Image Segmentation 1

李东晓

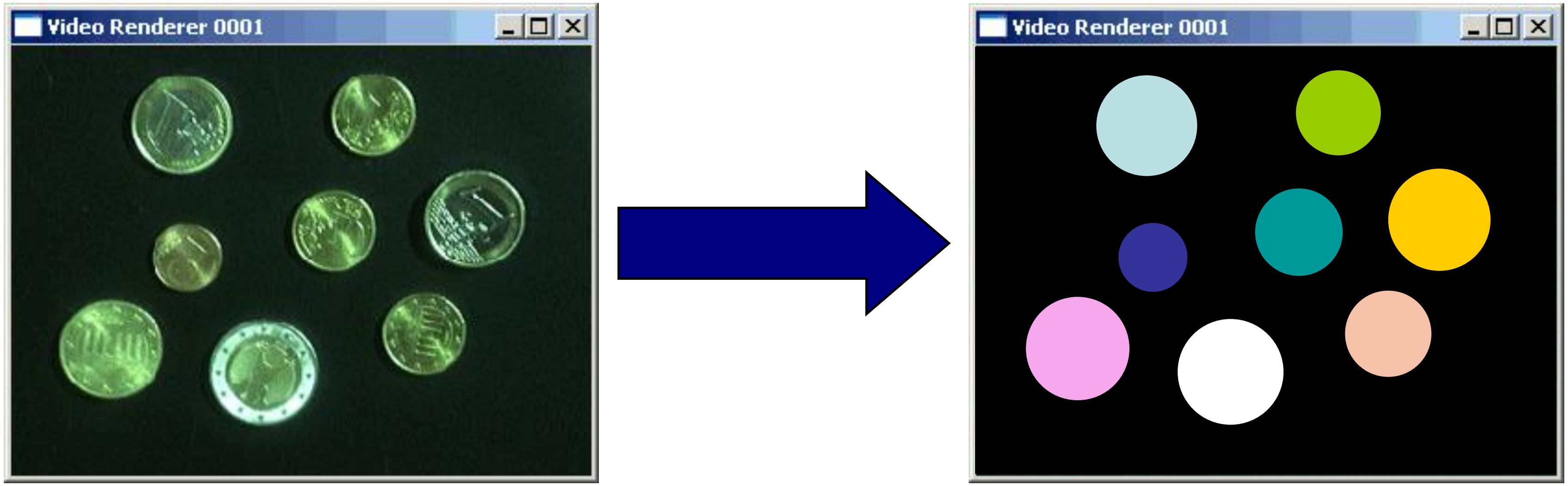
lidx@zju.edu.cn

浙江大学 信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

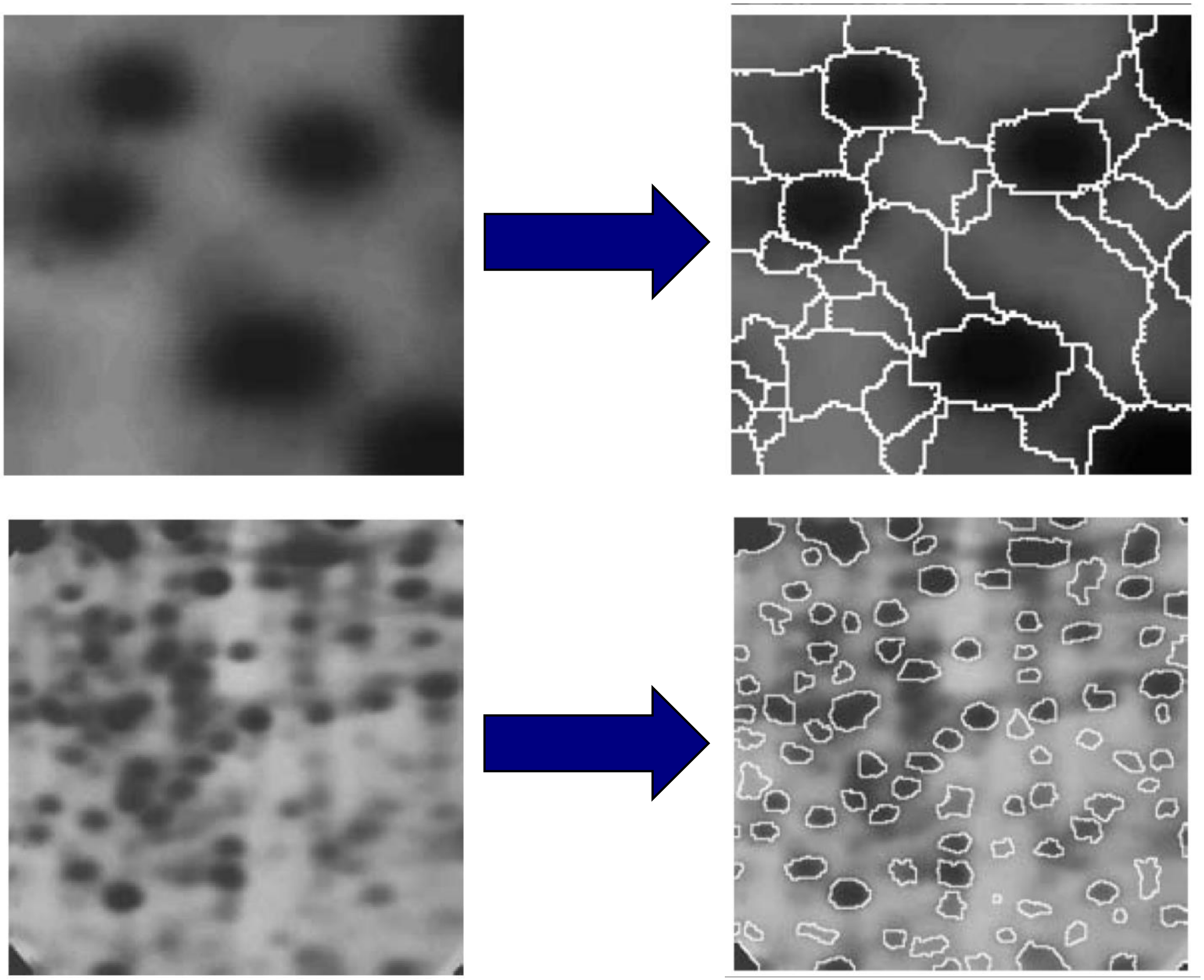
Contents

The Segmentation Problem

- Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image
- Typically the first step in any automated computer vision application

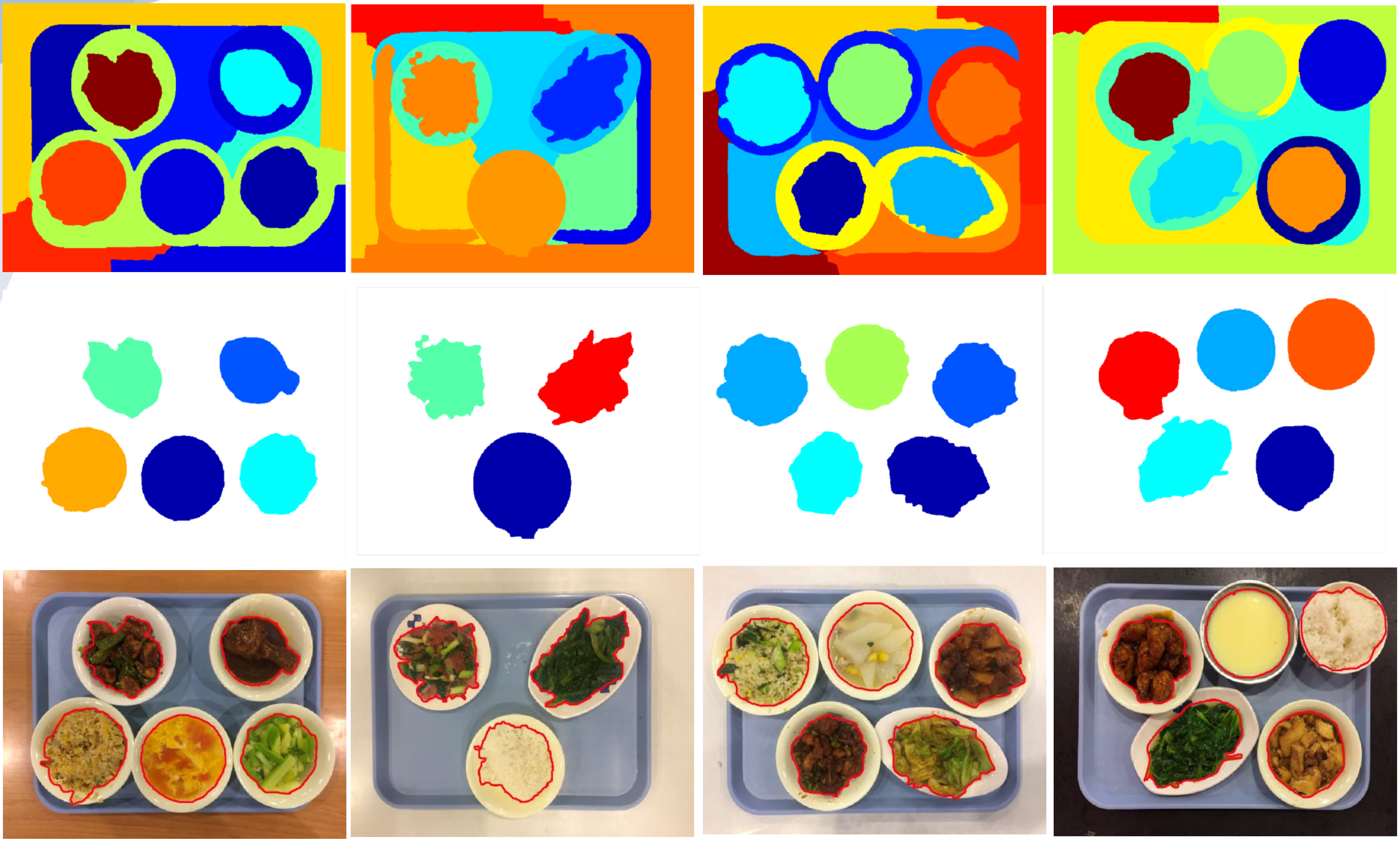


Segmentation Examples

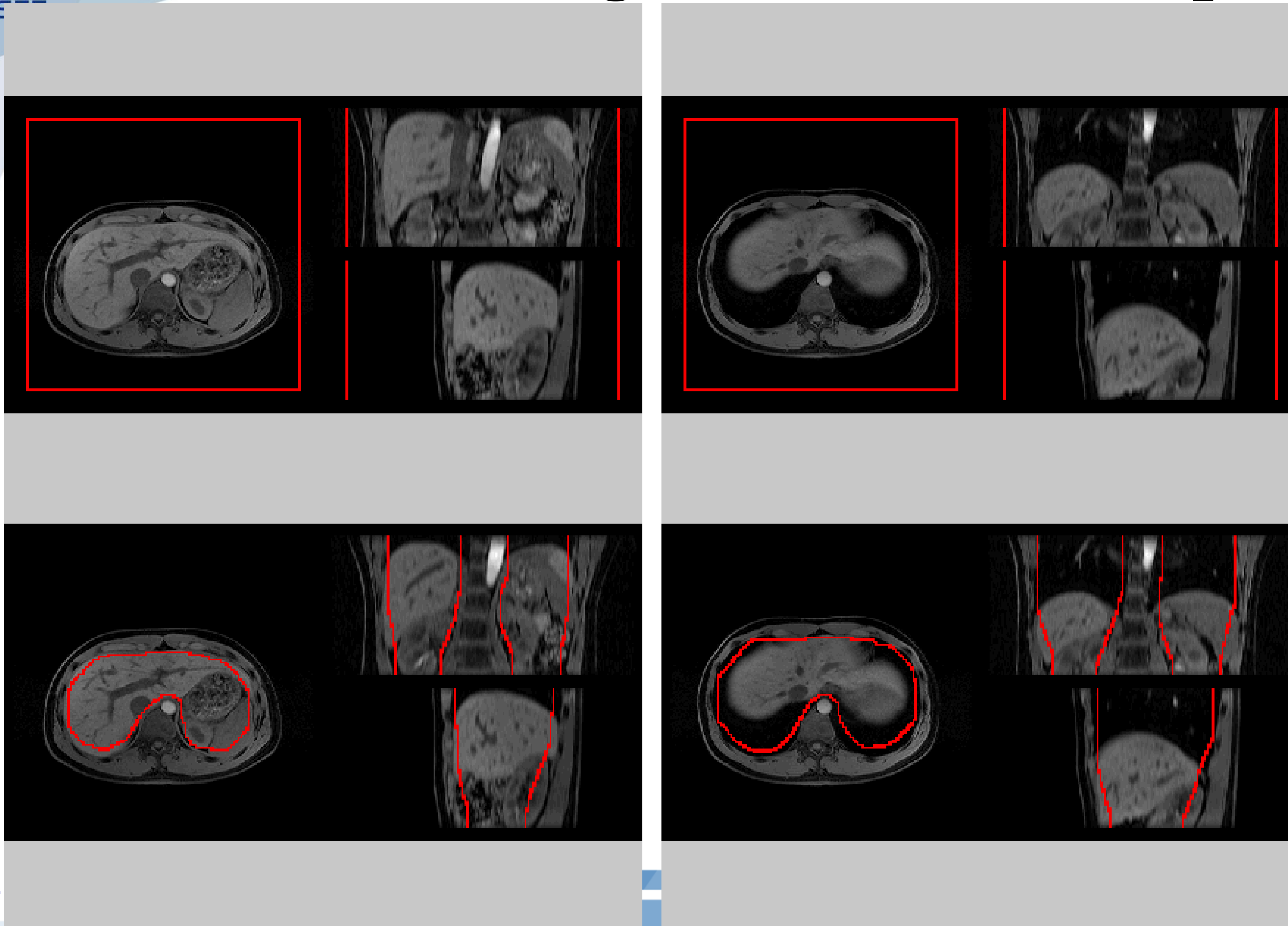


Segmentation Examples

Segmentation Examples



Segmentation Examples



Fundamentals

- Segmentation: partition image region R into n subregions, R_1, R_2, …, R_n

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

(b) R_i is a connected set, $i = 1, 2, \ldots, n$.

(c) $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j$.

