Image Segmentation

Line & Edge Detection

Fundamentals

- Let R represent the entire spatial region occupied by an image. Image segmentation is a process that partitions R into *n* sub-regions, R_1, R_2, …, R_n, such that

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

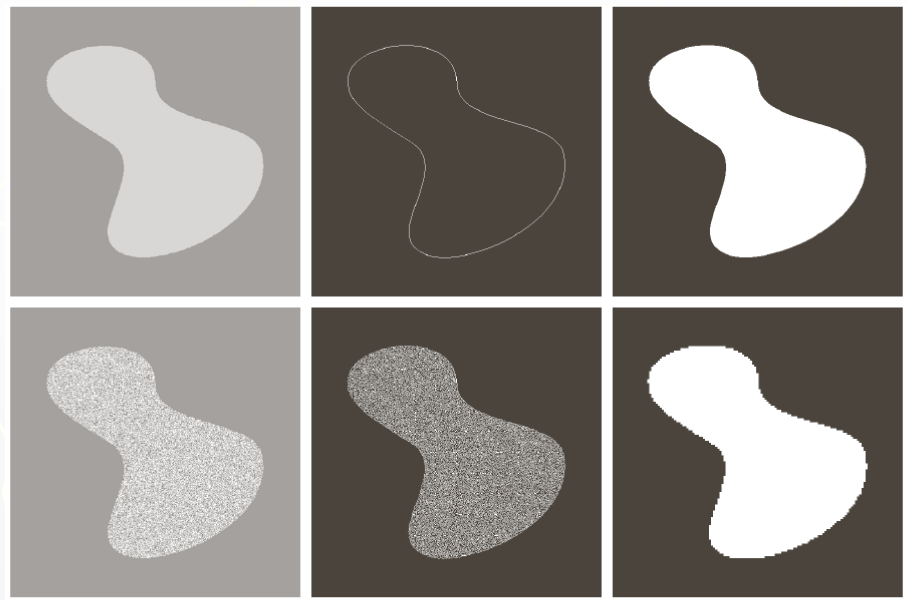
(b) R_i is a connected set. $i = 1,\ 2,\ ...,\ n.$

(c) $R_i \cap R_j = \Phi.$

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

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

R_i and R_j.



a	b	c
d	e	f

FIGURE 10.1 (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.

Characteristics of First and Second Order Derivatives

- First-order derivatives generally produce thicker edges in image
- Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
- Second-order derivatives produce a double-edge response at ramp and step transition in intensity
- The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light

Detection of Isolated Points

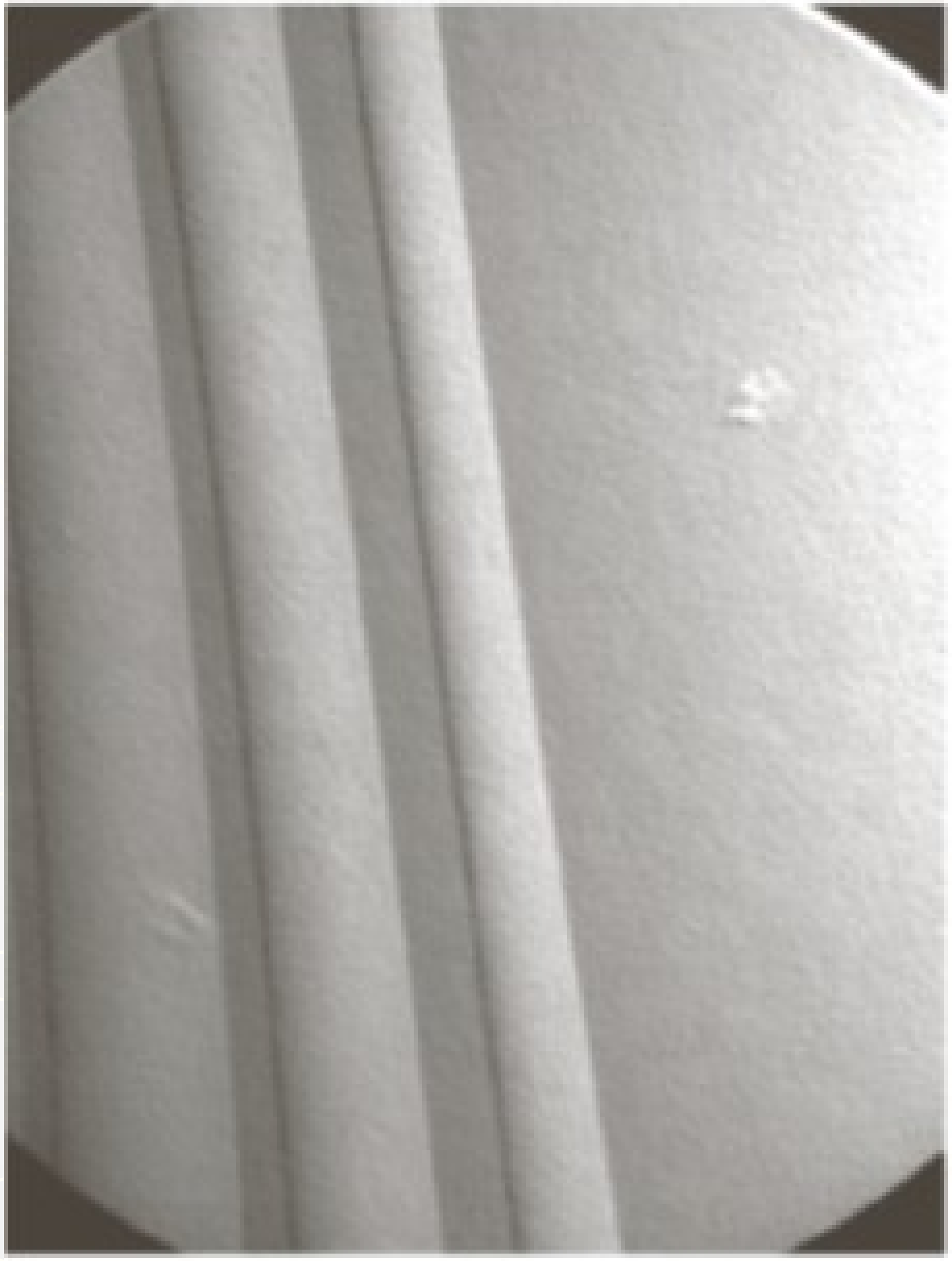
- The Laplacian

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

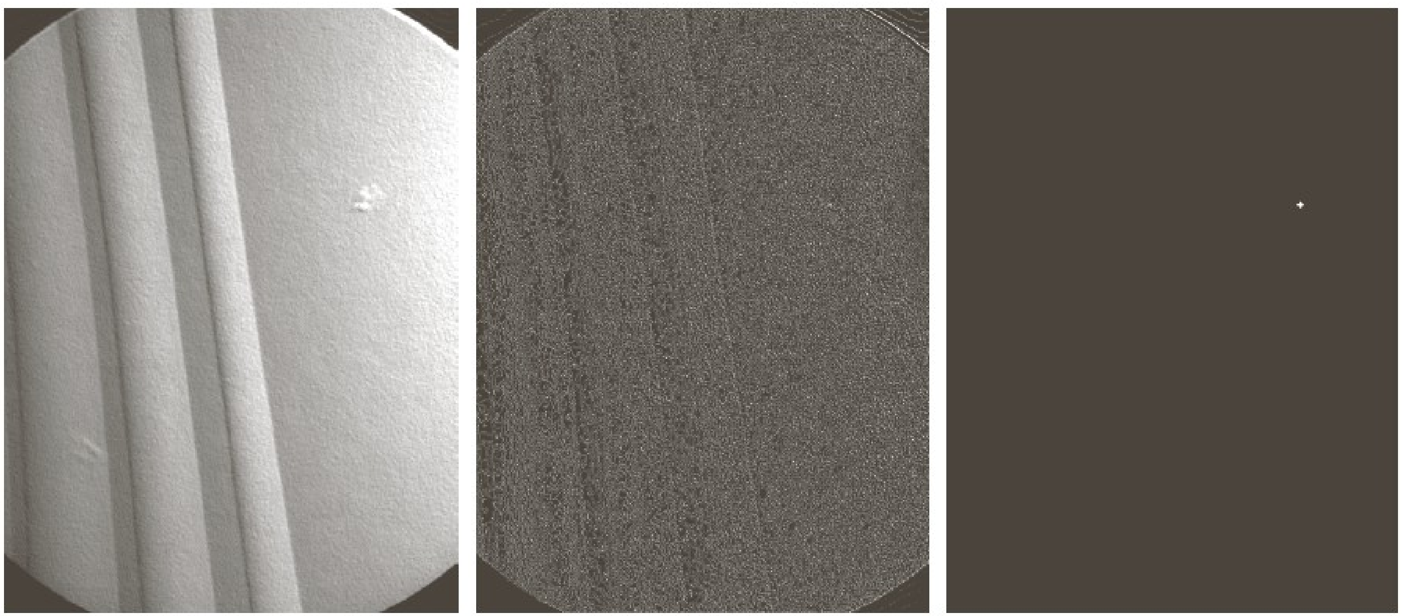
$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)$$

$$-4 f(x, y)$$

$$g(x, y) = \begin{cases} 1 & \text{if } |R(x, y)| \geq T \\ 0 & \text{otherwise} \end{cases} \qquad R = \sum_{k=1}^{9} w_k z_k$$



