Today’s Agenda

- Image Segmentation

Advanced Edge Detection Techniques

- **Deal with image noise**
- **Exploit the properties of image**

$\Rightarrow$ **Work much better for real images**

Advanced edge detectors:

- Laplacian of Gaussian (LoG)
- Difference of Gaussian (DoG)
- Canny

Marr-Hildreth Detector (LoG)

Observations:

- Intensity changes are dependent on the image scale
- A sudden intensity change (step) causes a peak/trough in the1st order derivative and a zero-crossing in the 2nd order derivative
- The 2nd order derivative is especially sensitive to noise

$\Rightarrow$ Smooth the image using a Gaussian filter first before applying the Laplacian

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

$$g(x,y) = \nabla^2[G(x,y) \otimes f(x,y)] = \underbrace{\boxed{\nabla^2 G(x,y)}}_{\textbf{Laplacian of a Gaussian (LoG)}} \otimes f(x,y)$$

Marr-Hildreth Detector (LoG)

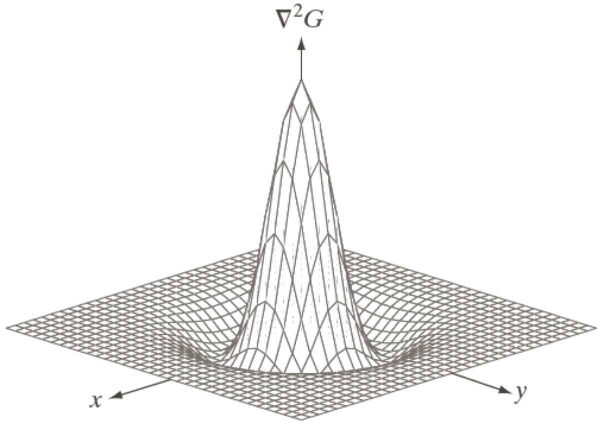
Laplacian of a Gaussian $(LoG):\qquad \nabla^2 G(x,y)$

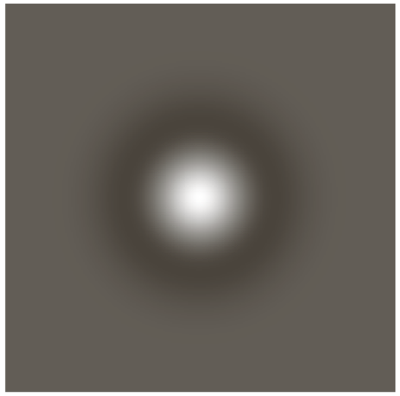
$$\text{Gaussian} \longrightarrow G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

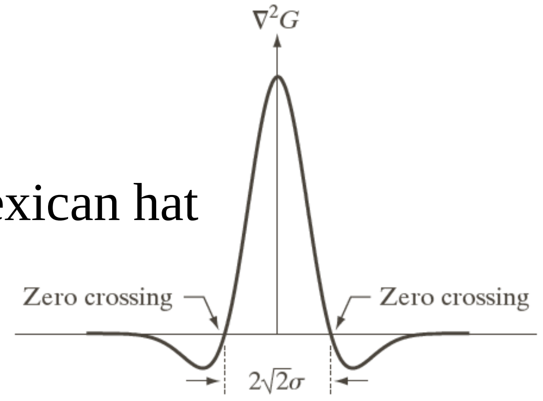
- Varying σ values for scale changes
- Rotation invariant in edge detection

LoG Cont'd





Mexican hat



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.

$\longrightarrow$ Sum to zero!

In practice, we use the negative of the mask

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

LoG Filtering

$$g(x,y) = \nabla^2 G(x,y) \otimes f(x,y)$$
$$= \nabla^2 [G(x,y) \otimes f(x,y)]$$

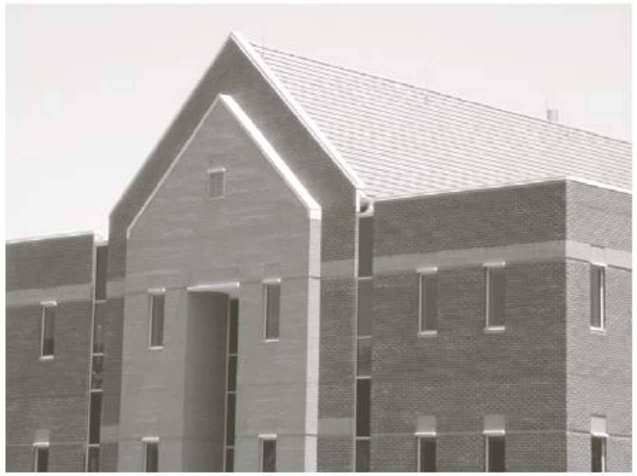
1. **Filter the input image with an $n \times n$ Gaussian filter.**
2. **Compute the Laplacian of the intermediate image resulting from Step 1.**
3. **Find the zero-crossings of the image from Step 2.**
   - opposite signs of the neighbors
   - the difference should be significant

Note:

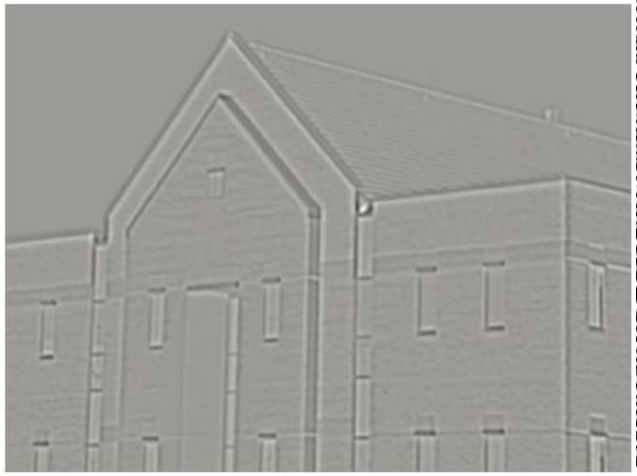
- Window size $n \geq 6\sigma$ and n is an odd number

