Image Features (II)

COMP 4900D

Winter 2006

Edge Detection using Derivatives

1-D image $f(x)$

1$^{\text{st}}$ derivative $f'(x)$

$|f'(x)|$ threshold

Pixels that pass the threshold are edge pixels

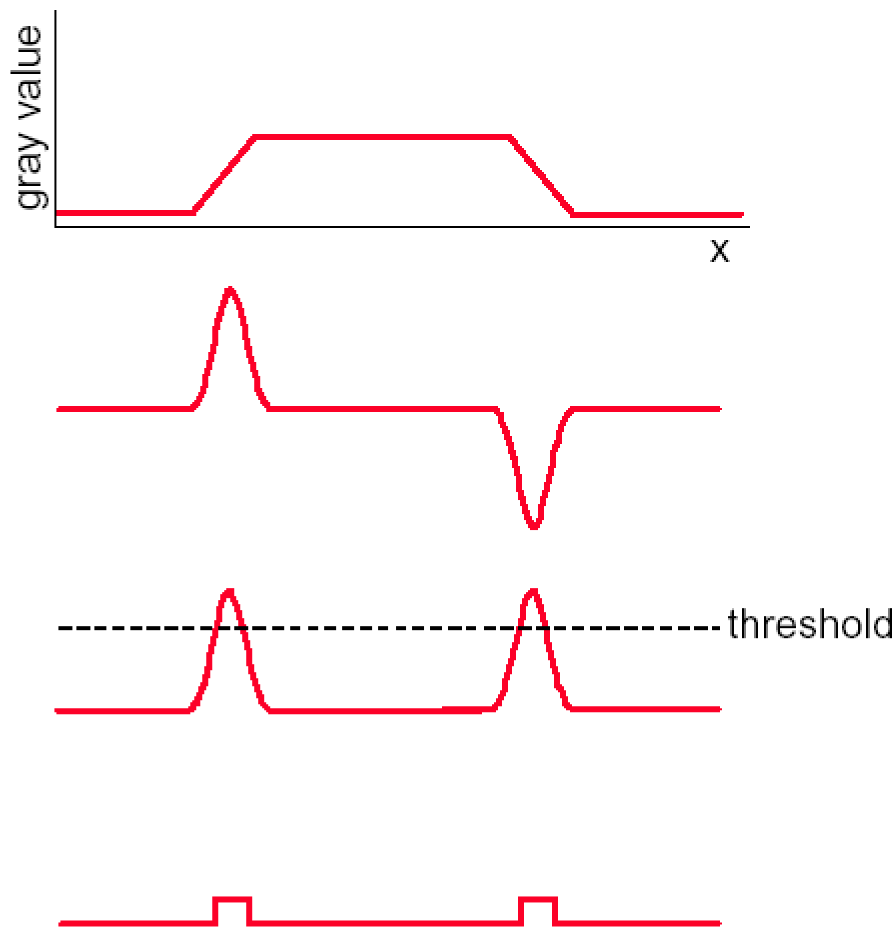
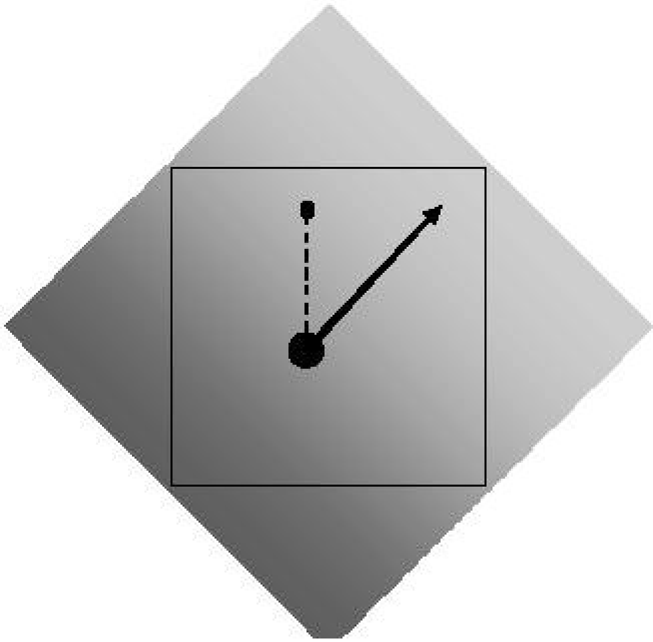


