Digital Image Processing

Lecture # 5
Edge Detection

Use of first derivatives for image enhancement: The Gradient

- The **gradient** of a function f(x,y) is defined as

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

Gradient Operators

Sobel Operator

-1	-2	-1
0	0	0
1	2	1

Extract horizontal edges

-1	0	1
-2	0	2
-1	0	1

Extract vertical edges

Emphasize more the current point (y direction)

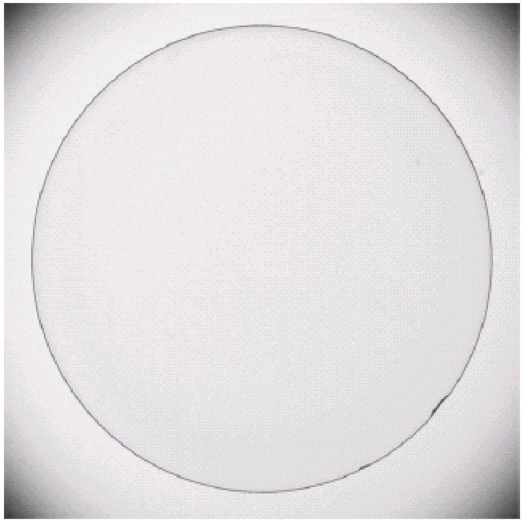
$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

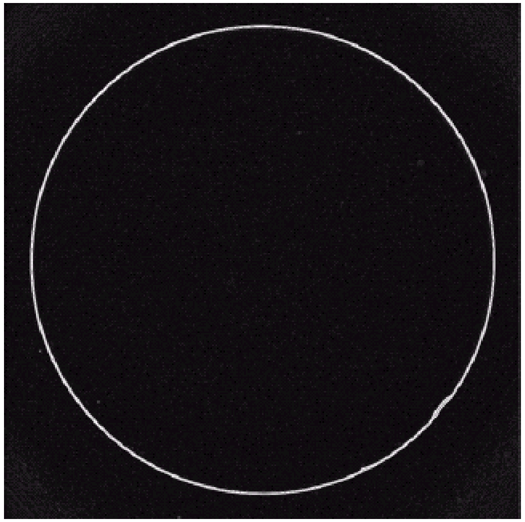
Emphasize more the current point (x direction)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel Arrangement

Sobel Operator: Example



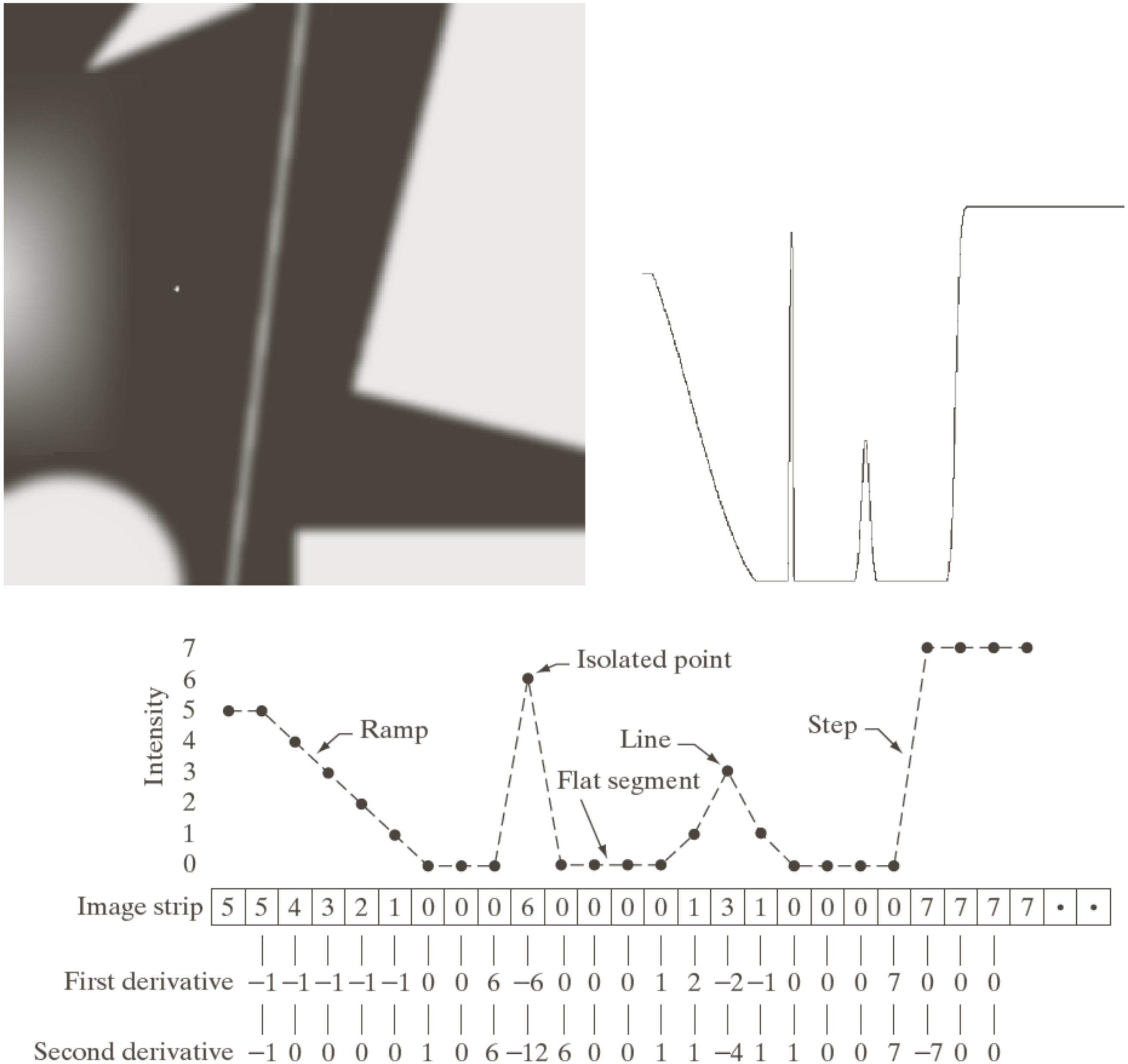


An image of a contact lens which is enhanced in order to make defects more obvious

Sobel filters are typically used for edge detection

Edge Detection

- Edge detectors can be based on the first and second derivatives which can detect abrupt intensity changes.
- Derivates of digital functions are defined in terms of differences
- See approximations on using first and second derivatives in Gonzalez section 10.2.1


Intensity
7
6
5
4
3
2
1
0
Isolated point
Ramp
Line
Step
Flat segment
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

Edge Detection

- First order derivatives produce thick edges while second order derivatives produce finer ones. See ramp edge on the figure 10.2 on previous slide
- Second order derivatives enhance sharp changes and fine details more aggressively than first order derivatives see the isolated point and the line in the same figure. This can be a problem if the noise is present in the image
- Second derivative changes its sign as it transitions into and out of a ramp or step edge. See the step edge. This “double edge” effect can be used to locate edges.
- The sign of the second derivative is also used to determine whether an edge is a transition from light to dark (-ve value) or from dark to light (+ve value). See step edge