1	1	1
1	−8	1
1	1	1



a
b c d

FIGURE 10.4 (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) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Line Detection

- Second derivatives to result in a stronger response and to produce thinner lines than first derivatives

- Double-line effect of the second derivative must be handled properly

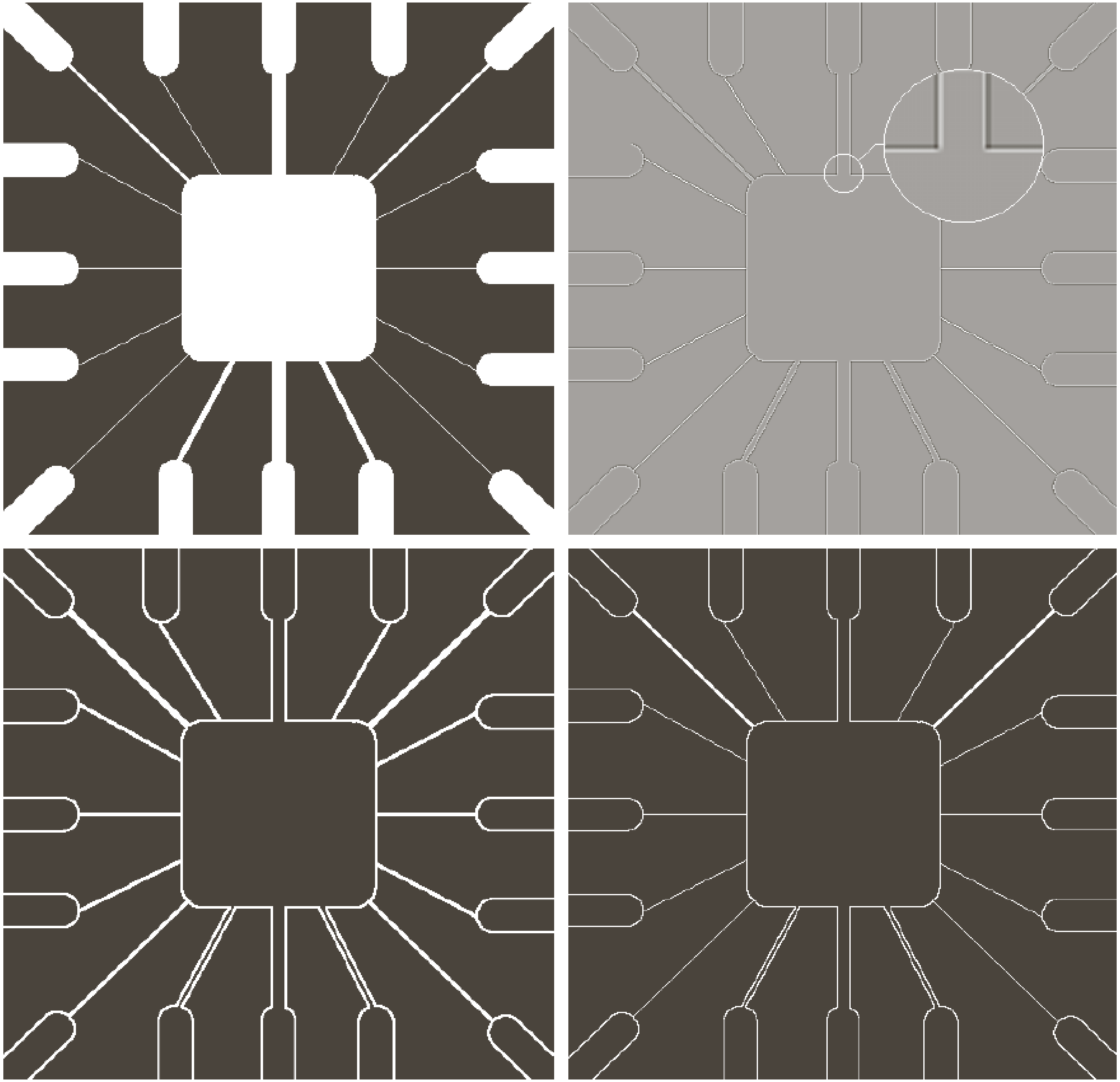


FIGURE 10.5
(a) Original image.
(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.
(c) Absolute value of the Laplacian.
(d) Positive values of the Laplacian.

(b) Intensity are scaled for display

To cater –ve values

(c) Absolute difference taken; so a 0 diff is black; -ve diff are made +ve

Doubles the thickness, reduce double-line effect

Detecting Line in Specified Directions

- Laplacian detector kernel is isotropic, so its response is independent of direction.

