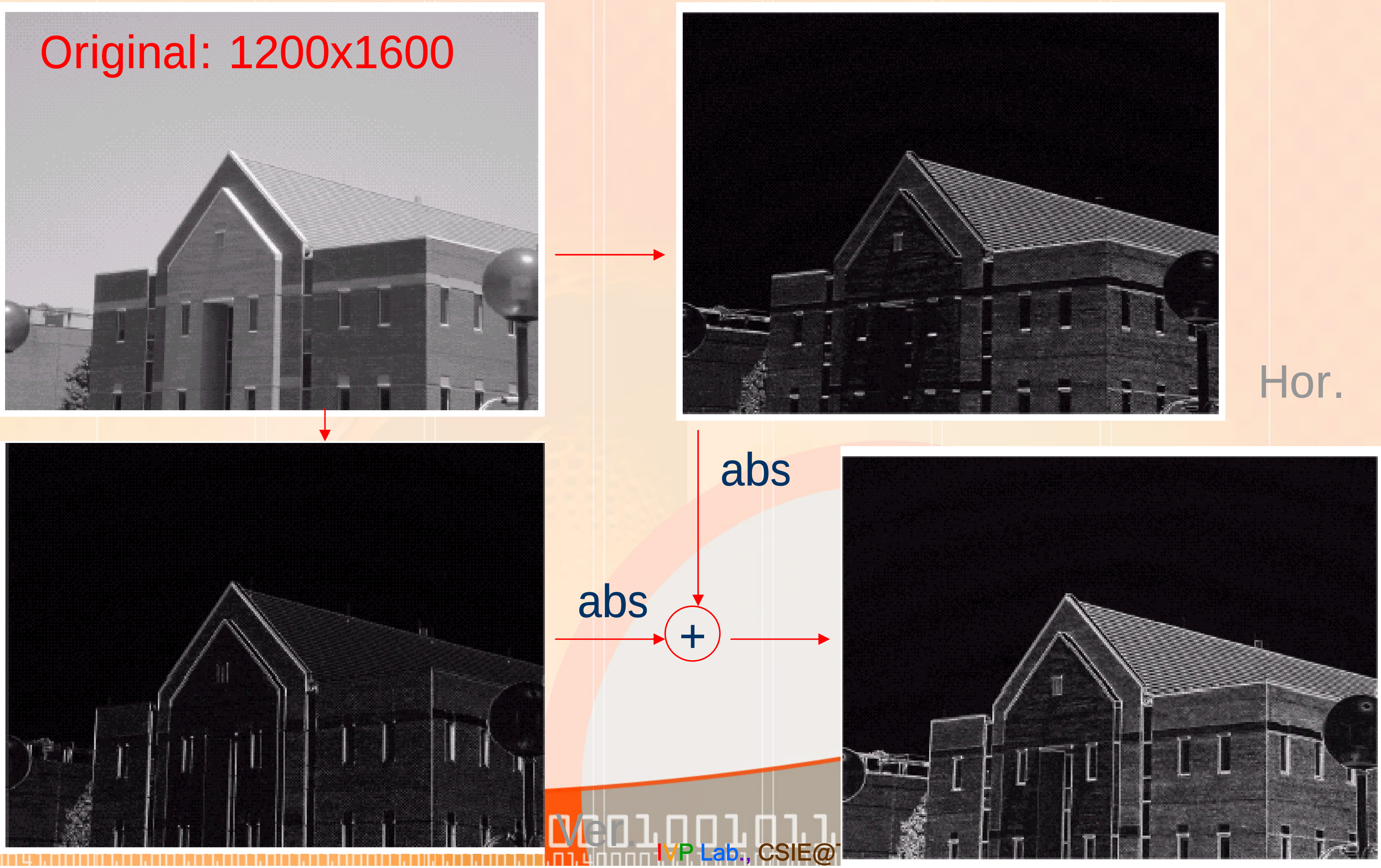
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

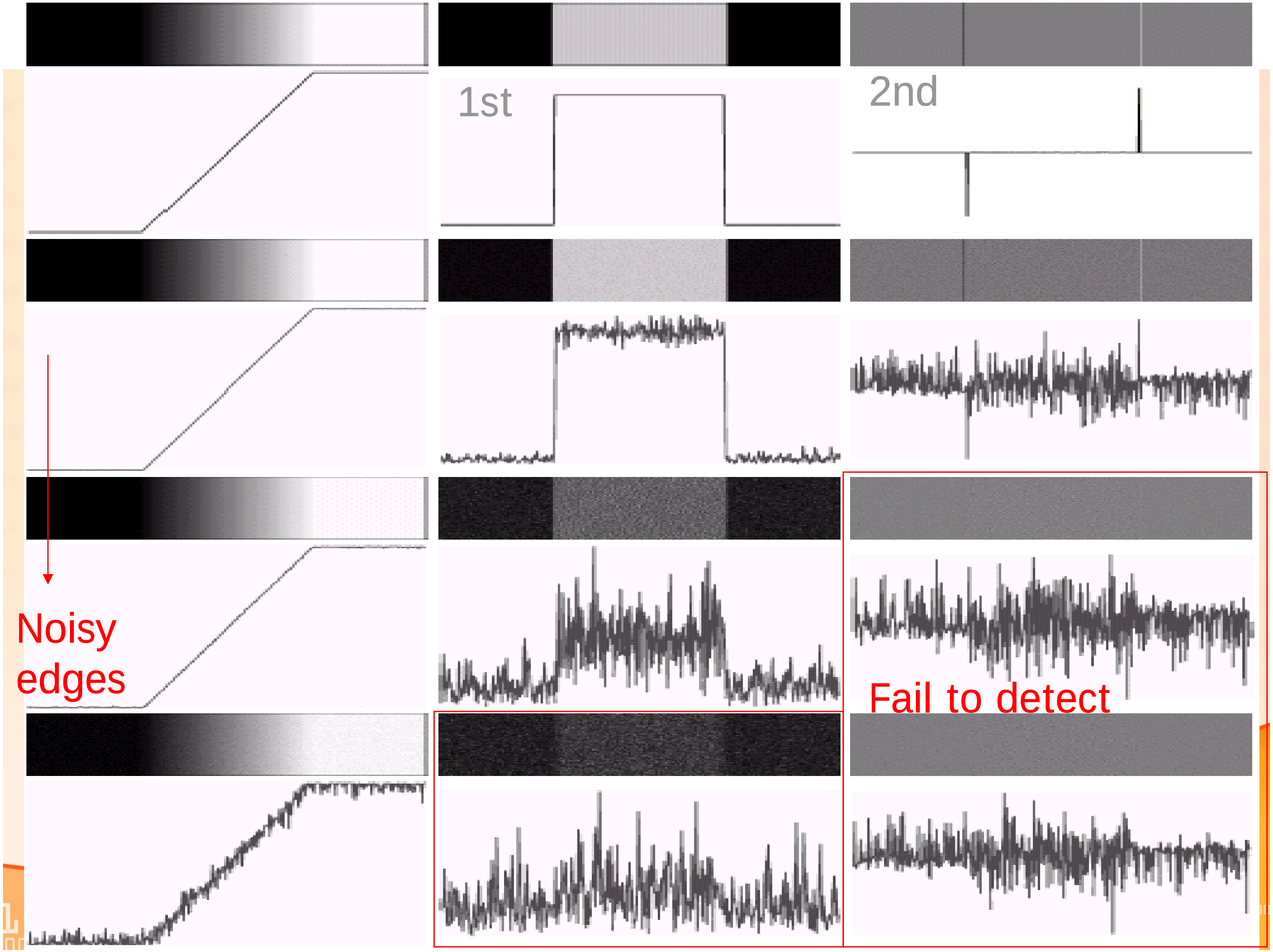
$\dfrac{\partial f}{\partial x}$ $\dfrac{\partial f}{\partial y}$

Example: Sobel filter

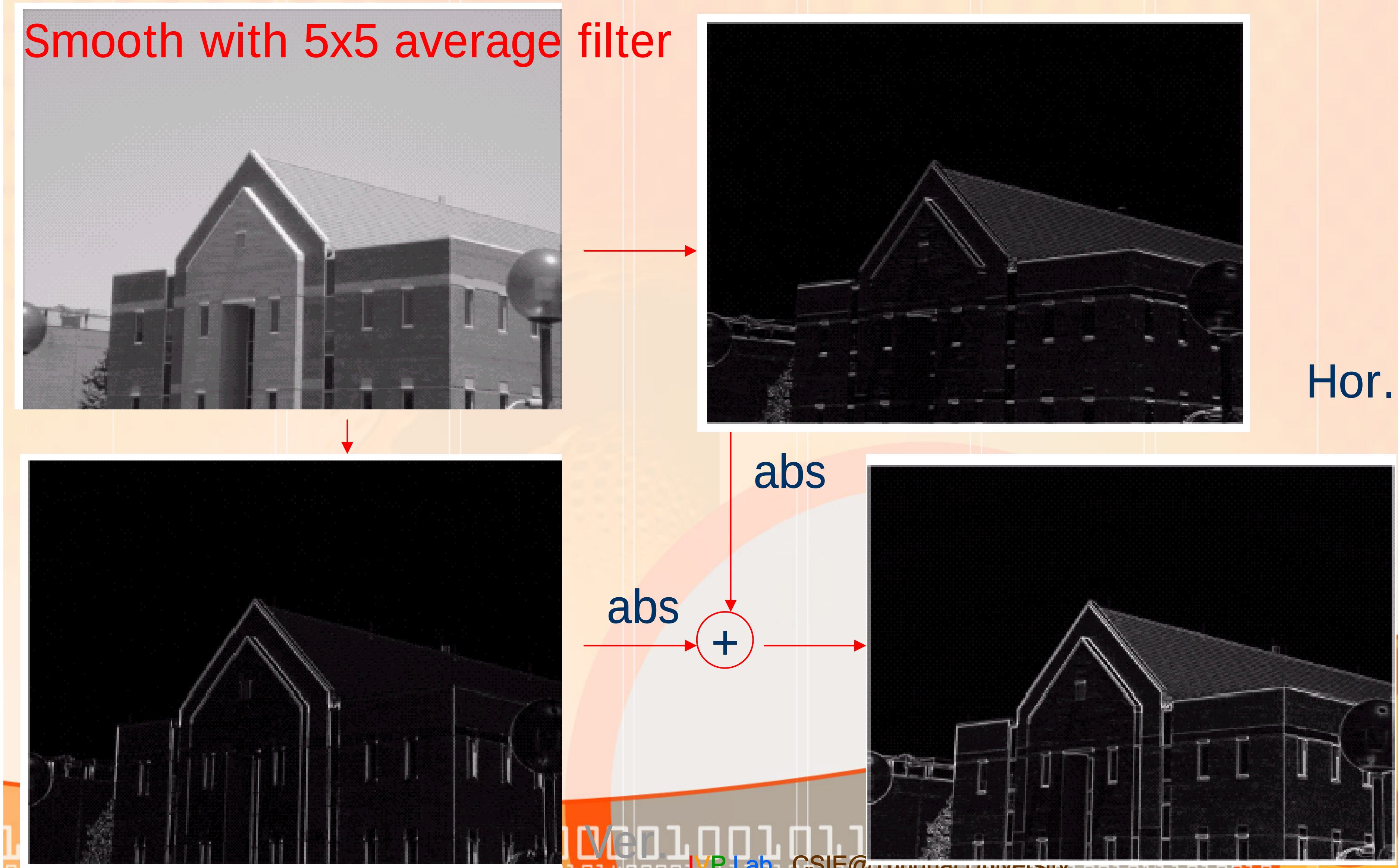


Effects of noise

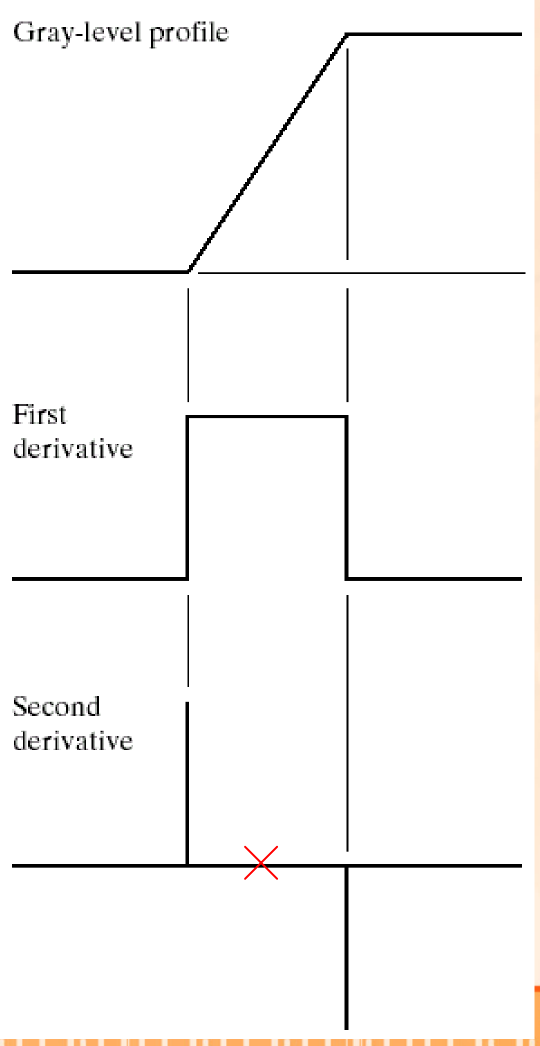
Unwanted
Detailed
Structure !


1st
2nd
Noisy
edges
Fail to detect

Reduce noise: Smoothing



Method: 2nd derivative



Edge detection method

Magnitude of 1st derivatives $\geq$ T

1. Sign determines Dark or light region

2. Zero-crossing

2nd derivative mask

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Edge Detection (cont.)

Laplacian-

- The Laplacian of a 2-D function *f*(*x*, *y*) is a second derivative defined as

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

- The spatial mask for Laplacian is

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8).$$

0	-1	0
-1	4	-1
0	-1	0

Edge Detection (cont.)

- The Laplacian is unsuitable for edge detection because
  - sensitive to noise
  - produce double edges
  - unable to detect edge direction
- The Laplacian is usually used to detect whether a pixel is on the dark or light side of an image.
- A more general use of Laplacian is in *finding the location of edges* using its zero-crossing property by convolving an image with the Laplacian of a 2-D gaussian function

$$h(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- where σ is the standard deviation.

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

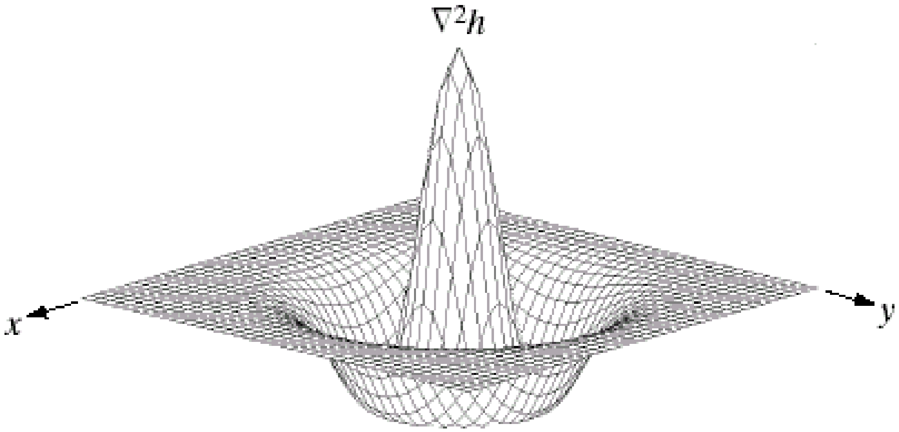
−1	−1	−1
−1	8	−1
−1	−1	−1

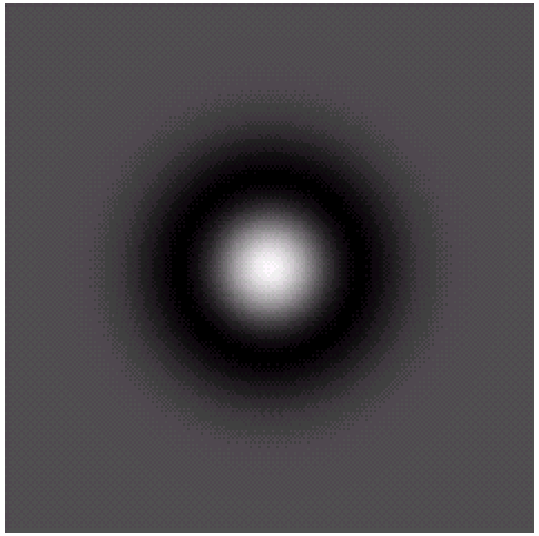
Edge Detection (cont.)

