Image Segmentation I

Hongkai Xiong
Department of Electronic Engineering
Shanghai Jiao Tong University

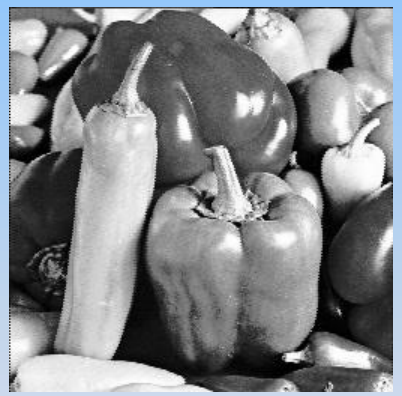
2016

Preview

- Segmentation is to subdivide an image into its component regions or objects.
- Segmentation should stop when the objects of interest in an application have been isolated.

Preview

- Example



Preview

- Segmentation algorithms generally are based on one of 2 basis properties of intensity values
  - ***discontinuity*** : to partition an image based on sharp changes in intensity (such as edges)
  - ***similarity*** : to partition an image into regions that are similar according to a set of predefined criteria.

Topic

Edge Detection

- Gradient-based methods
- Canny edge detector
- Hough transform

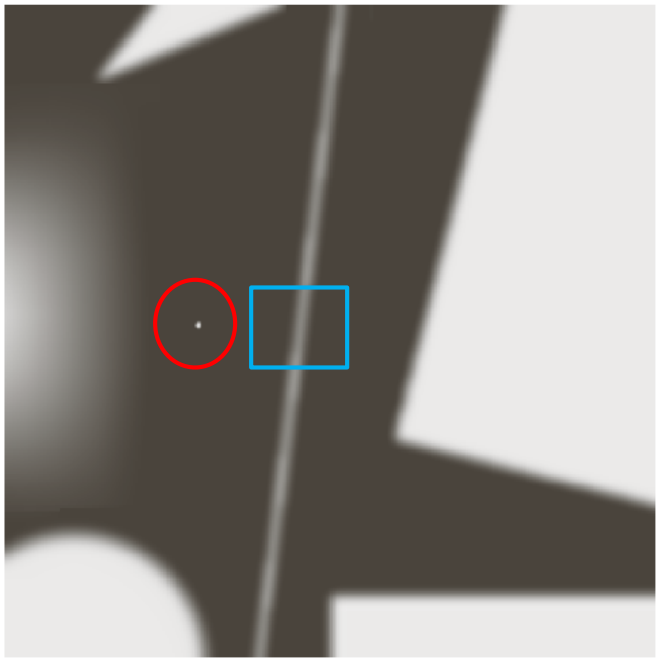
Detection of Discontinuities

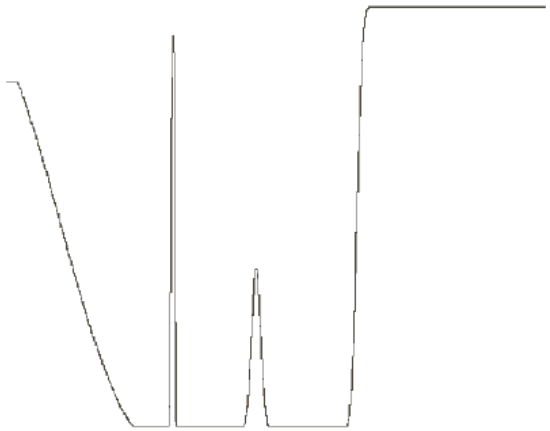
- Detect the three basic types of gray-level discontinuities

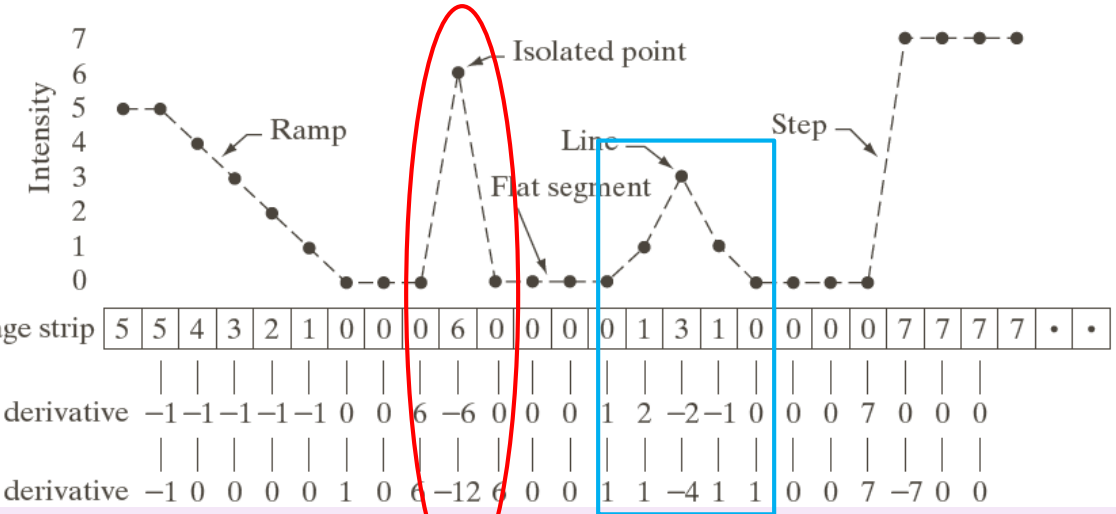
  points , lines , edges

Detection of Discontinuities

- What are their characteristics of points, lines, and edges ?






Intensity
7
6
5
4
3
2
1
0
Ramp
Isolated point
Line
Flat segment
Step
Image strip
5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •
First derivative
−1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0
Second derivative
−1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

Detection of Discontinuities

- We can arrive at the following conclusions:

1. 1-st order derivatives usually produce **thicker edges**;

2. 2-nd order derivatives have a **stronger** response to fine details, such as thin lines, isolated points, and noise.

3. 2-nd order derivatives produce **a double-edge** response at ramp and step transitions in intensity.

−1	−1	−1
−1	8	−1
−1	−1	−1

Point Detection

A point can be detected at the location where has a stronger 2-nd derivatives, which can be implemented by convolving with the mark and having response:

$$|R| \geq T$$

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.

