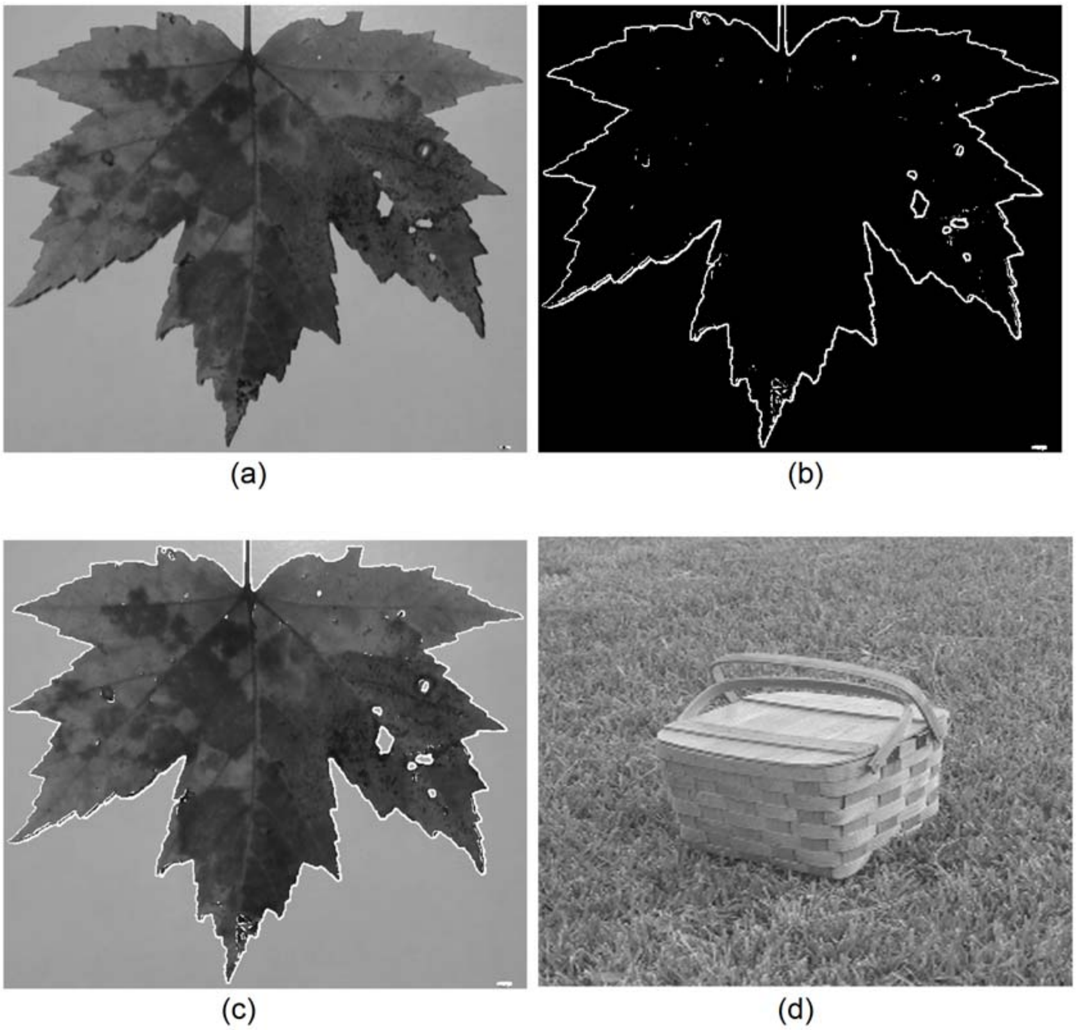
6 – EDGE DETECTION

Changes or discontinuities in an image attribute such as luminance or texture are important primitive features of an image since they often provide an indication of the physical extent of objects within the image. Edge detection contributes significantly to algorithms for feature detection, segmentation, and motion analysis.

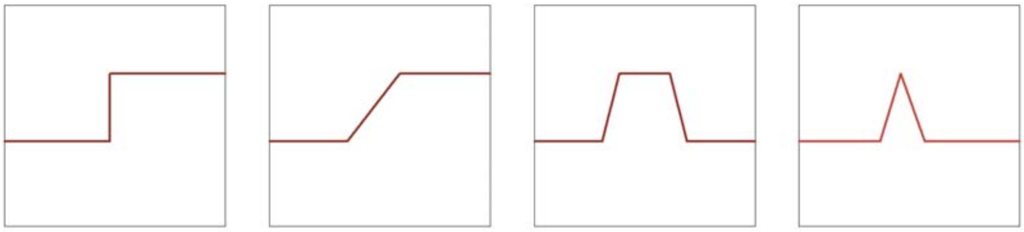
In this course, we consider edges to be local discontinuities in intensity.



