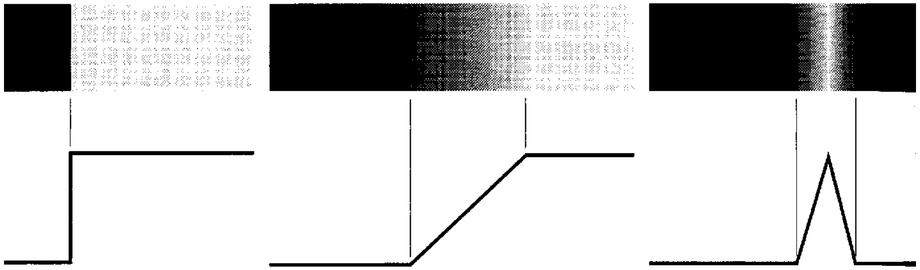
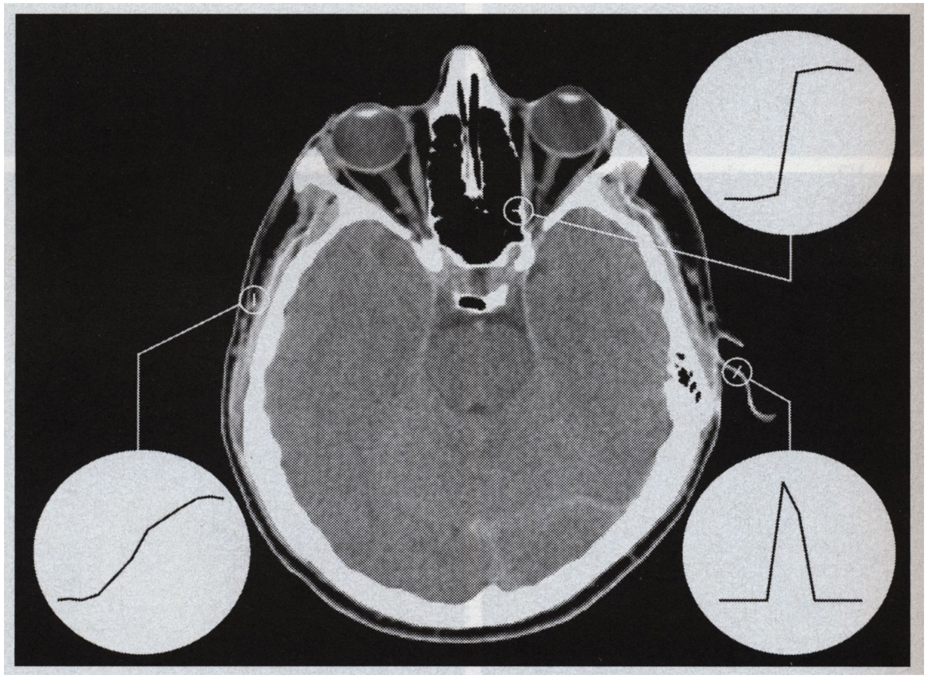
EDGE DETECTION AND SEGMENTATION OF IMAGES, I

- Ramp, roof and step edges
- First and second derivative of intensity profile
- Edge detection
- The Marr-Hildreth edge detector
- The Canny edge detector
- The image gradient and its properties
- The Canny edge detector
- The Canny and Marr-Hildreth edge detectors
- (The Marr-Hildreth edge detector – DoG)
- (Edge detection using gradient operators)
- (Edge linking)

Ramp, roof and step edges

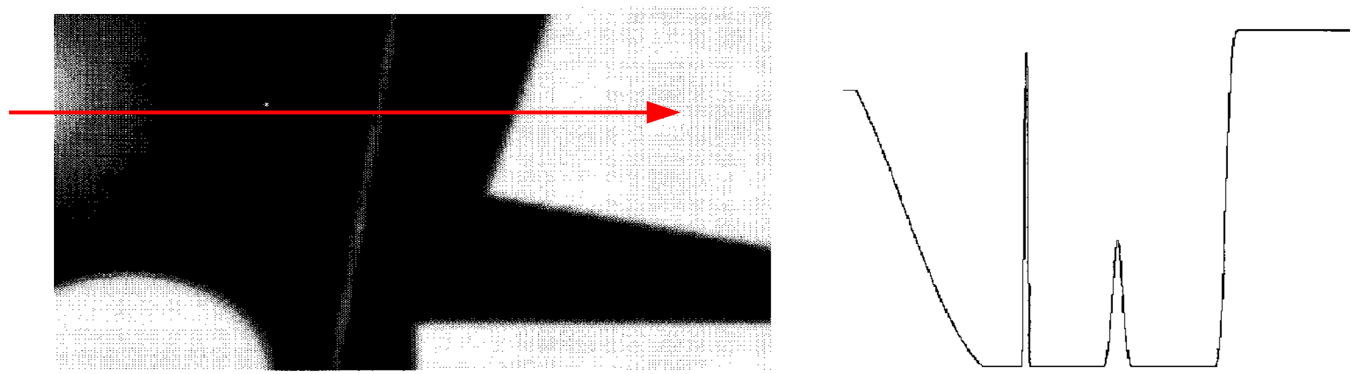


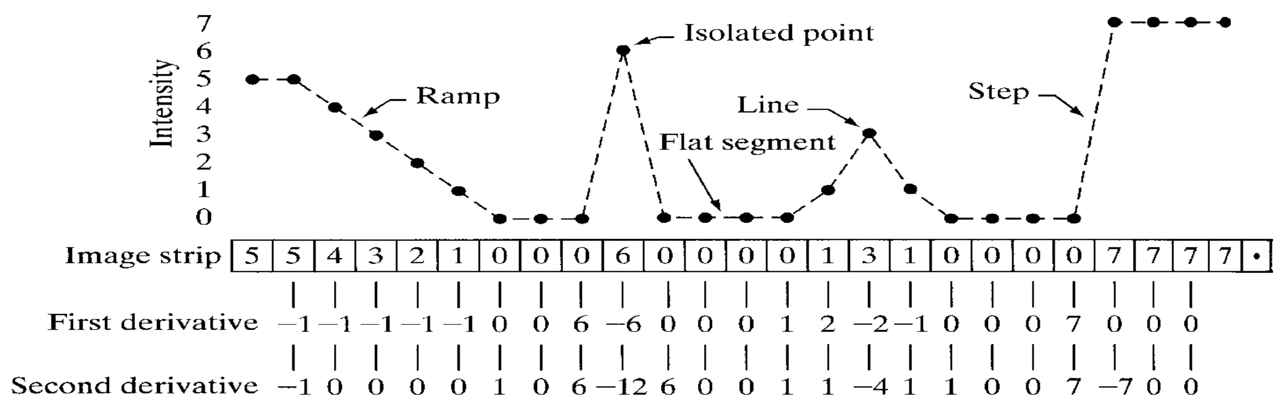


(Gonzales, Woods)