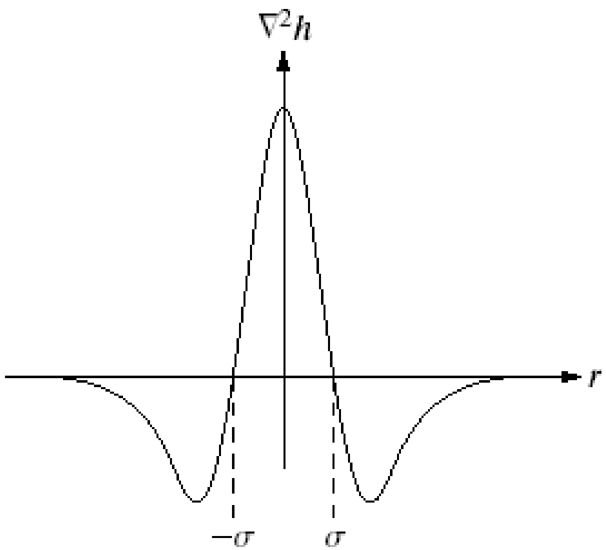
- Let $r^2 = x^2 + y^2$, the Laplacian of h is

$$\nabla^2 h = \left(\frac{r^2 - \sigma^2}{\sigma^2}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

- See Fig. 10.14 for the shape of a Laplacian function (classical Mexican hat shape).
- See Fig. 10.15 for illustartion.
- Note edge detection by gradient operations work well in cases involving images with <u>sharp intensity transition</u> and <u>relatively low noise</u>; while zero crossings offer an alternative in cases when <u>edges are blurry</u> or when a <u>high noise</u> content is present.







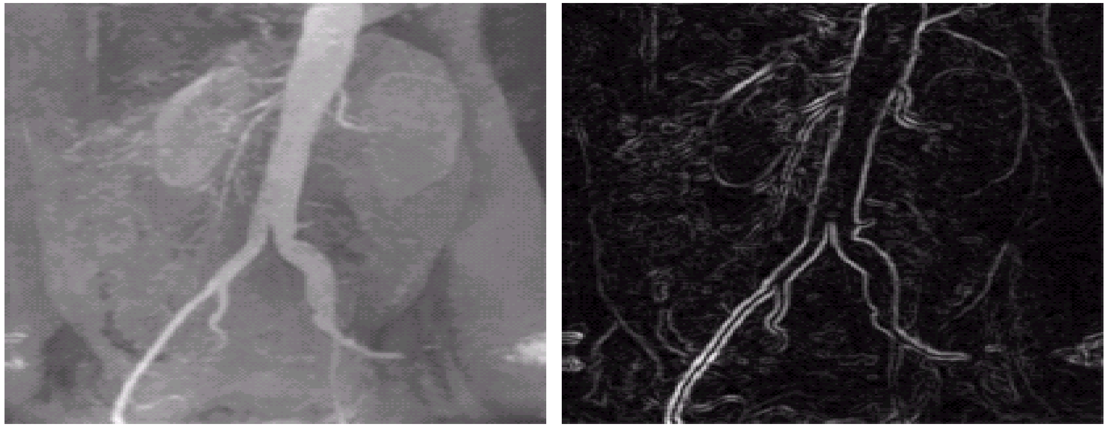
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

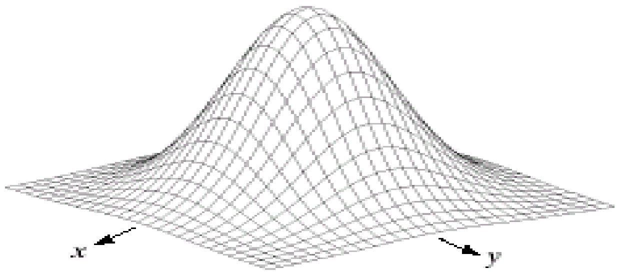
a	b
c	d

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).

a b
c d
e f g

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





−1	−1	−1
−1	8	−1
−1	−1	−1

Edge Linking and Boundary Detection

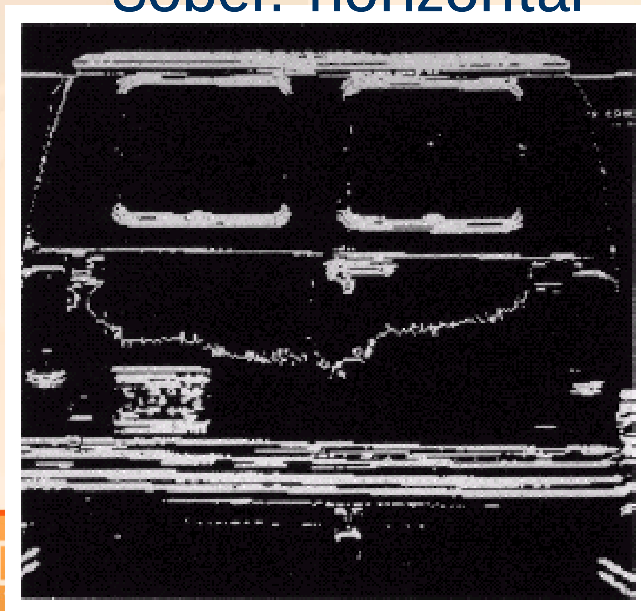
- Objective- to assemble edge pixels, which is yielded by edge detection algorithms and seldom characterize a boundary completely because of noise, breaks in the boundary from nonuniform illumination and other effects that introduce spurious intensity discontinuities, into meaningful boundaries.
- Approaches-
  - local processing
  - global processing via the Hough Transform

Motivation for edge linking

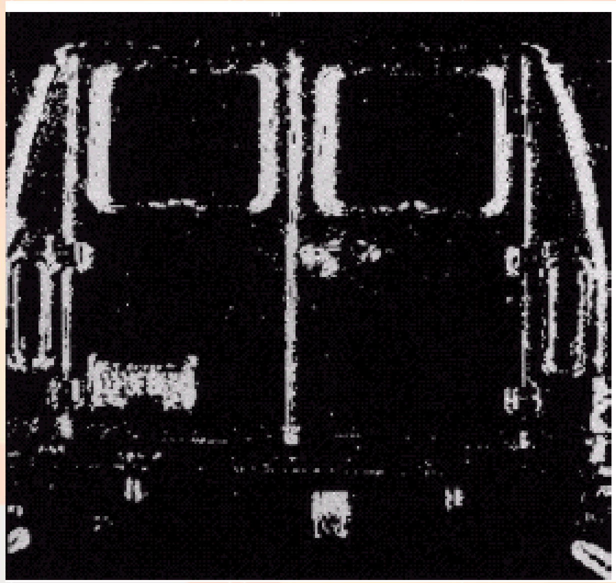
- Edge detection: find pixels lying on edges

=> Link edge points

original



Sobel: horizontal



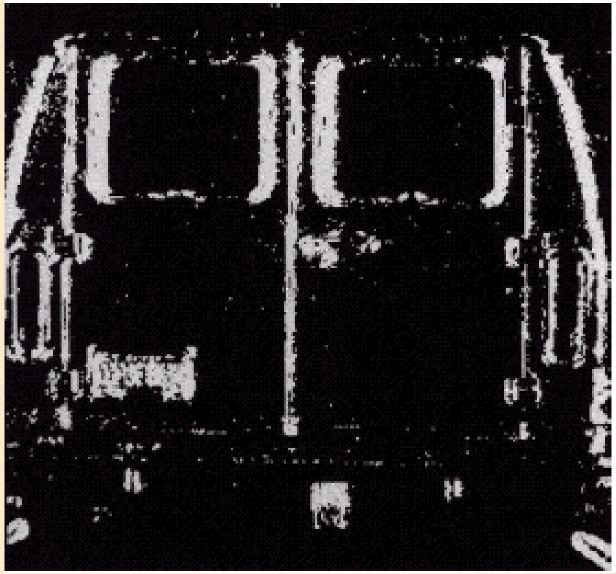
Sobel: vertical

Goal:找出車牌方塊

Edge linking methods

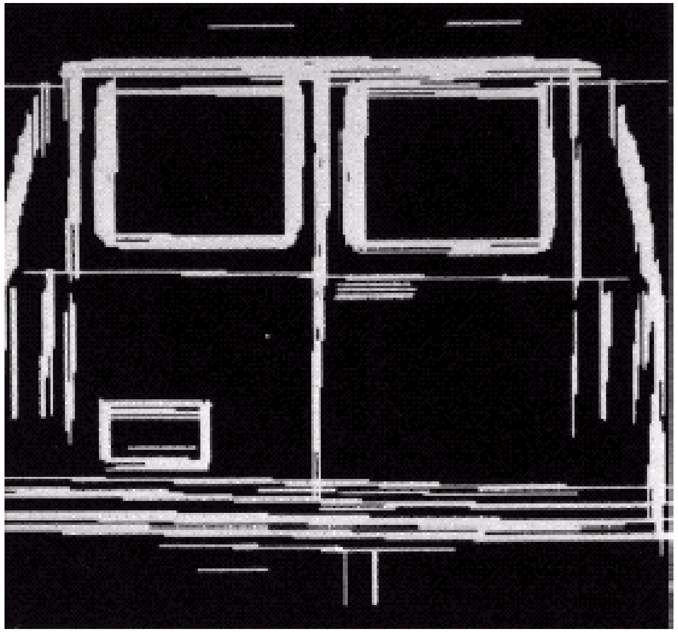
- Local processing
  - Analyze pixels in a small region (3x3 or 5x5)
  - Gradient magnitude: $$\left|\nabla f(x,y) - \nabla f(x_0, y_0)\right| \le E$$
  - Gradient direction: $$\left|\alpha(x,y) - \alpha(x_0, y_0)\right| \le A$$
- Global processing
  - Given n (edge) points in an image
  - Find all possible lines by every pairs of points
  - For each line, find subsets of points that close to it

Example: Local processing



Sobel: vertical

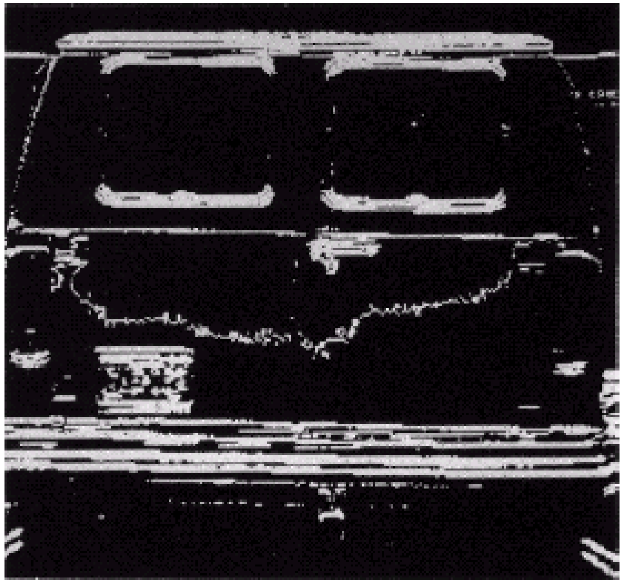
Vertical scan



Criterion:

1. Grad. mag. > 25
2. Grad. Direction difference < 15^o

Horizontal scan



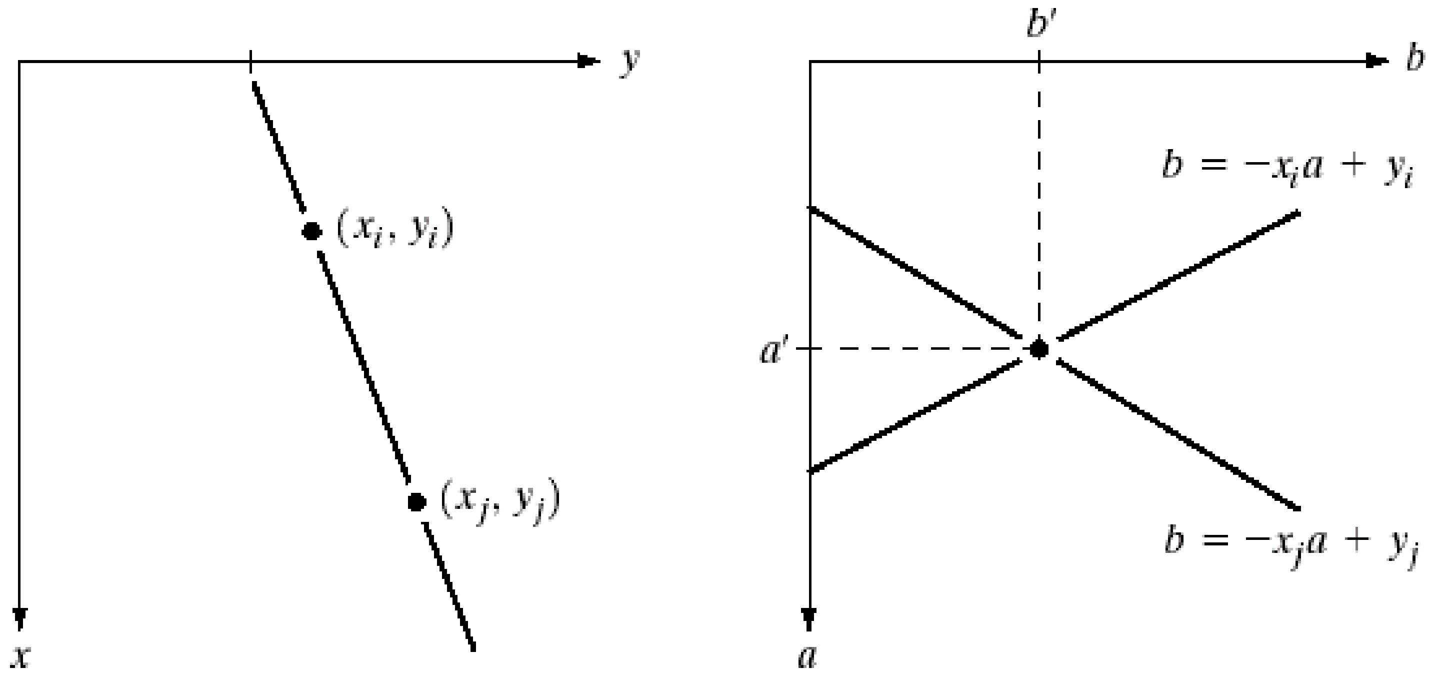
Sobel: horizontal

Global Processing via the Hough Transform

- Useful for detecting and allocating curve of specified shape (lines, circles, and ellipses, etc.).
- Advantages of Hough Transform-
  - immune to noise
  - tolerating gaps
  - applicable to partial shapes
- However, the shape must be analytic, i.e., the shape must be describable using an equation.

7.2.2 Global Processing via the Hough Transform

- Idea: （點共線 ⟶ 線共點）
  - Consider a point (x_i, y_i) and the general equation of a straight line in slope-intercept form, $y_i = ax_i + b$.
  - Rewriting the above equation as $b = -x_i a + y_i$ and considering the *ab* plane (called *parameter space*) yields the equation of a single line for a fixed pair (x_i, y_i).
  - Furthermore, a second point (x_j, y_j) also has a line in parameter space associated with (x_j, y_j) , this line will intersect the line associated with (x_i, y_i) at (a', b'), where a' is the slope and b' the intercept of the line containing (x_i, y_i) and (x_j, y_j) in the *xy* plane.



a b

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

7.2.2 Global Processing via the Hough Transform

- Implementation (*accumulator cells*):
  - The parameter space is subdivided into accumulator cells (see Fig. 10.18), where (a_{min}, a_{max}) and (b_{min}, b_{max}) are the expected ranges of slope and intercept values.
  - For each point (x_k, y_k) in the image plane, let a equal each possible value a_i in the range (a_{min}, a_{max}) and compute its corresponding b_j using the equation $b = -x_k a + y_k$, increment the accumulator cell count $A(i, j)$.
  - A value of M in $A(i, j)$ corresponds to M points in the xy plane lying on the line $y = a_i x + b_j$.
  - If there are n image points and the a axis is subdivided into K values, the method involves nK computations.