Image Gradient



gradient

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

magnitude

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

direction

$$\arctan\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Finite Difference for Gradient

Discrete approximation:

$$I_x(i, j) = \frac{\partial f}{\partial x} \approx f_{i+1,j} - f_{i,j}$$

$$I_y(i, j) = \frac{\partial f}{\partial y} \approx f_{i,j+1} - f_{i,j}$$

Convolution kernels:

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

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

magnitude $$G(i, j) = \sqrt{I_x^2(i, j) + I_y^2(i, j)}$$

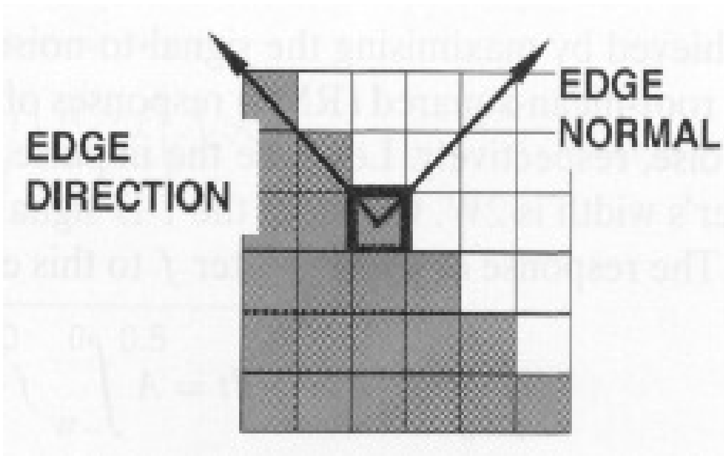
aprox. magnitude $$G(i, j) \approx \left|I_x\right| + \left|I_y\right|$$

direction $$\arctan(I_y / I_x)$$

Edge Detection Using the Gradient

Properties of the gradient:

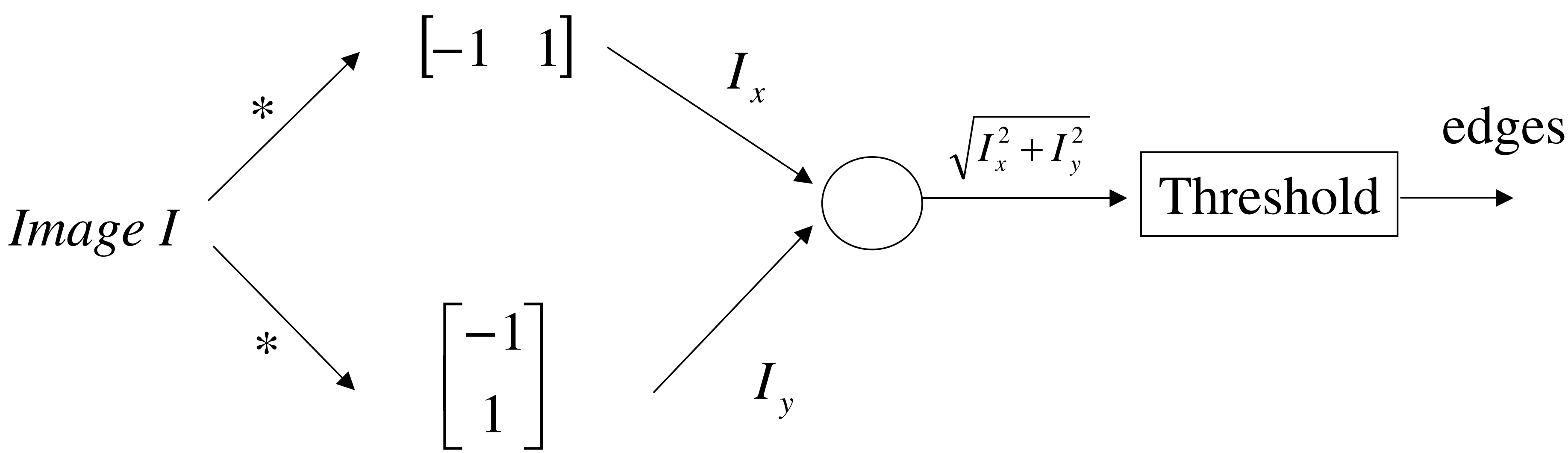
- The magnitude of gradient provides information about the strength of the edge
- The direction of gradient is always perpendicular to the direction of the edge