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

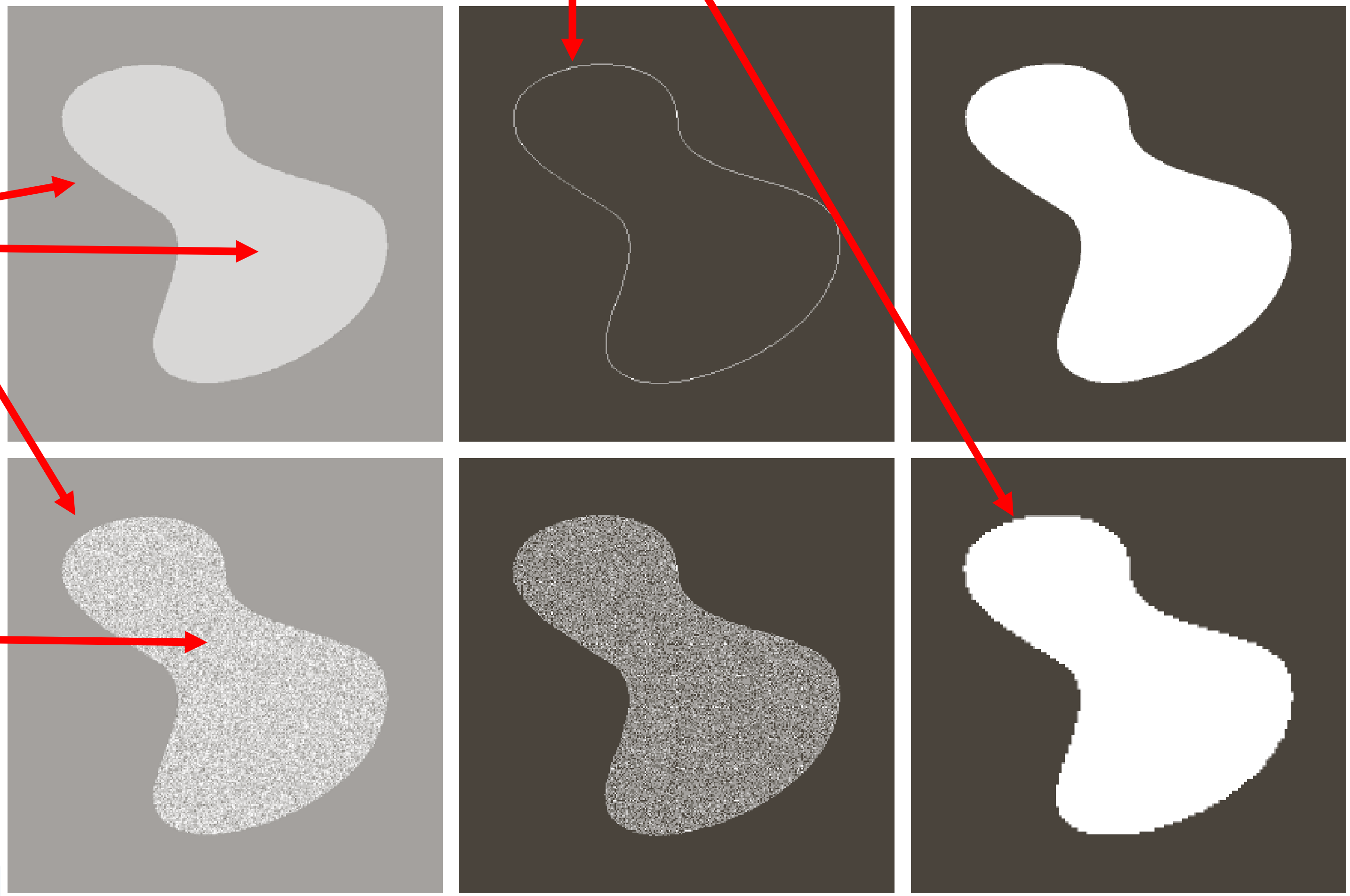
(e) $Q(R_i \cup R_j) = \text{FALSE}$ for any adjacent regions R_i and R_j.

Basic Idea in Segmentation

- Making the use of two basic properties:
  - Discontinuity → Edge-based
  - Similarity → Region-based

Constant

Texture



Detection Of Discontinuities

- Three basic types of grey level discontinuities
  - Points
  - Lines
  - Edges
- We typically find discontinuities using masks and correlation

Approximation of Derivatives

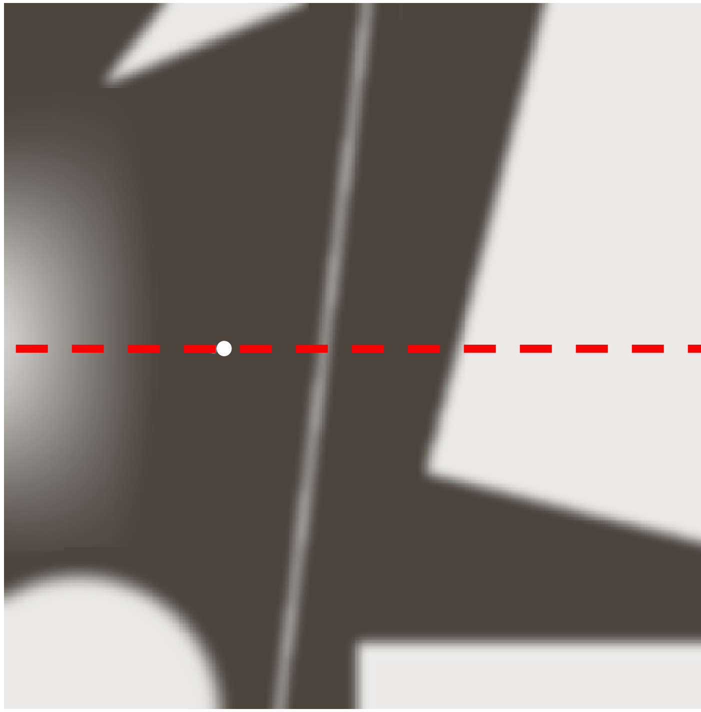
- Derivatives → Differences
- 1st-order derivative

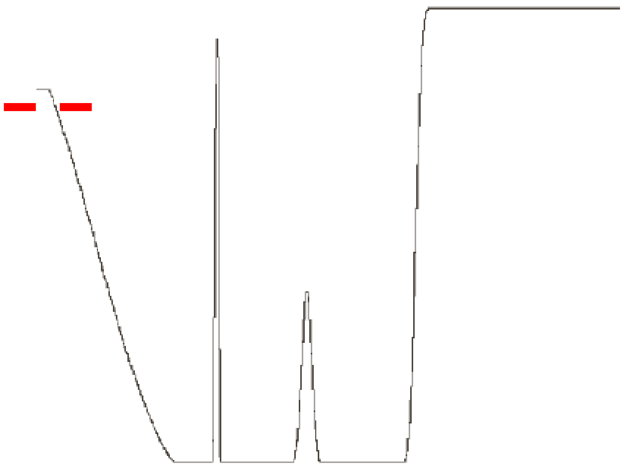
$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

- 2nd-order derivative

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x+1) + f(x-1) - 2f(x)$$

Approximation of Derivatives





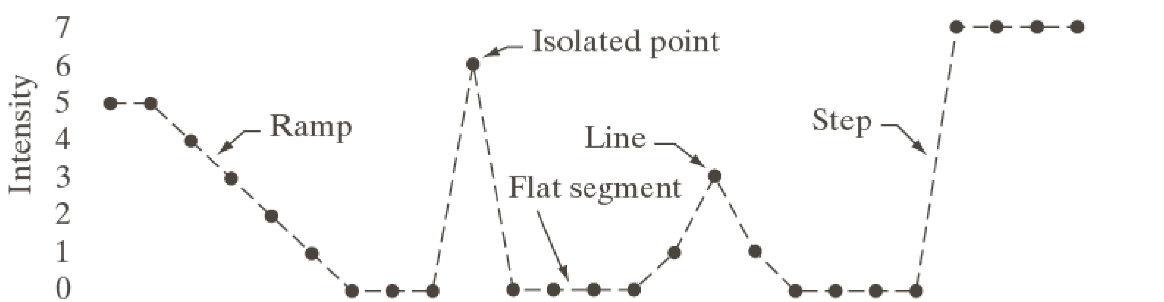
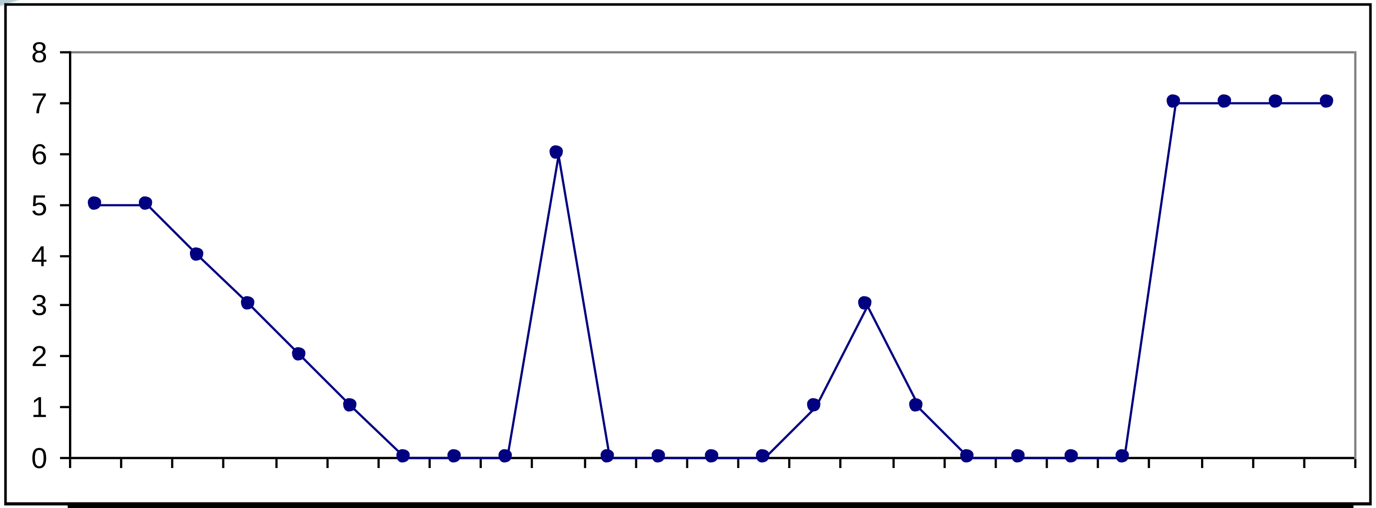


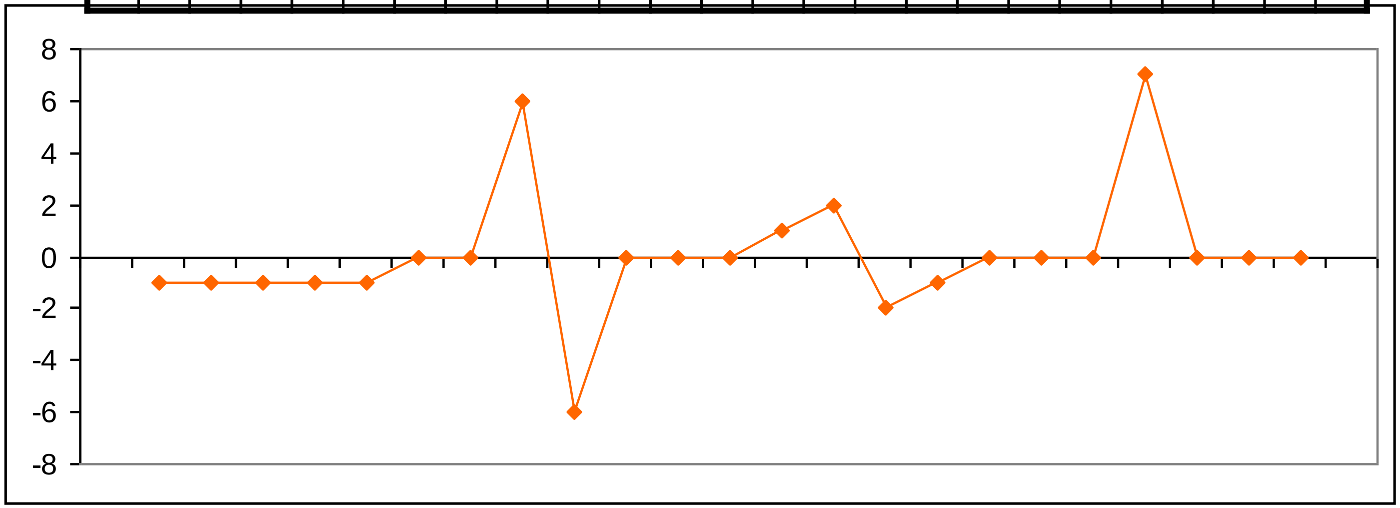
Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			

1^{st} Derivative (cont…)