Line Detection

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

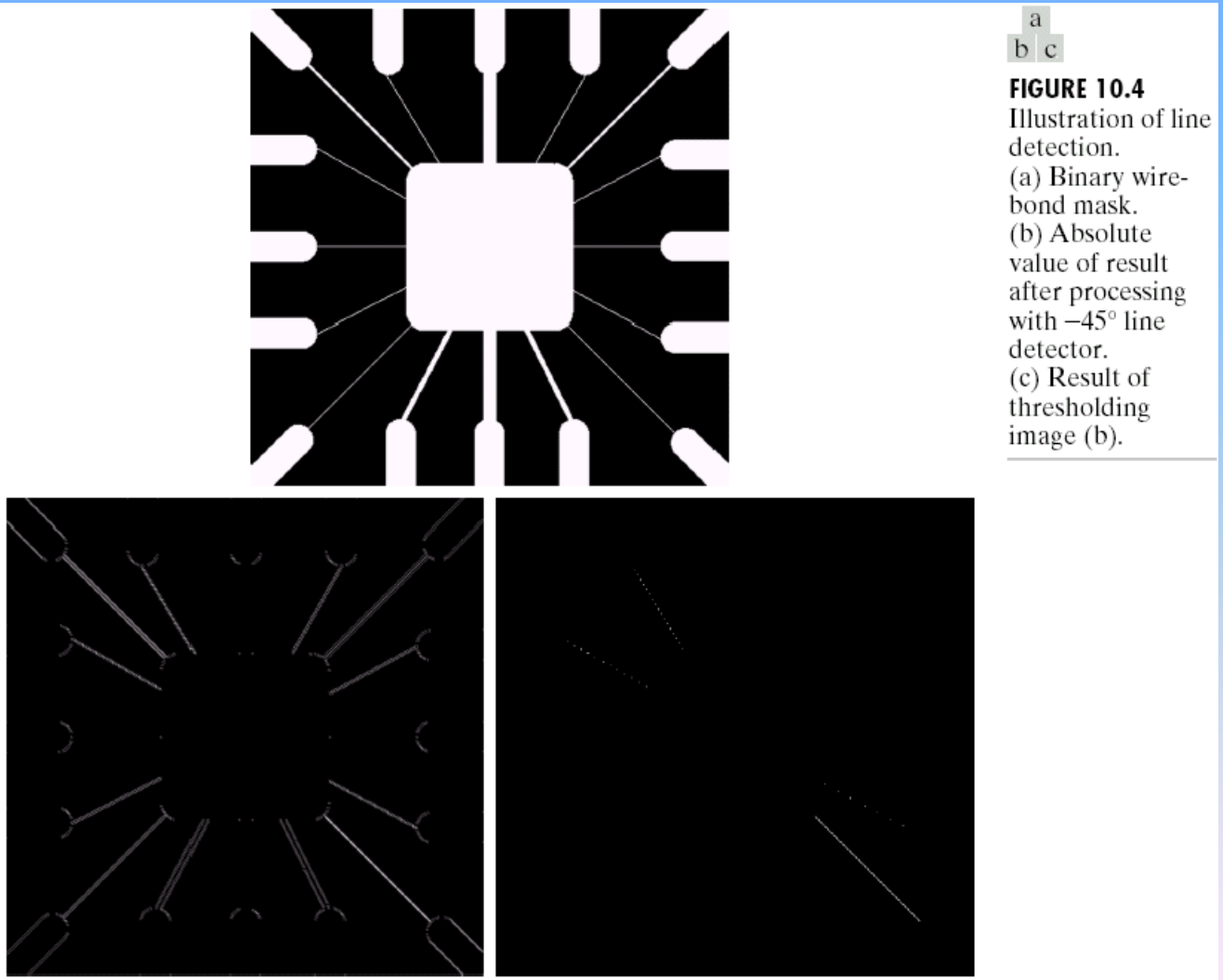
- Horizontal mask will result with max response when a line passed through the middle row of the mask with a constant background.
- the similar idea is used with other masks.
- note: the preferred direction of each mask is weighted with a larger coefficient (i.e.,2) than other possible directions.

Line Detection

- If we are interested in detecting all lines in an image in one direction, we simply run corresponding 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.

Example

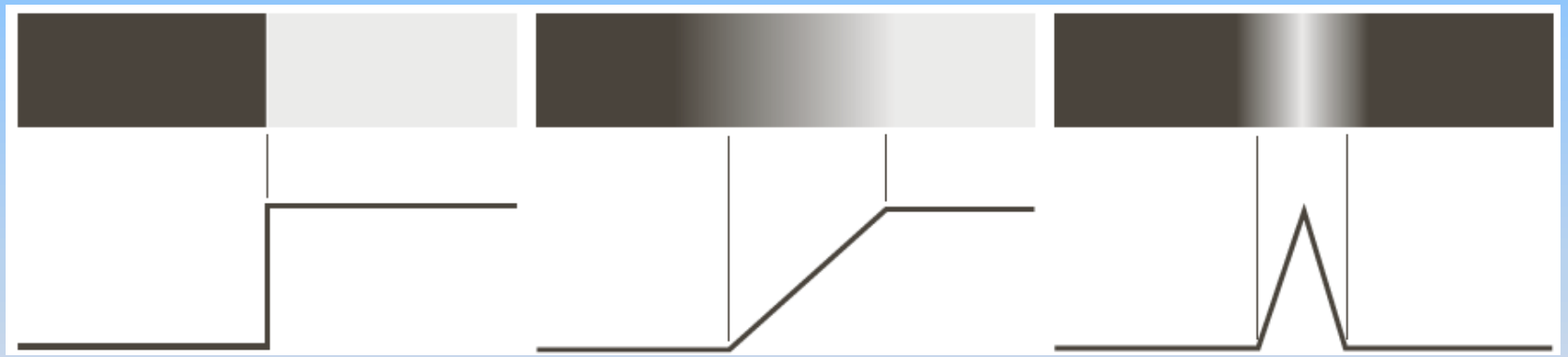


a
b c

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).

