10.2.5 Basic edge detection

The image gradient and its properties

Definition

$$\nabla \mathbf{f} = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \begin{pmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{pmatrix}$$

Magnitude

$$\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = [g_x^2 + g_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

Direction

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

The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point

Example

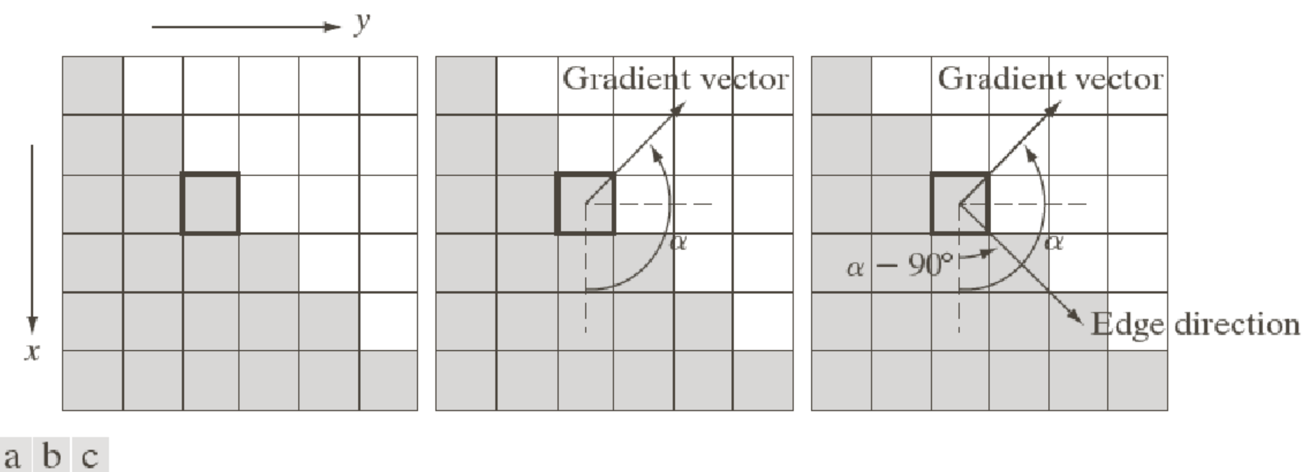


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

$$\nabla \mathbf{f} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}; \quad M(x,y) = 2\sqrt{2}; \quad \alpha(x,y) = -45^\circ$$

(135° in positive direction wrt x-axis)

Gradient operators

$$\boxed{g_x = \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y)} \quad \boxed{g_y = \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y)}$$

−1
1

−1	1

a b

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

Roberts: $$\boxed{g_x = \frac{\partial f(x,y)}{\partial x} = (z_9 - z_5)} \quad \boxed{g_y = \frac{\partial f(x,y)}{\partial y} = (z_8 - z_6)}$$

Prewitt: $$\boxed{g_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)} \quad \boxed{g_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)}$$

Sobel: $$\boxed{g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)} \quad \boxed{g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)}$$

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.

Convenient approximation:

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

Masks for detecting diagonal edges

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

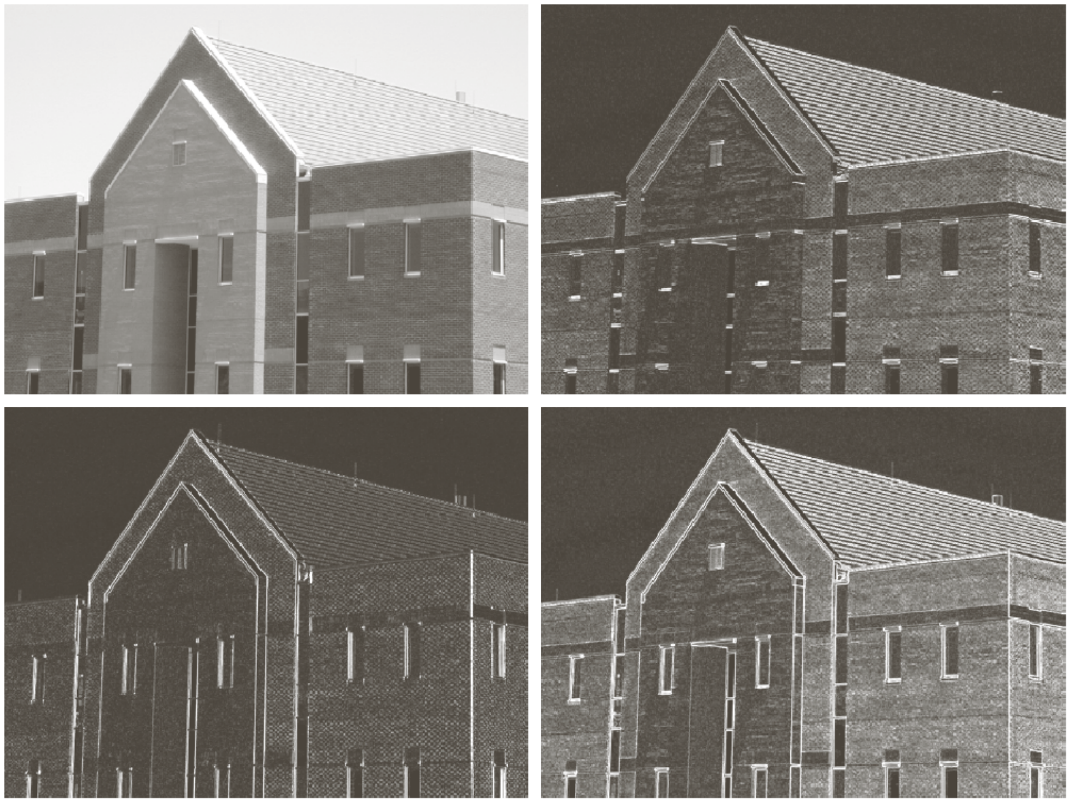
−2	−1	0
−1	0	1
0	1	2

Sobel

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

Example 10.6



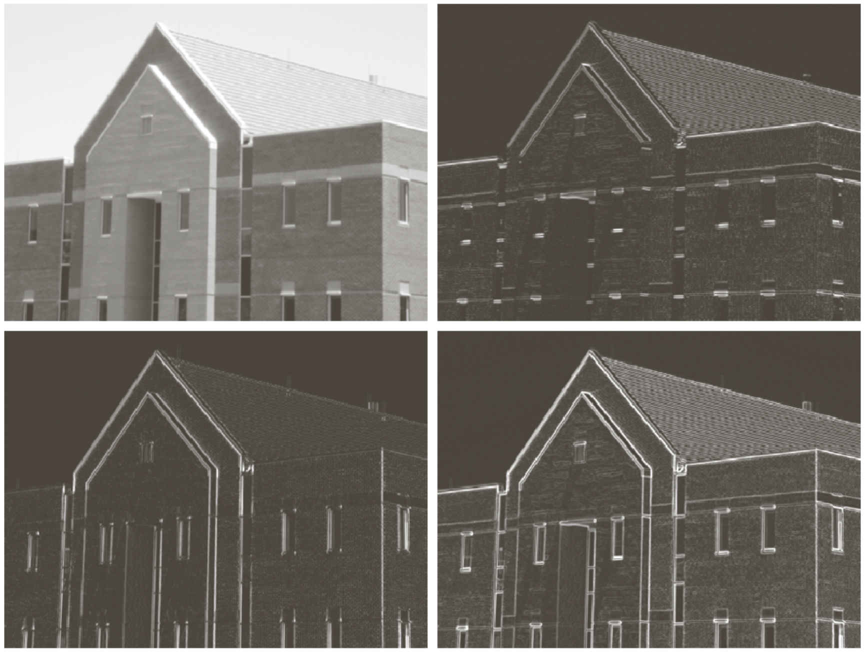
a b
c d

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.