Main idea:

- Compute derivatives in x and y directions
- Find gradient magnitude
- Threshold gradient magnitude

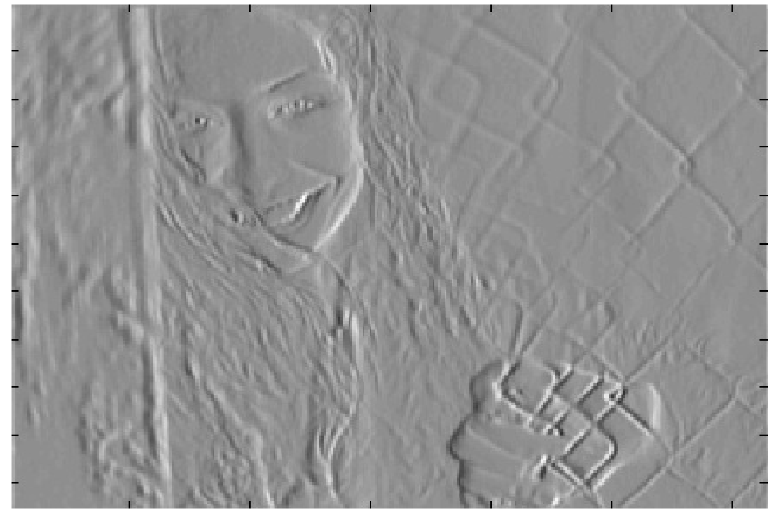
Edge Detection Algorithm



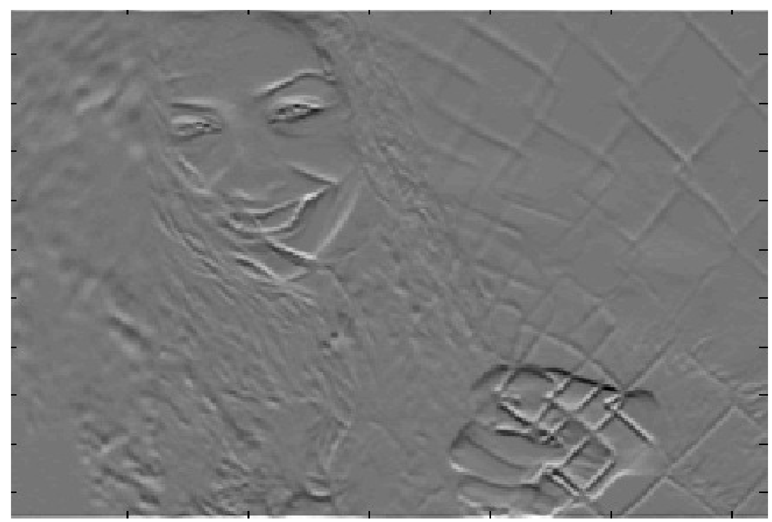