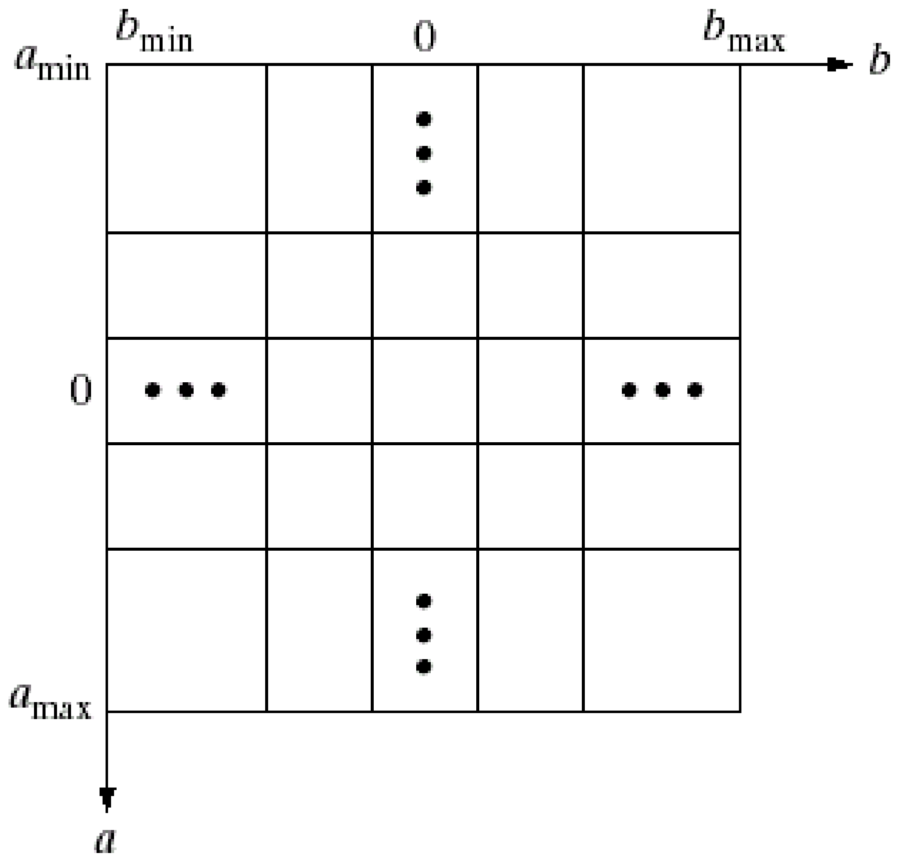


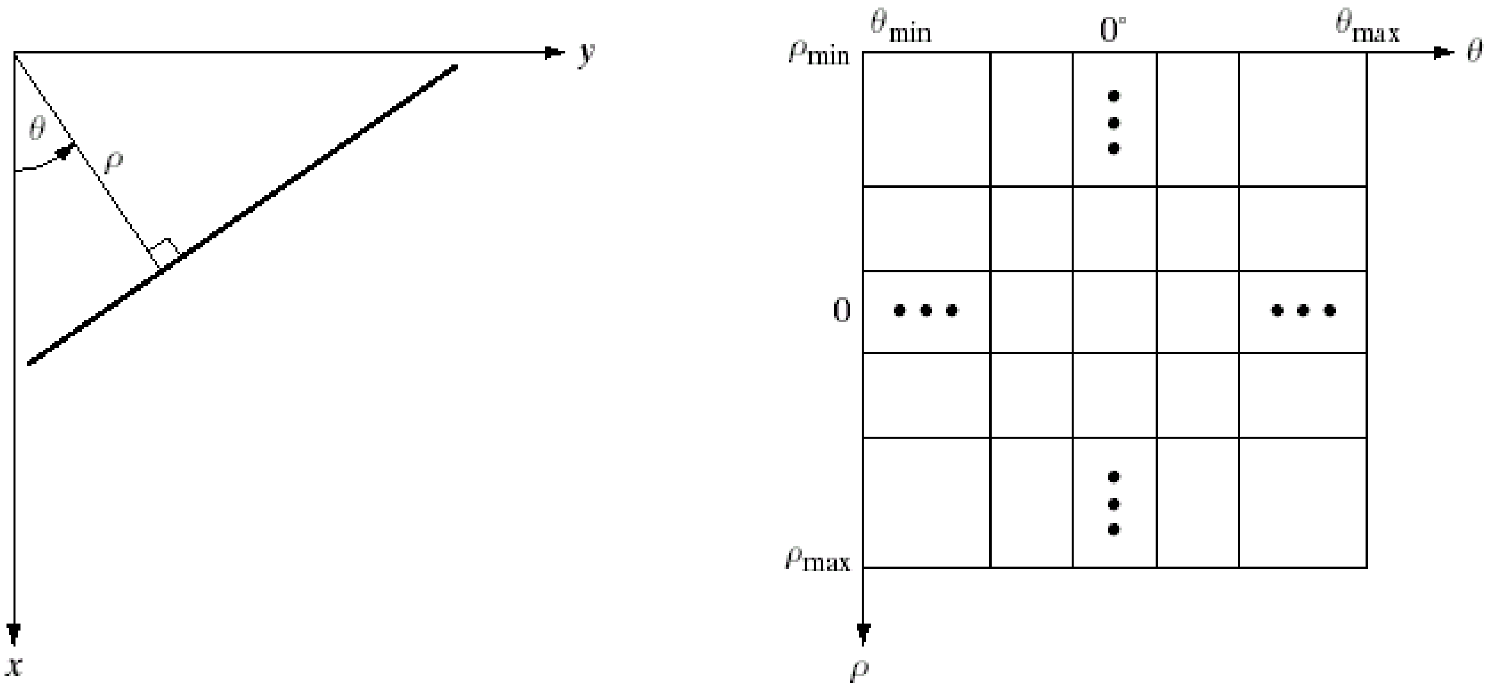
FIGURE 10.18 Subdivision of the parameter plane for use in the Hough transform.

7.2.2 Global Processing via the Hough Transform

- Drawback of the slope-intercept form, e.g., $y = ax + b$, is that both the slope and intercept approach infinity as the line approaches the vertical.
- To solve the above problem is to use normal representation:

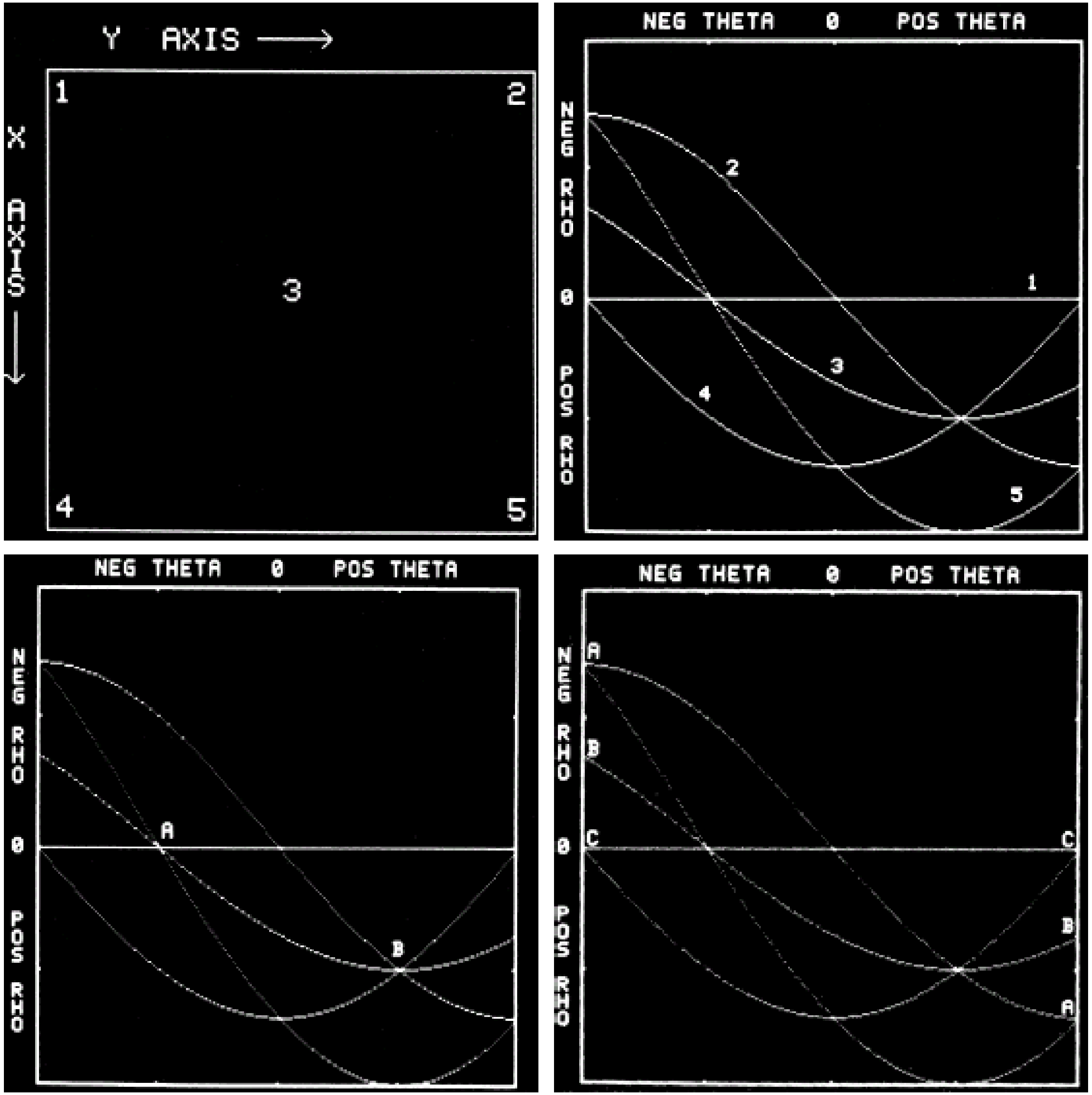
  $$x \cos\theta + y \sin\theta = \rho.$$ (see Fig. 10.19)
- The range of angle θ is [-90º, 90º], measured with respect to x axis.

horizontal line: $\theta = 0^o$ and $\rho = x$ intercept.
vertical line: $\theta = 90^o$ and $\rho =$ positive y intercept
or $\theta = -90^o$ and $\rho =$ negative y intercept



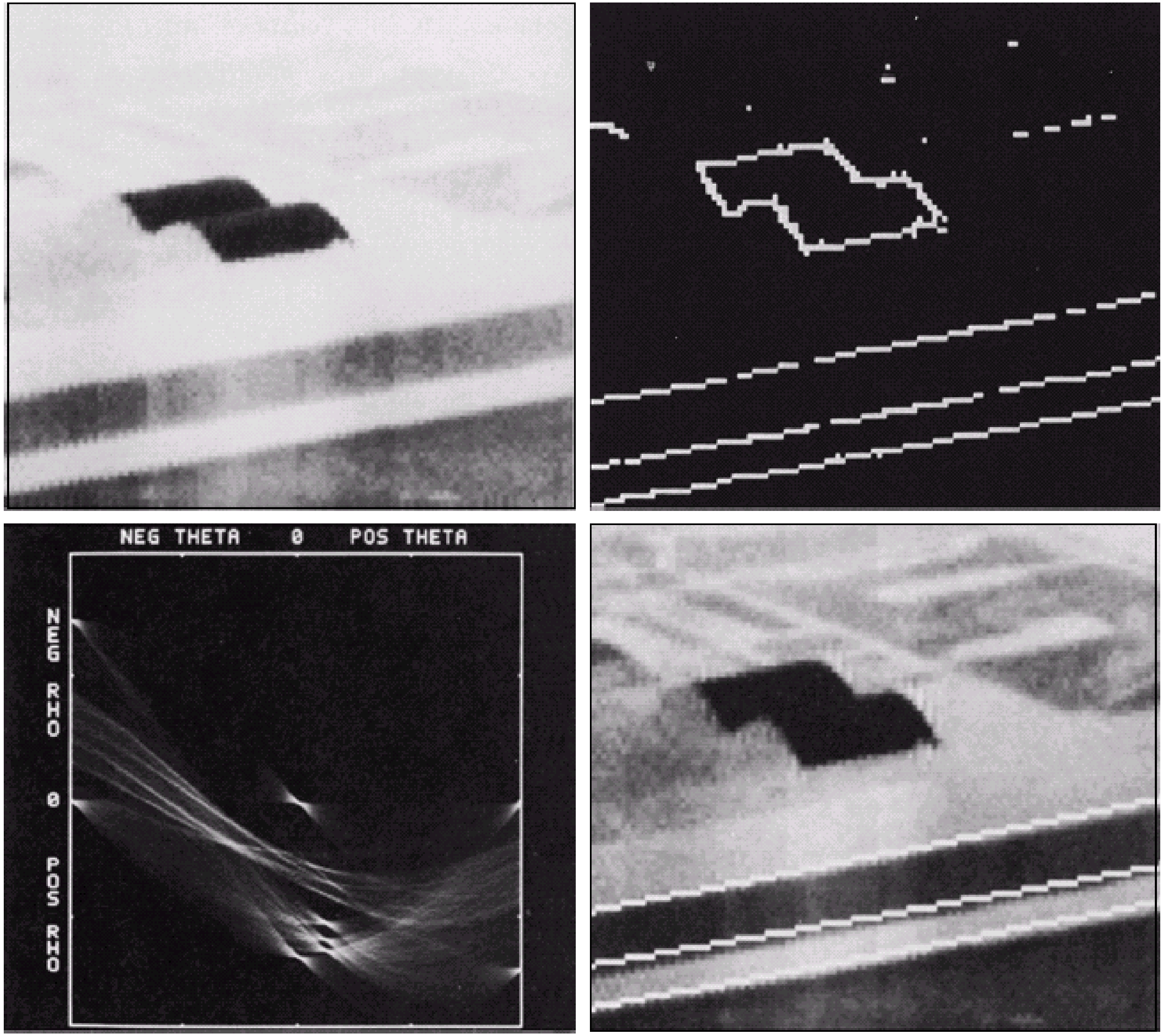
a b

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



a b
c d

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

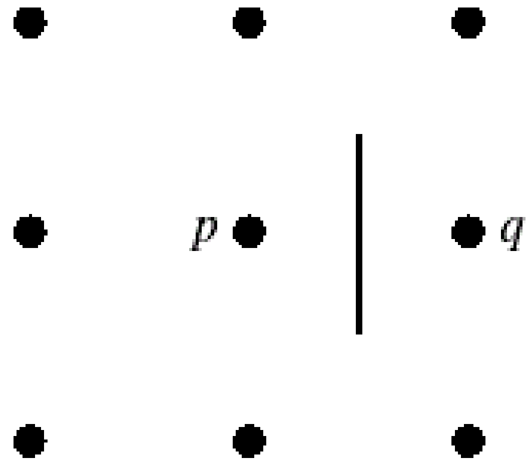


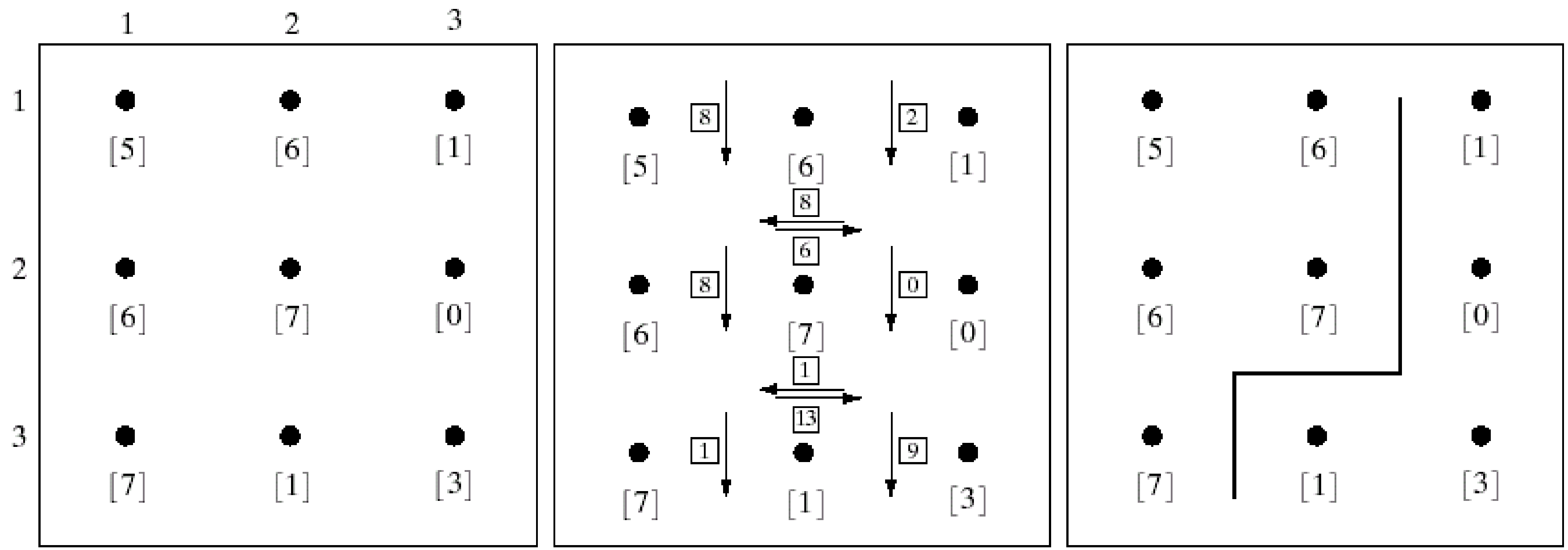
a b
c d

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.)

