Edge Detection

Computer Vision 2017

P. Zanuttigh
Figures from Gonzalez Woods
Some slides from B. Girod, S. Seitz, E. Petrakis

Feature Detection

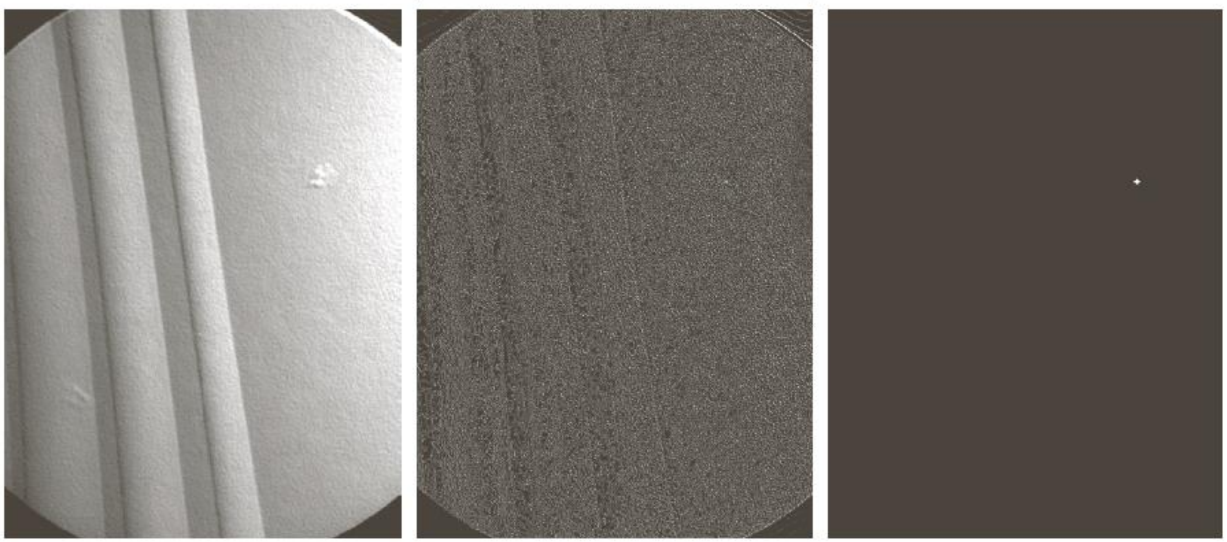
Extract relevant features from the images

- Points
- Lines
- **Edges**
- Corners
- Blobs
- Other…

- ***Feature detection***: extract the features of interest
- ***Feature description***: associate a descriptor to each feature point in order to distinguish it from the other feature points

Isolated Points

1	1	1
1	−8	1
1	1	1

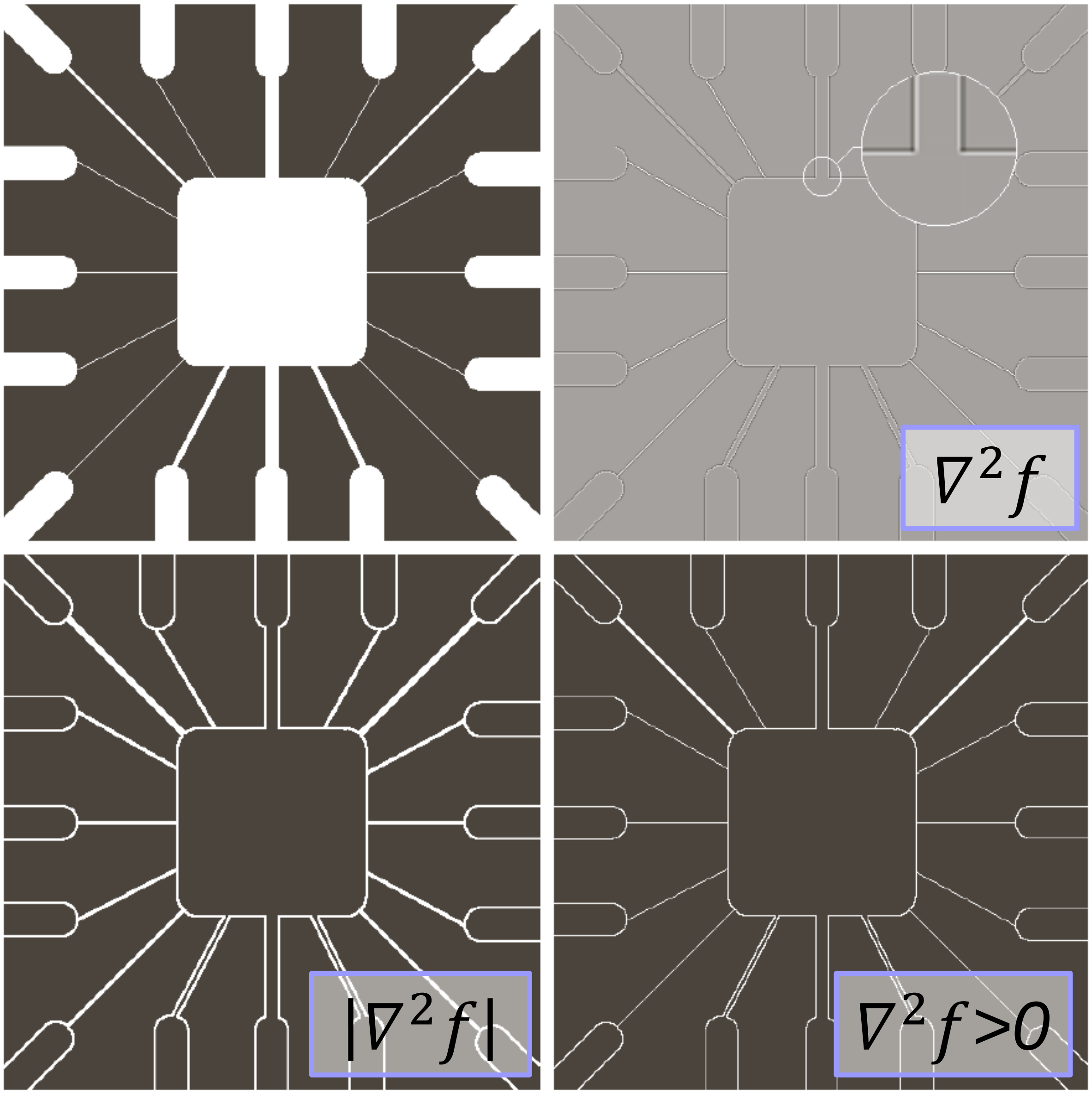


a
b c d

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.)

- *Apply the Laplacian mask*
- *Thresholding of the output* $|\nabla^2 f| > T$

A Simple Line Detector: The Laplacian



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.

Find Lines Along a Chosen Direction

−1	−1	−1
2	2	2
−1	−1	−1

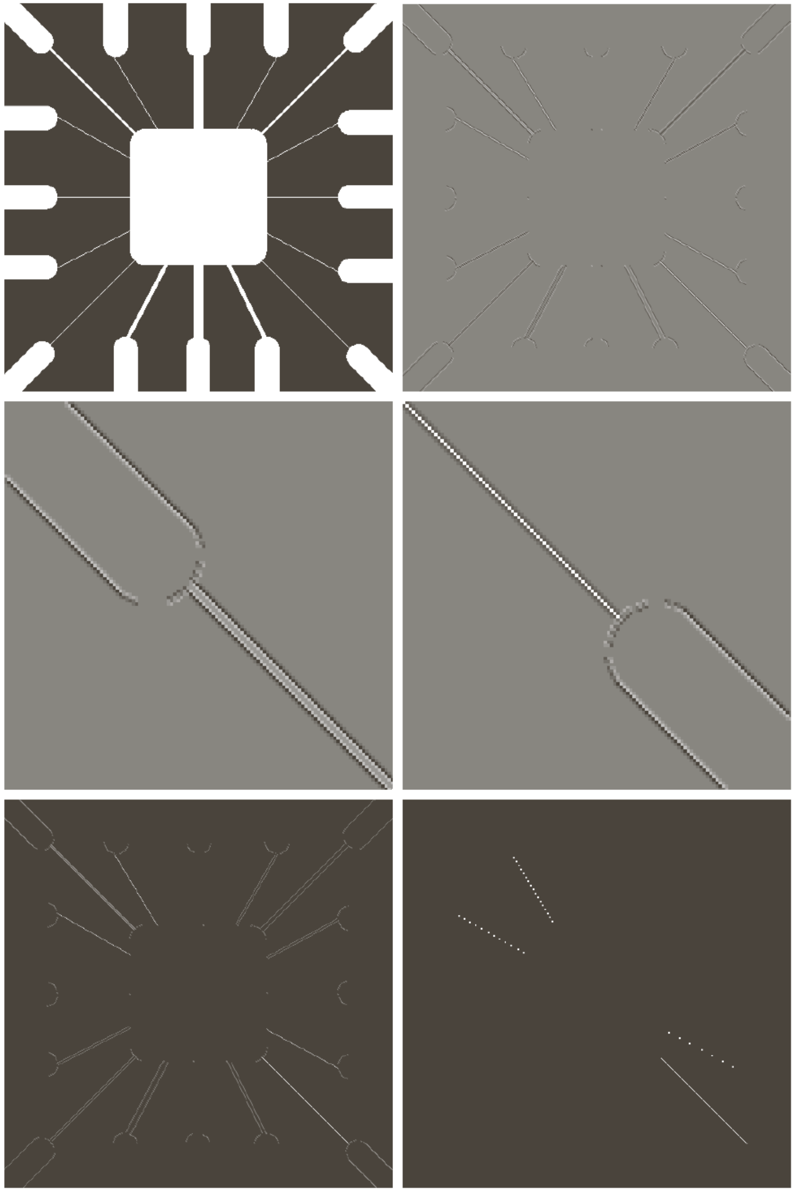
2	−1	−1
−1	2	−1
−1	−1	2

−1	2	−1
−1	2	−1
−1	2	−1

−1	−1	2
−1	2	−1
2	−1	−1

Oriented Masks
(derived from the Laplacian)

Search for Lines (Example)



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.)

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

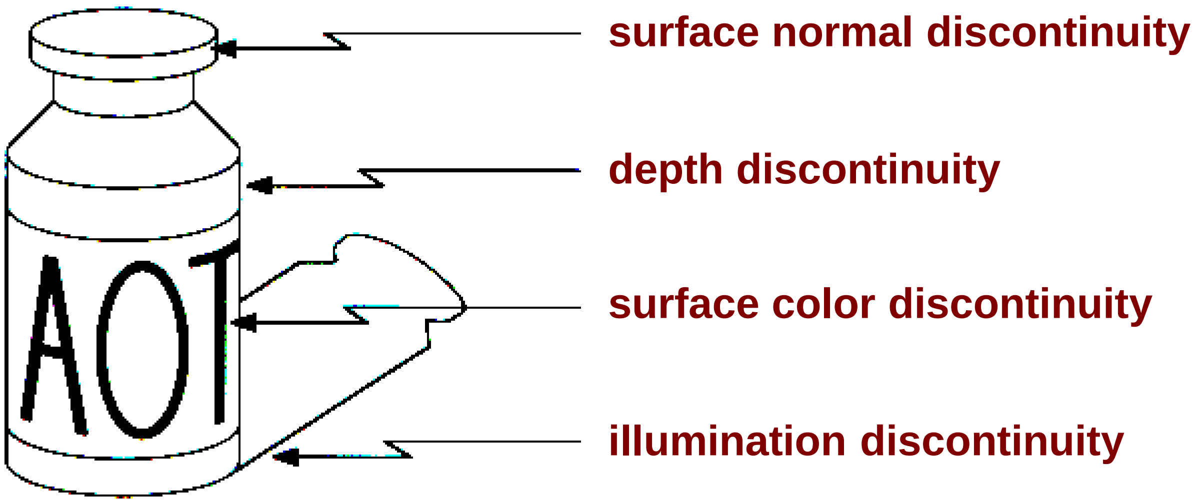
−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

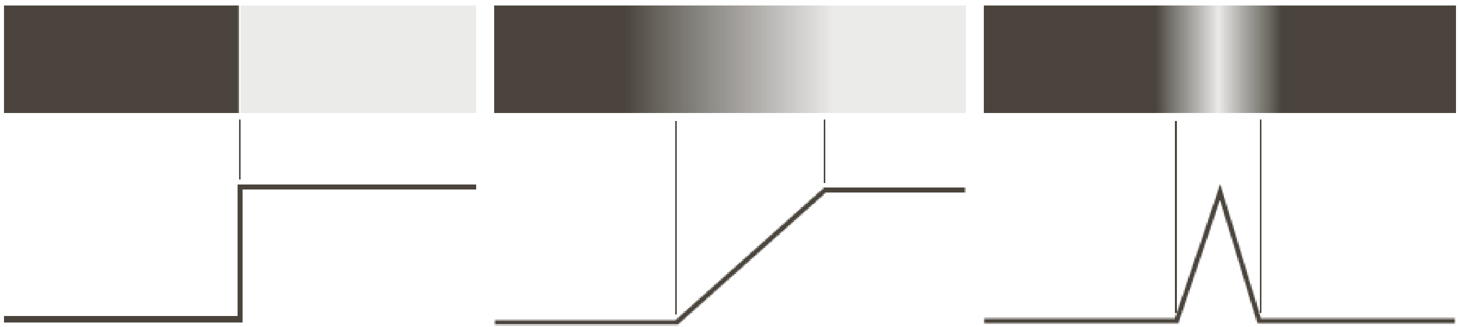
−45°

Origin of Edges



Edges can be caused by a variety of factors

Different Types of Edges



a) Step b) Ramp c) Roof

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.

