Introduction to Digital Image Processing

Dr. Kundan Kumar
Ph.D. (IIT Kharagpur)
Associate Professor
ECE Department (Cabin - E139)

Institute of Technical Education & Research (ITER)
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India-751030

Image Segmentation: Part I

Introduction

Image processing

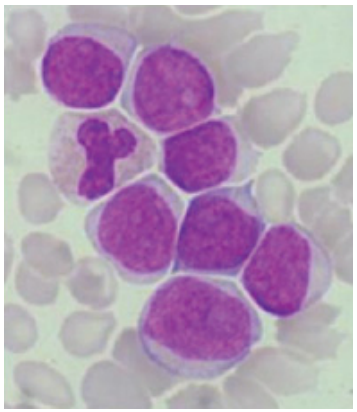
Any form of signal processing for which the input is an image. Output may be either an image, a set of characteristics or parameters related to the image.

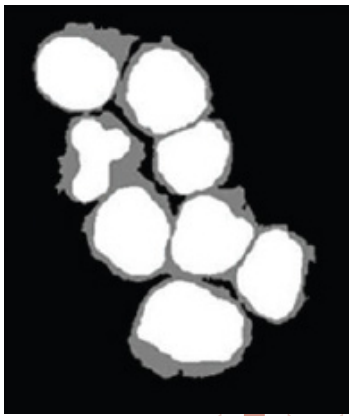
Ex. Point-wise operations, local neighborhood operations, Fourier transform.

Image analysis

Extraction of meaningful information from images.

Ex. Image segmentation





Introduction

- Dividing the image into different regions based on some similarity or discontinuity criteria.
- Separating objects from background and giving them individual ID numbers (labels).
- Segmentation is often the most difficult problem to solve in image analysis.
- The problem can be made much easier if solved in cooperation with the constructor of the imaging system (choice of sensors, illumination, background, etc.)
- There is no universal solution.

Fundamental Definition

- Let R is the entire spatial region occupied by an image then the image segmentation is the process of partitioning the region R in sub-regions $R_1, R_2, \ldots, R_n$, such that

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

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

  where $Q(R_k)$ is a logical predicate defined over the points in set R_k, and $\varnothing$ is the null set.

Segmentation

Segmentation algorithms for monochrome image are often based on one of the following two basic properties of intensity values:

Discontinuity

- Detecting boundaries of regions based on local discontinuity in intensity.
- Called *Edge-based segmentation*

Similarity

- Partitioning an image into regions that are similar according to a set of predefined criteria.
- Called *Region-based segmentation*.

Segmentation

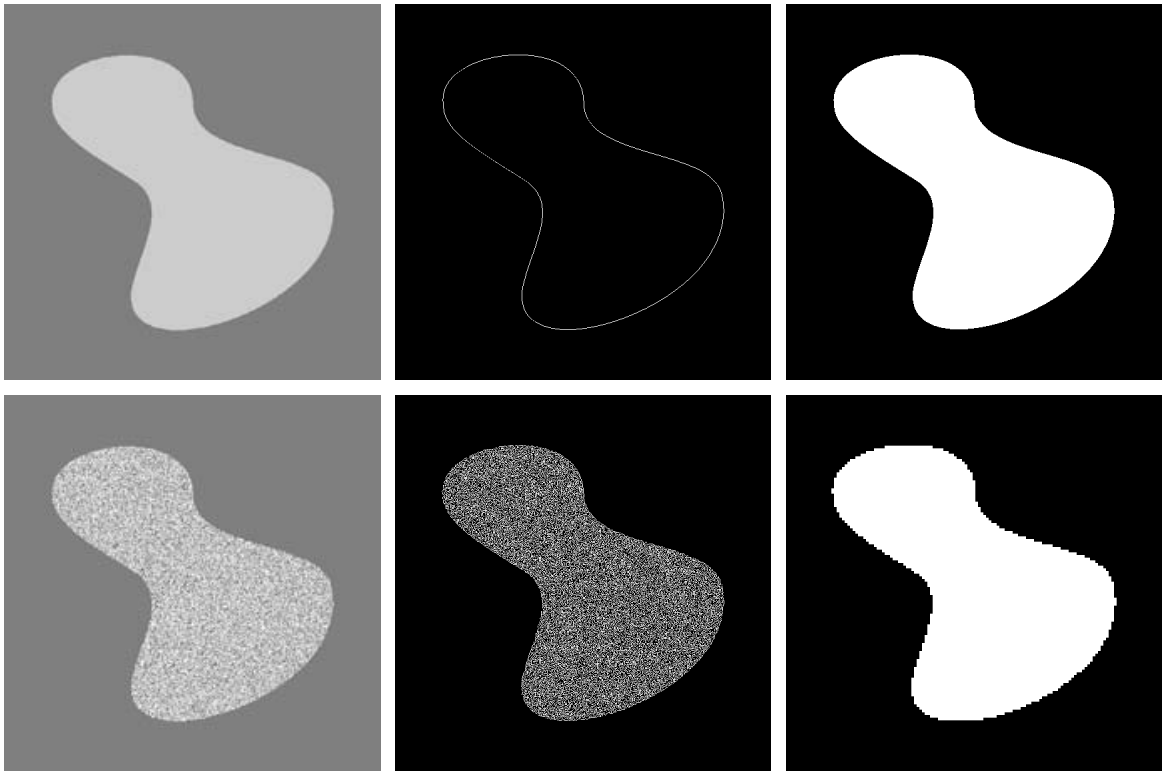


Figure: (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Detection of Discontinuities

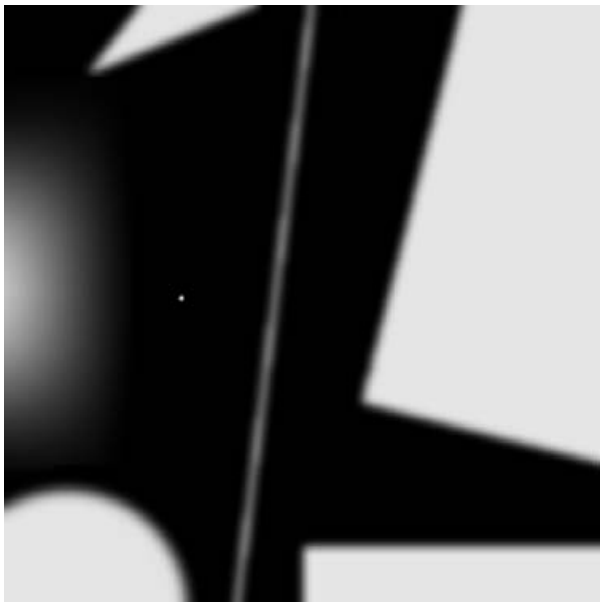
- Based on detecting sharp, local changes in intensity.
- Three type of image features
  - Point
  - Line
  - Edge

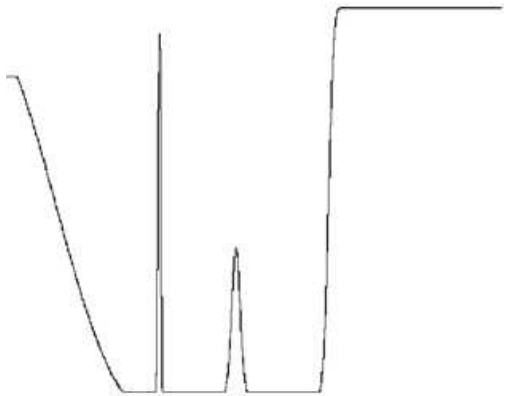
- Approximation for first and second derivatives

$$\frac{df}{dx} \approx \begin{cases} f(x+1,y) - f(x,y) \\ f(x,y) - f(x-1,y) \\ 0.5(f(x+1,y) - f(x-1,y)) \end{cases}$$

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

Detection of Discontinuities





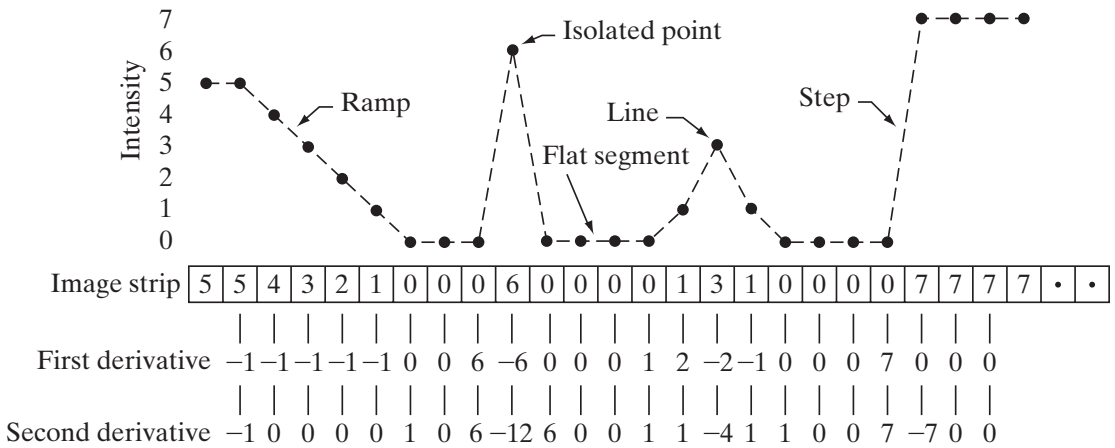


Figure: (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

Point Detection

- First Derivative
  - Thicker Edge;
  - Strong Response for step changes;
- Second Derivative
  - Strong response for fine details and isolated points
  - Double response at step changes

