Lecture 14 – Edge Detection（边缘检测）

This lecture will cover:

– Image segmentation (图像分割)

- Properties of Intensity values
- Derivatives and filtering

– Point, Line and Edge Detection（点、线和边缘检测）

- Point Detection
- Line Detection
- Edge Detectors
- Edge Linking

Segmentation

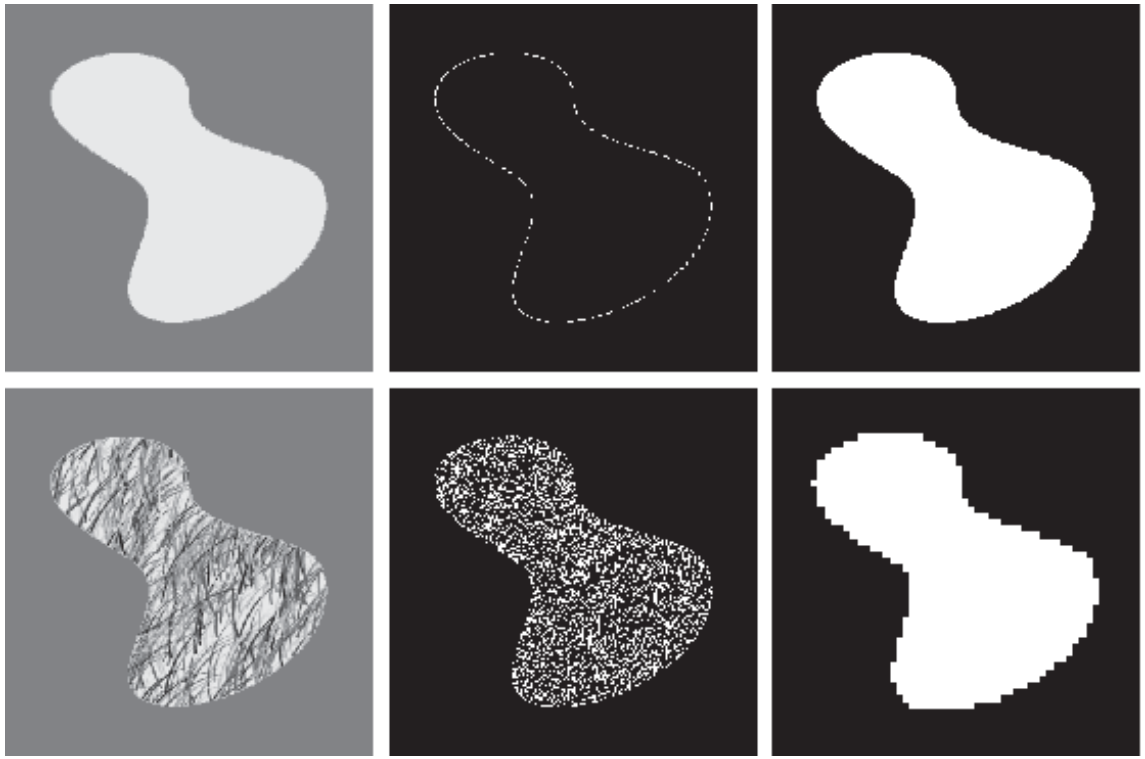
- **Subdivide an image into its constituent regions or object**
  - Determine the level of subdivision details by the problem to be solved
  - Stop the subdivision when the regions or objects have been detected
- **Improve segmentation accuracy during acquisition**
  - Environment control
  - Sensor selection
  - Imaging modality
- **Considered as a process that partitions region R into n subregions $R_1, R_2, \cdots, R_n$, such**
  a) $\cup_{i=1}^{n}(R_i) = R$
  b) R_i is a connected set, $i = 1,2,\cdots,n$
  c) $R_i \cap R_j = \emptyset$ for all i and j, $i \neq j$
  d) $Q(R_i) = True$ for $i = 1,2,\cdots,n$
  e) $Q(R_i \cup R_j) = False$ for any adjacent region R_i and R_j

Properties of Intensity values

- **Discontinuity** – Edge-based segmentation
- **Similarity** – Region-based segmentation

a b c
d e f

FIGURE 10.1
(a) Image of a constant intensity region.
(b) Boundary based on intensity discontinuities.
(c) Result of segmentation.
(d) Image of a texture region.
(e) Result of intensity discontinuity computations (note the large number of small edges).
(f) Result of segmentation based on region properties.