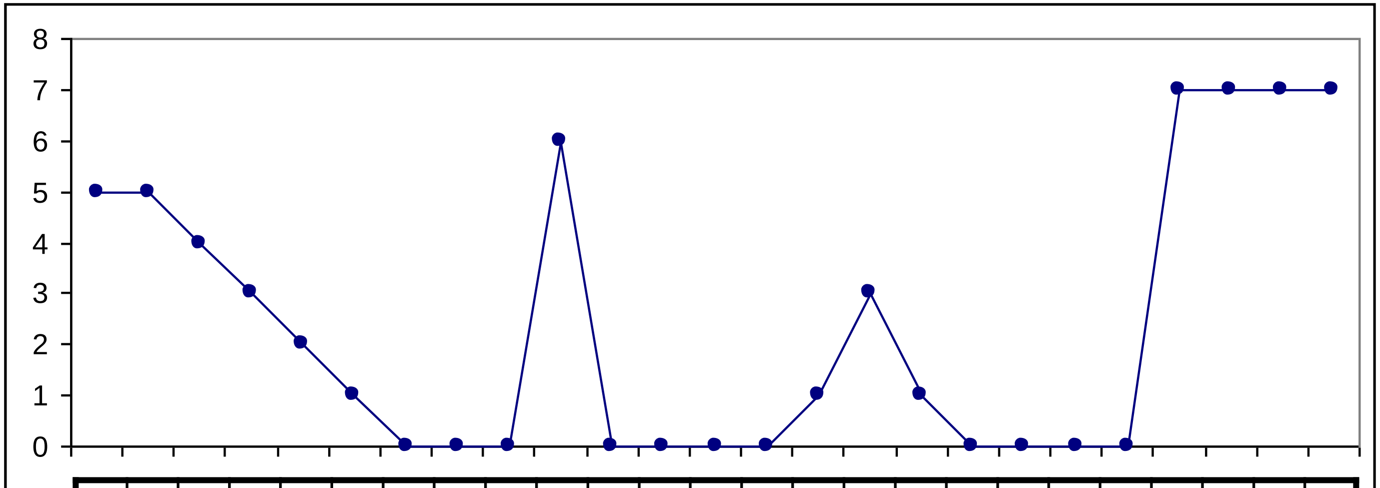


5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	

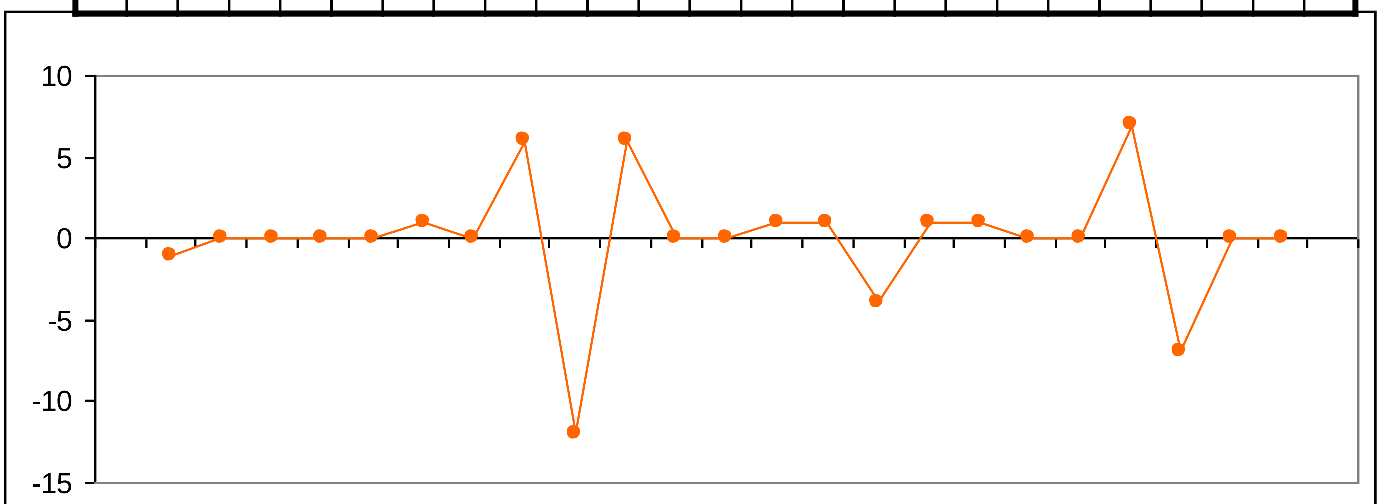


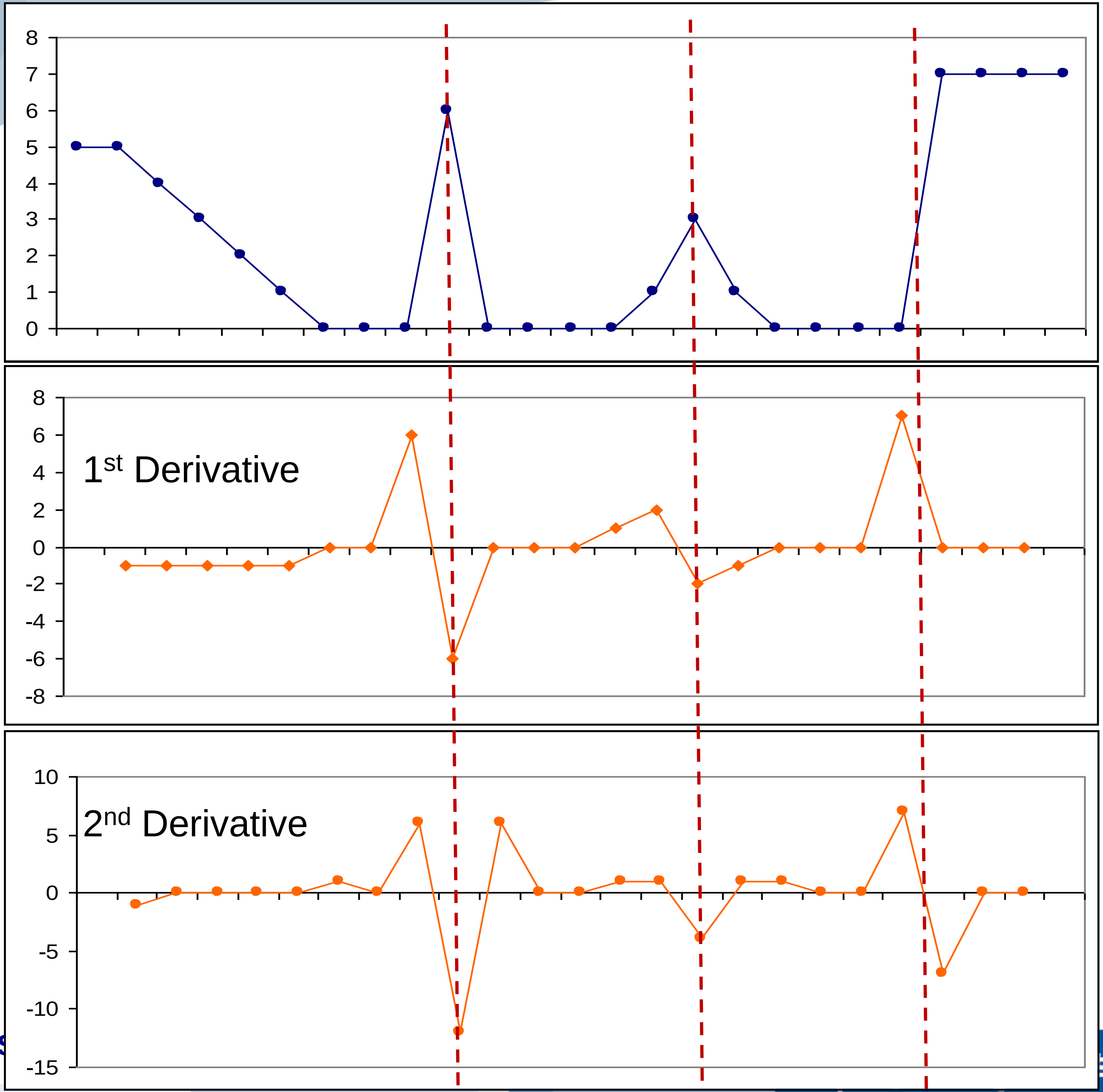
2nd Derivative (cont…)



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

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	





1st Derivative
2nd Derivative

Summary on Derivatives

- 1st-order derivatives produce thicker edges
- 2nd-order derivatives have a strong response to fine detail, such as thin lines, isolated points, and noise
- 2nd-order derivatives produce a double-edge response at ramp and step transitions
- The sign of 2nd-order derivatives indicates the transition of moving into the edge

  "-": light → dark

  "+": dark → light

Spatial Filter for Derivatives

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9$$

$$= \sum_{k=1}^{9} w_k z_k$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Contents

Detection of Isolated Points

- 2^{nd}-order derivative → Laplacian

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

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

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

Point Detection

- Laplacian filter for point detection

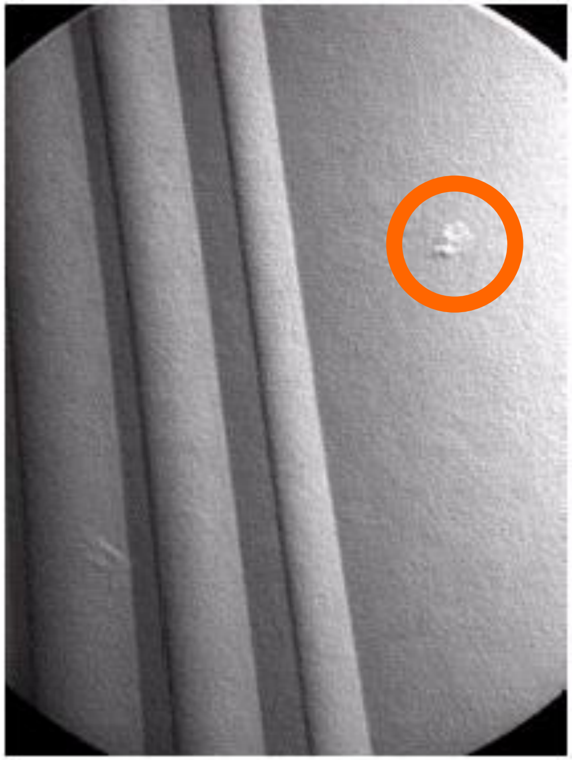
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

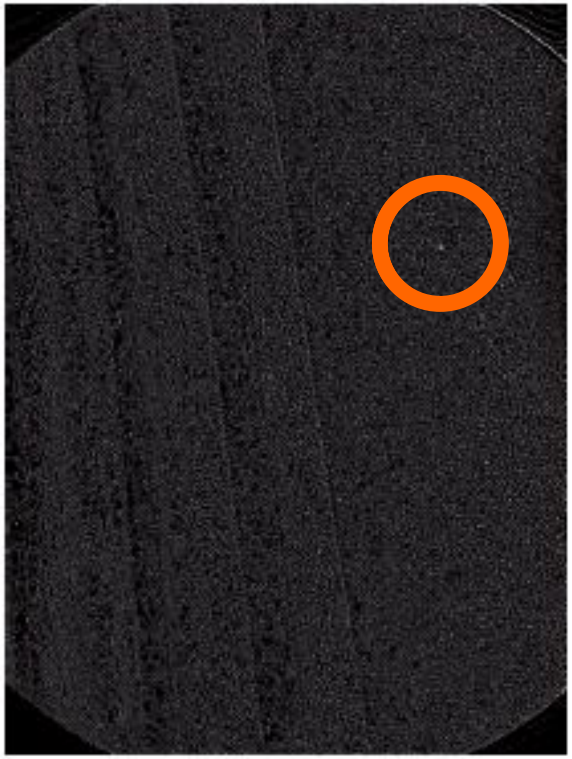
- Points detection by thresholding

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

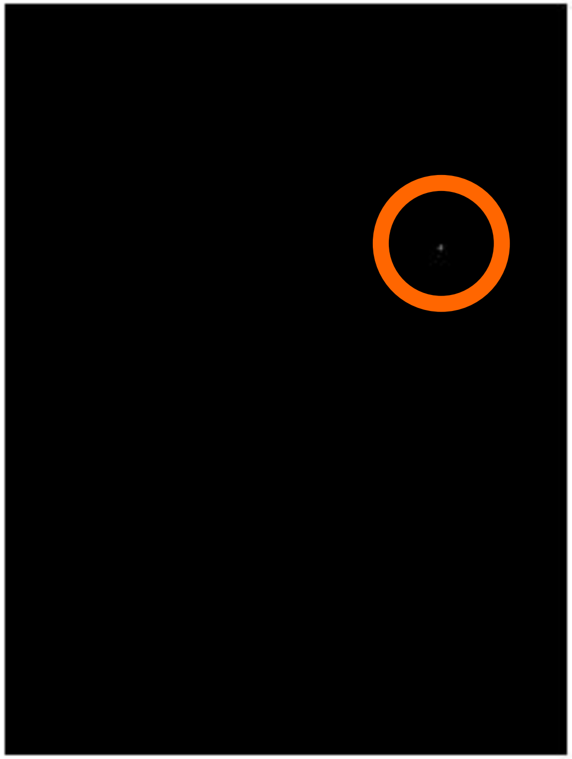
Point Detection (cont…)



X-ray image of a turbine blade （涡轮叶片）



Result of point detection



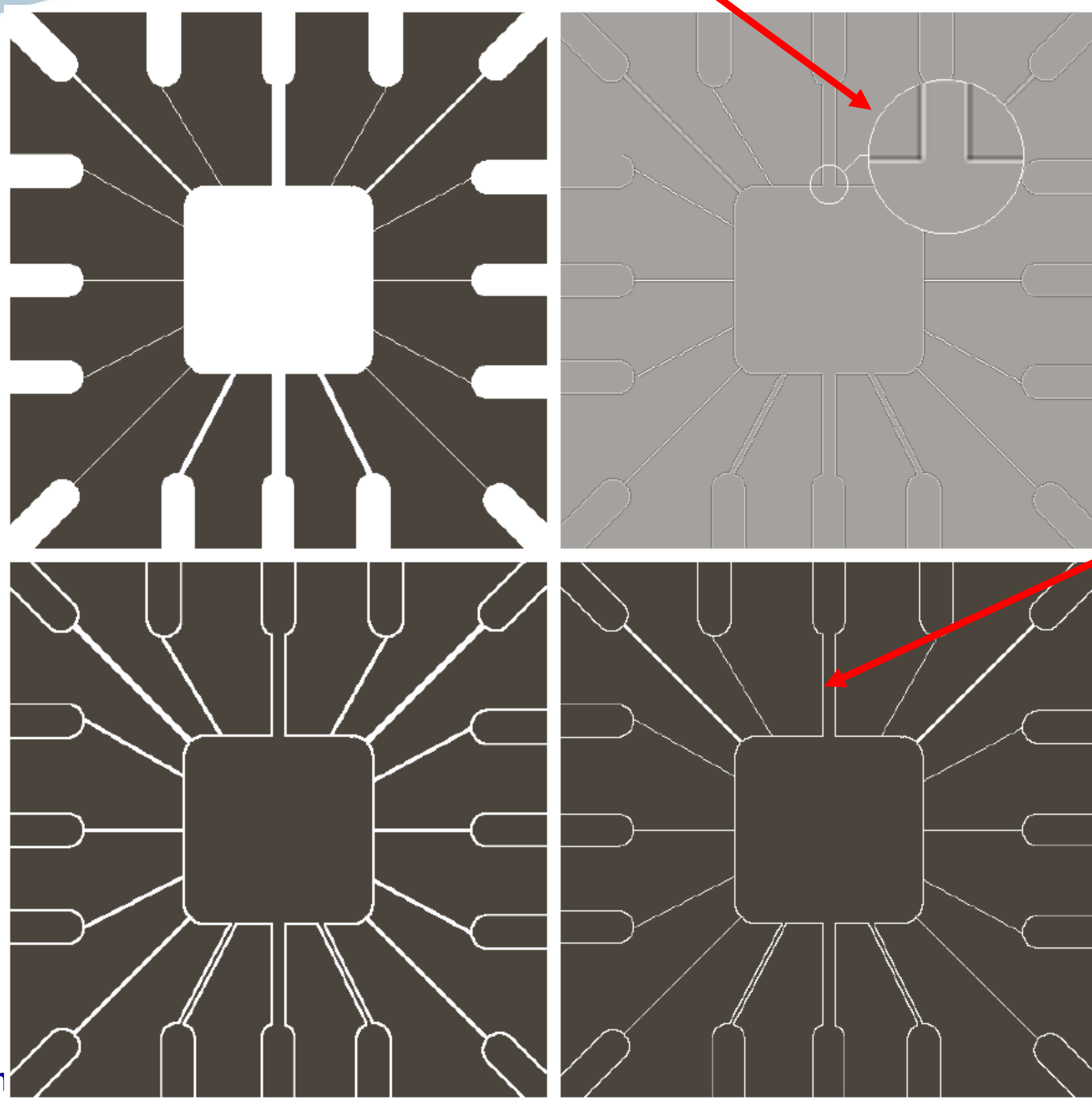
Result of thresholding

Line Detection

Double-line effect

Original image

Laplacian image



Valley (region)

Absolute value

Positive value

Line Detection

- The following masks will extract lines that are one pixel thick and running in a particular direction