First and second derivative of intensity profile

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \qquad \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

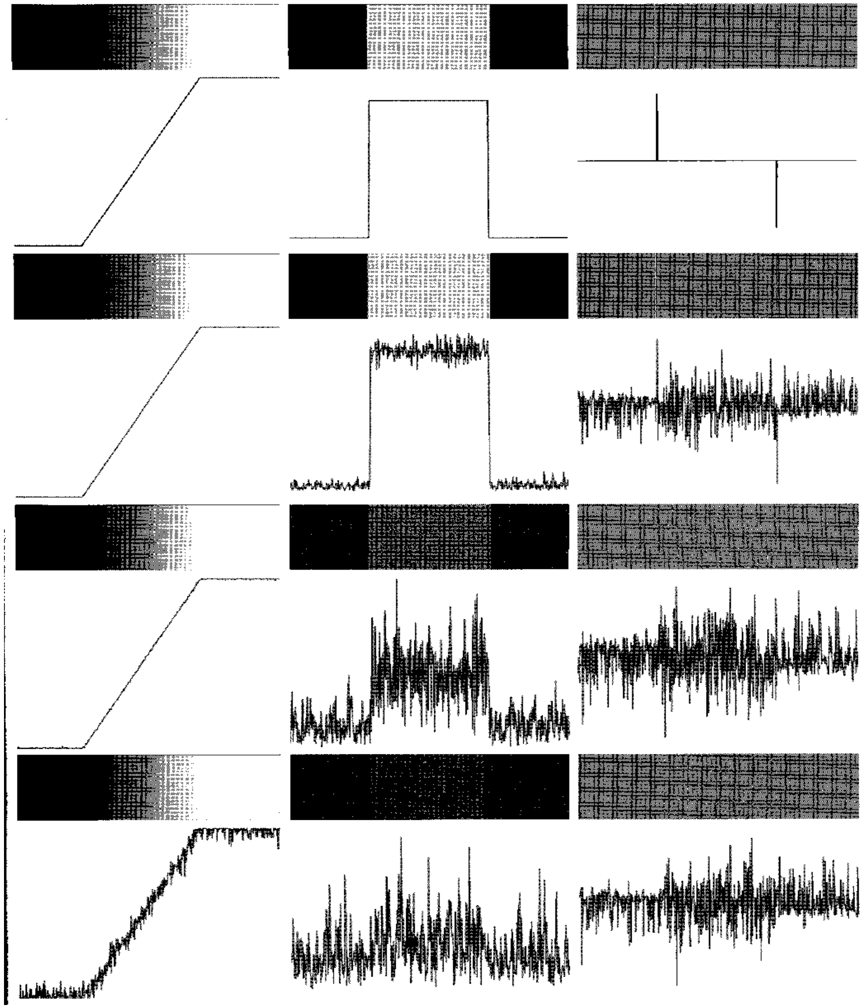




(Gonzales, Woods)

First and second derivative of intensity profile

- 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
- First-derivative images and intensity profiles
- Second-derivative images and intensity profiles



(Gonzales, Woods)

Edge detection

- Edge detection consists from three fundamental steps:

1. Image smoothing for noise reduction (preprocessing; or, smoothing feature is contained in the next step)
2. Detection of edge points (a local operation that extracts from an image all points that are potential candidates to become edge points)
3. Edge localization (selects from the candidate edge points only the points that are true members of the set of points comprising an edge)

The Marr-Hildreth edge detector

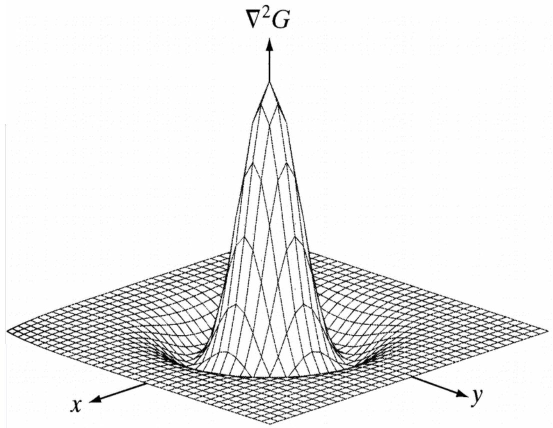
- Two-D Gaussian function

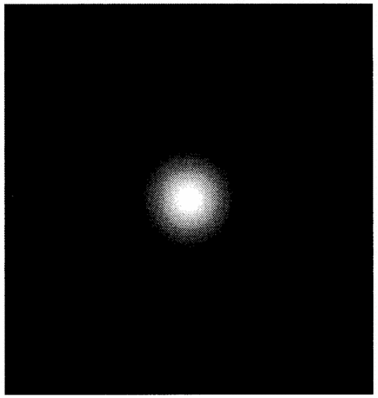
$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

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

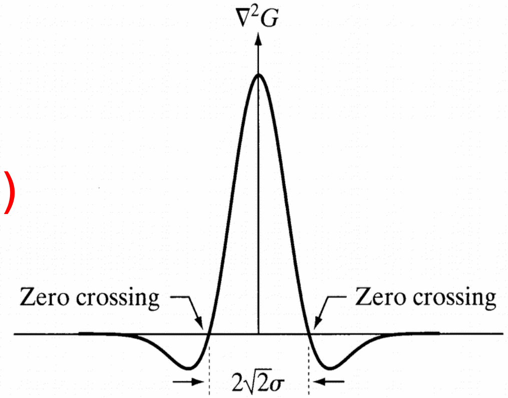
$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$