Edge detection

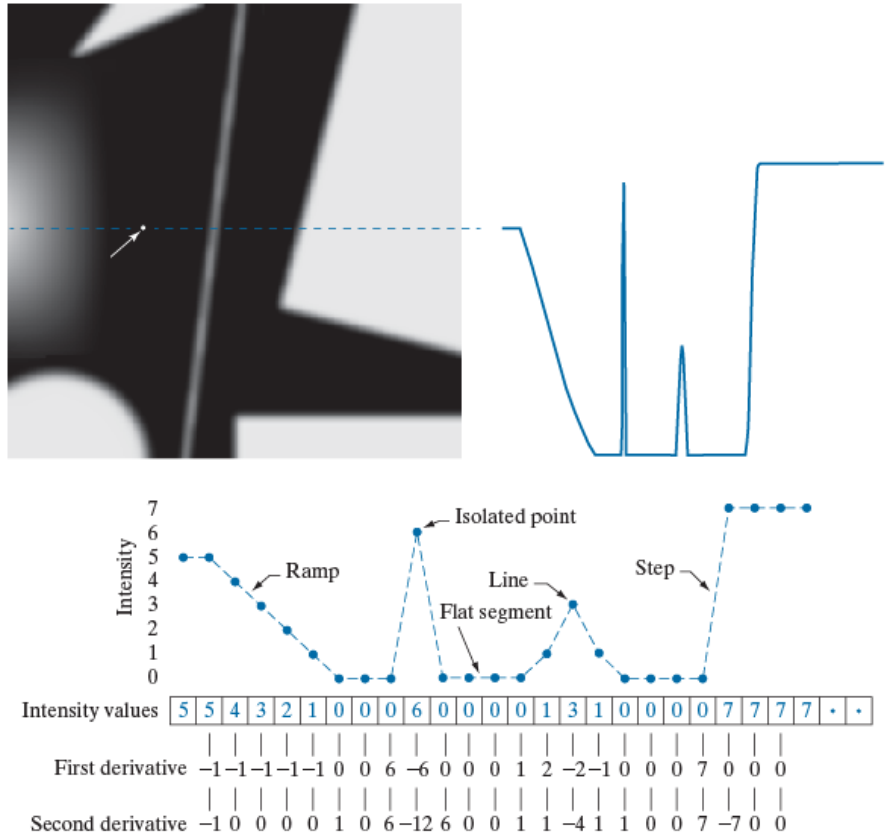
- **Derivatives**
  1. Zero in area of constant intensity
  2. Nonzero at the onset of intensity step or ramp
  3. (1) Nonzero along intensity ramp – 1st order derivative

     (2) Zero along intensity ramp with constant slope – 2nd order derivative

- **Vector Operation**

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{mn} z_{mn} = \sum_{k=1}^{mn} w_k z_k = w^T z$$

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



a b
c

FIGURE 10.2
(a) Image.
(b) Horizontal intensity profile that includes the isolated point indicated by the arrow.
(c) Subsampled profile; the dashes were added for clarity. The numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10-4) for the first derivative and Eq. (10-7) for the second.

FIGURE 10.3
A general 3×3 spatial filter kernel. The w's are the kernel coefficients (weights).

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Point Detection (点检测)

- **Output expression:**

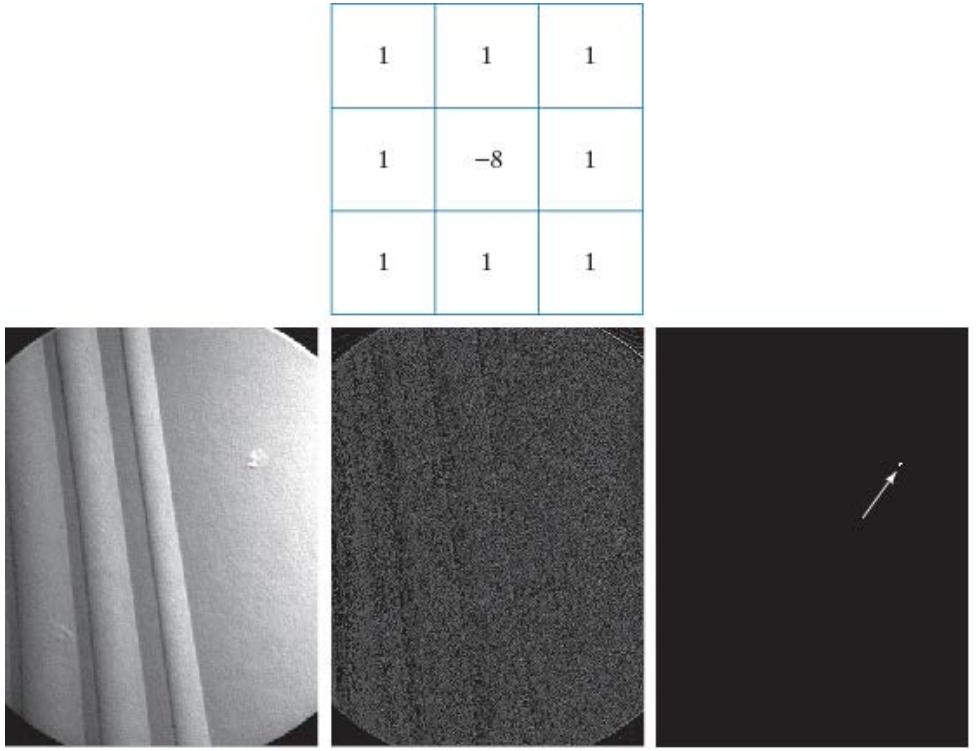
$$g(x,y) = \begin{cases} 1, & |R(x,y)| \geq T \\ 0, & \text{otherwise} \end{cases}$$

Where

g: output image

T: nonnegative threshold

R: the filtered image



a
b c d

FIGURE 10.4 (a) Laplacian kernel used for point detection. (b) X-ray image of a turbine blade with a porosity manifested by a single black pixel. (c) Result of convolving the kernel with the image. (d) Result of using Eq. (10-15) was a single point (shown enlarged at the tip of the arrow). (Original image courtesy of X-TEK Systems, Ltd.)