Edge Detection Example

I

I_x



I_y

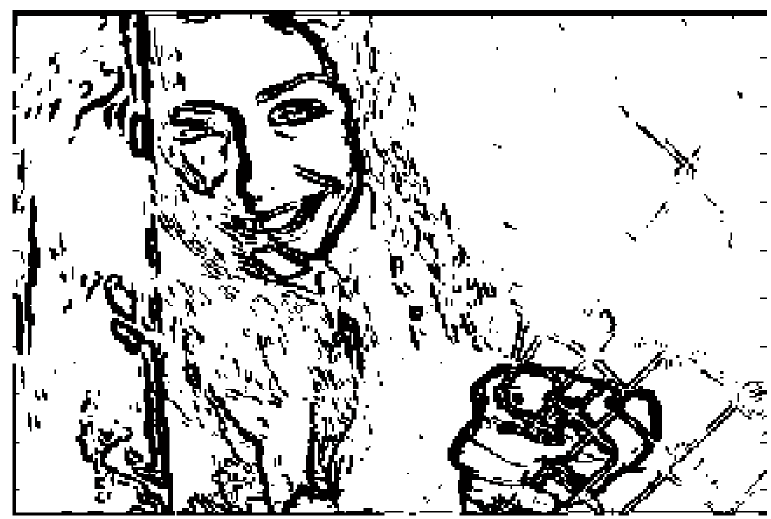


Edge Detection Example

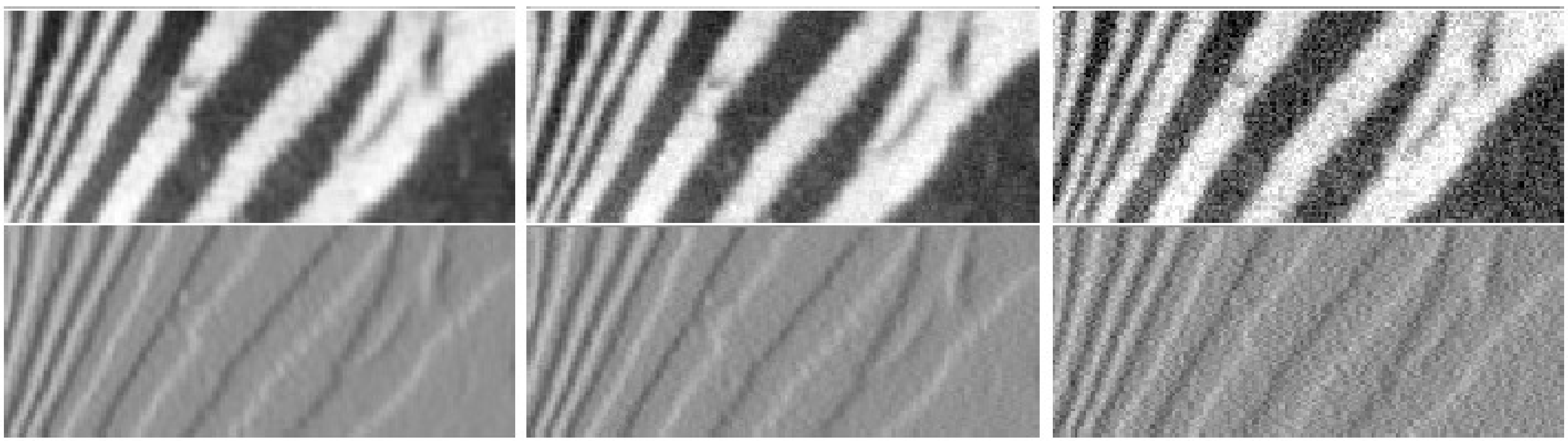
$$G(i, j) = \sqrt{I_x^2(i, j) + I_y^2(i, j)}$$