Edge Detection

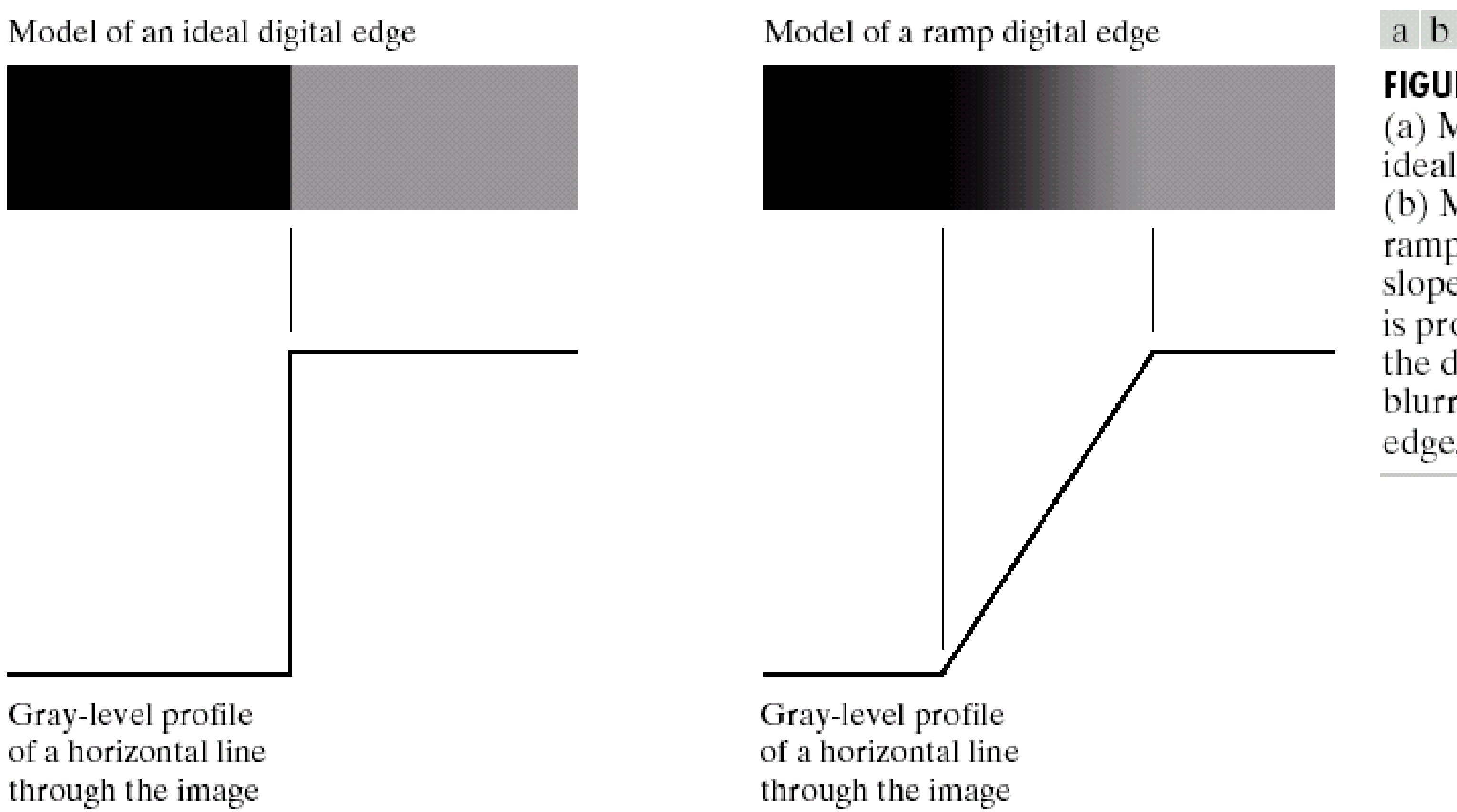
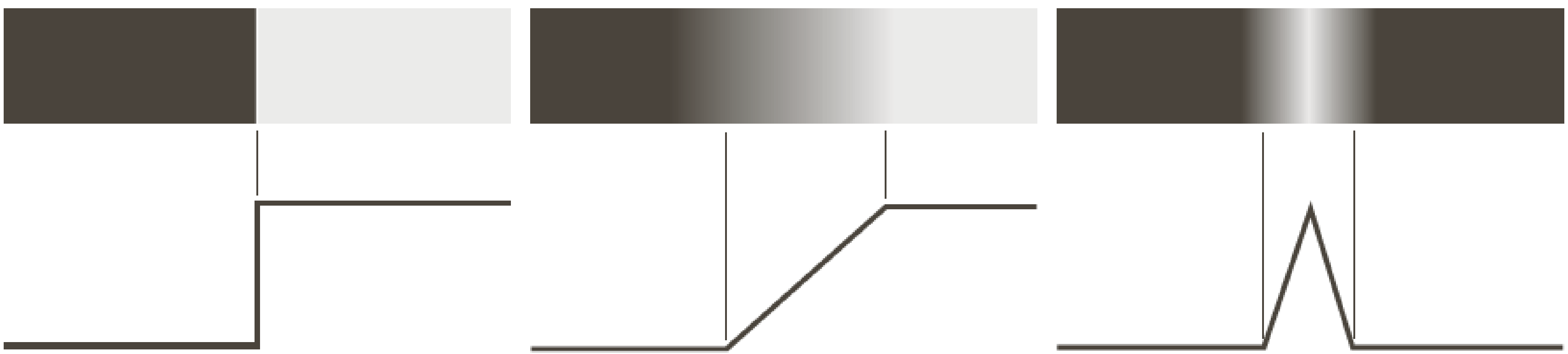


FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

Edges



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.