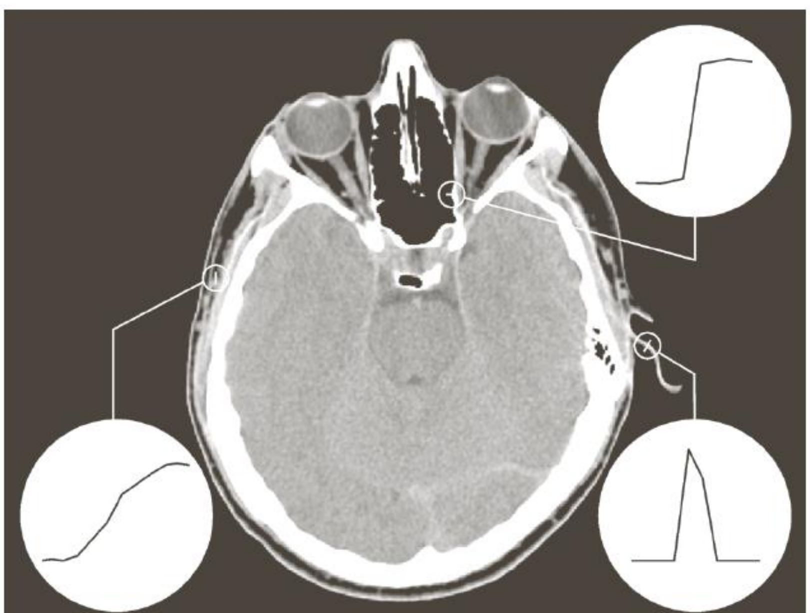


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

First and Second Order Derivatives

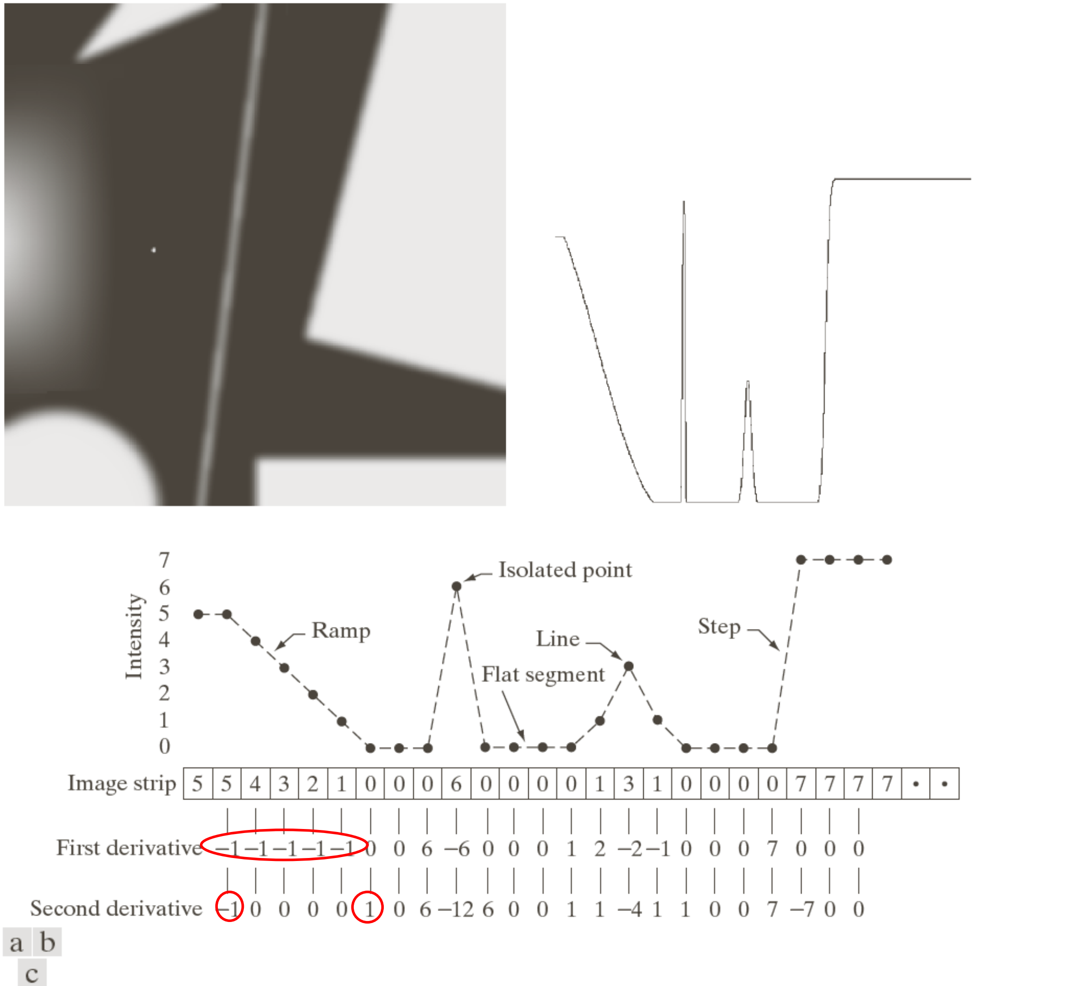
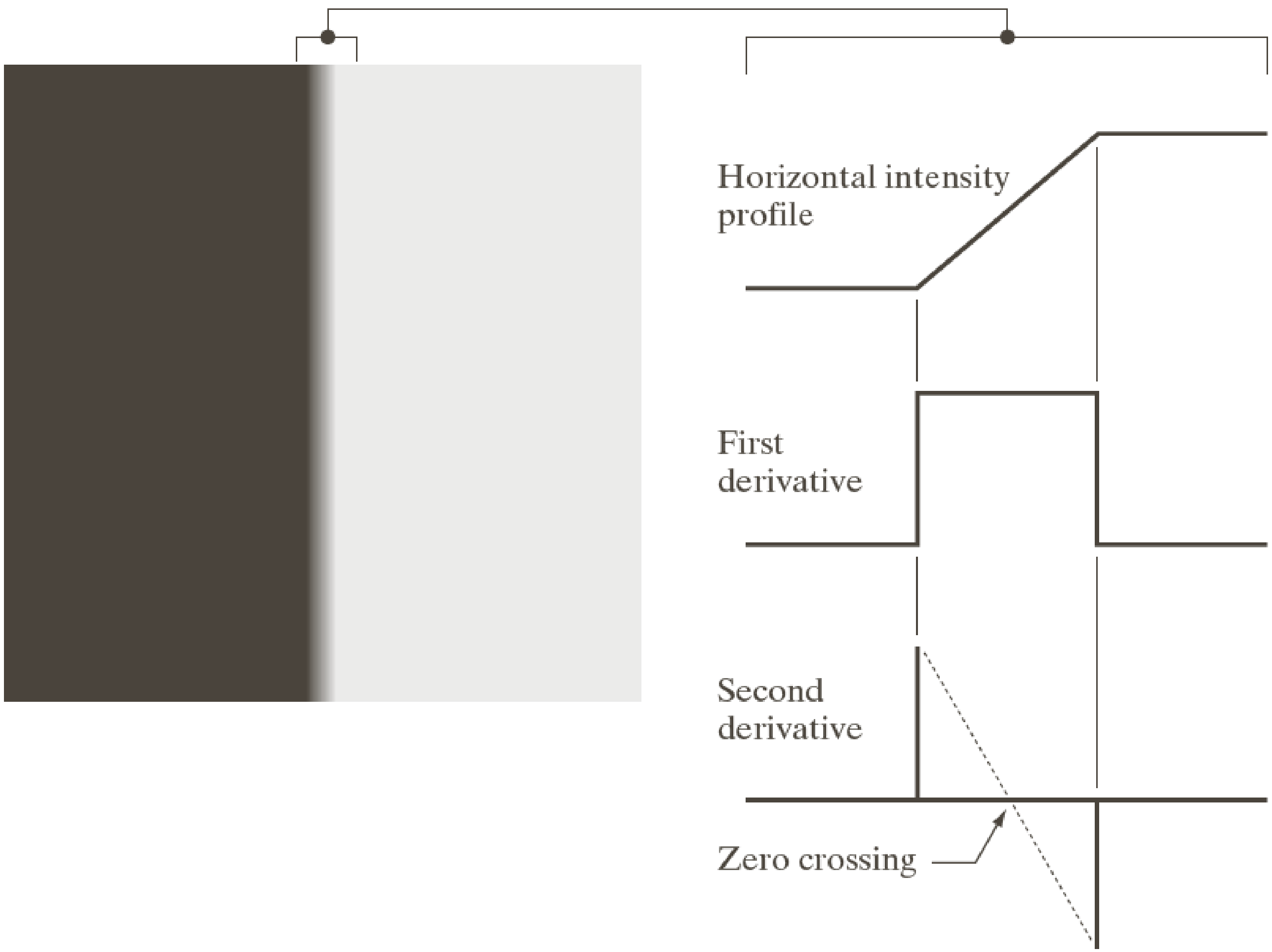


FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values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

	1st order derivative	***2nd order derivative***
Large values	On all the edge	Start and end of the edge
Edge type	Single	Double
Edge thickness	Thick	Thin
Noise sensitivity	Moderate	High

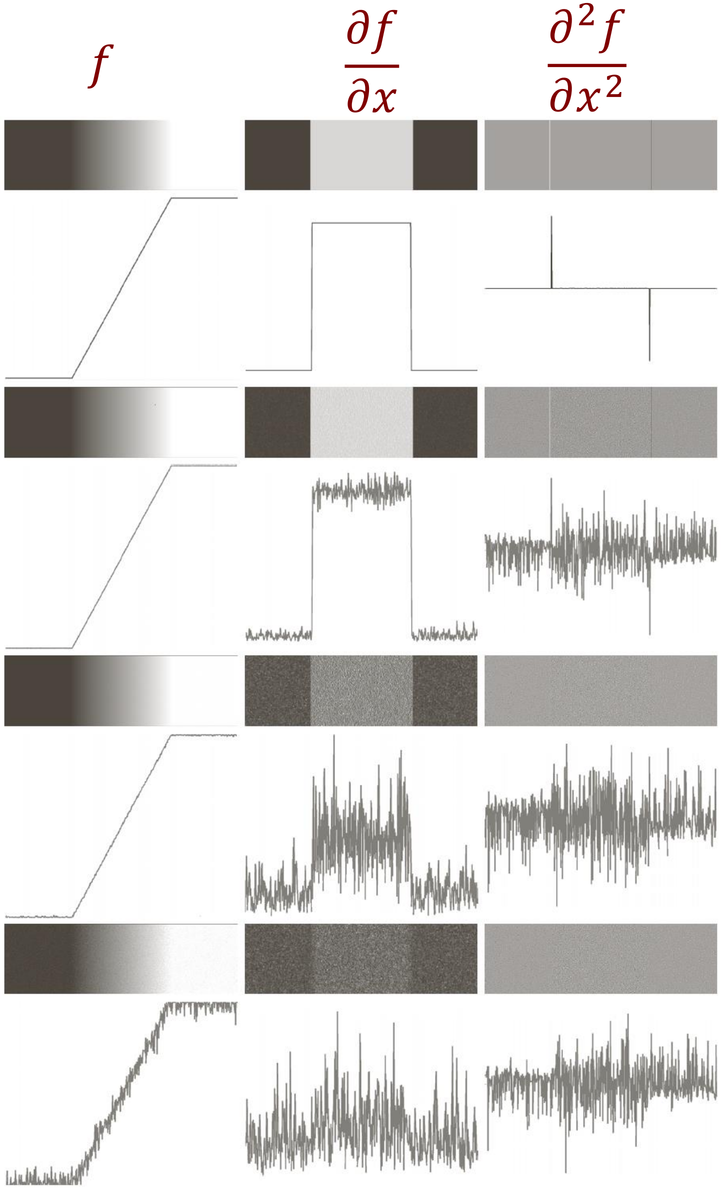
Edge and Derivatives



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.

Derivatives are Quite Sensible to Noise



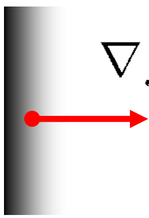
- Noise strongly affect the derivative computation
- 2nd order derivative is particularly sensible

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.

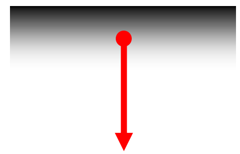
Image Gradient (1)

- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

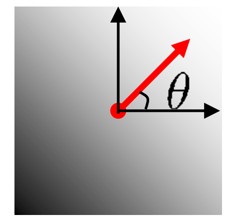
- The gradient points in the direction of most rapid change in intensity



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The gradient direction is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

- how does this relate to the direction of the edge?