I

$$G(i, j) > Threshold = \tau$$



Finite differences responding to noise

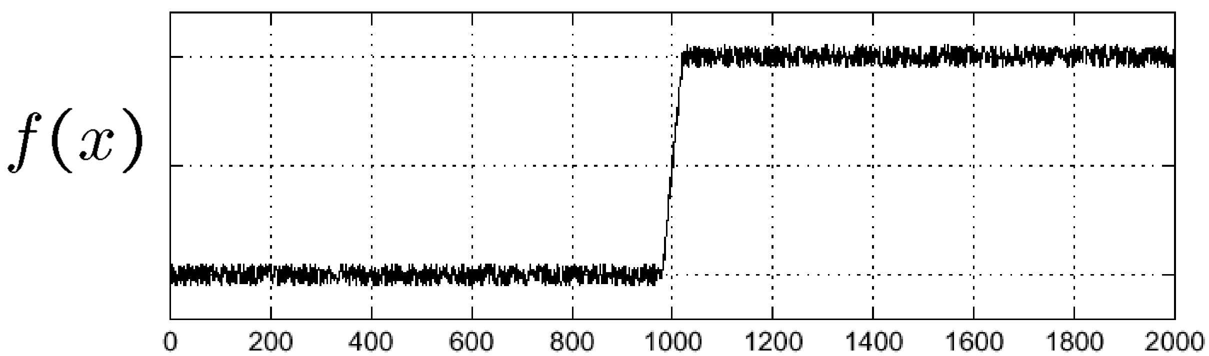


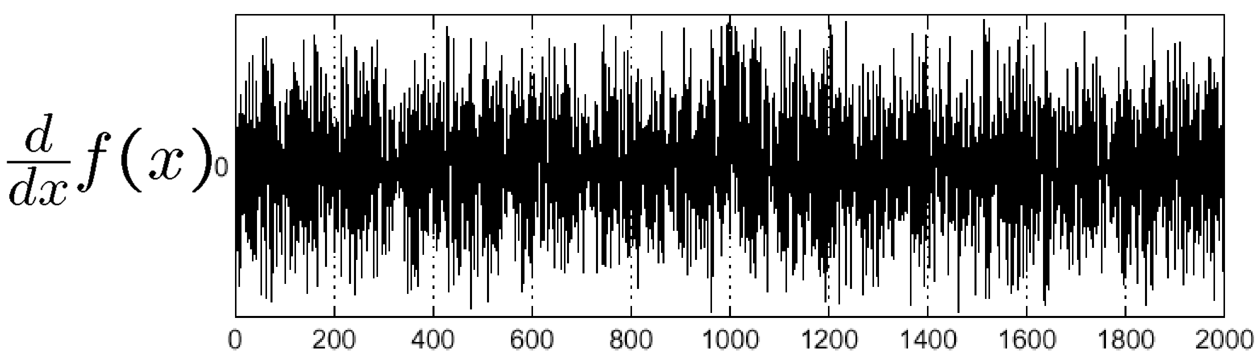
Increasing noise ->
(this is zero mean additive gaussian noise)

Effects of noise

Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal





Where is the edge?