Graph-Theoretic Technique

FIGURE 10.22
Edge element between pixels p and q.





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.

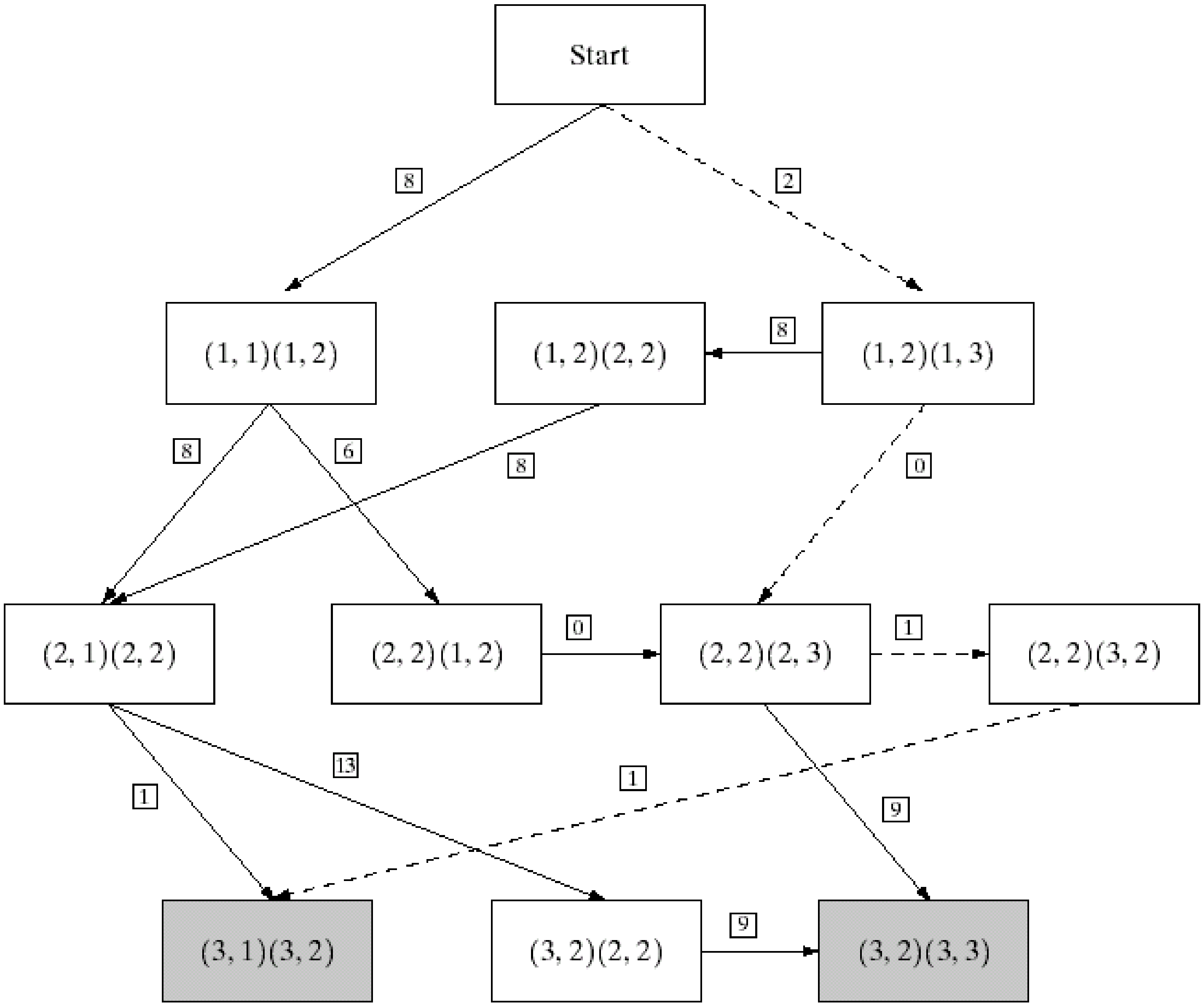


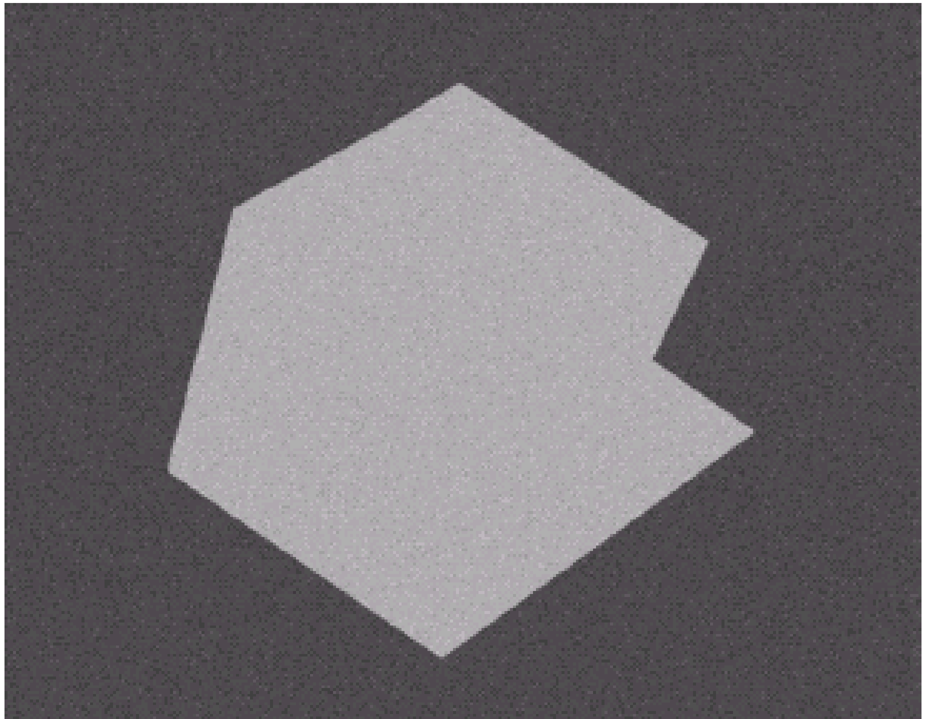
FIGURE 10.24 Graph for the image in Fig. 10.23(a). The lowest-cost path is shown dashed.

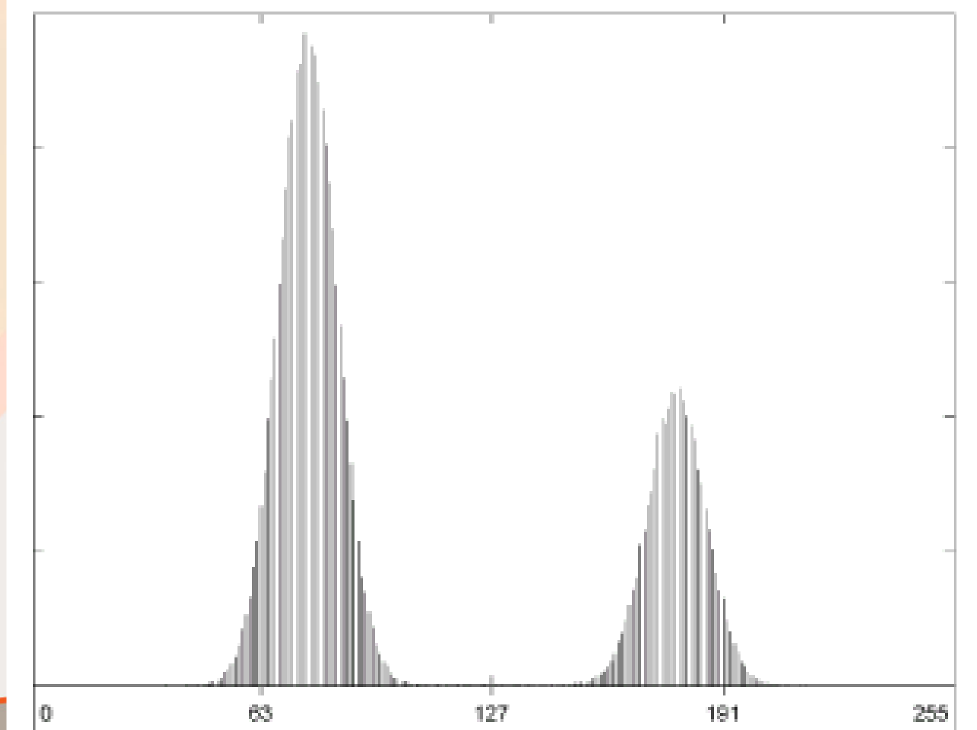
Chapter 10
Image Segmentation

FIGURE 10.25 Image of noisy chromosome silhouette and edge boundary (in white) determined by graph search.

Foundation of thresholding

- Idea: object and background pixels have gray levels grouped into two dominant modes





Original image → histogram

Foundation of thresholding

- Input *f(x,y)*, given threshold *T*

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

Issues of thresholding

- Selection of threshold T ?
- Complex environment – illumination
- Multiple thresholds – more than one object
- Global threshold
- Local threshold

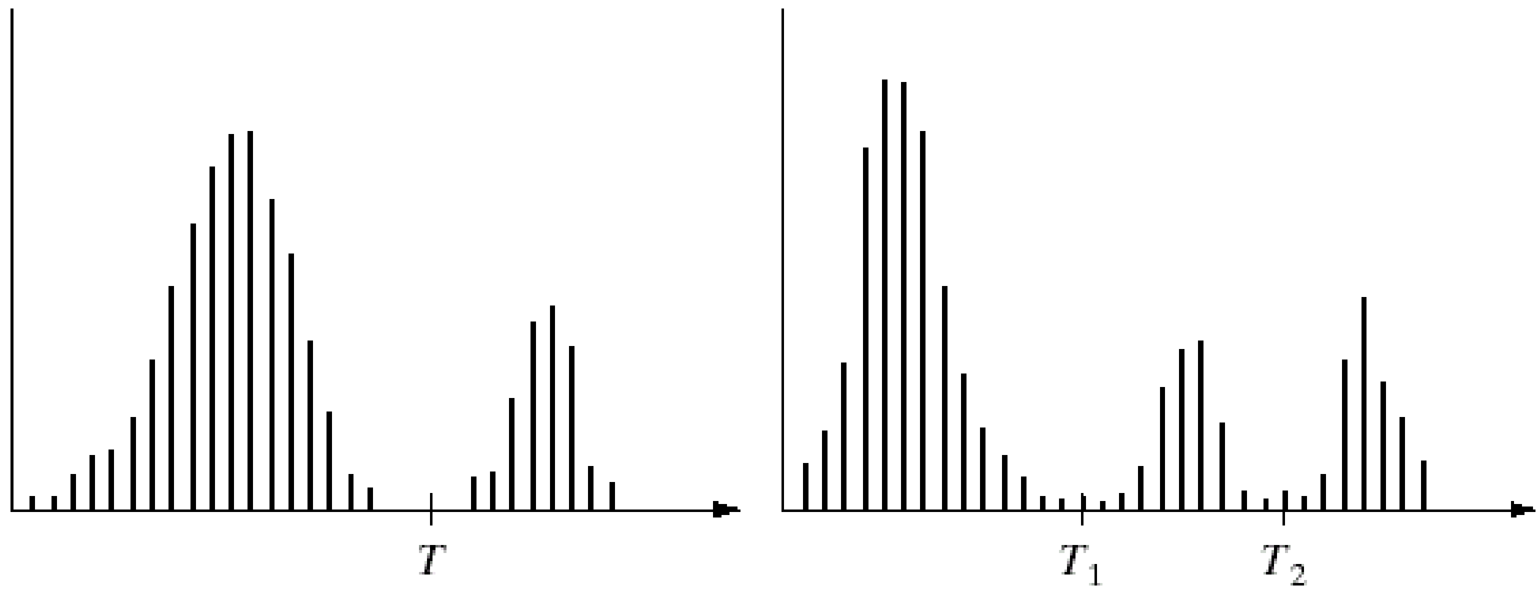
Thresholding as a multi-variable function:

$$g(x,y) = \mathbf{T}[\ f(x,y),\ \underbrace{x,\ y}_{\text{Adaptive}},\ p(x,y)\ ]$$

Adaptive: Depend on position

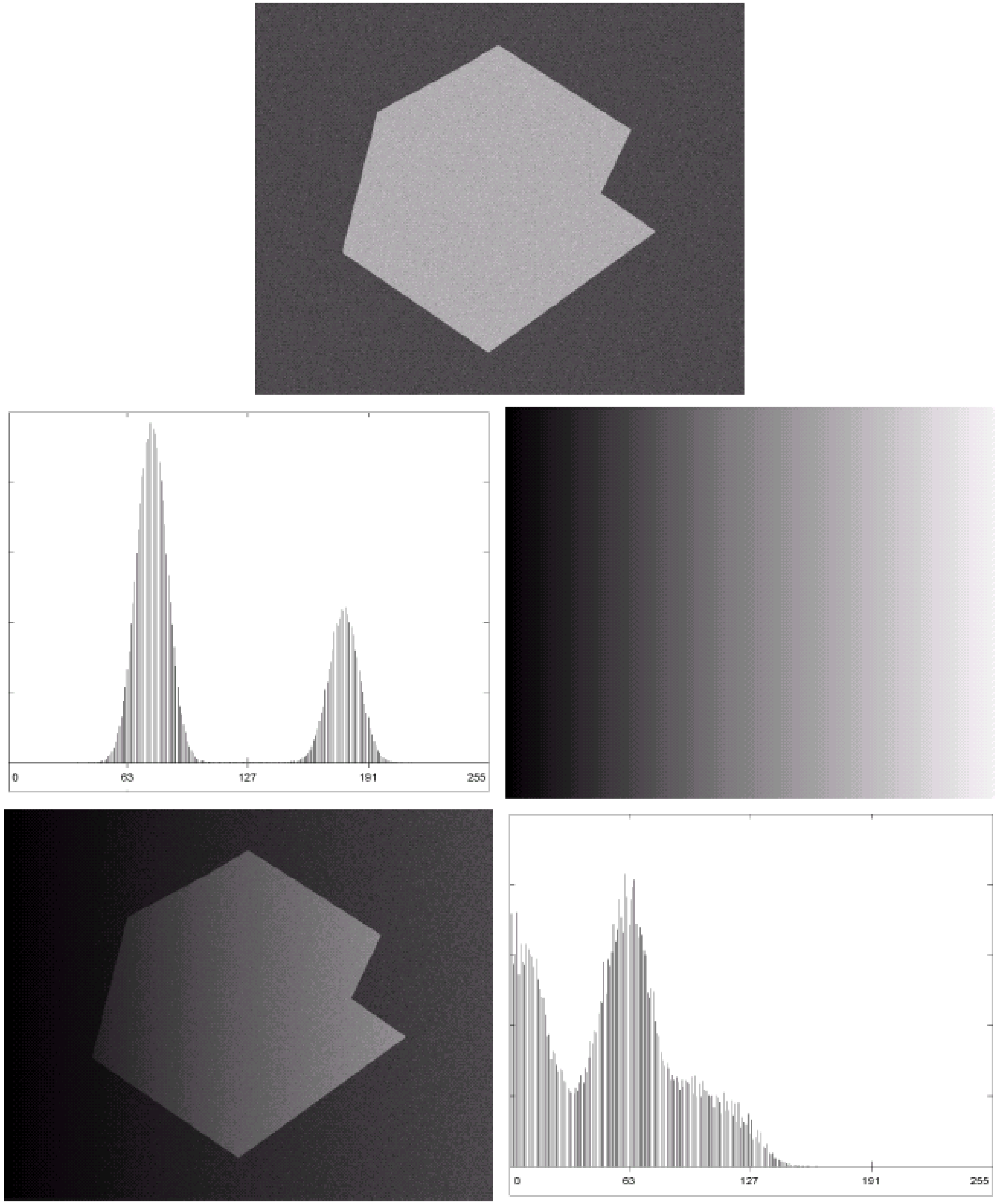
Local: local property func.