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

- Line direction estimation

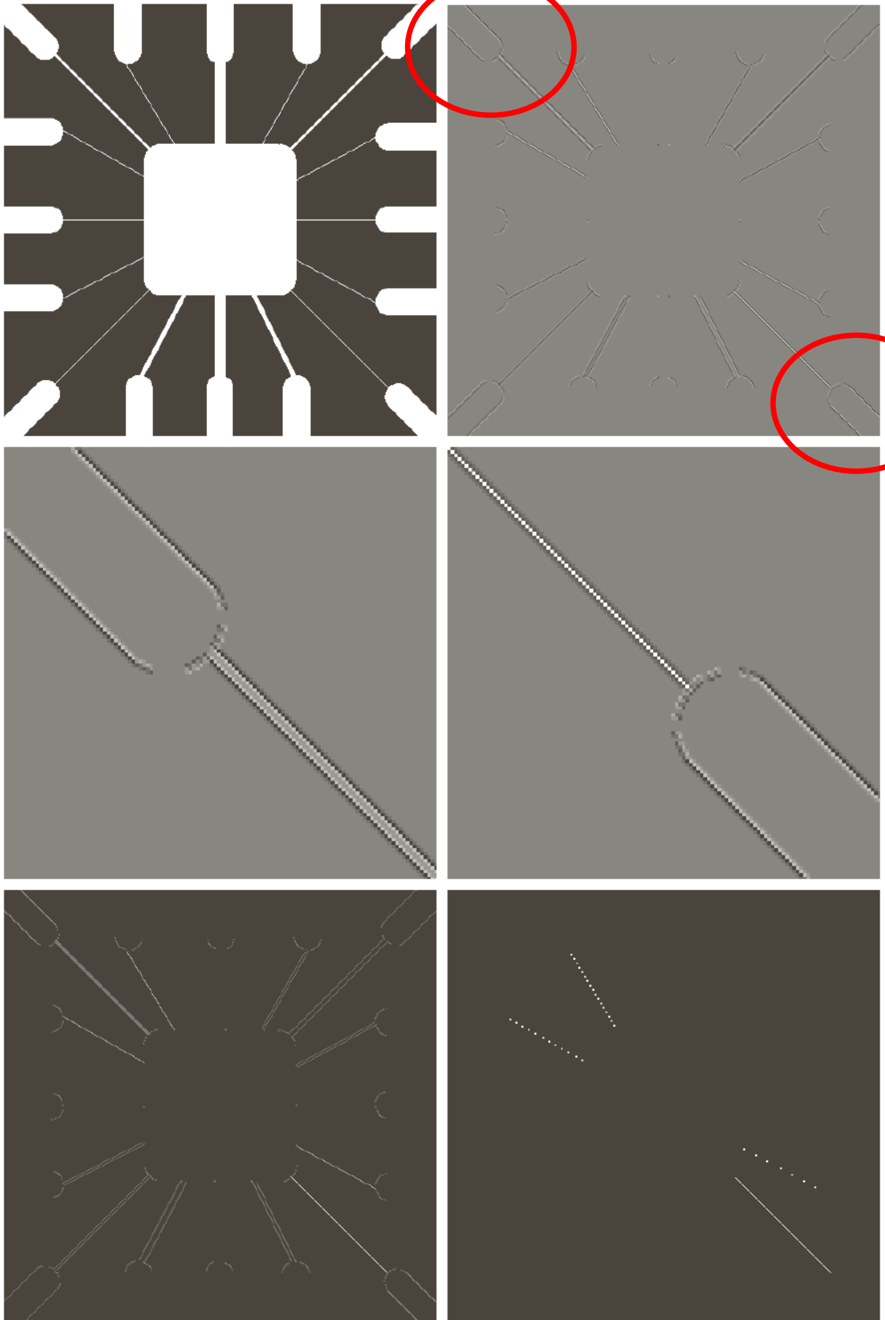
$$|R_k| > |R_j|, \text{ for all } j \neq k$$

Line Detection (cont…)

Binary wire-bond image

Zoomed view of the top left (thick line)

Negative values set to zero



2	−1	−1
−1	2	−1
−1	−1	2

+45°

Zoomed view of the bottom right (thin line)

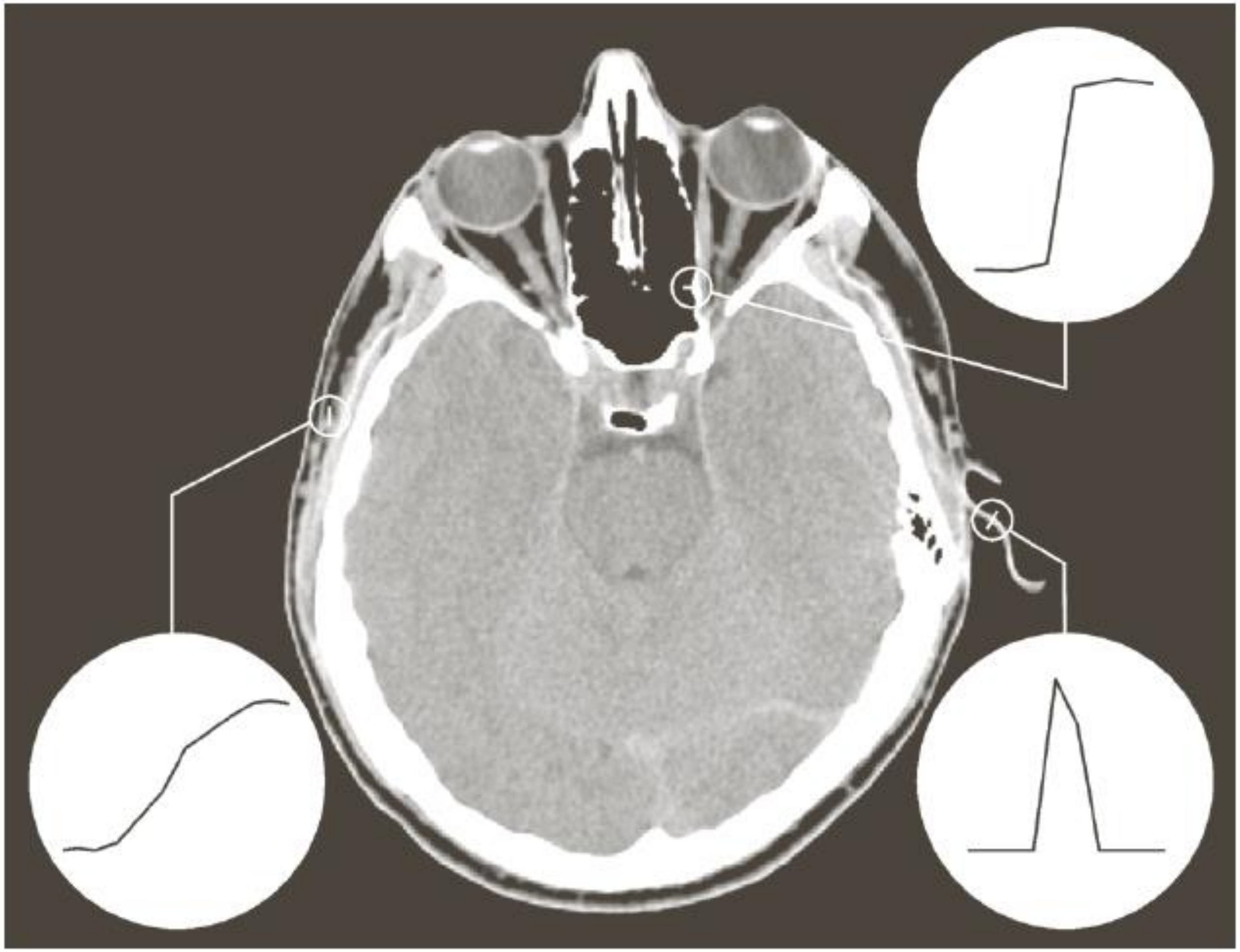
After thresholding

Edge Models

- An edge is a set of connected pixels that lie on the boundary between two regions

Step   Ramp   Roof

Actual Edge Profiles

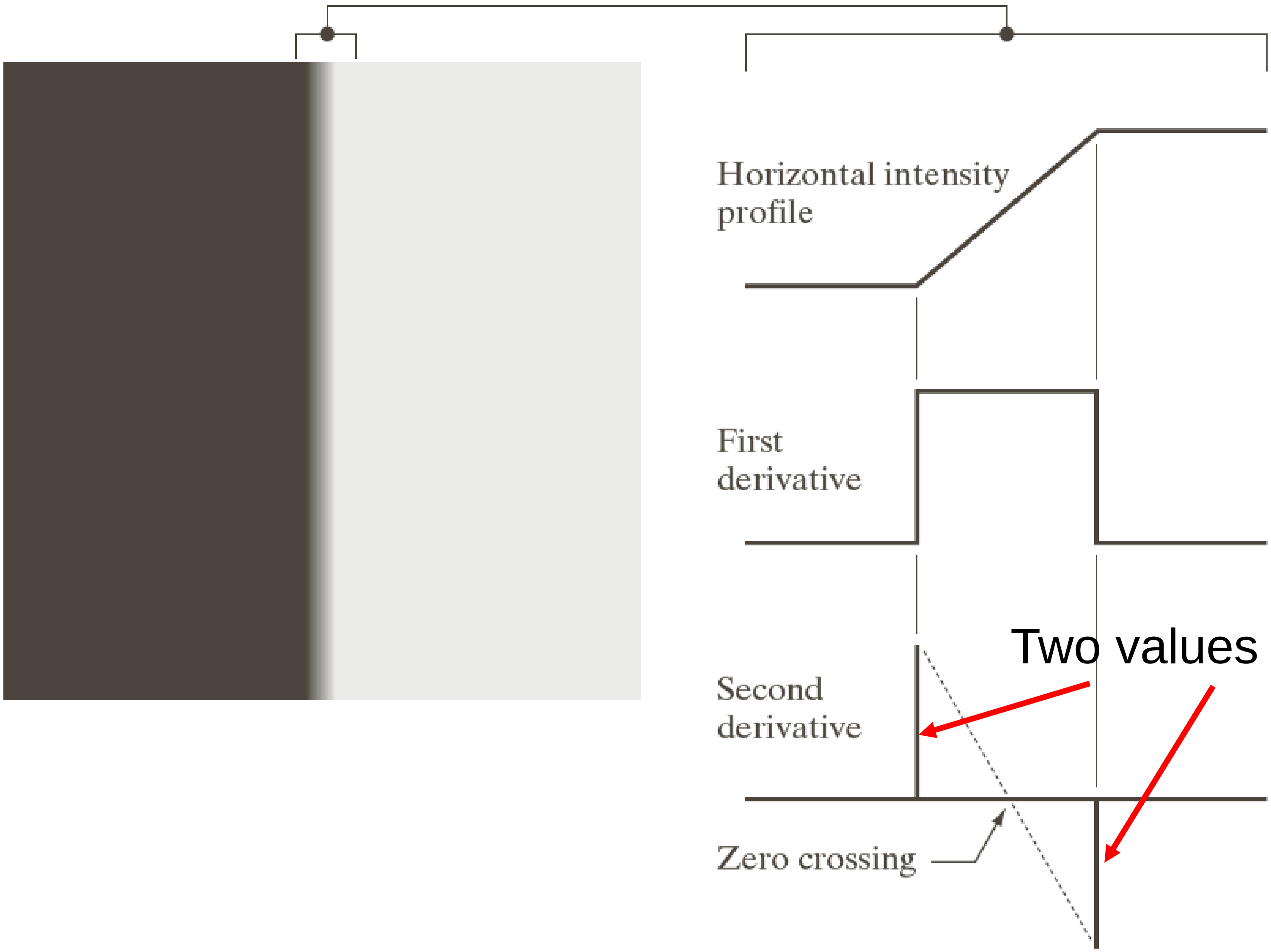


Step

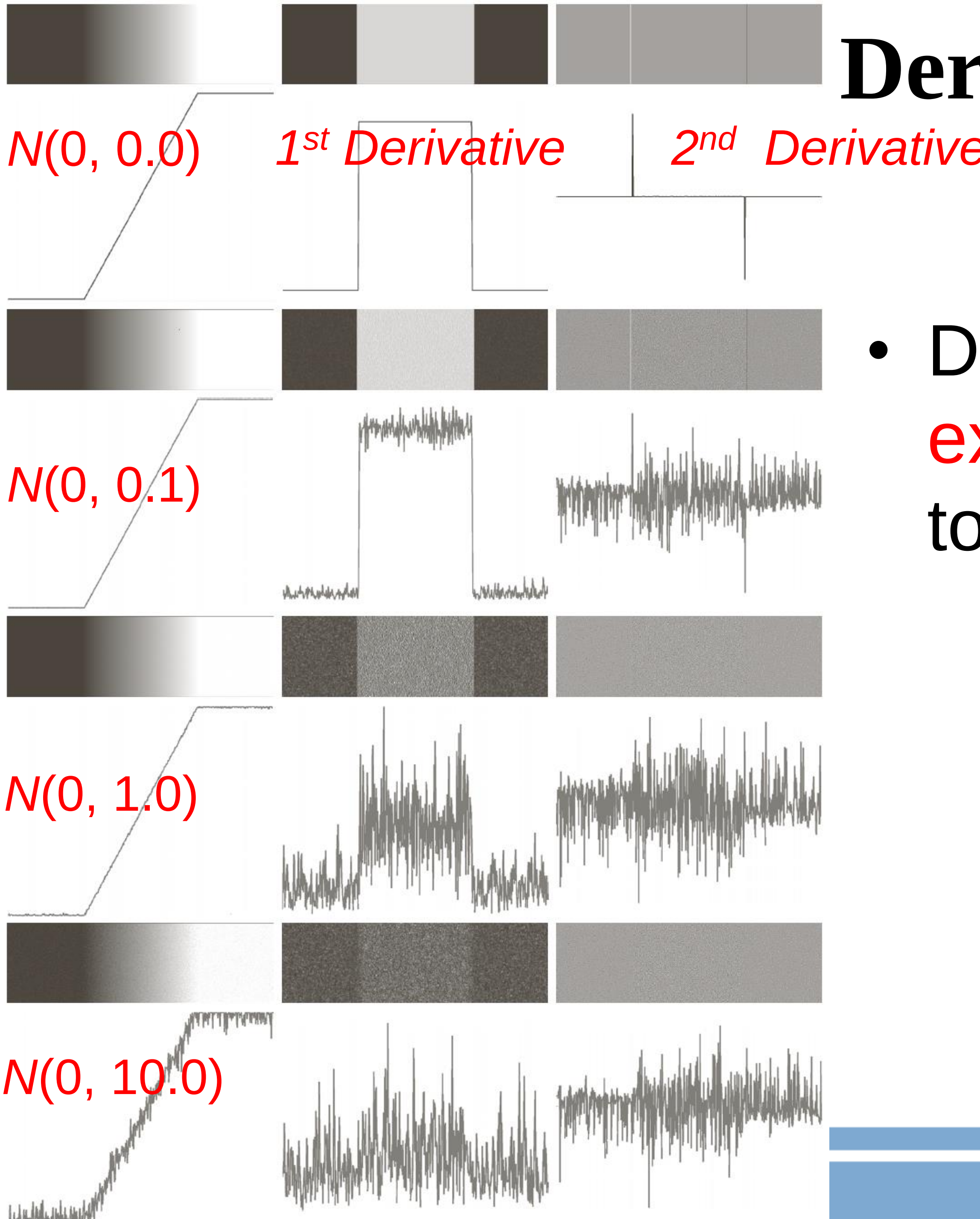
Ramp

Roof

Edges & Derivatives



Derivative & Noise



- Derivatives are **extremely sensitive** to noise

Fundamental Steps in Edge Detection

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

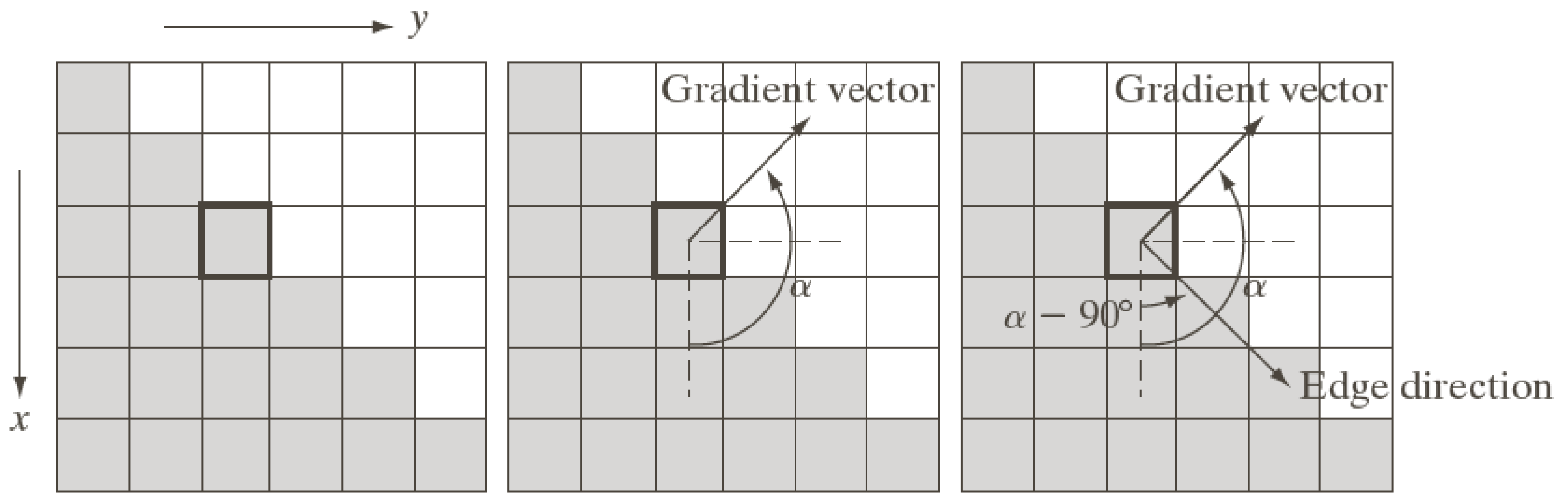
Basic Edge Detection using 1st-order Derivatives