- **Matlab function:**

```
g = abs(imfilter(double(f)), w))>=T
g = ordfilt2(f,m*n,ones(m,n))-ordfilt2(f,1,ones(m,n)); g=g>=T
```

Line Detection (线检测)

- **Output expression**

$$|R_i| > |R_j|, \qquad j \neq i$$

Where R_i is the response to the masks

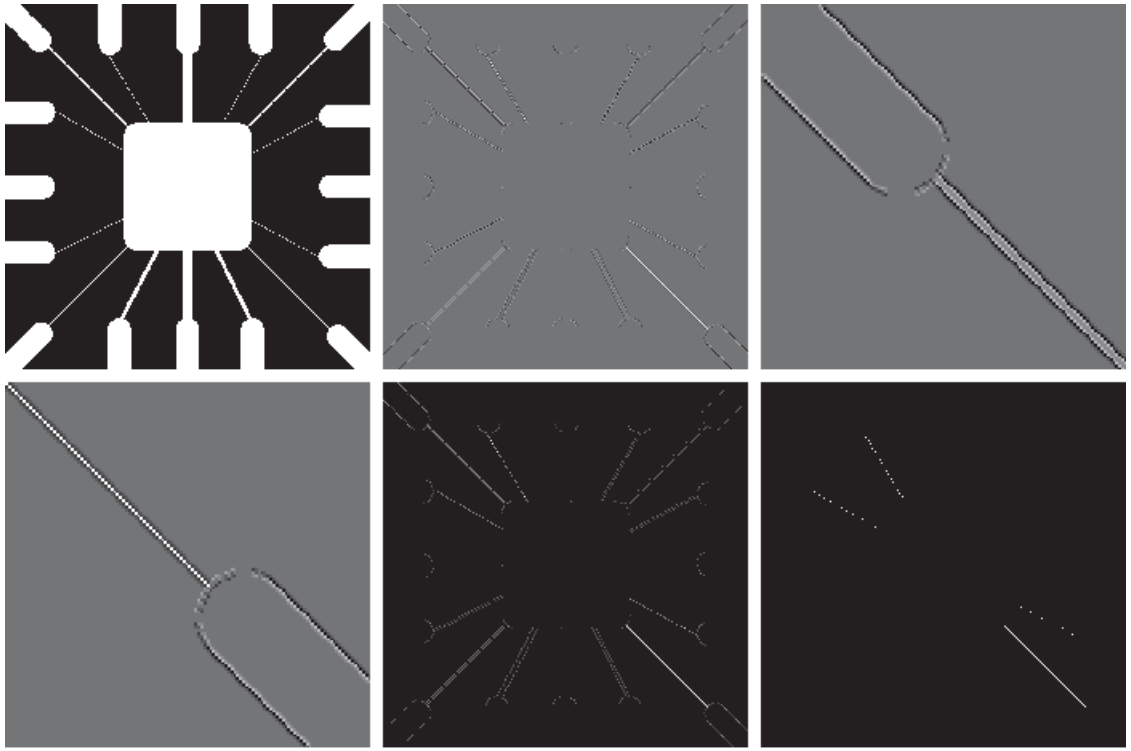
- **Matlab function**

g = abs(imfilter(double(f)), w))

−1	−1	−1	2	−1	−1	−1	2	−1	−1	−1	2
2	2	2	−1	2	−1	−1	2	−1	−1	2	−1
−1	−1	−1	−1	−1	2	−1	2	−1	2	−1	−1
Horizontal			+45°			Vertical			−45°		

a b c d

FIGURE 10.6 Line detection kernels. Detection angles are with respect to the axis system in Fig. 2.19, with positive angles measured counterclockwise with respect to the (vertical) x-axis.



a b c
d e f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the +45° line detector kernel in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g > T$, where g is the image in (e) and $T = 254$ (the maximum pixel value in the image minus 1). (The points in (f) were enlarged to make them easier to see.)

Edge Detection (边缘检测)

➢ Rules

- Seek the 1st derivative greater than a threshold
- Seek the zero-crossing point on the 2nd derivative

➢ Steps

1. Image smoothing for noise reduction
2. Detection of edge points
3. Edge localization

➢ Matlab function

[g, t] = edge(f, 'method', parameters);

Edge Detector	Description
Sobel	Finds edges using the Sobel approximation to the derivatives in Fig. 10.5(b)
Prewitt	Finds edges using the Prewitt approximation to the derivatives in Fig. 10.5(c).
Roberts	Finds edges using the Roberts approximation to the derivatives in Fig. 10.5(d).
Laplacian of a Gaussian (LoG)	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a Laplacian of a Gaussian filter.
Zero crossings	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a specified filter.
Canny	Finds edges by looking for local maxima of the gradient of $f(x, y)$. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges.

Edge Detectors (边缘检测器)

a b c d e f g

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

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0	0	−1
0	1	1	0

Roberts

−1	−1	−1	−1	0	1
0	0	0	−1	0	1
1	1	1	−1	0	1

Prewitt

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

Sobel

- **The first derivative detectors**

$$M(x,y) \approx |g_x| + |g_y|$$

- **Matlab function:** [g, t] = edge(f, method, T, dir)

where method is sobel/prewitt/roberts

−3	−3	5	−3	5	5	5	5	5	5	5	−3
−3	0	5	−3	0	5	−3	0	−3	5	0	−3
−3	−3	5	−3	−3	−3	−3	−3	−3	−3	−3	−3
N			NW			W			SW		

5	−3	−3	−3	−3	−3	−3	−3	−3	−3	−3	−3
5	0	−3	5	0	−3	−3	0	−3	−3	0	5
5	−3	−3	5	5	−3	5	5	5	−3	5	5
S			SE			E			NE		

a b c d e f g h

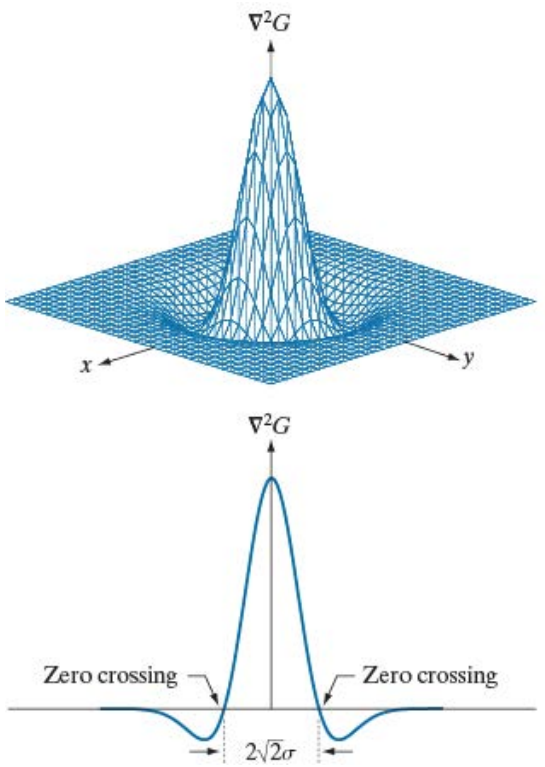
FIGURE 10.15
Kirsch compass kernels. The edge direction of strongest response of each kernel is labeled below it.