Thresholding



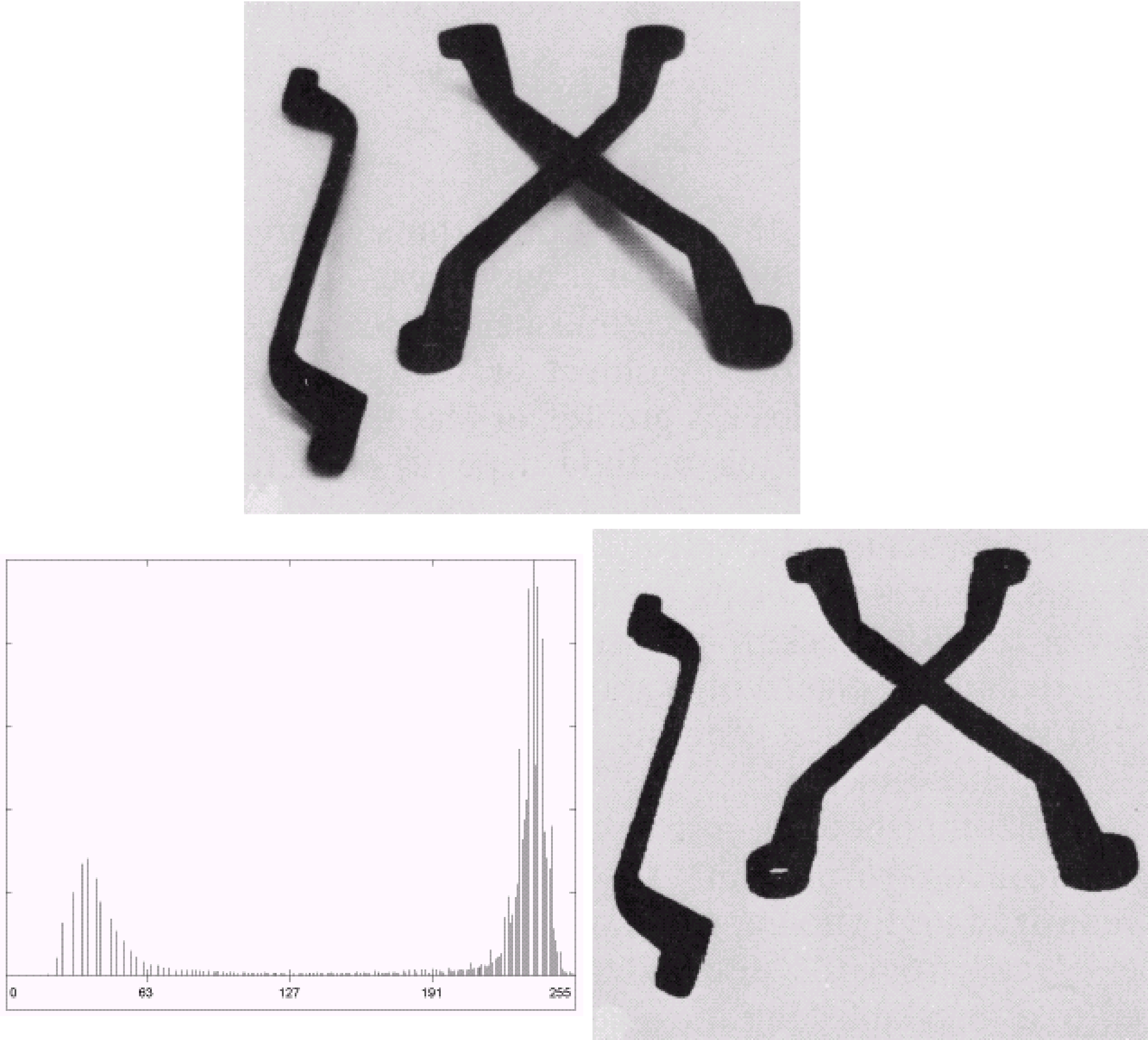
a b

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



a
b c
d e

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

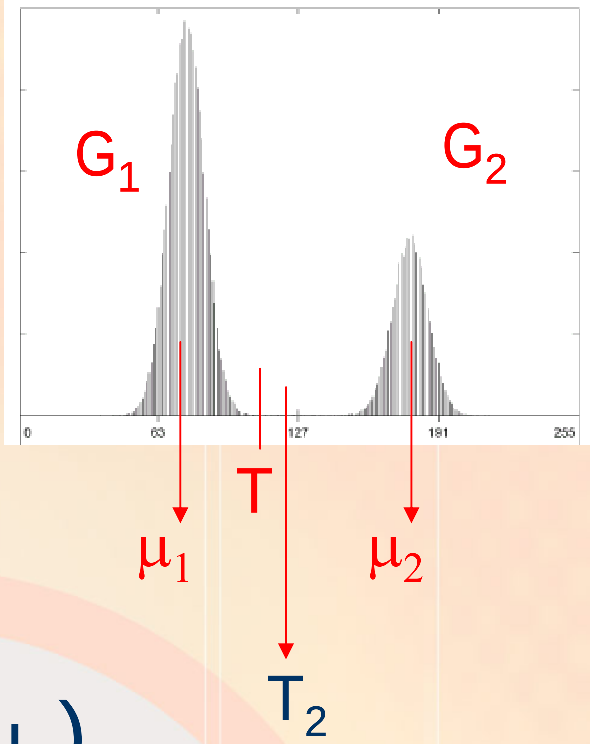


a
b c

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

1. Automatic selection of T

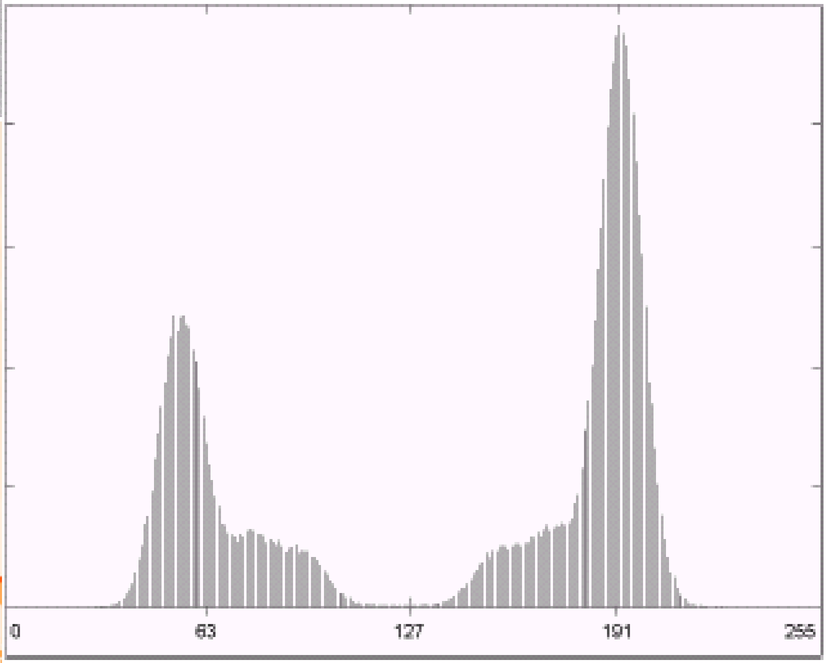
1. Select an initial T
   - Average gray level
   - Mean of max. and min. gray level
2. Segment the image using T
3. Calculate mean of G_1 and G_2
4. New threshold: $T_2 = 0.5(\mu_1 + \mu_2)$
5. Repeat steps 2~4 until difference in successive T is small