Focus on Edge

- What is an edge ?



a b c

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

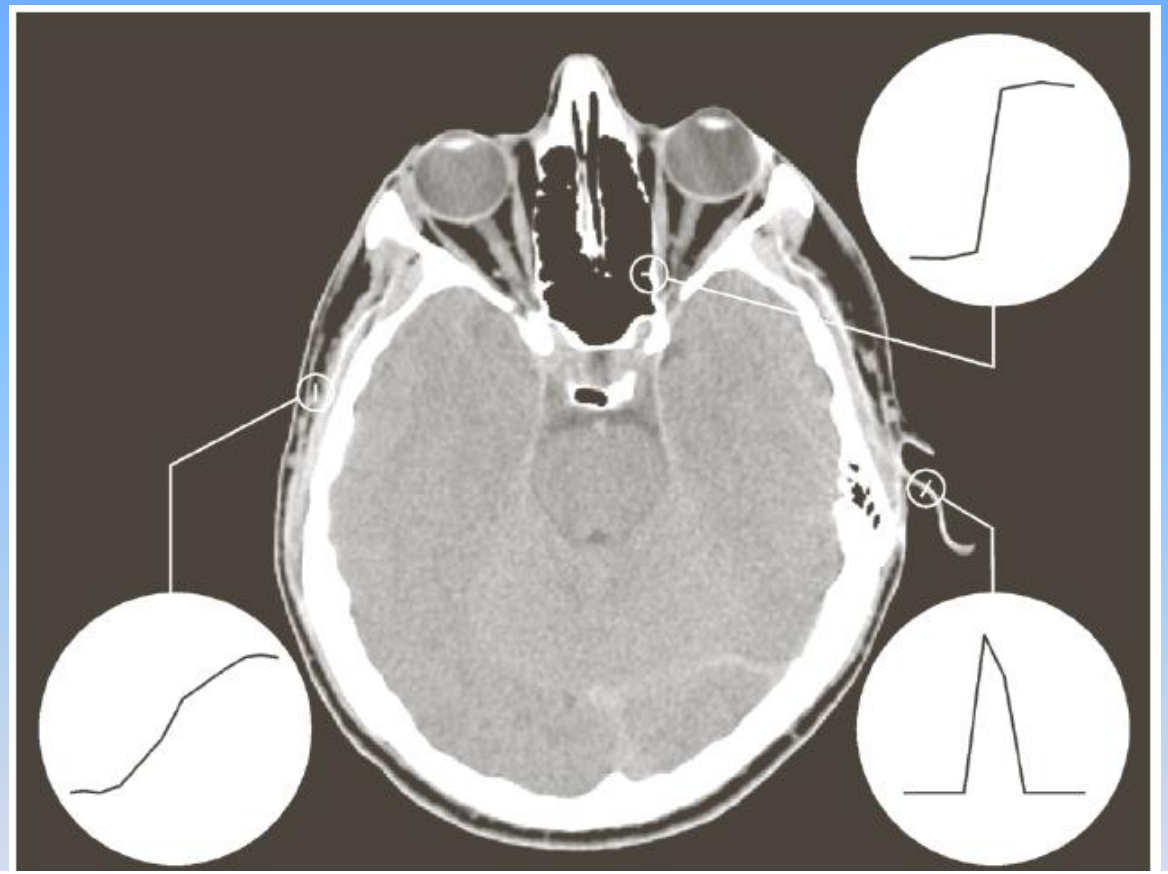
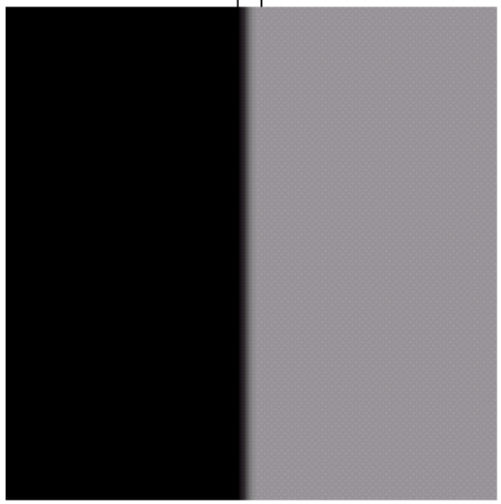


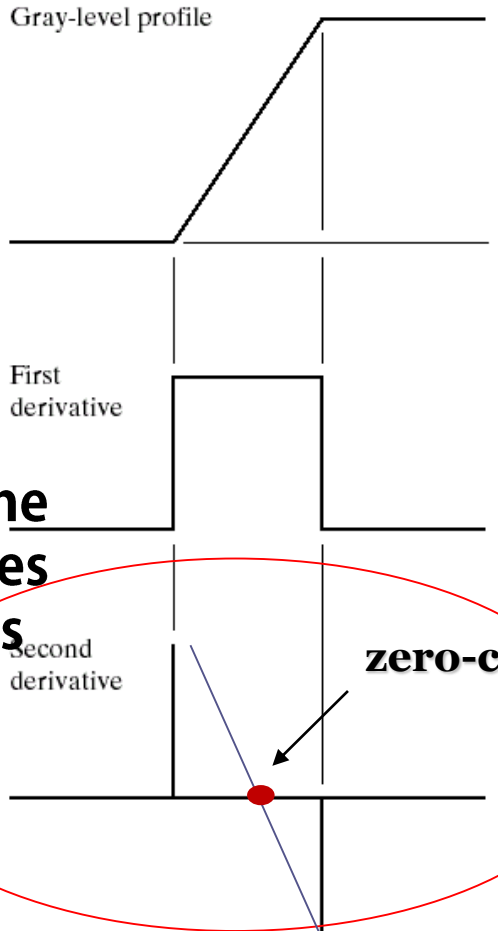
FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Character of an edge

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.





An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge

Review the conclusions:

- We can arrive at the following conclusions:

1. *1-st order derivatives usually produce* ***thicker edges****;*

2. 2-nd order derivatives have a **stronger** response to fine detail, such as thin lines, isolated points, and noise.

3. *2-nd order derivatives produce* ***a double-edge*** *response at ramp and step transitions in intensity.*