An Example – Edges are 1 Pixel Thick

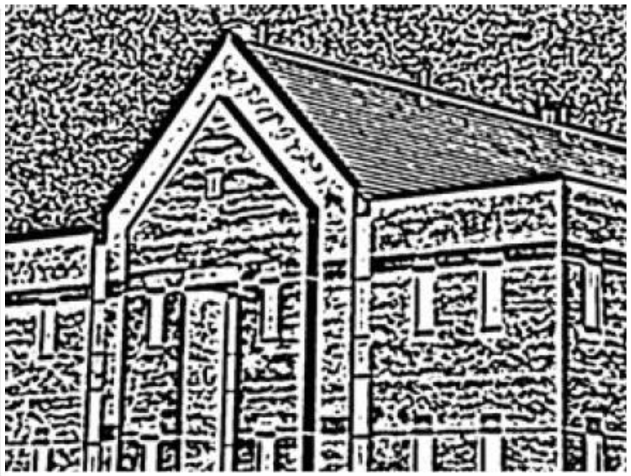
Original

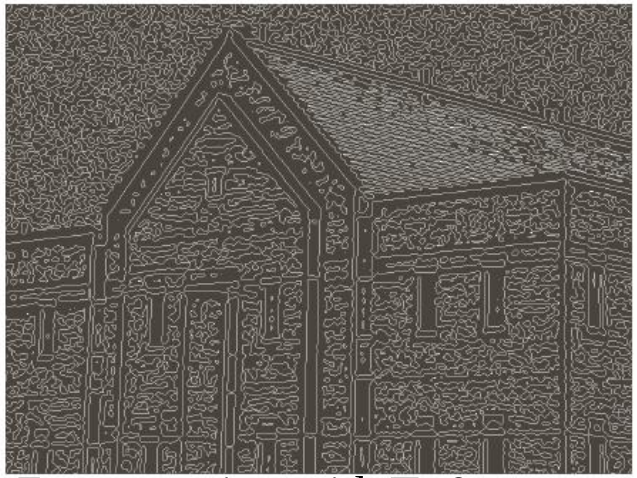


LoG filtering

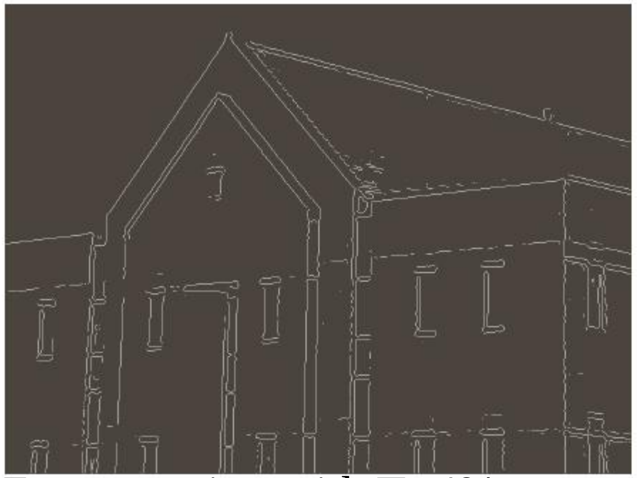


LoG filtering with T=0





Zero-crossing with T=0



Zero-crossing with T=4%max

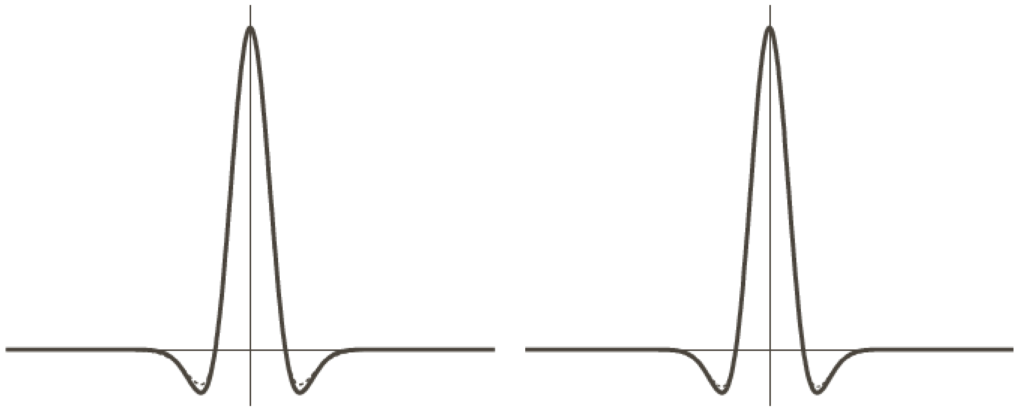
In practice, run LoG many times with different sigmas and keep the edges that are common for all sigmas

Approximate LoG by DoG

LoG needs multiple filters for scale variations

→ Difference of Gaussian (DoG)

$$DOG(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}, \quad \sigma_1 > \sigma_2$$



a b

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

Canny Edge Detector

The general goal:

1. Low error rate: all edges should be found and there should be no false response
2. Edge points should be well localized: the edges located must be as close as possible to the true edges
3. Single edge point response: the detector should return only one point for each true edge point

Canny edge detector: expressing these three criteria mathematically and then find optimal solutions

Canny Edge Detector

Gaussian smoothing + 1st order derivative

$\Rightarrow$ **optimal step edge detector**

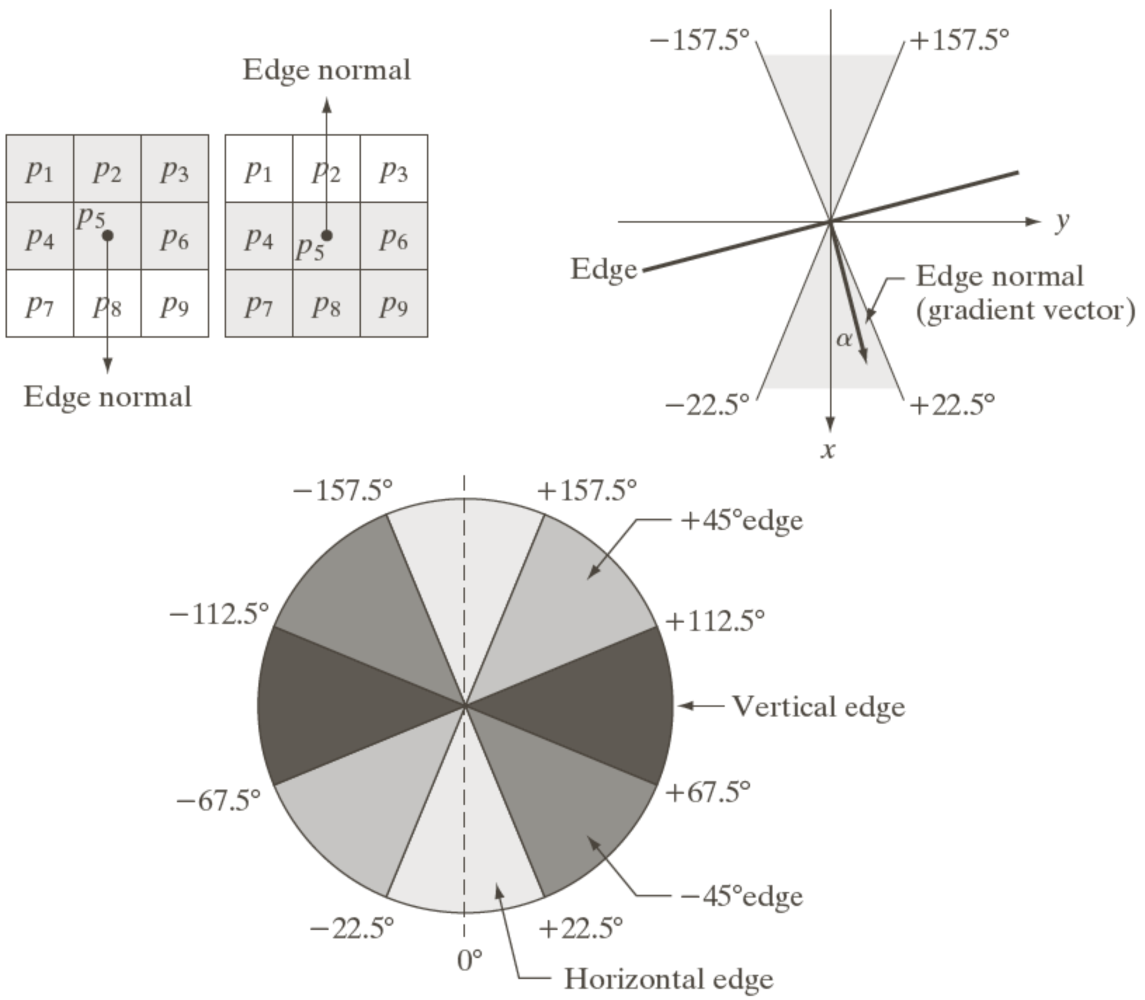
$$f_s(x,y) = G(x,y) \otimes f(x,y)$$

$$g_x = \partial f_s / \partial x \qquad g_y = \partial f_s / \partial y$$

$$M(x,y) = \sqrt{g_x^2 + g_y^2} \qquad \alpha(x,y) = \tan^{-1}\left(g_y / g_x\right)$$

$$\textbf{Thick edge} \xrightarrow{\text{Nonmaxima suppression}} \textbf{Single edge}$$

Quantize the Edge Direction



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3×3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3×3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Canny Detector -- Algorithm

1. **Smooth the input image with a Gaussian filter**
2. **Compute the gradient magnitude and angle images**
3. **Apply nonmaximum suppression on the gradient magnitude image**
    1. At (x, y), find the quantized edge normal d_k
    2. If the value $M(x, y)$ is less than at least one of its two neighbors along d_k, let $g_N(x, y) = 0$; otherwise $g_N(x, y) = M(x, y)$
4. **Reduce false edge: double thresholding and connectivity analysis to detect and link edges**
    1. High-threshold → strong edge pixels →valid edge pixels
    2. Low-threshold → weak edge pixels → valid only when connected to strong edge pixels