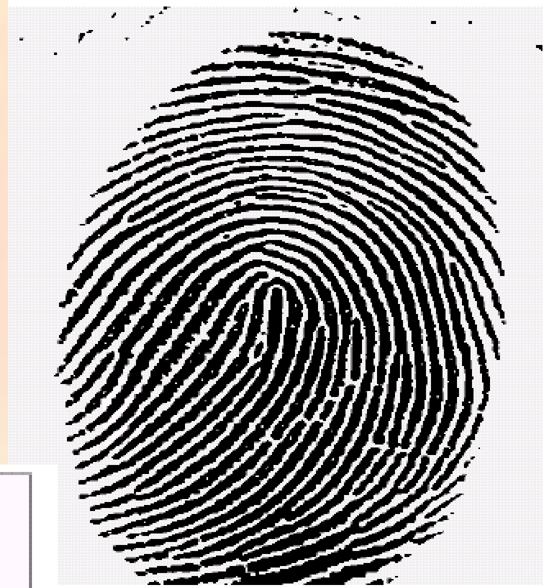


Example: automatically select T

fingerprint

Initial: gray level mean
3 iterations
T = 125.4





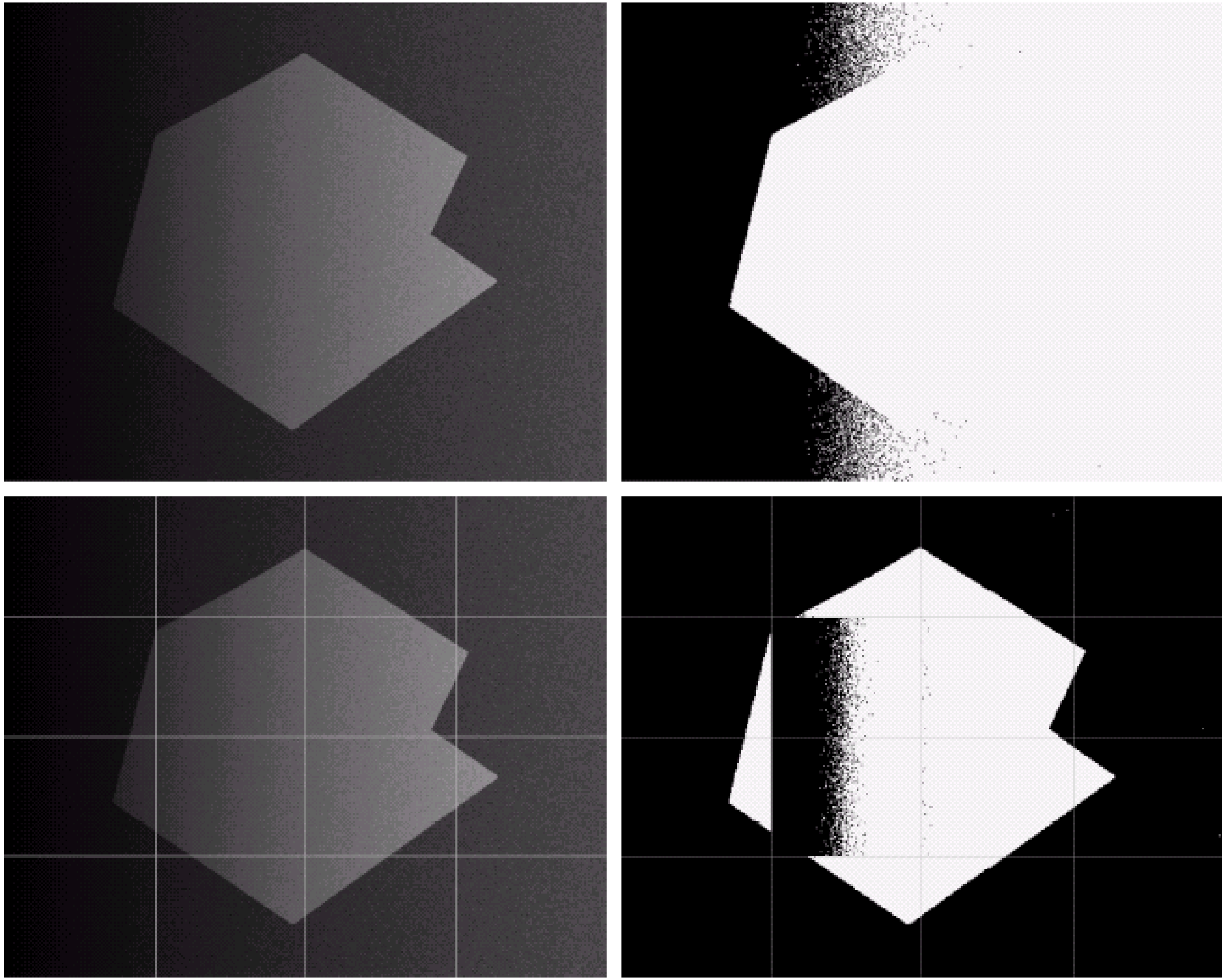


FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

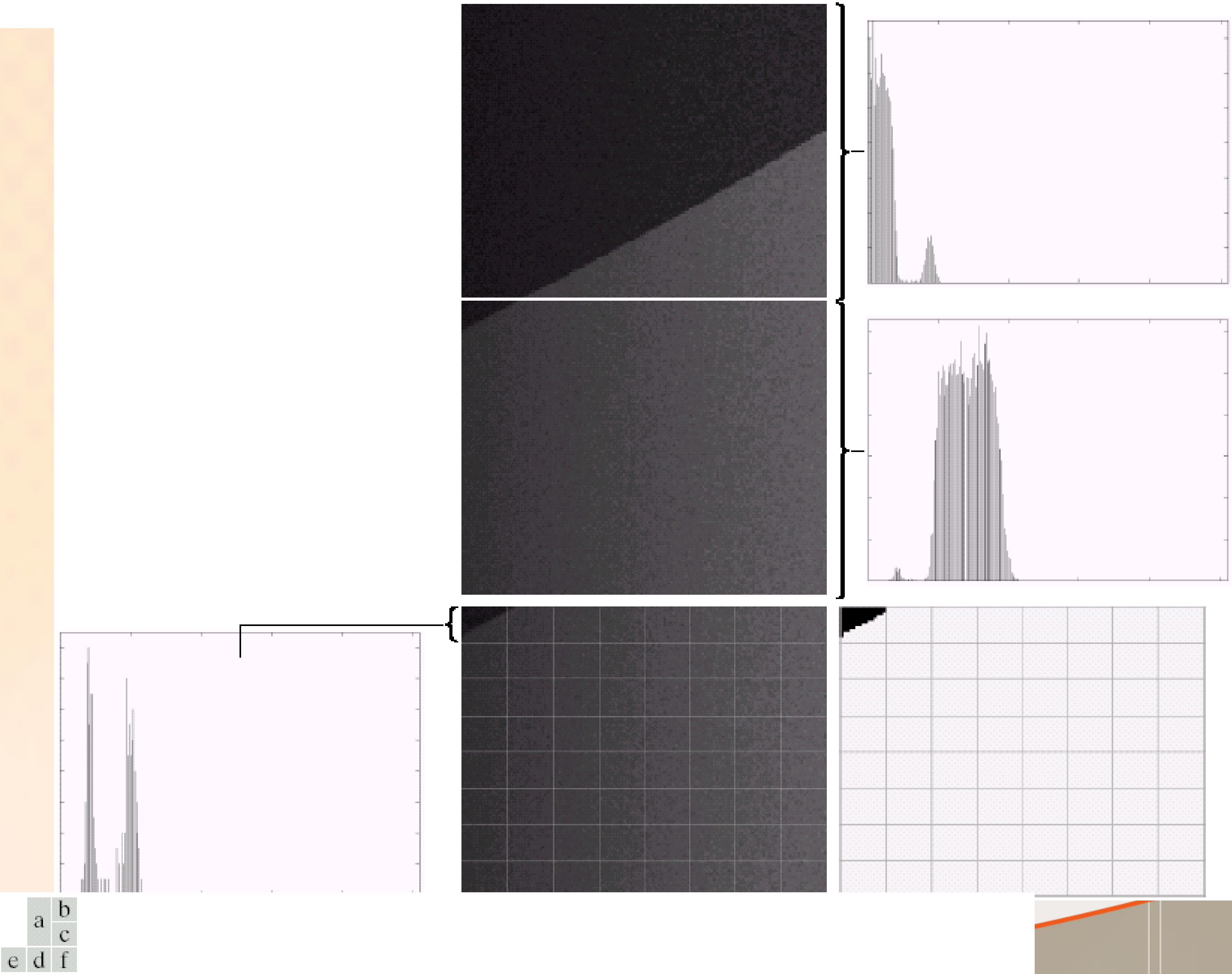
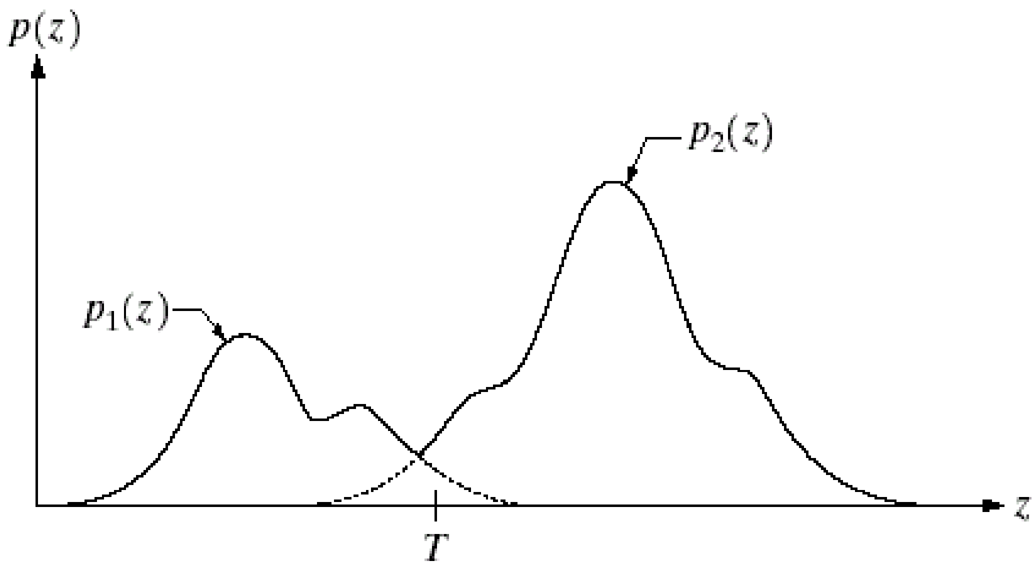


FIGURE 10.31 (a) Properly and improperly segmented subimages from Fig. 10.30. (b)–(c) Corresponding histograms. (d) Further subdivision of the improperly segmented subimage. (e) Histogram of small subimage at top, left. (f) Result of adaptively segmenting (d).

Optimal Thresholding

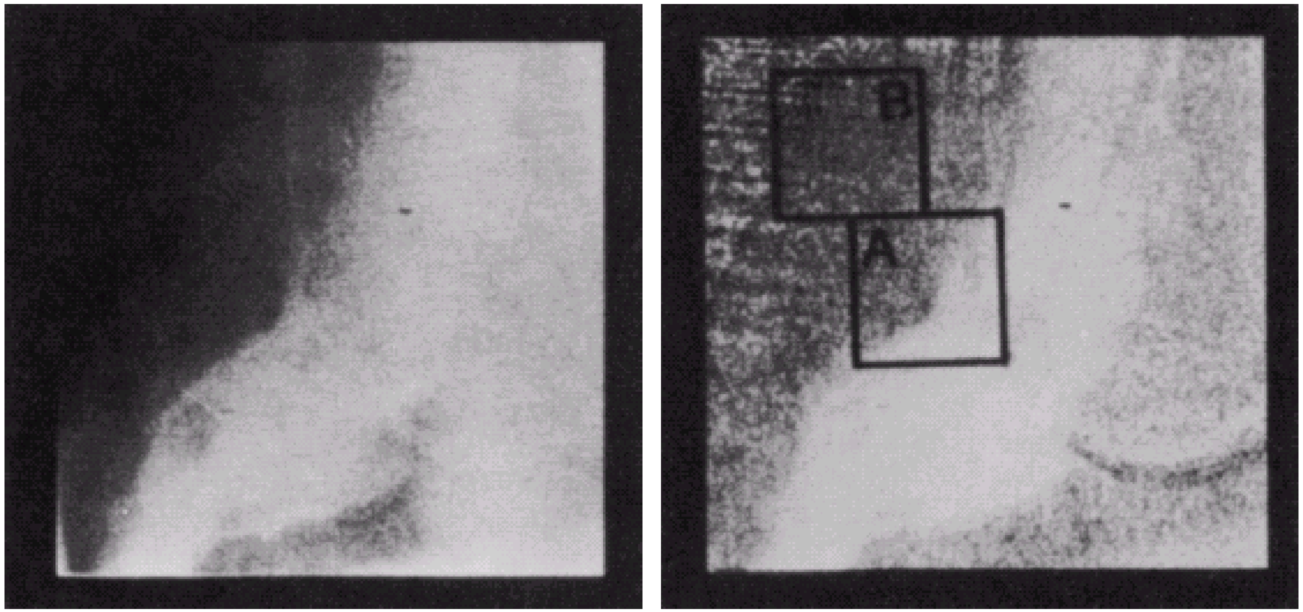
FIGURE 10.32 Gray-level probability density functions of two regions in an image.



Optimal Thresholding Example

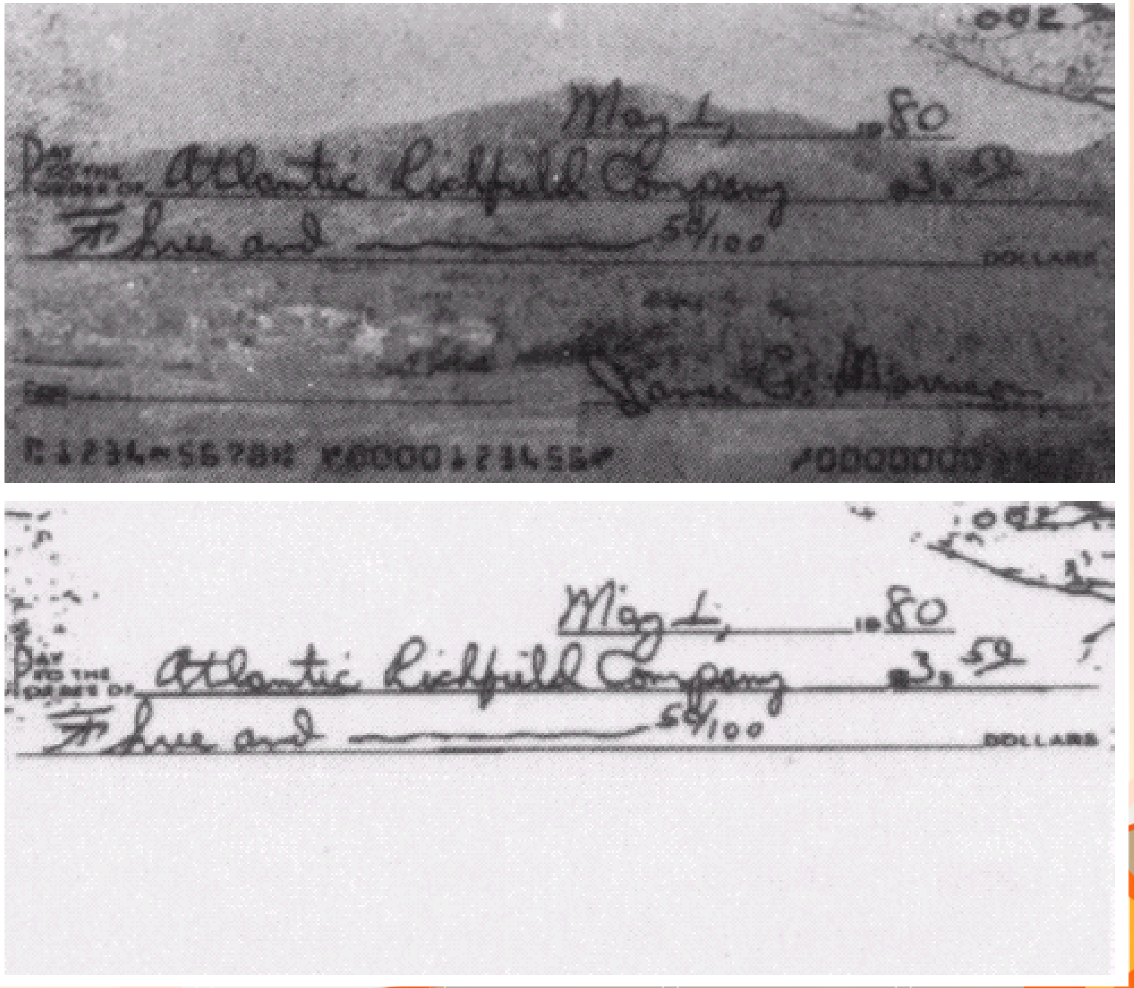
a b

FIGURE 10.33 A cardioangiogram before and after preprocessing. (Chow and Kaneko.)



Boundary Characteristics
Eq. 10.3-6

FIGURE 10.36 Image of a handwritten stroke coded by using Eq. (10.3-16). (Courtesy of IBM Corporation.)



a
b

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

Multispectral Thresholding Example

a b c

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

REGION-ORIENTED SEGMENTATION

- Let R represent the entire image region
- Partitions R into n subregions, R_1, R_2, ..., R_n, such that

1. $\bigcup_{i=1}^{n} R_i = R$
2. R_i is a connected region, i = 1, 2, ..., n
3. $R_i \bigcap R_j = \phi \quad for \quad all \;\; i \;\; and \;\; j, i \neq j$
4. P(R;) = TRUE for i = 1, 2, ..., n, and
5, $P(R_i \cup R_j) = FALSE \quad for \;\; i \neq j$

where P(R_i) is a logical predicate over the points in set R;

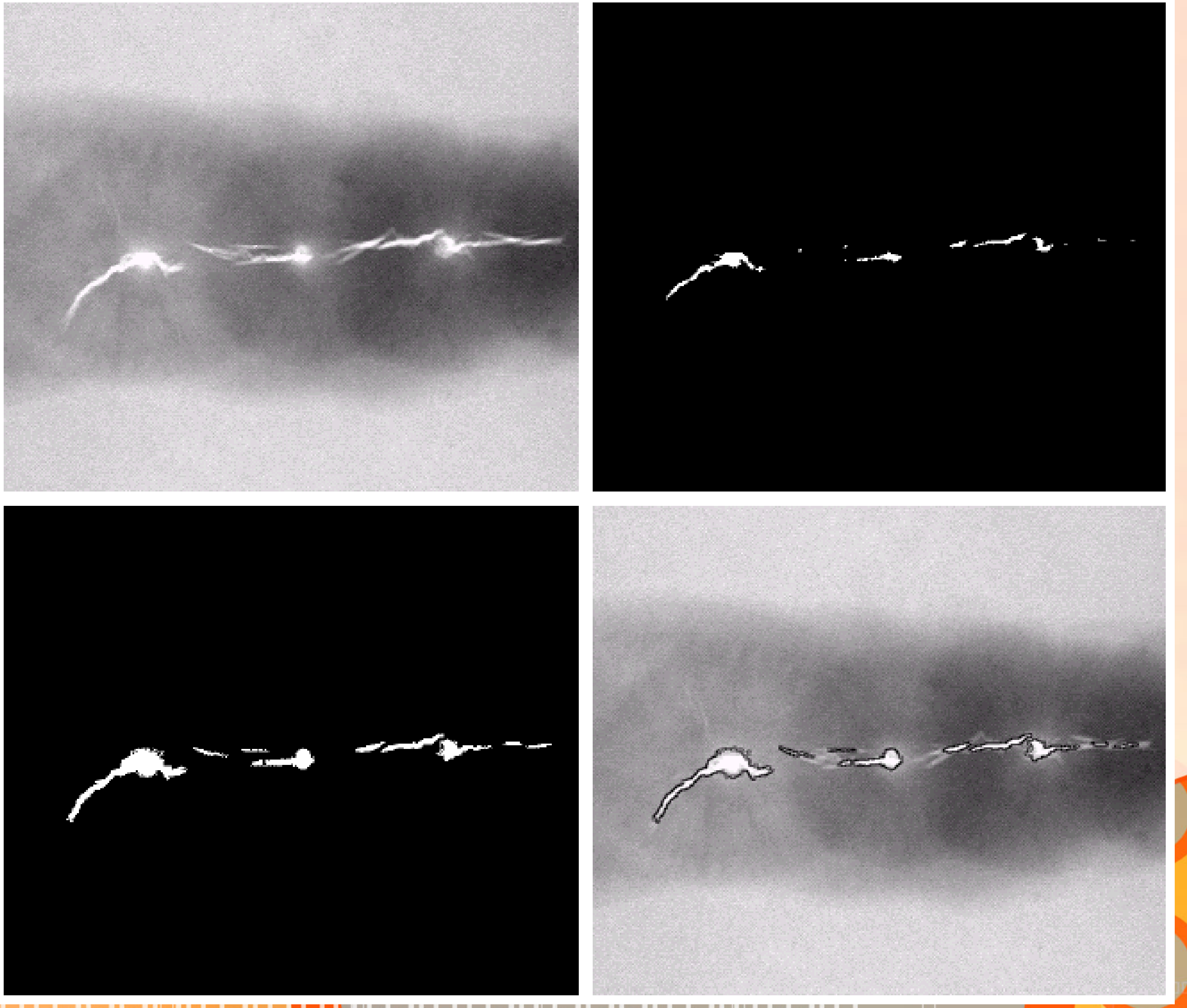
Region Growing by Pixel Aggregation

- Starts with a set of "seed" points
- Appending to each seed point those neighboring pixels with similar properties (such as gray level, texture, color)

Three problems for region growing

1. Selection of initial seeds
2. Selection of suitable properties
3. The formulation of a stopping rule

- Selecting starting points can be based on the nature of the problem
- For example, in military applications of infrared imaging, targets of interest are hotter (and thus appear brighter) than the background. Choosing the brightest pixels as starting point



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.).