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

2	−1	−1
−1	2	−1
−1	−1	2

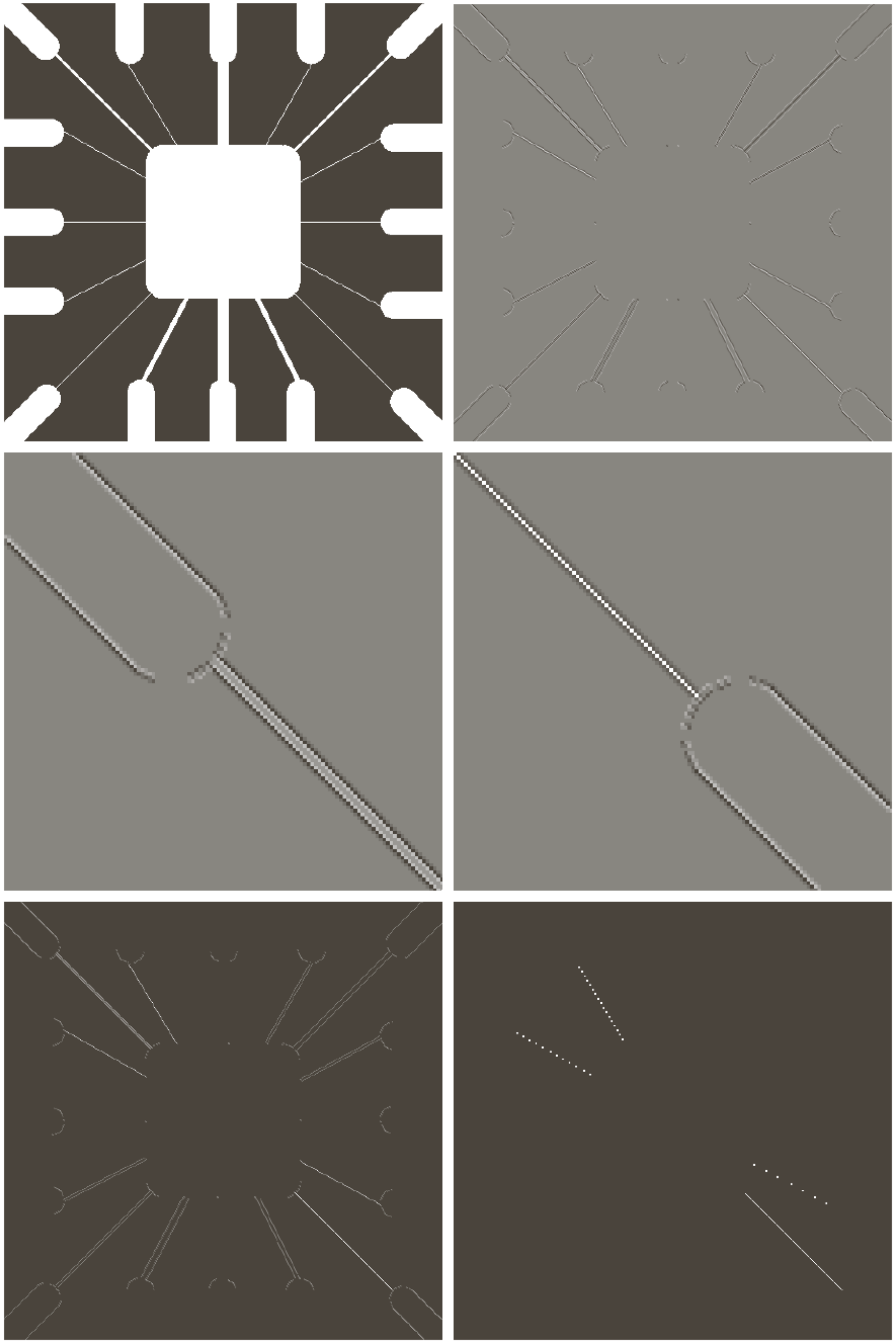
Vertical

−1	2	−1
−1	2	−1
−1	2	−1

−45°

−1	−1	2
−1	2	−1
2	−1	−1

FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

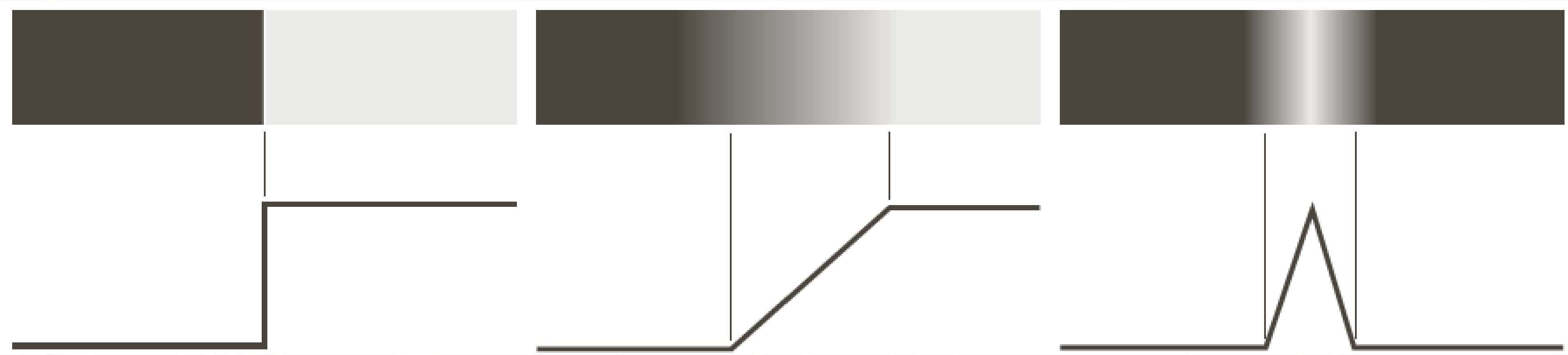


a	b
c	d
e	f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask 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 $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Edge Detection

- Edges are pixels where the brightness function changes abruptly
- Edge models



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.

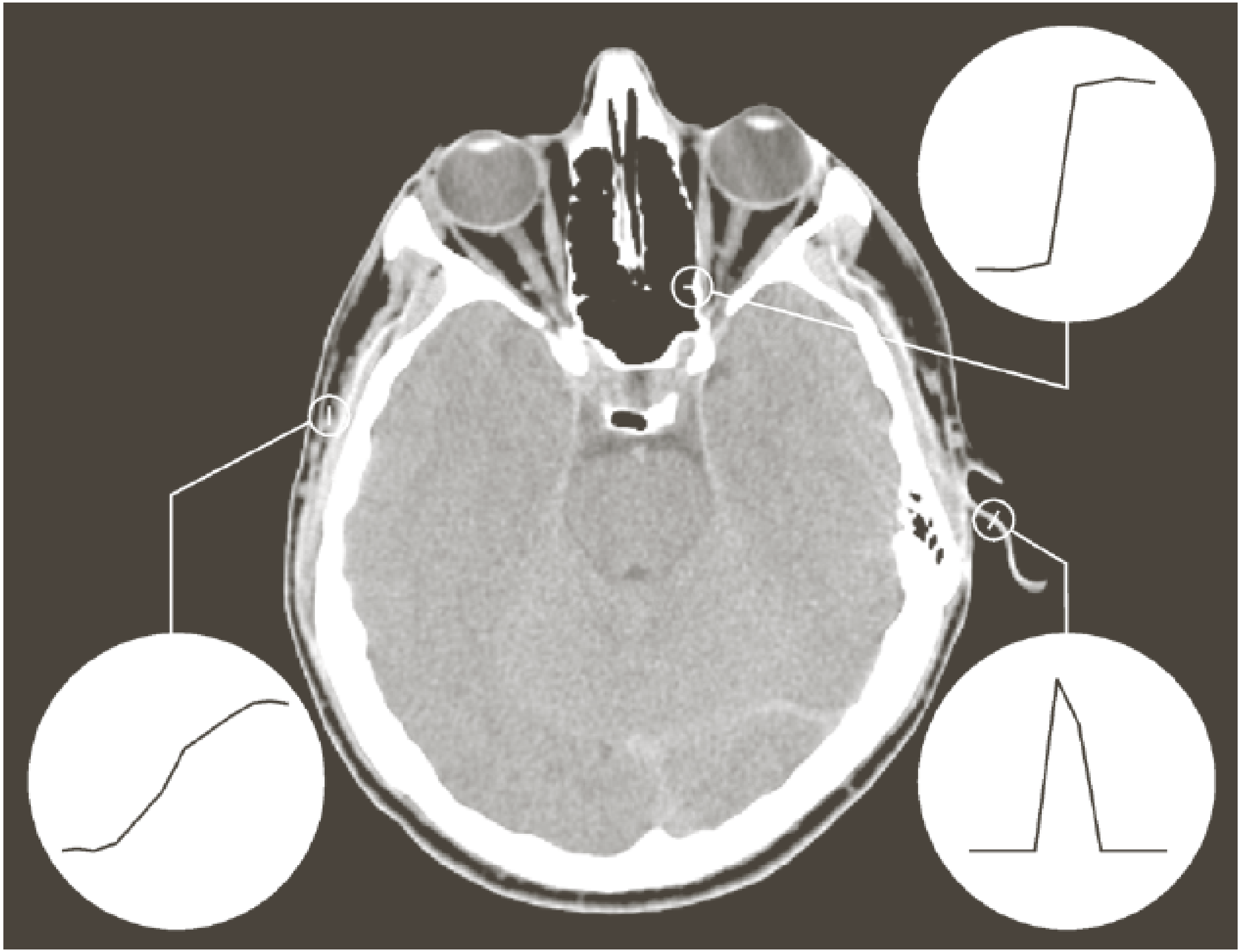
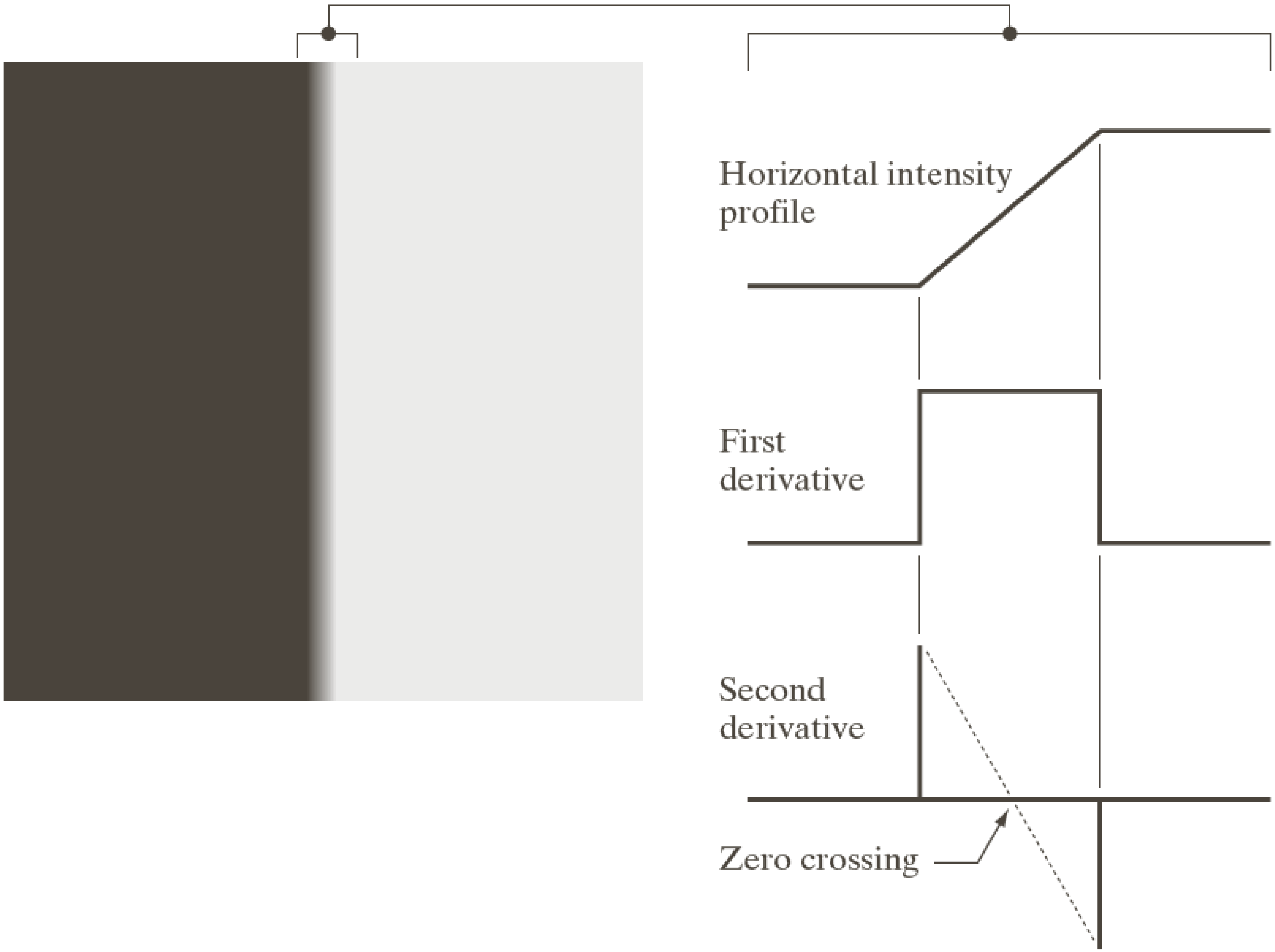


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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



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.