Isolated point detection

1. Compute $R(x, y) = \sum_{i=1}^{9} w_i z_i$
2. Apply thresholding

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

Point Detection

1	1	1
1	−8	1
1	1	1

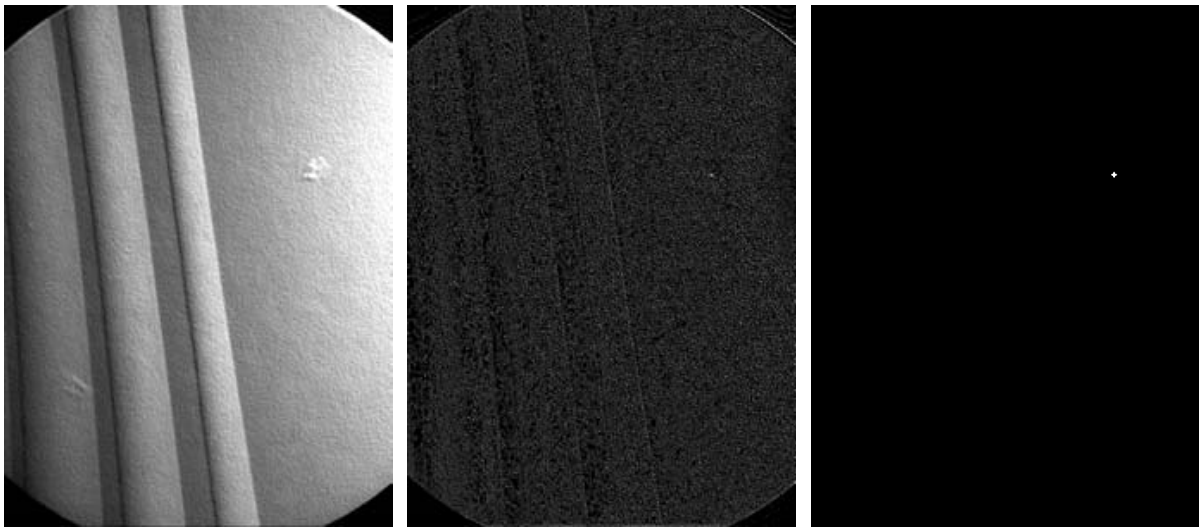


Figure: (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) A single point detection using thresholding

Line Detection

- Let R_1, R_2, R_3, and R_4 denote the responses of the masks from left to right.
- If at a given point in the image,

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

that point is said to be more likely associated with a line in the direction of mask k.

−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

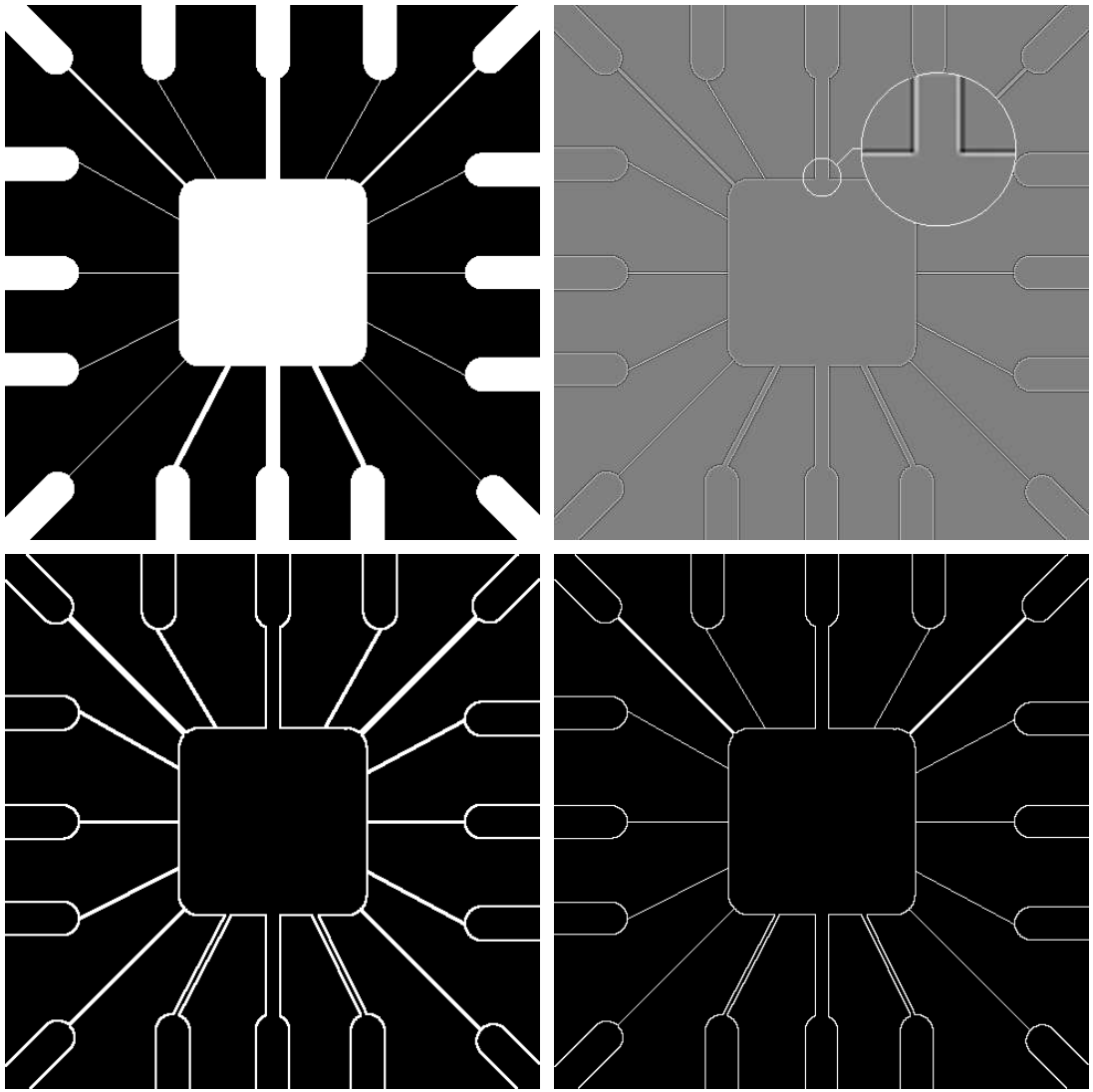
Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

- Alternatively, to detect lines in a specified direction use the mask associated with that direction and threshold its output.

Line Detection: Laplacian operator example



a b
c d

FIGURE 10.5
(a) Original image. (b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian. (c) Absolute value of the Laplacian. (d) Positive values of the Laplacian.