An Example

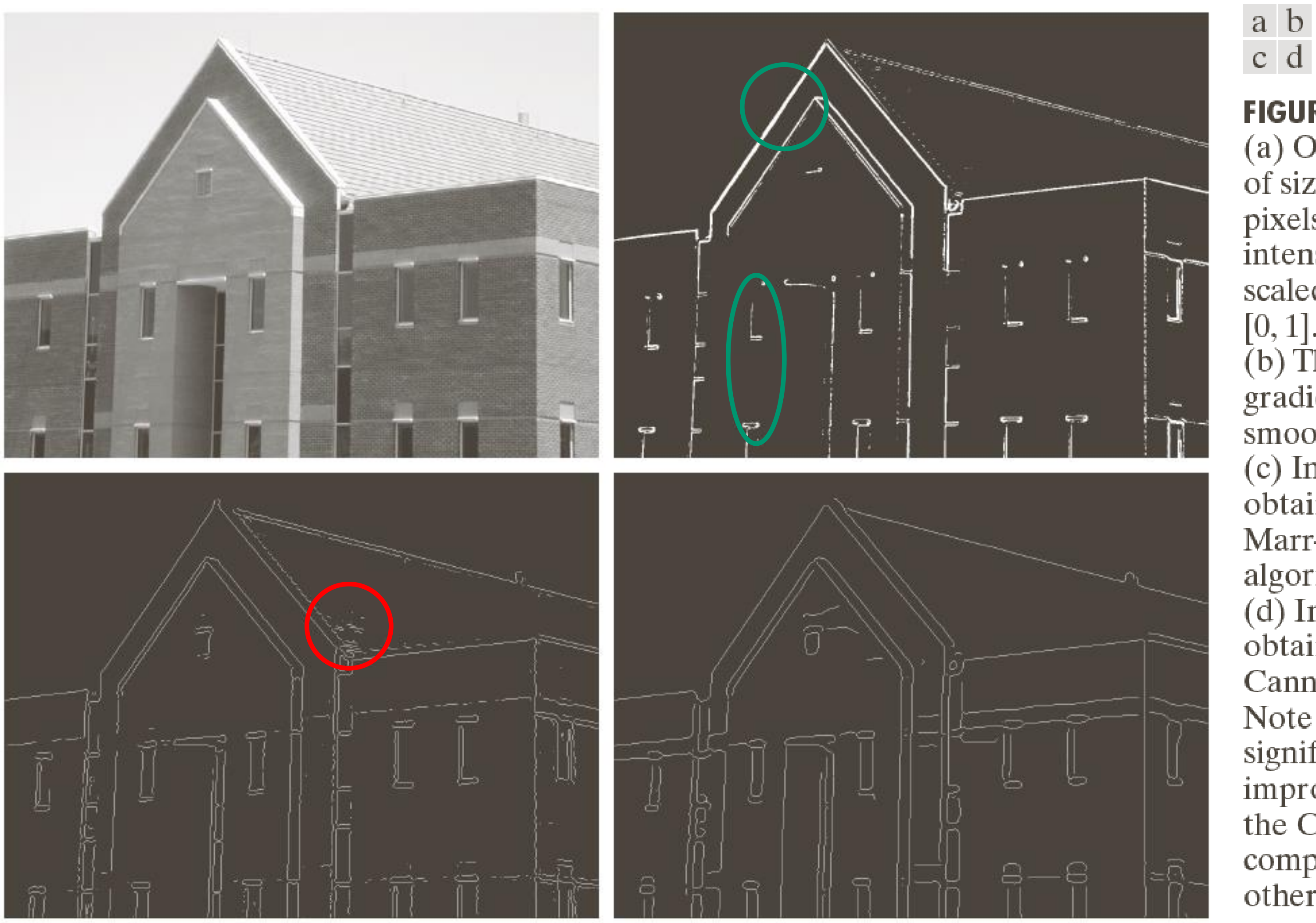


FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

An Example

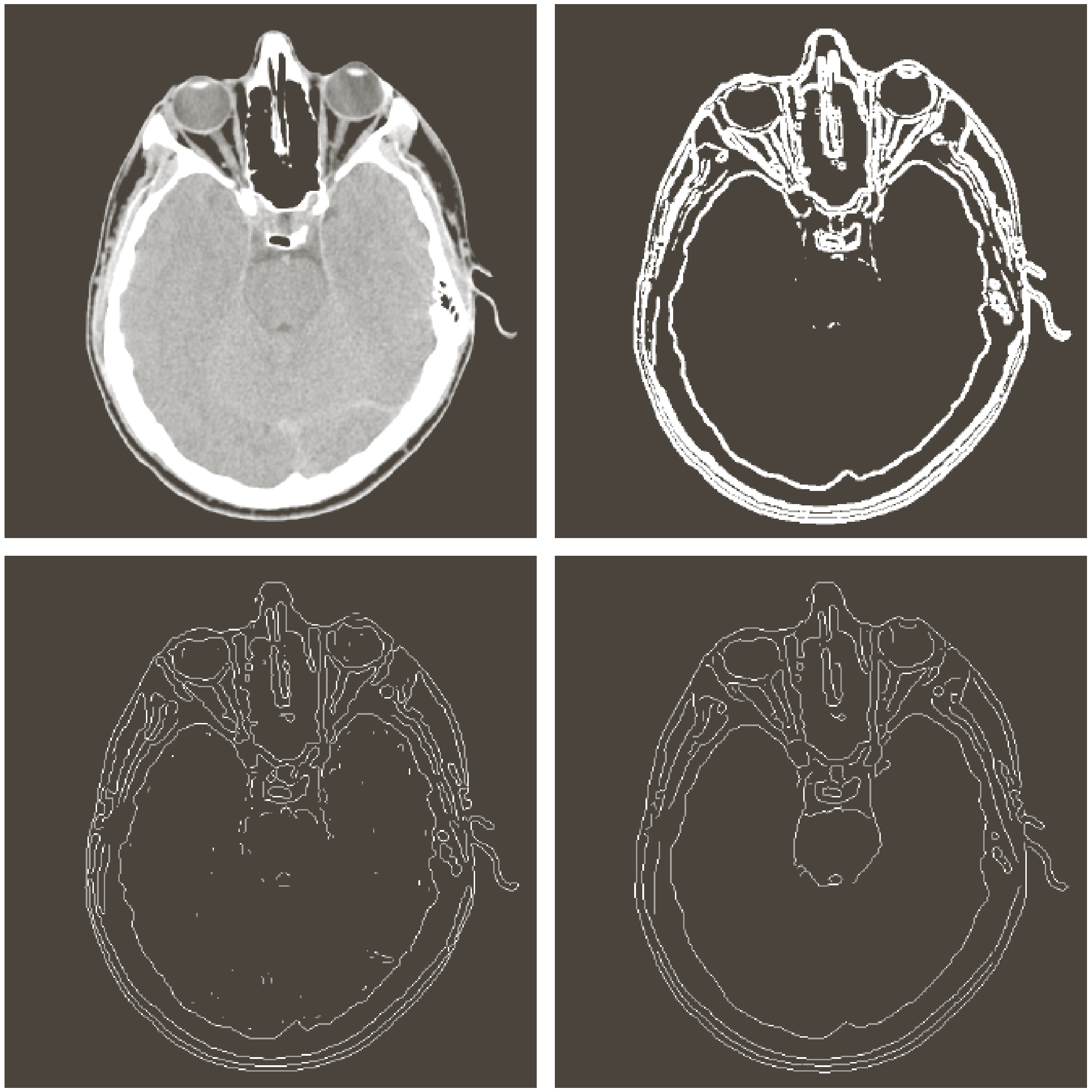


FIGURE 10.26 (a) Original head CT image of size 512 × 512 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Edge Linking and Boundary Detection

All edge detection algorithms can only detect fragments of boundaries, due to image noise, non-uniform illuminations, or other effects

Edge linking: link edges into longer meaningful edges or a full region boundaries

Three classic methods:

- Local processing
- Regional Processing
- Hough transform

Edge Linking – Local Processing

Link the edge points with similar properties:

- Strength
- Direction

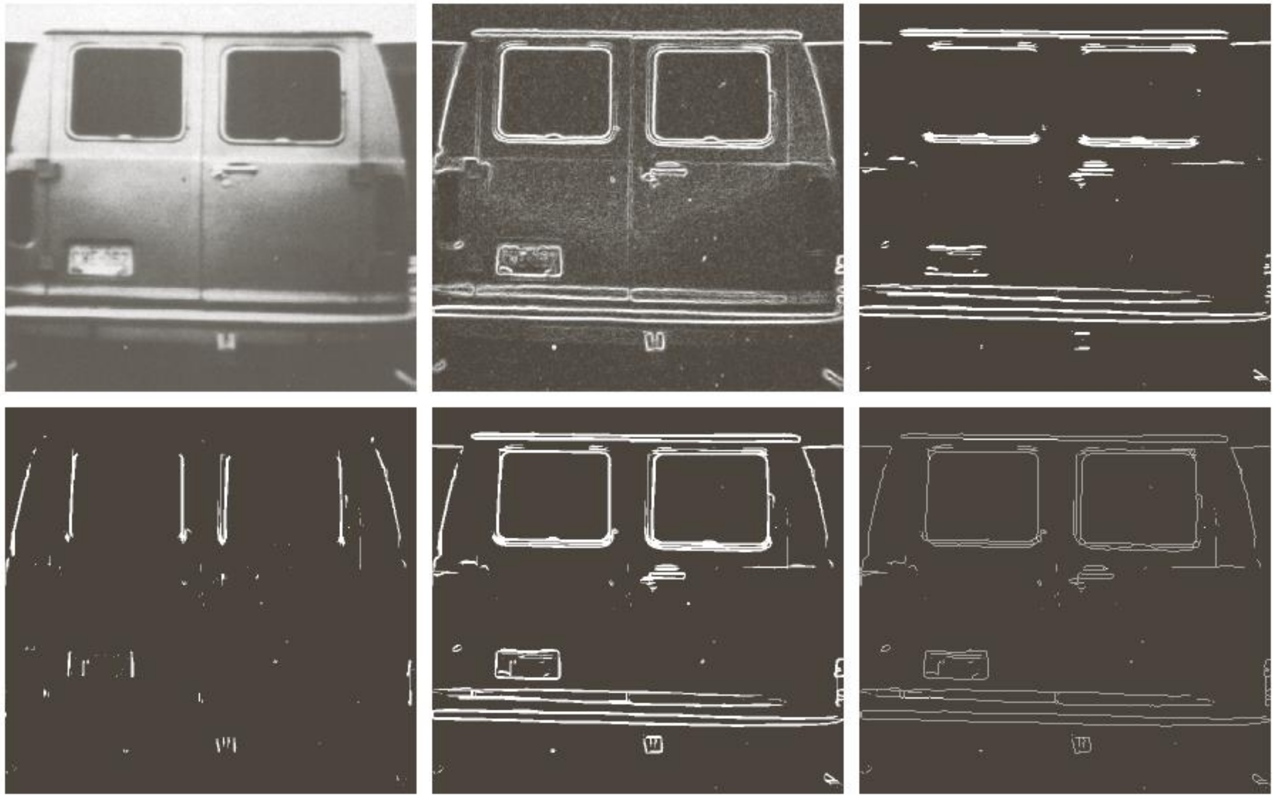
Two edge pixels are linked if

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

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

A Local Processing Example

Objective: find rectangular region for license plate recognition.



