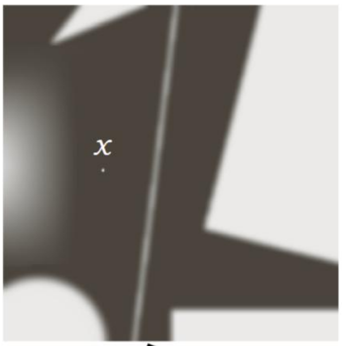
An image is represented by intensity values z_i



w_i represent weights within a 3x3 pre-defined window

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Mask response at a pixel, $R = \sum_{i=1}^{9} w_i z_i$

Working principle: For each pixel, x, in an image, if the absolute value of the mask response, $|R|$, is larger than or equal to some threshold T, $|R| \geq T$, then an intensity discontinuity is detected and located at the pixel.

[Gonzalez and Woods]

Point Detection

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

a b
c d

FIGURE 3.37 (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

[Gonzalez and Woods]

Edge Detection

First derivative

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

where z_i represent intensity values.

−1	0
0	1

0	−1
1	0

Roberts

For horizontal edge detection

−1	−1	−1
0	0	0
1	1	1

For vertical edge detection

−1	0	1
−1	0	1
−1	0	1

Prewitt

For horizontal edge detection

−1	−2	−1
0	0	0
1	2	1

For vertical edge detection

−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.

[Gonzalez and Woods]

Gradient operator

23. Diagonal edge masks for detecting discontinuities in the diagonal directions.

Prewitt

0	1	1		−1	−1	0
−1	0	1		−1	0	1
−1	−1	0		0	1	1

Sobel

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

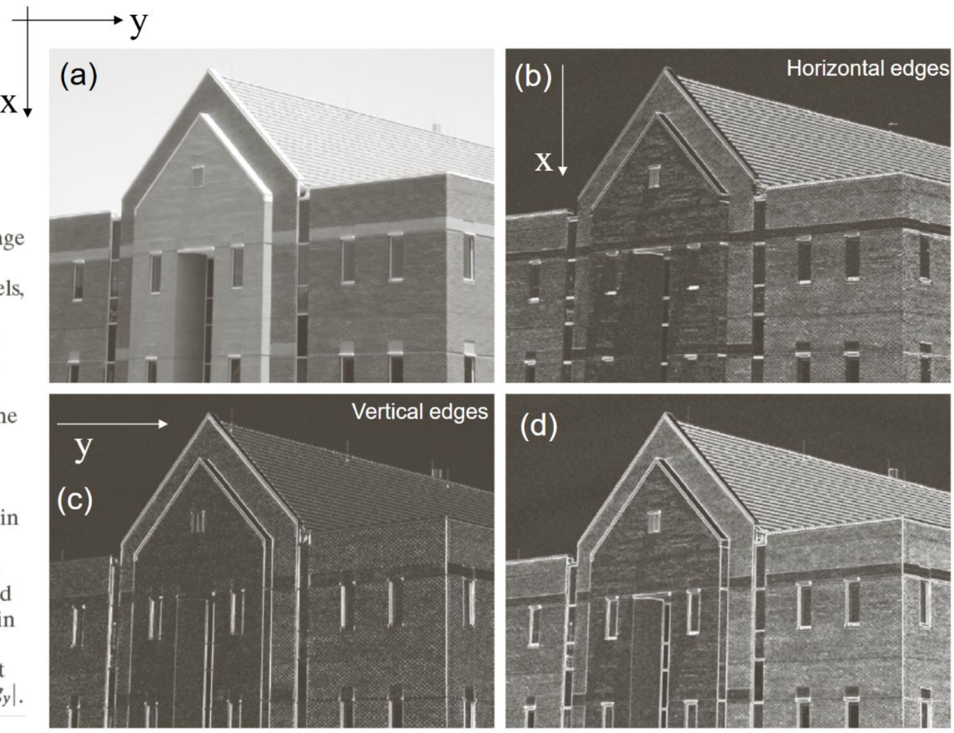
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

25

[Gonzalez and Woods]



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

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).