The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Image Gradient (2)

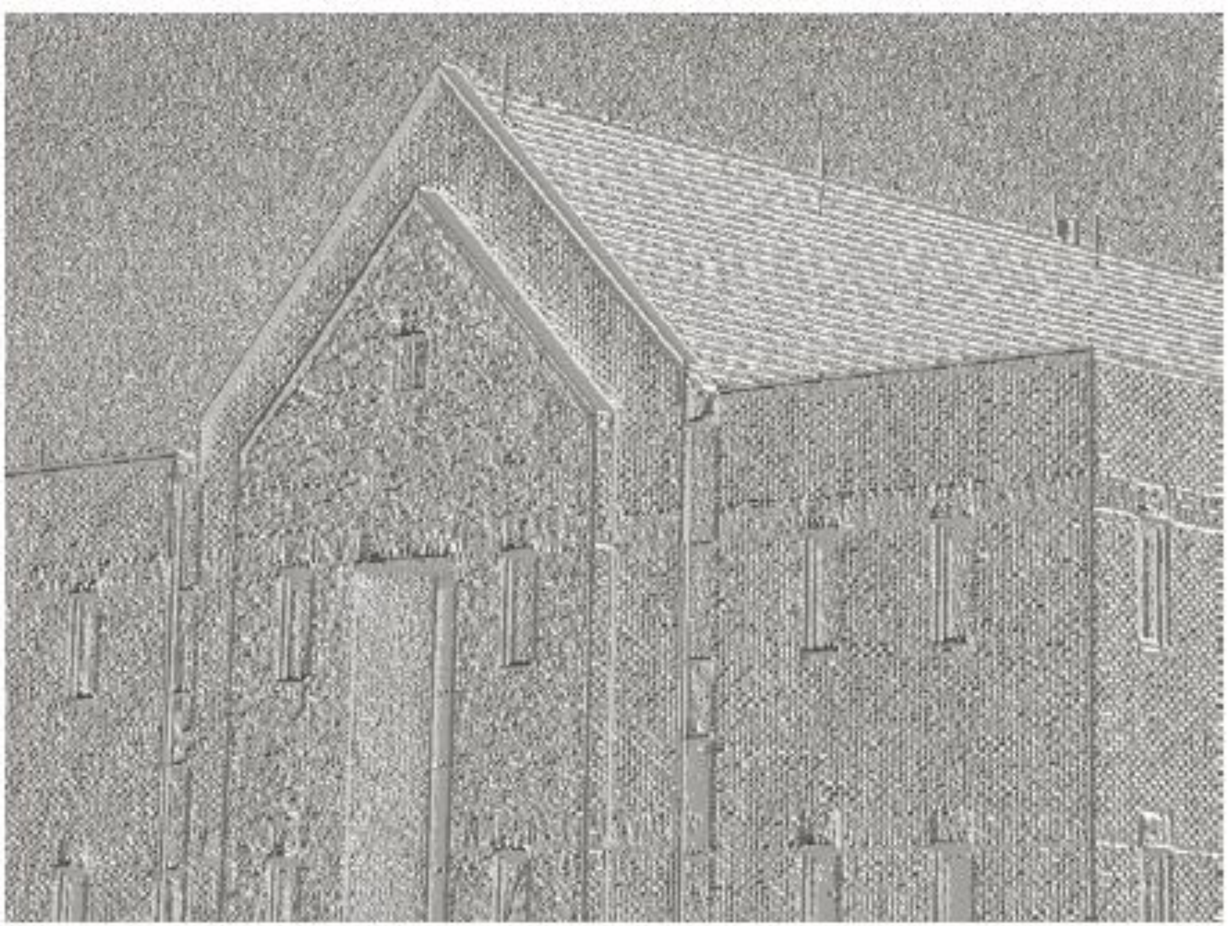
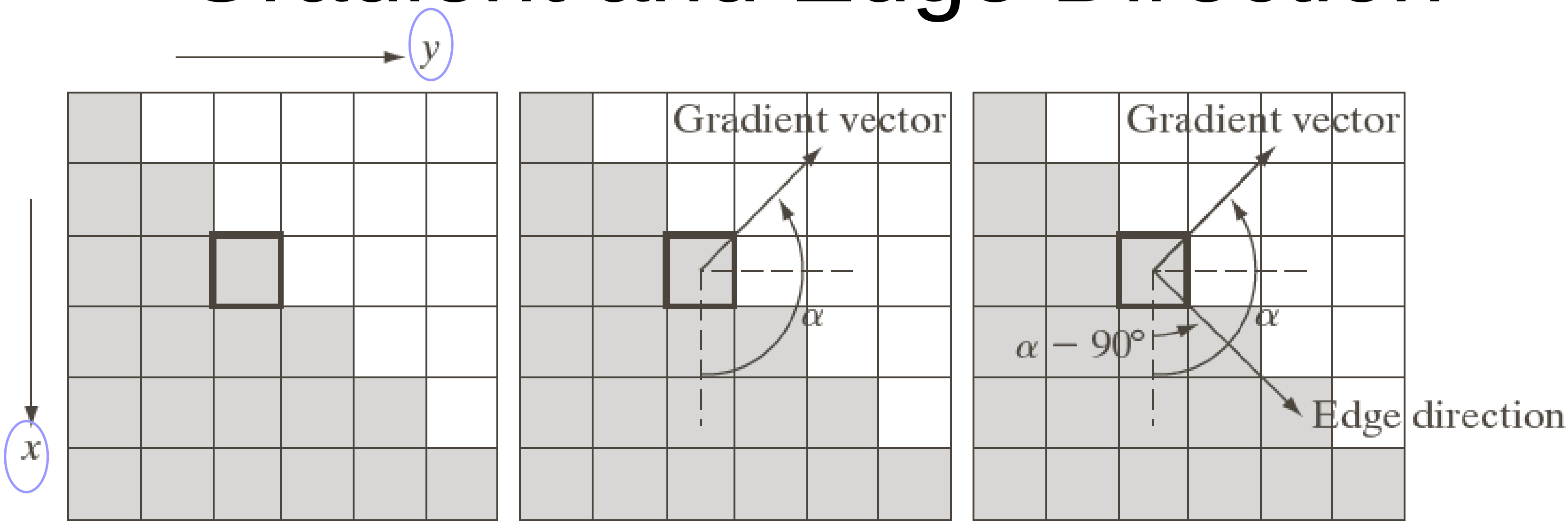


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

2 Informaton types

- *Module*
- *Direction*

Gradient and Edge Direction



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.

Edge: $|\nabla f| = \sqrt{f_x^2 + f_y^2} > T$

The orientation is perpendicular to $\alpha = \tan^{-1}\left(\frac{f_y}{f_x}\right)$