- Image gradient and its properties

$$\nabla f \equiv \mathrm{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x, y) = \mathrm{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$



Common Edge Detectors

- Given a 3*3 region of an image, the following edge detection filters can be used

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	0
0	1

0	−1
1	0

Roberts

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Diagonal Prewitt & Sobel Masks

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

Edge Detection Example

Original Image

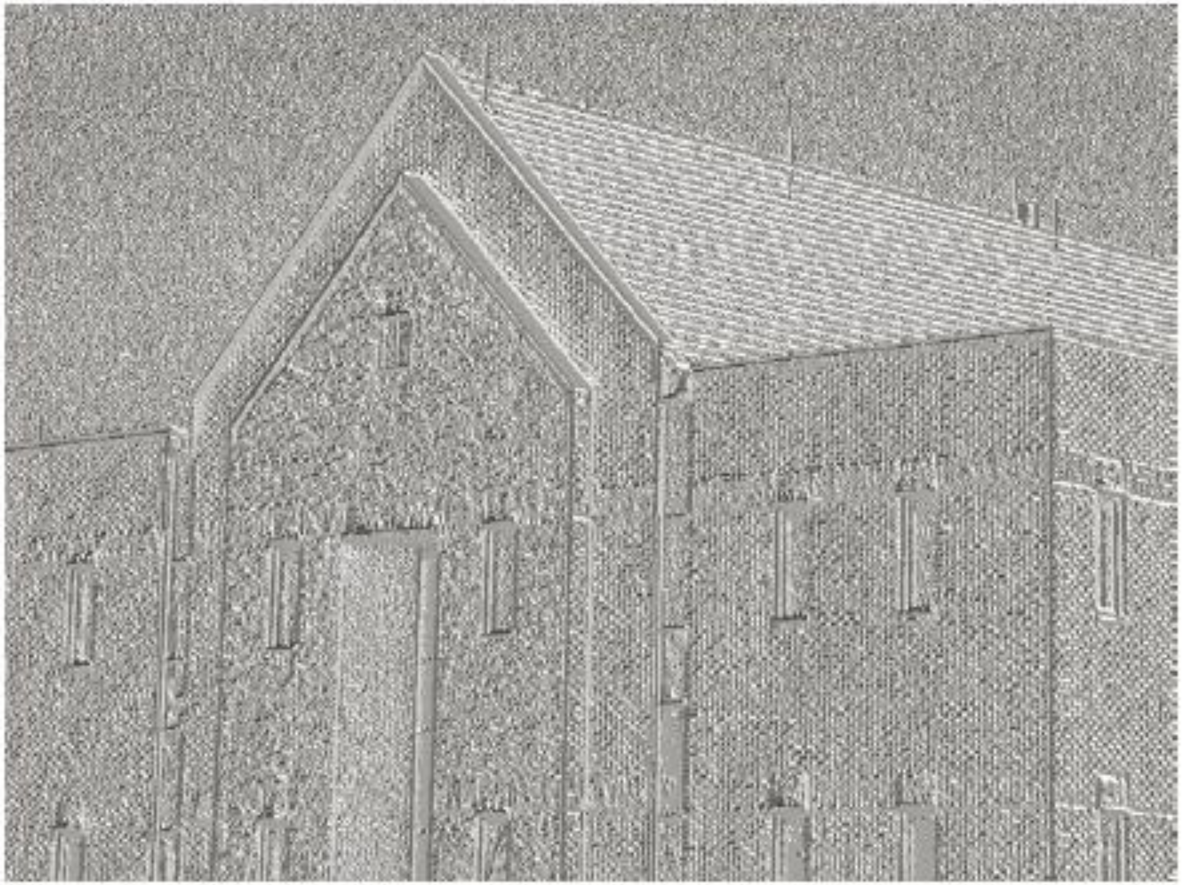
$|g_x|$

$|g_y|$

$|g_x| + |g_y|$

Gradient Angle Image

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$



Edge Detection Problems

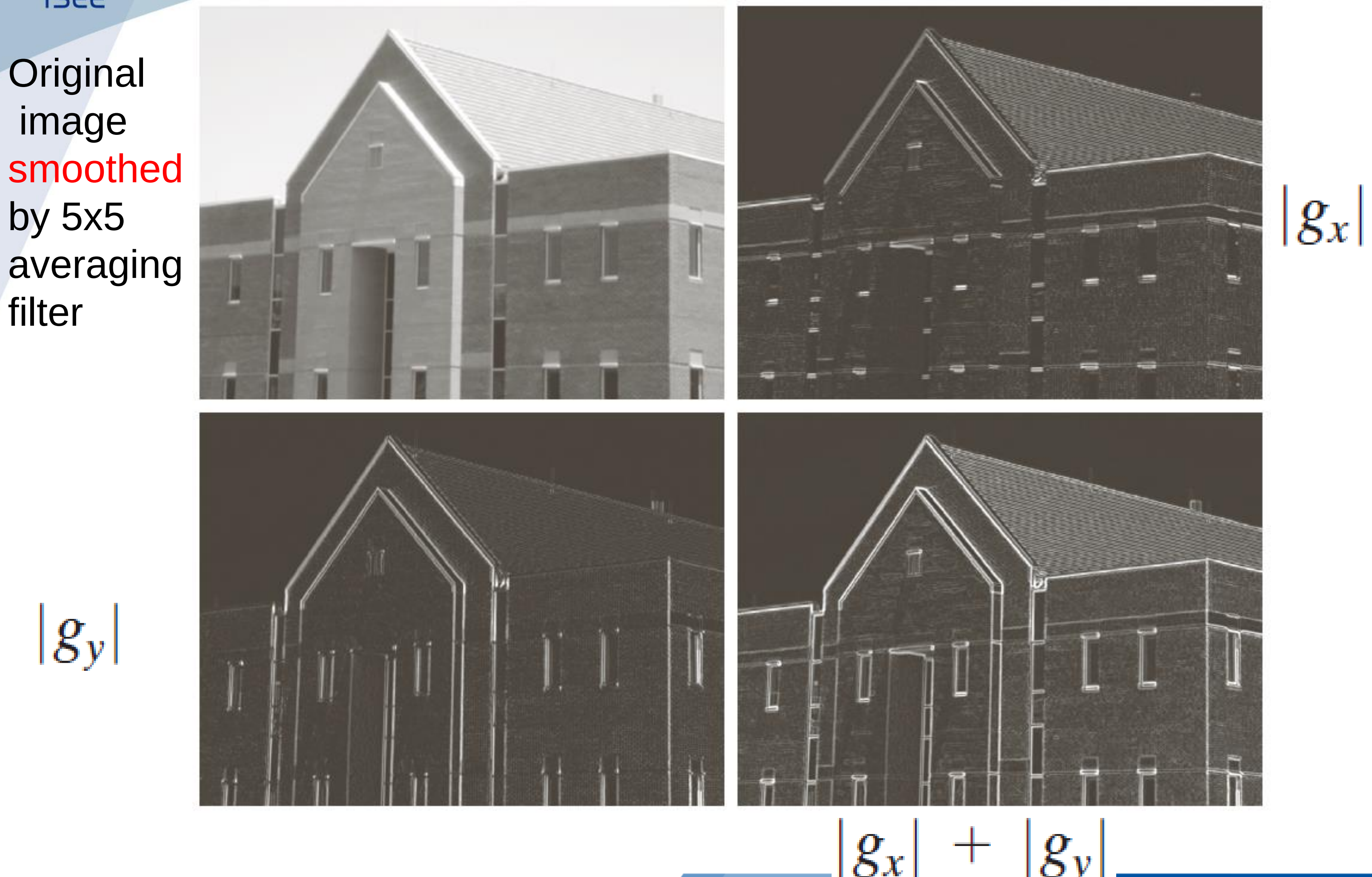
Often, problems arise in edge detection is that there are too much details

For example, the brickwork in the previous example

One way to overcome this is to smooth images prior to edge detection

Edge Detection Example

Original image smoothed by 5x5 averaging filter



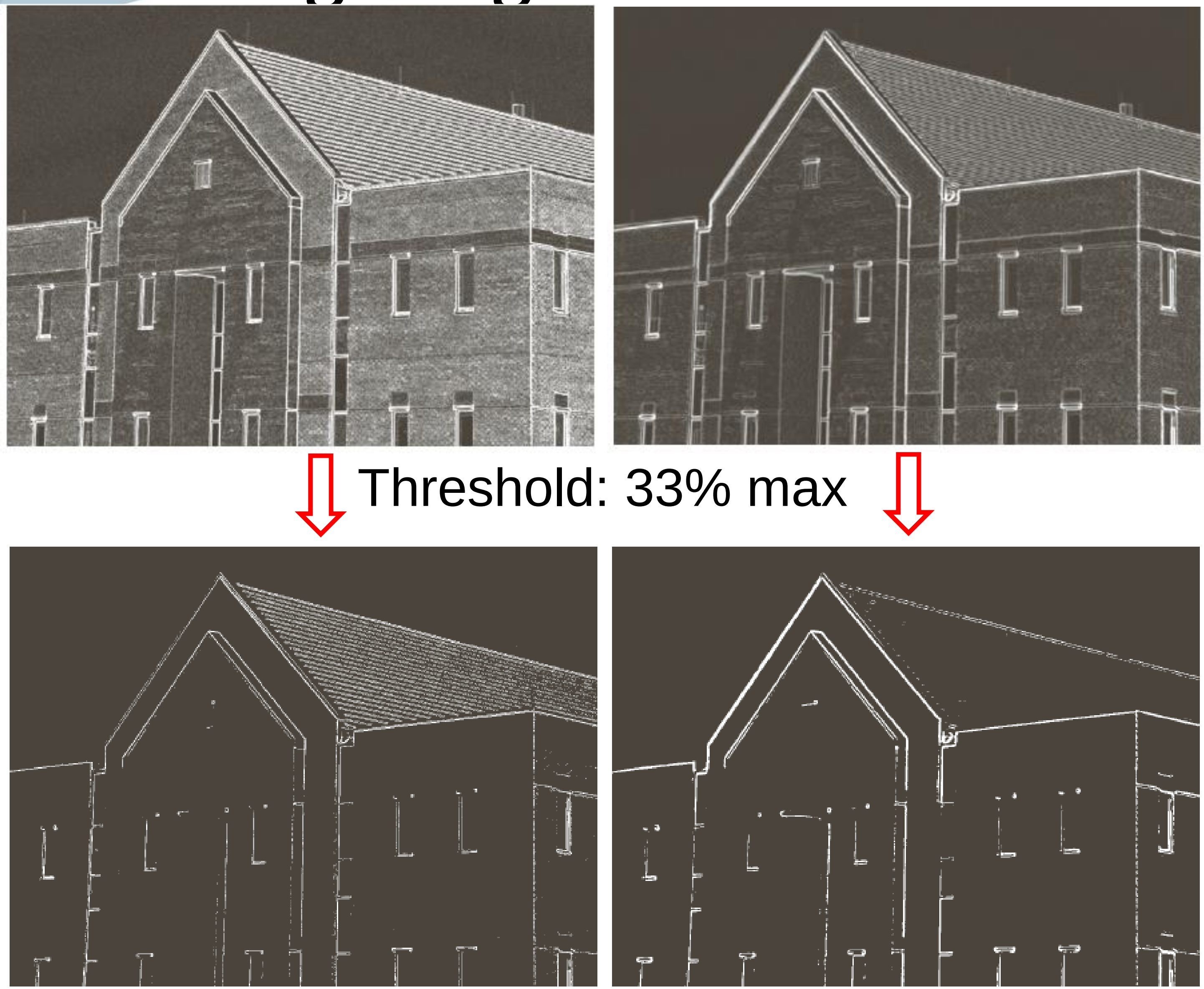
Diagonal Edge Detection

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Combining the gradient with thresholding



Laplacian Edge Detection

- 2^{nd}-order derivative based Laplacian filter

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

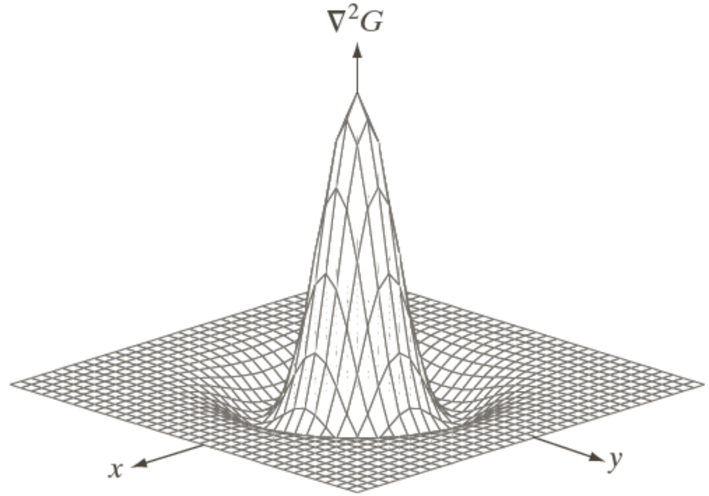
- The Laplacian is typically not used by itself as it is too sensitive to noise
- Usually Laplacian is combined with a smoothing Gaussian filter

Laplacian of Gaussian (LoG)

- LoG (Marr-Hildreth, Mexican hat)

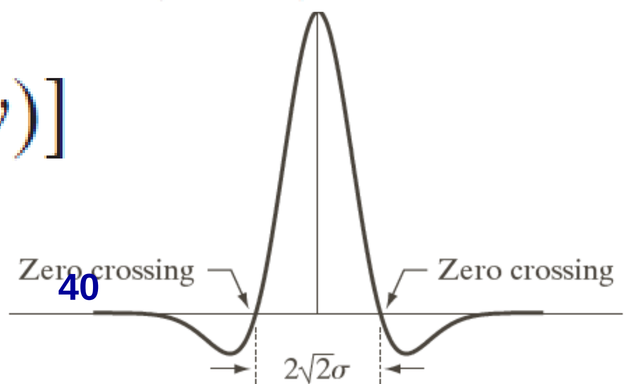
$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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