(a) Leaf (b) Detected edges (c) Edges superimposed on original (d) Texture example

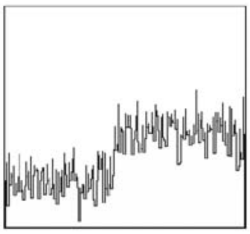

Original
Detected edges
• THE LATEST ON
CAD FRAMEWORKS
• EVALUATE HIGH-
PERFORMANCE ADCs
• ANALOG PROCESSES

Example - Edge profiles

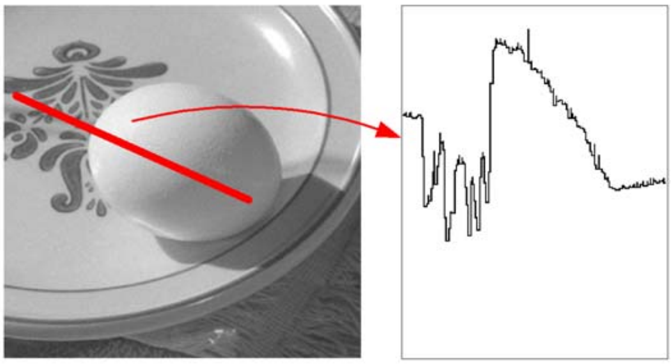


Step edge Ramp edge Line edge Roof edge

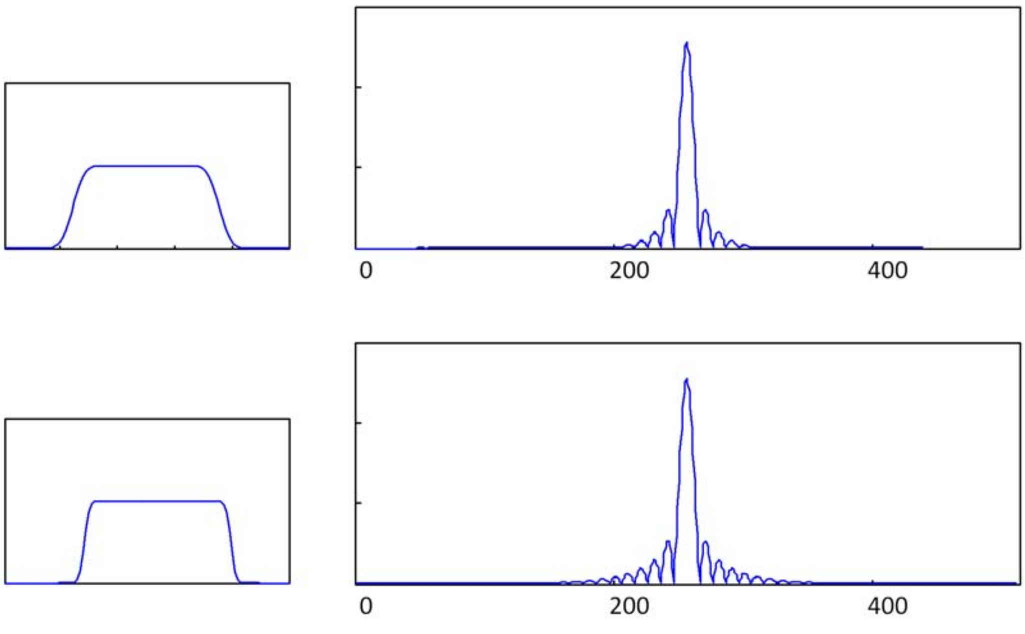
Example of
a “real” edge



Noisy edge



Where are the edges?



Sharpness of edge and Fourier spectrum

Gradient Operators

For an image function $f(x, y)$, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ are the rates of change in the x and y directions, respectively. The gradient at any point (x, y) is defined as the vector

$$\mathbf{G}(x, y) = \begin{bmatrix} \partial f(x, y)/\partial x \\ \partial f(x, y)/\partial y \end{bmatrix} \equiv \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (1)$$

The magnitude of the gradient is

$$|\mathbf{G}(x, y)| = [G_x^2 + G_y^2]^{1/2}. \quad (2)$$

An approximation is

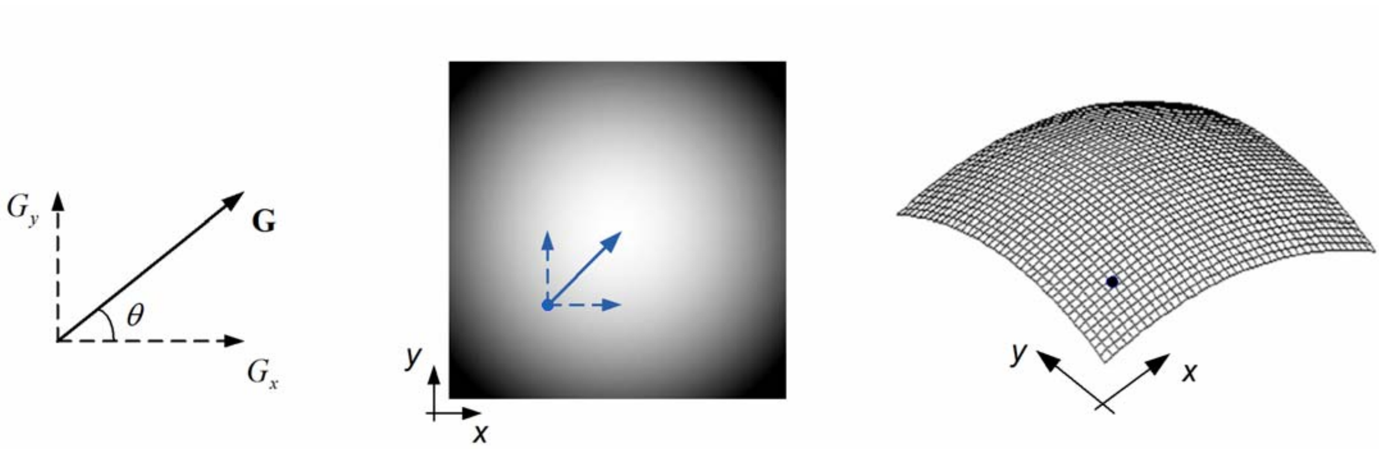
$$|\mathbf{G}(x, y)| \approx |G_x| + |G_y|. \quad (3)$$