Gonzalez book : different x and y convention

Masks for Oriented Gradient

−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

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

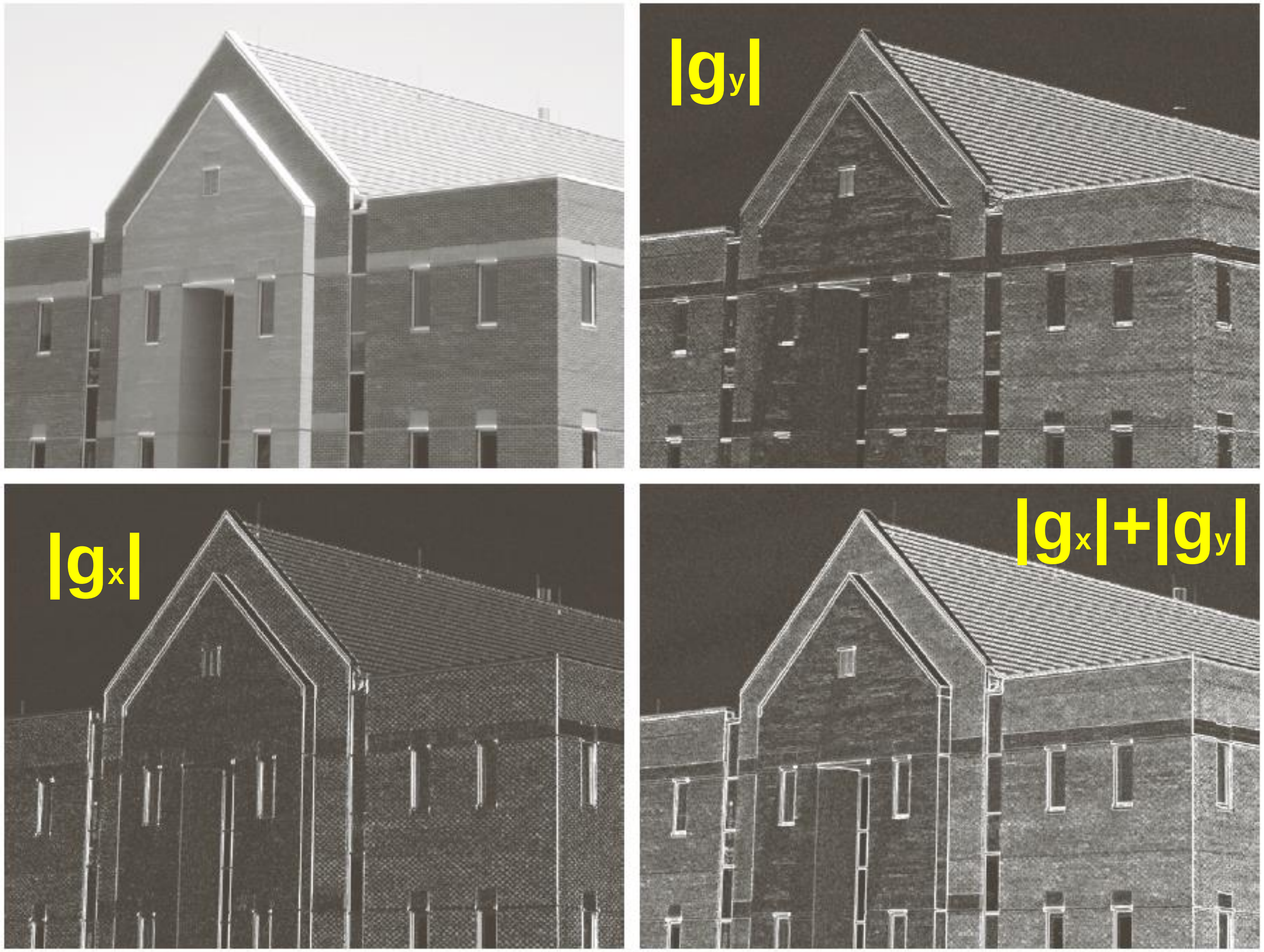
Sobel

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Sobel Masks



$|g_y|$

−1	−2	−1
0	0	0
1	2	1

$|g_x|$

−1	0	1
−2	0	2
−1	0	1

Sobel

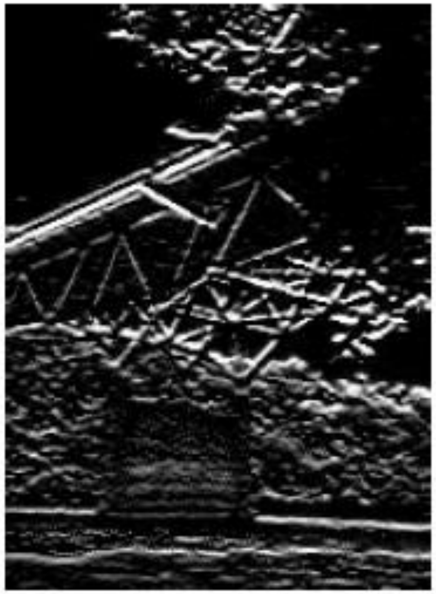
Demo

Prewitt Masks

Original *Bridge*
220x160

magnitude of
image filtered with

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$$



magnitude of
image filtered with

$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Demo

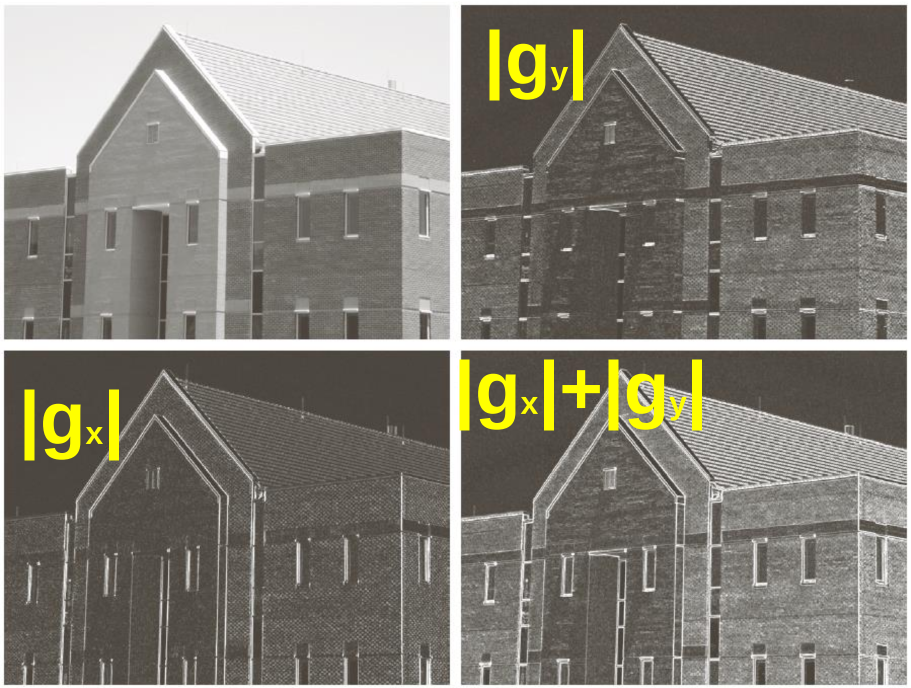
A Simple Edge Detector

3 Steps

1. Low pass filtering (noise removal)
2. Compute the Gradient ∇f
3. Thresholding of the gradient module $|\nabla f| > T$

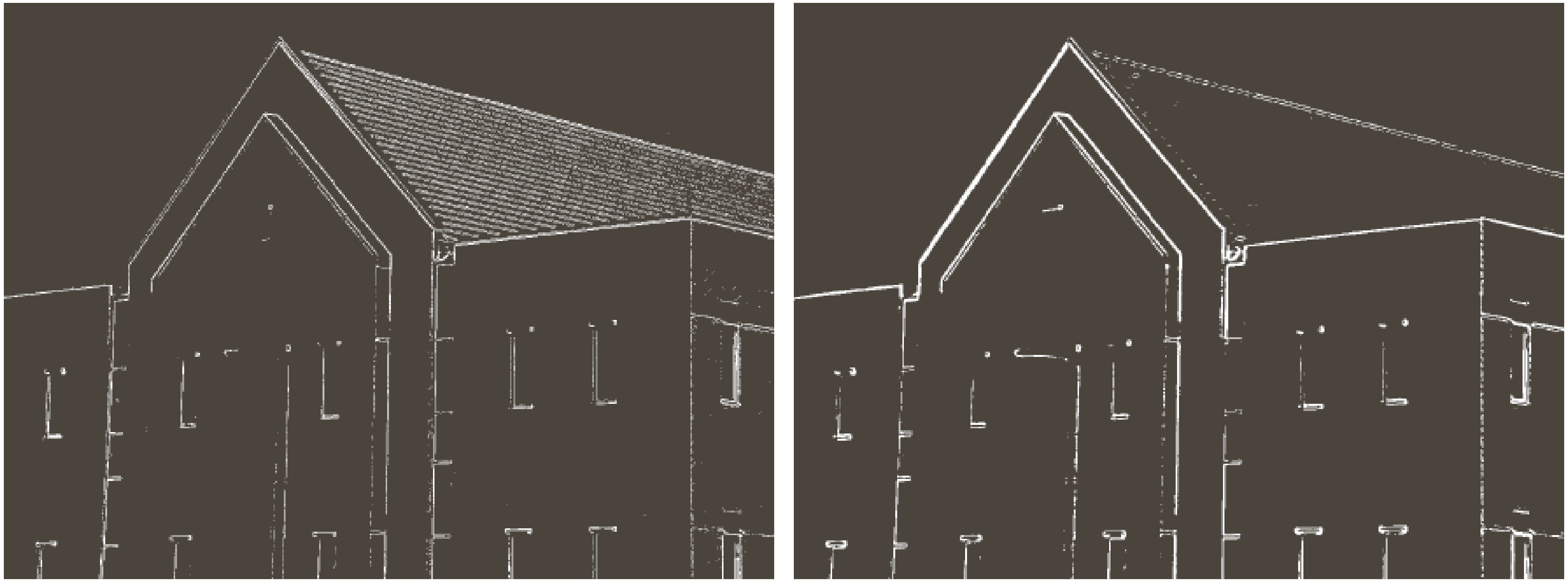
1 2 3

Low Pass Filtering and Edge Detection



Low pass smoothing 5x5

Thresholding



Without smoothing *With 5x5 smoothing*

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.

Diagonal Edges

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).

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

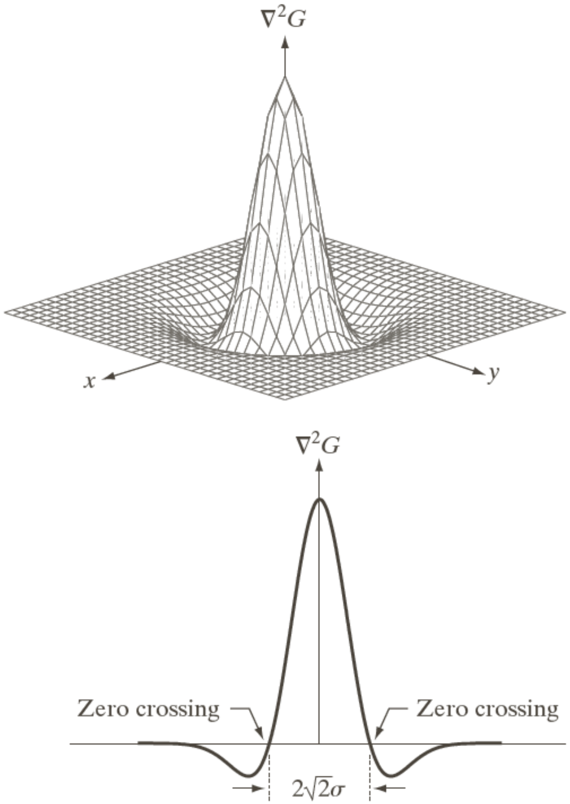
Marr-Hilderth Edge Detector

- Exploit the *second order derivative* for edge detection, but two main issues
  → Very sensitive to noise
  → Double edges

- Noise sensitivity
  → Need a differential operator that is also robust to noise

- Single Edge
  →maxima of *f'* or *zero crossing of f''*

Laplacian of a Gaussian (LoG)



plot of -LoG

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.

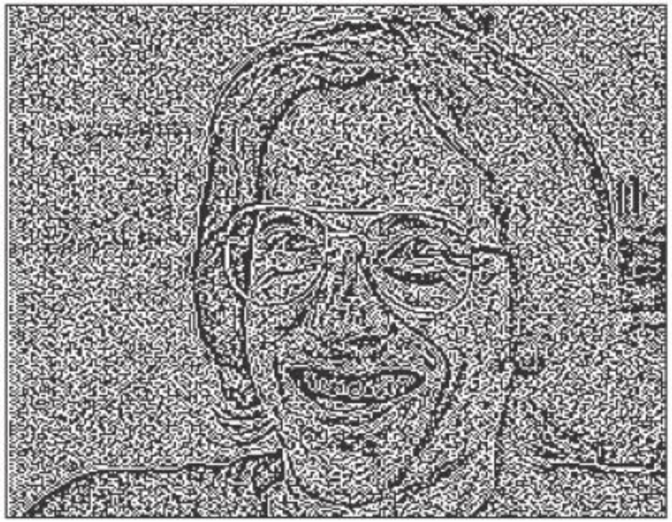
- *Smoothing*: noise removal
- *No ringing*: no false edges
- *Isotropic* (Laplacian)
- Matches the Human Visual System characteristics

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

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

** Book definition: missing* $1/2\pi\sigma^2$

Marr-Hildreth and Laplacian Issues



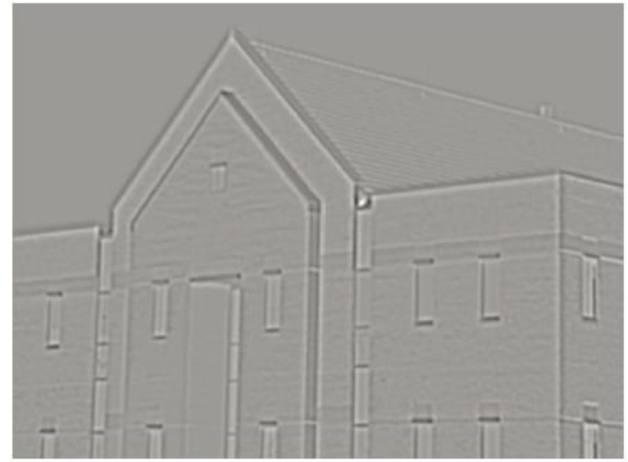
- Sensitive to very fine detail and noise ➔ blur image first
- Responds equally to strong and weak edges
  ➔ suppress edges with low gradient magnitude

Gaussian Filter

Threshold

Marr-Hildreth Edge Detector: Procedure

LoG



a	b	c
d	x	e
f	g	h

Zero crossing:

$$(ah<0)\vee(bg<0)\vee(cf<0)\vee(de<0)$$

Threshold

(on couples of zero-crossing)

Zero Crossing of the LoG

w/o Gaussian

$\sigma = 1.4$

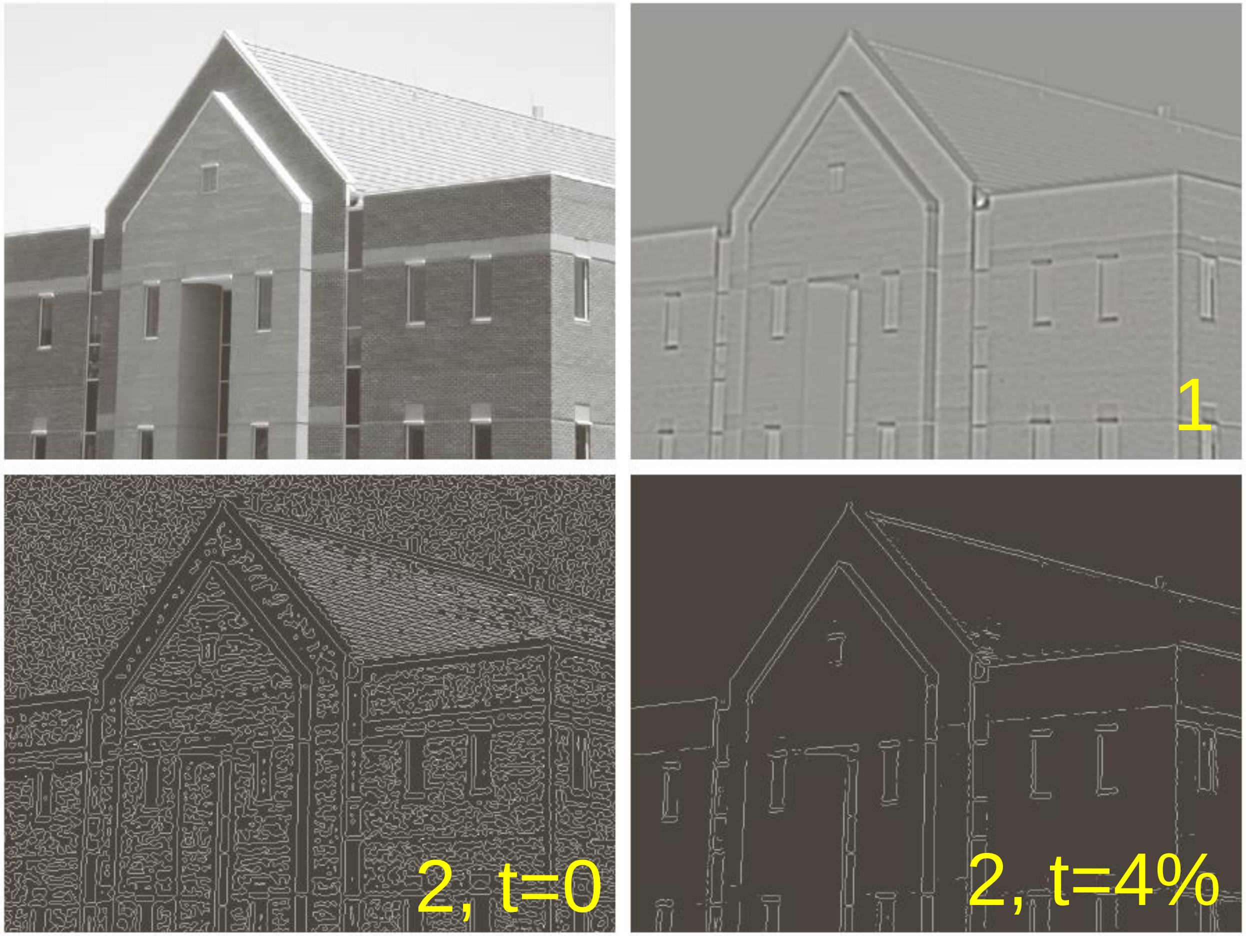
$\sigma = 3$

$\sigma = 6$

Parameter	Value
σ (Gaussian filter)	Remove structures of size much smaller than σ
n (mask size: nxn)	n ≥ 6σ (99.7% volume)
T (zero-crossing threshold)	4% max{LoG}

The locuses of the zero-crossings have a closed loop shape

Marr-Hildreth Edge Detector



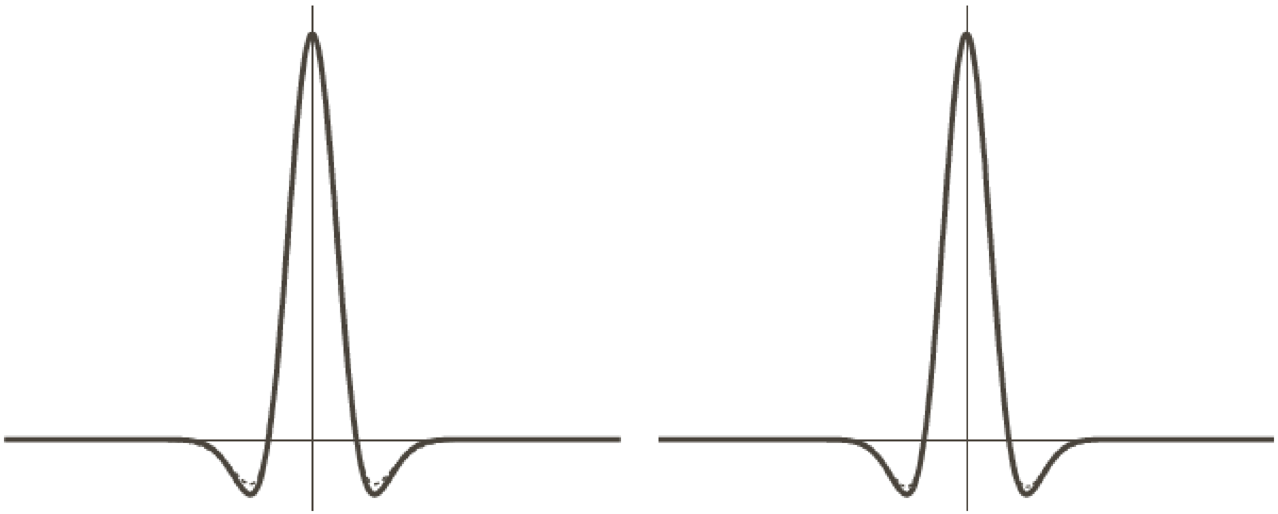
a b
c d

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.