[Gonzalez and Woods]

Second derivative

Marr-Hildreth algorithm

1. Filter the input with an n-by-n Gaussian blurring filter $G(r)$.

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

2. Compute the Laplacian of the image resulting from Step 1 using one of the following 3-by-3 masks.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

[Gonzalez and Woods]

Marr-Hildreth algorithm

3. Find the zero crossings of the image from Step 2.
   a. A zero crossing at a pixel implies that the signs of at least two of its opposing neighboring pixels must be different.

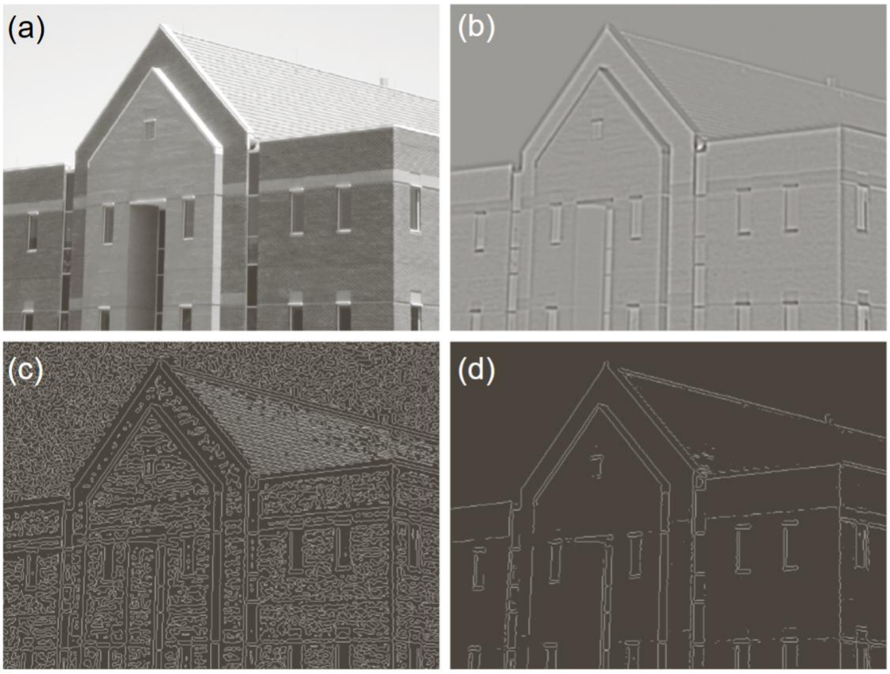
   b. There are four cases to test: left/right, up/down, and the two diagonals.

   c. For testing, the signs of the two opposing neighboring pixels must be different and their absolute Laplacian values must be larger than or equal to some threshold.

   d. If yes, we call the current pixel a zero-crossing pixel.

[Gonzalez and Woods]

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.

[Gonzalez and Woods]

If precise edge localization is more important and noise is not a significant issue, the second derivative might be preferred. On the other hand, if the goal is to capture broader edge details with some tolerance for noise, the first derivative might be sufficient.