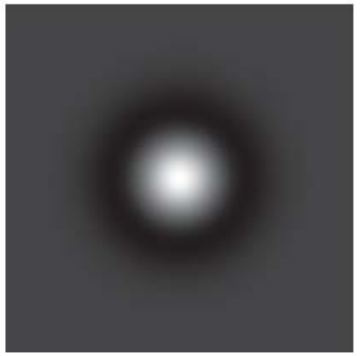
Edge Detectors (边缘检测器)

- **LoG (Laplacian of a Gaussian, 高斯拉普拉斯算子):**

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

1. Filter the input image with a Gaussian lowpass kernel;
2. Compute the Laplacian of the image from Step 1;
3. Find the zero crossing of the image from Step 2.





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) 3-D 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 kernel approximation to the shape in (a). The negative of this kernel would be used in practice.

- **Matlab function:**
  - [g, t] = edge(f, 'log', T, sigma)
  - [g, t] = edge(f, 'zerocross', T, H)

Edge Detectors (边缘检测器)

➢ **Canny Detector (坎尼边缘检测器):**

- **Basic objectives**
  1. Low error rate
  2. Edge points should be well localized
  3. Single edge point response
- **Step**
  1. Smooth the input image with a Gaussian filter;
  2. Compute the gradient magnitude image $M(x, y)$ and angle image $\alpha(x, y)$;

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

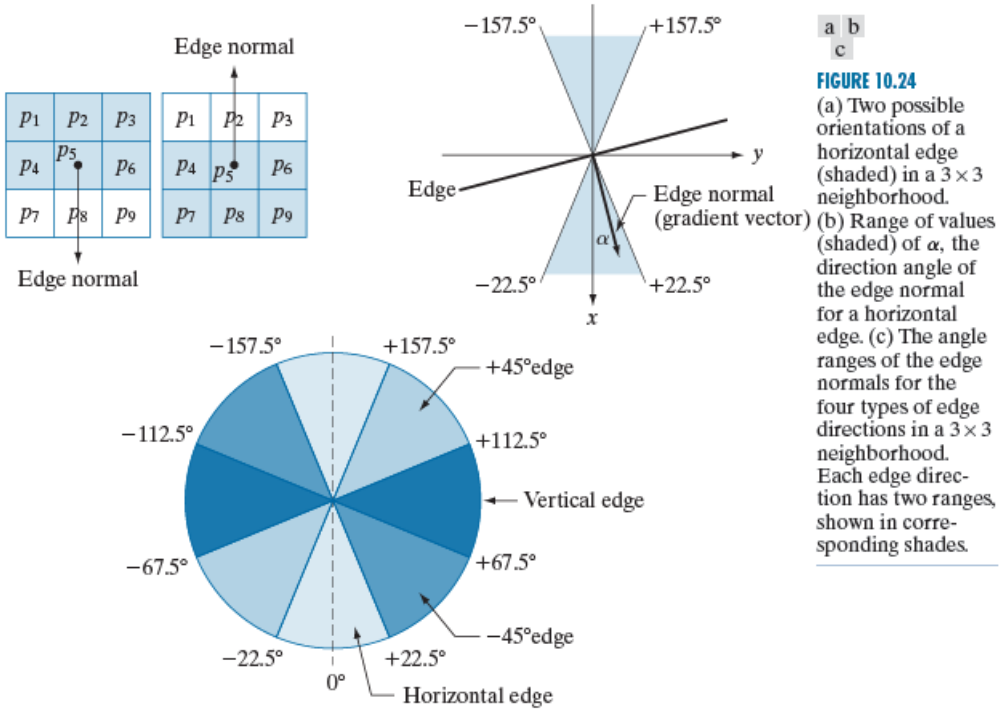
  3. Apply nonmaxima suppression (非最大值抑制) to the gradient magnitude image;
  4. Use double thresholding and connectivity analysis to detect and link edge.
- **Matlab function:** [g, t] = edge(f, 'canny', T, sigma), where T=[T1, T2]

Canny Edge Detector

- **Nonmaxima suppression (非最大值抑制)**

1. Find the direction d_k that is closest to $\alpha(x,y)$, where d_k denote 4 basic edge directions;

2. $$M(x,y) = \begin{cases} M(x,y), & \text{if greater than 2 neighbor points along } d_k \\ 0, & \text{otherwise} \end{cases}$$



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (shaded) in a 3 × 3 neighborhood. (b) Range of values (shaded) 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.