Line Detection: Laplacian operator example

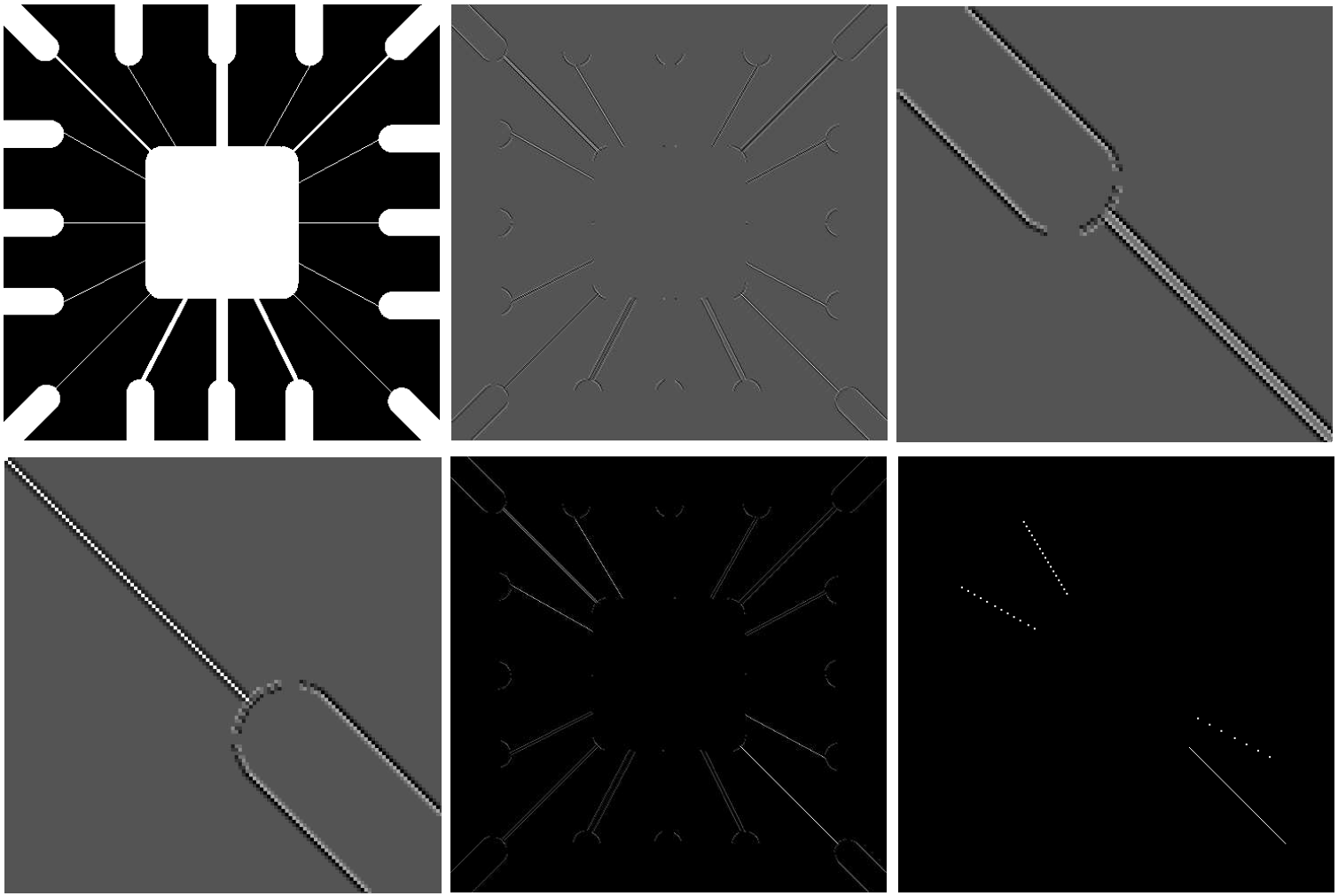


Figure: Processing with $+45^o$ line detector mask

Edge Detection

- Most frequently used for segmenting images based on abrupt (local) changes in intensity.
- Three mathematical models to detect edge
  1. Step edge
  2. Ramp edge
  3. Roof edge

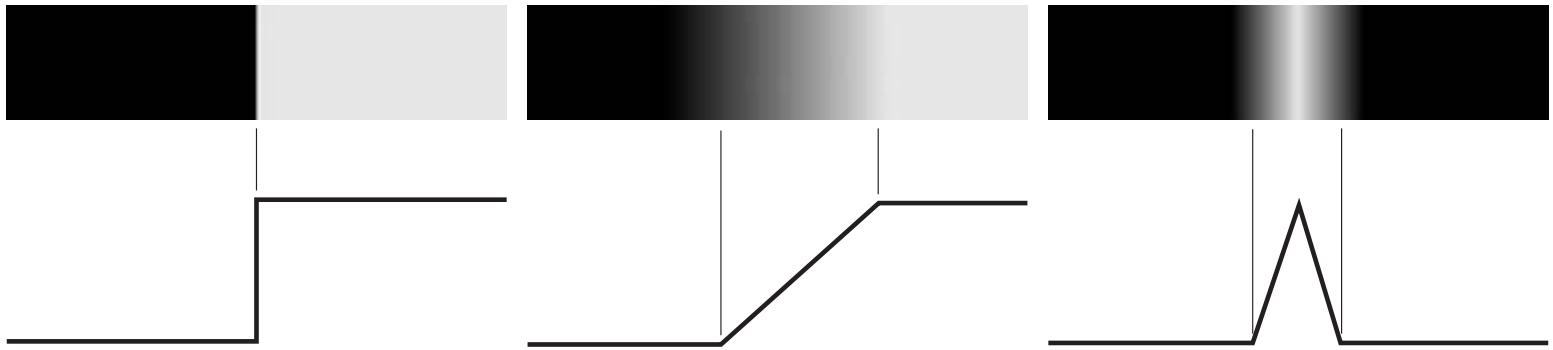
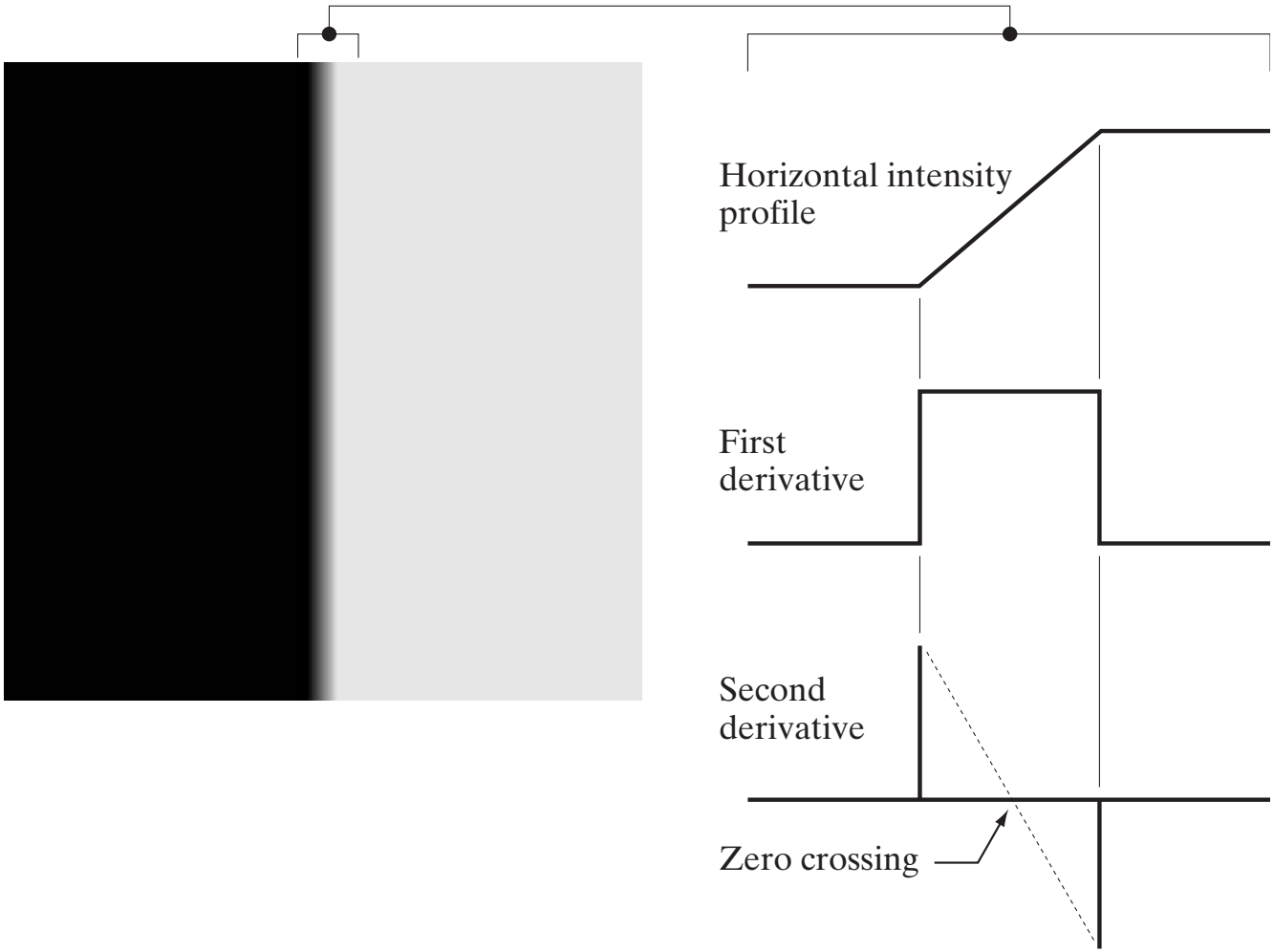


Figure: From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding *intensity profiles*.

Ramp Edge Model



a b

FIGURE 10.10
(a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Effect of noise on Ramp Edge Models