1. Filter the image with the LoG
2. Find zero-crossing of the result

Matlab example

LoG vs DoG



a b

FIGURE 10.23 (a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1. (b) Profiles obtained using a ratio of 1.6:1.

*The plots show –LoG and -DoG

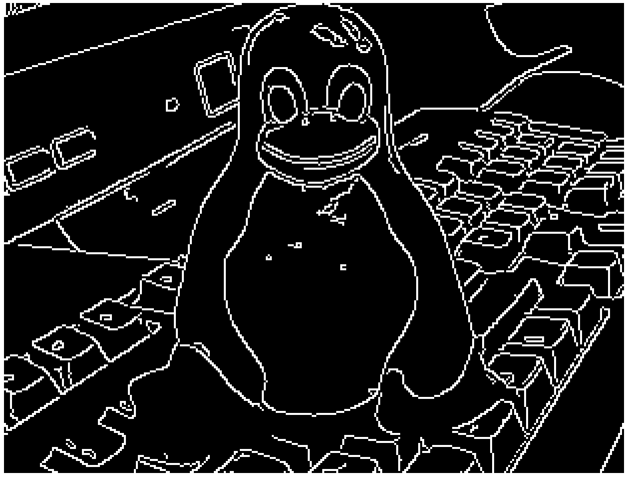
$$LoG: g(x,y) = \nabla^2\left(\frac{1}{2\pi\sigma_{LoG}^2} e^{-\frac{x^2+y^2}{2\sigma_{LoG}^2}}\right)$$

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

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

$$\sigma_1 > \sigma_2 \qquad (\sigma_1 = 1.6\sigma_2)$$

Canny Edge Detector

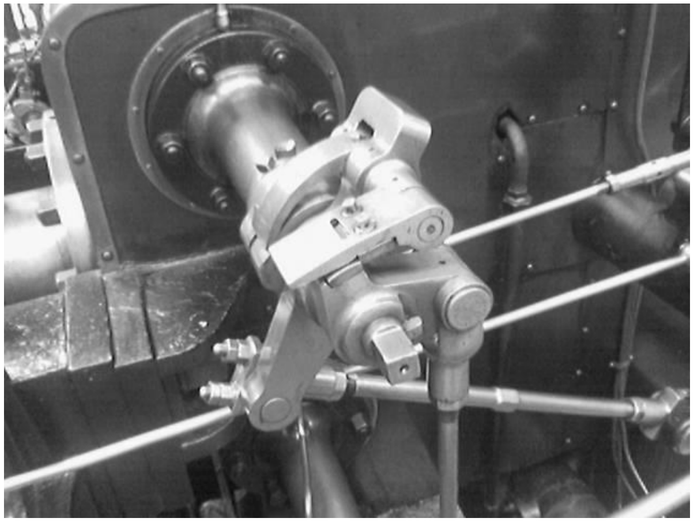


Targets:

1. Low error rate
2. Precisely locate edge points
3. “Single” edges

Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

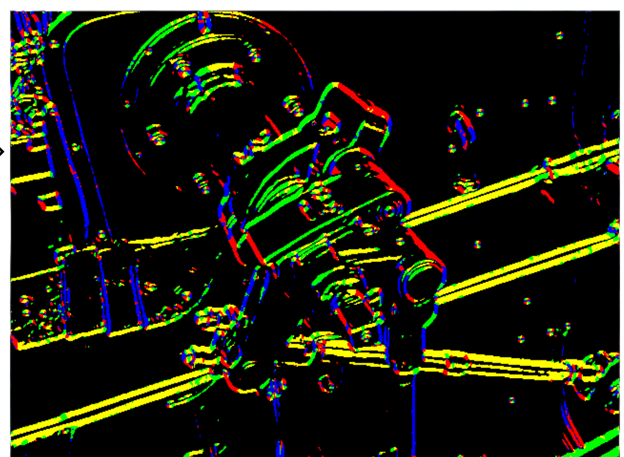
Derivative of the Gaussian



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

$$M(x,y) = \sqrt{\left(\frac{\partial f_s}{\partial x}\right)^2 + \left(\frac{\partial f_s}{\partial y}\right)^2}$$

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



0°

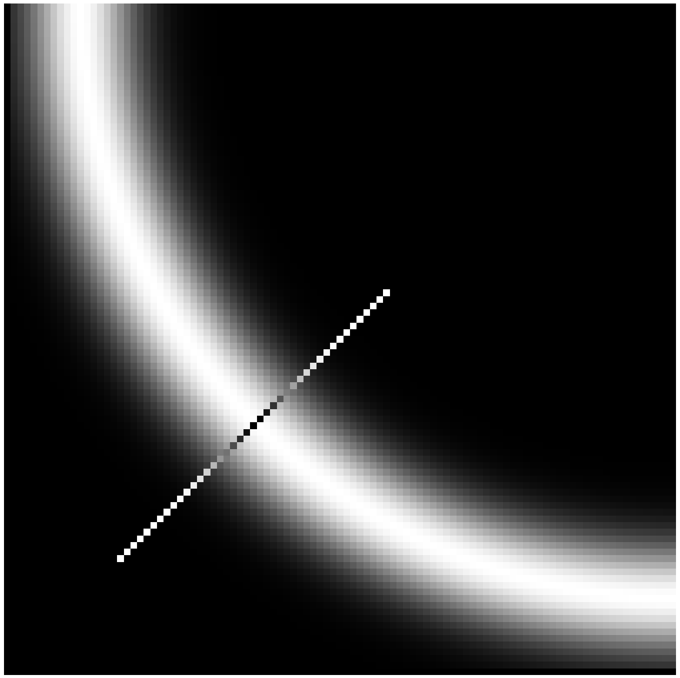
45°

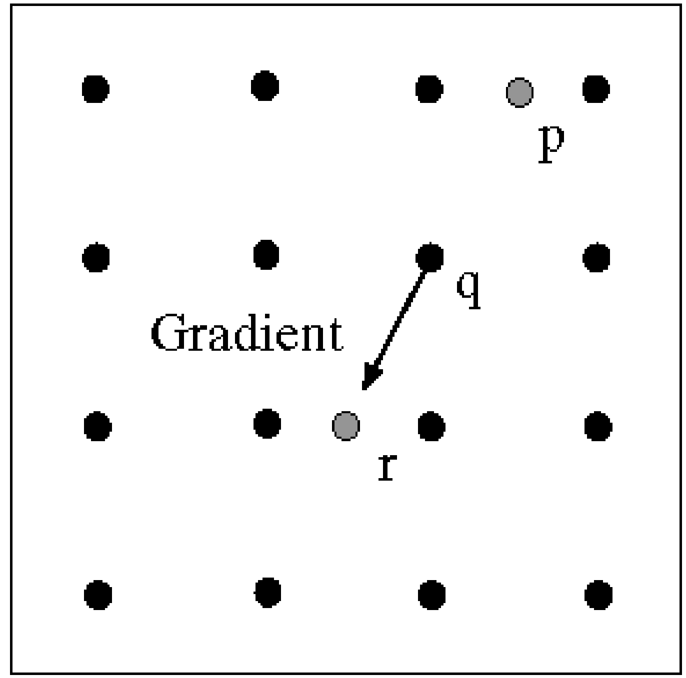
90°

135°

Thick edges!

Non-Maxima Suppression (1)





Check if pixel is local maximum along gradient direction

- Requires checking interpolated pixels p and r
- Canny : Approximation!

Non-Maxima Suppression (2)

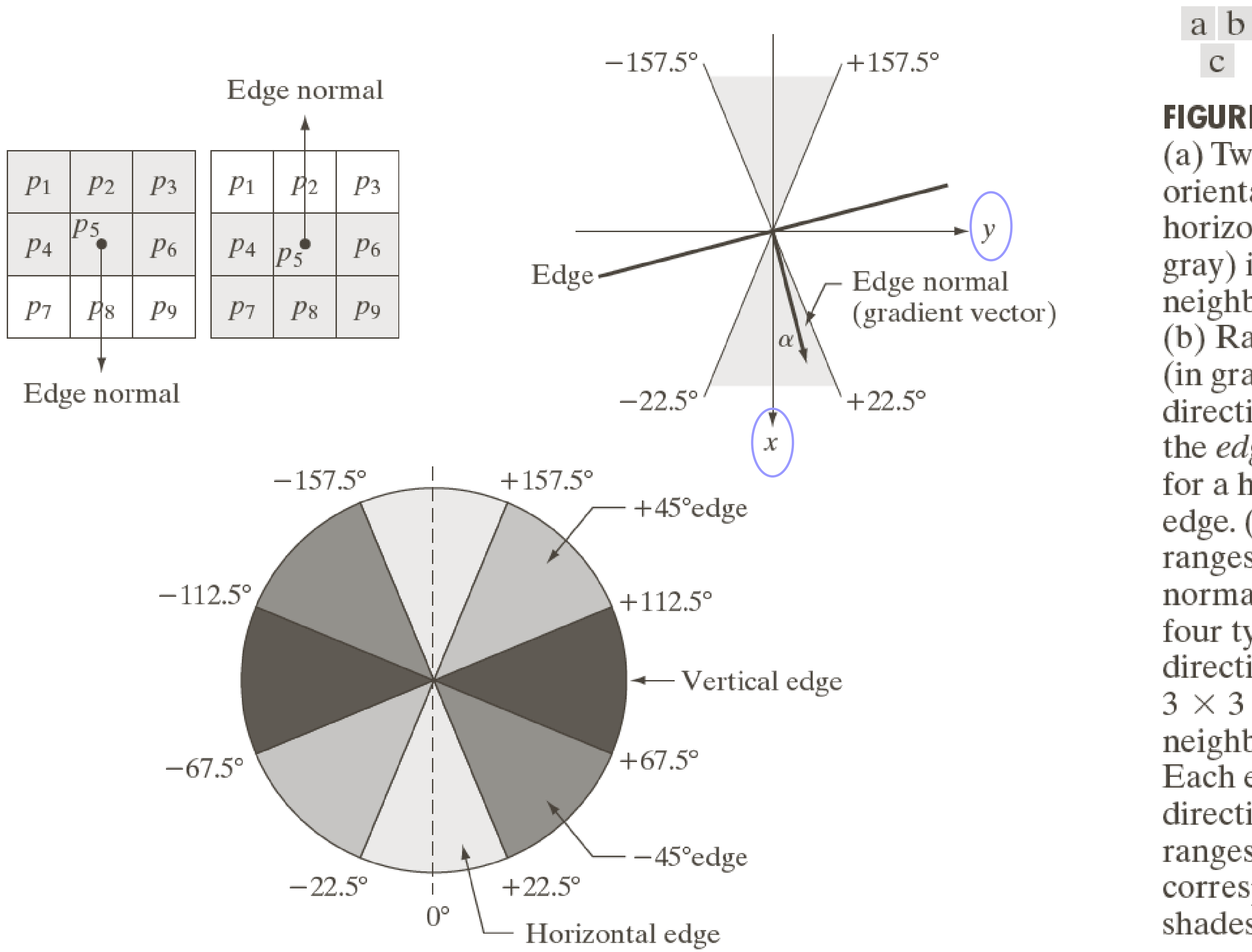
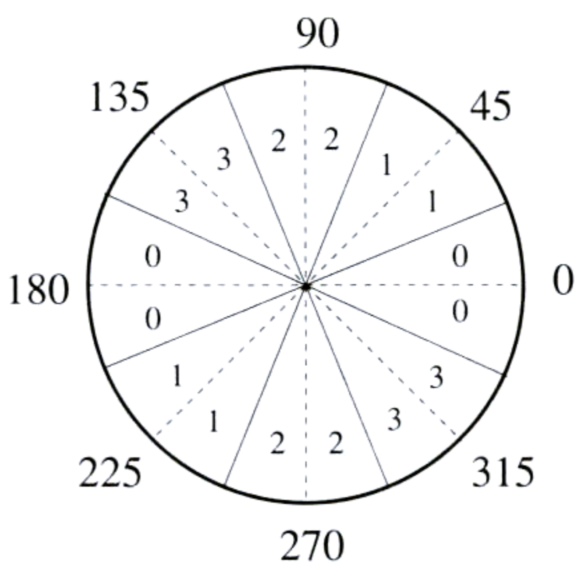


FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3 × 3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3 × 3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Non-Maxima Suppression (3)

- Assign angle of Gradient α(x,y) to one of the 4 sectors
- Check the 3x3 region of each M(x,y)
- If the value at the center is not greater than the 2 values along the gradient, then M(x,y) is set to 0



0

M_1	M	M_2

1

		M_2
	M	
M_1		

2

	M_1	
	M	
	M_2	

3

M_1		
	M	
		M_2

Hysteresis Tresholding

T_h=2 (M≥2)

0	0	0	0	0	0	0	3
0	0	0	0	1	1	3	0
0	0	2	1	1	0	0	0
0	0	3	0	0	0	0	0
0	3	2	0	0	0	0	0
0	1	0	0	0	1	0	1
0	3	0	0	1	0	1	0

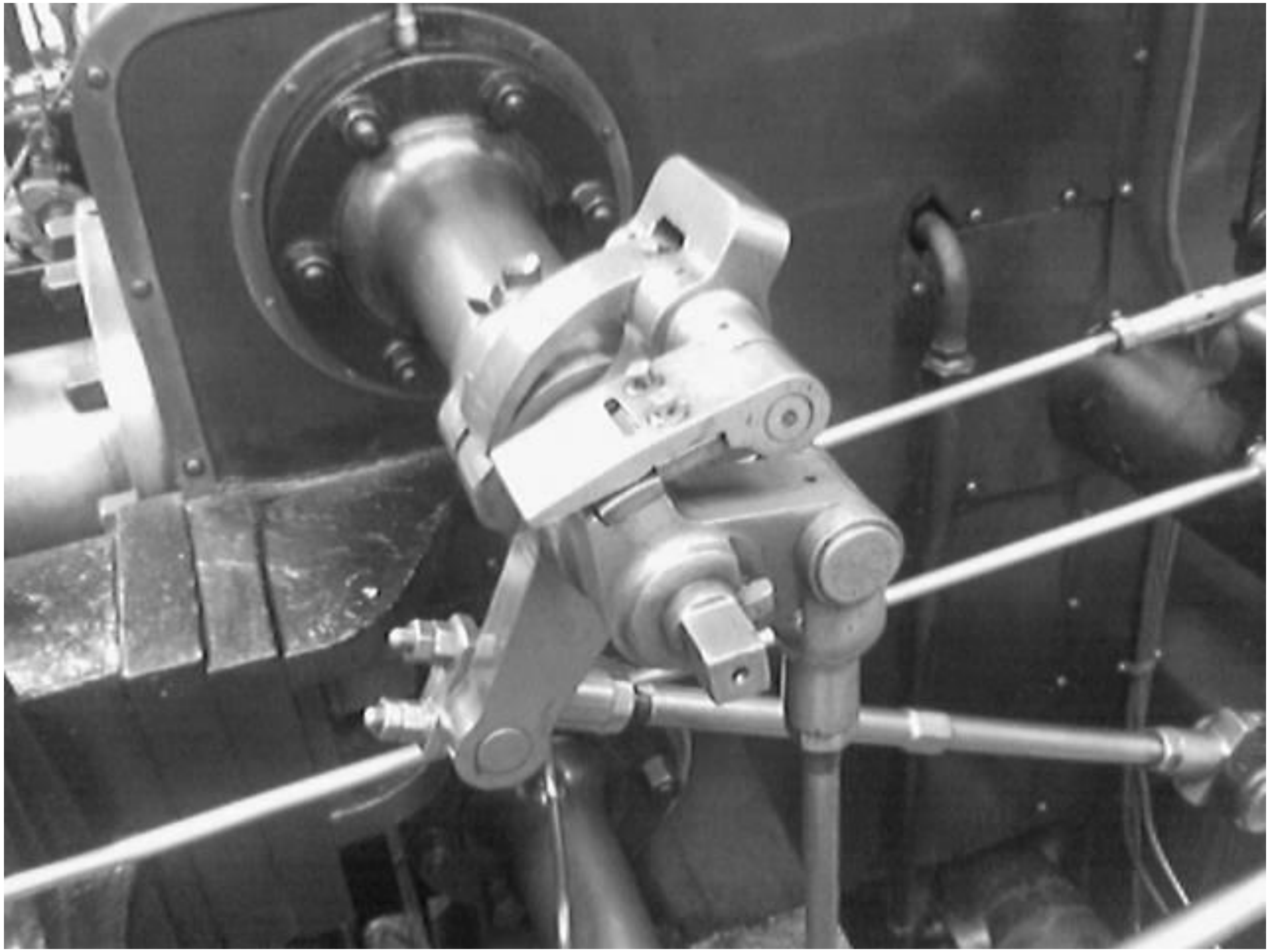
gaps filled from T_l

T_l=1 (1≤M<2)

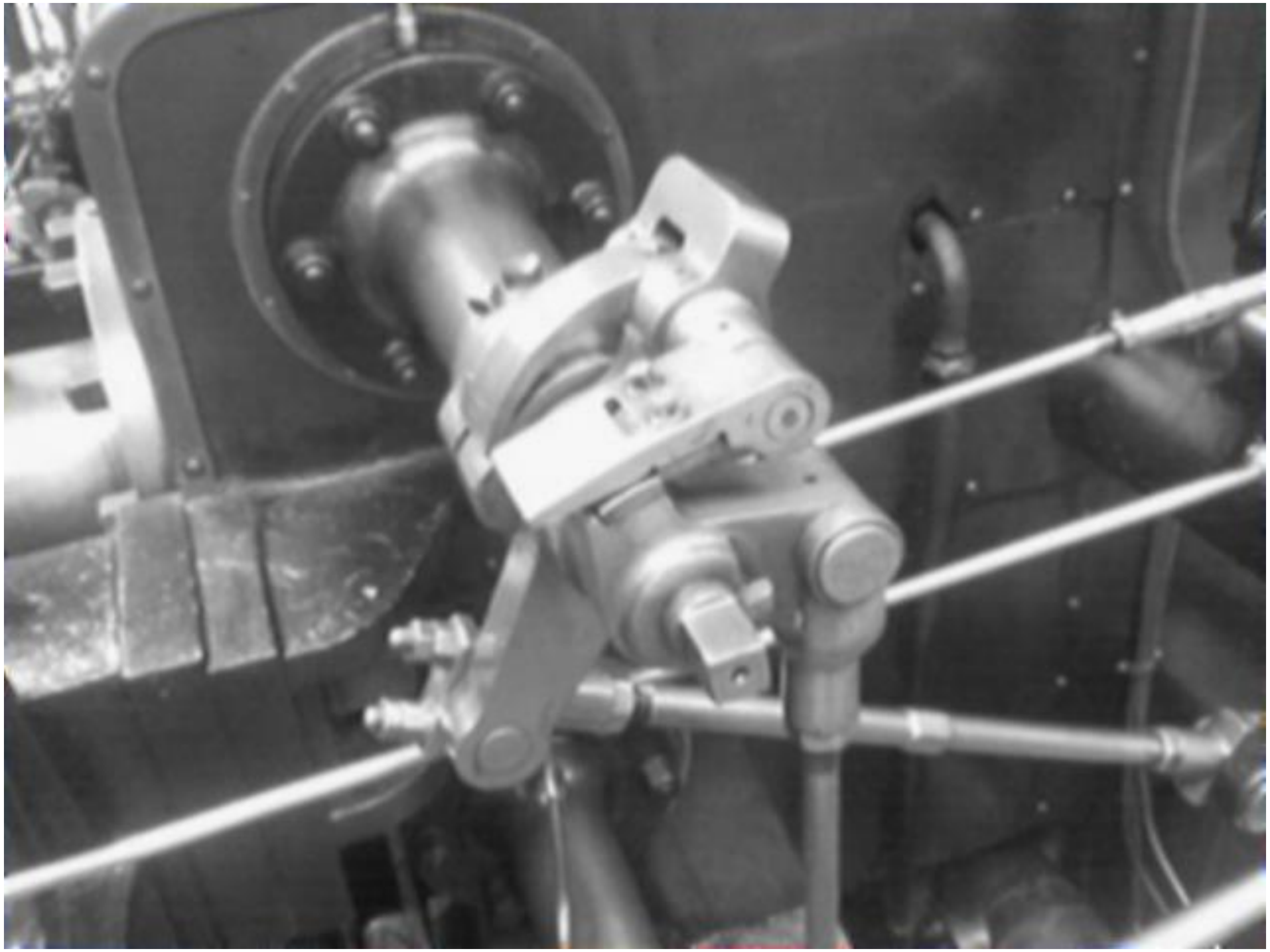
0	0	0	0	0	0	0	3
0	0	0	0	1	1	3	0
0	0	2	1	1	0	0	0
0	0	3	0	0	0	0	0
0	3	2	0	0	0	0	0
0	1	0	0	0	1	0	1
0	3	0	0	1	0	1	0

Canny Edge Detector: Procedure

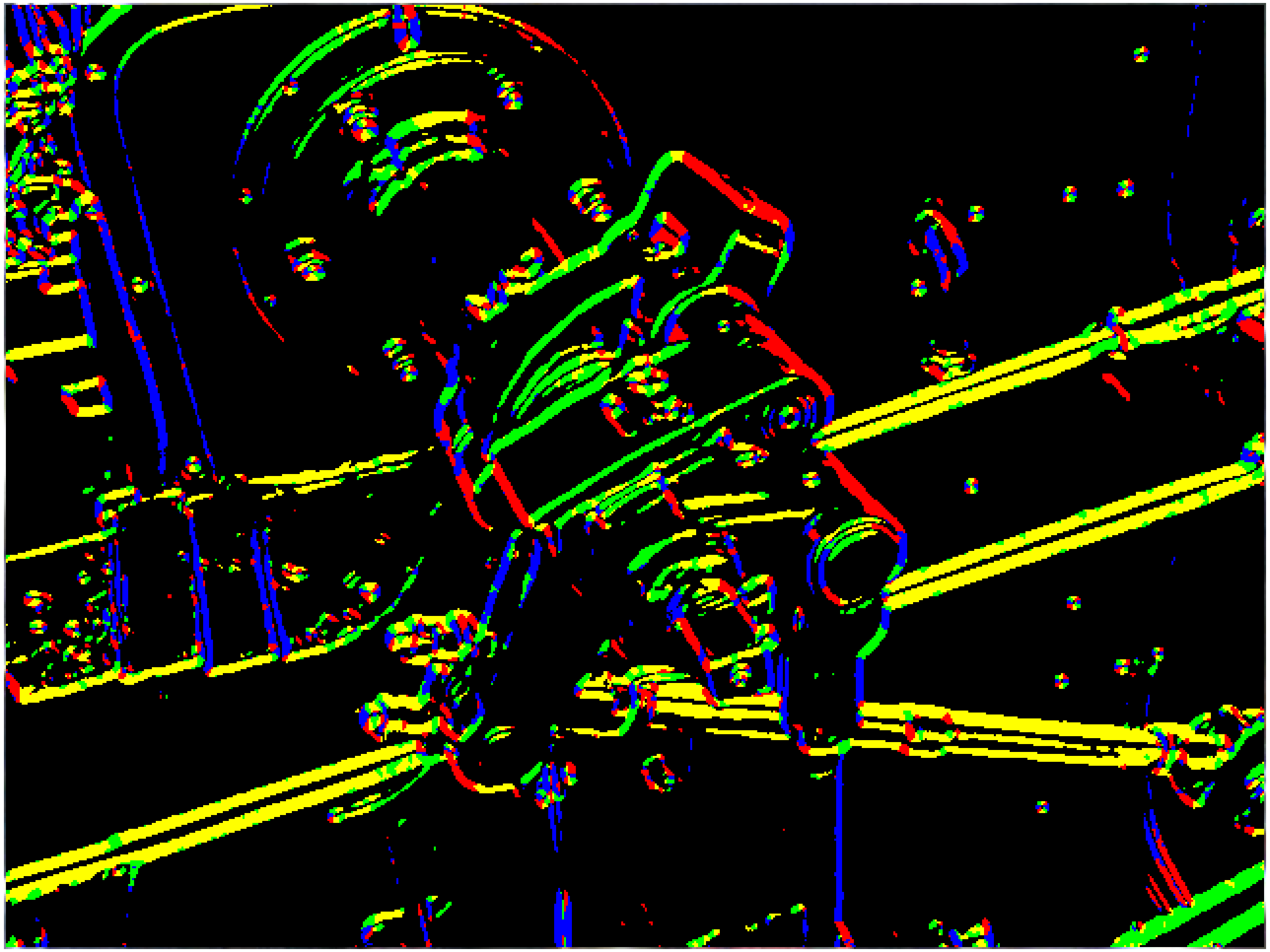
1. Smoothing with a Gaussian filter
2. Compute gradient (module and direction)
3. Quantize the gradient angles
4. Non maxima suppression
5. Thresholding with double threshold