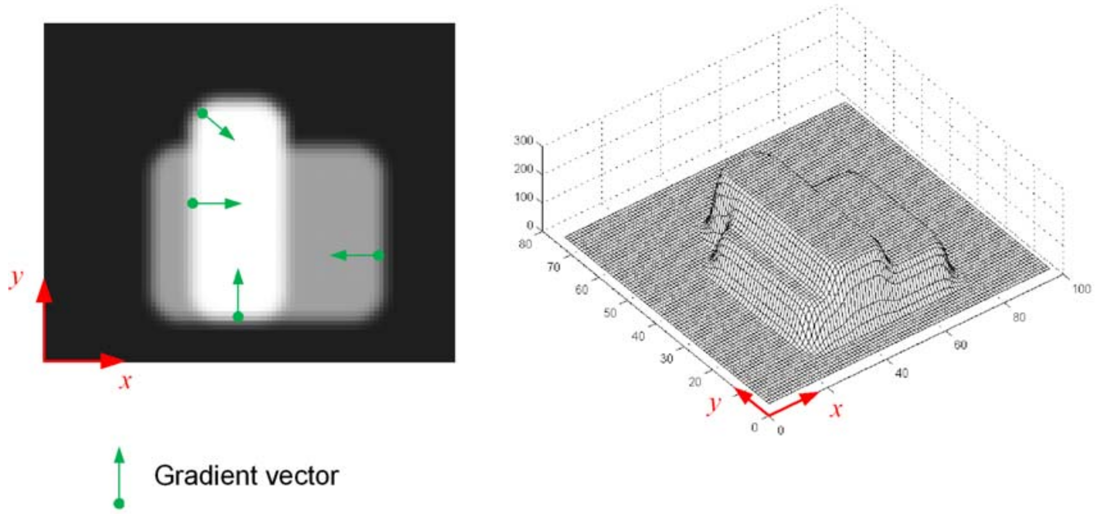
The direction of the gradient vector $\mathbf{G}$ at (x, y) measured with respect to the x axis is

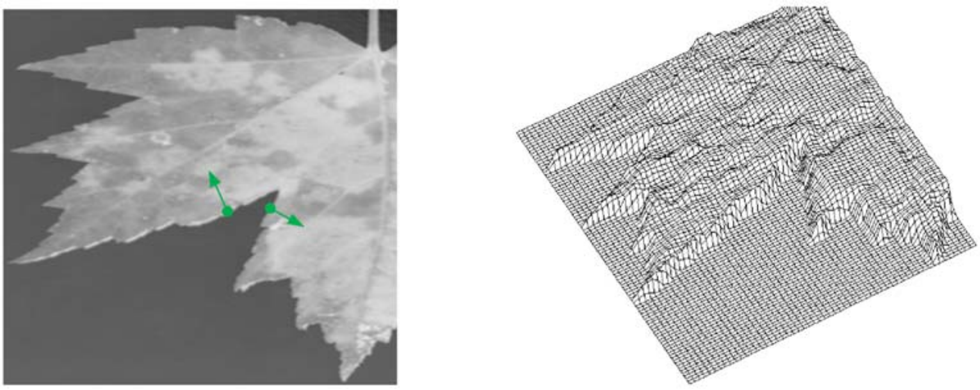
$$\theta(x, y) = \tan^{-1}(G_y/G_x) \quad (4)$$

It can be shown that the gradient vector points in the direction of maximum rate of change of f at location (x, y).

An edge is a pixel whose gradient magnitude is "sufficiently large".







How do we obtain the gradient at any point in a digital image?

We can use the forward derivative approximation, D_1, defined as

$$D_1 = \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{5}$$

to obtain

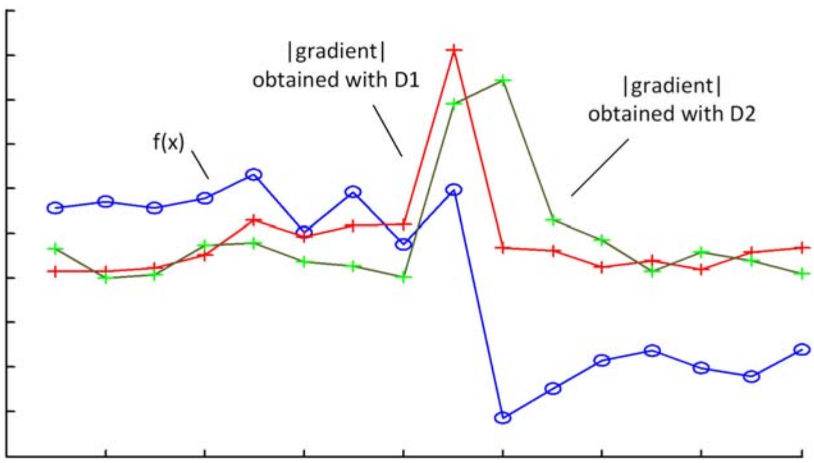
$$\frac{\partial f(x,y)}{\partial x} \approx D_{1x} = \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x} \tag{6}$$

$$\frac{\partial f(x,y)}{\partial y} \approx D_{1y} = \frac{f(x, y+\Delta y) - f(x,y)}{\Delta y} \tag{7}$$

This corresponds to masking operations with

$$[-1 \quad 1] \qquad \text{and} \qquad \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

(Δx and Δy may be set to 1, x axis points to the right, y axis points up.)