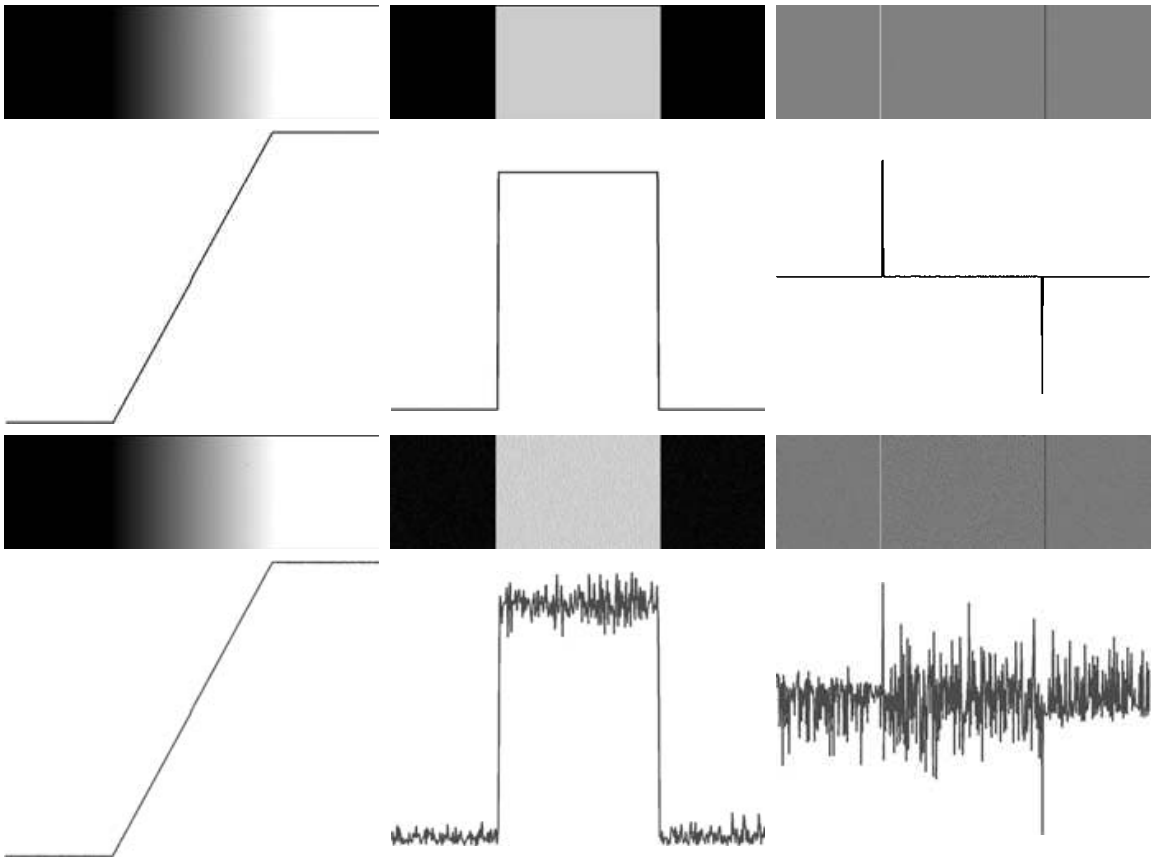


Figure: *First column*: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0 and 0.1 intensity levels, respectively. *Second column:* First-derivative images and intensity profiles. *Third column:* Second-derivative images and intensity profiles

Effect of noise on Ramp Edge Models

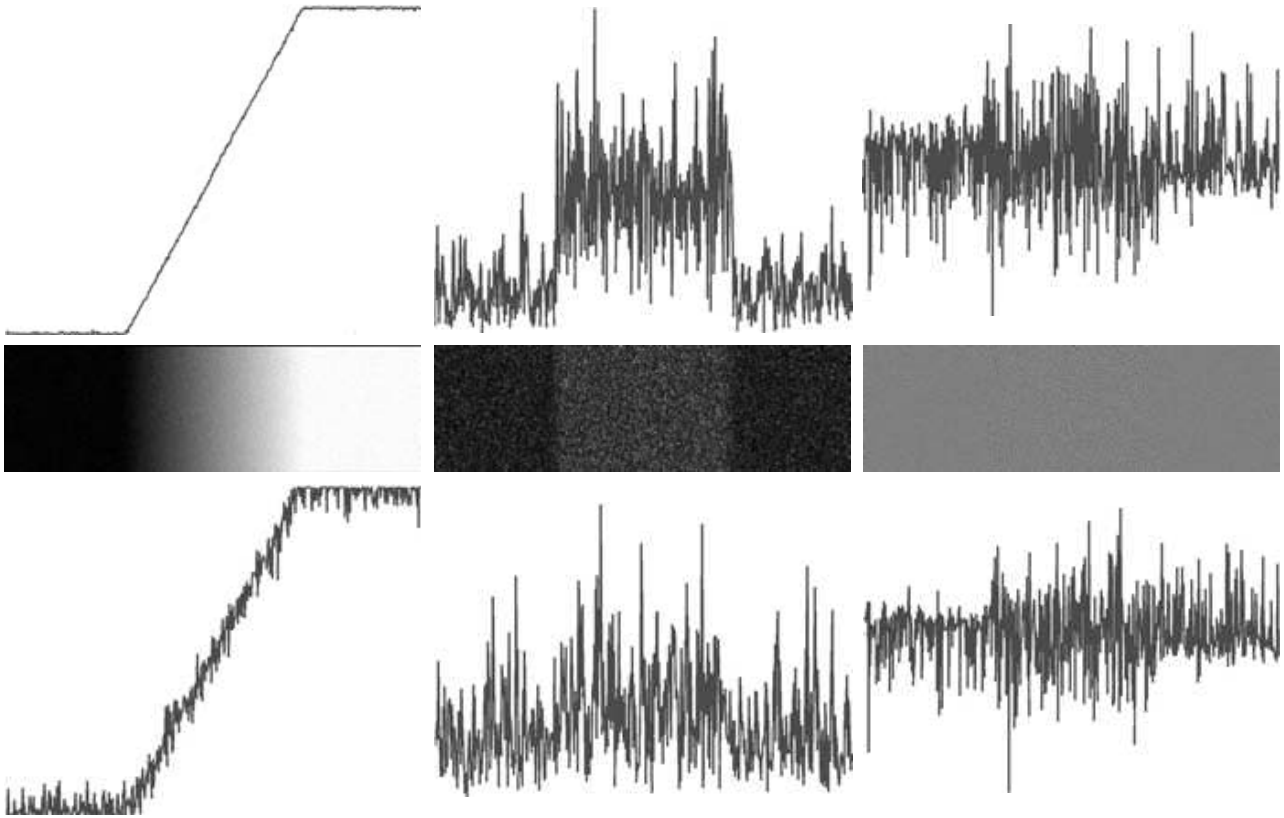


Figure: *First column:* Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 1.0 and 10.0 intensity levels, respectively. *Second column:* First-derivative images and intensity profiles. *Third column:* Second-derivative images and intensity profiles

Fundamental steps for Edge Detection

1. *Image smoothing*: This step is needed to reduce the noise.

2. *Detection of edge points*: This is a local operation that extracts from an image all points that are potential candidates to become edge points.

3. *Edge localization*: The objective of this step is to select from the candidate edge points only the points that are true members of the set of points comprising an edge.

Basic Edge Detection

Image gradient and its properties

- The tool for finding edge strength and direction at location (x, y) of an image, f, is the gradient, denoted by ∇f, and defined as the vector

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

This vector has the important geometrical property that it points in the direction of the greatest rate of change of f at location (x, y).

Basic Edge Detection

Image gradient and its properties

- The magnitude of vector ∇f, denoted as $M(x, y)$, as

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

  The magnitude value of the rate of change in the direction of the gradient vector.

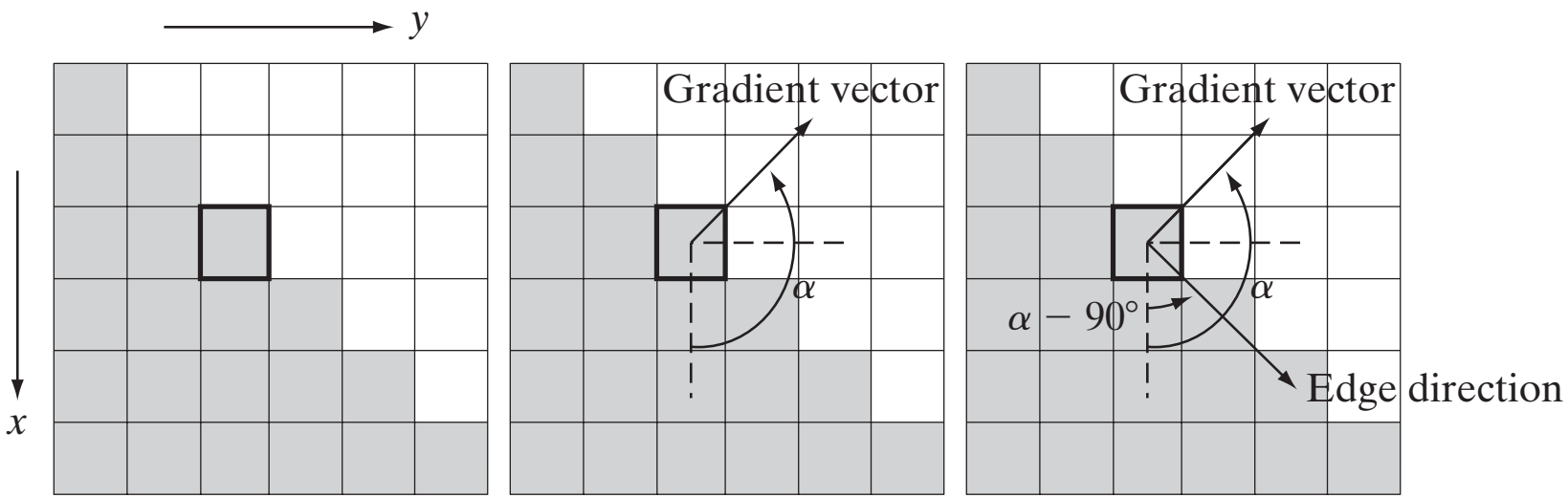
- The direction of the gradient vector is given by the angle

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

  measured with respect to the x-axis.

- The gradient image, $\alpha(x, y)$ also is an image of the same size as the original

Basic Edge Detection



a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Basic Edge Detection

Gradient operators:

- Roberts
- Prewitt
- Sobel

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

Basic Edge Detection

Prewitt and Sobel masks for detecting diagonal edges.