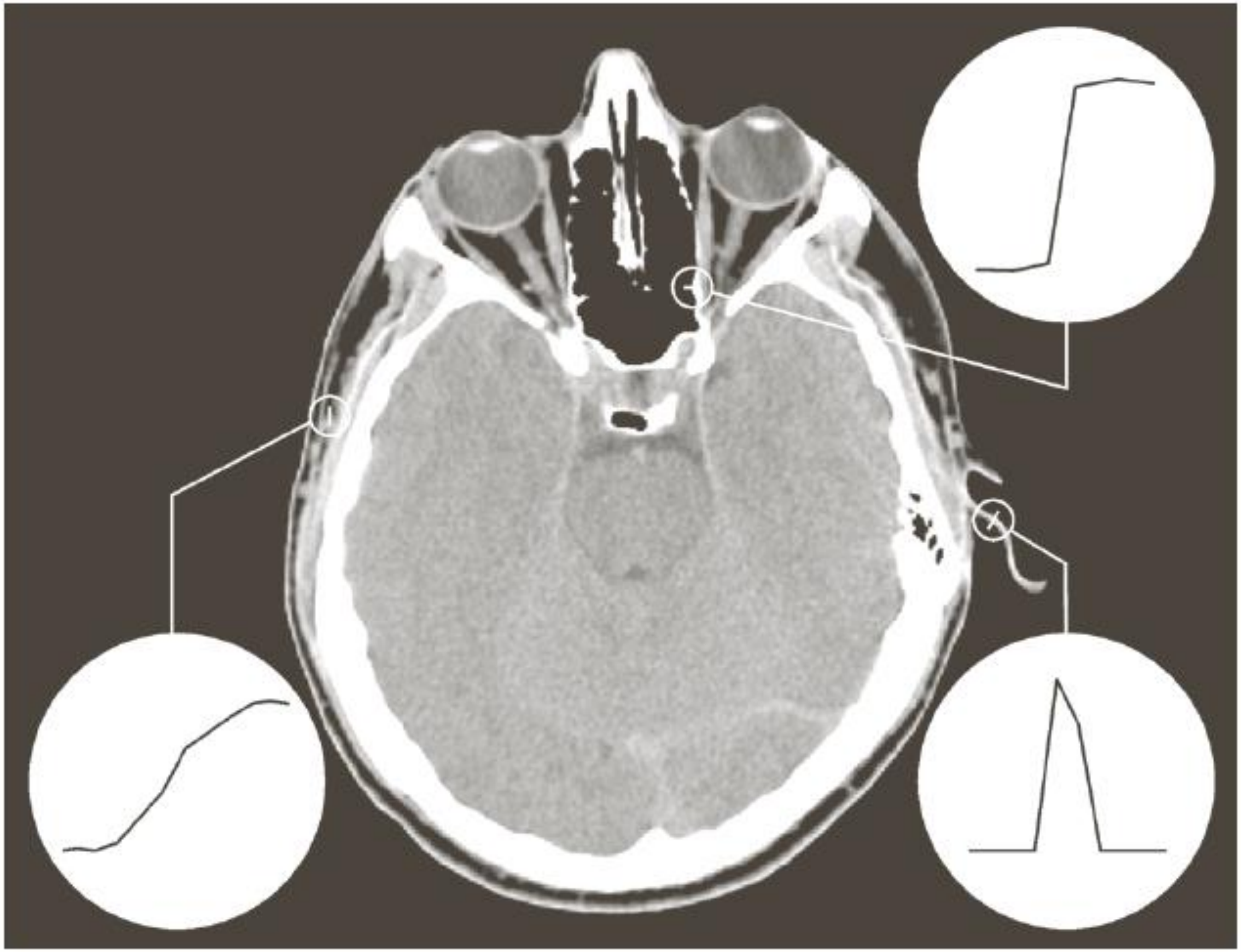


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Gradient Operators

- Most common differentiation operator is the gradient vector.

$$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

Magnitude:

$$\left|\nabla f(x,y)\right| = \left[G_x{}^2 + G_y{}^2\right]^{1/2} \approx \left|G_x\right| + \left|G_x\right|$$

Direction:

$$\angle f(x,y) = \tan^{-1}\left[\frac{G_y}{G_x}\right]$$

Gradient Operators

Some common gradient operators

- Roberts and Prewitt masks are the simplest but not robust against noise
- Sobel edge detection masks are the most common and give satisfactory results in presence of noise

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.

Gradient Operators

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.

Direction of an Edge

•The direction of an edge at a point is orthogonal to the direction of the gradient vector at the point

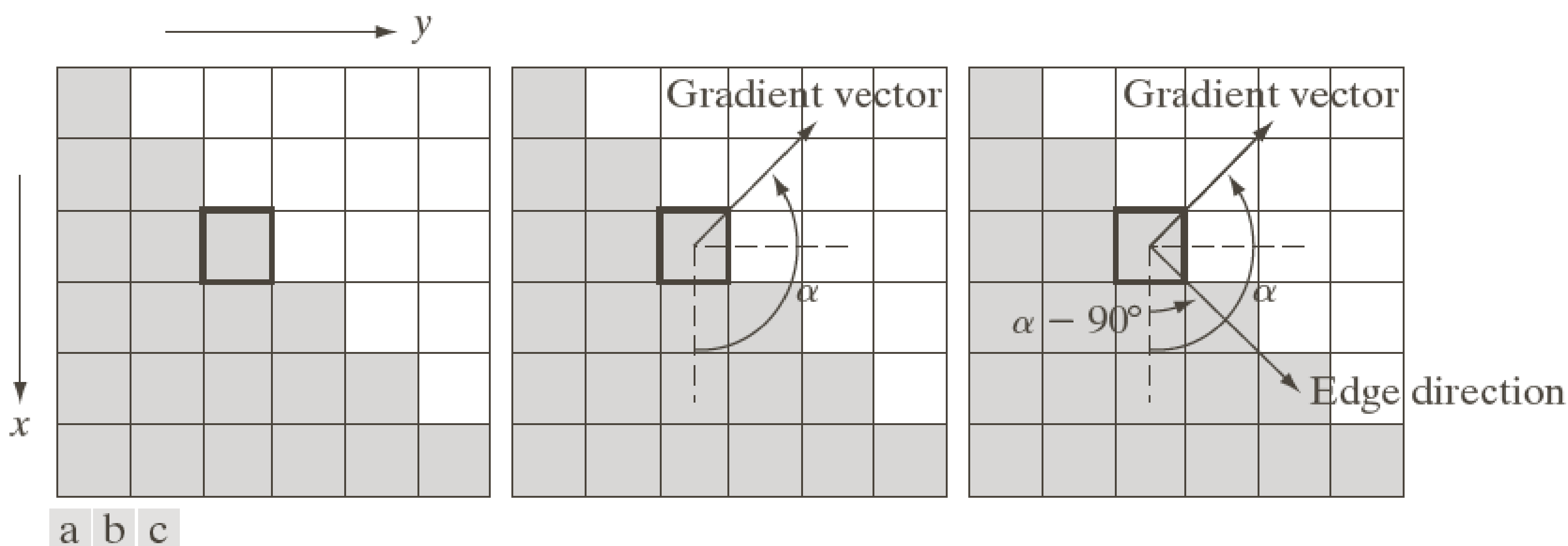


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

a b
c d

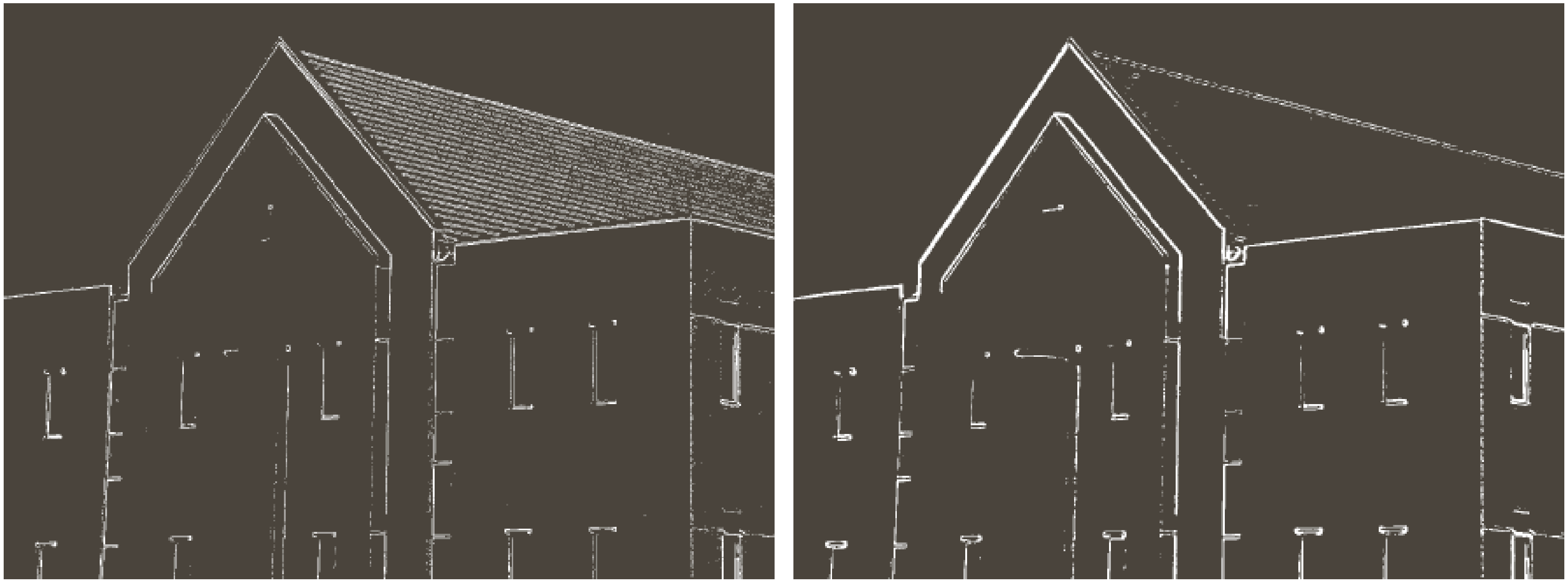
FIGURE 10.16 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The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