Gaussian Noise μ=0, σ=0.001

Gaussian Noise μ=0, σ=0.1

Gaussian Noise μ=0, σ=1

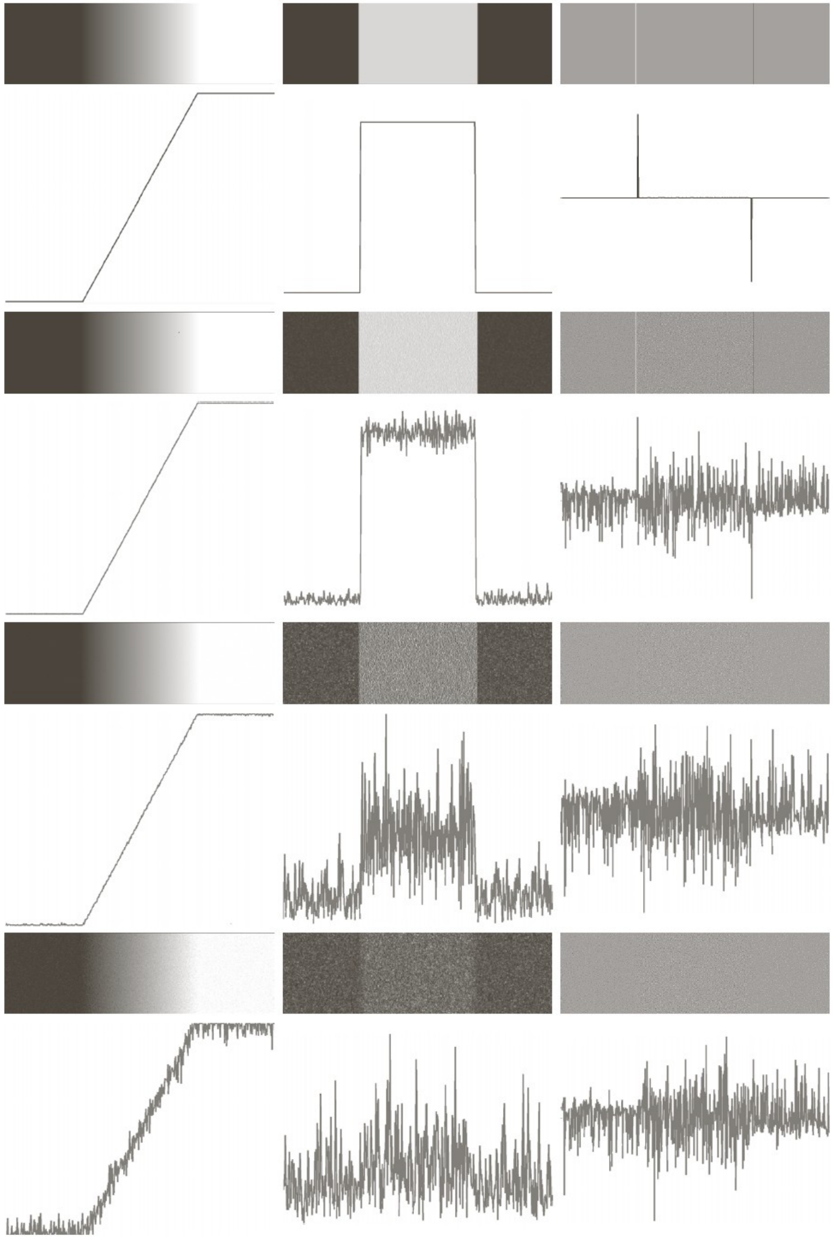


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

Image Smoothing to reduce noise

Extracting all potential edge pixels

Edge localization: selecting only pixels which are edges

Basic Edge Detection by Using First-Order Derivative

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

The magnitude of ∇f

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

The direction of ∇f

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

The direction of the edge

$$\phi = \alpha - 90^\circ$$

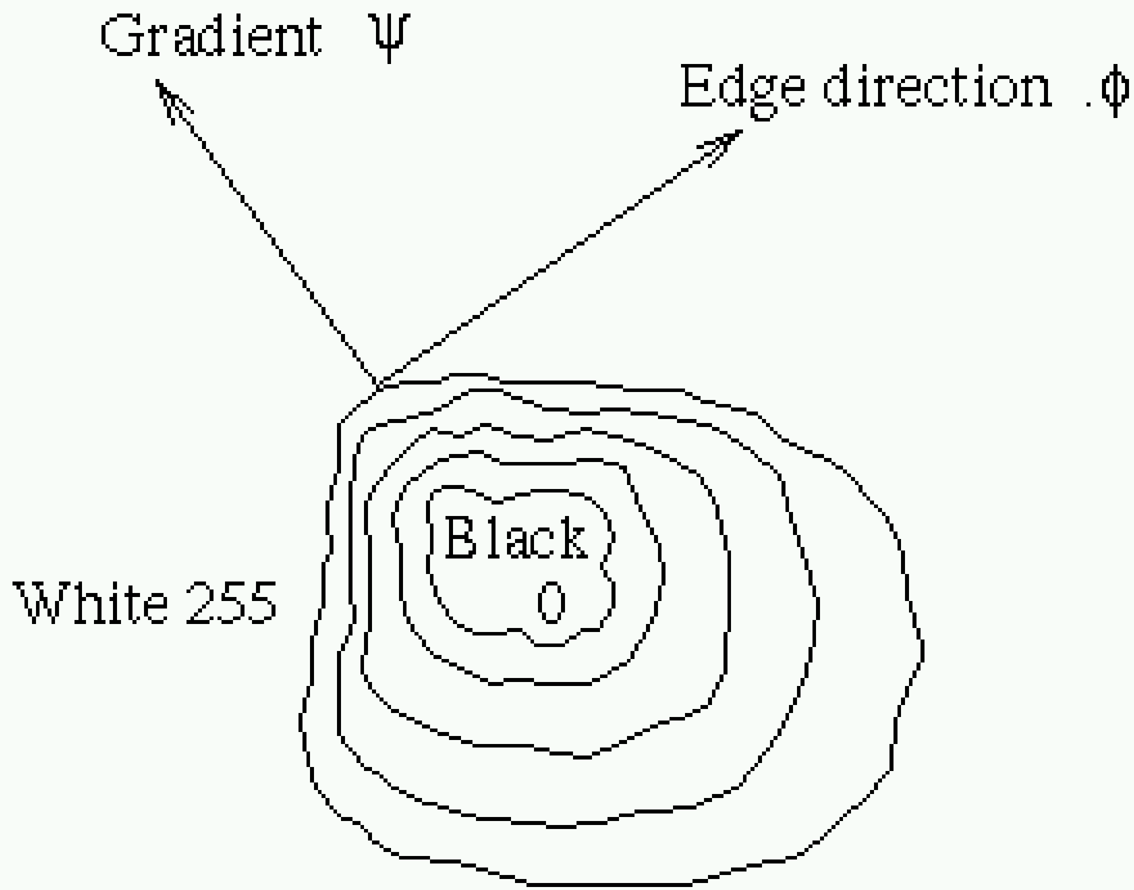
Basic Edge Detection by Using First-Order Derivative

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

Edge unit normal: $\nabla f \,/\, \mathrm{mag}(\nabla f)$

In practice,sometimes the magnitude is approximated by

$$\mathrm{mag}(\nabla f) = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| \text{ or } \mathrm{mag}(\nabla f) = \max\left(\left|\frac{\partial f}{\partial x}\right|, \left|\frac{\partial f}{\partial y}\right|\right)$$


Gradient ψ
Edge direction .ϕ
Black
0
White 255

−1
1

−1	1

a b

FIGURE 10.13 One-dimensional masks used to implement Eqs. (10.2-12) and (10.2-13).

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

	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 masks used to compute the gradient at the point labeled z_5.