- The Laplacian of a Gaussian (LoG)

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

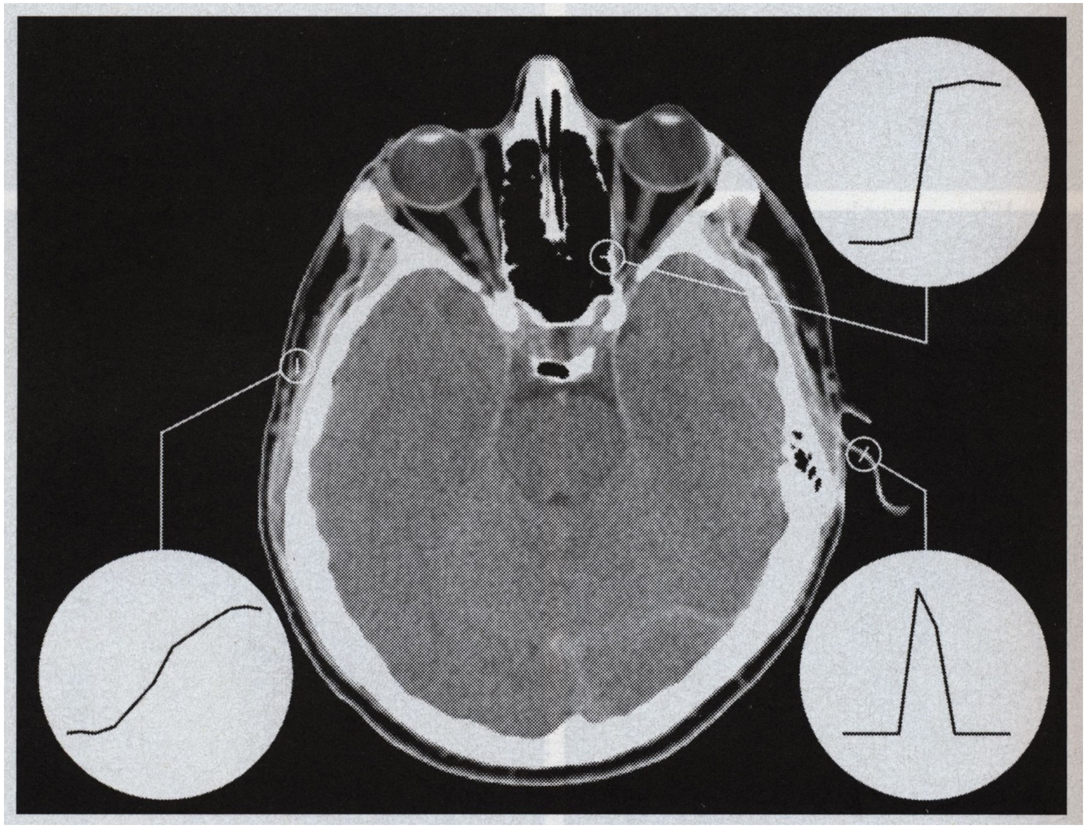


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

(Gonzales, Woods)

The Marr-Hildreth edge detector

- Exercises 2:
  - Detecting contours of human organs in CT images using the Marr-Hildreth edge detector



(Gonzales, Woods)

The Marr-Hildreth edge detector

- The Marr-Hildreth algorithm consists of convolving the LoG filter with an input image, *f(x,y)*, and then finding the zero crossings of *g(x,y)* to determine the locations of edges in *f(x,y)*
- Because these are linear processes, we can smooth the image first with a Gaussian filter and then compute the Laplacian of the result

1. Filter the input image with an *n x n* Gaussian lowpass filter obtained by sampling

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

2. Compute the Laplacian of the image resulting from step 1 using one of these masks

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

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

(Gonzales, Woods)

The Marr-Hildreth edge detector

- Zero crossings are the key feature of the Marr-Hildreth edge detection method
- An approach to find zero crossings at any pixel, *p(x,y)*, of the filtered image, *g(x,y)*:

1. Use a 3 x 3 neighborhood centered at *p*
2. The *signs* of at least two of opposing neighboring pixels of a pixel *p* considered must differ (there are four cases: left/right, up/down, and the two diagonals)
3. If the values of *g(x,y)* are being compared to a threshold, *T*, then besides the requirement of different signs of opposing neighbors, also the absolute value of their numerical differences must exceed the threshold *T*