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).

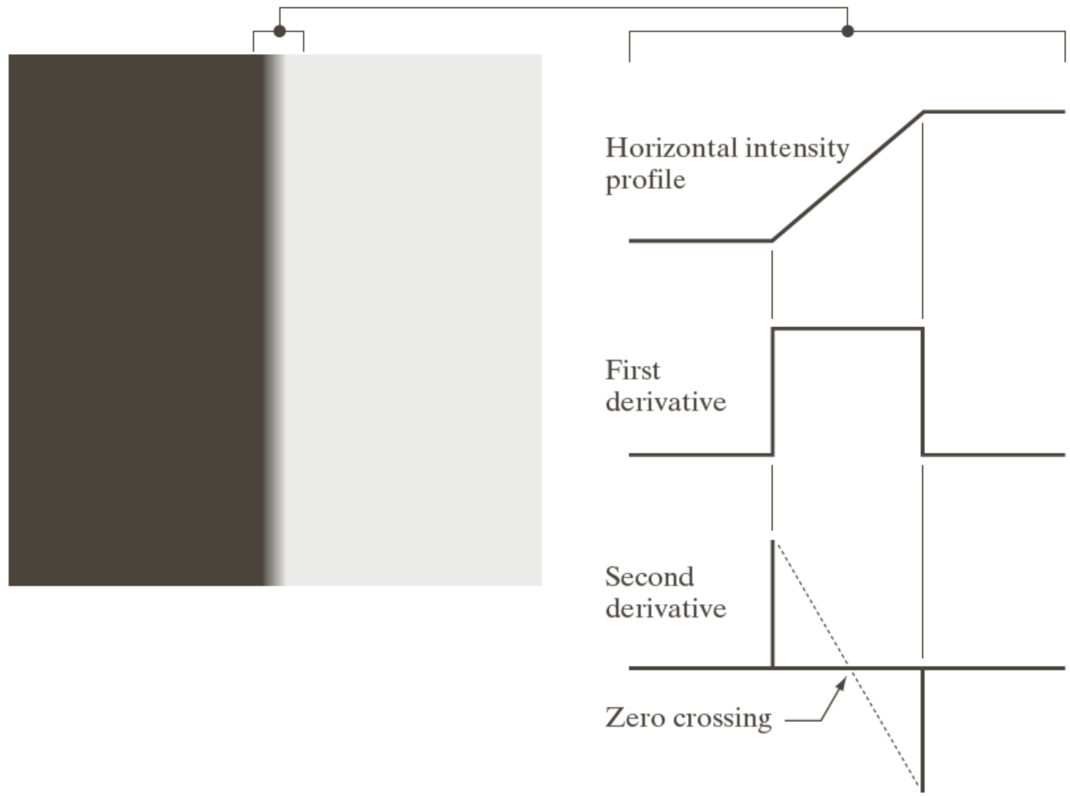


a b

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.

Edge Detection

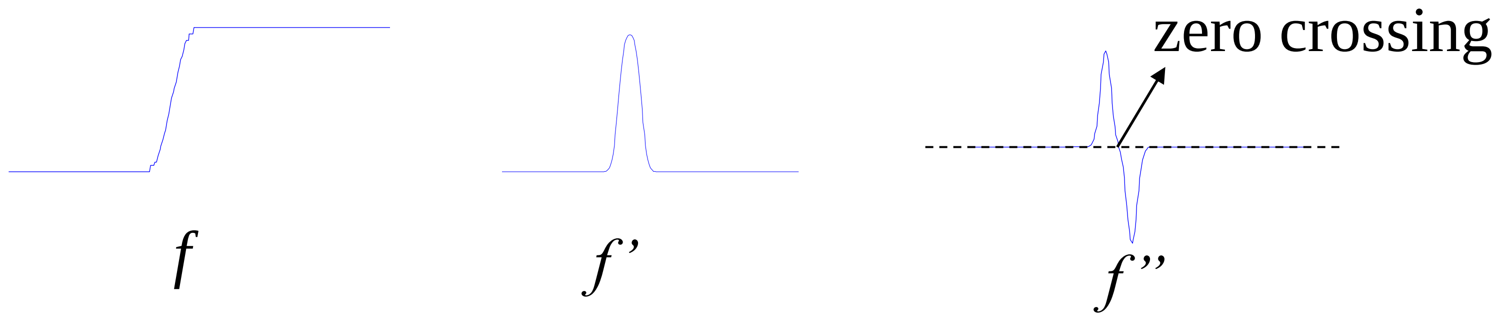
- First and second derivative for smooth noiseless edge
- The zero crossings of the second derivative can be used for locating edges in an image.

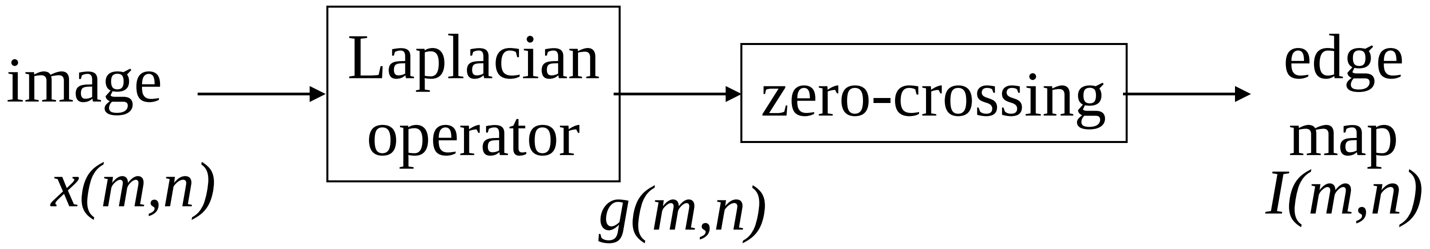


a b

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.