- **Double thresholding and connectivity analysis**

$$g_{NH}(x,y) = g_N(x,y) \geq T_H \qquad g_{NL}(x,y) = g_N(x,y) \geq T_L$$

where T_H is high threshold and T_L is low threshold

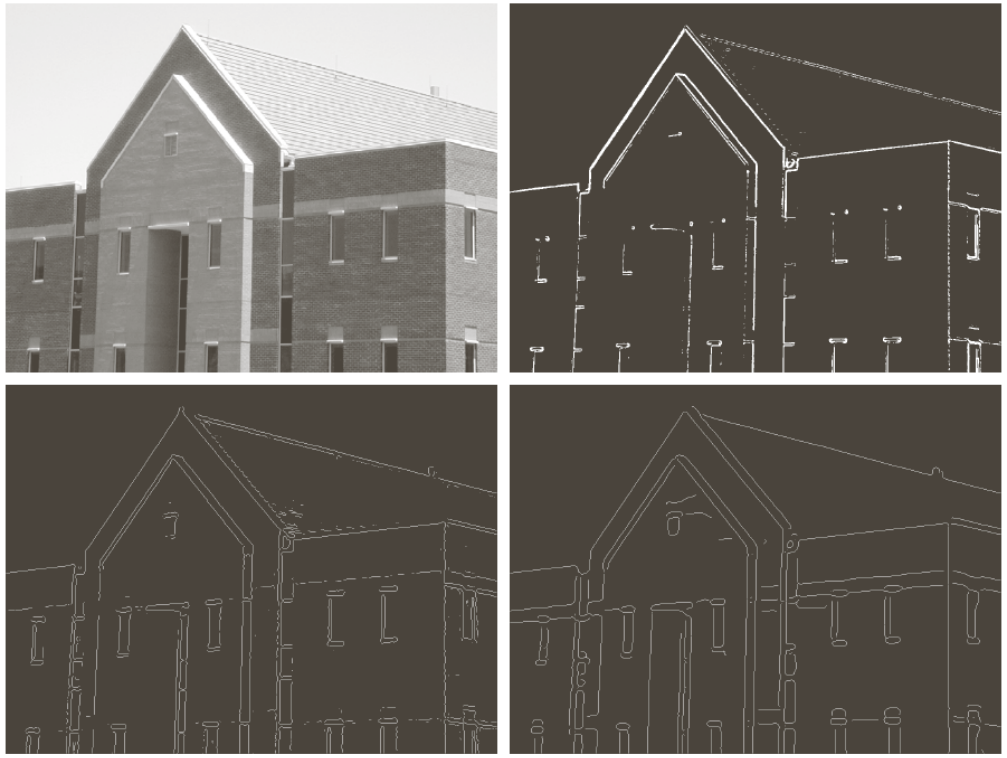
$$g_{NL}(x,y) = g_{NL}(x,y) - g_{NH}(x,y)$$

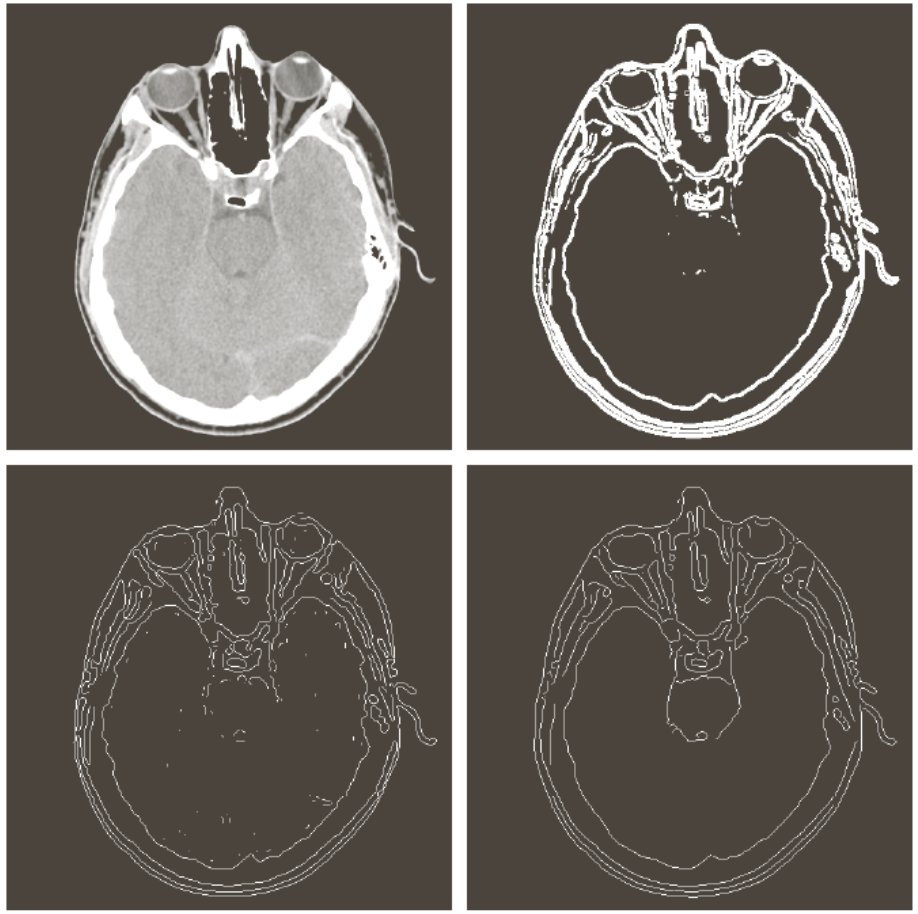
1. Locate the next unvisited edge pixel p in $g_{NH}(x,y)$;
2. Mark as valid edge pixels all the weak pixels in $g_{NL}(x,y)$ that are connected to p;
3. If all nonzero pixels in $g_{NH}(x,y)$ have been visited, go to Step 4; else, return to Step 1;
4. Set to zero all pixels in $g_{NL}(x,y)$ that are not marked as valid edge pixels.