a b c
d e f

FIGURE 10.27 (a) A 534 × 566 image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)

Global Processing

A basic idea: Given *n* points

1. Find $n(n-1)/2$ lines between each pair of points
2. Find all subset of points that are close to particular lines. This needs $n(n-1)n/2$ comparisons

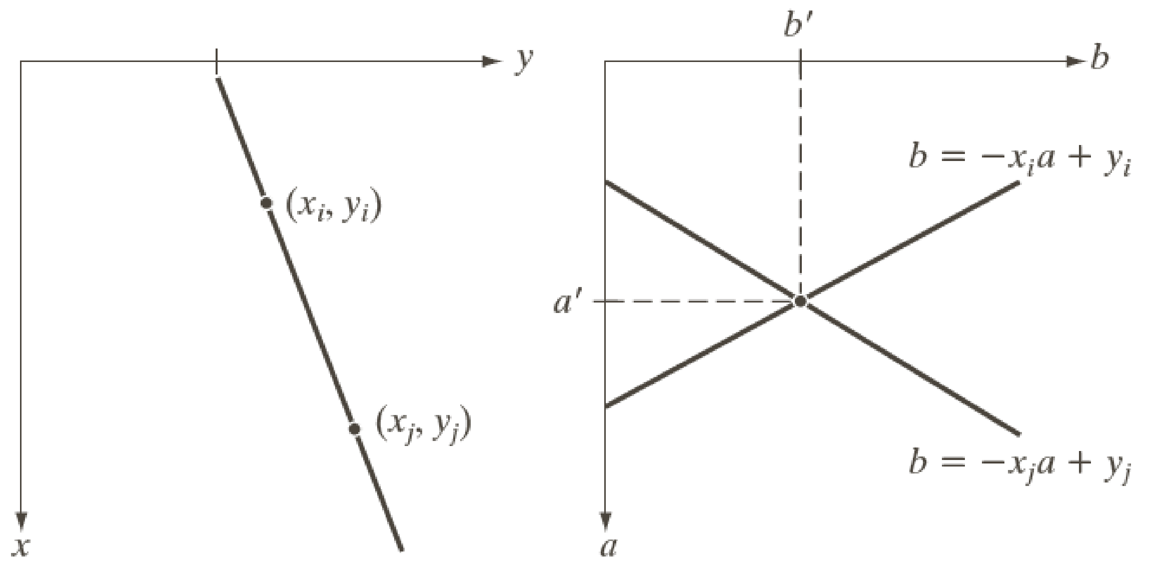
This is computationally expensive!

→ **Hough transform**

Hough Transform

A line in x-y plane is a point in the parameter plane.
A point in x-y plane is a line in the parameter plane.

X-Y plane ⟷ Parameter plane



a b

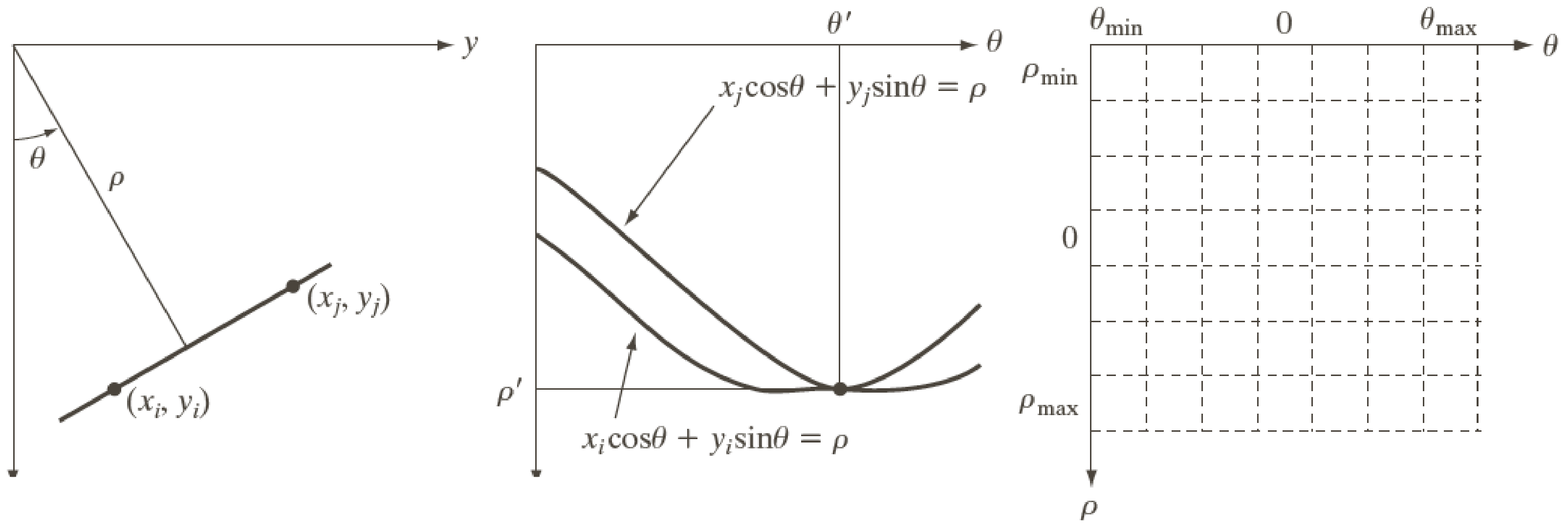
FIGURE 10.31
(a) *xy*-plane.
(b) Parameter space.

Problem of slope-intercept form: the slope a **approaches infinity for vertical lines**

Hough Transform

A line in x-y plane is a point in the parameter plane.
A point in x-y plane is a sinusoidal in the parameter plane (polar space).

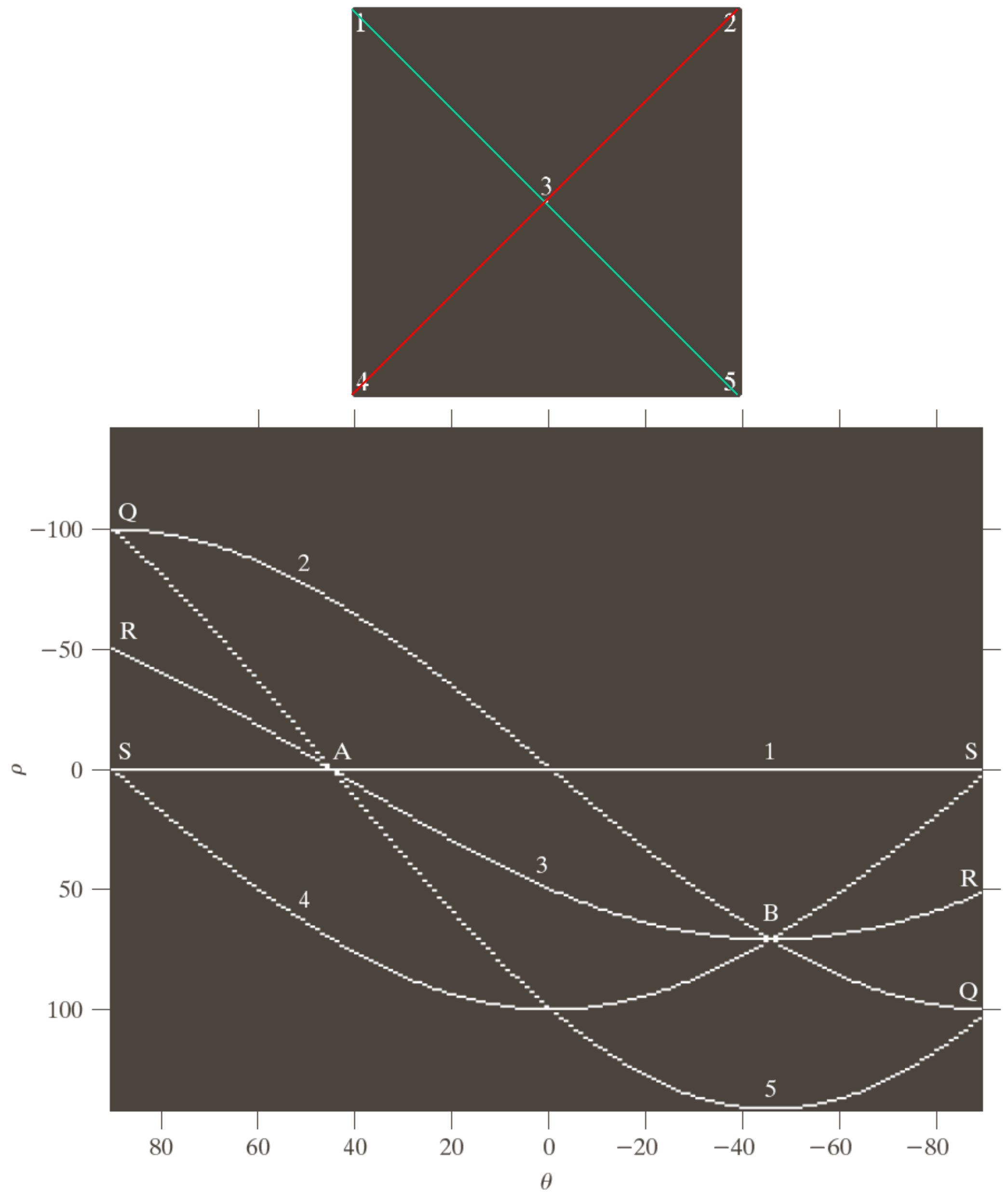
$$x\cos\theta + y\sin\theta = \rho$$



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.