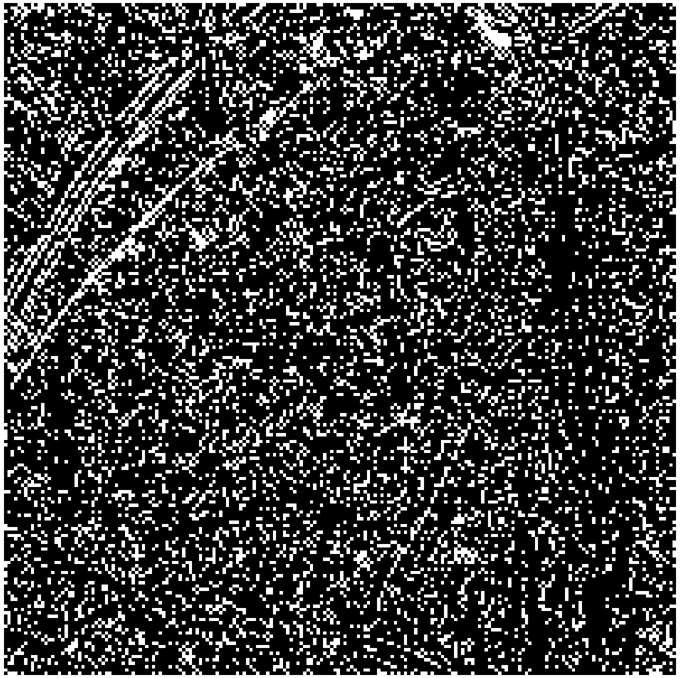
Zero Crossings





Examples

original image

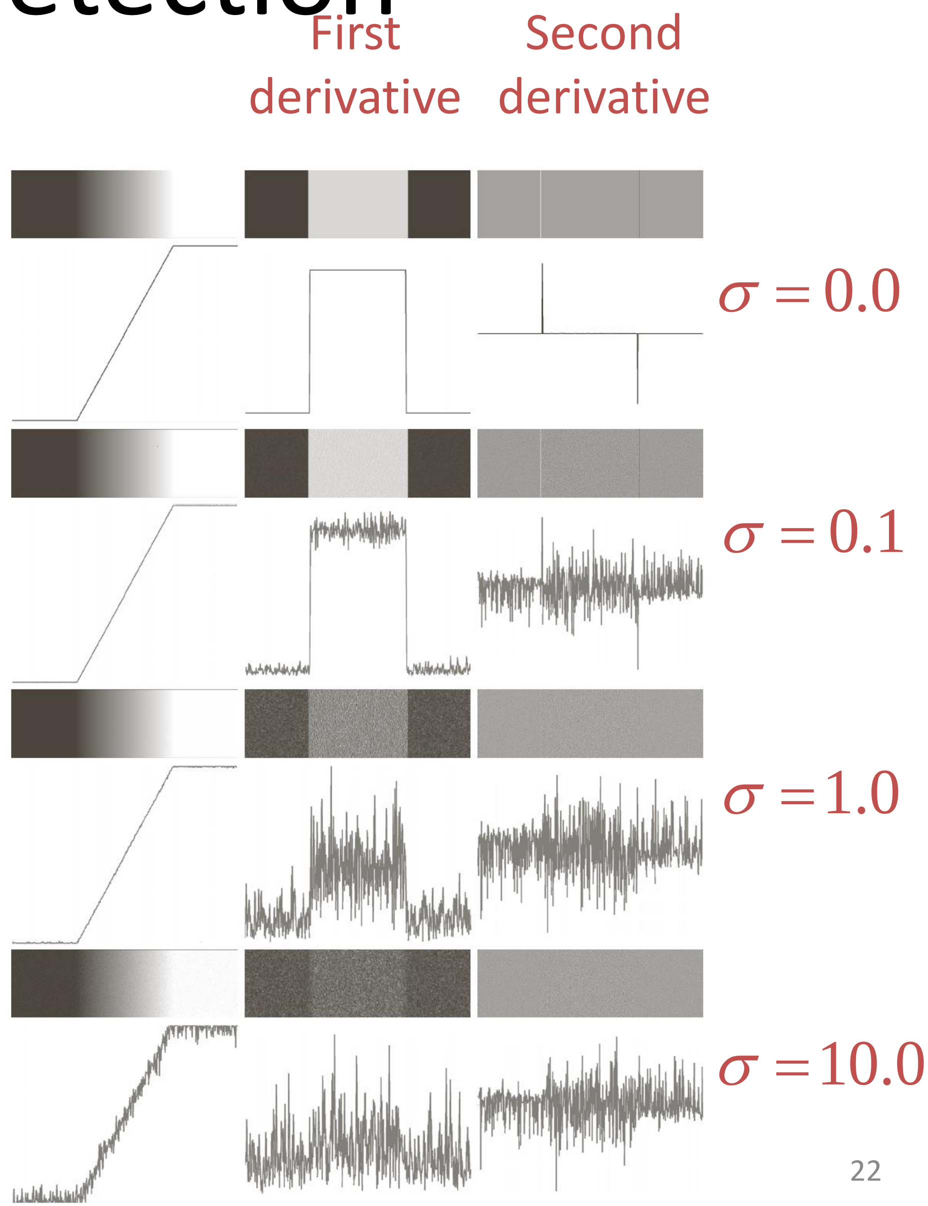


zero-crossings

Question: why is it so sensitive to noise (many false alarms)?
Answer: a sign flip from 0.01 to -0.01 is treated the same as from 100 to -100

Edge Detection

Results of first and second derivative for edges with Gaussian noise of mean = 0.



•Smoothing step is a must before taking derivative for edge detection

Ideas to Improve Robustness

- Linear filtering
  - Use a Gaussian filter to smooth out noise component $\rightarrow$ Laplacian of Gaussian
- Spatially-adaptive (Nonlinear) processing
  - Apply different detection strategies to smooth areas (low-variance) and non-smooth areas (high-variance) $\rightarrow$ Robust Laplacian edge detector
- Return single response to edges (not multiple edge pixels)
  - Hysteresis thresholding $\rightarrow$ Canny's edge detector

Laplacian of a Gaussian (LoG)

- A filter which combines the smoothing function (Gaussian) with the Laplacian is called Laplacian of a Gaussian (LoG) filter.
- Robust against noise.
- Consider a smoothing Gaussian function:

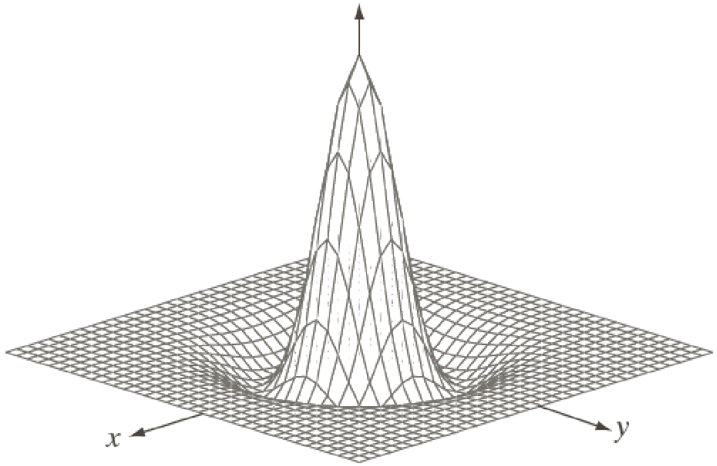
$$G(r) = e^{-\frac{r^2}{2\sigma^2}}$$

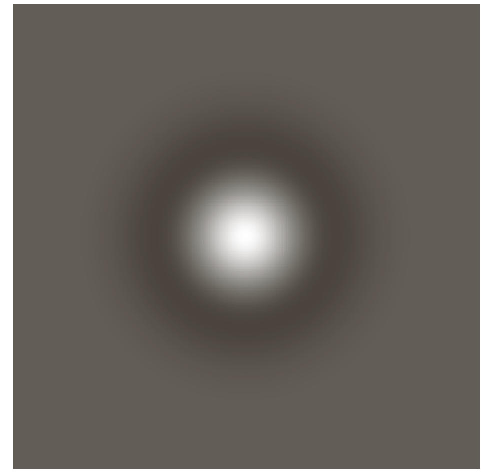
where $r^2 = x^2 + y^2$, σ: standard deviation