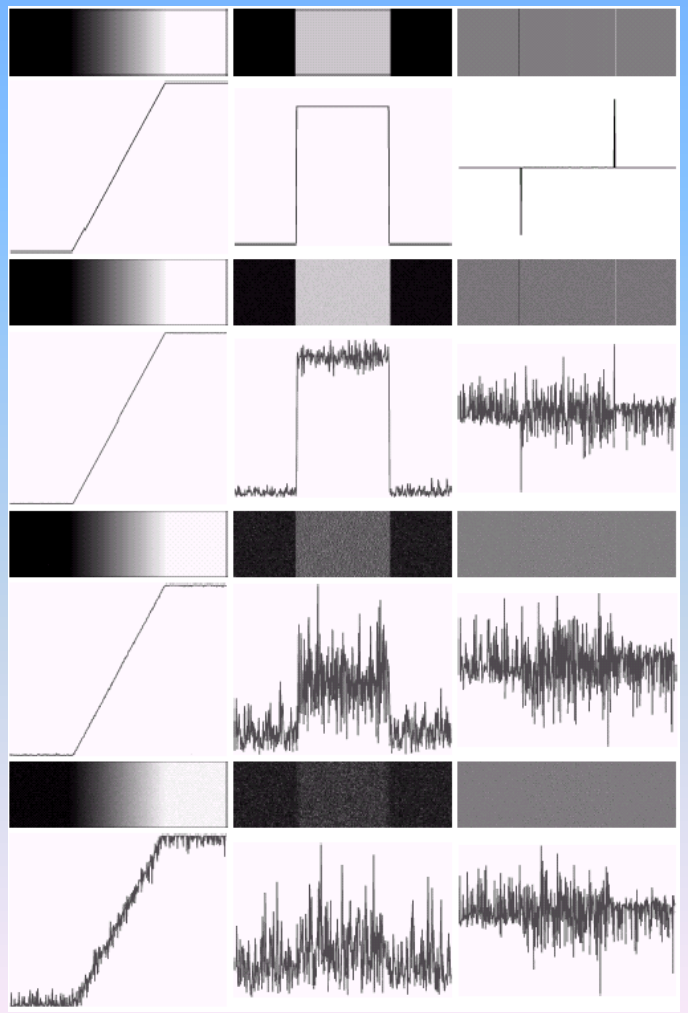
Edge detection

- We find a way to detect edges! Yeh!

Just use 1-st derivatives or zero-crossing points of 2-nd derivatives.

Really?

How about noisy image?



First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and σ = 0.0, 0.1, 1.0 and 10.0, respectively.

Notation:

Fairly little noise can have such a significant impact on the two key derivatives used for edge detection in images

Image smoothing is necessary!

Gradient-based edge detection

- Procedure:

1. ***Image smoothing for noise reduction.***
2. ***Detection of edge points.*** Using 1-st or 2-nd derivatives.
3. ***Edge localization.*** To select from the candidate edge points that are true members of edge set.

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

1-st order Gradient Operator

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

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

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

commonly approx.

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

the magnitude becomes nonlinear

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

Diagonal edges detect mask

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

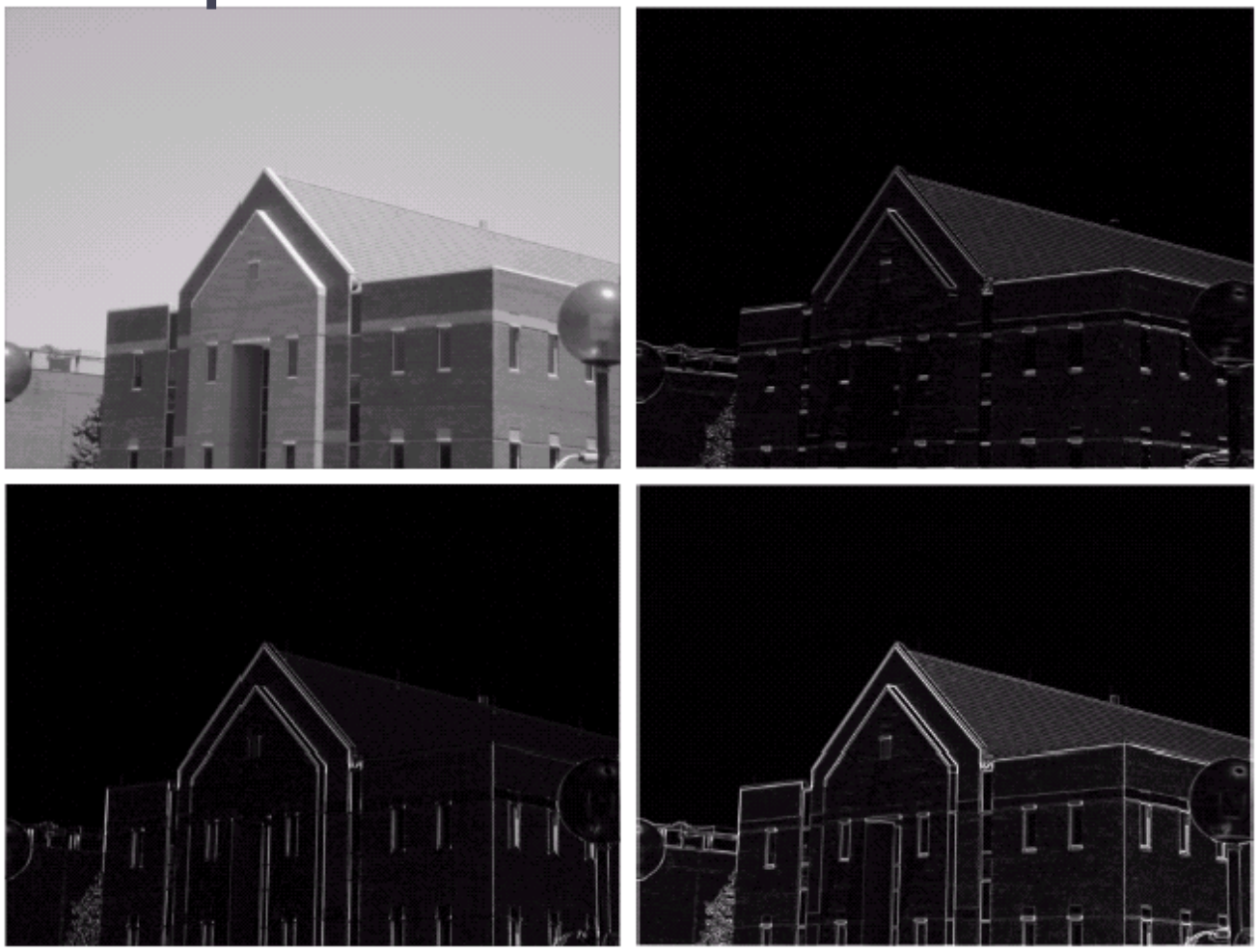
Sobel

Example

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, $|G_x| + |G_y|$.

Example



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.