Angle info plays key supporting role in Canny edge detection

Result with prior smoothing



a b
c d

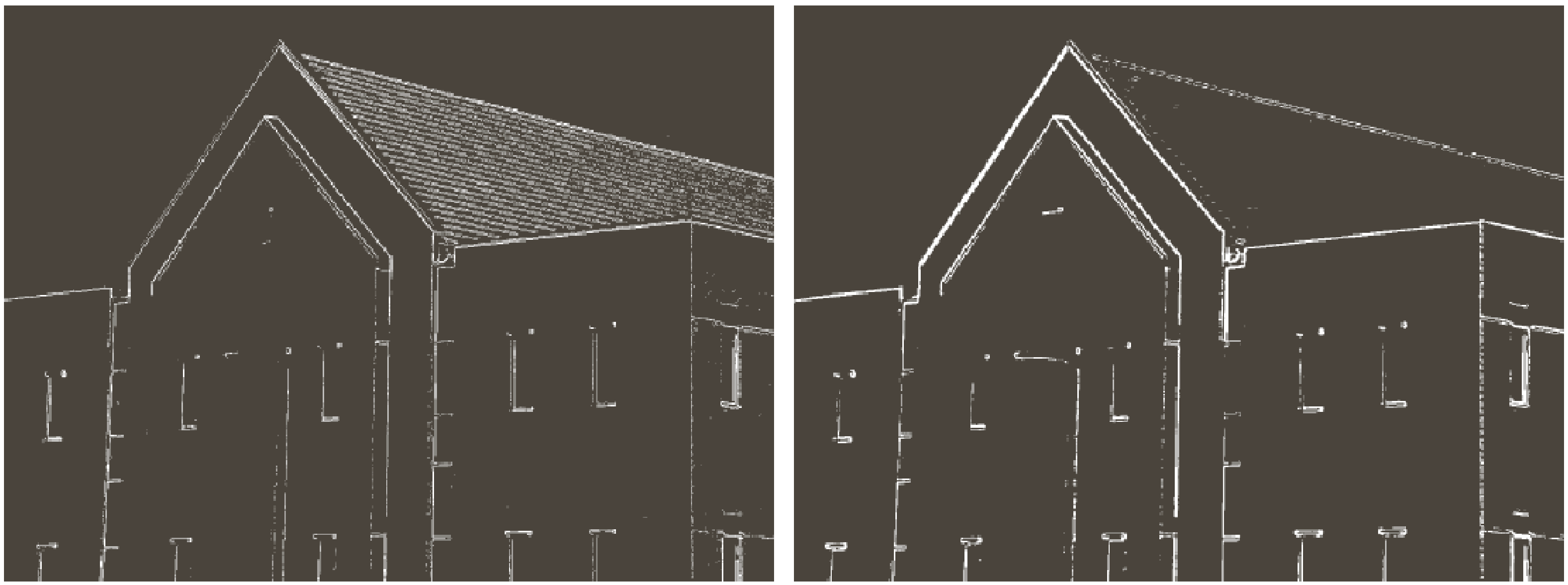
FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.

Diagonal 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).

Combining the gradient with 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.

10.2.6 More advanced techniques for edge detection

The Marr-Hildreth edge detector [1980]

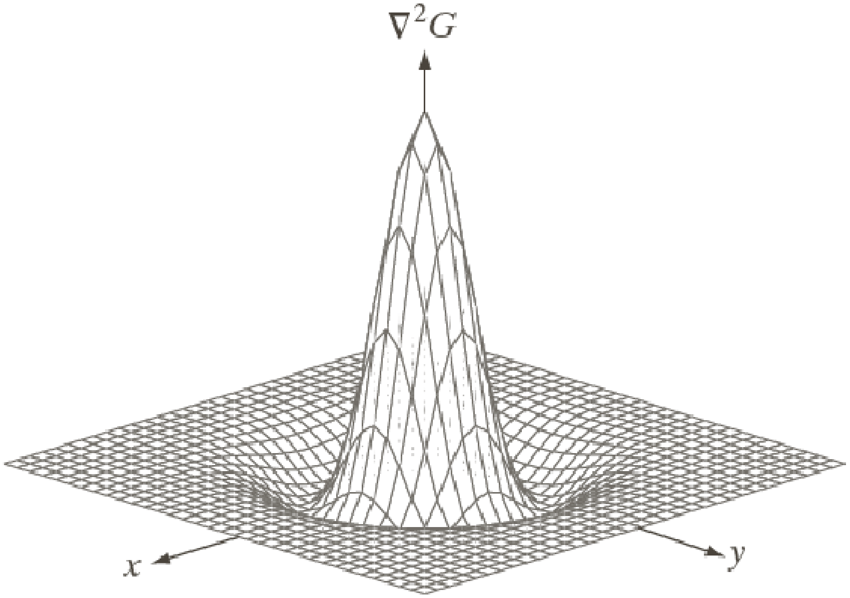
- Image should be smoothed first (to reduce noise)
- Larger operators should be used for larger images
- Zero-crossings of second derivative should be exploited
- The Laplacian of a Gaussian (LoG) operator is therefore empolyed