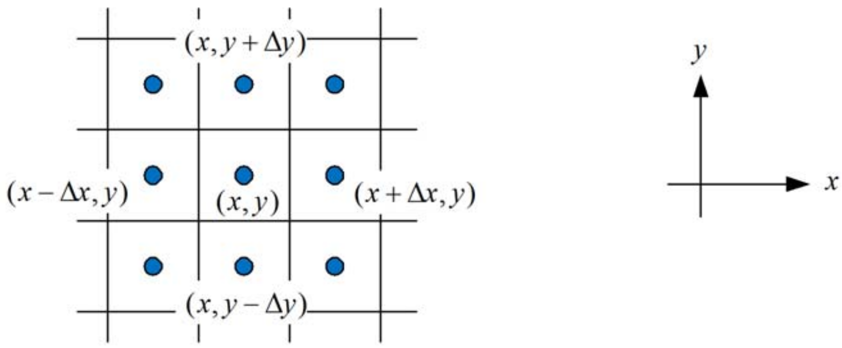
Better estimates are obtained using the *centred* difference approximation

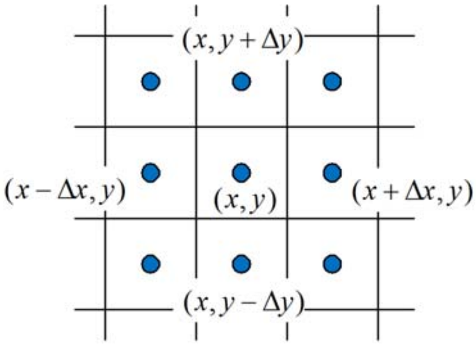
$$D_{2x} = \frac{f(x+\Delta x, y) - f(x-\Delta x, y)}{2(\Delta x)} \tag{8}$$

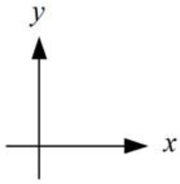
$$D_{2y} = \frac{f(x, y+\Delta y) - f(x, y-\Delta y)}{2(\Delta x)} \tag{9}$$

equivalent to masking operations with

$$[-1 \quad 0 \quad 1] \qquad \text{and} \qquad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$







Variants on D_2 are the Prewitt masks:

$$D_{Px} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad D_{Py} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

which form smoothed, centred difference operators.

The *Sobel* weighting masks emphasize the central pixel:

$$D_{Sx} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad D_{Sy} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The Sobel edge filter provides good edge detection and is less sensitive to noise in the image compared to Prewitt.

The *Roberts* operator uses 2×2 masks, and is defined by

$$D_+ = f(x+\Delta x, y+\Delta y) - f(x, y) \tag{10}$$

$$D_- = f(x, y+\Delta y) - f(x+\Delta x, y) \tag{11}$$

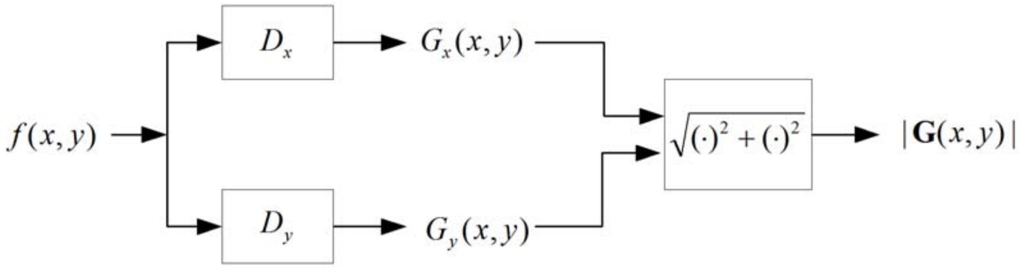
corresponding to filtering the image function with masks

$$D_+ = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad \text{and} \qquad D_- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The centre of each mask may be taken to be the lower left pixel.

The output image $G(x, y)$ can be set equal to the gradient of the input image $f(x, y)$ at that point:

$$G(x, y) = |\mathbf{G}[(x, y)]|. \tag{12}$$

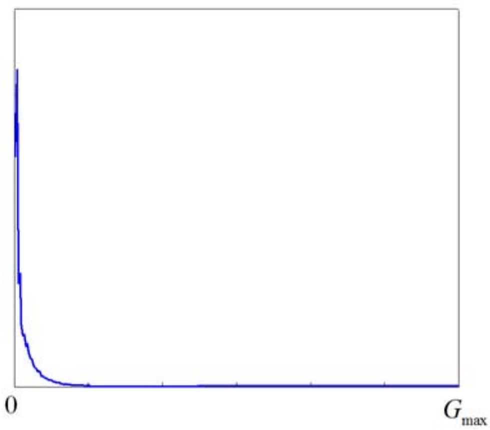


To obtain an edge map:

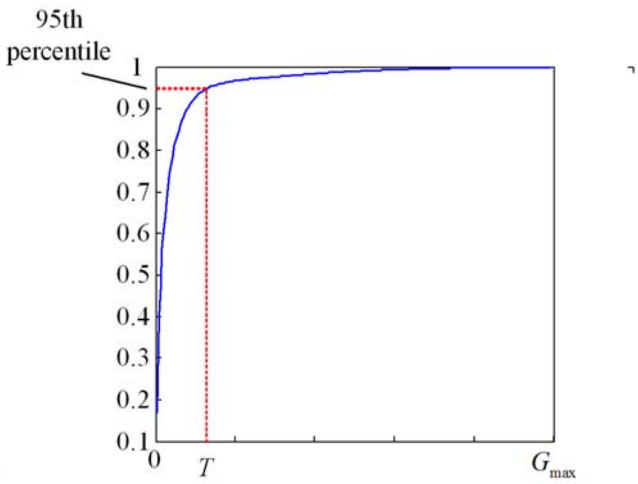
$$G(x, y) = \begin{cases} 1 & \text{if } |\mathbf{G}[(x, y)]| > T \\ 0 & \text{if } |\mathbf{G}[(x, y)]| \leq T. \end{cases} \tag{13}$$

The edge map is used for obtaining the object boundaries in an image. A suitable threshold T may be selected using the cumulative histogram of $G(x, y)$; typically, about 5 to 10% of pixels with the largest gradients are declared as edges.