A Toy Example



a
b

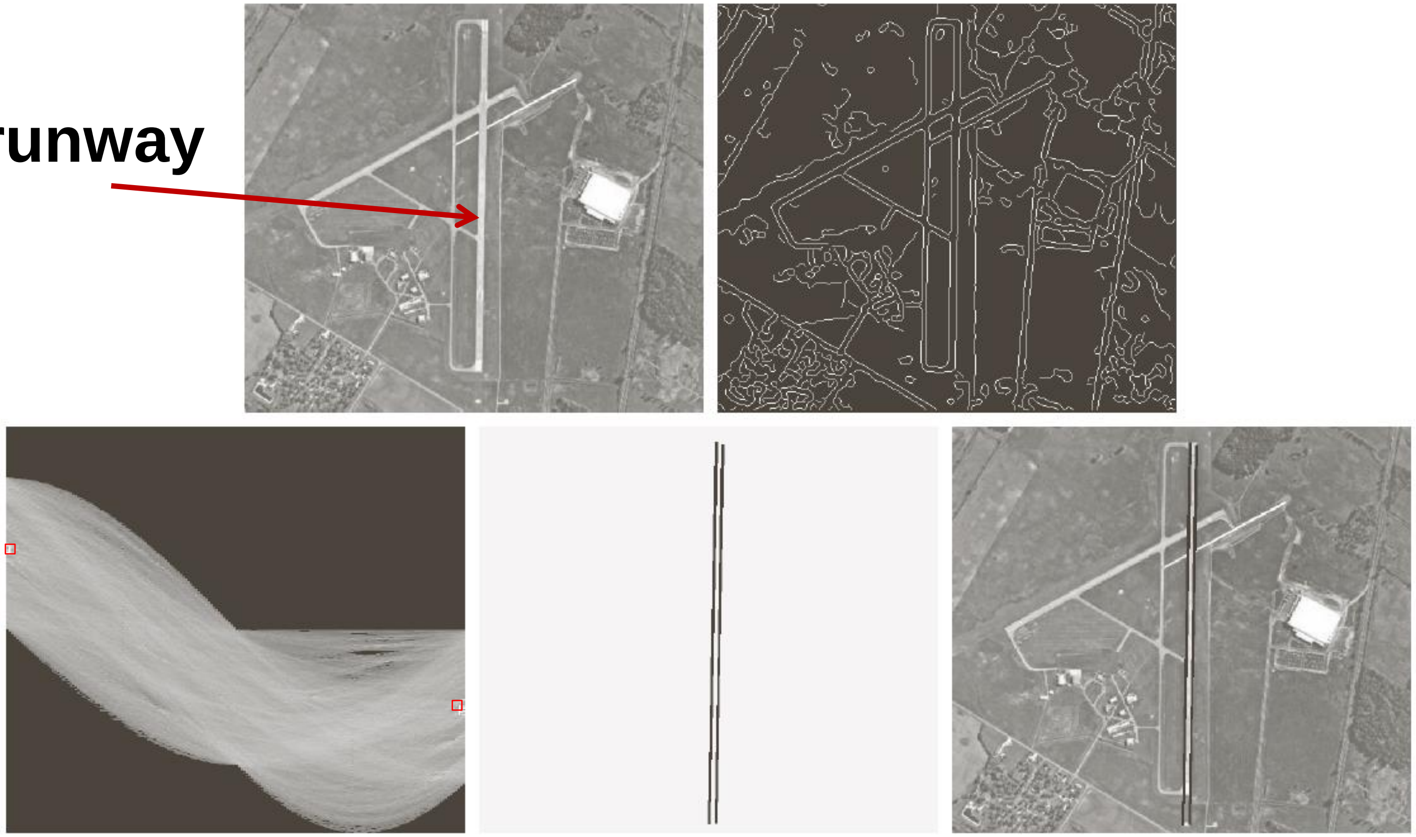
FIGURE 10.33 (a) Image of size 101 × 101 pixels, containing five points. (b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

A: green line

B: red line

Hough Transform – Real Example

Find the runway

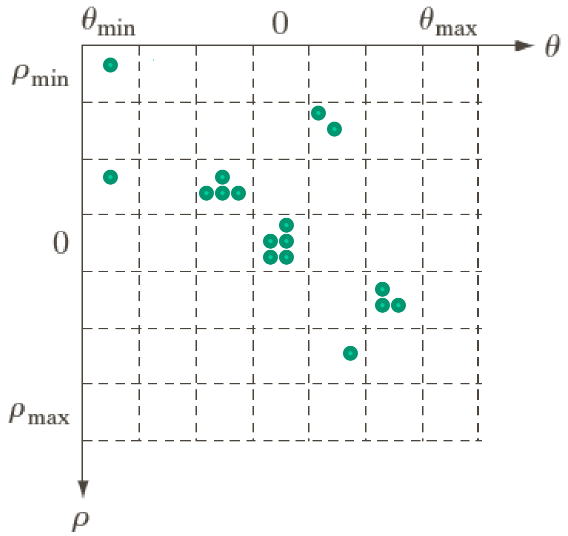


a b
c d e

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

Hough Transform Algorithm

- Obtain a binary edge image using any edge detector
- Specify subdivisions in the ρ-θ plane
- For each edge point
    - For each θ value in the accumulator cell
        - Update the accumulator cell with the corresponding ρ
- Examine the counts of accumulator cell for high pixel concentrations
- For each chosen cell, link the pixels based on the continuity



$$-90^0 \le \theta \le 90^0, -D \le \rho \le D$$

Note

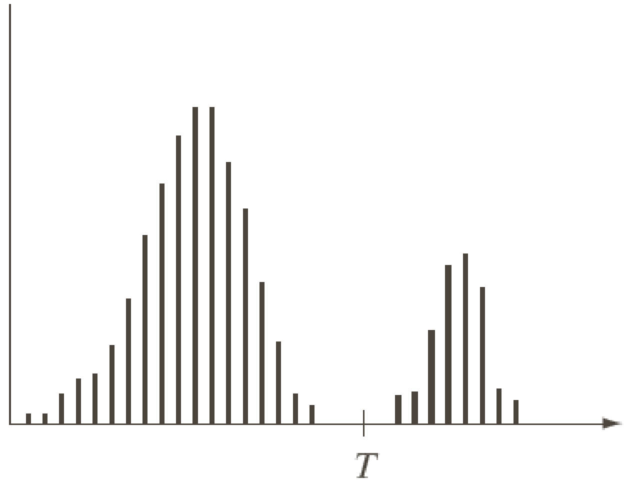
Thresholding is required to get edge pixels and realize edge-based segmentation

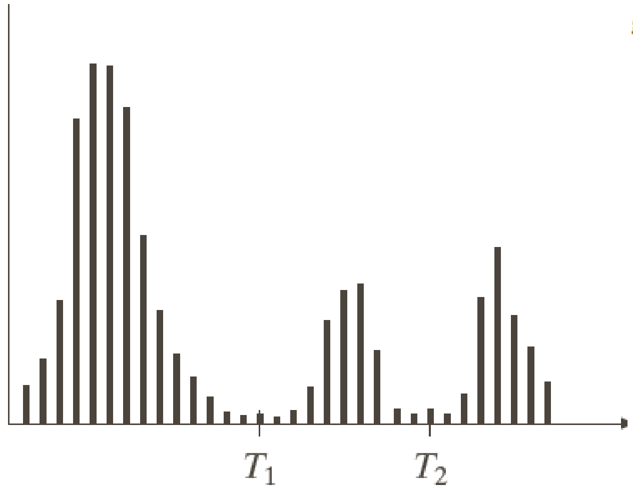
Boundary detection (edge linking) is still a hot research topic in image processing and computer vision

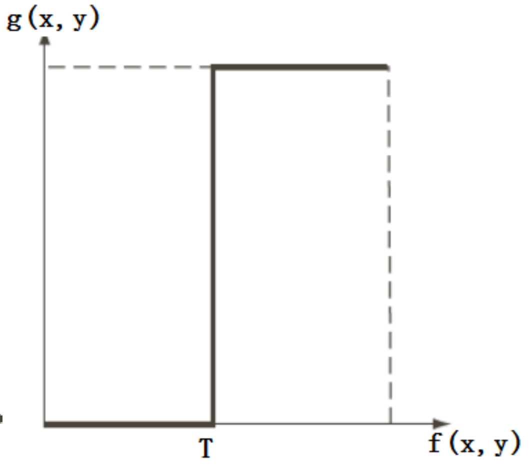
Incorporate domain knowledge:

- Psychology rules on the boundary: smooth, convex, symmetry, closed, complete, etc
- Template shape information: hand, stomach, lip, etc
- Appearance information: region-based texture, intensity/color, etc.