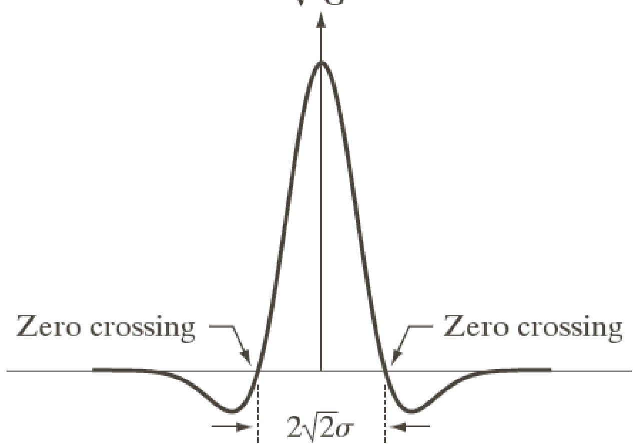
- The Laplacian of this function gives the LoG function:

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

Laplacian of a Gaussian (LoG) Filter







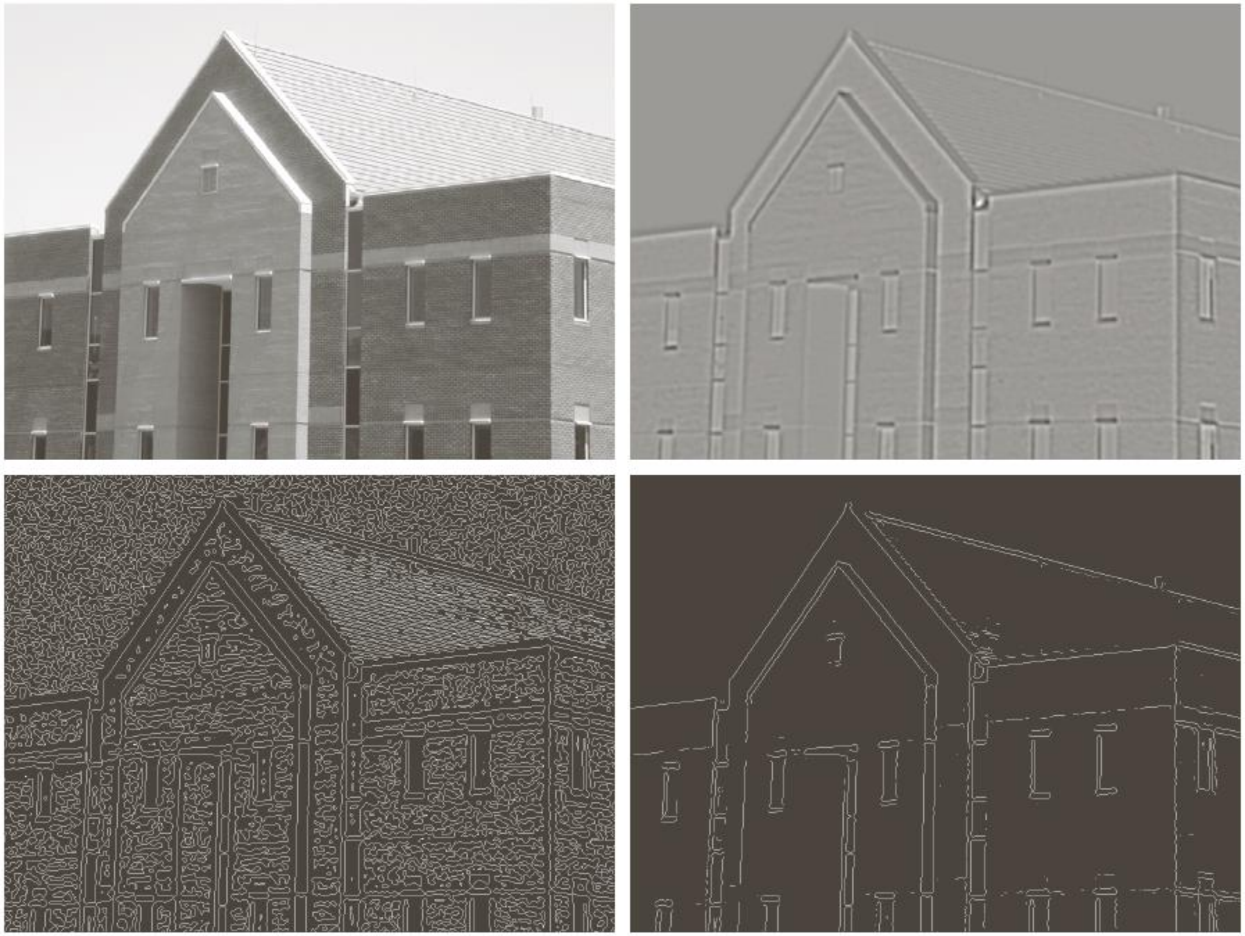
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

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.

Edge detection by LoG

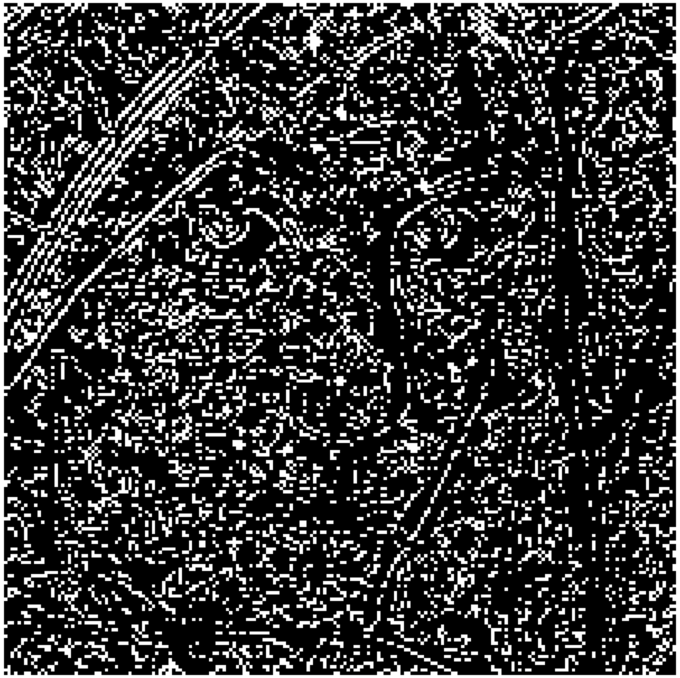
- Due the shape of this function it is also called Mexican hat function (or Mexican hat filters).
- Better performance against noise. Reduces the intensity of structures or noise, which are at scales much smaller than sigma.
- Choose size of Gaussian mask to be n >= 6*sigma
- Then use a 3x3 Laplacian
- Find the zero crossings



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.