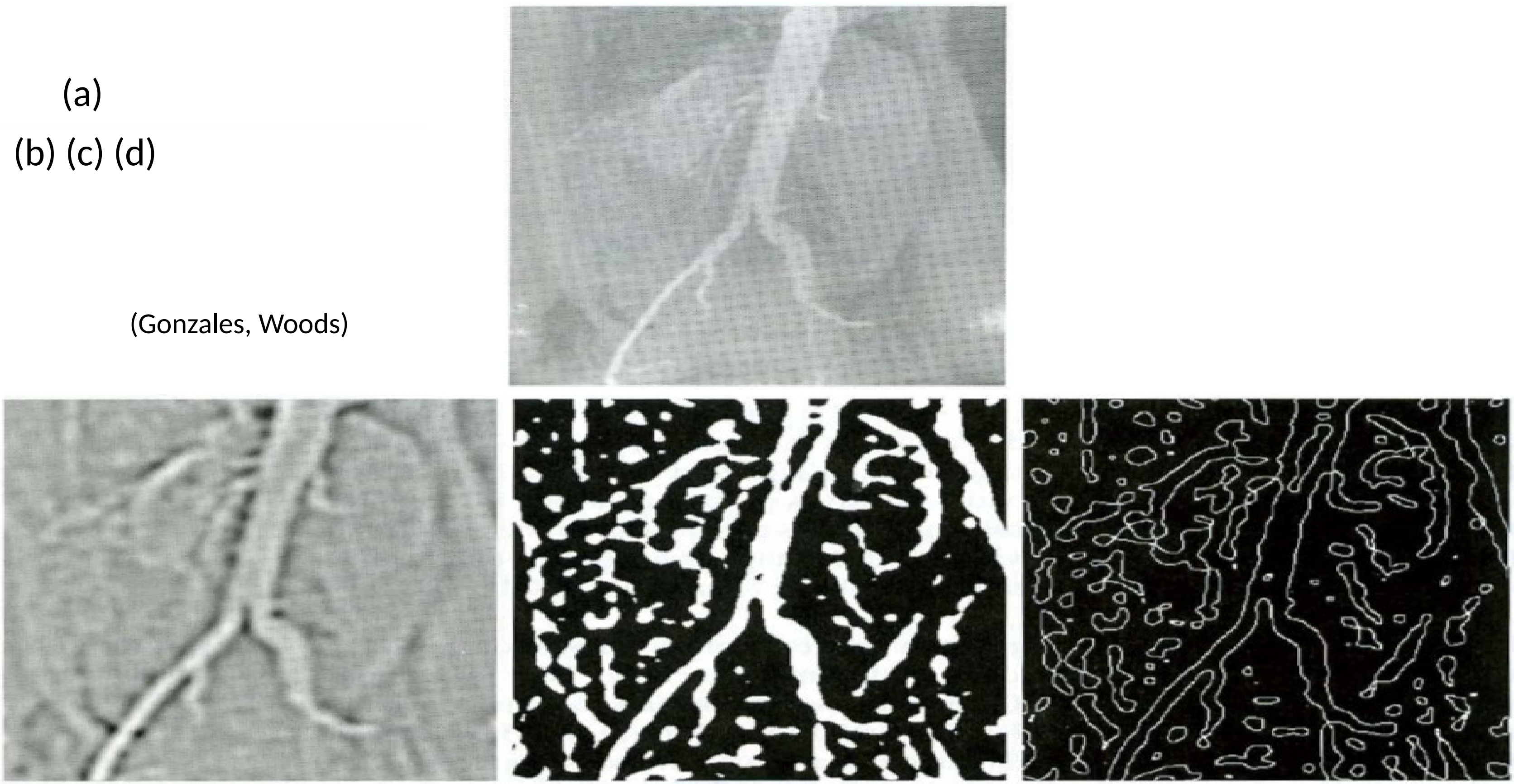
The Marr-Hildreth edge detector

- (a) Angiogram image (b) The LoG image (σ = 5, n = 27) (c) Thresholded LoG (positive values set to white, negative values set to black) (d) Zero crossings

(a)

(b) (c) (d)

(Gonzales, Woods)

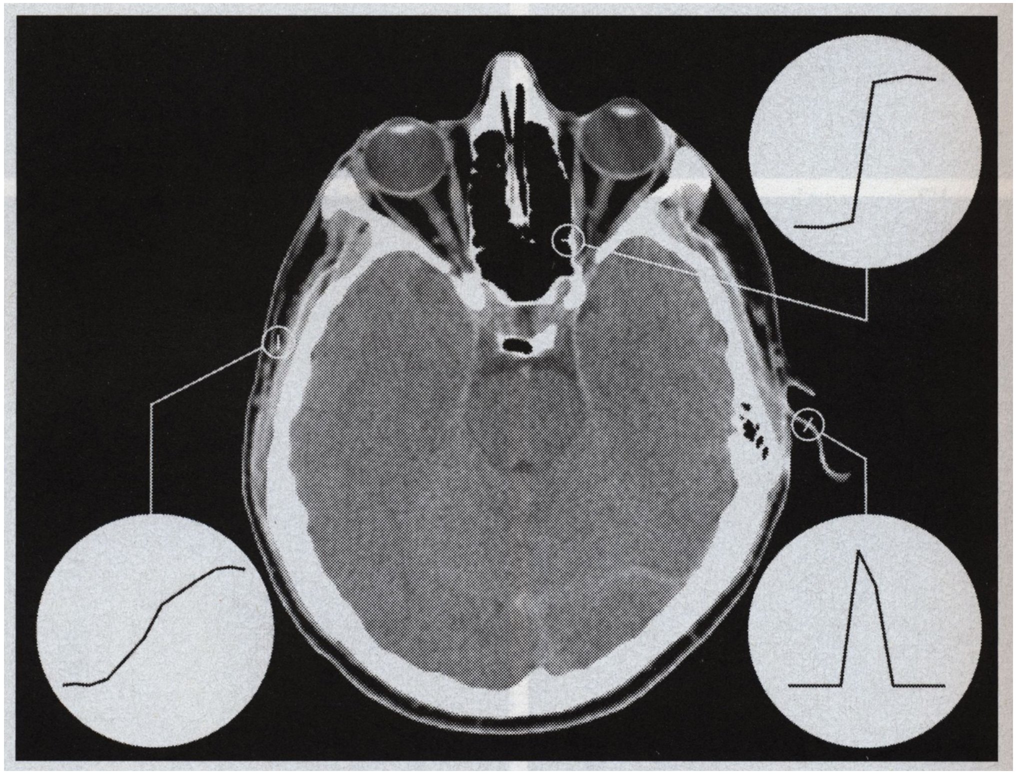


The Canny edge detector

- Basic objectives of Canny's approach
  1. Low error rate (all edges should be found – the edges detected must be as close as possible to the true edges)
  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 detector should return only one point for each true edge point – the number of local maxima around the true edge should be minimum)
- Basic steps of the Canny edge detection algorithm
  1. Smooth the input image with a Gaussian filter
  2. Compute the gradient magnitude and angle images
  3. Apply nonmaxima suppression to the gradient magnitude image
  4. Use double (hysteresis) thresholding and connectivity analysis to detect and link edges

The Canny edge detector

- Exercises 2:
  - Detecting contours of human organs in CT images using the Canny edge detector



(Gonzales, Woods)

The Canny edge detector

- A good approximation to the optimal step detector is the first derivative of a Gaussian

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

- 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, *fs(x,y)*, by convolving *G* and *f*

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

- Compute the gradient magnitude and direction (angle)

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

with $gx = \delta f / \delta x$, $gy = \delta f / \delta y$, and *gx* and *gy* are any of filter mask pairs to compute the gradient

The image gradient and its properties

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

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

$g_x = (0 + 0 + 0) - (0 + 1 + 1) = -2$

$g_y = (1 + 1 + 0) - (0 + 0 + 0) = 2$

(to compute g_x and g_y use 3 x 3 mask at *x*, *y*)