Intensity Thresholding







Object/background segmentation:

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

- A constant T - global thresholding
- A variable T - local/regional thresholding; adaptive thresholding
- Multiple T - multiple thresholding

Key Factors Affect Thresholding

- Separation between peaks
- Noise level
- Relative sizes of objects and background
- Uniformity of the illumination source
- Uniformity of the reflectance of the image

The Role of Noise in Image Thresholding

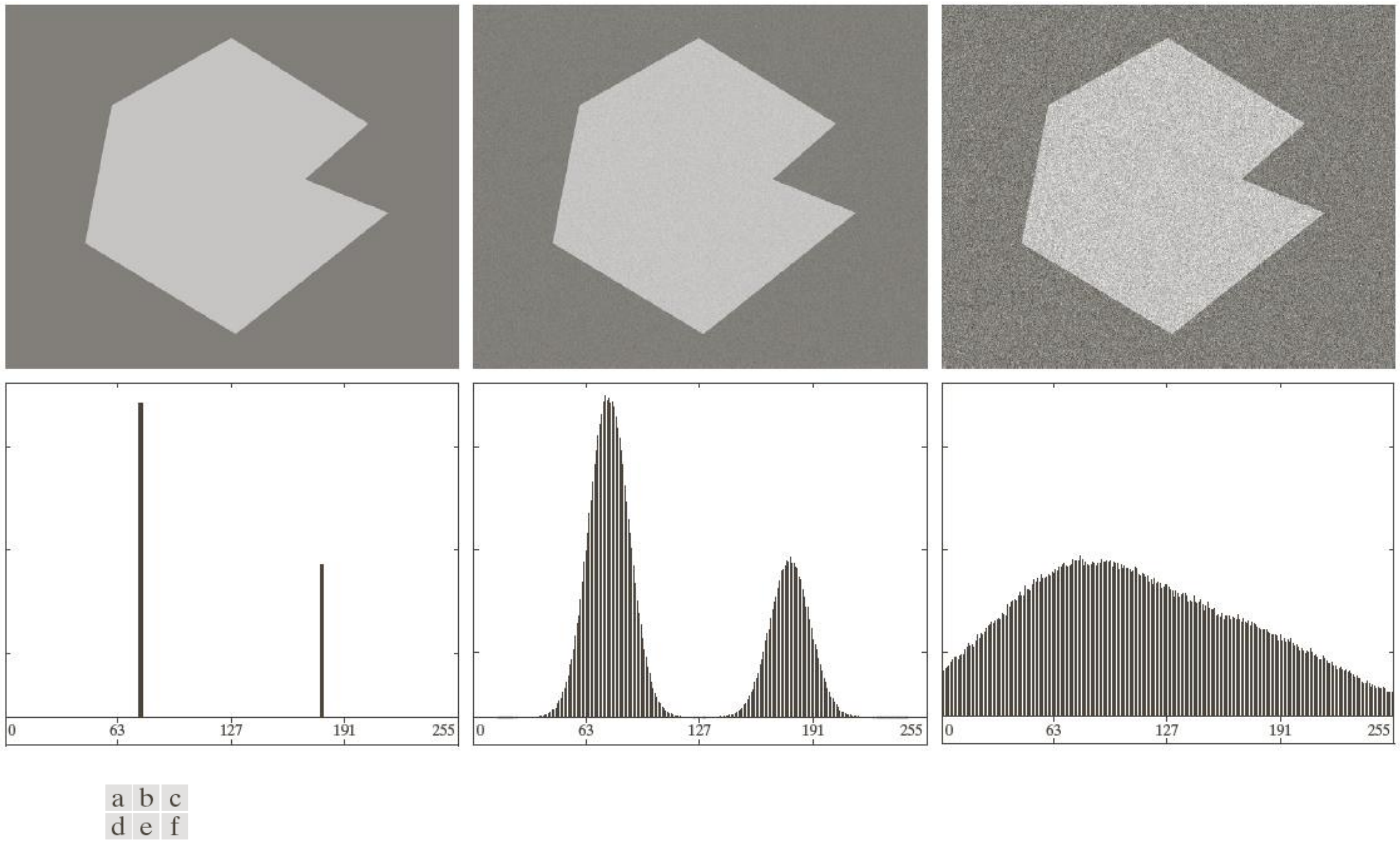


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

The Role of Illumination in Thresholding

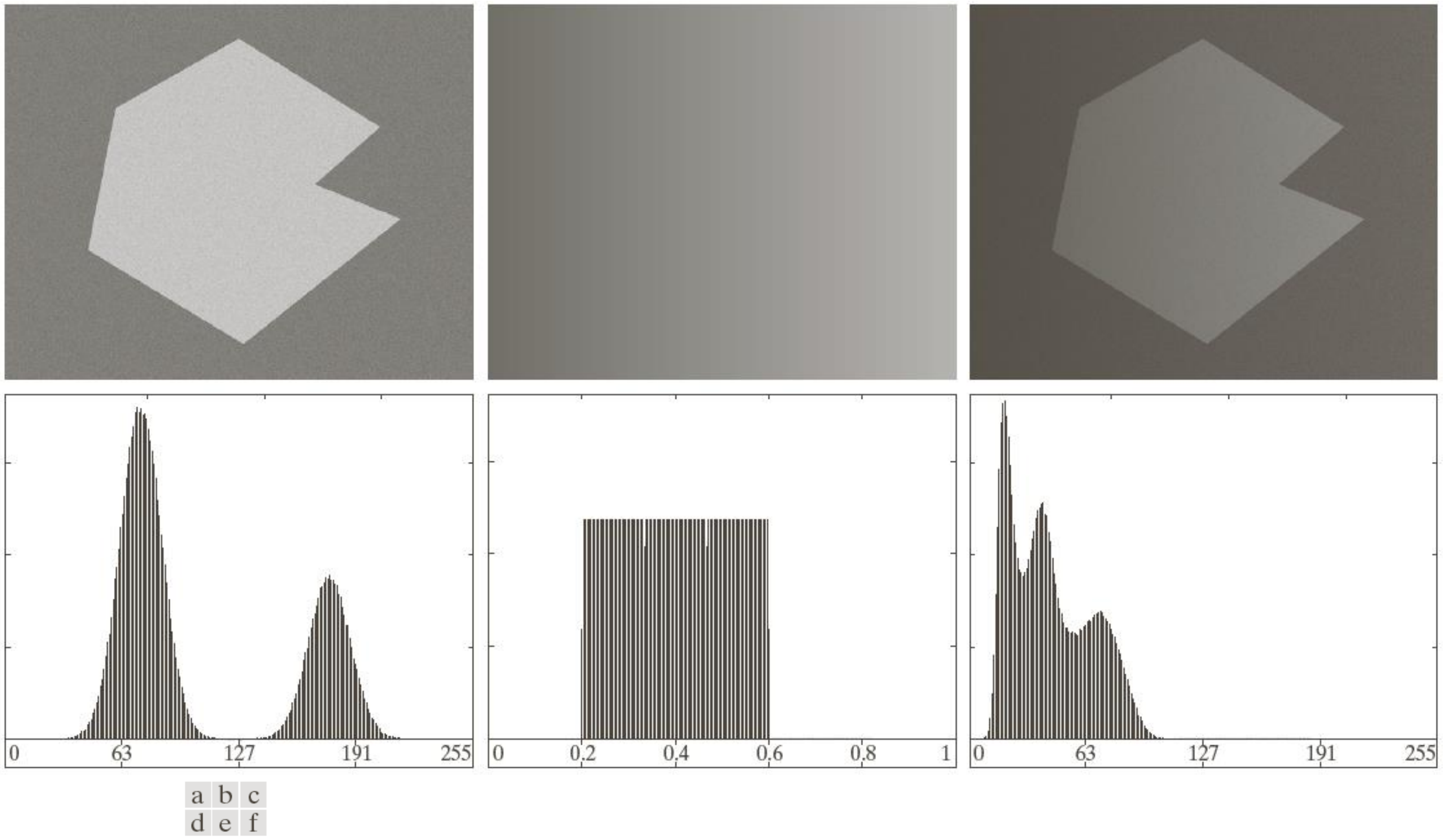
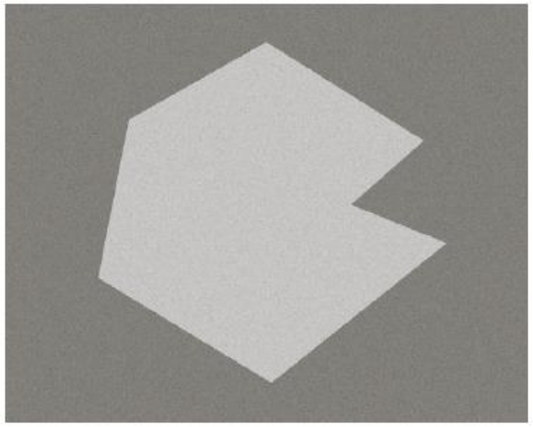
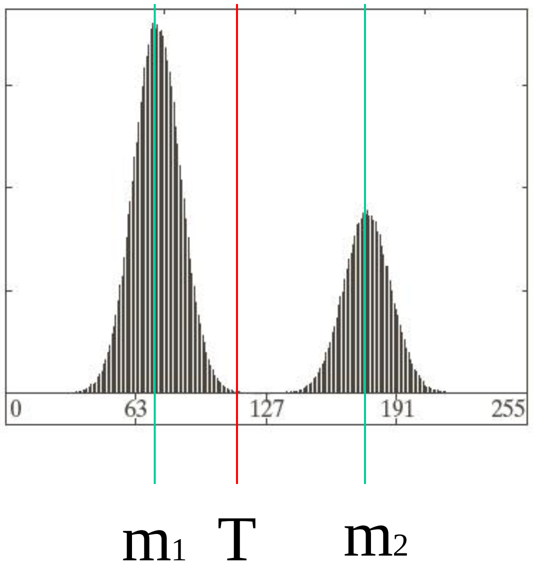