1. Original image

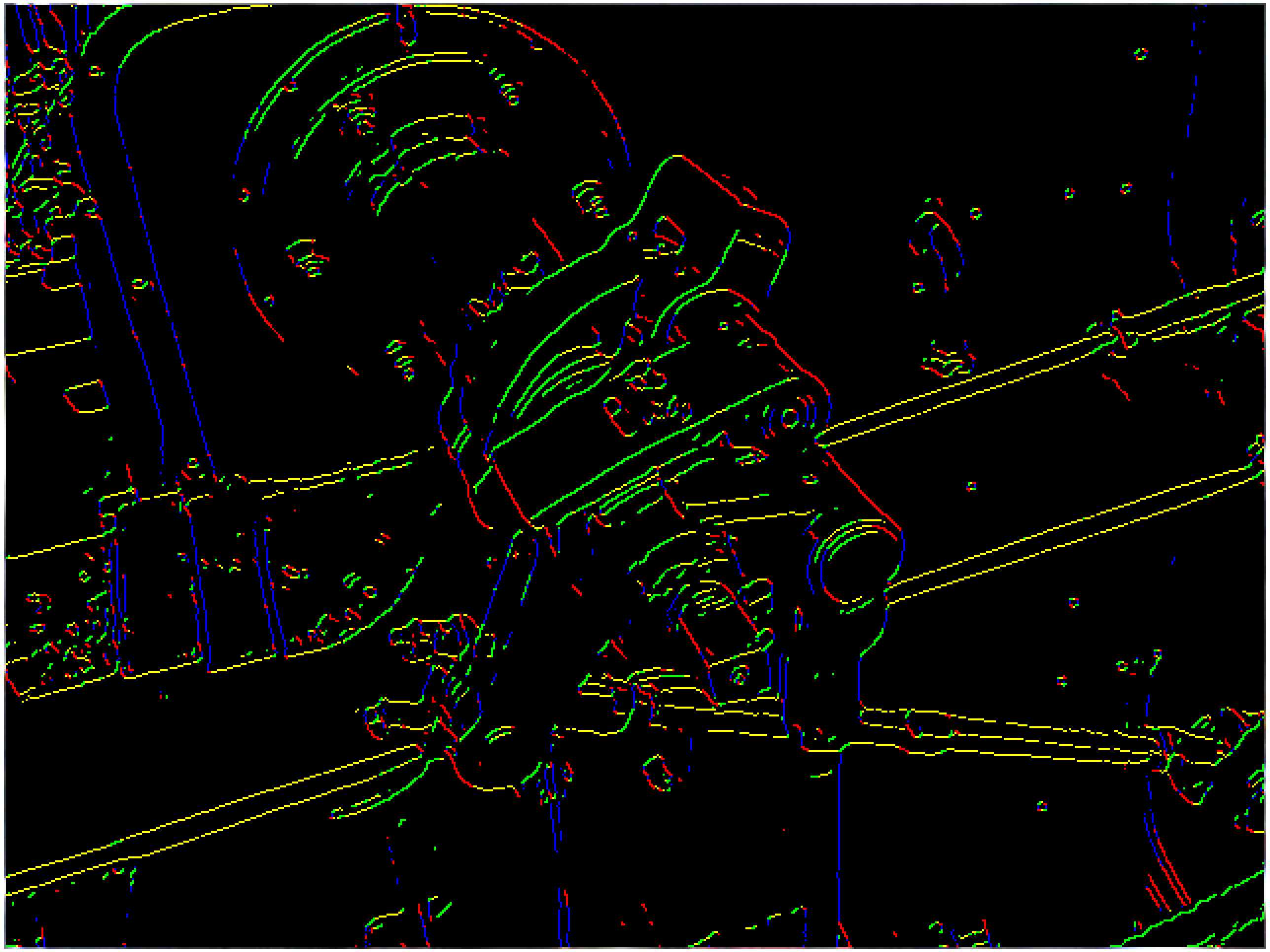


2. Gaussian smoothing

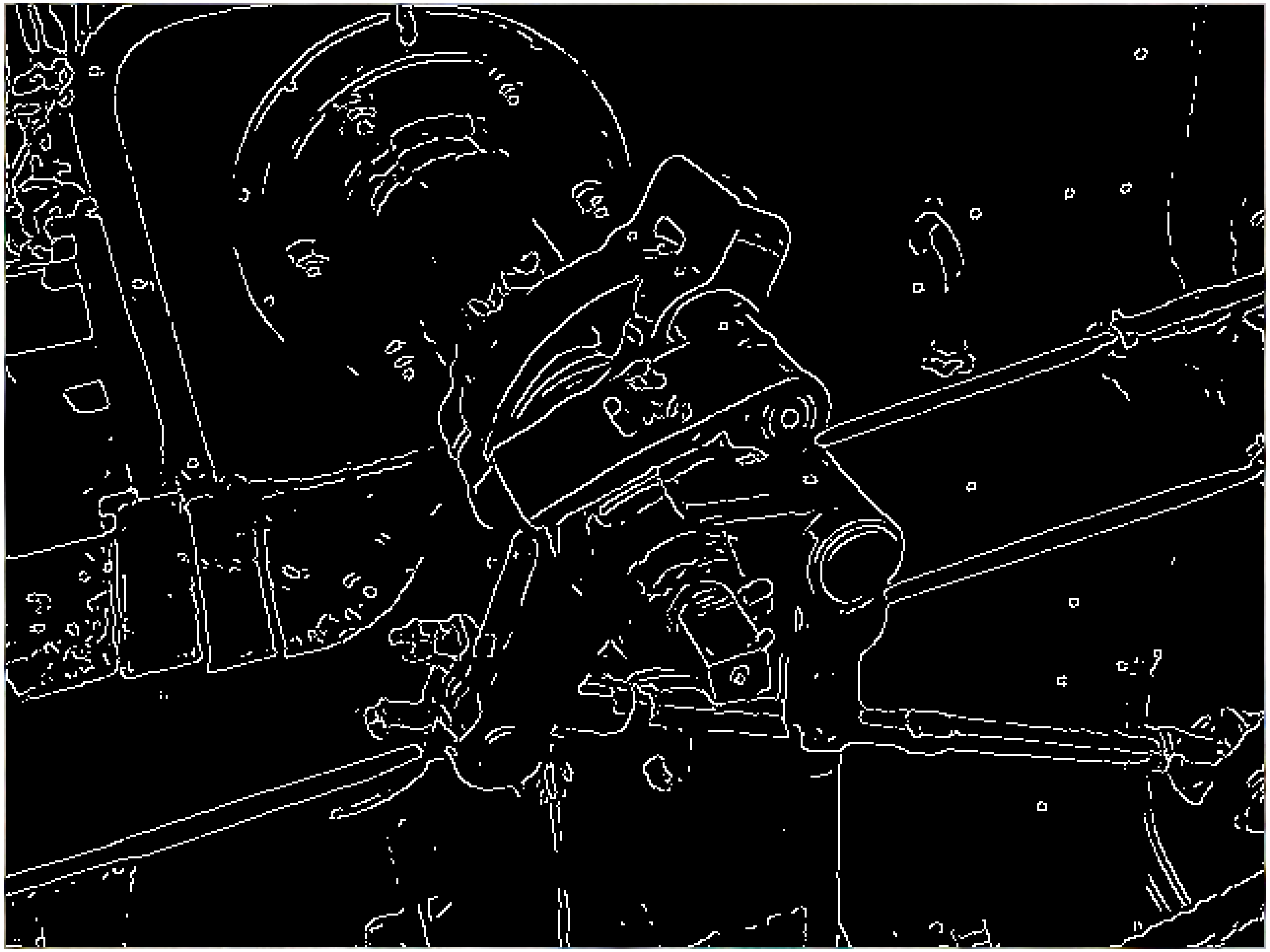


0°
45°
90°
135°

3. Gradient computed with Sobel's mask



4. Non maxima suppression



5. Output of the Canny edge detector

Effect of σ (Gaussian Kernel Size)

Original

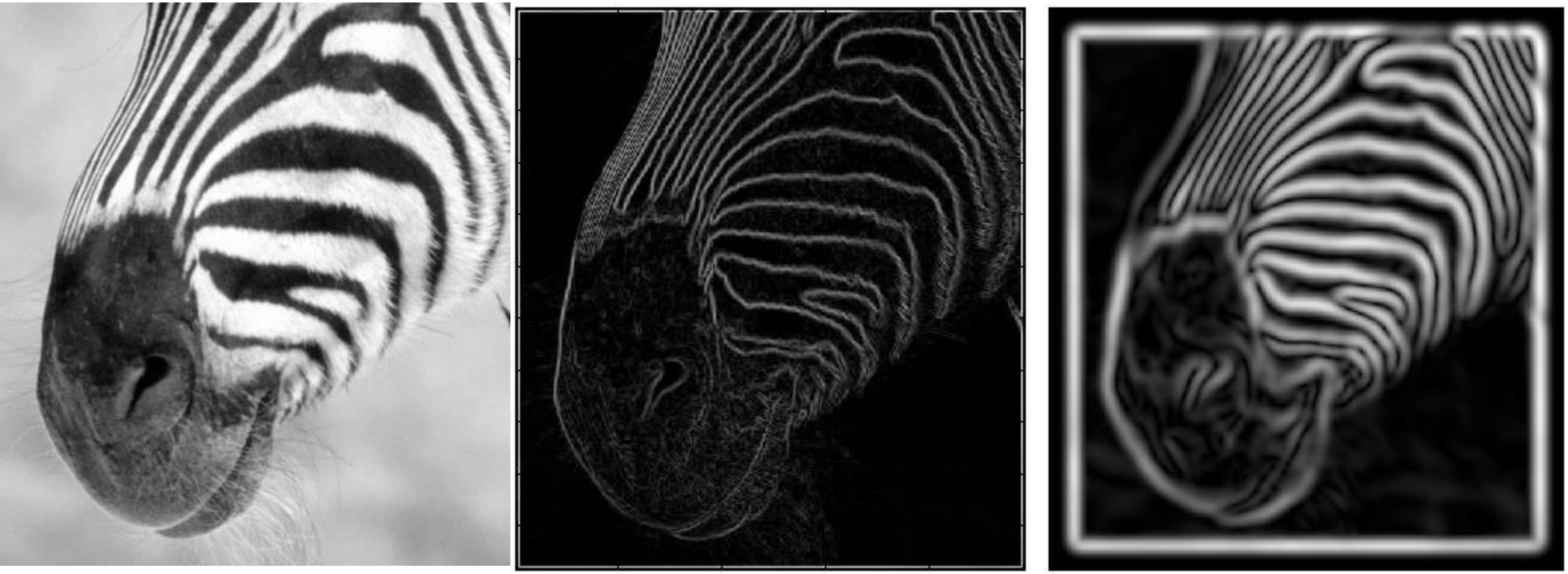
Canny with $\sigma = 1$

Canny with $\sigma = 2$

The choice of σ depends on desired behavior

- large σ detects large scale edges
- small σ detects fine features

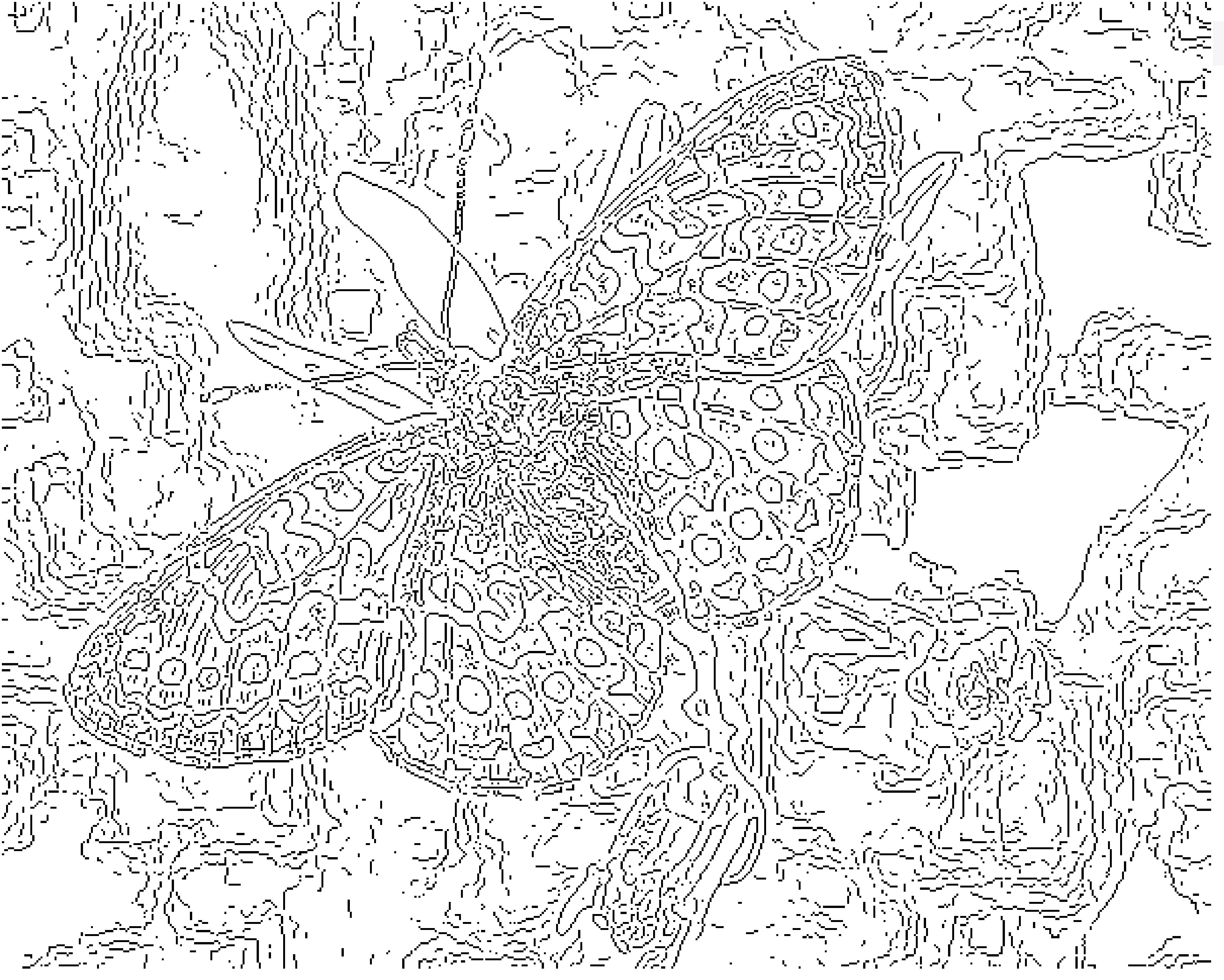
σ : Scale



- Smoothing
- Eliminates noise edges
- Makes edges smoother
- Removes fine details