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

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



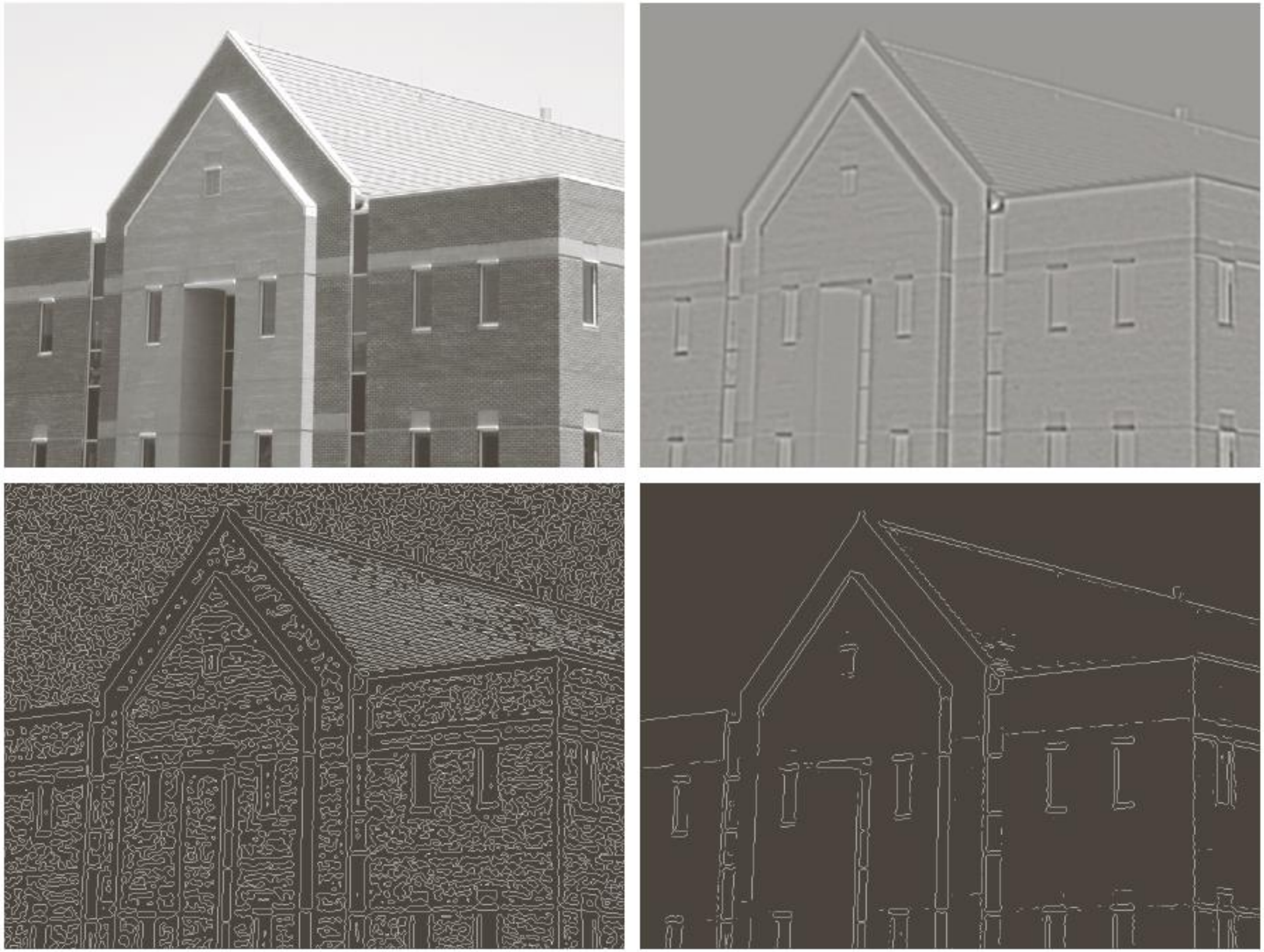
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

LoG Example

$\sigma = 4$ and $n = 25$

Original

LoG



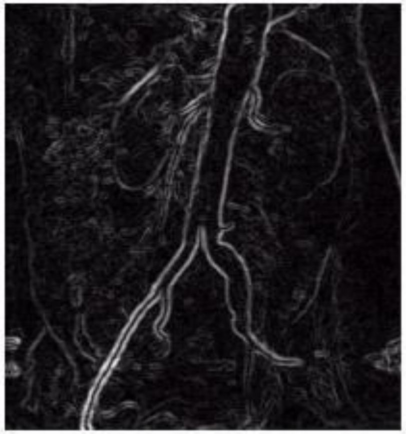
Zero crossing

Threshold = 0

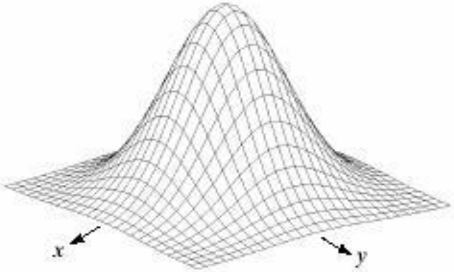
Threshold = 4% max

LoG Example

(a) 原图



(b)Sobel梯度图



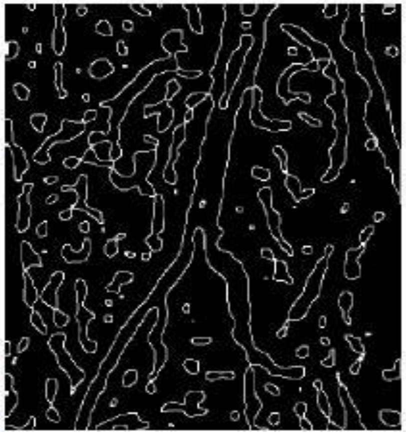
(c) 高斯函数

–1	–1	–1
–1	8	–1
–1	–1	–1

(d) 拉普拉斯算子

(e) 运用LoG后的图

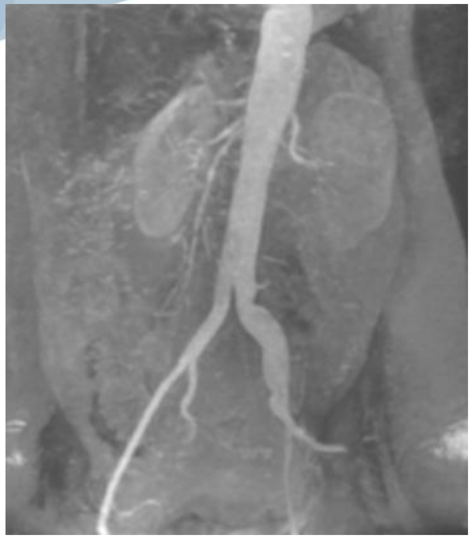
(f) 图（e）的二值化



(g) 过零点形成边缘

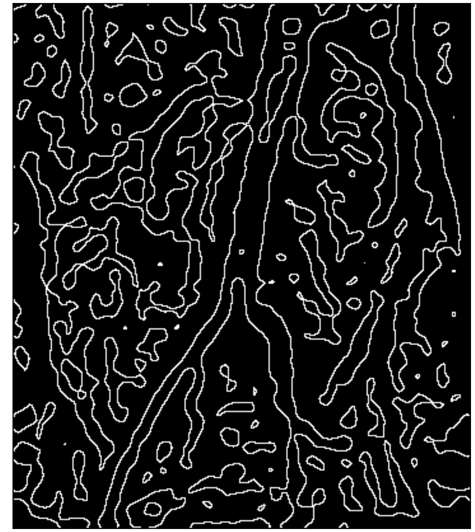
用Marr算子得到的边缘

LoG Example

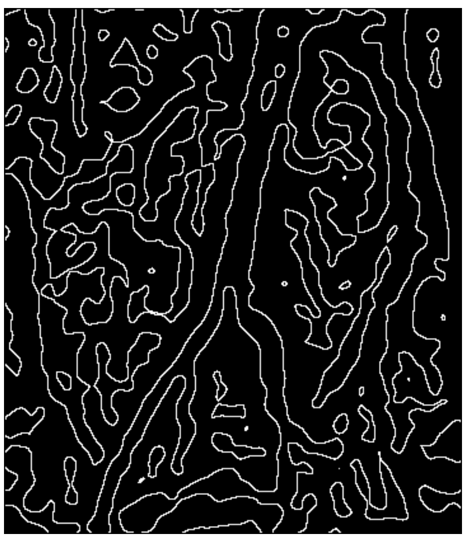


(a) 原图

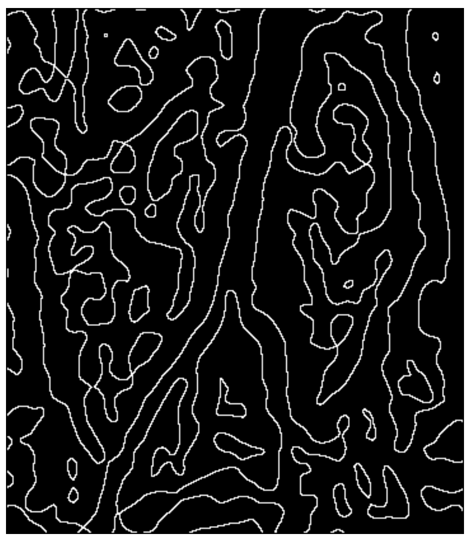
(b) σ =3



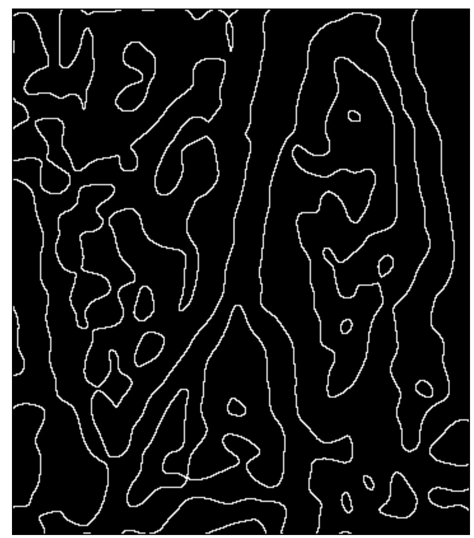
(c) σ =5



(d) σ =6



(e) σ =7



(f) σ =9

不同 σ 的Marr算子得到的边缘

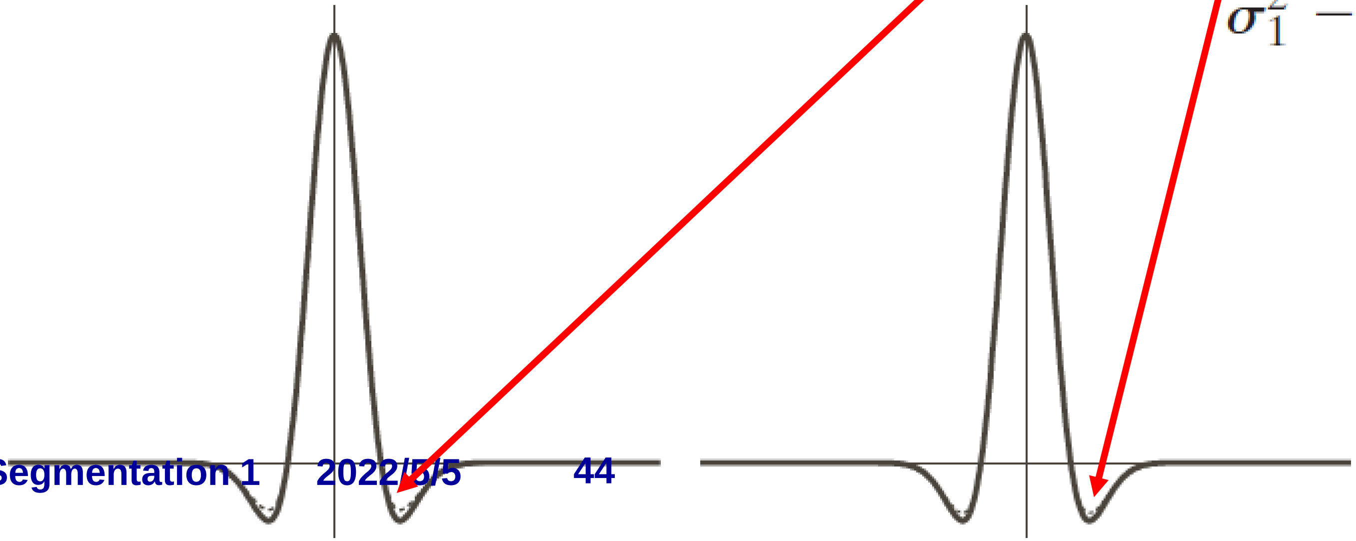
Difference of Gaussian (DoG)

- Approximate LoG by DoG

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

- Human vision system: $\sigma_1 : \sigma_2 = 1.75 : 1$
- Marr & Hildreth suggest: $\sigma_1 : \sigma_2 = 1.6 : 1$
- Compatible LoG:

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left[\frac{\sigma_1^2}{\sigma_2^2}\right]$$



Canny Edge Detector

- Three basic objectives
  - Low error rate
  - Edge points should be well localized
  - Single edge point response
- A good approximation to the optimal step edge detector is the 1st derivative of a Gaussian

$$\frac{d}{dx}e^{-\frac{x^2}{2\sigma^2}} = \frac{-x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}$$

Canny Edge Detector

- 2D Gaussian function $G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$
- Smoothed image $f_s(x, y) = G(x, y) \star f(x, y)$
- Gradient $g_x = \partial f_s / \partial x \quad g_y = \partial f_s / \partial y$

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

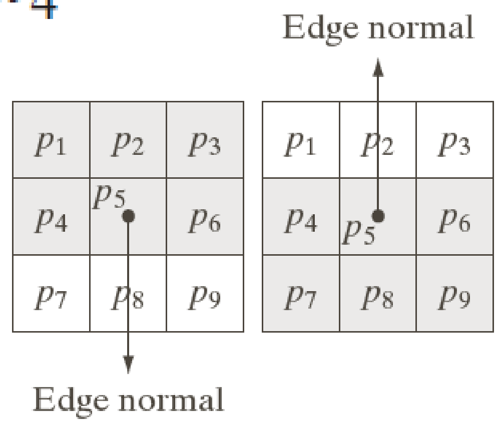
- How to thin the ridges? Nonmaxima suppression

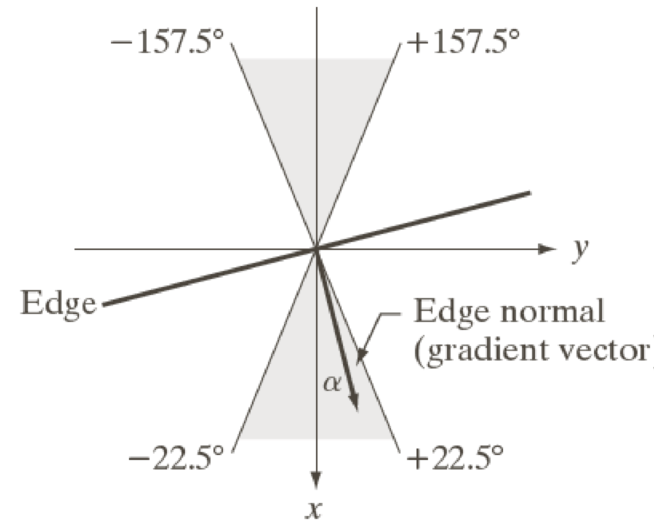
1. Find the direction d_k that is closest to $\alpha(x, y)$.
2. If the value of $M(x, y)$ is less than at least one of its two neighbors along d_k, let $g_N(x, y) = 0$ (suppression); otherwise, let $g_N(x, y) = M(x, y)$