Edge Detectors (边缘检测器)

a b
c d

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.





a b
c d

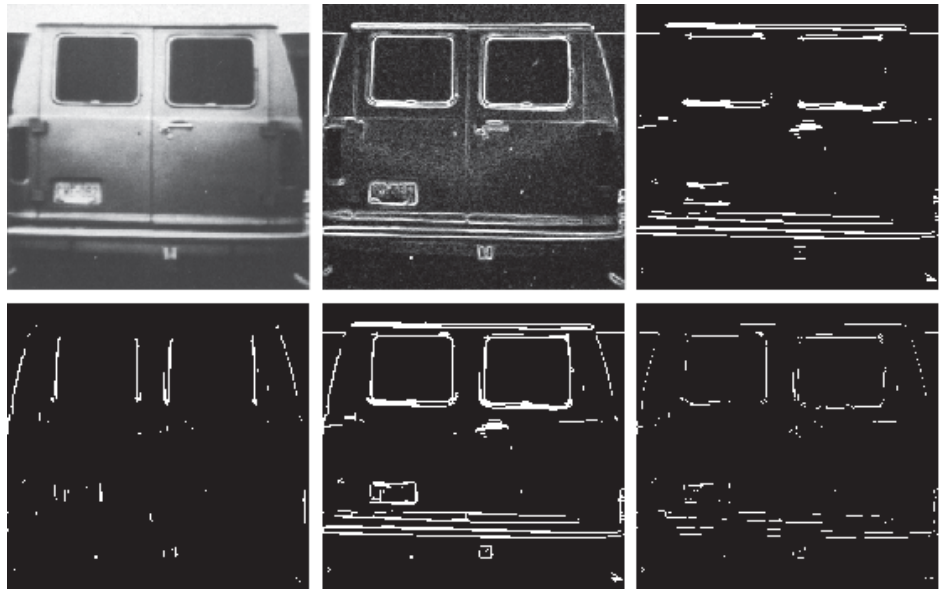
FIGURE 10.26
(a) Original head CT image of size 512 × 512 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R.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

Edge Linking (边缘连接)

➢ **Local processing (局部处理)**

- Based on the characteristics of pixels in a small neighborhood about every point (x, y);
- Two principal properties
  1. The magnitude of gradient vector
     $$|M(s,t) - M(x,y)| \leq E$$
  2. The direction of gradient vector
     $$|\alpha(s,t) - \alpha(x,y)| \leq A$$
- Simplified steps
  1. Compute $M(x, y)$ and $\alpha(x, y)$;
  2. Form a binary image $g(x, y)$, where

$$g(x,y) = \begin{cases} 1, & \text{if } M(x,y) > T_M \text{ AND } \alpha(x,y) = A \pm T_A \\ 0, & \text{otherwise} \end{cases}$$



a b c
d e f

FIGURE 10.27
(a) 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 (c) and (d). (f) Final result, using morphological thinning. (Original image courtesy of Perceptics Corporation.)

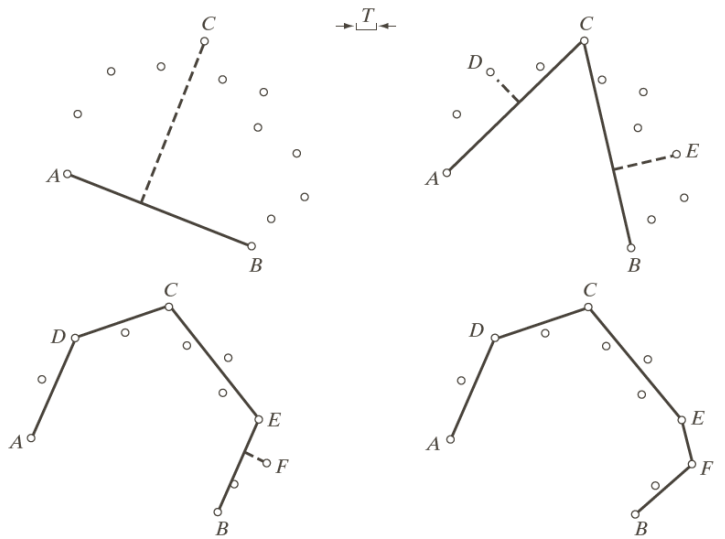
  3. Scan the rows of g and fill (set to 1) all gaps in each row that do not exceed a specified length L;
  4. To detect gaps in any other direction θ, rotate g by θ and apply Step 3, and rotate the result back by $-\theta$

Edge Linking (边缘连接)

- **Regional processing (区域处理)**

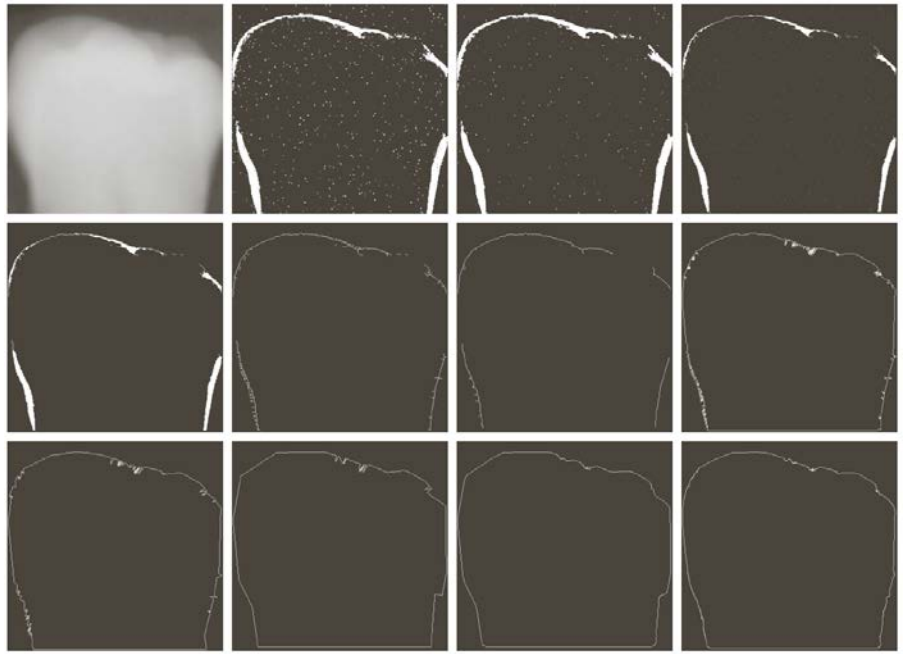
  Linking pixels on a regional basis for an approximate boundary of the region;