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

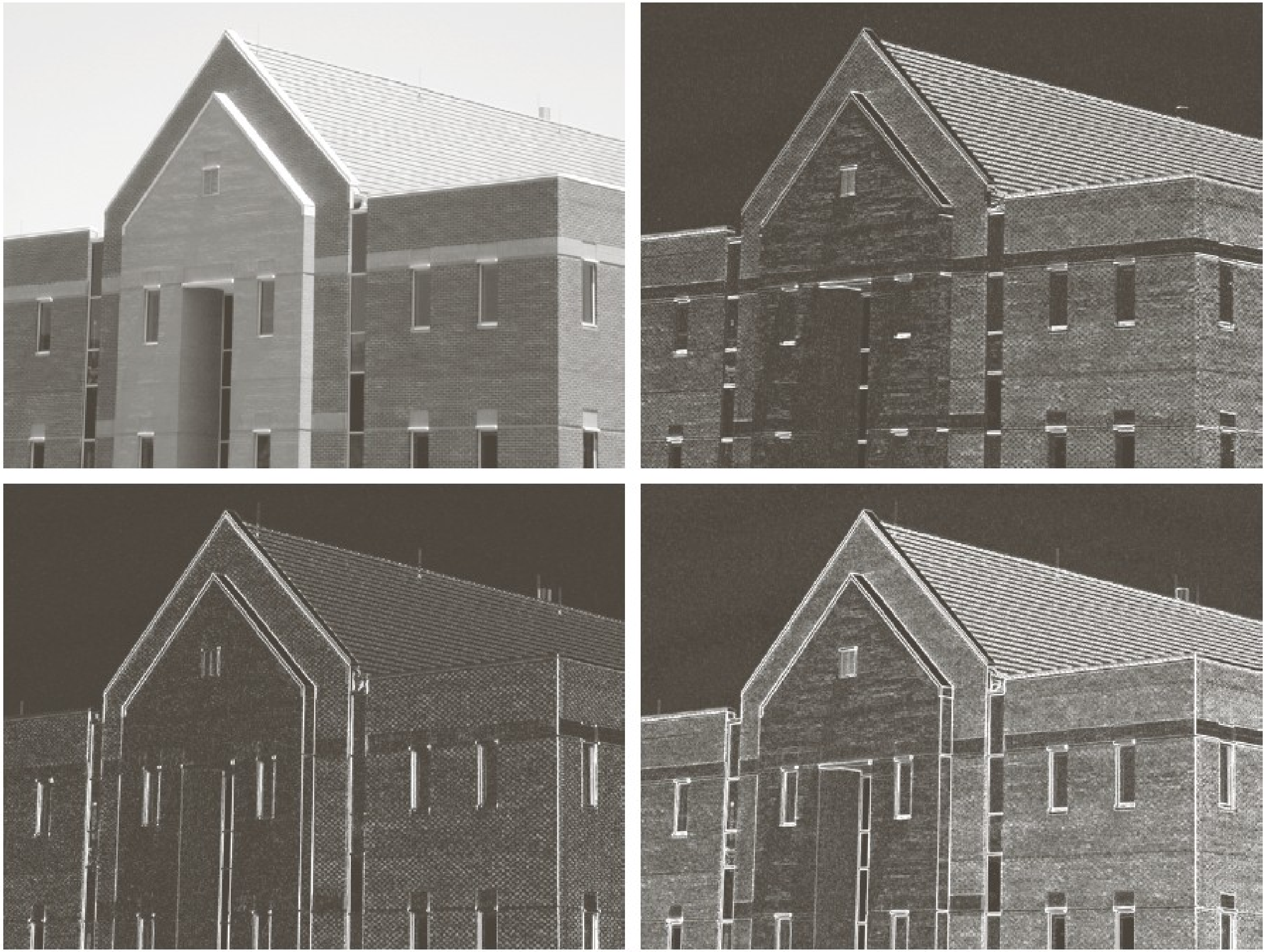
0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a	b
c	d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.



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, $|g_x| + |g_y|$.

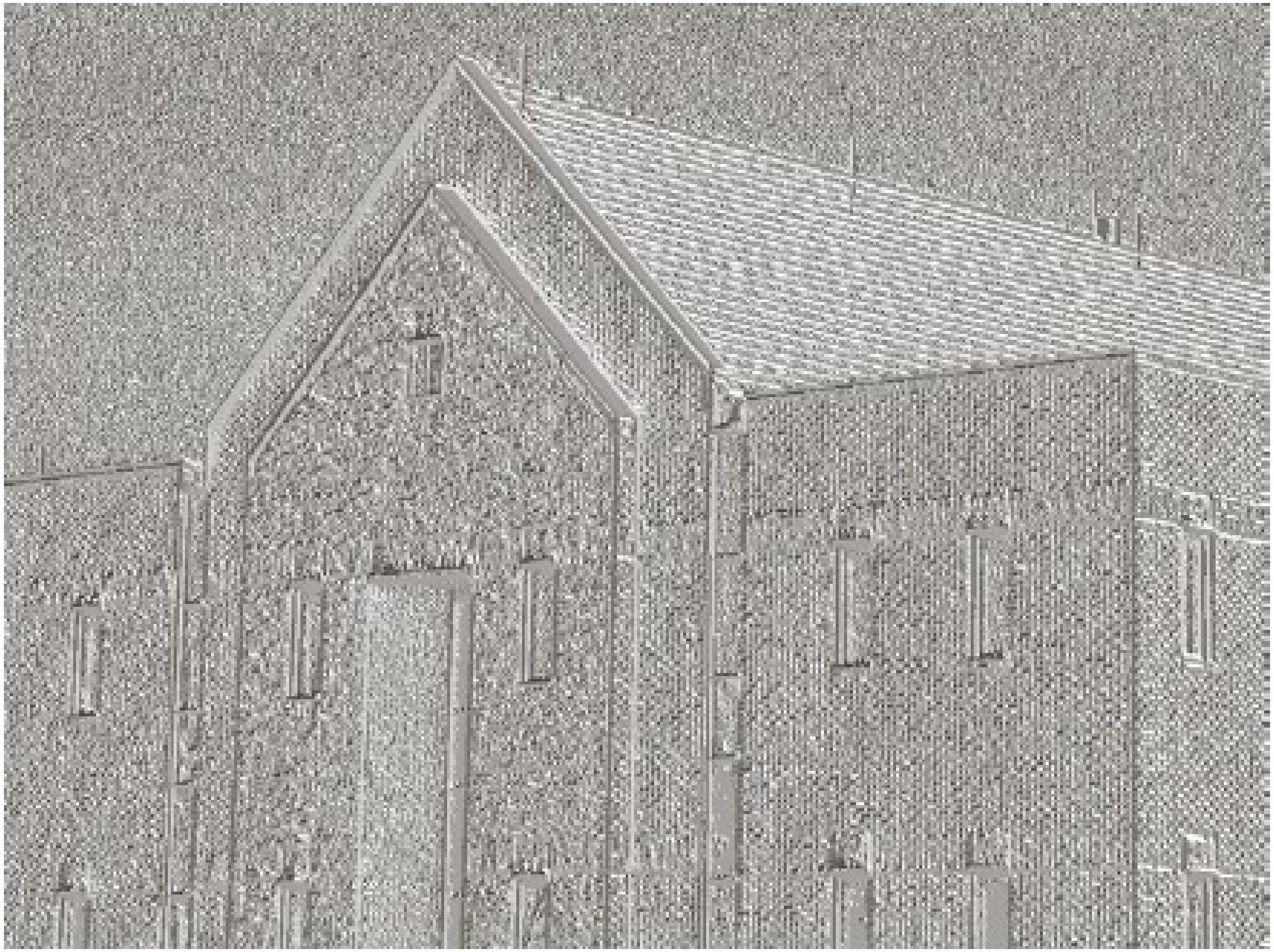
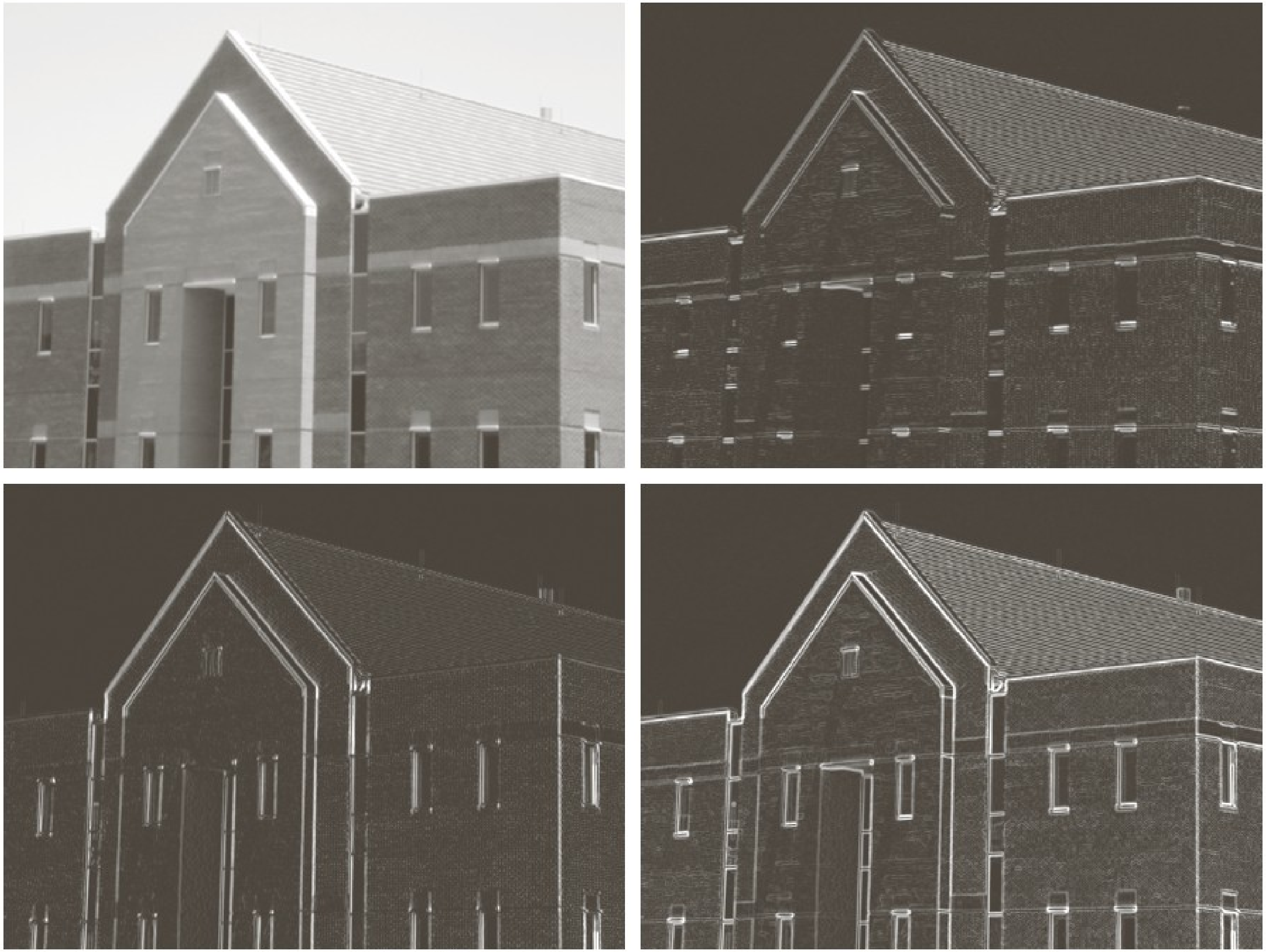