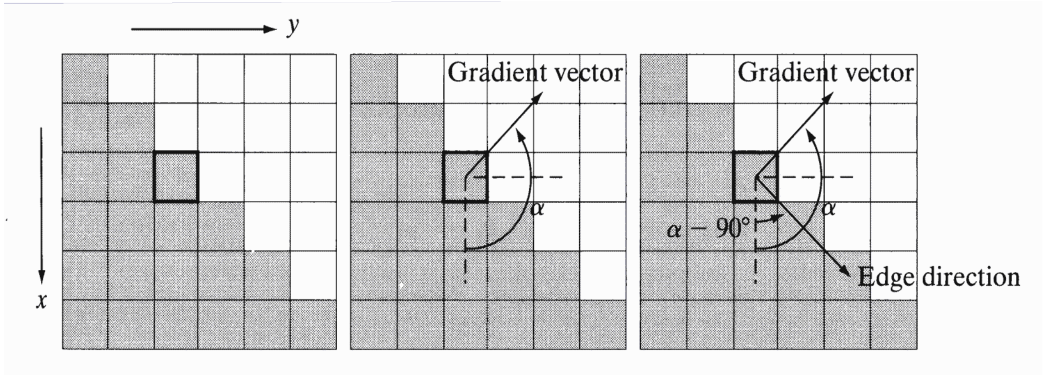
- The magnitude (length) of the vector, *M(x,y)*, is the value of the rate of change in the direction of the gradient vector

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

- The direction of the gradient vector, *α*, is given by the angle measured with respect to the *x*-axis

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

$\alpha \leftarrow \alpha(x,y) + 180° = 135°$



($\delta f / \delta x = -2$, $\delta f / \delta y = 2$, $M(x,y) = 2.\mathrm{sqrt}(2)$, $\alpha(x,y) = -45°$, $\alpha - 90° = 45°$)

The Canny edge detector

- Various masks to compute the gradient ($gx = \delta f / \delta x$, $gy = \delta f / \delta y$) (intensity levels)

−1	0
0	1

0	−1
1	0

Roberts

−1
1

−1	1

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

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

−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

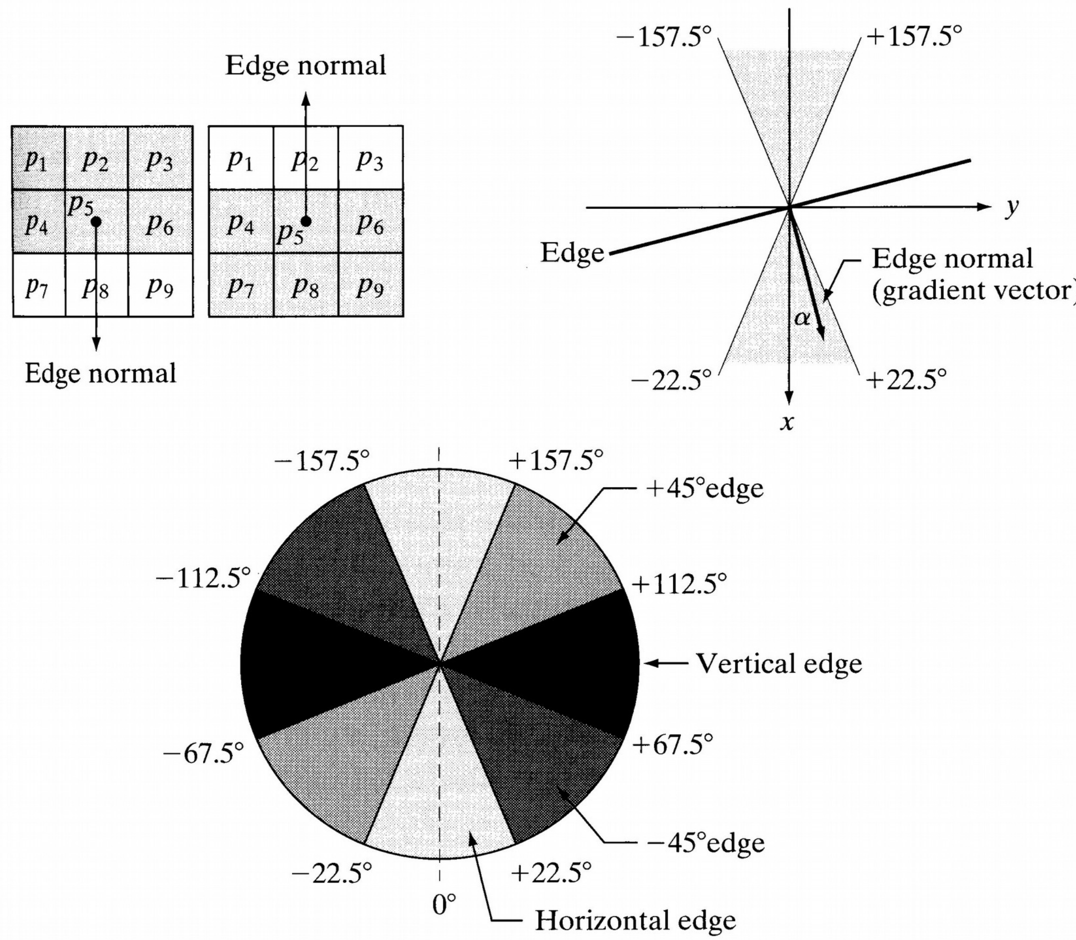
0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

The Canny edge detector

- Thin the ridges using nonmaxima suppression scheme:
  1. Find the direction d_k that is closest to $\alpha(x,y)$ where d_1, d_2, d_3, and d_4 denote the four basic edge directions just discussed for a 3 x 3 region: horizontal, -45°, vertical, and + 45°
  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 $g_N(x,y) = M(x,y)$



(Gonzales, Woods)

The Canny edge detector

- Hysteresis thresholding

  - Use two thresholds, *TH*, *TL*: *TH* > *TL*, *TH* = *2(3).TL*

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

  - Eliminate from $g_{NL}(x,y)$ all the nonzero pixels from $g_{NH}(x,y)$ by letting:

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

  - The nonzero pixels in $g_{NH}(x,y)$ and $g_{NL}(x,y)$ are viewed as being “strong” and “week” edge pixels
  - All strong pixels in $g_{NH}(x,y)$ are assumed to be valid edge pixels. 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 pixels all the week pixels in $g_{NL}(x,y)$ that are connected to *p* using, *say, 8-connectivity*