  - Polygonal approximation
  - The algorithm for finding a polygonal fit to open and closed curve.



a b
c d

FIGURE 10.28 Illustration of the iterative polygonal fit algorithm.



a b c d
e f g h
i j k l

FIGURE 10.30 (a) A 550 × 566 X-ray image of a human tooth. (b) Gradient image. (c) Result of majority filtering. (d) Result of morphological shrinking. (e) Result of morphological cleaning. (f) Skeleton. (g) Spur reduction. (h)–(j) Polygonal fit using thresholds of approximately 0.5%, 1%, and 2% of image width ($T = 3$, 6, and 12). (k) Boundary in (j) smoothed with a 1-D averaging filter of size 1 × 31 (approximately 5% of image width). (l) Boundary in (h) smoothed with the same filter.

Edge Linking (边缘连接)

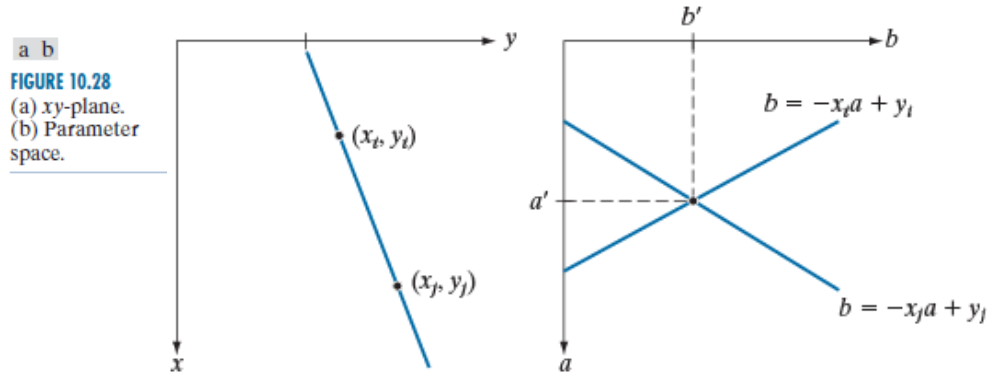
➢ Hough Transform (霍夫变换)

- Global processing - based on predefined global properties
- An approach based on Hough Transform
  1. Obtain a binary edge image using any edge detector;
  2. Specify subdivisions in the ρθ-plane;
  3. Examine the counts of the accumulator cells (累加器单元) for high pixel concentrations;
  4. Examine the relationship between pixels in a chosen cell.
- Matlab function:

  ≫ [H, theta, rho] = hough(f);

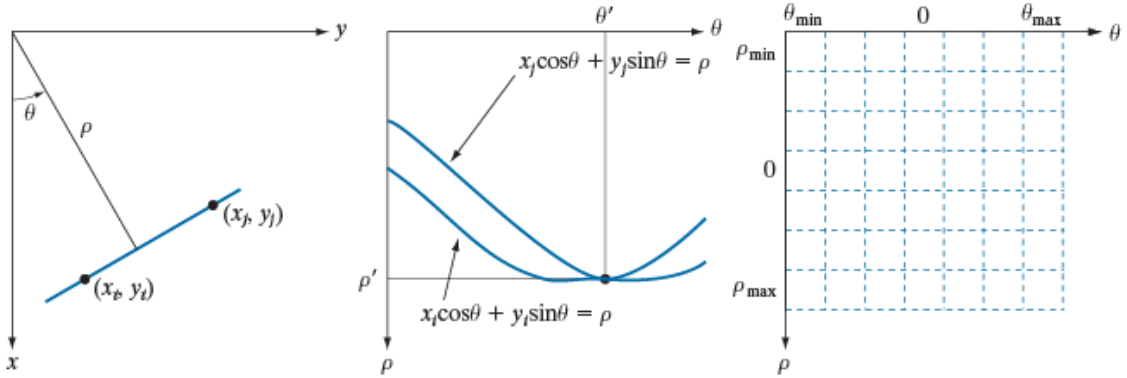
  ≫ peaks = houghpeaks(H, NumPeaks);

  ≫ lines = houghlines(f, theta, rho, peaks);