Example

a b

FIGURE 10.12 Diagonal edge detection. (a) Result of using the mask in Fig. 10.9(c). (b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

2-nd order gradient operator

- Laplacian

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

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

2-nd order gradient operator

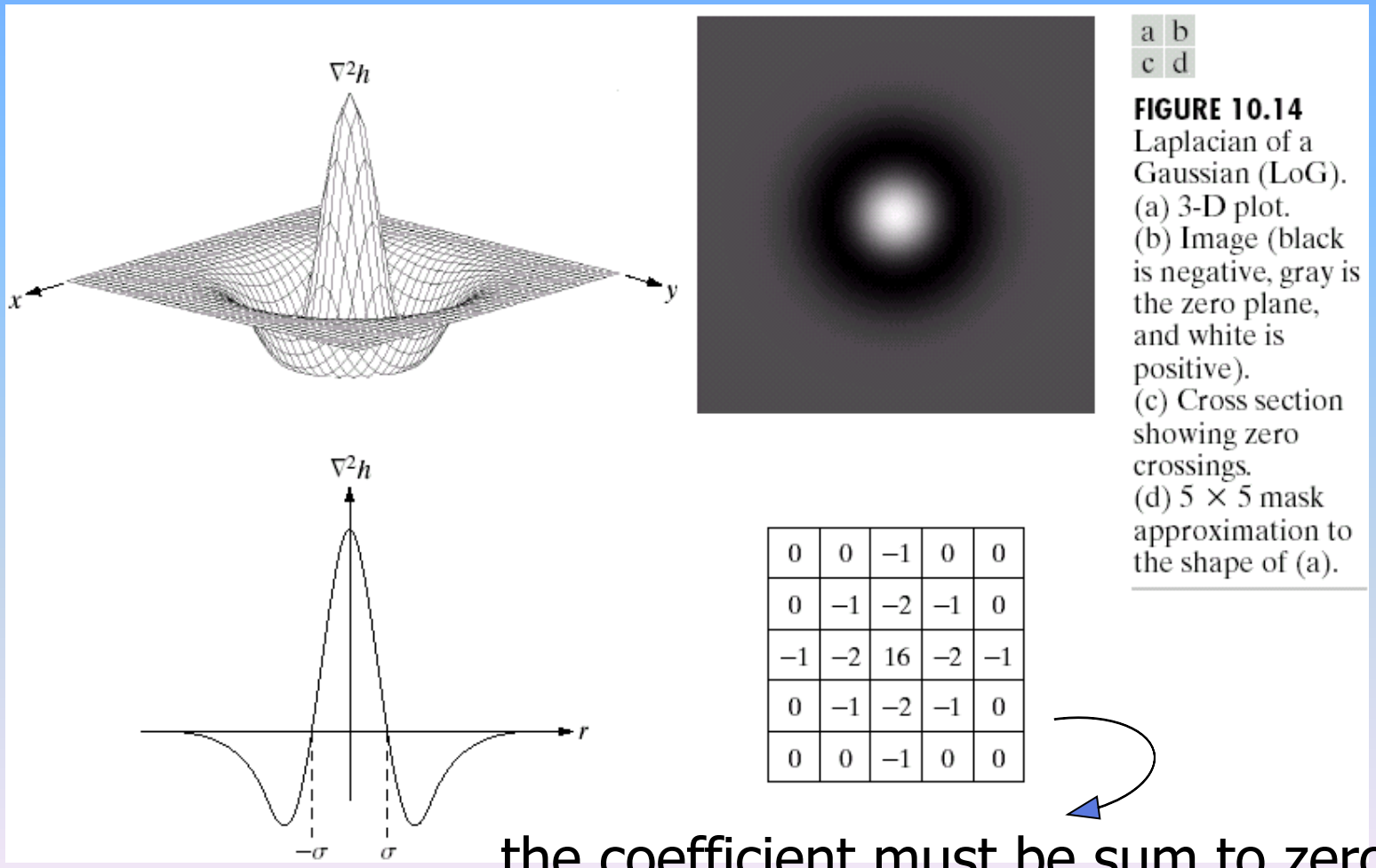
- Laplacian of Gaussian (LoG)

1.First using Gaussian function to smooth image, then applying Laplacian operator.

According to linear property, we can change the order.

$$\nabla^2(G(x,y) * I(x,y)) = (\nabla^2 G(x,y)) * I(x,y)$$

LoG



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

a b
c d

FIGURE 10.14 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

2-nd order gradient operator

- Laplacian of Gaussian (LoG)
  2. Calculate the zero-crossing points.
  3. Threshold.

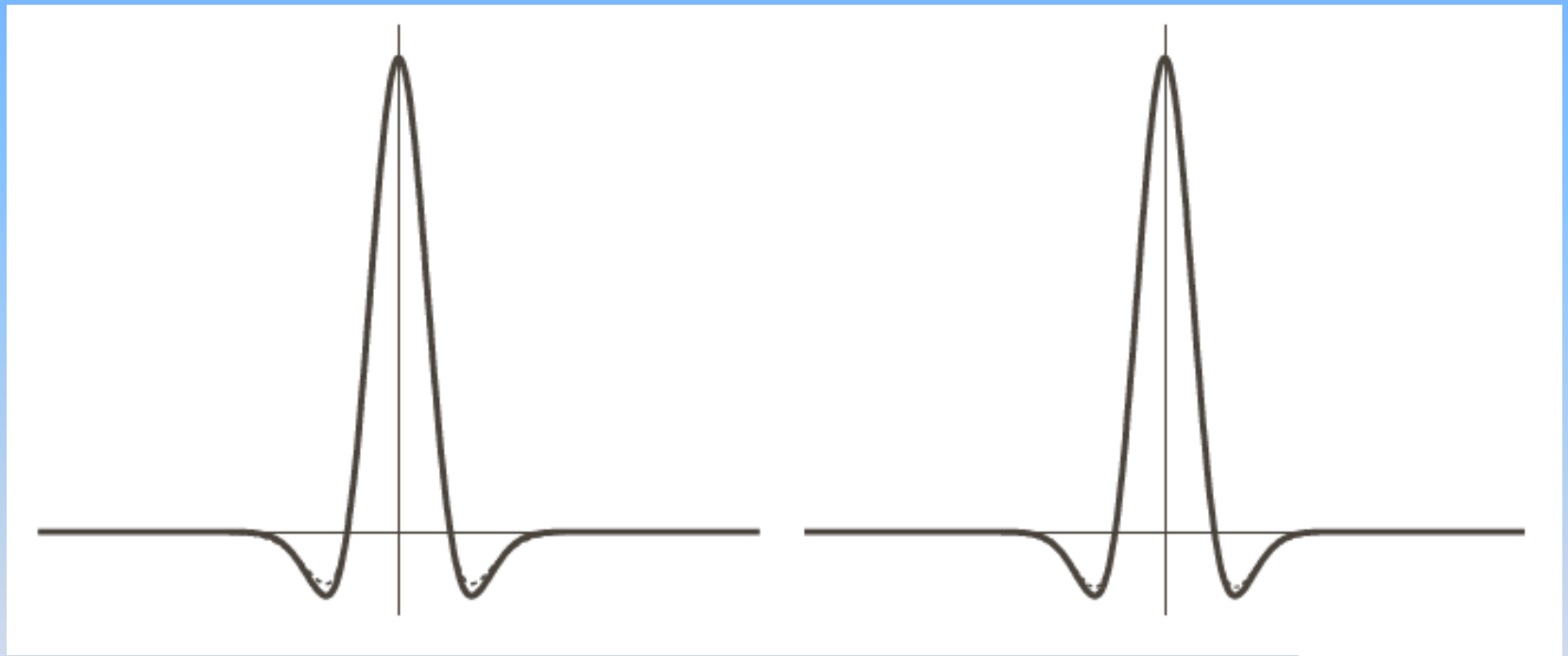
2-nd order gradient operator

- Difference of Gaussian (DoG)

LoG is complicated, we can approximate the LoG filtering by a DoG operator.

$$DoG(x, y) = \frac{1}{2\pi\sigma_1^2}\exp\left(-\frac{x^2 + y^2}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2}\exp\left(-\frac{x^2 + y^2}{2\sigma_2^2}\right)$$

2-nd order gradient operator



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 optimal one:**

So far, canny operator has been considered as the best of edge detector.