Canny edge detector

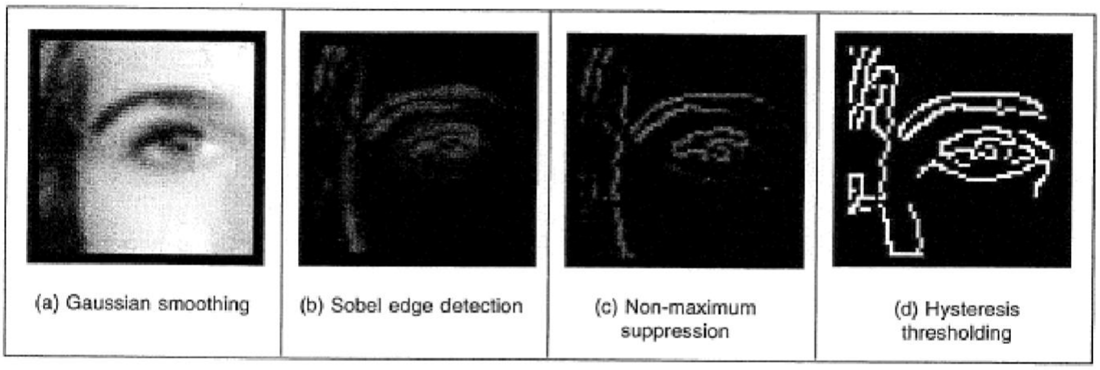
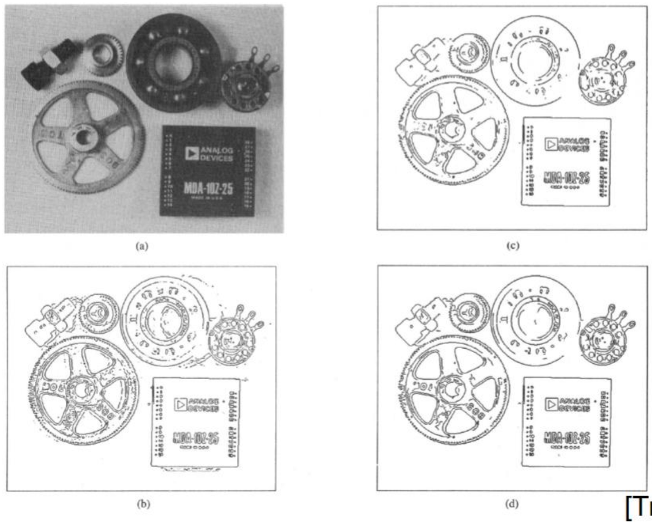


Figure 4.15 Stages in Canny edge detection

A Gaussian filter followed by the Sobel operator is often referred to an approximation of an optimal filter.

Canny edge detection algorithm

- There are three main steps in the Canny edge detection
- [1] Gradient calculation for computing gradient magnitudes at all pixels
- [2] Non-maxima suppression
- [3] Hysteresis thresholding



T_{high}

T_{low}

Hysteresis thresholding

10

[Treiber-10, Canny-86]

Non-maximum suppression:

1. **Gradient Calculation:** First, calculate the gradient magnitude and direction at each pixel in the image. This involves using an edge detection operator like Sobel, Prewitt, or Roberts.
2. **Directional Consideration:** The gradient direction at each pixel points in the direction of the steepest ascent from dark to light, giving an indication of the edge orientation.
3. **Suppression Process:** During non-maximum suppression, for each pixel, you compare its gradient magnitude to the magnitudes of the pixels in the positive and negative gradient directions. Only the pixel with the maximum gradient magnitude along this line (i.e., perpendicular to the edge direction) is considered to be part of an edge. All other pixels that are not local maxima are suppressed (set to zero), which thins the edges.

Hysteresis thresholding:

1. **Two Thresholds:** Hysteresis thresholding involves two thresholds—a lower threshold and a higher threshold.
   - **Lower Threshold:** Pixels with intensities above the lower threshold are considered as potential edges.
   - **Higher Threshold:** Pixels with intensities above the higher threshold are confirmed as edge pixels.
2. **Edge Linking:** Pixels that are between these two thresholds are only considered as edge pixels if they are connected to pixels that are above the higher threshold. This helps in preserving genuine edge structures while eliminating small, spurious edges caused by noise or other irrelevant variations.

3. **Propagation of Edge Status:**
   - If a weak edge pixel is directly connected to a strong edge pixel (either horizontally, vertically, or diagonally), it is also considered to be a true edge pixel.
   - This connectivity is checked recursively, meaning if a weak pixel is connected to another weak pixel that is already confirmed as an edge pixel (due to its connection to a strong edge pixel), it too becomes part of the edge.

Line Detection

1. A line is detected when more than one aligned, connected points are detected;
2. Or, the response of line mask is greater than some threshold, e.g.,

y

x

−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°

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

The above are line masks for detecting lines (1 pixel thick) in 4 different specific directions.

4. If 4-line masks are used, then the final response is equal to the largest response among the masks.

$$R = \max\left(\left|R_{horizontal}\right|, \left|R_{45^o}\right|, \left|R_{vertical}\right|, \left|R_{-45^o}\right|\right)$$