a b

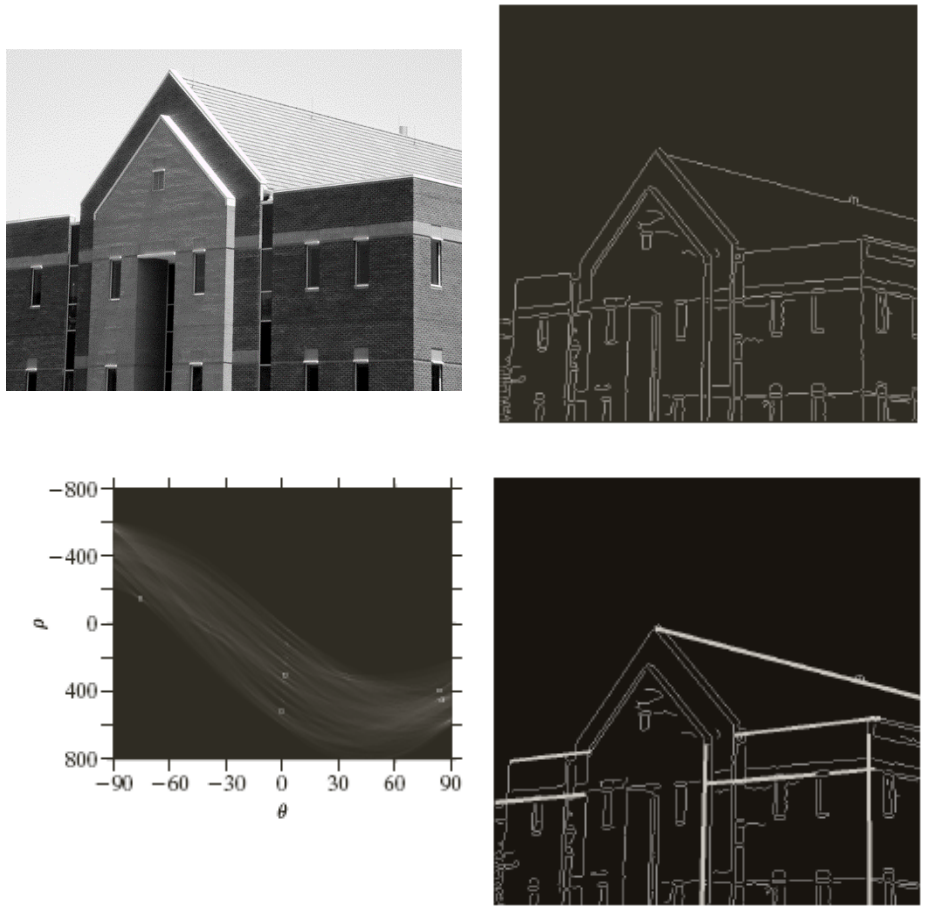
FIGURE 10.28 (a) *xy*-plane. (b) Parameter space.



a b c

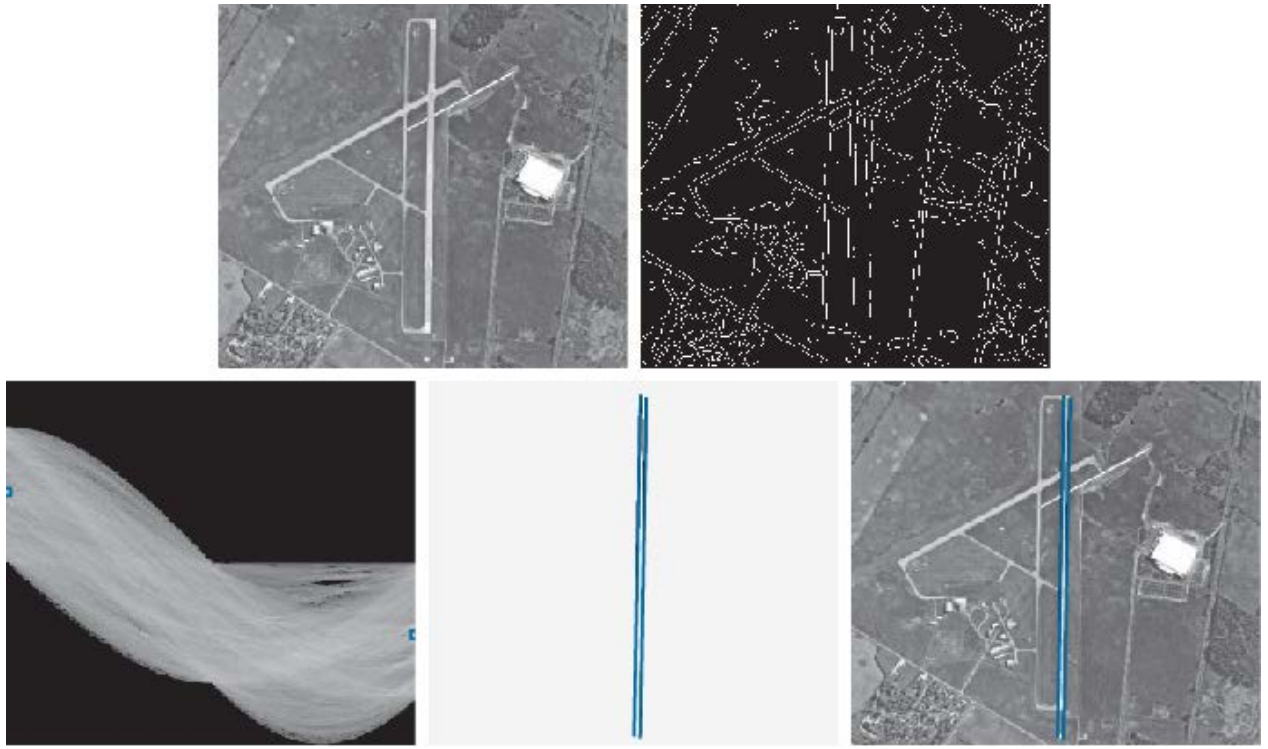
FIGURE 10.29 (a) (ρ,θ) parameterization of a line in the *xy*-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ',θ') corresponds to the line passing through points (x_i,y_i) and (x_j,y_j) in the *xy*-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.

Hough Transform (霍夫变换)



a b
c d

Fig. (a) Original image; (b) Result of Canny edge detector; (c) Hough transform with 5 peaks locations selected (d) Line segmentation (in bold) corresponding to the Hough transform peaks.



a b
c d e

FIGURE 10.31 (a) A 502×564 aerial image of an airport. (b) Edge map 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.