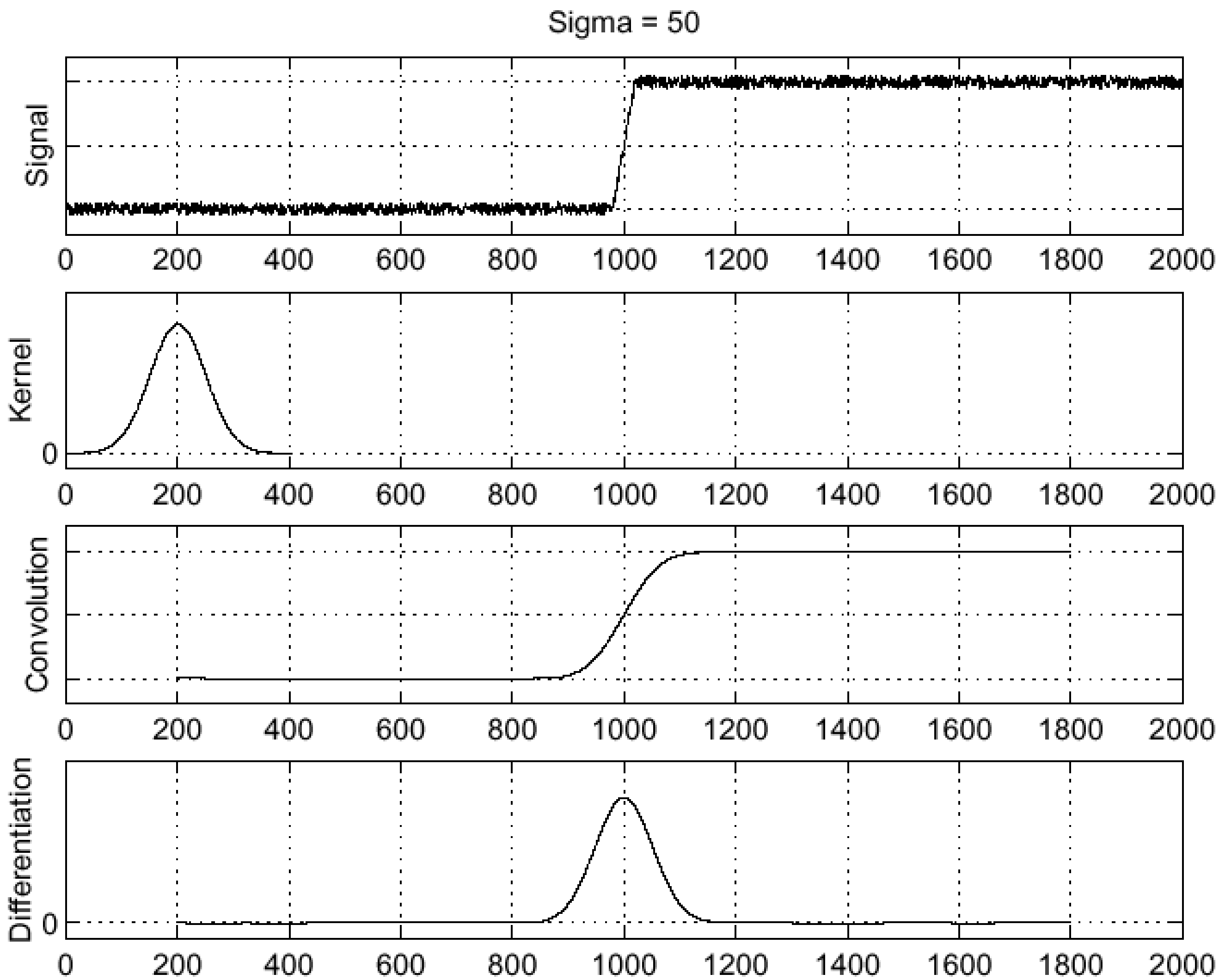
Solution: smooth first

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



Where is the edge? Look for peaks $\frac{\partial}{\partial x}(h \star f)$

Sobel Edge Detector

Approximate derivatives with central difference

$$I_x(i, j) = \frac{\partial f}{\partial x} \approx f_{i-1,j} - f_{i+1,j}$$

Convolution kernel

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

Smoothing by adding 3 column neighbouring differences and give more weight to the middle one

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Convolution kernel for I_y

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel Operator Example

$$\begin{array}{|c|c|c|} \hline a_1 & a_2 & a_3 \\ \hline a_4 & \boxed{a_5} & a_6 \\ \hline a_7 & a_8 & a_9 \\ \hline \end{array} \quad * \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