- **Four Steps:**

1. Image smoothing using Gaussian filter;

$$J = I * G$$

2. Calculating the magnitude M and direction D of J.

Canny Edge Detector

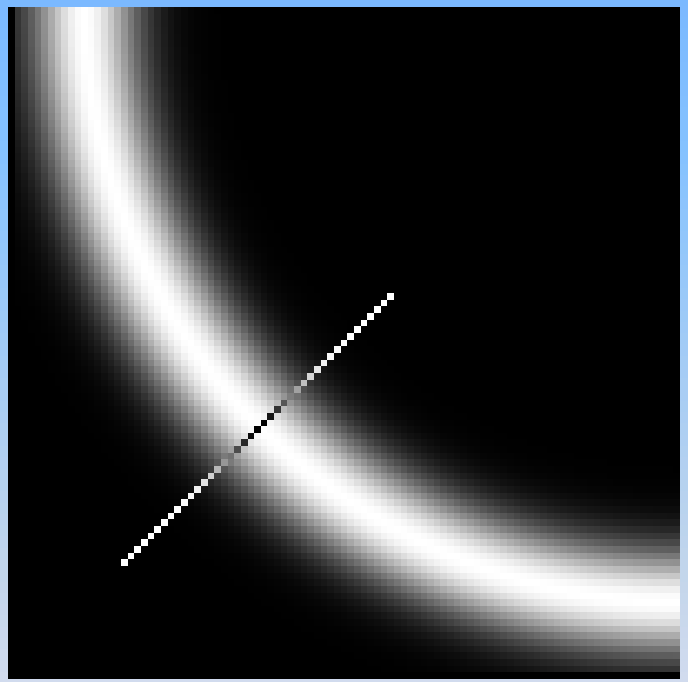
3. Non-maxima suppression

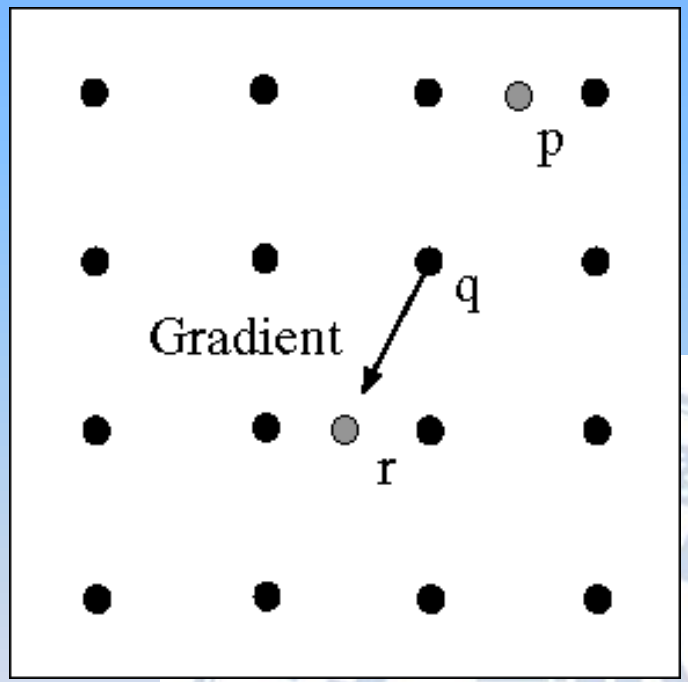
- Let d1, d2, d3, d4 denote four basic edge directions (0^0,45^0,90^0,135^0), then for each point (x,y) in D:

I. find direction dk which best approximates the direction of D(x,y).

II. Along the direction dk, check the two neighbors of pixel [x,y] in M. If M[x,y] is greater than both of its neighbors, set I [x,y] = M[x,y]; otherwise, I [x,y] = 0.

Canny Edge Detector





Check if pixel is local maximum along gradient direction

Canny Edge Detector

- Double threshold and edge link

Choose two thresholds Th and TL such that Th >TL， and threshold I with Th and TL respectively:

$$I_H(x,y) = I(x,y) \geq T_h$$
$$I_L(x,y) = I(x,y) \geq T_l$$

Then eliminate from I_L all the nonzero pixels from I_H :

$$I_L(x,y) = I_L(x,y) - I_H(x,y)$$

Canny Edge Detector

- Double threshold and edge link

After the thresholding operation, all strong pixels in I_H are assumed to be valid edge pixels and marked immediately. However, such pixels have gaps, we link edge as follows:

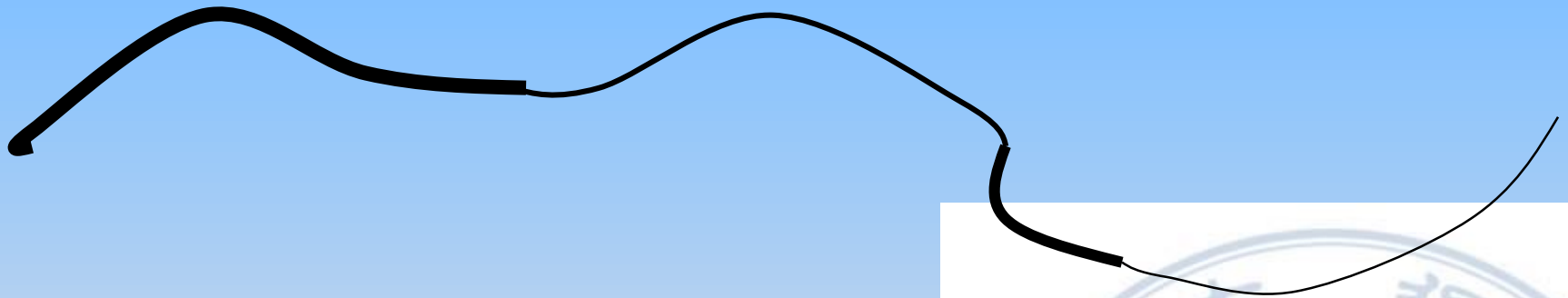
(a) Locate the next unvisited edge pixel p in I_H .