(Forsyth & Ponce)

Example image

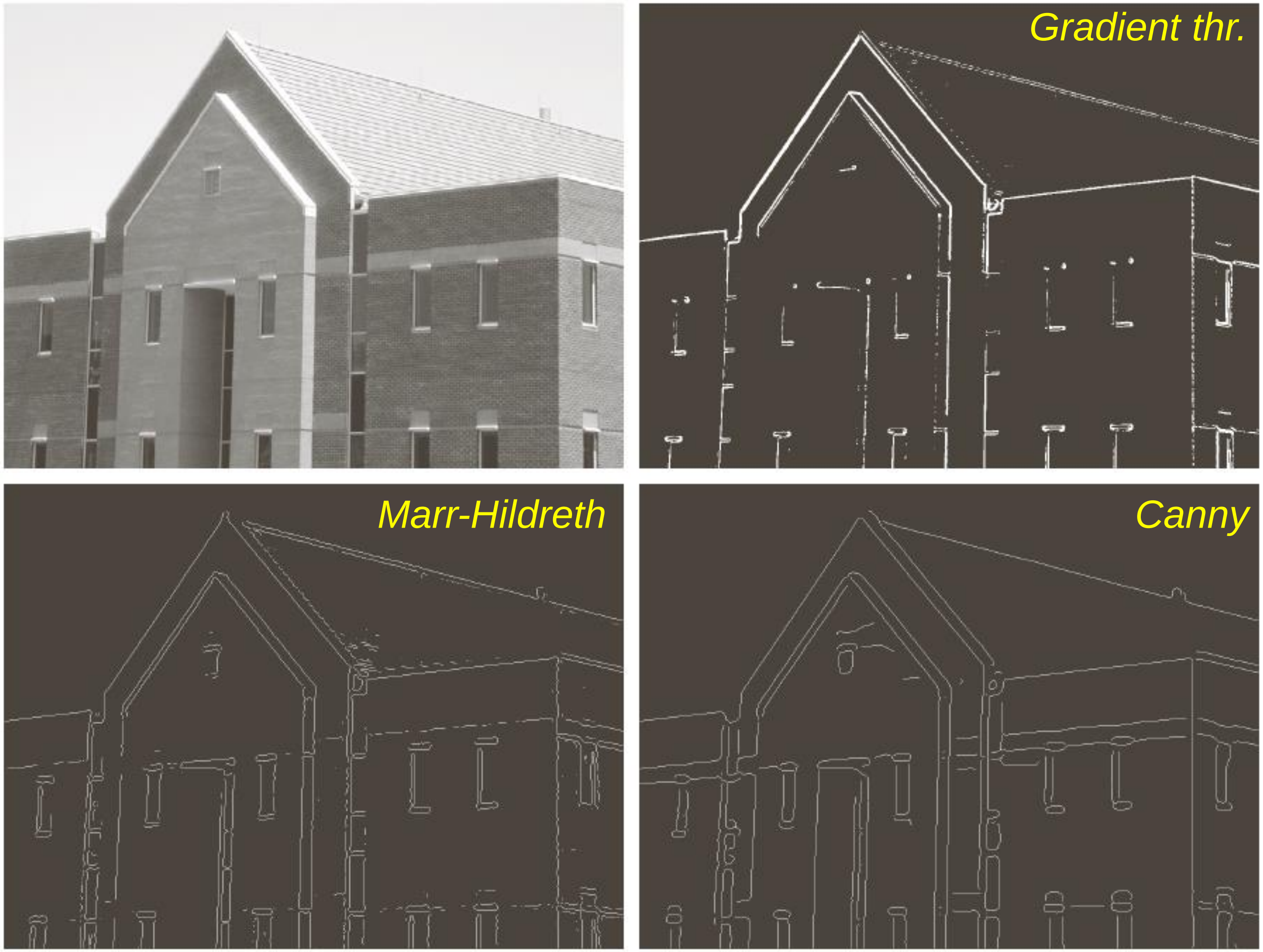


Fine scale (small σ); high thresholds

Coarse scale (large σ); high thresholds

Coarse scale (large σ); low thresholds

Gradient vs Canny vs Marr-Hildreth

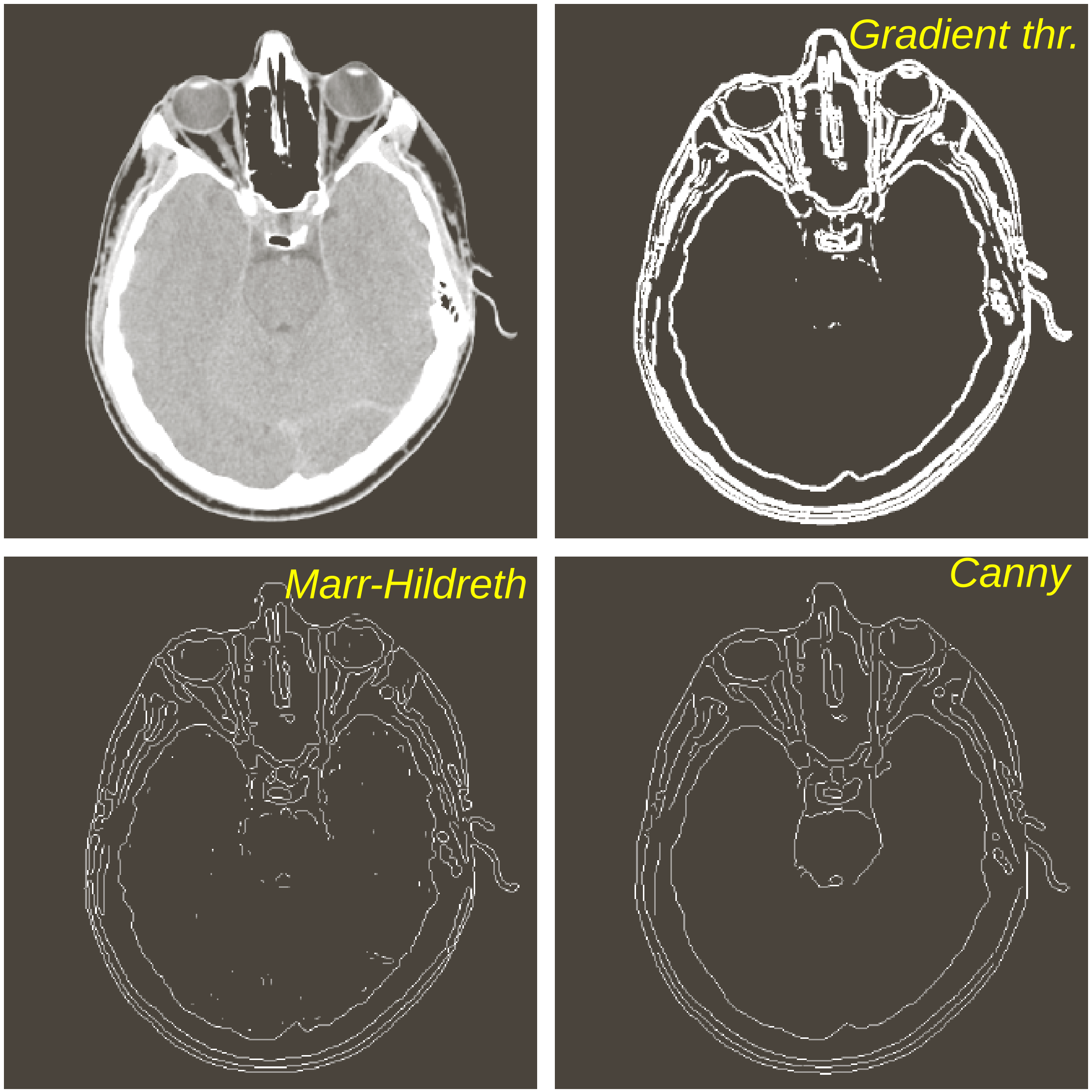


a	b
c	d

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.

Matlab example

Gradient vs Canny vs Marr-Hildreth

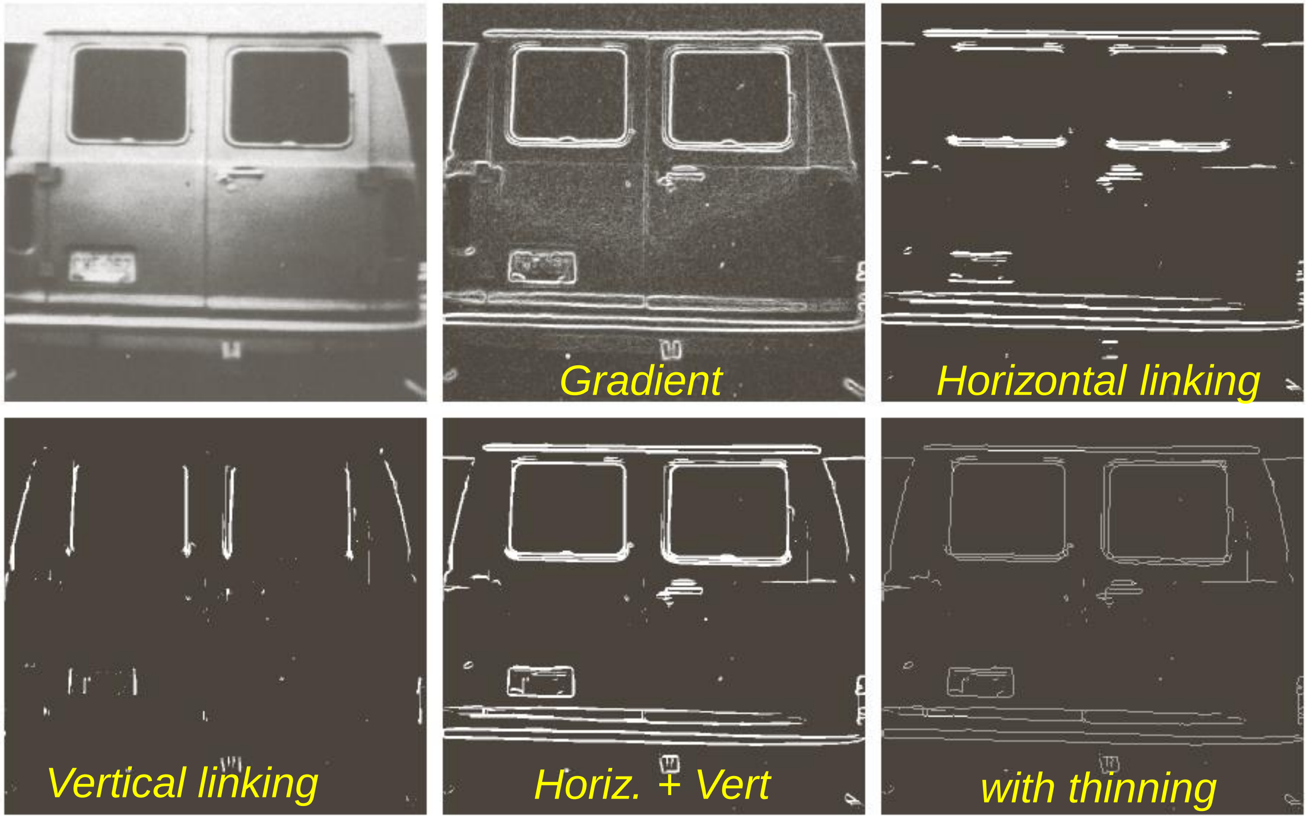


a b
c d

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.)

Edge Connection

Not part of the course



a b c
d e f

FIGURE 10.27 (a) A 534 × 566 image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)