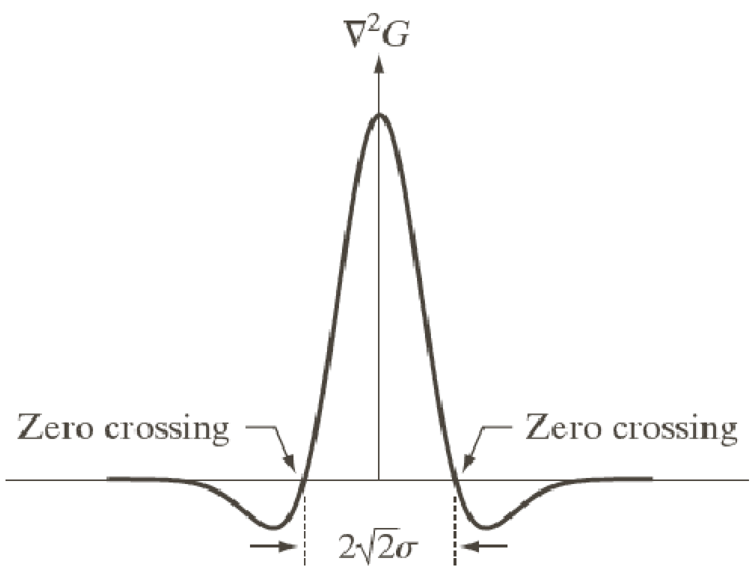
2-D Gaussian operator:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \ldots \quad (1)$$

Laplacian of Gaussian (LoG)

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





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 LoG filter is first convolved with the input image $f(x,y)$,

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

after which the zero-crossings of $g(x,y)$ is found

Summary of algorithm:

(1) Filter input image with $n \times n$ Gaussian lowpass filter (sample eqn (1))

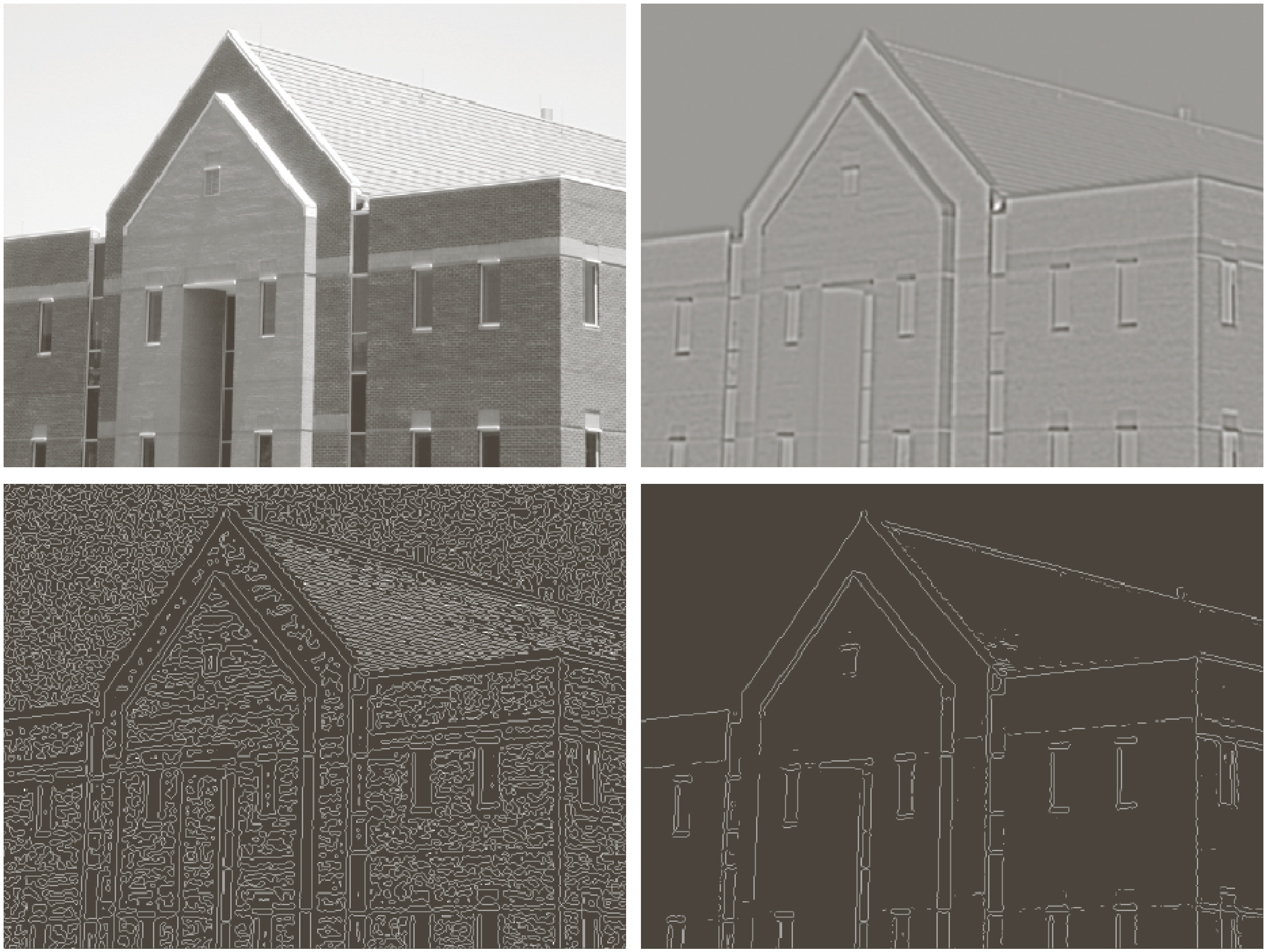
(2) Compute the Laplacian of the result in step (1)