(b) Mark as valid edge pixels all the weak pixels in I_L that are connected to p.

(c) If all nonzero pixels in I_H have been visited, go to (d),else to (a).

(d) Set to zero all pixels in I_L that are not marked as valid edge pixels.

Hysteresis thresholding



Use a high threshold to start edge curves and a low threshold to continue them.

Canny Edge Detector

Effect of σ (Gaussian Kernel Size)

original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

The choice of σ depends on desired behavior

- large σ detects large scale edges
- small σ detects fine features

Hough Transform

Edge detection yields sets of pixels lying on edge. We use Hough transform to link them.

Hough Transform

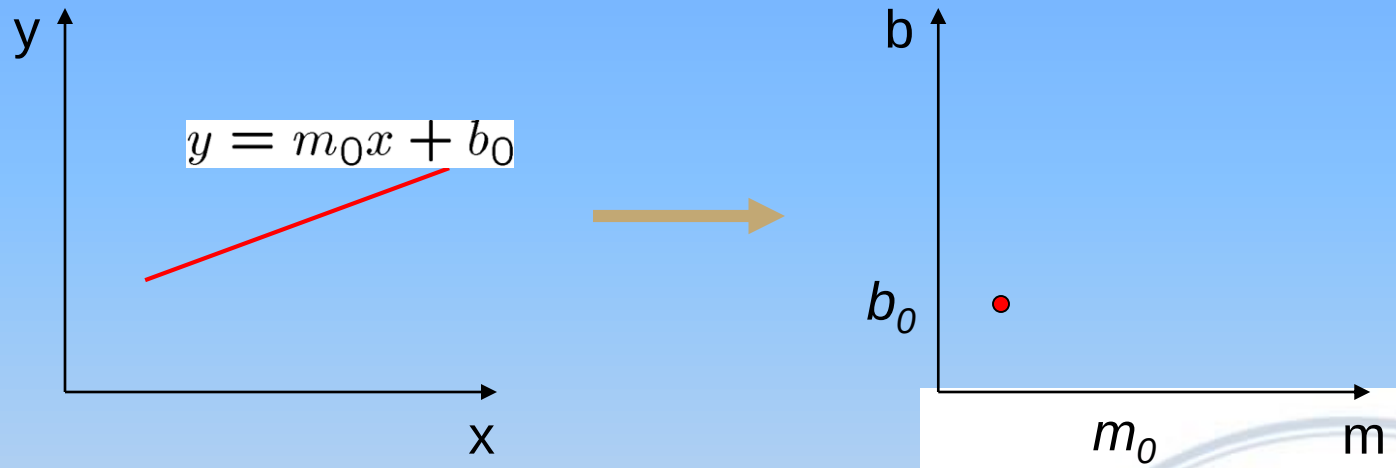
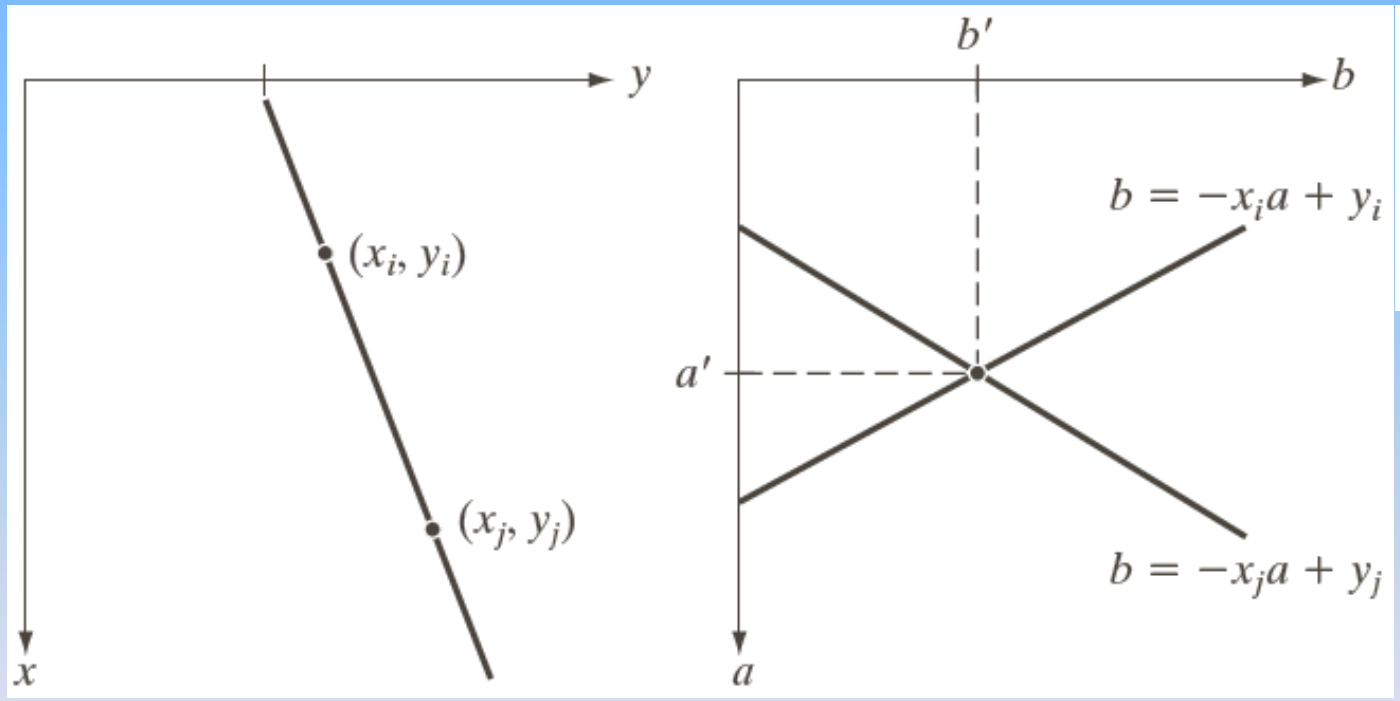


image space Hough space

- Connection between image (x, y) and Hough (m ,b) spaces
  - A line in the image corresponds to a point in Hough space
  - To go from image space to Hough space:
    - given a set of points (x,y), find all (m,b) such that y = mx + b
  - What does a point (x_o, y_o) in the image space map to?