Grad. magnitude image



Histogram of grad. magnitude



Cumulative histogram of grad. magnitude

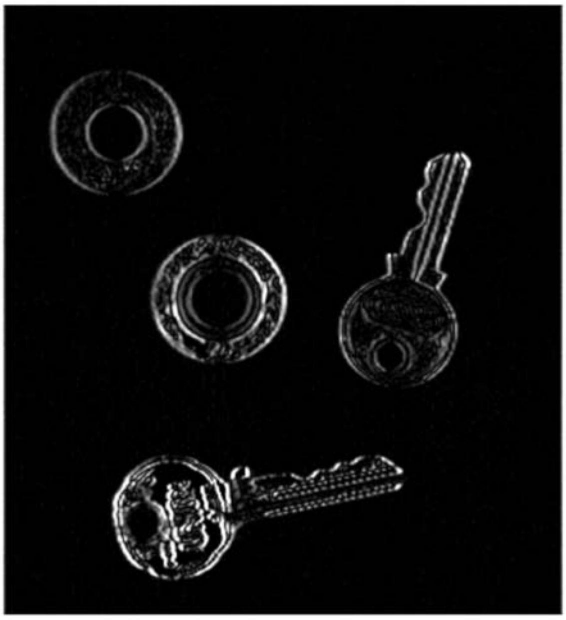
Edge detection

$|G_x|$

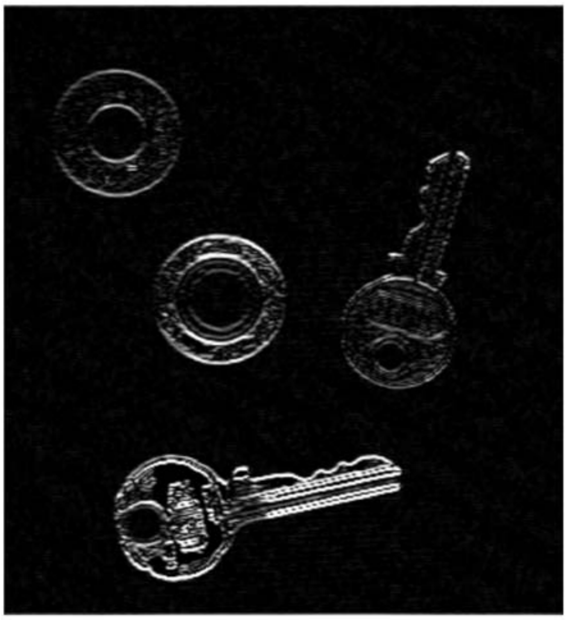
$|G_y|$

$\left(G_x^2 + G_y^2\right)^{1/2}$

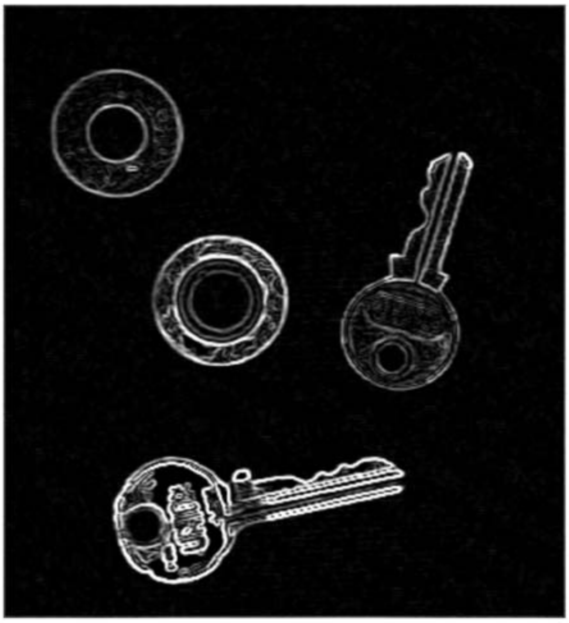
Edge detection



$|G_x|$



$|G_y|$

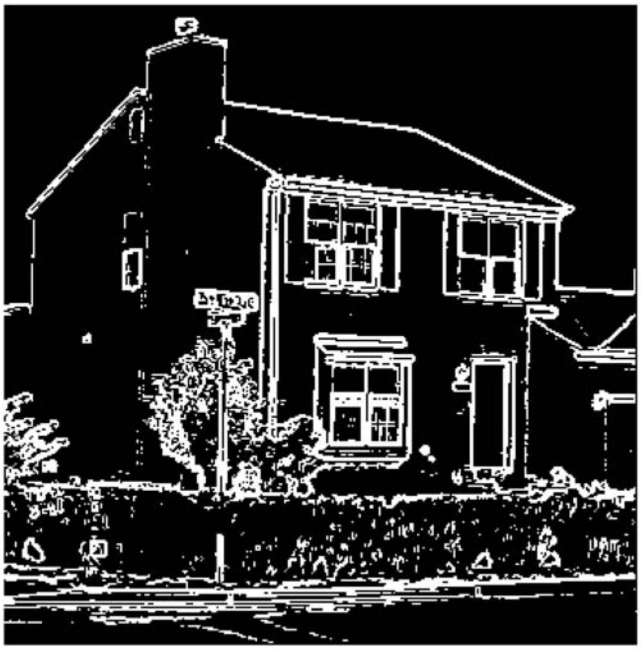


$\left(G_x^2 + G_y^2\right)^{1/2}$

Edge map

Gradient magnitude

With low threshold



With moderate threshold

With high threshold

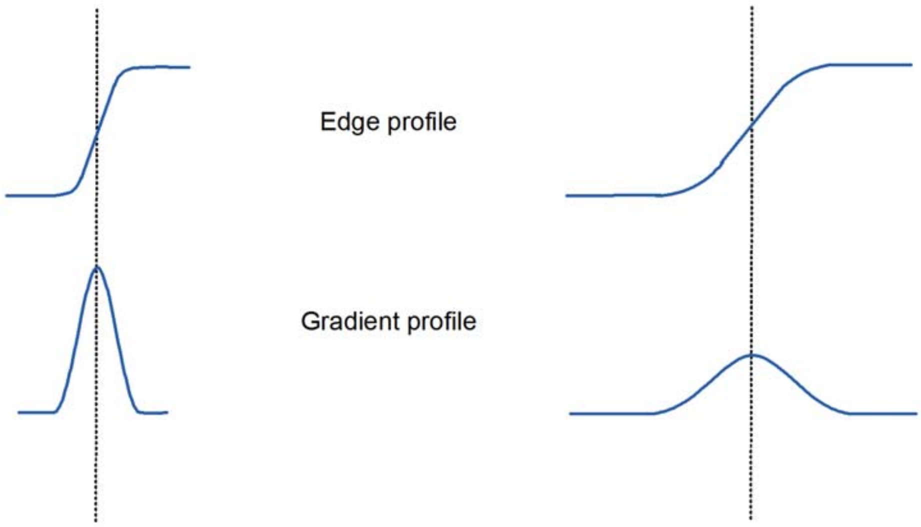
Problems

Problems may be encountered with the use of gradient operators:

1. Boundaries may be diffused instead of sharp; the former have lower gradients than the latter, thus complicating the threshold selection.

Slightly blurred rectangle

Heavily blurred rectangle



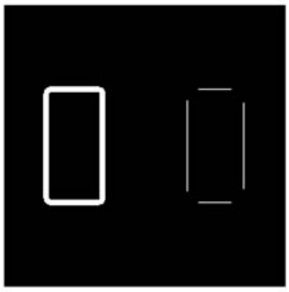
Profiles across one edge of the rectangle