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

The direction of ∇f

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

Not much useful than magnitude image
This is complement of magnitude image
Will be of good help in Canny edge detector



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.

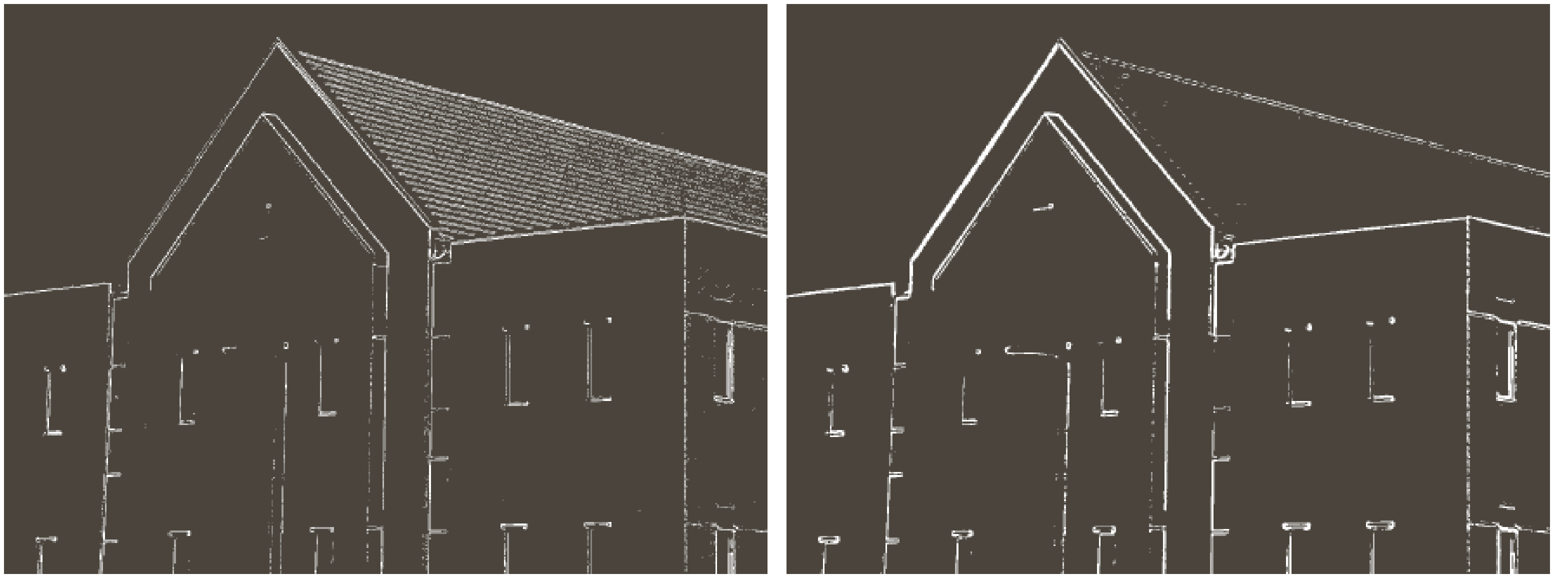
a b

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).

Diagonal edge detection

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

Sobel



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.

- The **Kirsch operator** or **Kirsch compass kernel** is a non-linear edge detector that finds the maximum edge strength in a few predetermined directions.

$$h_{n,m} = \max_{z=1,\ldots,8} \sum_{i=-1}^{1} \sum_{j=-1}^{1} g_{ij}^{(z)} \cdot f_{n+i,m+j}$$

$$\mathbf{g}^{(1)} = \begin{bmatrix} +5 & +5 & +5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix},\ \mathbf{g}^{(2)} = \begin{bmatrix} +5 & +5 & -3 \\ +5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix},\ \mathbf{g}^{(3)} = \begin{bmatrix} +5 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & -3 & -3 \end{bmatrix},\ \mathbf{g}^{(4)} = \begin{bmatrix} -3 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & +5 & -3 \end{bmatrix}$$

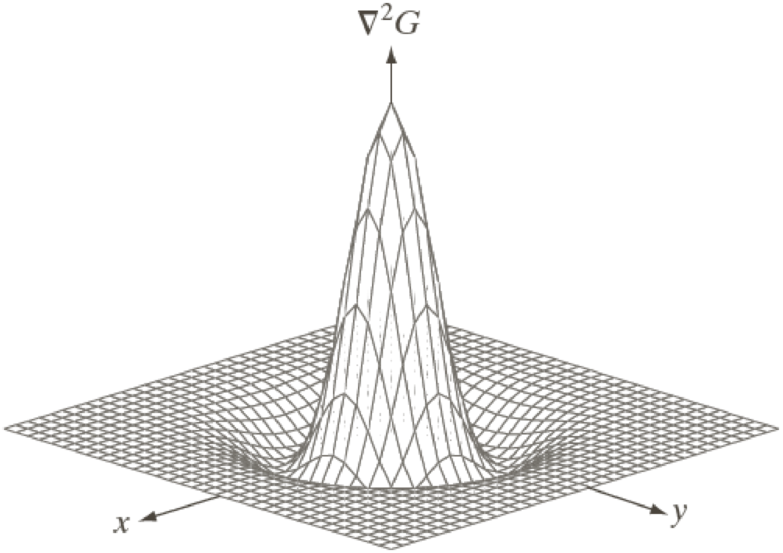
Advanced Techniques for Edge Detection

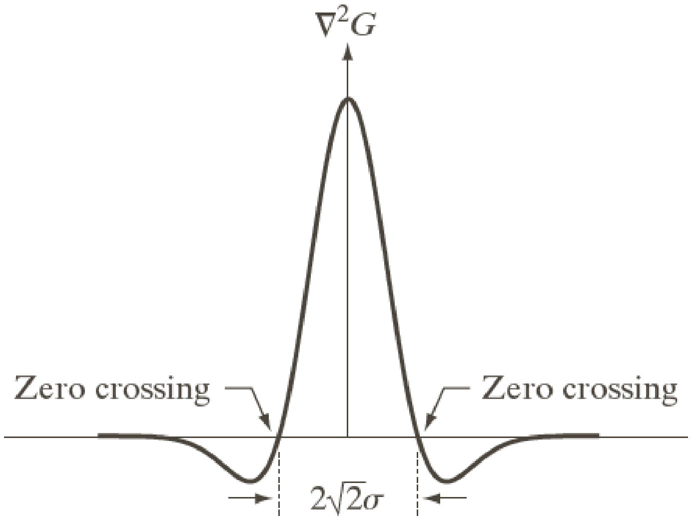
- The Marr-Hildreth edge detector

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

Laplacian of Gaussian (LoG)

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





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) 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.