Discrete Orientations of Gradient Vector)

- 4 basic edge directions for a 3x3 region

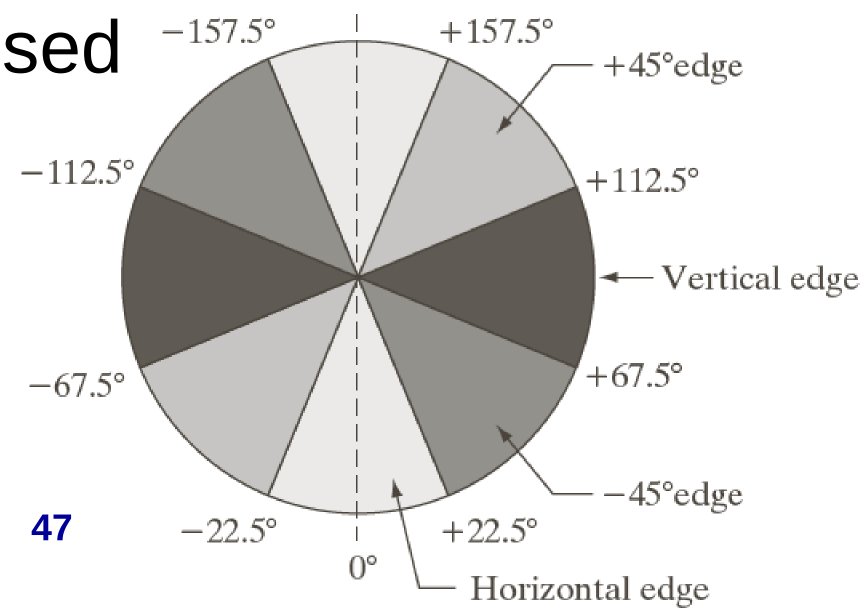
$d_1, d_2, d_3,$ and d_4





- Nonmaxima-suppressed

Image $g_N(x, y)$



Hysteresis Thresholding with Two Thresholds

- Low threshold T_L, & High threshold T_H
- "Strong" edge pixels

$$g_{NH}(x, y) = \boxed{g_N(x, y) \geq T_H}$$

- "Weak" edge pixels

$$\begin{aligned} g_{NL}(x, y) &= g_N(x, y) \geq T_L \\ g_{NL}(x, y) &= g_{NL}(x, y) - g_{NH}(x, y) \end{aligned}$$

- Form longer edge

(a) Locate the next unvisited edge pixel, p, in $g_{NH}(x, y)$.

(b) Mark as valid edge pixels all the weak pixels in $g_{NL}(x, y)$ that are connected to p using, say, 8-connectivity.

(c) If all nonzero pixels in $g_{NH}(x, y)$ have been visited go to Step d. Else, return to Step a.

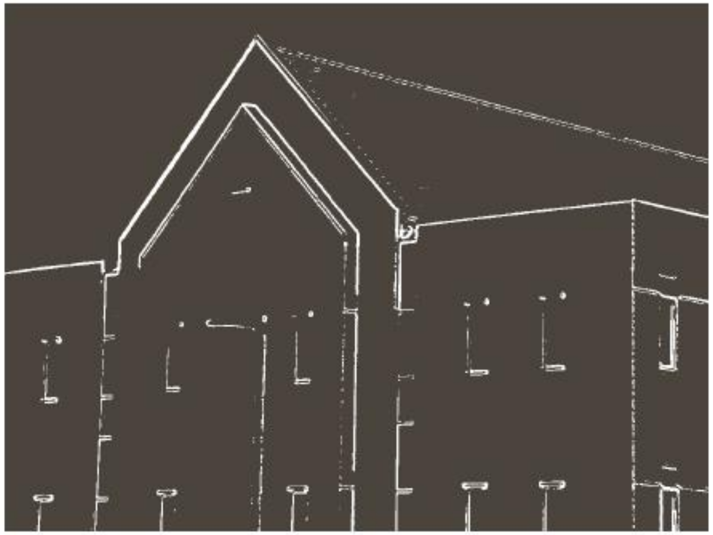
(d) Set to zero all pixels in $g_{NL}(x, y)$ that were not marked as valid edge pixels.

Canny Edge Detector

1. Smooth the input image with a Gaussian filter.
2. Compute the gradient magnitude and angle images.
3. Apply nonmaxima suppression to the gradient magnitude image.
4. Use double thresholding and connectivity analysis to detect and link edges.

Original image

Thresholded gradient of smoothed image



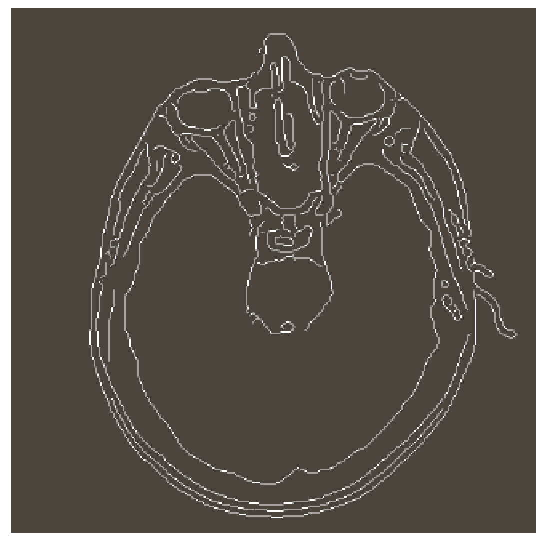
LoG

Canny

Original image

Thresholded gradient of smoothed image

LoG



Canny

Edge Linking and Boundary Detection

Method 1: local processing

- Similar in magnitude

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

- Similar in angle

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

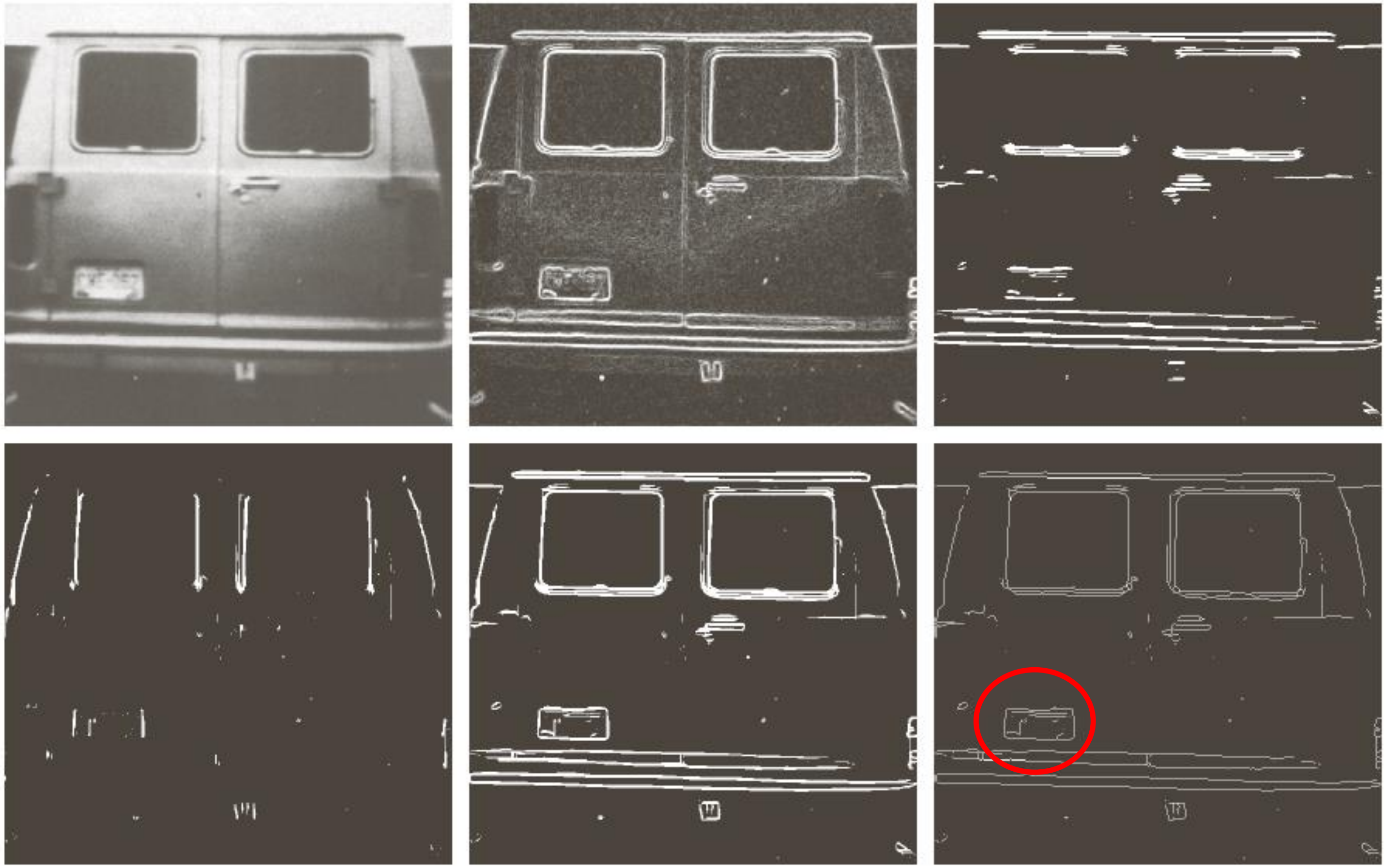
- Simplified approach:

Scan a series of directions (radial scanning)

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

Example: License Plate Extraction

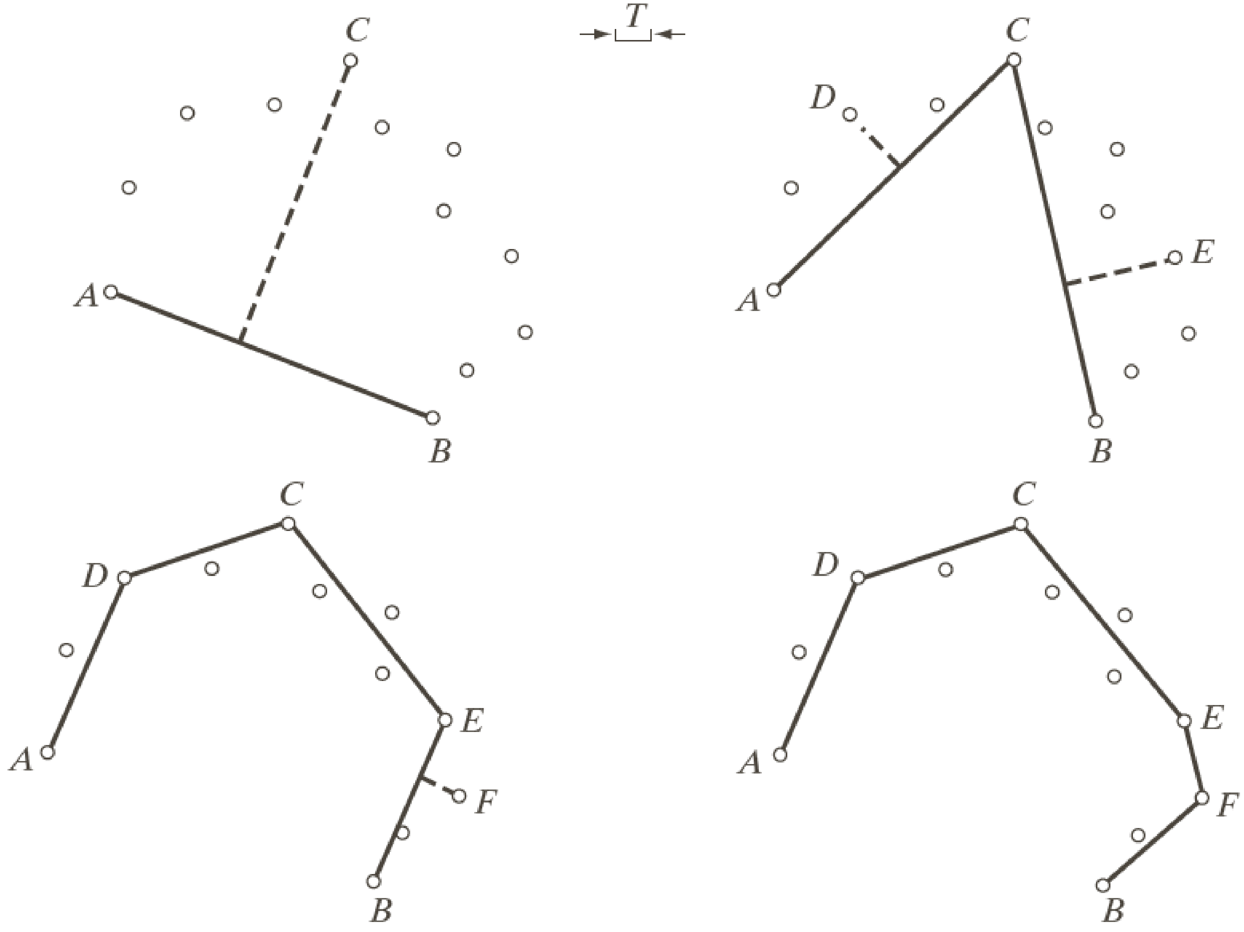
Original image Gradient magnitude Horizontally connected