Profiles across one edge of the rectangle

Original

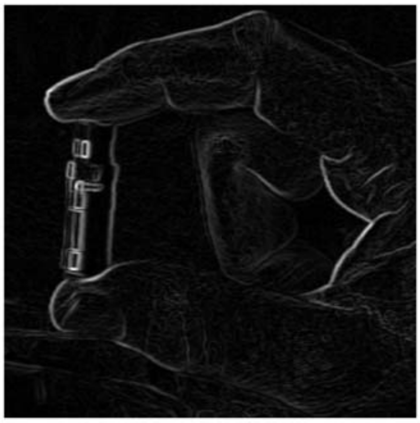
After Sobel

Profile



After thresholding

Original

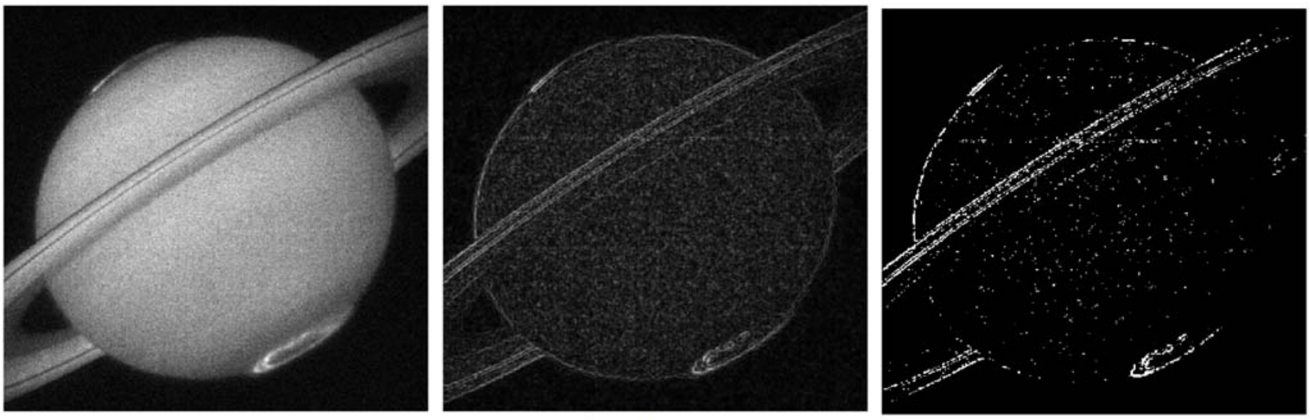


After Sobel

High threshold

Low threshold

2. Noise can result in stronger gradients than meaningful edges.

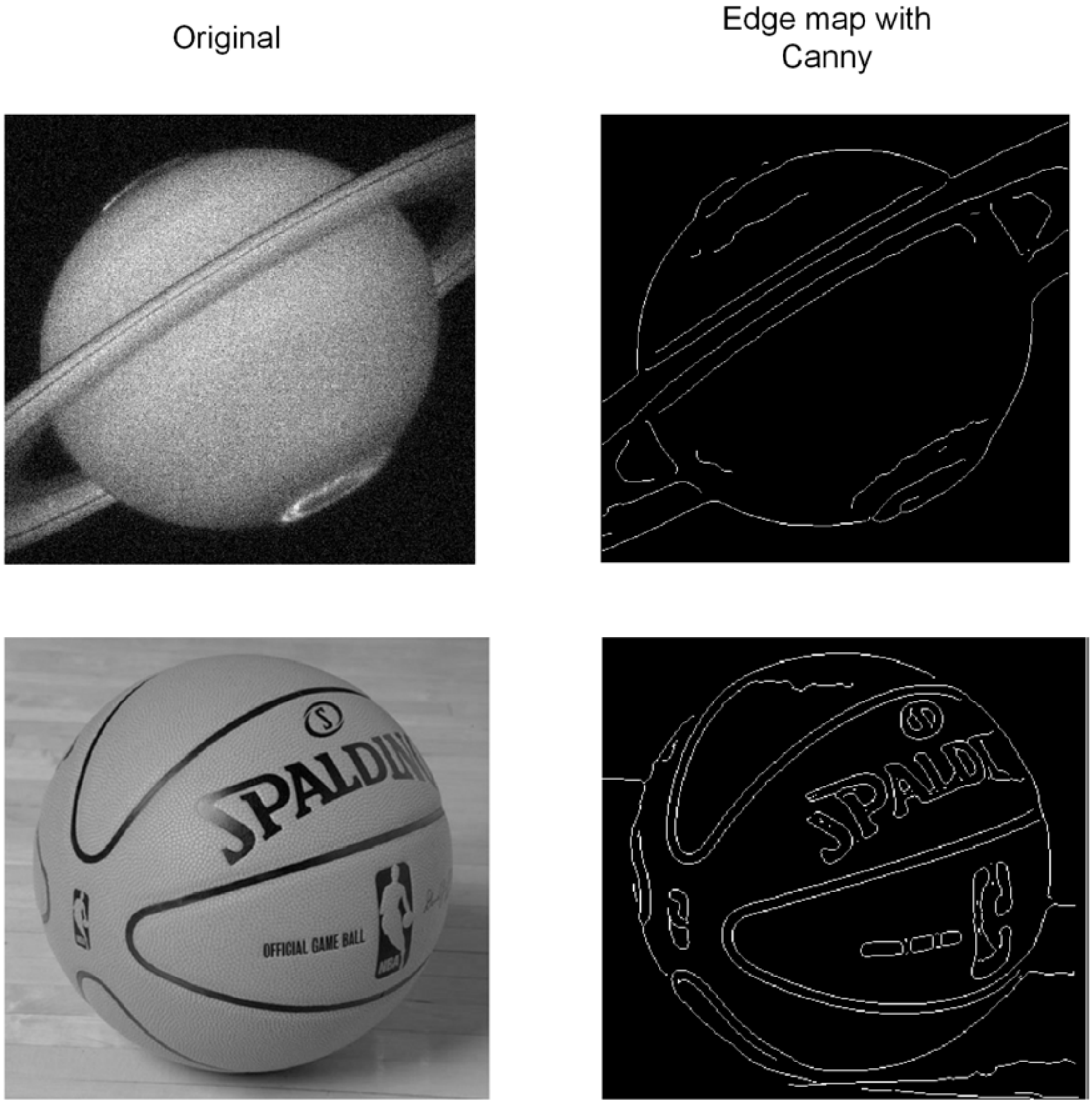


Original | Gradient magnitude | Edge map

3. Edges after thresholding are often thick.

Original | Gradient magnitude | Edge map

More sophisticated algorithms (e.g., Canny operator) can be used to give better results.



Laplacian Operator

The Laplacian is a second-order derivative operator defined as

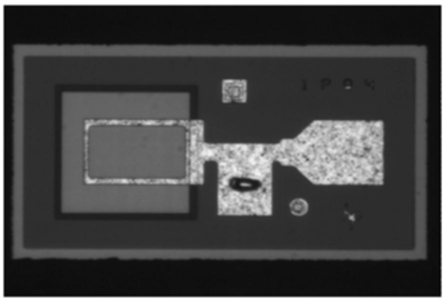
$$L[f(x,y)] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{14}$$

A Laplacian mask is

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

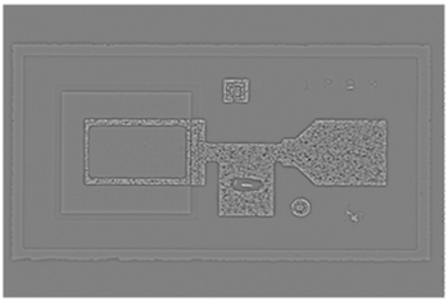
The Laplacian operator is seldom used by itself for edge detection because it is unacceptably sensitive to noise. It is used in conjunction with a Gaussian filter for edge detection. This is known as a Laplacian of Gaussian (LOG) filter.