(3) Find the zero-crossings in the result in step (2)

- **Rule of thumb: $n \equiv$ smallest odd integer greater than or equal to 6σ**

- **A zero-crossing at p implies that the sign of at least two of its oposing neibouring pixels must differ and that the absolute value of this difference must exceed a certain threshold**

Example 10.7



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.

- $\sigma = 4$ **($\approx 0.5\%$ of smallest image dimension)**
- **DoG (READ)**

The Canny edge detector [1986]

- Driven by 3 objectives:

(1) Low error rate: All edges should be found

(2) Edge points should be well localized: Edges found must be as close as possible to true edges

(3) Single edge point response: Only one point for each true edge point should be returned

Canny concluded that a good approximation to the optimal step edge detector is the first derivative of a Gaussian

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

For 2-D, the 1-D approach still applies in the direction of the edge normal

Since normal direction is unknown beforehand, 1-D detector has to be applied for all directions

This is approximated by (1) smoothing image with 2-D Gaussian; (2) compute the gradient of the result; (3) use gradient magnitude and direction to estimate edge strength and direction at every point

Smoothed image f_s is first formed by convolving G and f:

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

The gradient magnitude and direction are then calculated:

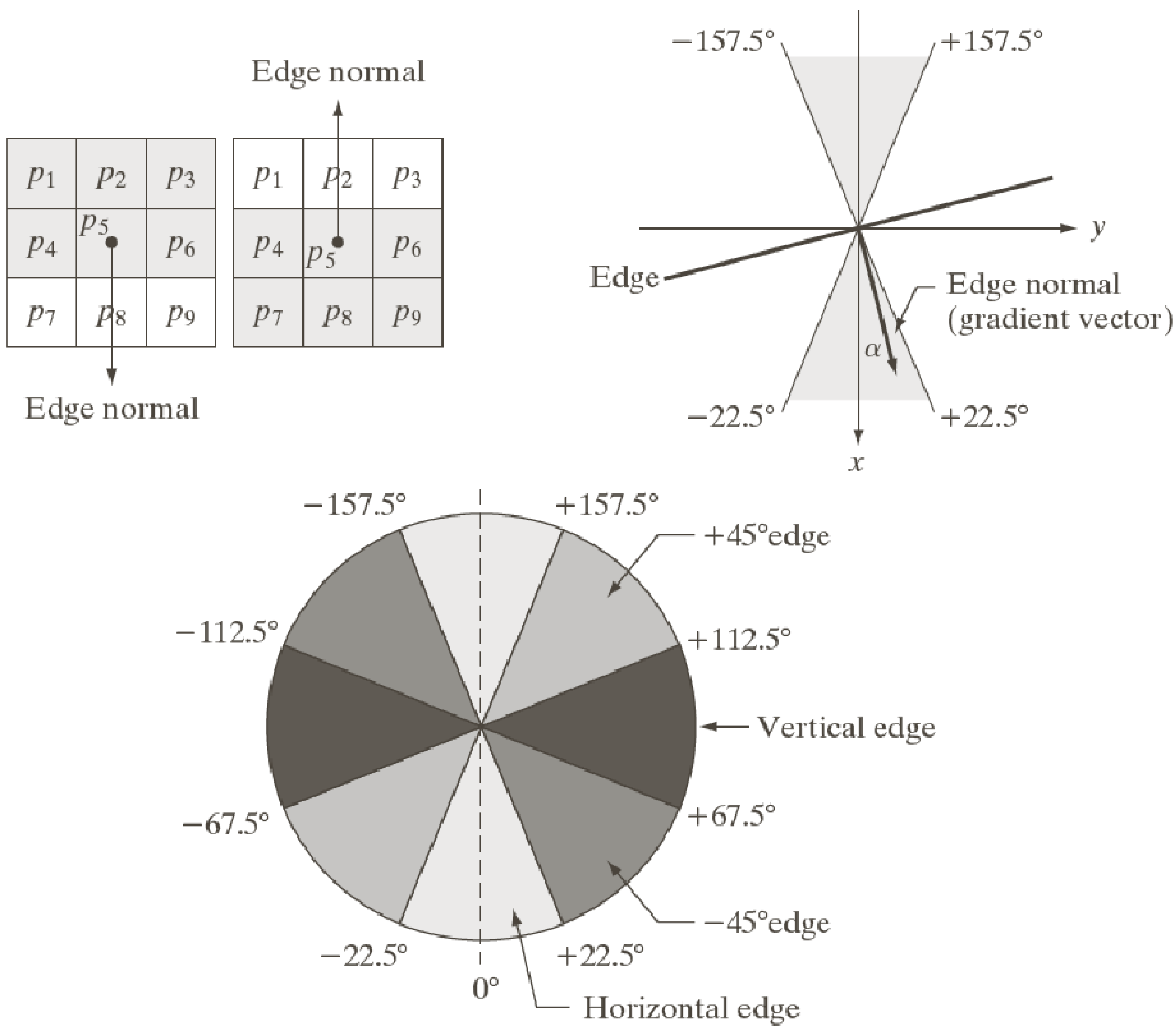
$$M(x,y) = \sqrt{g_x^2 + g_y^2}; \quad \alpha(x,y) = \tan^{-1}\left\{\frac{g_y}{g_x}\right\}$$

Wide ridges are then thinned using nonmaxima supression:

- **Specify number of discrete orientations of the edge normal (for a 3×3 region, we can specify four orientations: horizontal, vertical, $+45°$, and $-45°$)**

- **If edge normal is in range of directions from $-22.5°$ to $+22.5°$, or from $-157.5°$ to $+157.5°$, the edge is deemed "horizontal"**

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



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.

Nonmaxima supression scheme for 3×3 region centered at every point (x, y) in $\alpha(x, y)$:

(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 neighbours along d_k, then $g_N(x, y) = 0$ (supression); otherwise $g_N(x, y) = M(x, y)$

Final operation is to threshold $g_N(x, y)$ to reduce false edge points

- **Hysteresis thresholding (Section 10.3.6): Two thresholds, T_L and T_H are selected using Otsu's method**

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

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

$g_{\mathrm{NH}}(x, y) \equiv$ **"strong" edge pixels** $\qquad g_{\mathrm{NL}}(x, y) \equiv$ **"weak" edge pixels**

Strong pixels in $g_{\mathrm{NH}}(x, y)$ are assumed to be valid and marked accordingly

Also, the edges in $g_{\mathrm{NH}}(x, y)$ typically have gaps!