Vertically connected Logical OR Morphological thinning

Method 2: Regional Processing

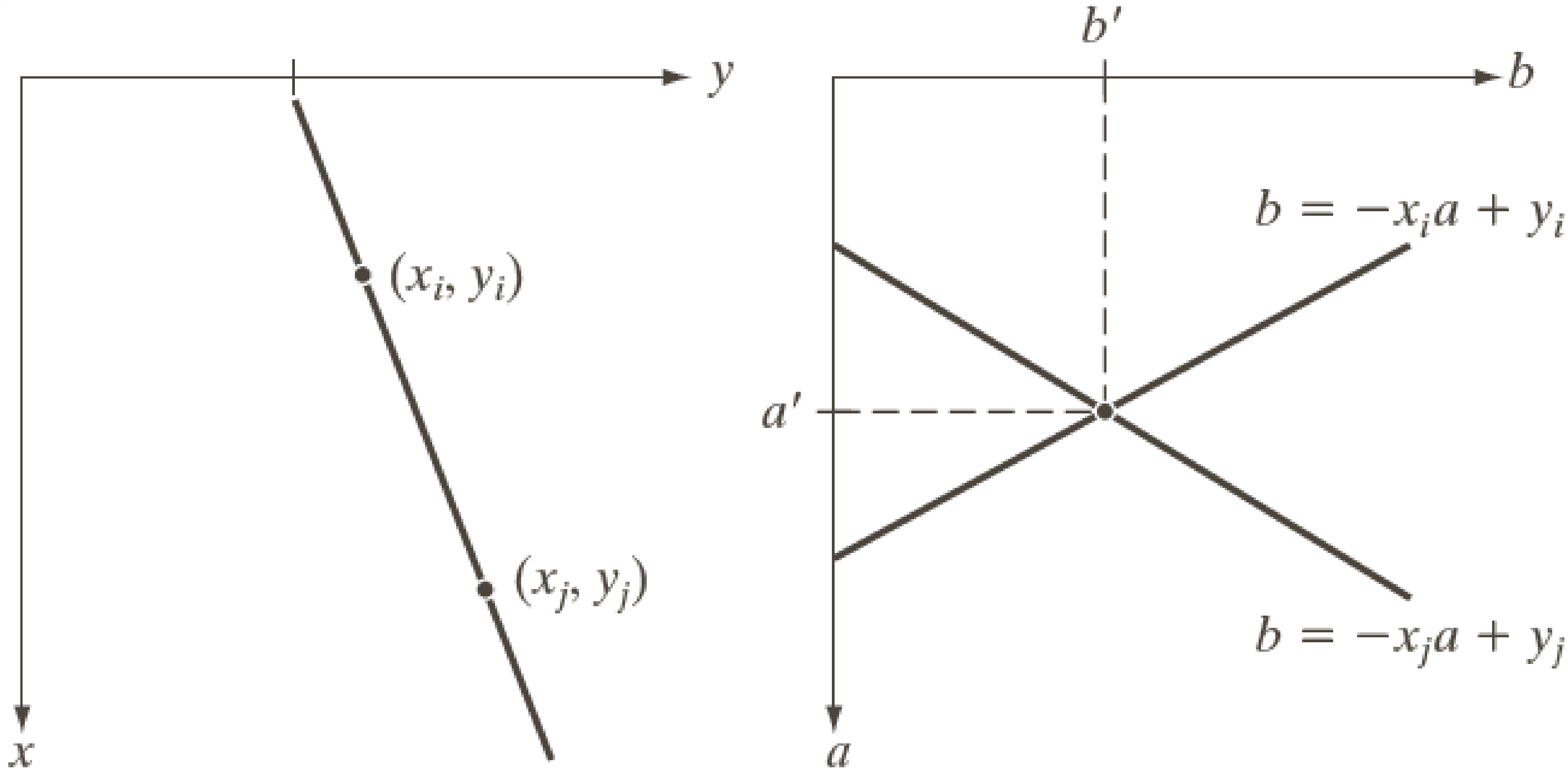
- Iterative polygonal fit algorithm



Method 3: Global Processing Using the Hough Transform

- Hough Transform of Lines

xy - plane ⇨ Parameter space



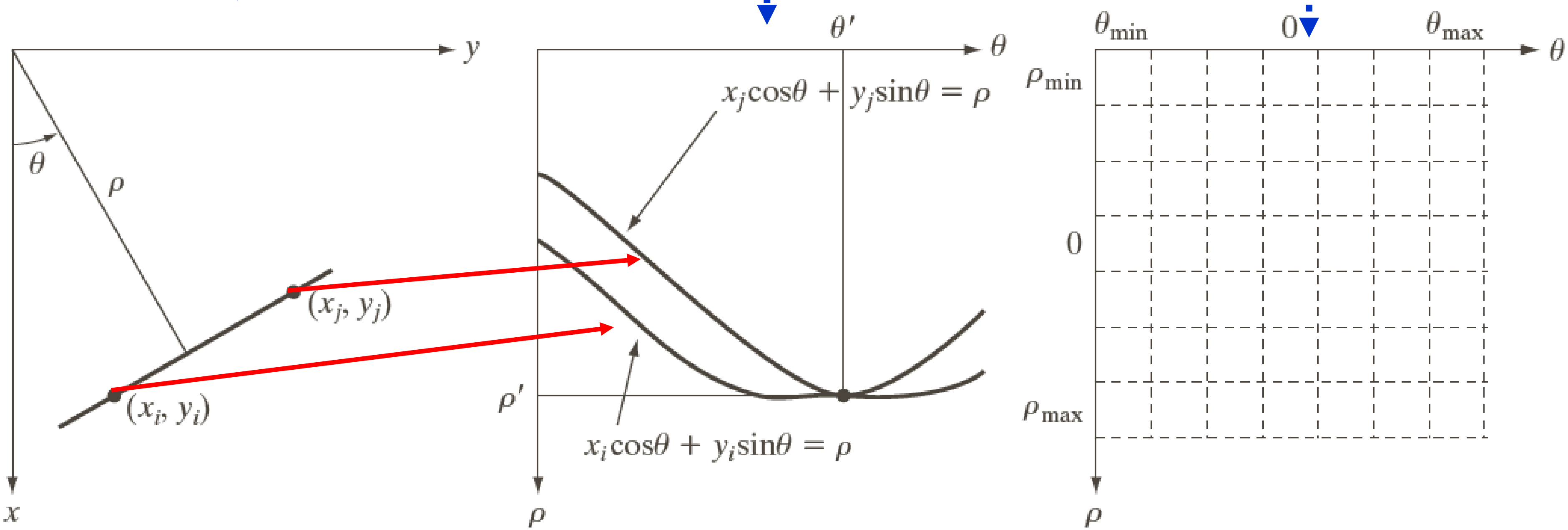
Hough Transform of Lines

(ρ, θ) parameterization of line

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

Sinusoidal curves in the $\rho\theta$-plane

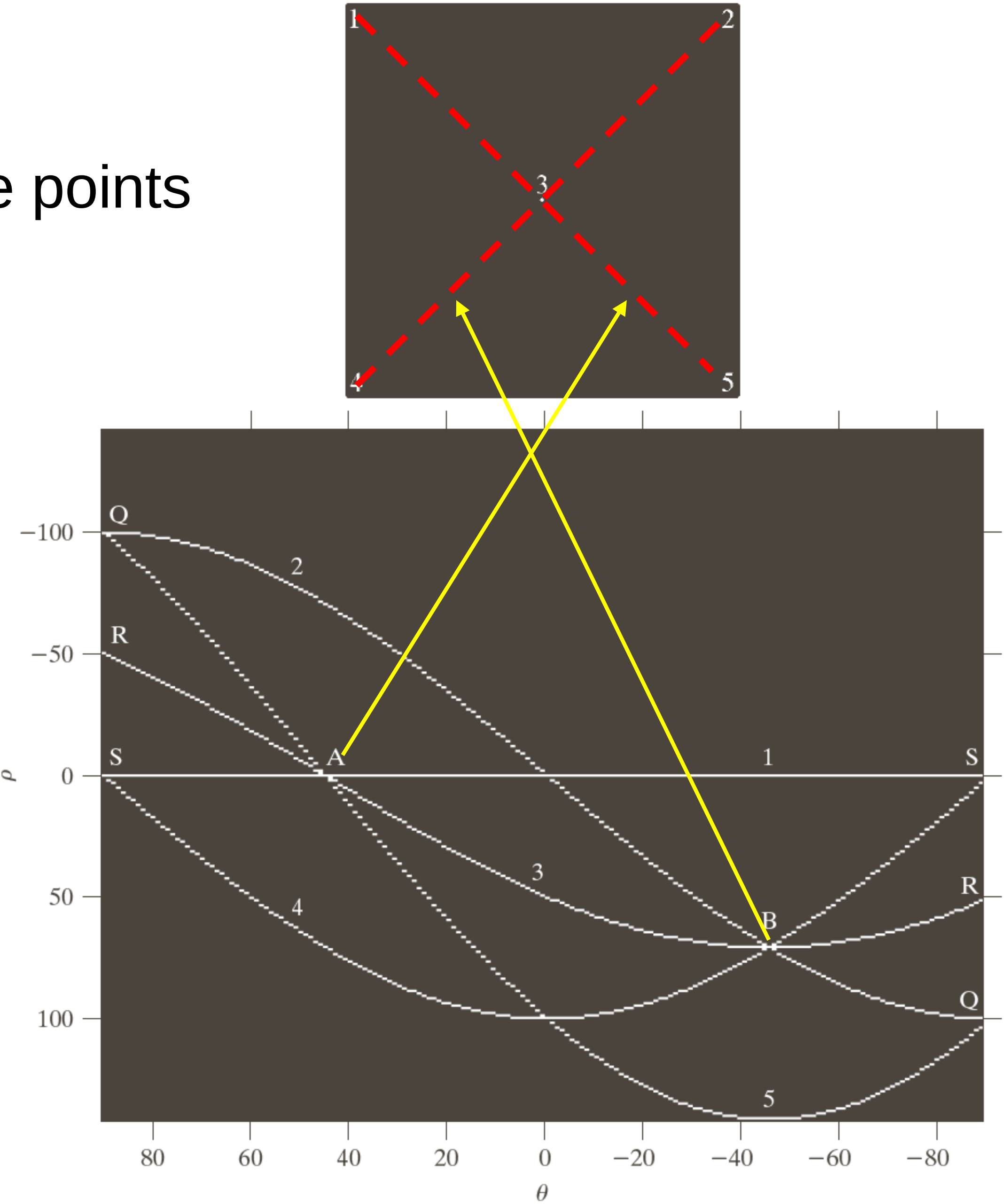
accumulator cells



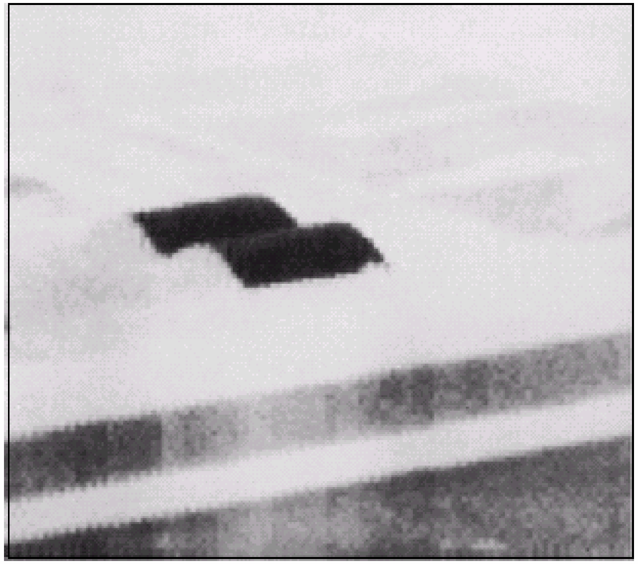
Hough Transform of Lines

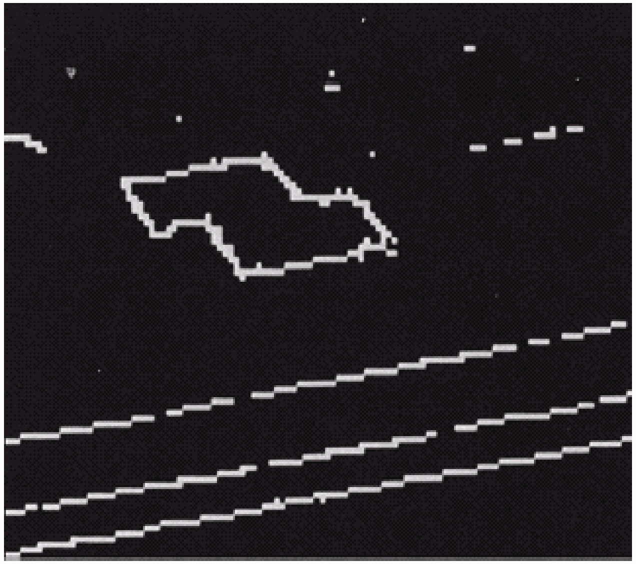
Five points

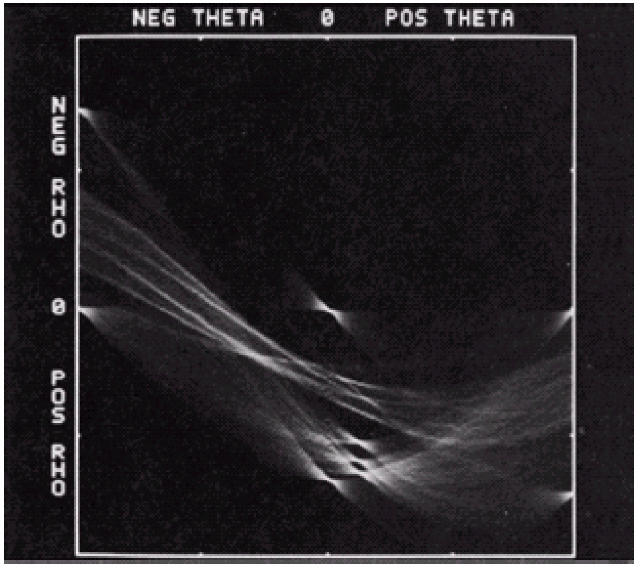
Parameter space



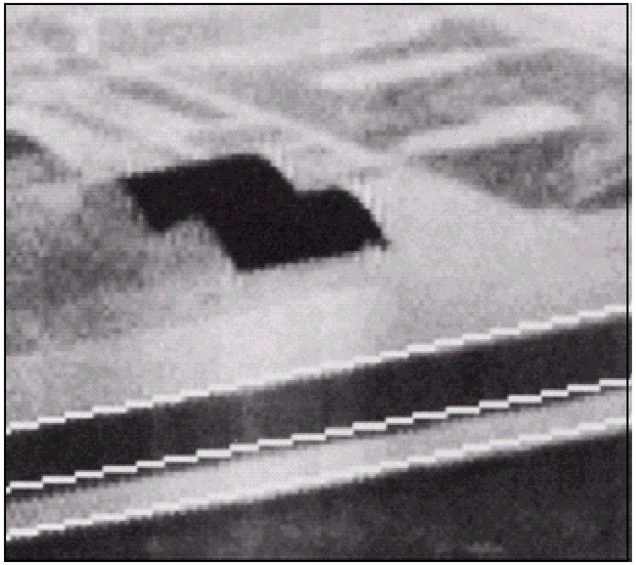
Example


(a) 一幅红外航空图像


(b) 利用Sobel算子检测到的边缘



(c) 霍夫变换结果


(d) 检测到的三条直线

霍夫变换的应用实例

Example

Airport

Canny

Parameter space

Lines

Hough Transform of Circles

- 3 parameters → 3D parameter space

$$(x-a)^2+(y-b)^2=r^2$$

- 3D → 2D: （*a*, *b*）

$$2(x-a)+2(y-b)\cdot\frac{dy}{dx}=0$$

- 3D → 2D: （*a*, *r*） or （*b*, *r*）

$$\frac{dy}{dx}=\tan\theta \Longrightarrow \begin{cases} a=x-r\cos(\theta-\frac{\pi}{2})=x-r\sin\theta \\ b=y-r\sin(\theta-\frac{\pi}{2})=y+r\cos\theta \end{cases}$$

Assignments

- 10.3, 10.6, 10.25

课后作业题目请对照参考第4版英文原版

- 第5次编程作业

从Laboratory Projects_DIP3E.pdf的Proj10-xx中选做1个题目。也可针对DIP4E Chapter 10内容，自拟任务。

Assignments

每个编程作业要求递交1份实验报告，命名“学号姓名_prjX.pdf”，内容提纲包括：

- 实验任务： 描述本次实验的任务，即所选择的ProjXX-xx题目，或自拟题目。
- 算法设计：理论上描述所设计的算法。
- 代码实现：描述编程环境，给出自己编写的核心代码。
- 实验结果：描述具体的实验过程，给出每个小实验的输入数据、算法参数和实验结果，并对结果做简要的讨论。
- 总结：简要总结本次实验的技术内容，以及心得体会