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

How to Pick the Threshold

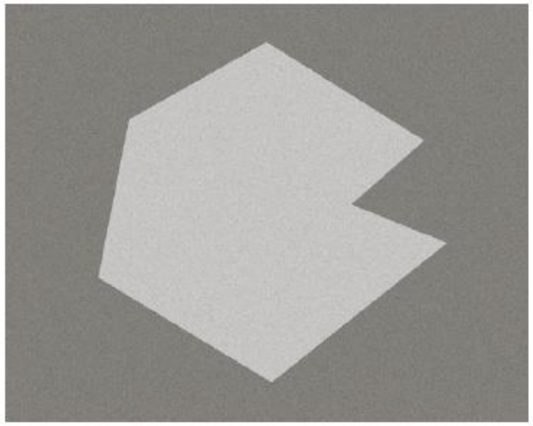
1. Select an initial estimate for the global threshold, *T*.
2. Segment the image using *T* by producing two groups of pixels
3. Compute the mean of these two groups of pixels, say m_1 and m_2.
4. Update the threshold T=(m_1 +m_2)/2
5. Repeat Steps 2 through 4 until convergence

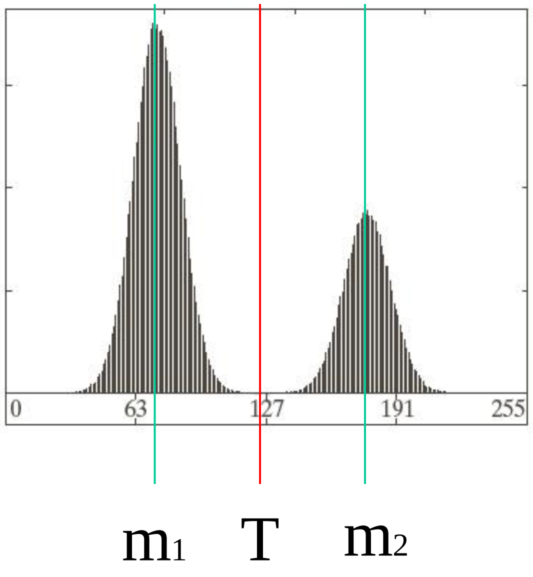




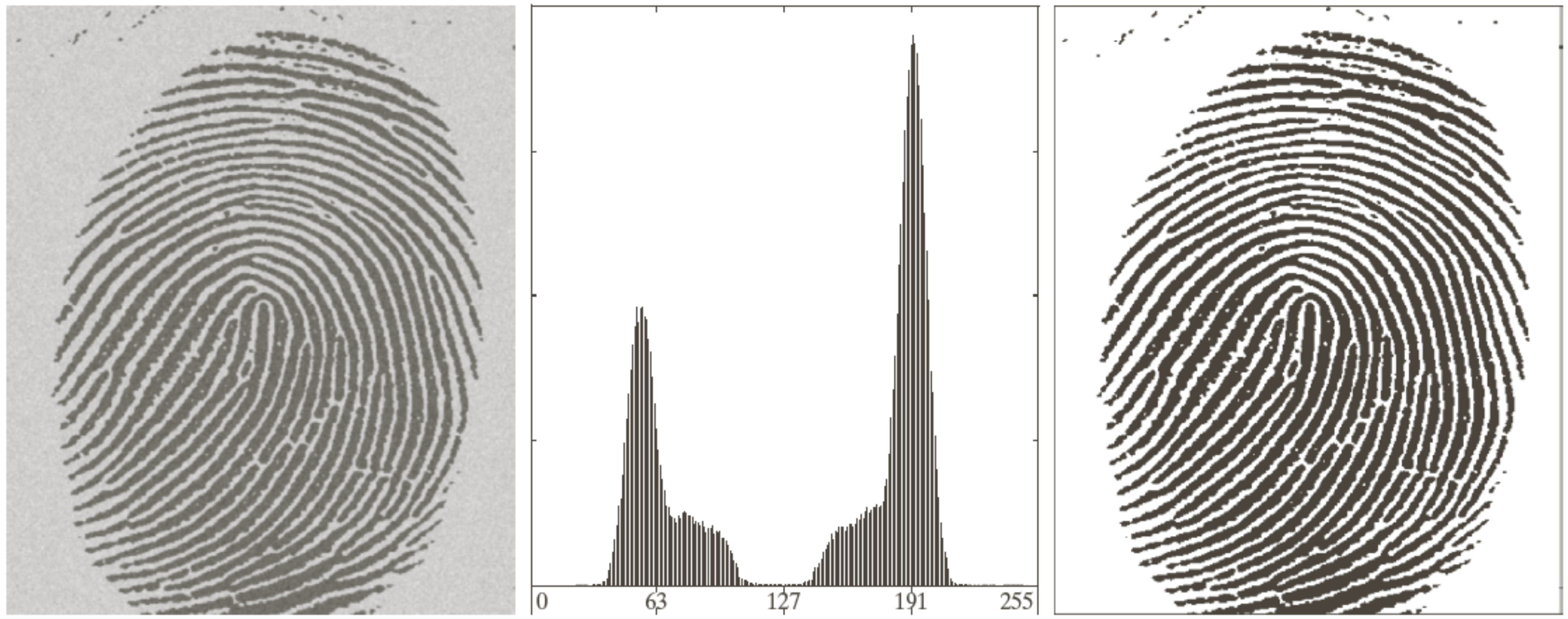
How to Pick the Threshold

1. Select an initial estimate for the global threshold, *T*.
2. Segment the image using *T* by producing two groups of pixels
3. Compute the mean of these two groups of pixels, say m_1 and m_2.
4. Update the threshold $T=(m_1 + m_2)/2$
5. Repeat Steps 2 through 4 until convergence





An Example



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.)