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

Prewitt

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

Sobel

Basic Edge Detection: Example

a b
c d

FIGURE 10.16
(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

Basic Edge Detection: Example

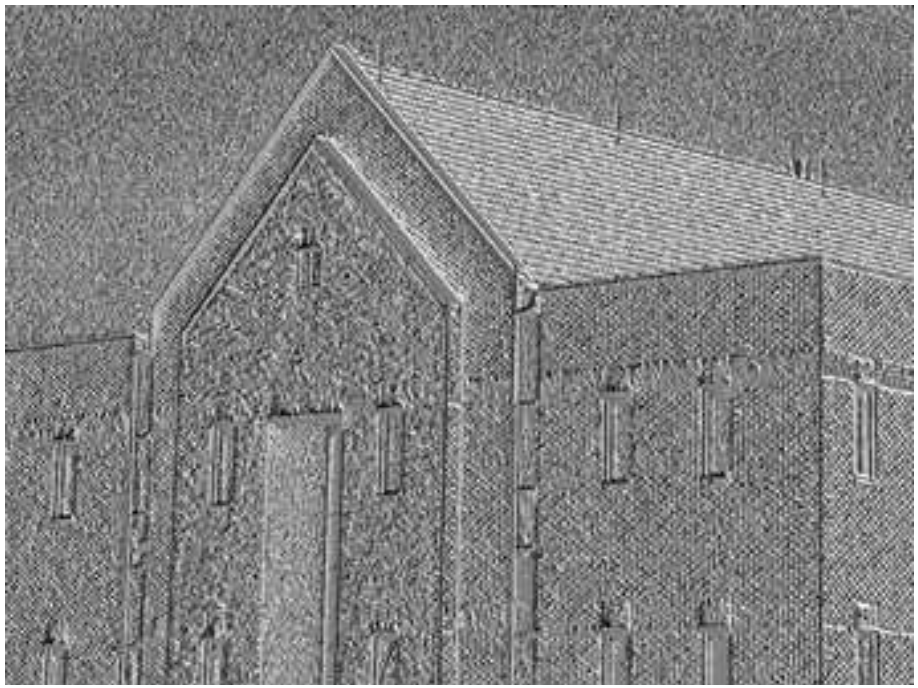
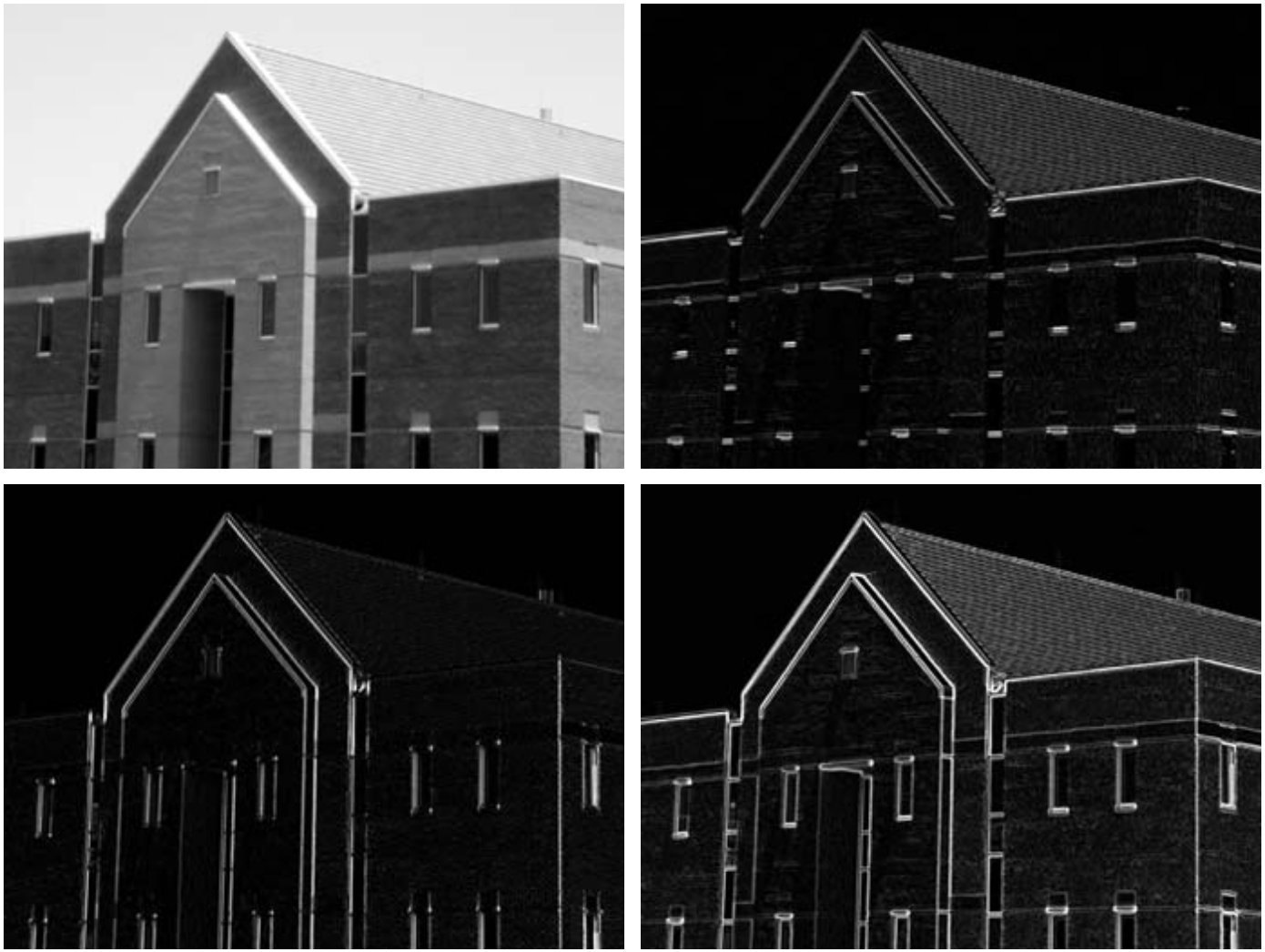


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

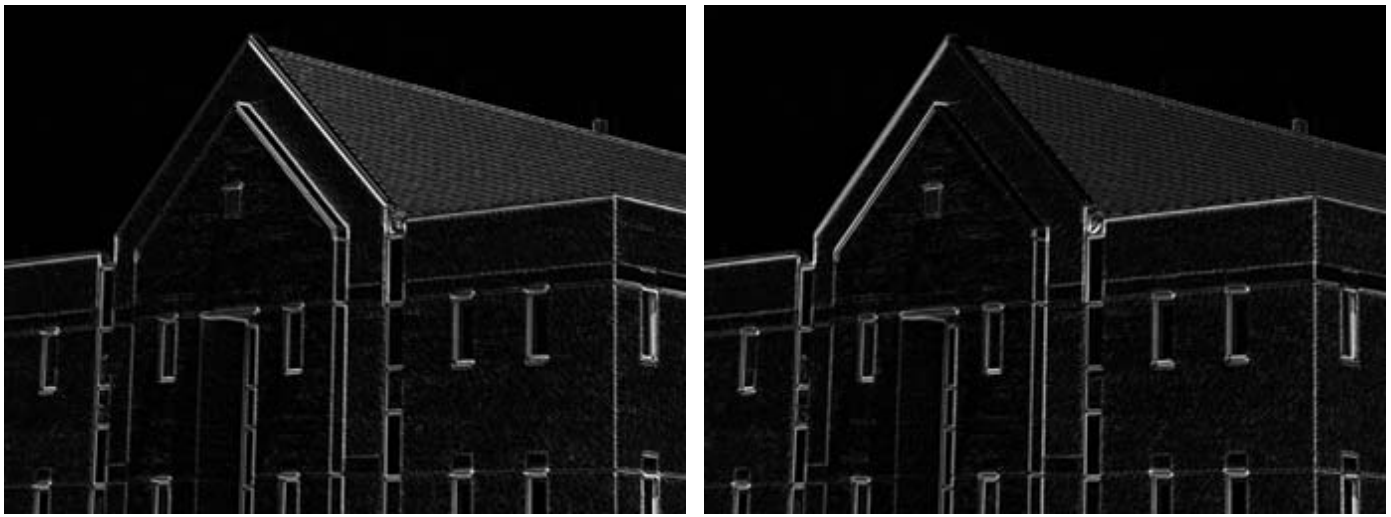
Basic Edge Detection: Pre-smoothing effect

a b
c d

FIGURE 10.18
Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.