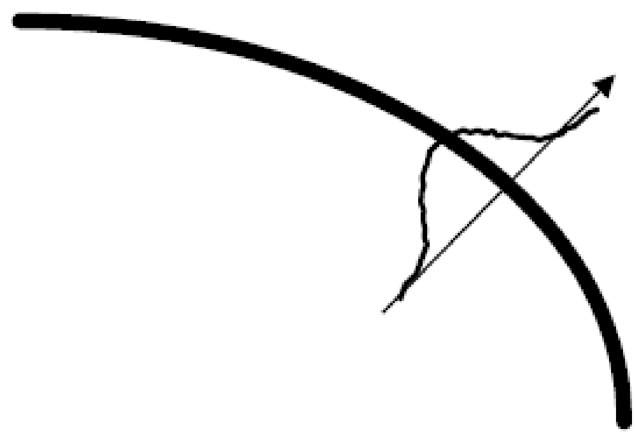
Examples



Better than Laplacian alone but still sensitive due to zero crossing

Combining smoothing and differentiation

- *Non-maxima supression - Retain a point as an edge point if:*
  - *its gradient magnitude is higher than a threshold*
  - *its gradient magnitude is a local maxima in the gradient direction*



simple thresholding will
compute thick edges

Canny Edge Detector

(*J. Canny'1986*)

Original image

↓

Smoothing by Gaussian convolution

↓

Differential operators along x and y axis

↓

Non-maximum suppression
finds peaks in the image gradient

↓

Hysteresis thresholding locates edge strings

↓

Edge map

Canny Edge Detector (smoothing and enhancement)

CANNY_ENHANCER

Given image I

1. Apply Gaussian Smoothing to I.
2. For each pixel (i, j):
   1. Compute the gradient components $I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}$
   2. Estimate the edge strength $\sqrt{I_x^2 + I_y^2}$
   3. Estimate the orientation of the edge normal

$$tan^{-1}\frac{I_x}{I_y}$$

Canny Edge Detector (Nonmaximum suppression)

The input is the output of CANNY_ENHANCER. We need to thin the edges. Given Es, Eo, the edge strength and orientation images. For each pixel (i, j),

1. Find the direction best approximate the direction Eo(i, j).
2. If Es(i, j) is smaller than at least one of its two neighbors along this direction, suppress this pixel.

The output is an image of the thinned edge points after suppressing nonmaxima edge points.

Canny Edge Detector (Hysteresis Thresholding)

Performs edge tracking and reduces the probability of false contours.

Input I is the output of nonmaximum_suppression, Eo and two threshold parameters

Scan I in a fixed order:

$$\tau_l < \tau_h$$

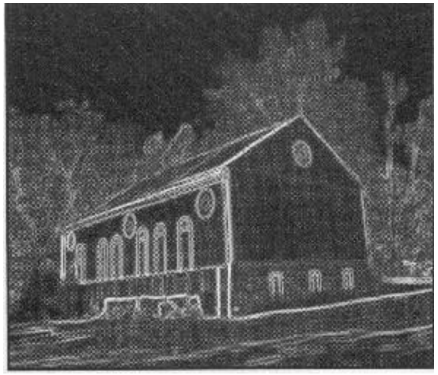
1. Locate the next unvisited edge pixel (i, j) such that I(i, j) $> \tau_h$
2. Starting from (I, j), follow the chains of connected local maxima in both directions perpendicular to the edge normal as long as I $> \tau_l$
3. Marked all visited points and save a list of the locations of all points in the connected contour found.

Hysteresis thresholding

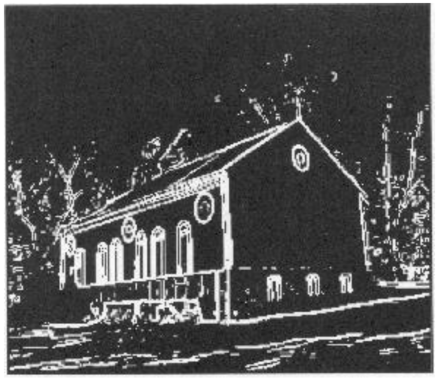
- Standard thresholding:

  - Can only select “strong” edges.
  - Does not guarantee “continuity”.

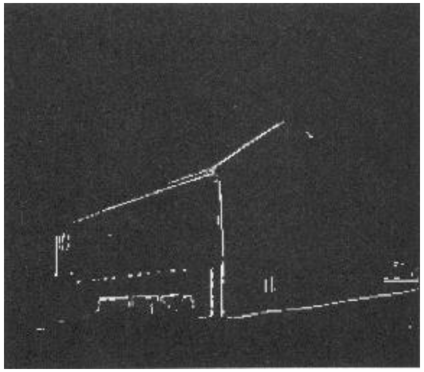
gradient magnitude



low threshold



high threshold



Hysteresis thresholding (cont'd)

- Hysteresis thresholding uses two thresholds:

  - low threshold t_l
  - high threshold t_h (usually, $t_h = 2t_l$)

$$\|\nabla f(x,y)\| \geq t_h \quad \text{definitely an edge}$$

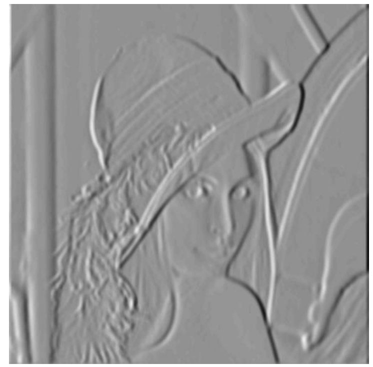
$$t_l \geq \|\nabla f(x,y)\| < t_h \quad \text{maybe an edge, depends on context}$$

$$\|\nabla f(x,y)\| < t_l \quad \text{definitely not an edge}$$

- For "maybe" edges, decide on the edge if <u>neighboring</u> pixel is a strong edge.