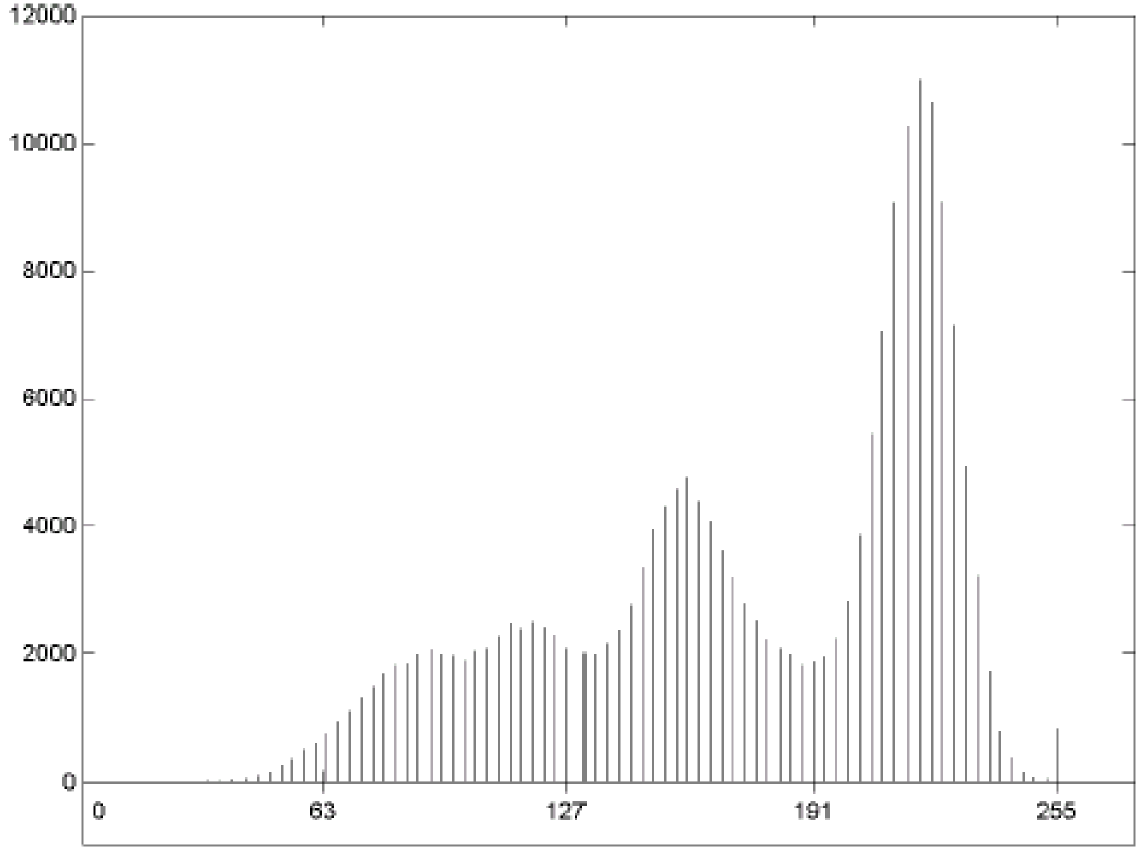
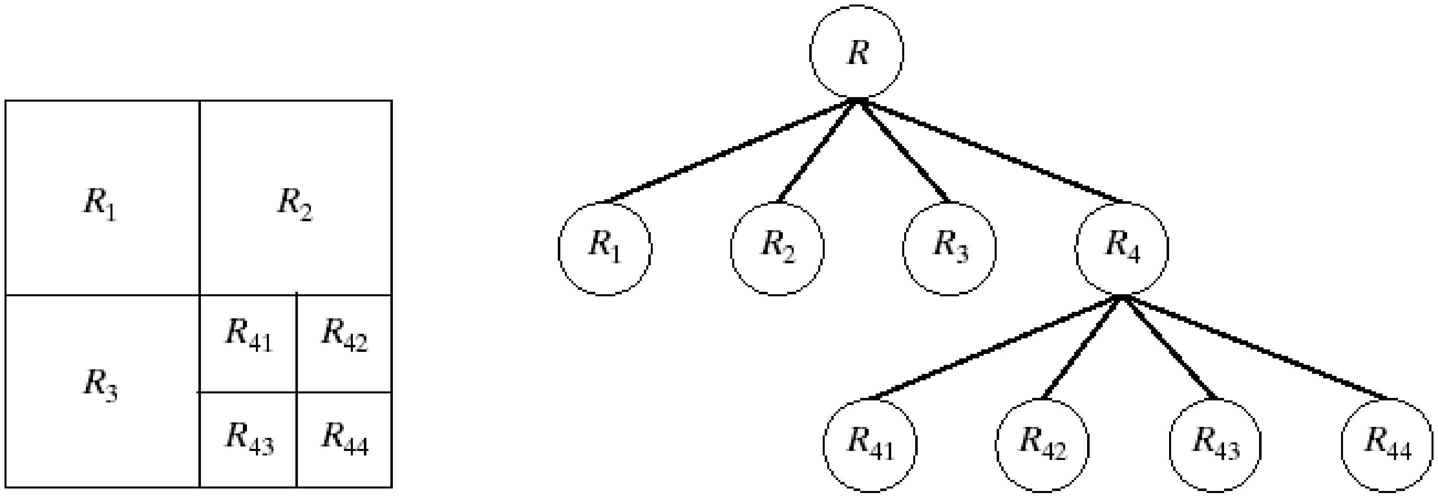


FIGURE 10.41 Histogram of Fig. 10.40(a).

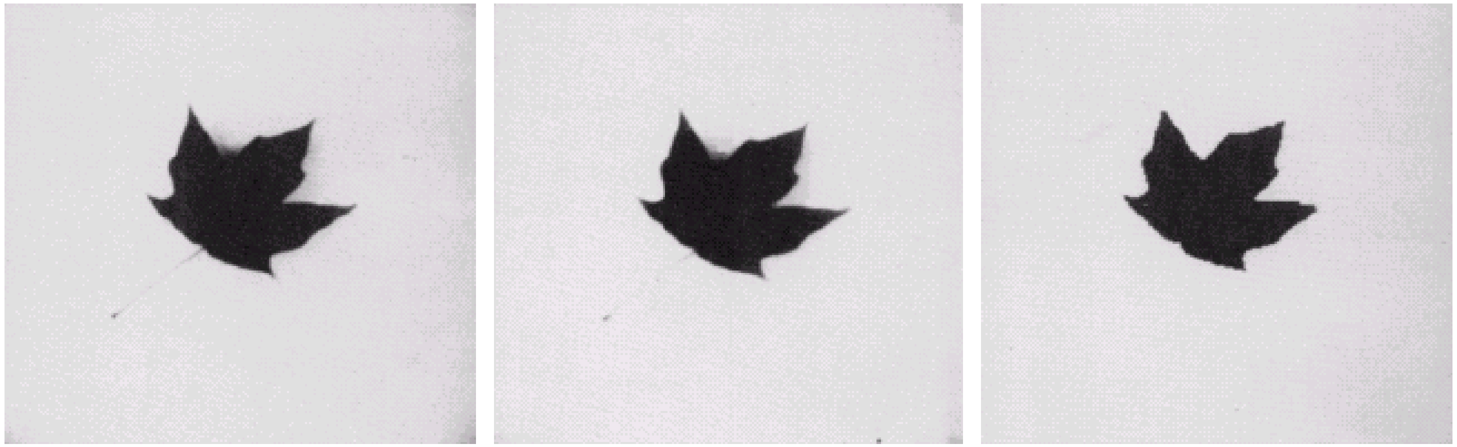
Region Splitting and Merging

- ❑ Let R represent the entire image region and select a predicate P
- ❑ For a square image, subdivide it successively into smaller and smaller quadrant regions so that, for any region R_i , $P(R_i) = \text{TRUE}$
- ❑ If $P(R_i) = \text{FALSE}$, divide the image into quadrants
- ❑ If P is FALSE for any quadrant, subdivide that quadrant into subquadrants, and so on
- ❑ This particular splitting technique has a convenient representation in the form of a so-called quad tree (that is, a tree in which each node has exactly four descendants), as illustrated in Fig. 10.42



a b

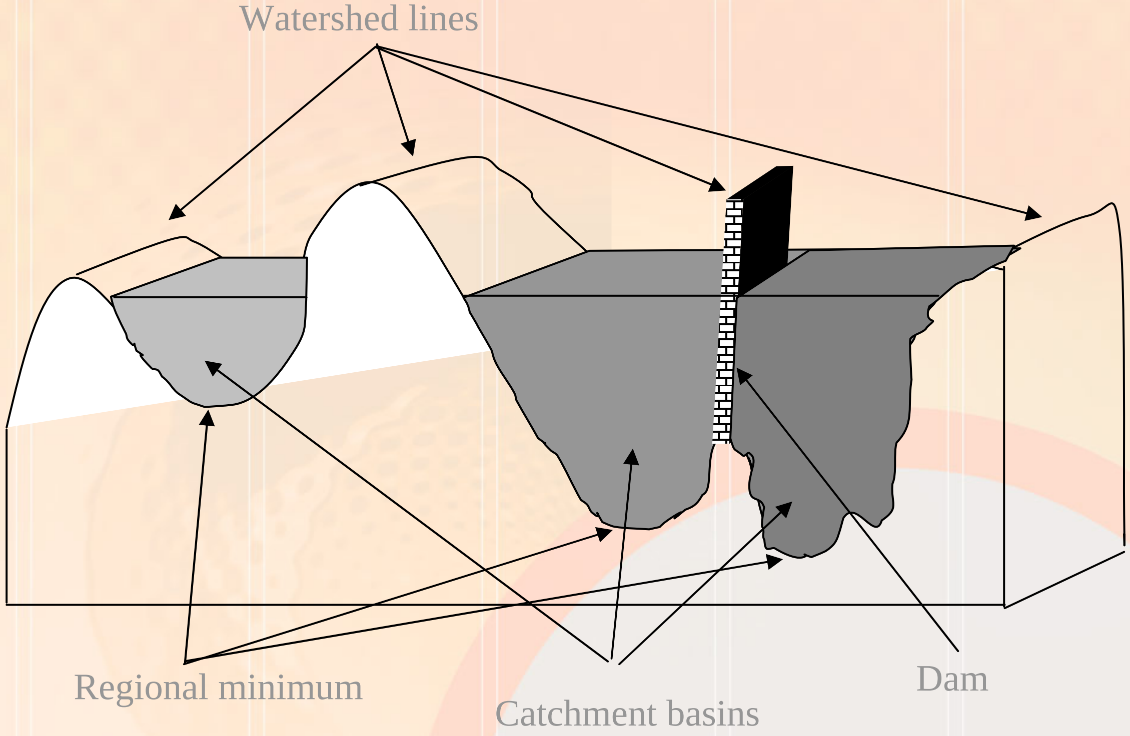
FIGURE 10.42
(a) Partitioned image.
(b) Corresponding quadtree.



a b c

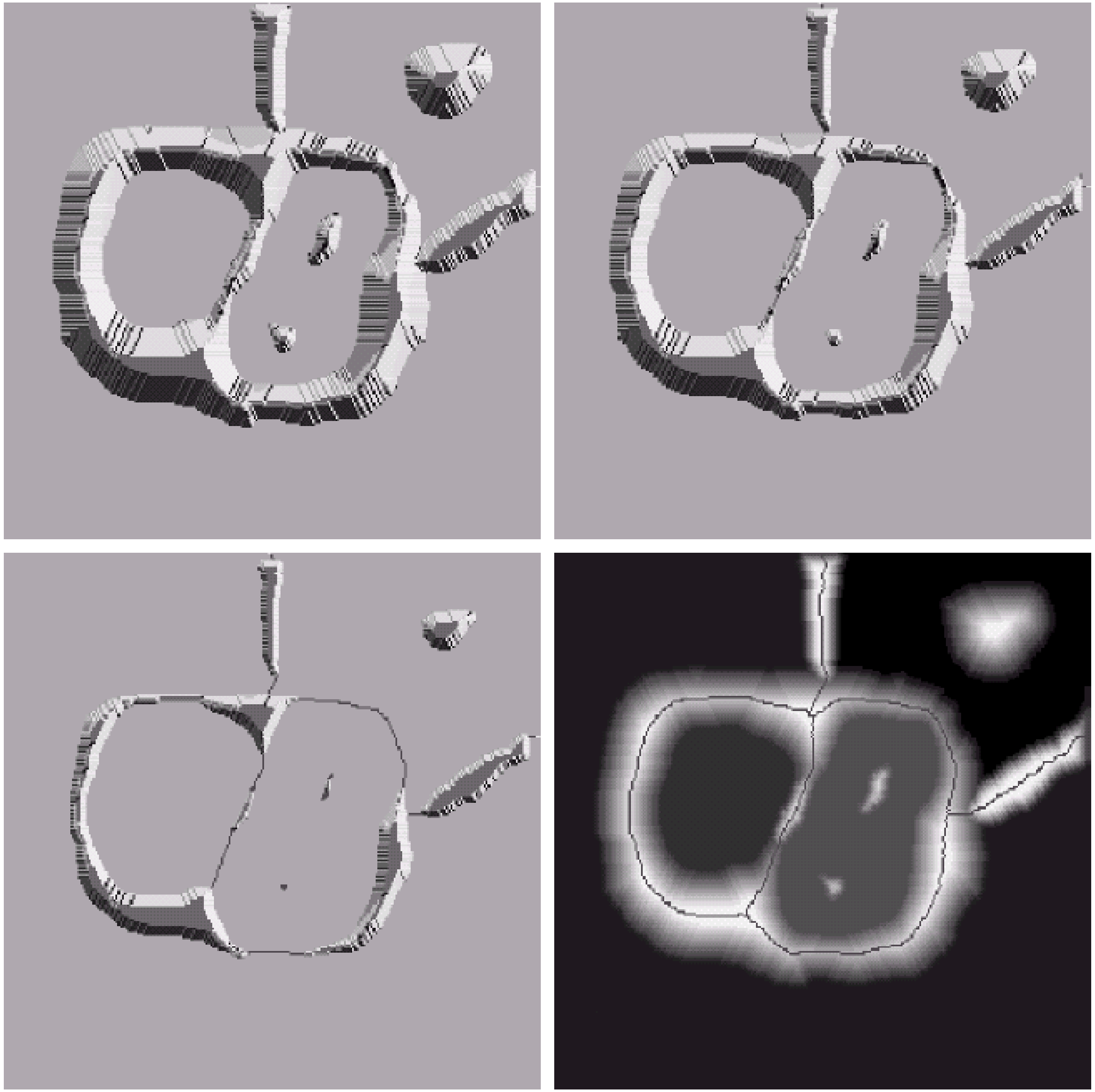
FIGURE 10.43
(a) Original image. (b) Result of split and merge procedure.
(c) Result of thresholding (a).

Morphological Watersheds



a b
c d

FIGURE 10.44
(a) Original image.
(b) Topographic view. (c)–(d) Two stages of flooding.



e	f
g	h

FIGURE 10.44 *(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.)