Original

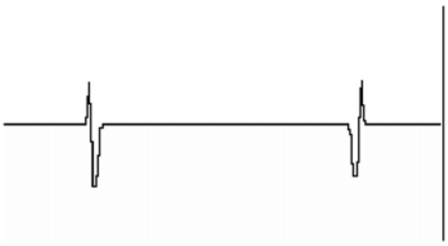


Original

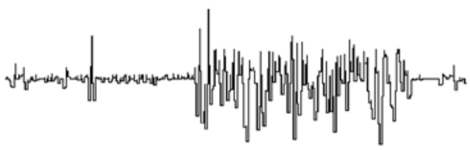
After Laplacian
(gray levels + 128)



After Laplacian
(gray levels + 128)

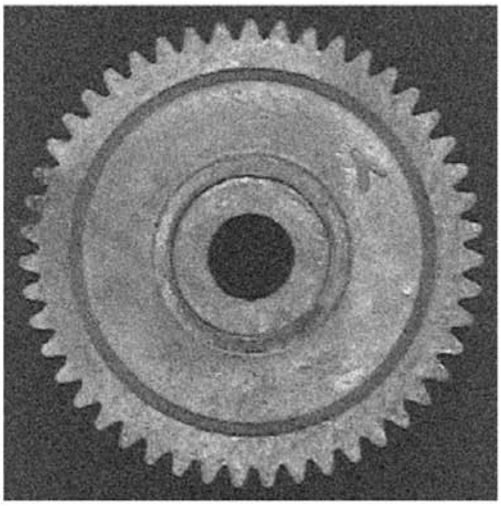


Profile through center

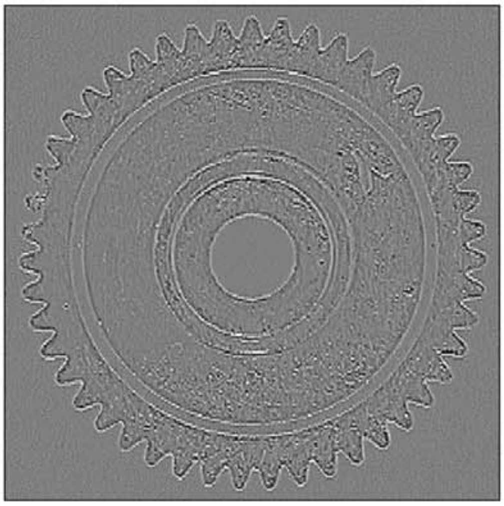


Profile through center

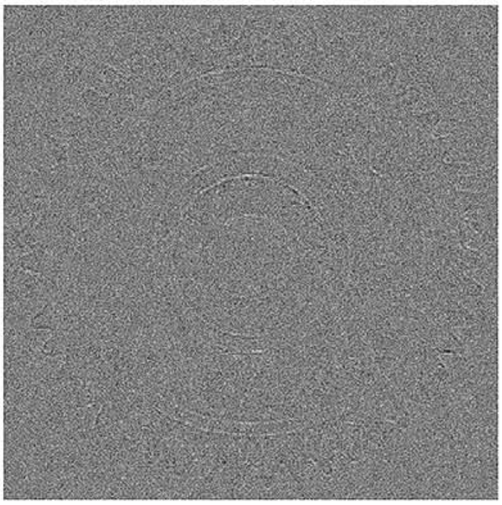
Original



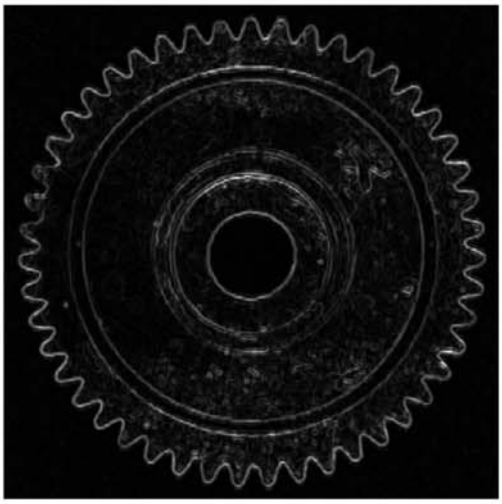
With Gaussian noise



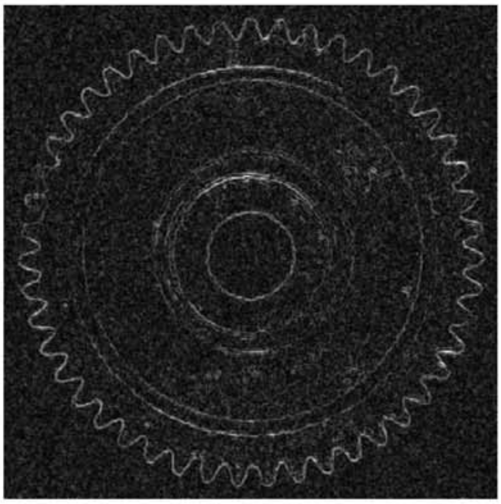
Laplacian



Laplacian



Sobel



Sobel