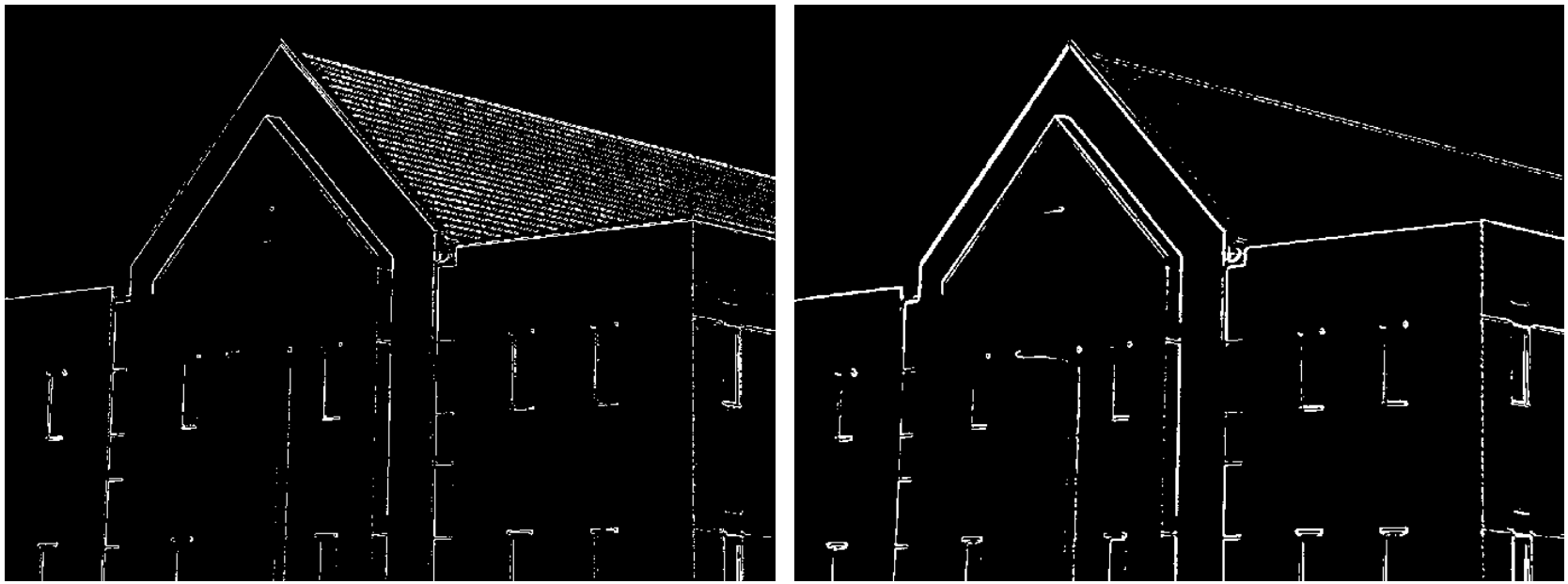
Basic Edge Detection: Pre-smoothing effect



a b

FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

Basic Edge Detection: After thresholding



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

More Advanced Techniques for Edge Detection

- In previous approaches, there was no provisions being made for edge characteristics and noise content.
- Edge detection can be improved by taking into account factors such as *image noise* and the *nature of edge themselves*.
- *Marr and Hildreth* have suggested
  - Intensity changes are not independent of image scale and so their detection requires the use of operators of different sizes
  - Sudden change will give rise to a peak in first derivative or, equivalently, to a zero crossing in the second derivative.
- Most satisfactory fulfilling these characteristics is the *Laplacian of Gaussian* (LoG) operator ($\nabla^2 G$)

Laplacian of Gaussian

- Gaussian function is defined as

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

- The Laplacian of gaussian obtained as

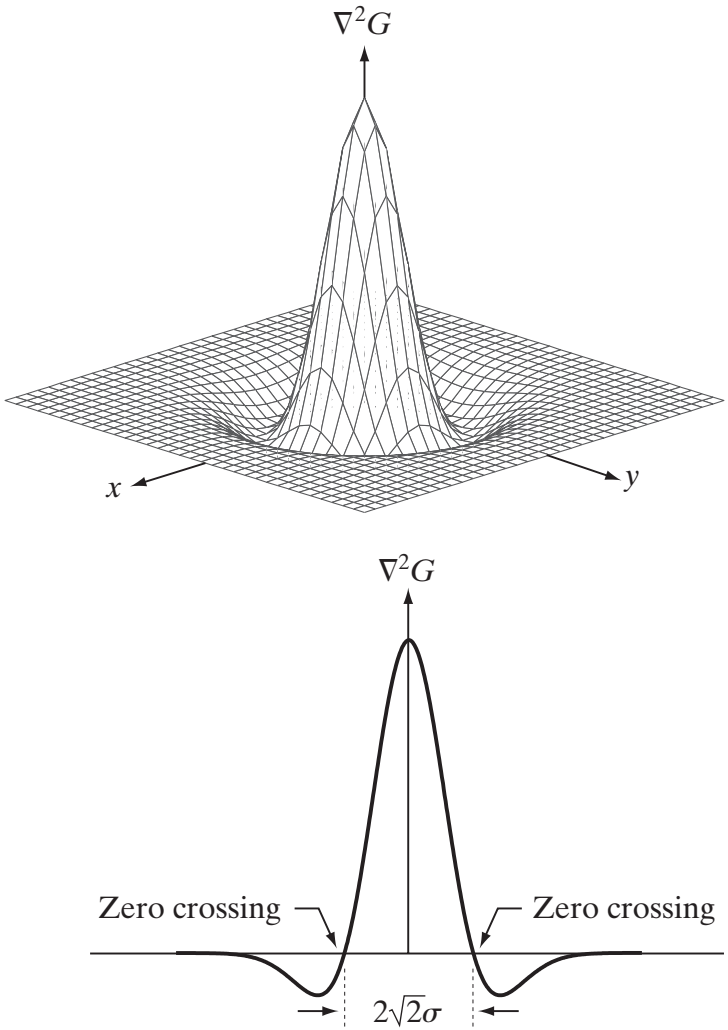
$$\nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2}$$

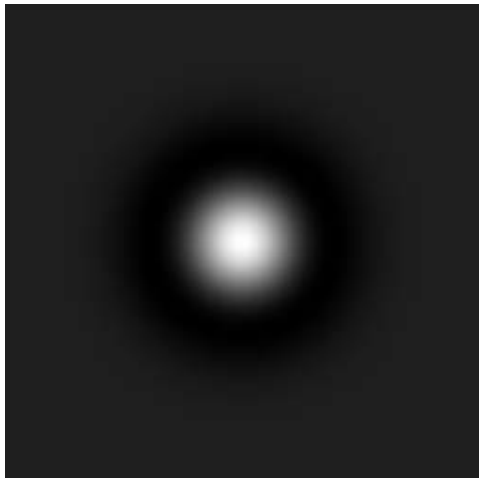
$$= \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

Laplacian of Gaussian





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.

The Marr-Hildreth Edge Detector:

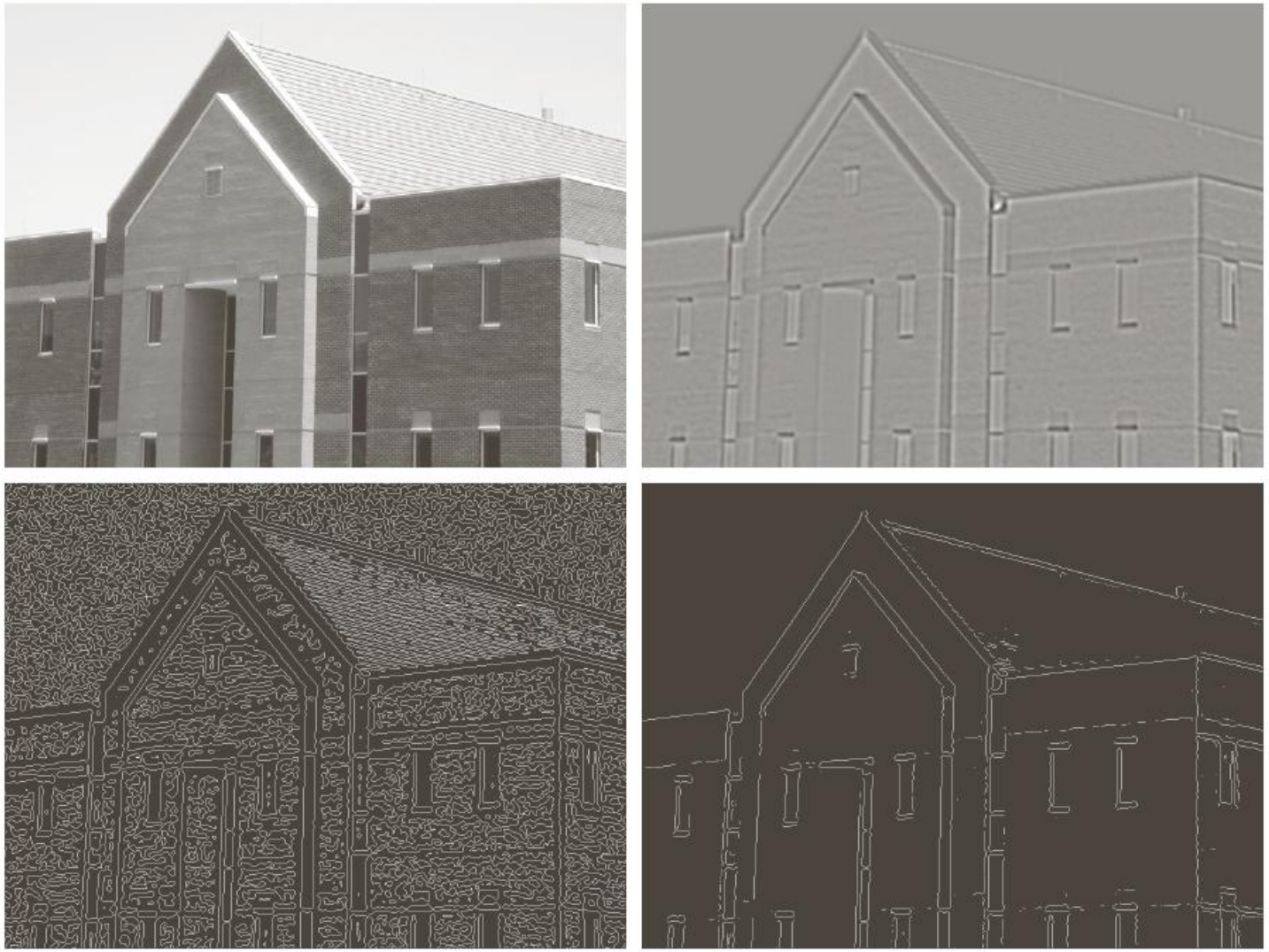
- Idea
  - Smoothing
  - Sharpening

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

- *Algorithm*
  1. Filter the input image with an $n \times n$ Gaussian lowpass filter.
  2. Compute the Laplacian of the image resulting from Step 1.
  3. Find the zero crossing of the image from Step 2 to get the edge.
- Marr and Hildreth noted that it is possible to approximate the LoG filter by *difference of Gaussians* (DoG)