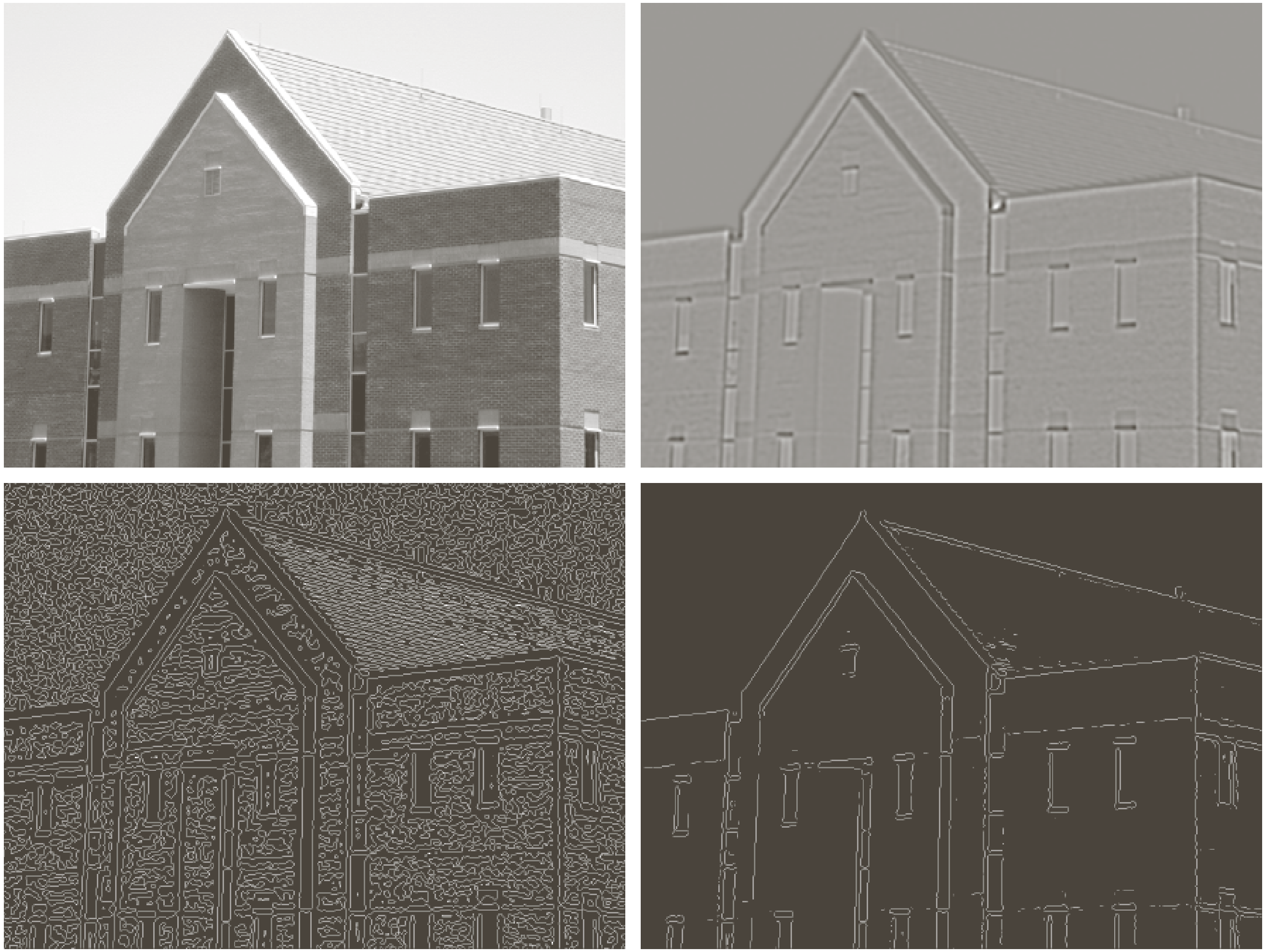
Marr-Hildreth Algorithm

1. Filter the input image with an nxn Gaussian lowpass filter. N is the smallest odd integer greater than or equal to 6σ

2. Compute the Laplacian of the image resulting from step1

3. Find the zero crossing of the image from step 2

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



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.

The Canny Edge Detector

- **Optimal for step edges corrupted by white noise.**
- **The Objective**

1. **Low error rate**
   The edges detected must be as close as possible to the true edge
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 number of local maxima around the true edge should be minimum

The Canny Edge Detector: Algorithm (1)

Let $f(x, y)$ denote the input image and $G(x, y)$ denote the Gaussian function:

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

We form a smoothed image, $f_s(x, y)$ by convolving G and f:

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

The Canny Edge Detector: Algorithm(2)

Compute the gradient magnitude and direction (angle):

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

and

$$\alpha(x, y) = \arctan(g_y / g_x)$$

where $g_x = \partial f_s / \partial x$ and $g_y = \partial f_s / \partial y$

Note: any of the filter mask pairs in Fig.10.14 can be used to obtain g_x and g_y

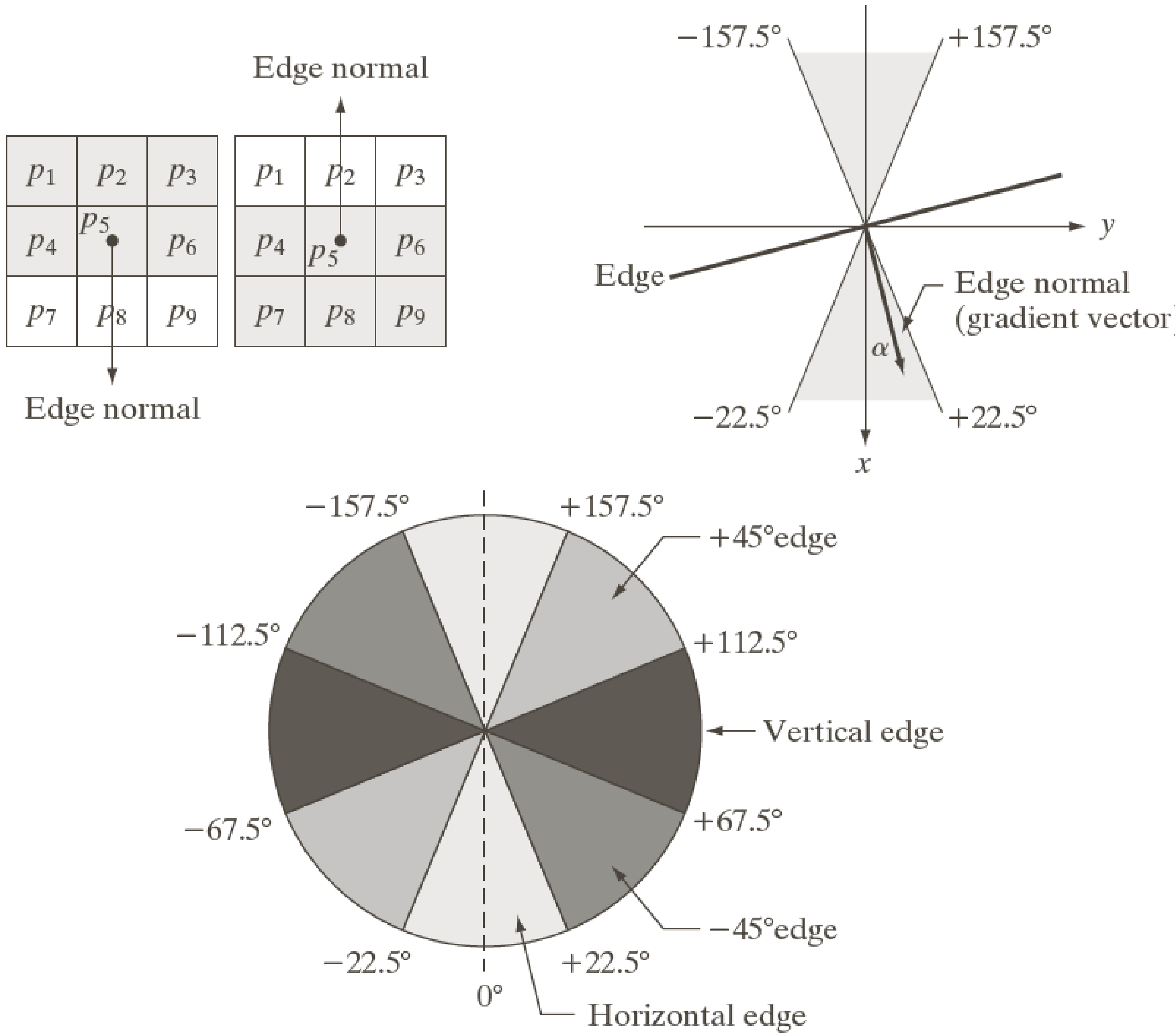
The Canny Edge Detector: Algorithm(3)

The gradient $M(x, y)$ typically contains wide ridge around local maxima. Next step is to thin those ridges.