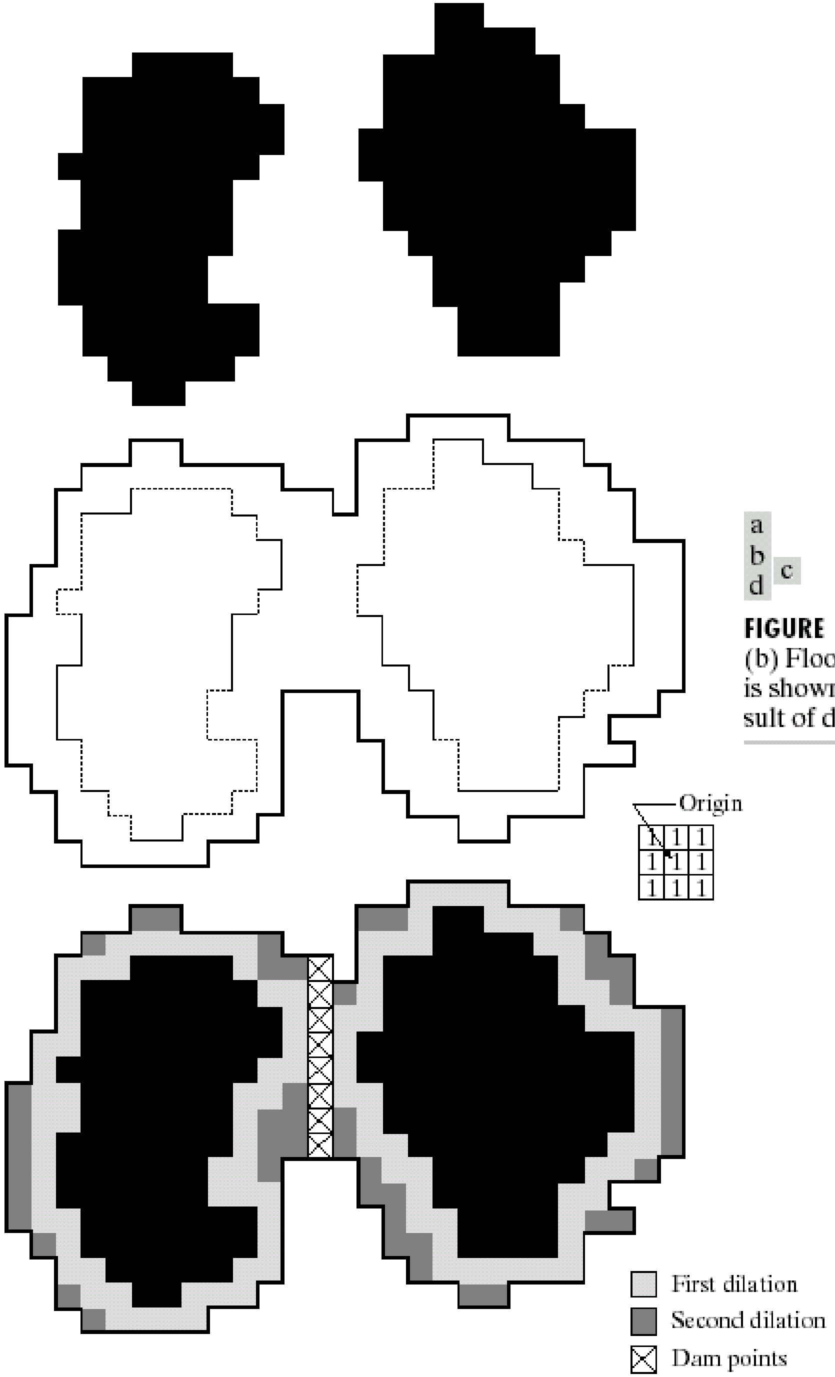
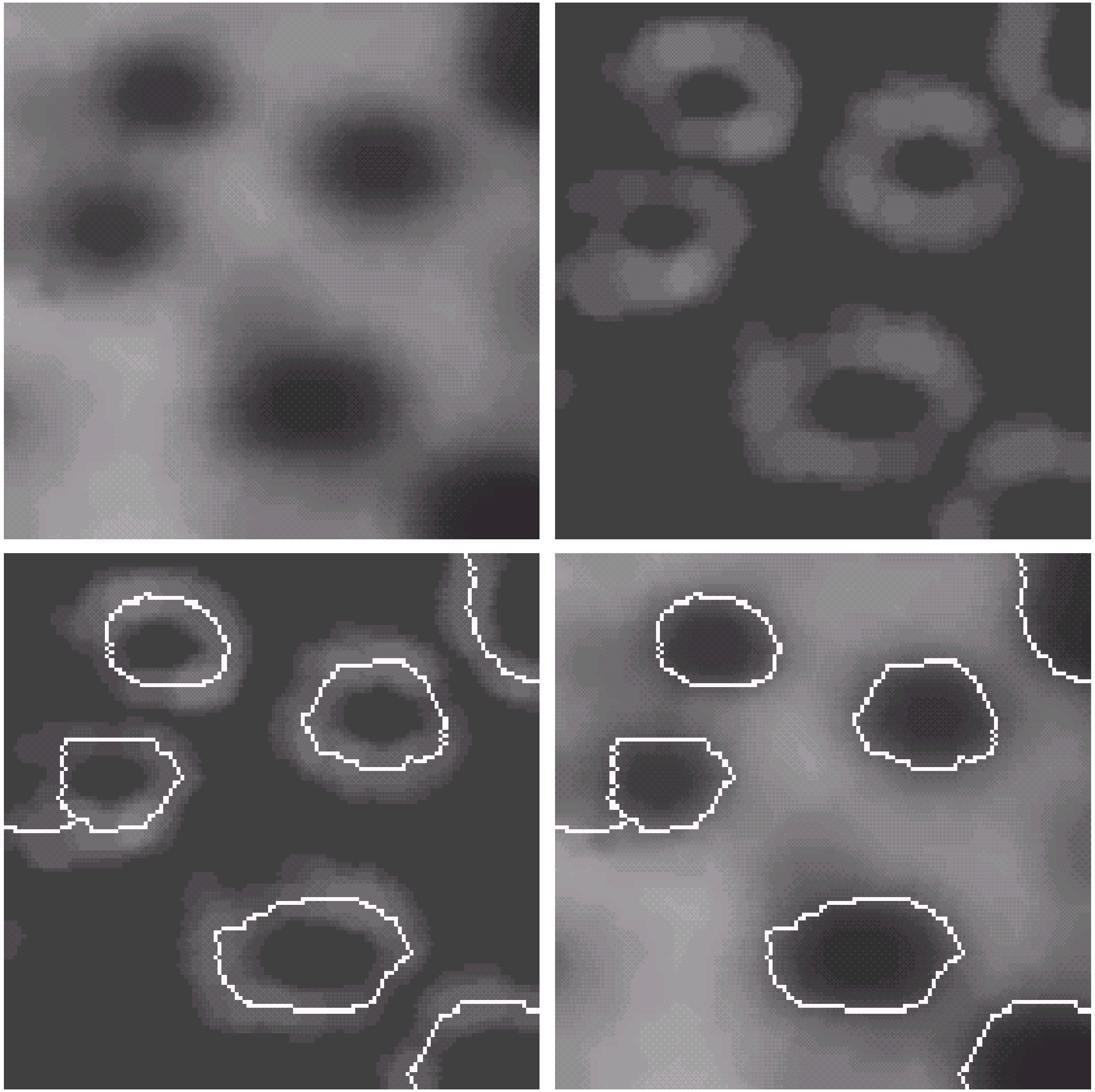
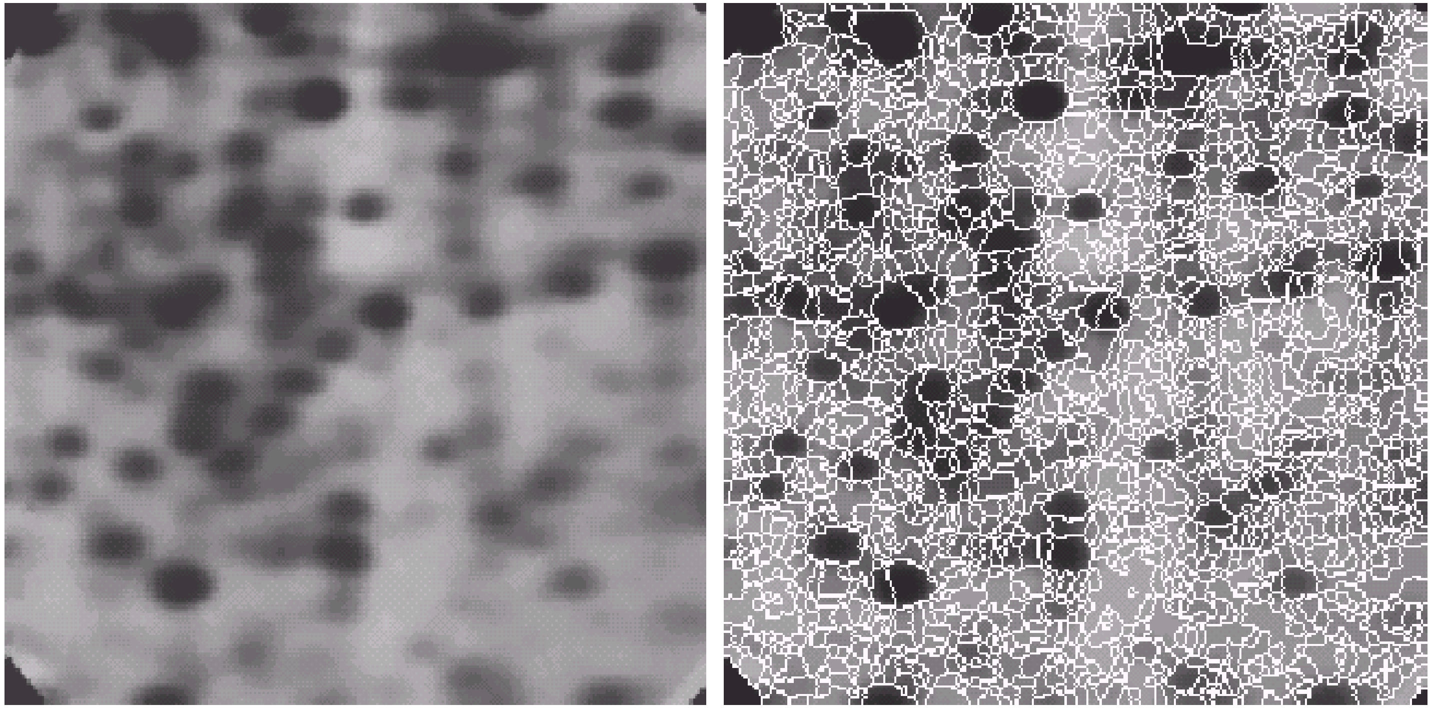


FIGURE 10.45 (a) Two partially flooded catchment basins at stage $n - 1$ of flooding. (b) Flooding at stage n, showing that water has spilled between basins (for clarity, water is shown in white rather than black). (c) Structuring element used for dilation. (d) Result of dilation and dam construction.



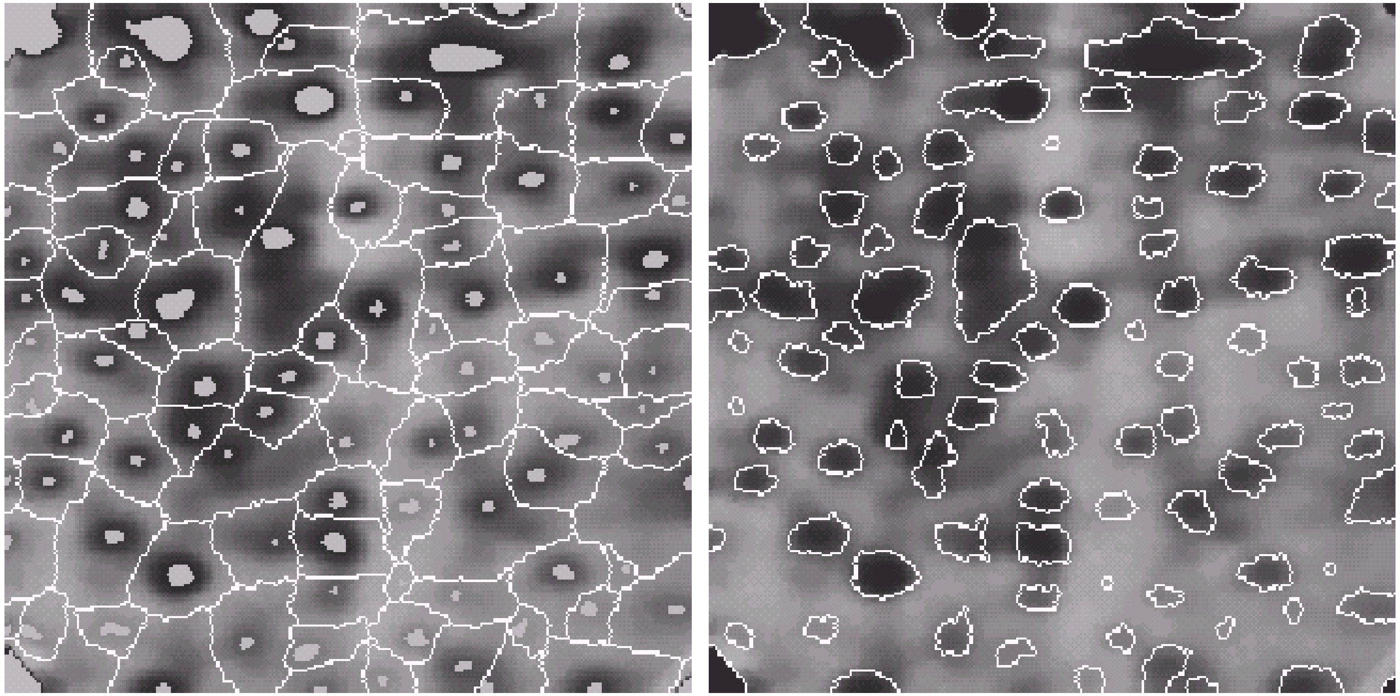
a b
c d

FIGURE 10.46
(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.)



a b

FIGURE 10.47 (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.)



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.)

THE USE OF MOTION IN SEGMENTATION

- Detecting changes between two image frames $f(x, y, t_i)$ and $f(x, y, t_j)$ is to compare the two images pixel by pixel.

- Suppose that we have a reference image containing only stationary components, comparing this image against a subsequent image having the same environment but including a moving object results in the difference of the two images canceling the stationary components, leaving only nonzero entries that correspond to the nonstationary image components.

- A difference image between two images taken at times ti and tj may be defined as

$$d_{ij} = \begin{cases} 1 & if \ \left| f(x,y,t_i) - f(c,y,t_j) \right| > \theta \\ 0 & otherwise \end{cases} \quad (7.5.1)$$

- In dynamic image analysis, all pixels with $d_{ij}(x, y)=1$ are considered as the result of object motion
- This approach is applicable only if the two images are registered and the illumination is relatively constant within the bounds established by θ.
- In practice, 1-valued entries in $d_{ij}(x, y)$ often arise as a result of noise. Typically, these entries are isolated points in the difference image

- A simple approach to their removal is to form 4- or 8-connected regions of 1 's in $d_{ij}(x, y)$ and then ignore any region that has less than a predetermined number of entries
- Although it may result in ignoring small and/or slow-moving objects, this approach improves the chances that the remaining entries in the difference image actually are the result of motion.

Accumulative differences

- A difference image often contains isolated noise. Although the number of these entries can be reduced or completely eliminated by a thresholded connectivity analysis, this filtering process can also remove small or slow-moving objects

- To address this problem by considering changes at a pixel over several frames thus introducing a "memory" into the process

- The basic idea is to ignore changes that occur only sporadically over a frame sequence

Implementation

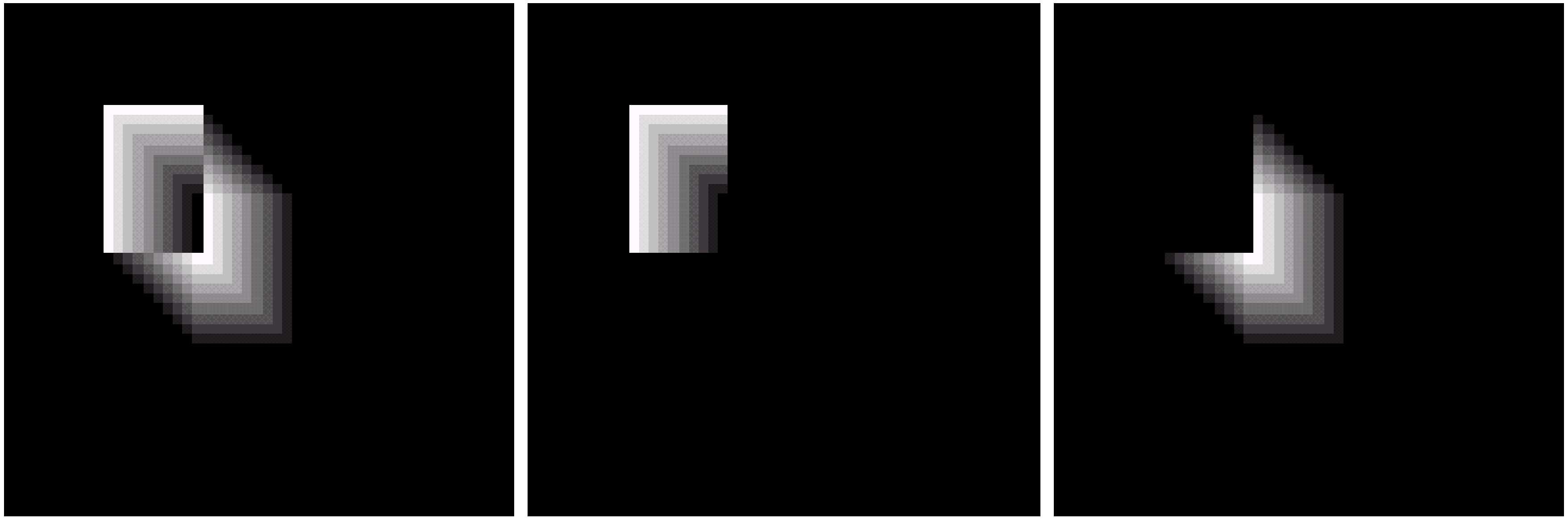
- Consider a sequence of image frames $f(x,y,t_1)$, ..., $f(x, y, t_n)$

- Let $f(x, y,t_1)$ be the reference image

- An accumulative difference image is formed by comparing this reference image with every subsequent image in the sequence

- A counter for each pixel in the accumulative image is incremented every time a difference occurs at that pixel between the reference and an image in the sequence

Accumulative difference image (ADI)
Eq. 10.6-2~4



a b c

FIGURE 10.49 ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.

a b c

FIGURE 10.50 Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)