Longer edges are formed using the following procedure:

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

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

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

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

Final output image is formed by appending to $g_{\mathrm{NH}}(x, y)$ all the nonzeo pixels from $g_{\mathrm{NL}}(x, y)$

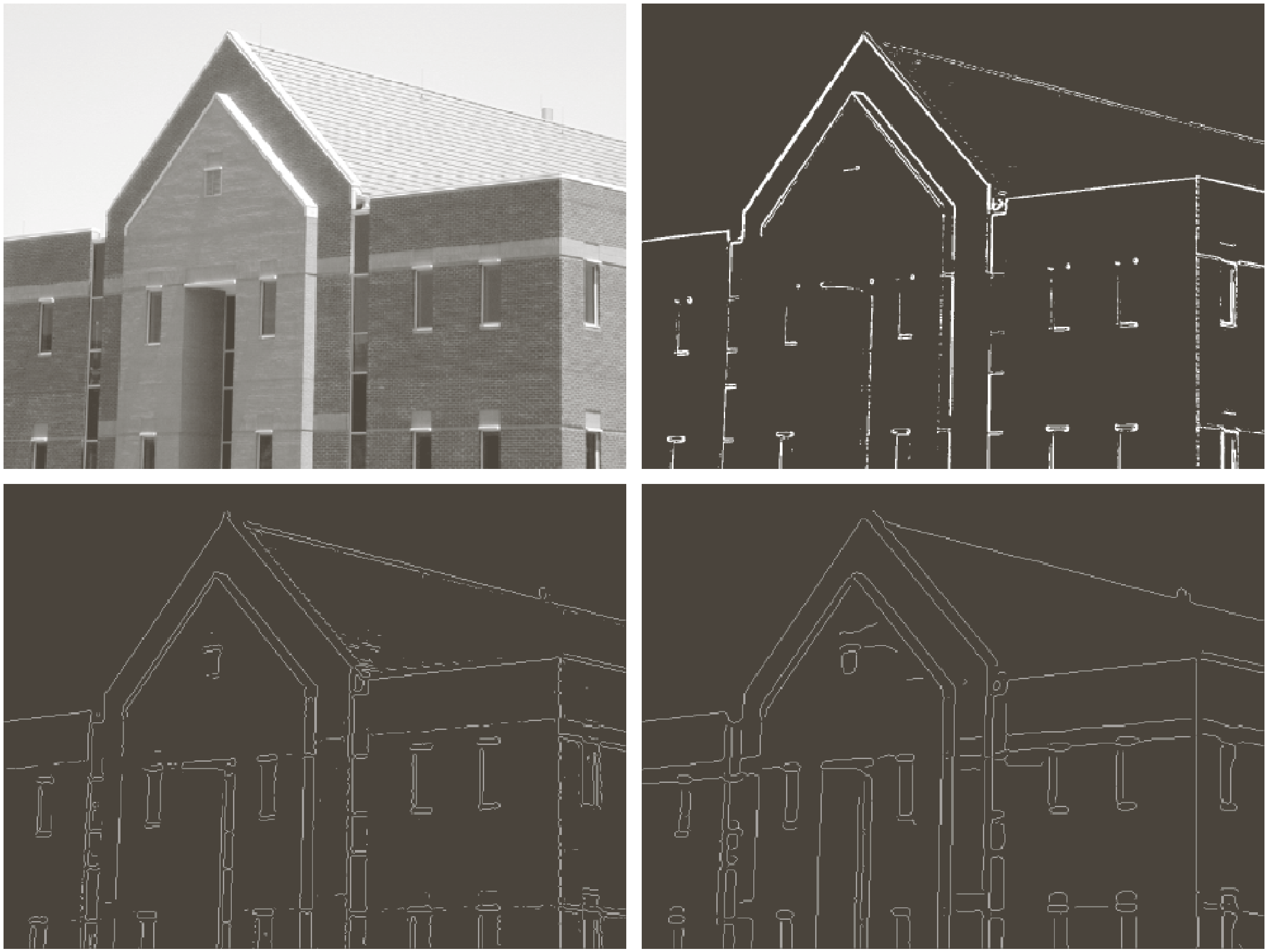
Summary: (Step (4) is typically followed by one pass of edge thinning)