Canny Edge Detector Example

original image



vertical edges

horizontal edges

norm of the gradient

after thresholding

after thinning

Marr and Hildreth's Method*

Edge is scale-dependent

↓

A different edge map can be generated at different scale

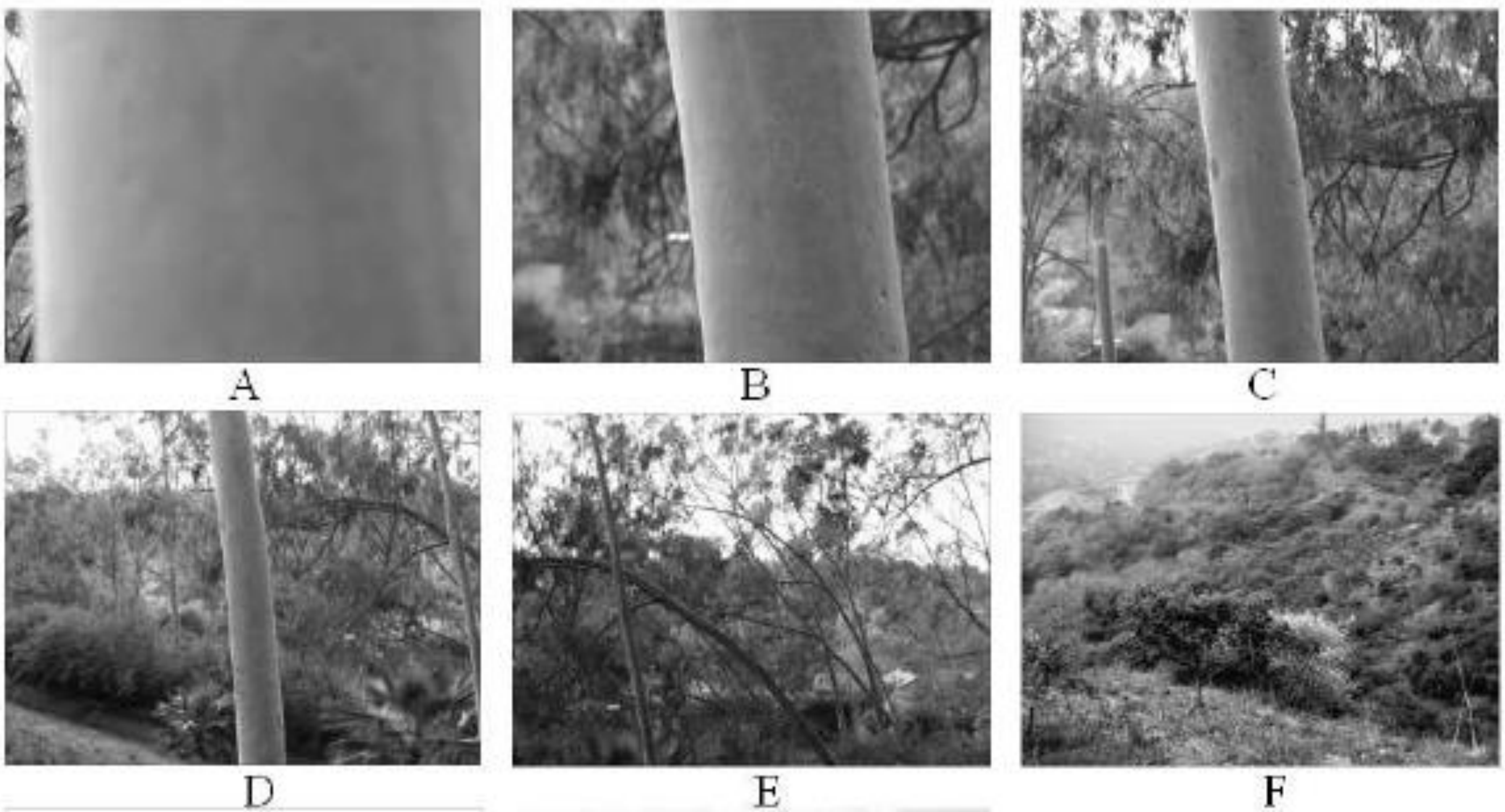
- Scale space representation

$$f(x,y;s) = f(x,y;0) * g(x,y;s)$$

coarse-scale image | *fine-scale image* | *Gaussian kernel with width of s*

$$g(x,y;s) = \frac{1}{\sqrt{2\pi s}} \exp\left(-\frac{x^2+y^2}{2s}\right)$$

Importance of Scale



Scale-Space Edge Detection Examples

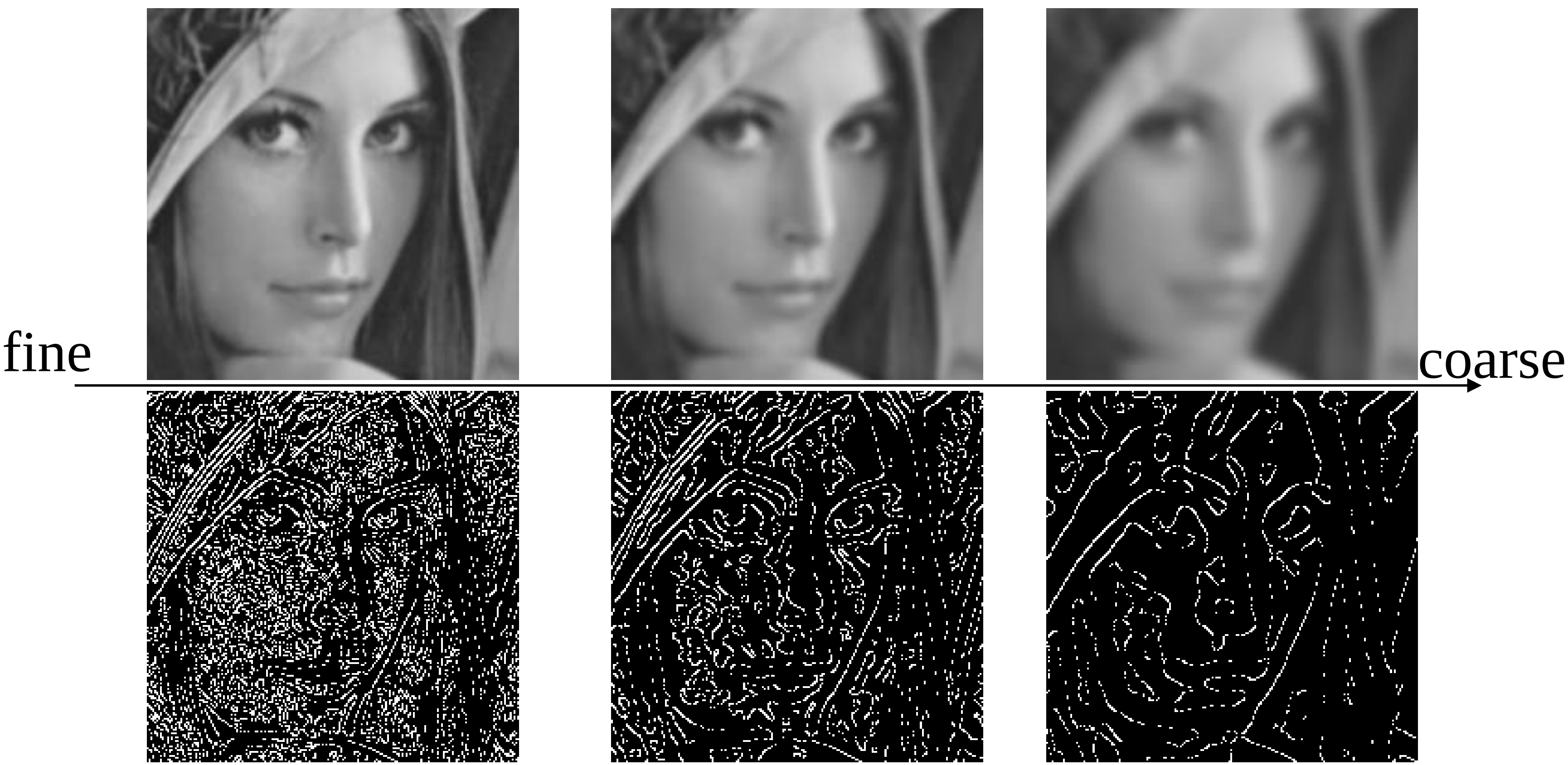


Image to Sketch Online Apps

http://sporkforge.com/imaging/sketch.php

Readings from Book (3rd Edn.)

- Sharpening Filters (Chapter – 3)
- Edge Detection (Chapter – 10)
- 10.2

Reading Assignment

- High Boost Filtering (Chap – 3)
- Laplacian of Gaussian (LoG) (Chap – 10)

Acknowledgements

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Computer Vision for Computer Graphics, Mark Borg

Material in these slides has been taken from, the following resources