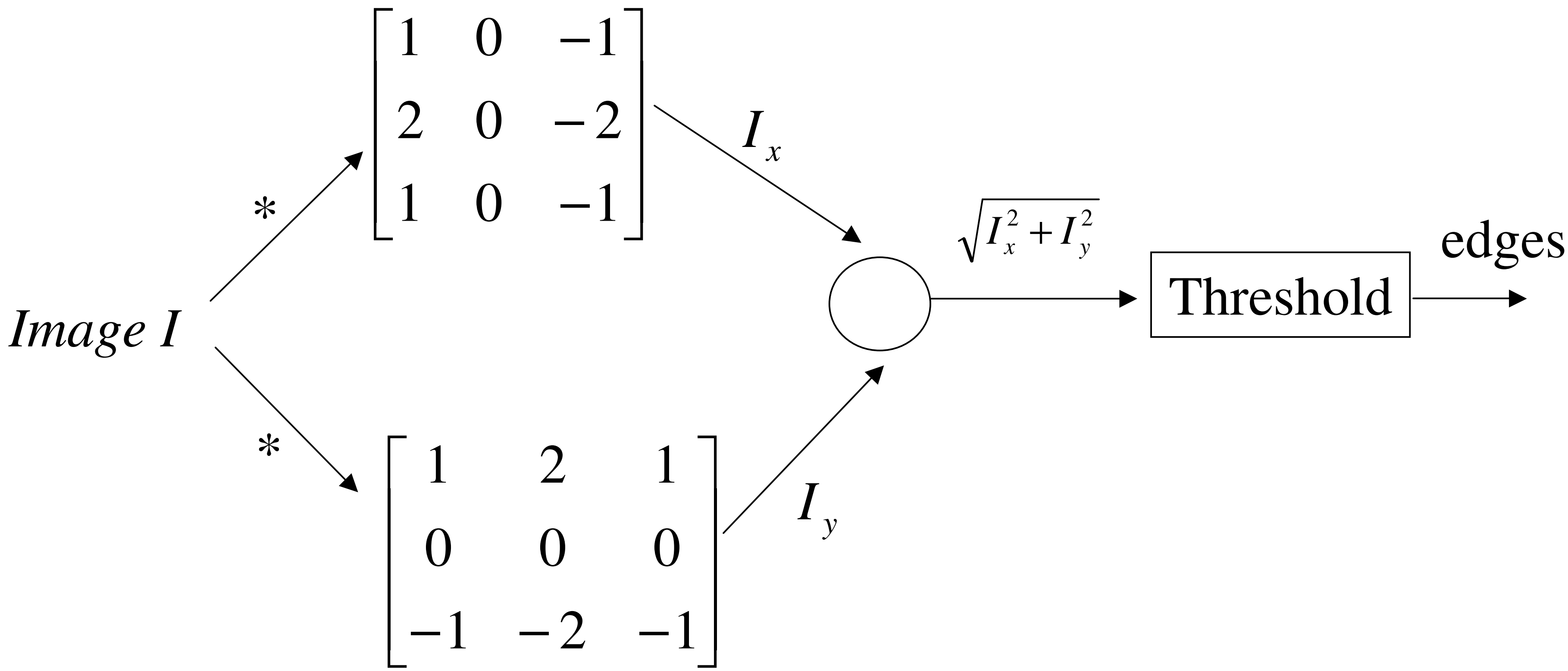
$$\begin{array}{|c|c|c|} \hline a_1 & a_2 & a_3 \\ \hline a_4 & \boxed{a_5} & a_6 \\ \hline a_7 & a_8 & a_9 \\ \hline \end{array} \quad * \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The approximate gradient at a_5

$$I_x = (a_1 - a_3) + 2(a_4 - a_6) + (a_7 - a_9)$$

$$I_y = (a_1 - a_7) + 2(a_2 - a_8) + (a_3 - a_9)$$

Sobel Edge Detector



Edge Detection Summary

Input: an image I and a threshold τ.

1. Noise smoothing: $I_s = I * h$
   (e.g. h is a Gaussian kernel)

2. Compute two gradient images I_x and I_y by convolving I_s with gradient kernels (e.g. Sobel operator).

3. Estimate the gradient magnitude at each pixel

$$G(i, j) = \sqrt{I_x^2(i, j) + I_y^2(i, j)}$$

4. Mark as edges all pixels (i, j) such that $G(i, j) > \tau$