Hough Transform



a b

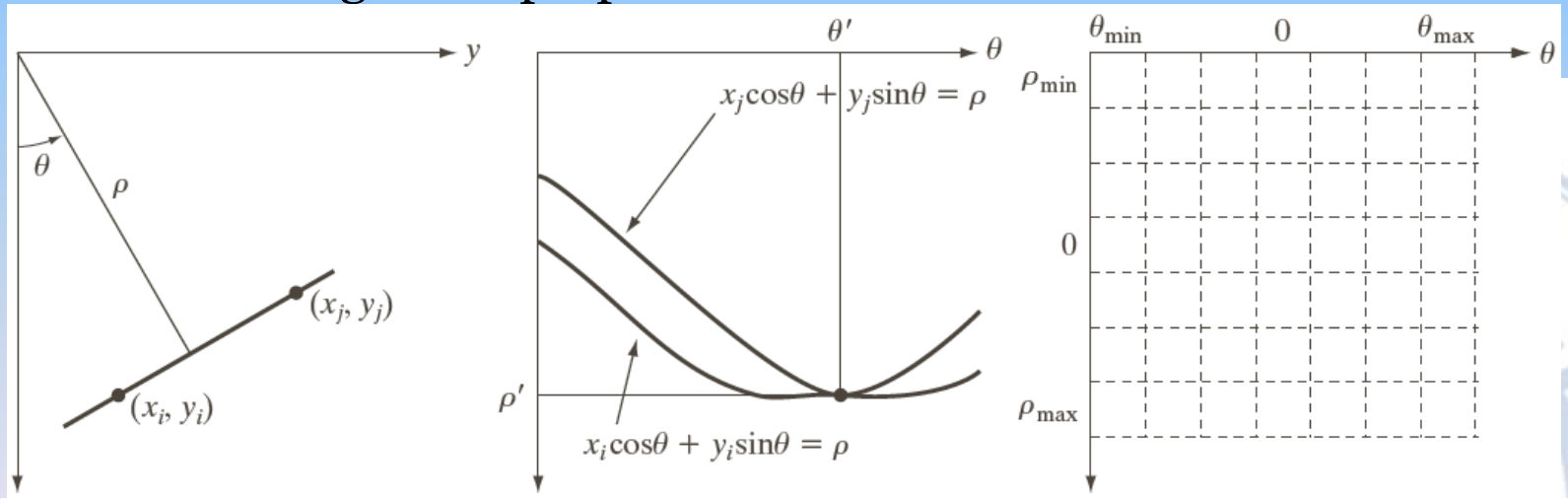
FIGURE 10.31
(a) *xy*-plane.
(b) Parameter space.

Hough Transform Algorithm

- Typically use a different parameterization

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

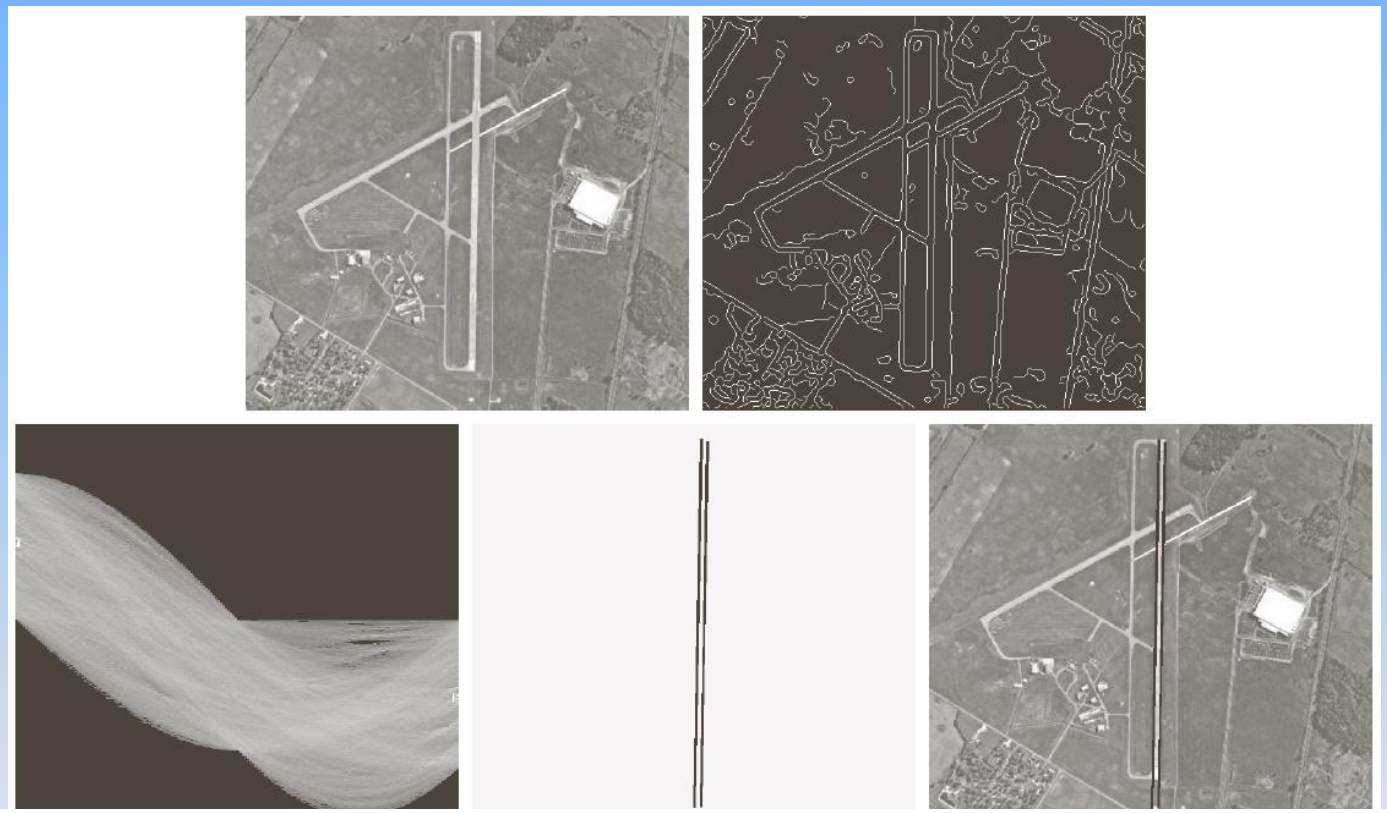
  - d is the perpendicular distance from the line to the origin
  - θ is the angle this perpendicular makes with the x axis



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.

Example



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.

Considering question

- Problem 10.23 (don't need to hand in)

Thank You!