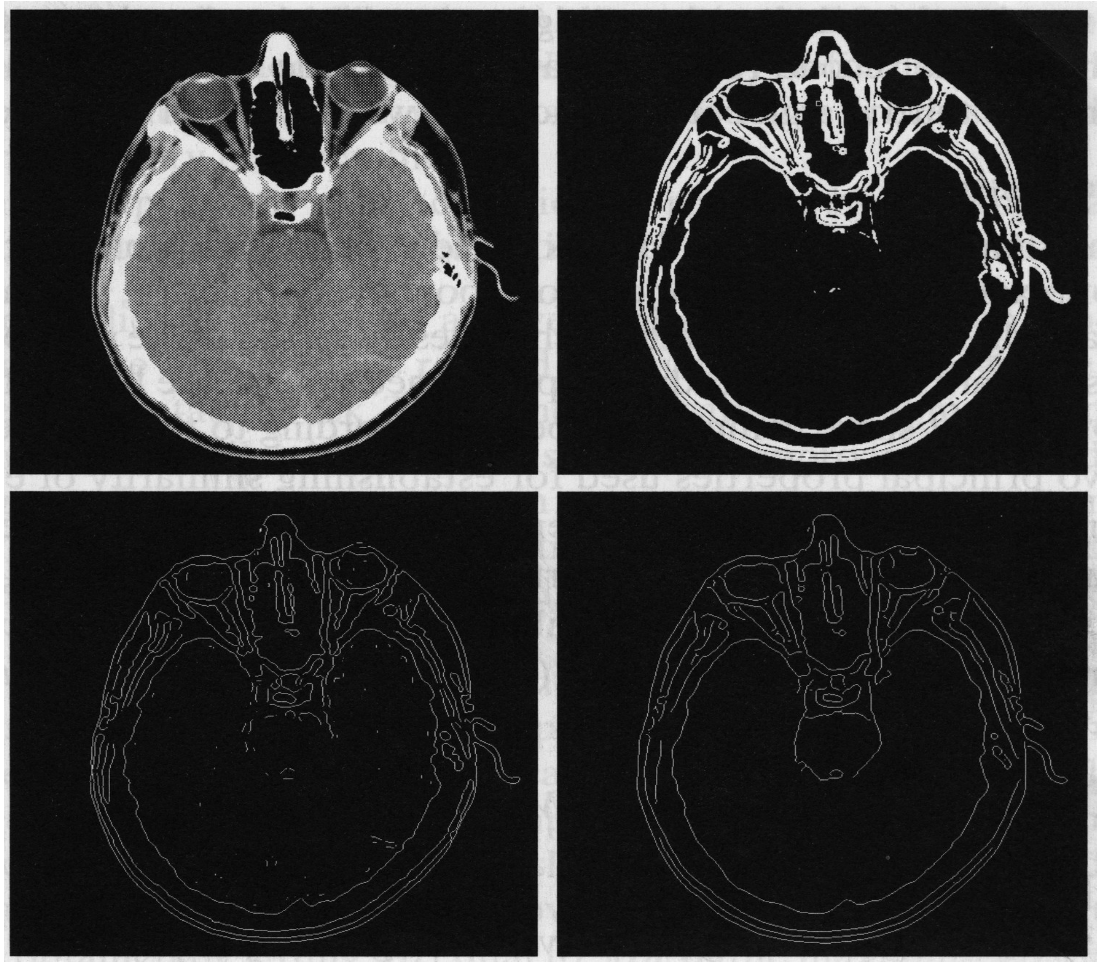
    (c) If all nonzero pixels in $g_{NH}(x,y)$ have been visited go to step d; else return to a

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

    (e) Append to $g_{NH}(x,y)$ all the nonzero pixels from $g_{NL}(x,y)$

The Canny edge detector

- Objective: to extract the edges of the outer contour of the brain (the gray region), the contour of the spinal region (directly behind the nose, toward the front of the brain), and the outer contour of the head
- Objective: to generate the thinnest, continuous contours possible, while eliminating edge details related to the gray content in the eyes and brain areas



(Gonzales, Woods)

The Canny edge detector

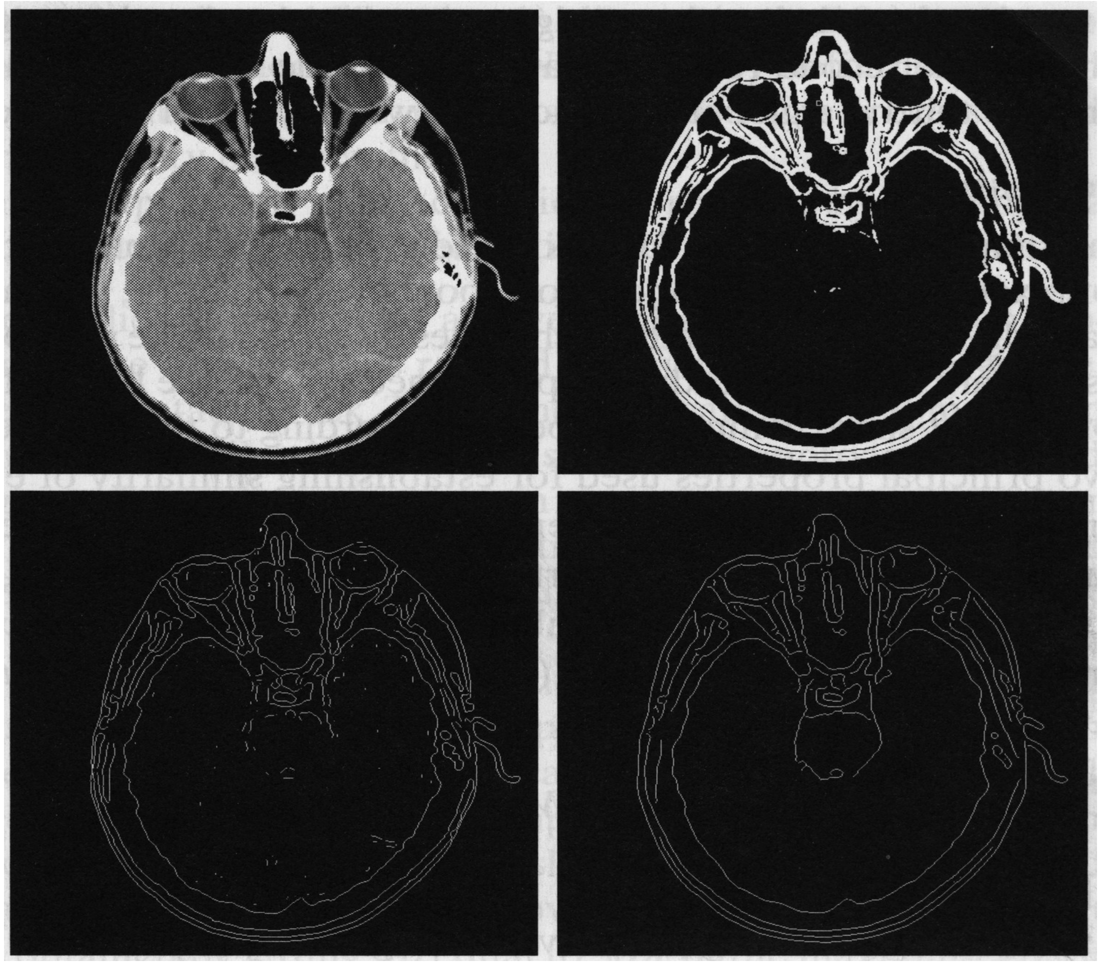
(a) (b)
(c) (d)