Region-Based Segmentation

- Region growing
- Region splitting and merging

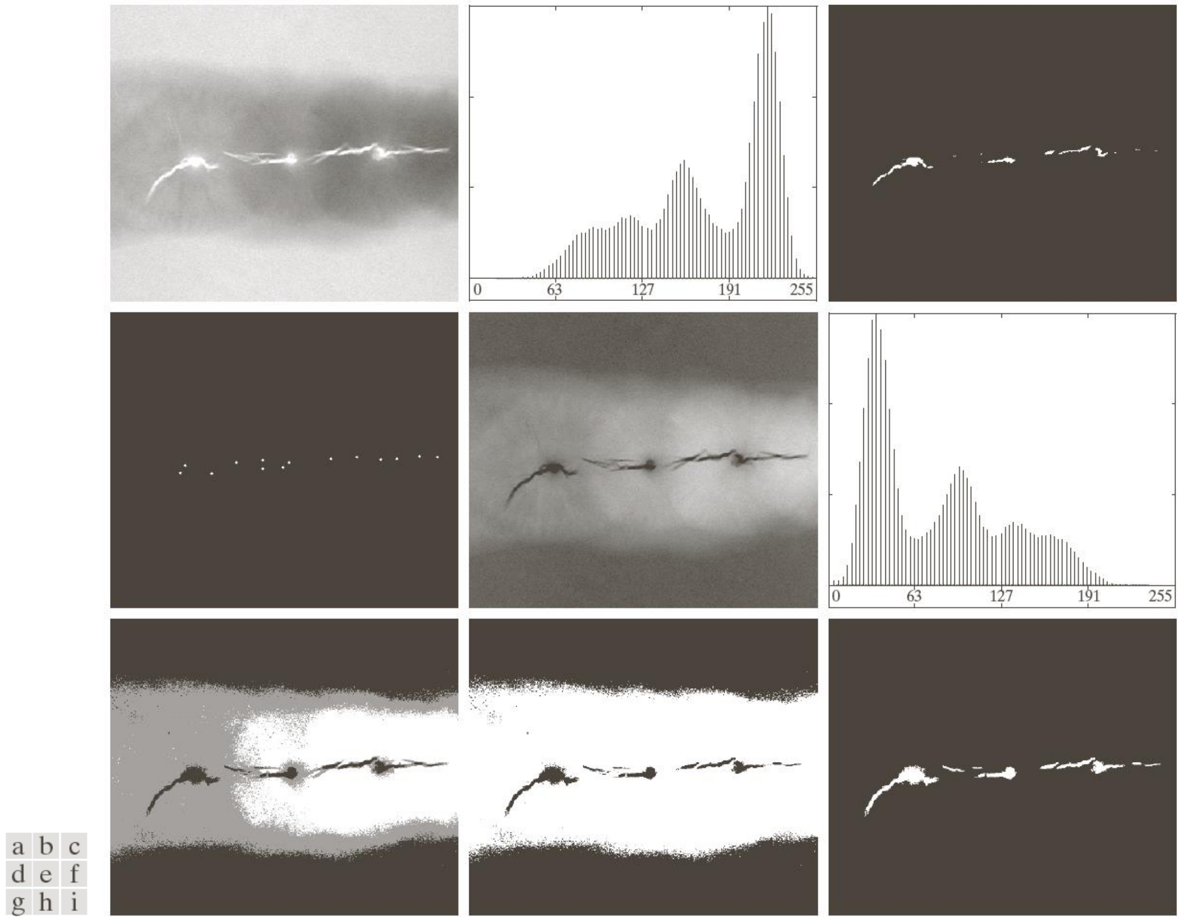
Region Growing Algorithm

-A procedure that groups pixels or subregions into larger regions based on predefined criteria for growth

-**Start with a set of “seed” points and grow regions by appending neighboring pixels that satisfy the given criteria**

- Connectivity
- Stopping rules
  - Local criteria: intensity values, textures, color
  - Prior knowledge: size and shape of the object

An Example



$$Q = \begin{cases} True & if\ |I_{seed} - I(x,y)| \le T \\ False & otherwise \end{cases}$$

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

Region-Splitting and Merging Algorithm

Step1: Keep splitting the region while $Q(R_i) = FALSE$ and $R_i > \min Size$

Step 2: Merge the subregions while $Q(R_i \cup R_j) = TRUE$

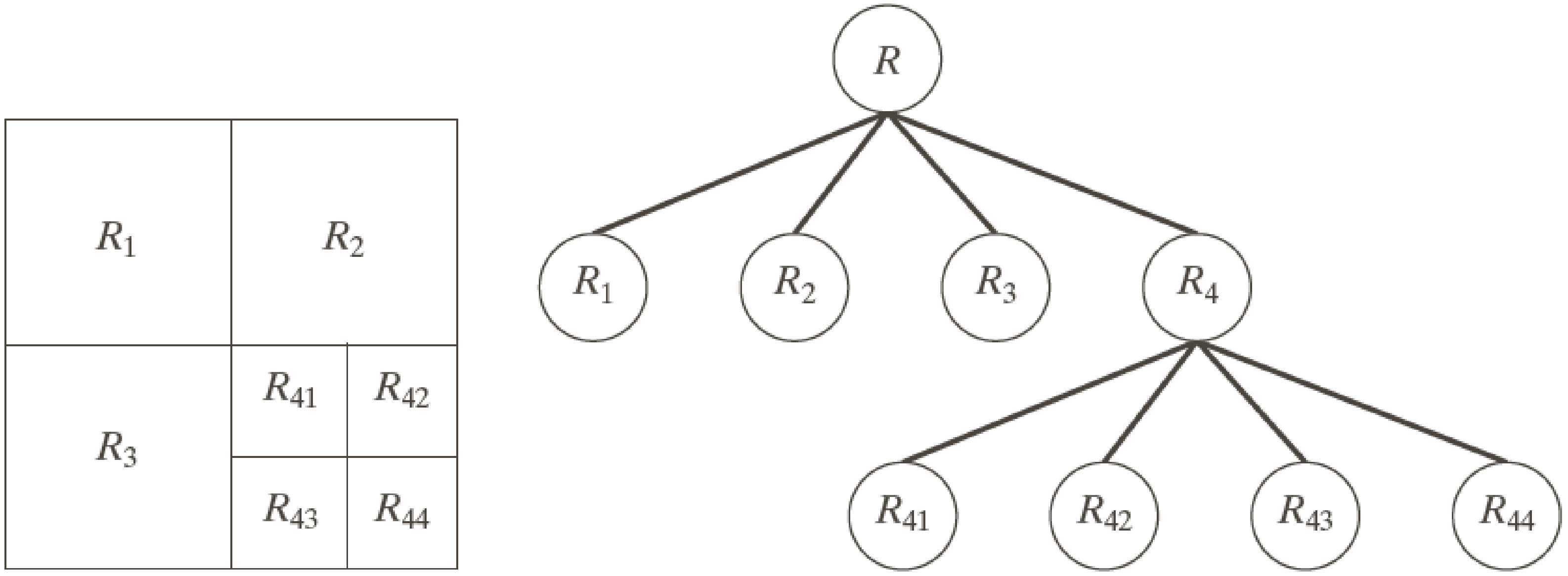


FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. *R* represents the entire image region.

An Example

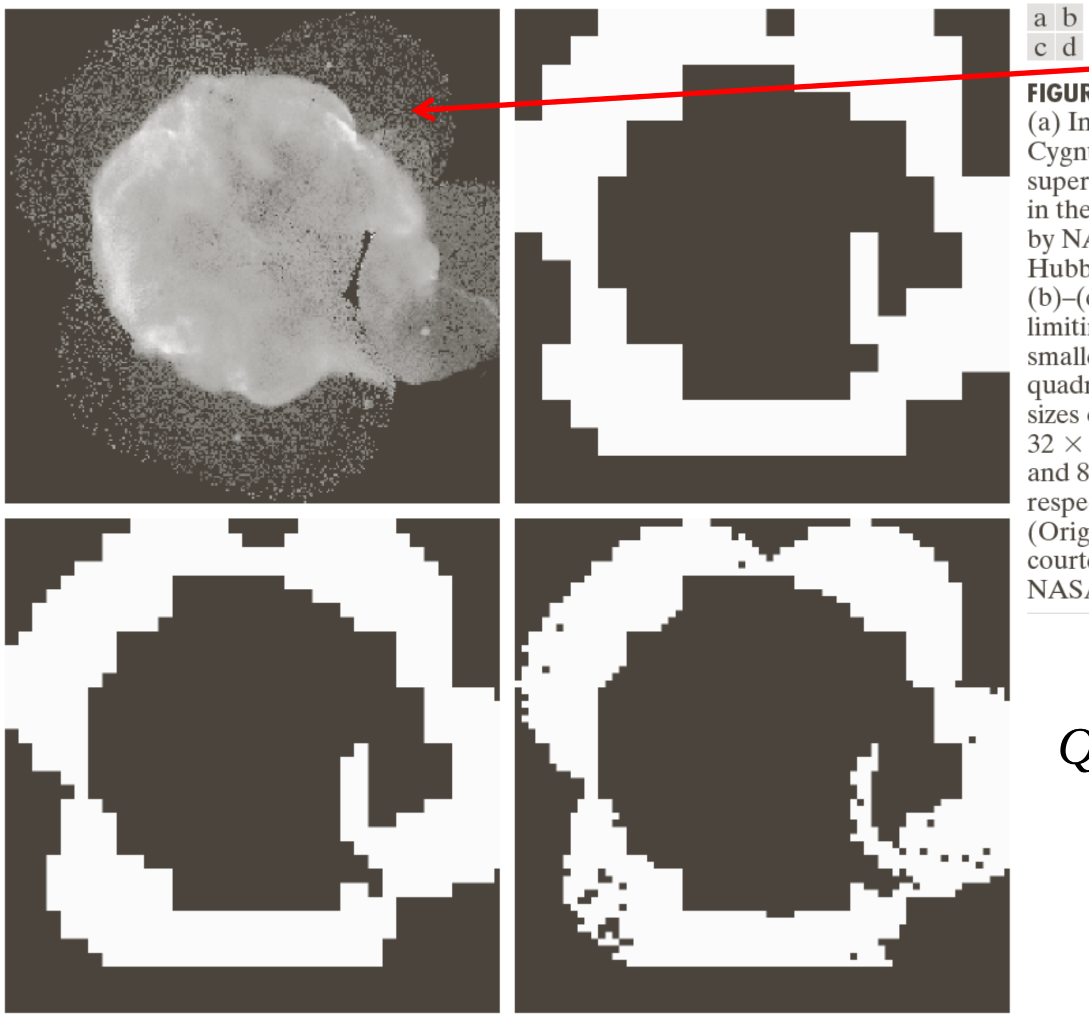


FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32 × 32, 16 × 16, and 8 × 8 pixels, respectively. (Original image courtesy of NASA.)

Extract the outer ring

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

Additional Notes and Readings

Image segmentation is a fundamental problem in image processing and computer vision

There are huge number of algorithms available for different applications

General-purpose image segmentation is far from well solved

It is still a research problem that is being investigated by many researchers

Deep-learning-based Tools and resources:

Image Segmentation with Deep Learning (Guide) - viso.ai

Segment Anything | Meta AI (segment-anything.com)

Paper tables with annotated results for Segment Everything Everywhere All at Once | Papers With Code