Nonmaxima suppression:

Let d_1, d_2, d_3, and d_4 denote the four basic edge directions for a 3×3 region: horizontal, -45°, vertical,+45°, respectively.

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



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.

The Canny Edge Detector: Algorithm(4)

The final operation is to threshold $g_N(x, y)$ to reduce false edge points.

<u>Hysteresis thresholding:</u>

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

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

and

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

Weak Edge pixels

Strong Edge pixels

The Canny Edge Detector: Algorithm(5)

Depending on the value of T_H, the edges in $g_{NH}(x, y)$ typically have gaps. Longer edges are formed using the following procedure:

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

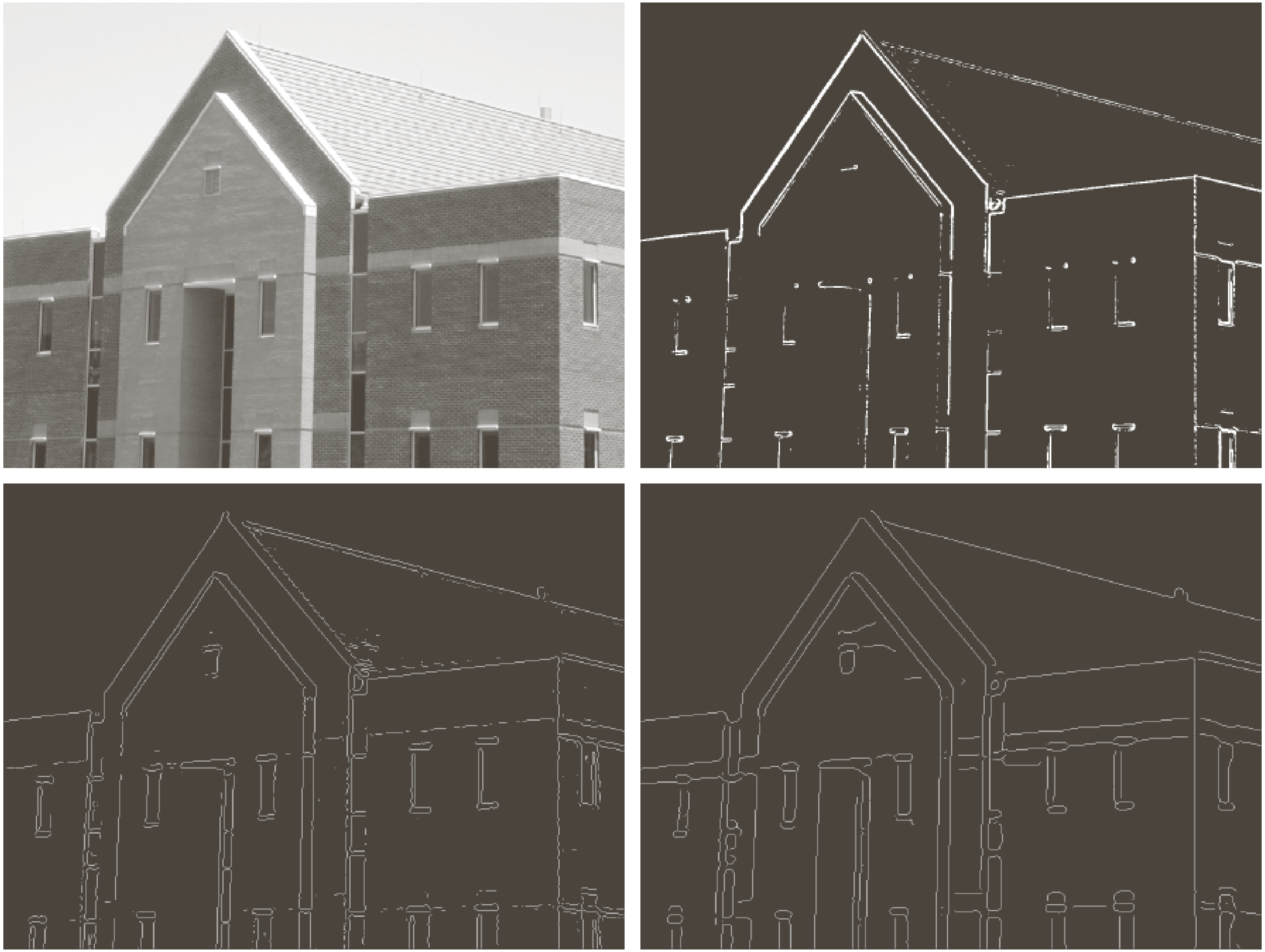
(b). Mark as valid edge pixel all the weak pixels in $g_{NL}(x, y)$ that are connected to p using 8-connectivity.

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

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

The Canny Edge Detection: Summary

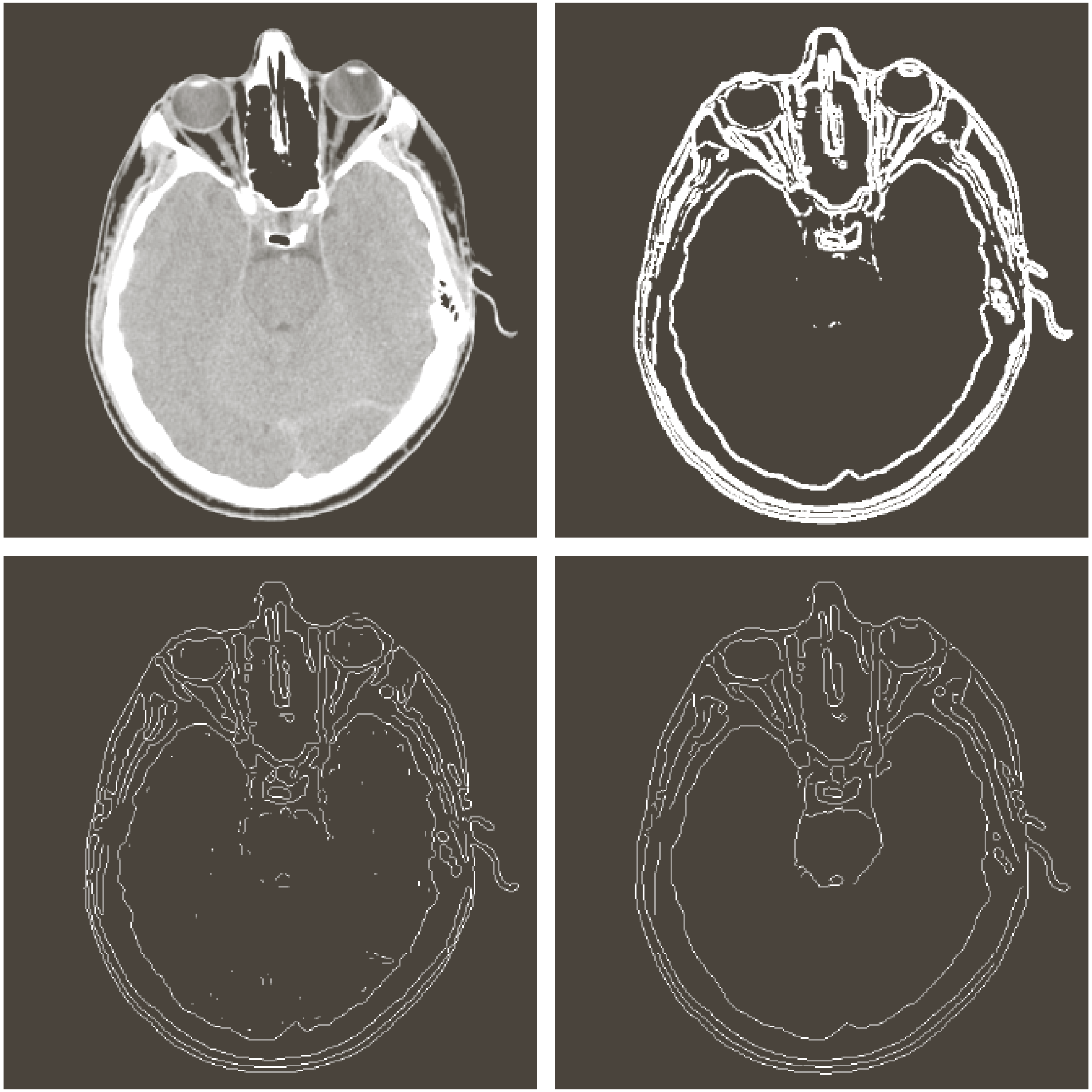
- Smooth the input image with a Gaussian filter
- Compute the gradient magnitude and angle images
- Apply nonmaxima suppression to the gradient magnitude image
- Use double thresholding and connectivity analysis to detect and link edges



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.

$T_L = 0.04; T_H = 0.10; \sigma = 4$ and a mask of size 25×25



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.)

$T_L = 0.05; T_H = 0.15; \sigma = 2$ and a mask of size 13×13