(a) Original head CT image of size 512 x 512 pixels

(b) Thresholded gradient of smoothed image (5 x 5 moving avearge)

(c) Image obtained using the Marr-Hildreth algorithm (threshold of 0.002, σ = 3, the mask of size 19 x 19 – the smallest odd integer greater that 6σ)

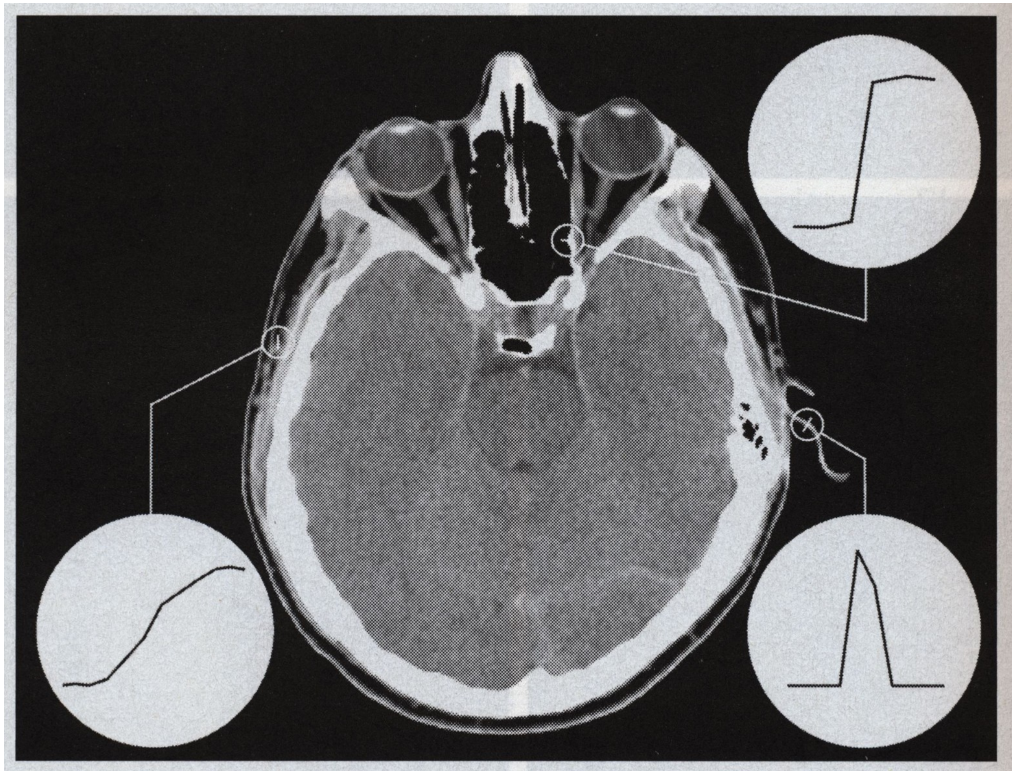
(d) Image obtained using the Canny algorithm (TL = 0.05, TH = 0.15, σ = 2, the mask of size 13 x 13 – the smallest odd integer greater that 6σ)



(Gonzales, Woods)

The Canny and Marr-Hildreth edge detectors

- Exercises 2:
  - Detecting contours of human organs in 3D CT images using the Canny edge detector (hint: link edges between image slices using 24-connectivity)
  - Detecting contours of human organs in 3D CT images using the Marr-Hildreth edge detector (hint: link edges between image slices using 24-connectivity)



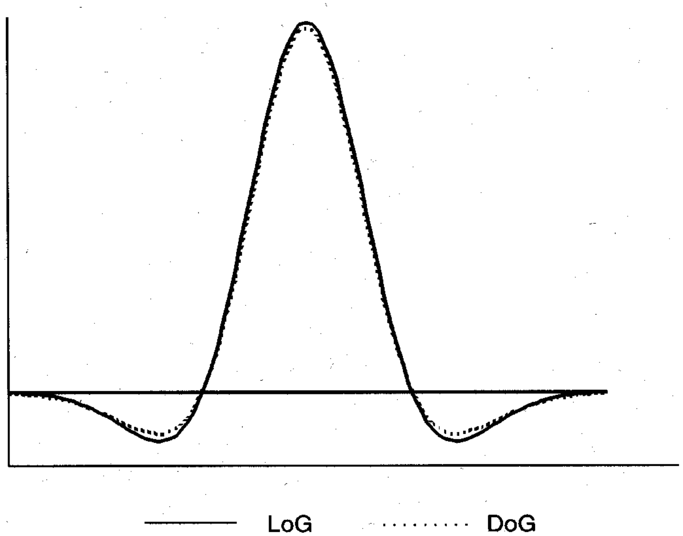
(Gonzales, Woods)