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

The Marr-Hildreth Edge Detector



a b
c d

FIGURE 10.22
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Canny Edge Detector

- The algorithm is more complex but the performance of the Canny edge detection is better than previous approaches.
- Three basic objective:
  - Low error rate
  - Edge points should be well localized
  - Single edge point response
- Canny Edge Detector Steps:
  - Smoothing
  - Compute Gradients
  - Non-maximum Suppression
  - Edge Tracking by hysteresis (double) thresholding.

Canny Edge Detector: Smoothing

- Using Gaussian filter:

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

$$G(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \quad \sigma = 1.4$$

- Practical Implementation:

$$G(x,y) = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Canny Edge Detector: Gradient

- Using any gradient kernel (Sobel or prewitt)

$$g_x(x,y) = \mathrm{w}_x(x,y) \bigstar f_s(x,y)$$

$$g_y(x,y) = \mathrm{w}_y(x,y) \bigstar f_s(x,y)$$

- Practical Implementation

$$\mathrm{w}_x(x,y) = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad \mathrm{w}_y(x,y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

- Compute magnitude and angle of gradient

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

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

Canny Edge Detector: Non-maximum Suppression

- Next step is to thin those ridges.
- One approach is to use *nonmaxima suppression*.
- Specify a number of discrete orientations of the edge normal (gradient vector).

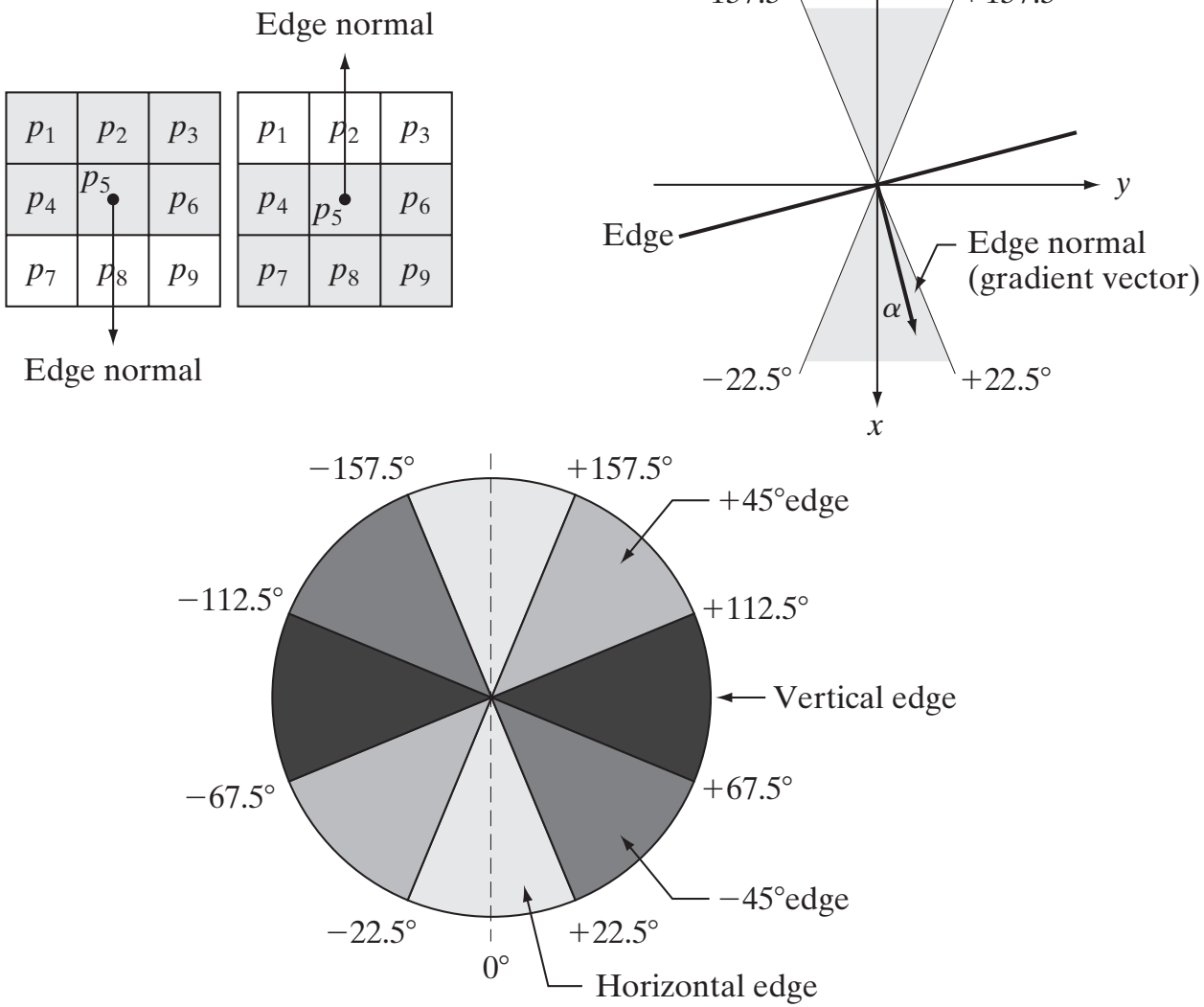
Algorithm:

Let d_1, d_2, d_3, and d_4 denote the four basic edge directions: horizontal, -45^o, vertical, and $+45^o$.

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 $g_N(x, y) = M(x, y)$.

Where $g_N(x, y)$ is the nonmaxima-suppressed image.

Canny Edge Detector: Non-maximum Suppression



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 Edge Detector: Edge Tracking

- Select two threshold (T_H, T_L)
- Form two images using two threshold:

$$g_{NH}(x,y) = \begin{cases} 1 & g_N(x,y) \geqslant T_H \\ 0 & g_N(x,y) < T_H \end{cases}$$ Fewer and Strong Edge

$$g_{NL}(x,y) = \begin{cases} 1 & g_N(x,y) \geqslant T_L \\ 0 & g_N(x,y) < T_L \end{cases}$$ More and Weak/Strong Edge

- Eliminate strong edge from g_{NL}

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

- The nonzero pixel in $g_{NH}(x,y)$ and $g_{NL}(x,y)$ may be viewed as being "strong" and "week" edge pixels, respectively.

Canny Edge Detector: Edge Tracking

- All strong edge in $g_N H$ are store and marked immediately.
- Gaps in g_{NH} with fill using g_{NL}
  1. Locate the next unvisited edge pixel, p, in $g_{NH}(x, y)$
  2. Mark as valid edge pixel all the weak pixels in $g_{NL}(x, y)$ that are connected to p using $8-$connectivity.
  3. If all nonzero pixels in $g_{NL}(x, y)$ have been visited go to Step 4 else, return to Step 1.
  4. Set to zero all pixels in $g_{NL}(x, y)$ that were not marked as valid edge pixels.

Canny Edge Detector: Example 01

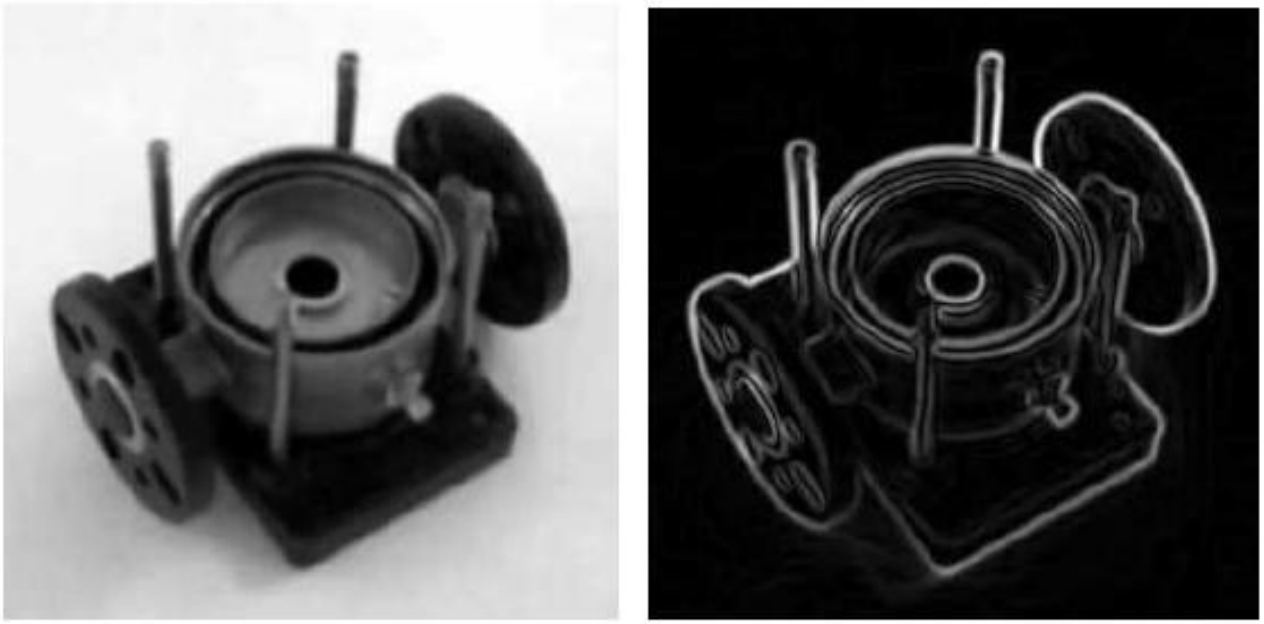
– Original and Smoothed

(a) Original

(b) Smoothed

Canny Edge Detector: Example 01

– Gradient images:

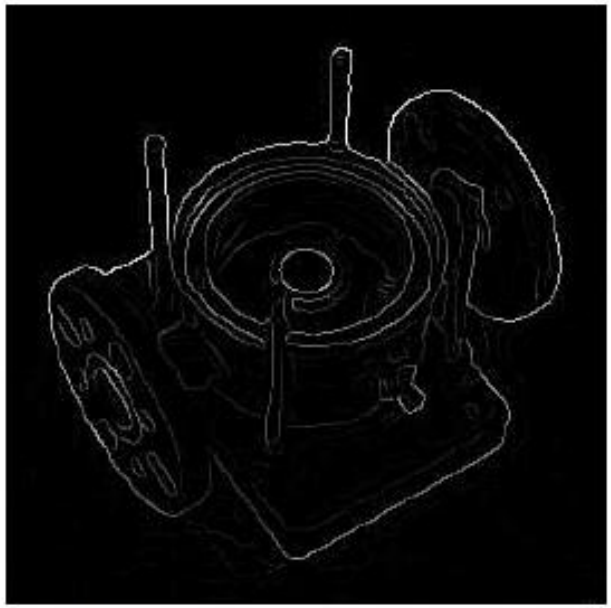


(a) Smoothed (b) Gradient magnitudes

Canny Edge Detector: Example 01

– Non-maximum suppression:

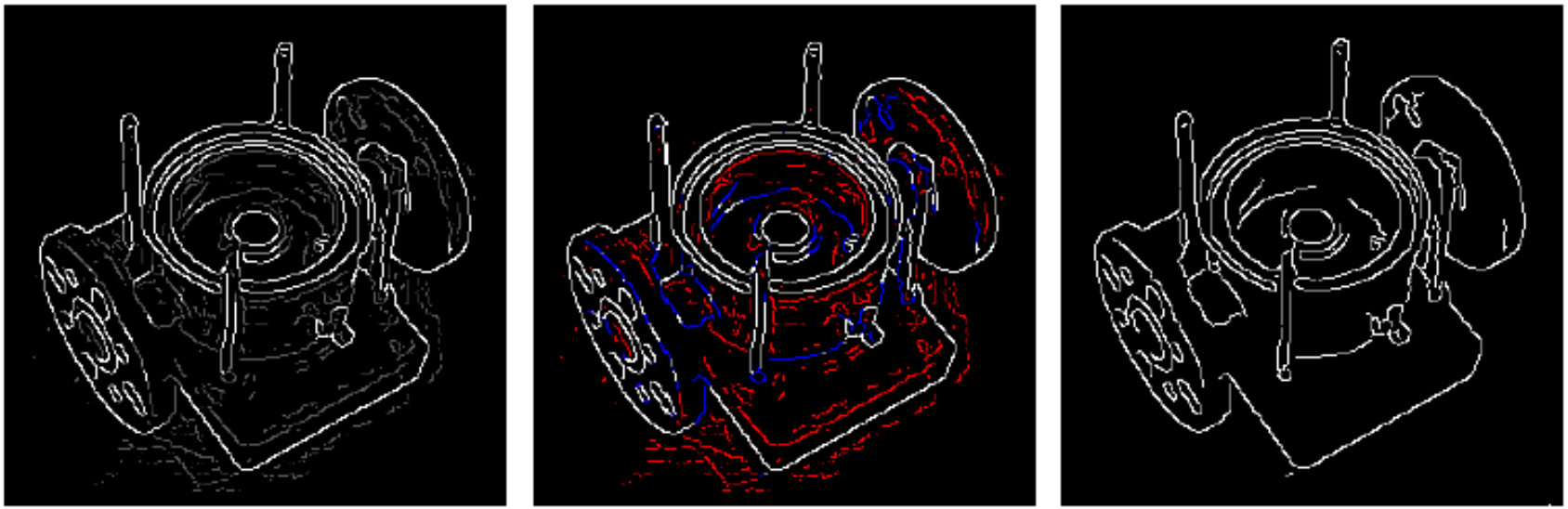
(a) Gradient values



(b) Edges after non-maximum suppression

Canny Edge Detector: Example 01

– Edge Tracking:



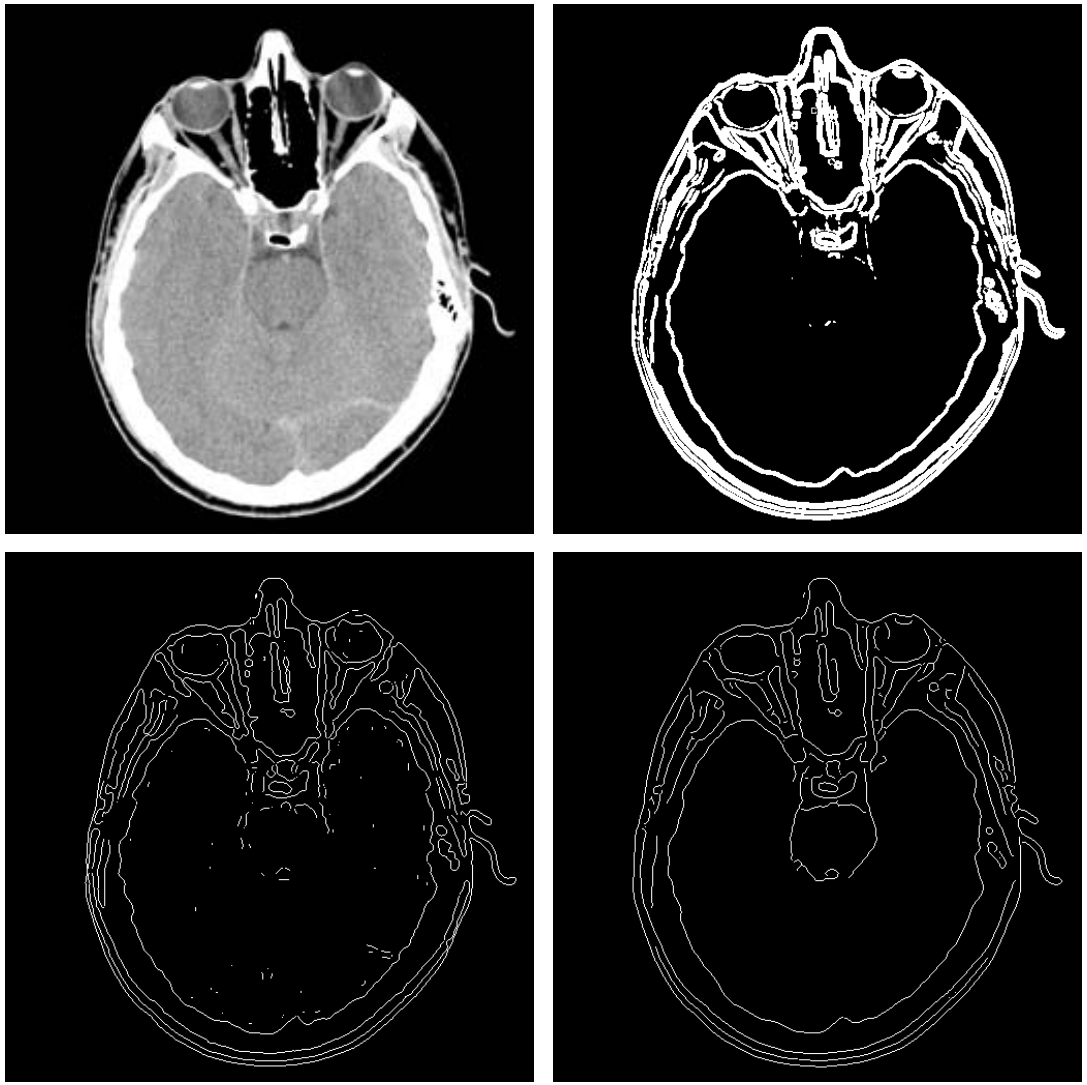
(a) Double thresholding (b) Edge tracking by hysteresis (c) Final output

Canny Edge Detector: Example 02

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.

Canny Edge Detector: Example 03



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.)


Thank You
Queries?