(1) Smooth the input image with a Gaussian filter

(2) Compute the gradient magnitude and angle images

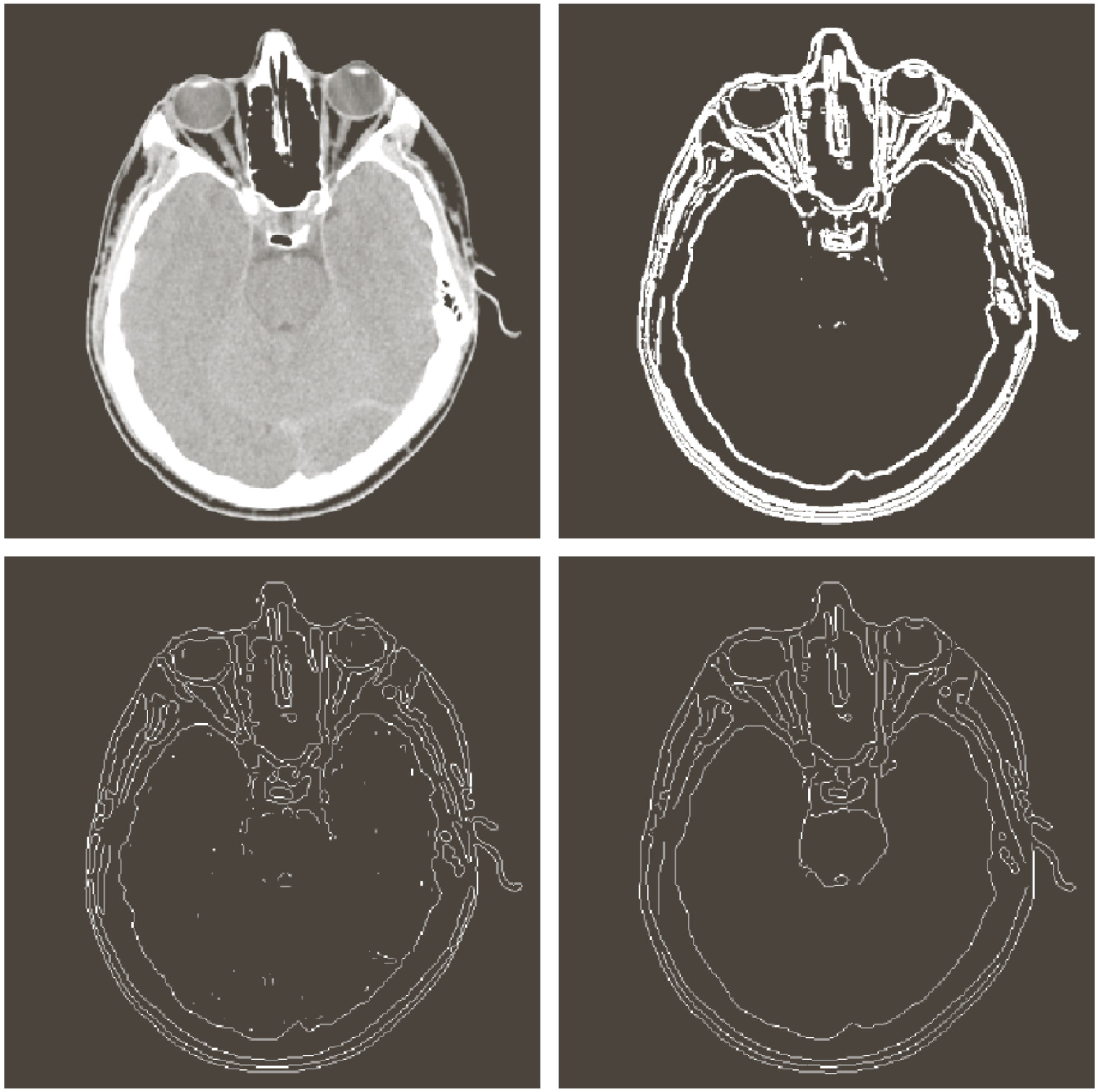
(3) Apply nonmaxima supression to the gradient magnitude image

(4) 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.



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.)