(The Marr-Hildreth edge detector - DoG)

- The LoG is possible to approximate by a difference of Gaussians (DoG) with $\sigma 1 > \sigma 2$

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

- The ratio $\sigma 1 / \sigma 2 = 1.6$ results in a good approximation of the LoG
- A LoG function ($\sigma = 12.35$), a DoG function ($\sigma 1 = 10$, $\sigma 2 = 16$)



7 × 7 mask

0	0	−1	−1	−1	0	0
0	−2	−3	−3	−3	−2	0
−1	−3	5	5	5	−3	−1
−1	−3	5	16	5	−3	−1
−1	−3	5	5	5	−3	−1
0	−2	−3	−3	−3	−2	0
0	0	−1	−1	−1	0	0

9 × 9 mask

0	0	0	−1	−1	−1	0	0	0
0	−2	−3	−3	−3	−3	−3	−2	0
0	−3	−2	−1	−1	−1	−2	−3	0
−1	−3	−1	9	9	9	−1	−3	−1
−1	−3	−1	9	19	9	−1	−3	−1
−1	−3	−1	9	9	9	−1	−3	−1
0	−3	−2	−1	−1	−1	−2	−3	0
0	−2	−3	−3	−3	−3	−3	−2	0
0	0	0	−1	−1	−1	0	0	0

(Edge detection using gradient operators)

- Various masks to compute the gradient ($gx = \delta f / \delta x$, $gy = \delta f / \delta y$) (intensity levels)

−1
1

−1	1

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

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

−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

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

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\begin{aligned} \nabla f &= \mathrm{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned}$$

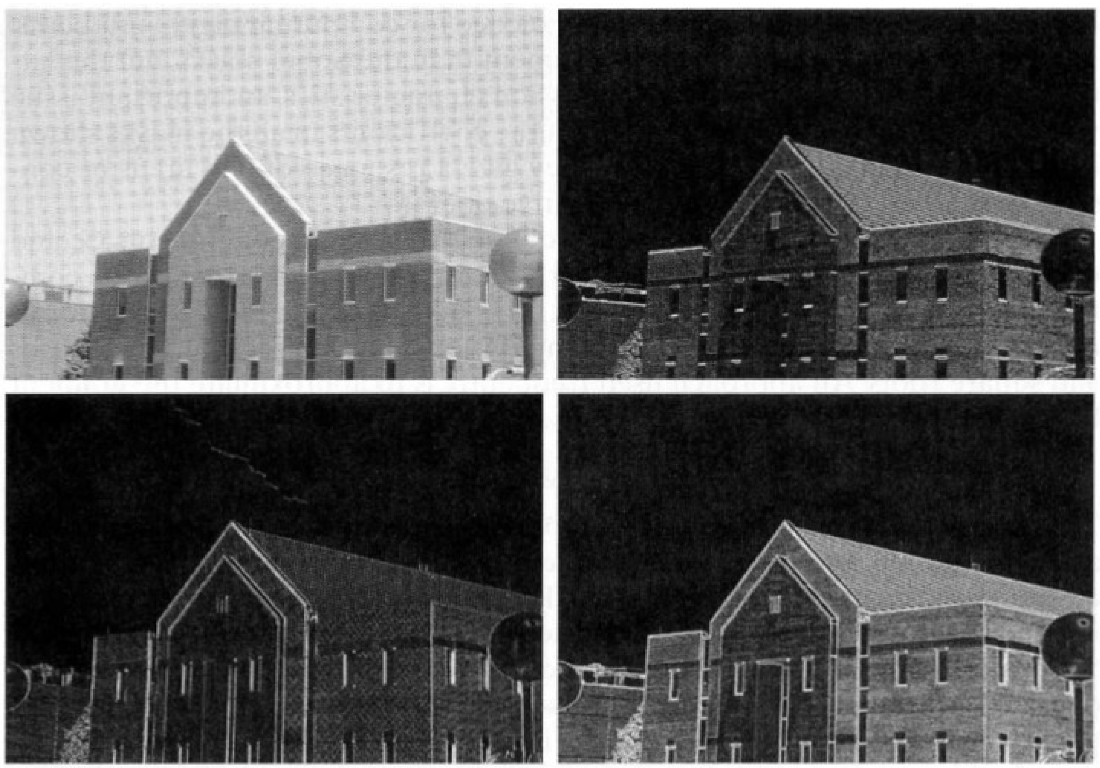
$$\nabla f \approx |G_x| + |G_y|$$

(Gonzales, Woods)

(Edge detection using gradient operators)

- (a) Original image of size 834 x 1114 pixels (b) The component of the gradient in the *x*-direction, $|g_x|$, (Sobel) (c) The component of the gradient in the *y*-direction, $|g_y|$, (Sobel) (d) The gradient image, $|g_x| + |g_y|$

−1	−2	−1
0	0	0
1	2	1



−1	0	1
−2	0	2
−1	0	1

(Gonzales, Woods)

(Edge linking)

- Local processing, a simplified procedure

1. Compute the gradient magnitude and angle arrays, $M(x,y)$, and $\alpha(x,y)$, of the input image, $f(x,y)$
2. Form a binary image, g, whose value at any pair of coordinates (x,y) is given by

$$g(x, y) = \begin{cases} 1 & \text{if } M(x, y) > T_M \text{ AND } \alpha(x, y) = A \pm T_A \\ 0 & \text{otherwise} \end{cases}$$

   where T_M is a threshold, A is a specified angle direction, and $\pm T_A$ defines a "band" of acceptable directions about A

3. Scan the rows of g and fill (set to 1) all gaps (sets of 0s) in each row that do not exceed a specified length, K (a gap bounded at both ends by one or more 1s) to obtain g_1
4. Rotate g by Θ (e.g. = 90°) and apply the horizontal scanning procedure in step 3.
5. Rotate the result back by $-\Theta$ to obtain g_2
6. Form the logical OR between g_1 and g_2 to obtain the final image

(7. Thin the final image using morphological thinning)

Suitable values are: T_M = 30% of the max. gradient, A = 0°, T_A = 45°,
$K \approx$ 5% of the image width (e.g., 25 pixels)