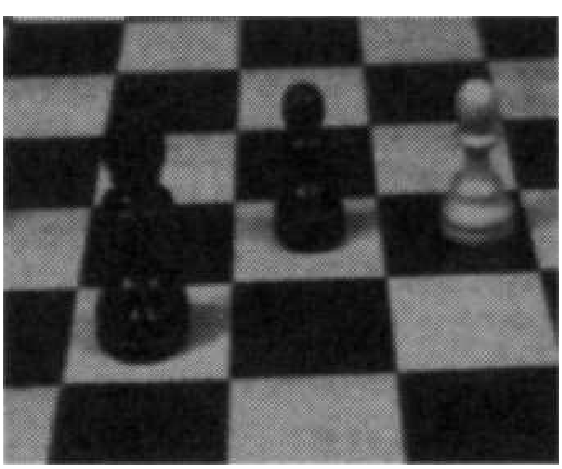
Edge Detection

- Edge detection masks
- Gradient Detectors
- Compass Detectors
- Second Derivative - Laplace detectors
- Edge Linking
- Hough Transform



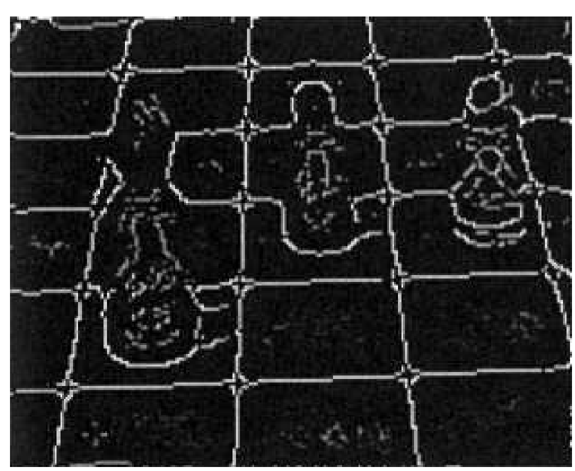


Image Processing - Computer Vision

Low Level

Image Processing

representation,
compression,transmission

image enhancement

edge/feature finding

image "understanding"

Computer Vision

High Level

UFO - Unidentified Flying Object

Point Detection

Convolution with:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Large Positive values = light point on dark surround
Large Negative values = dark point on light surround

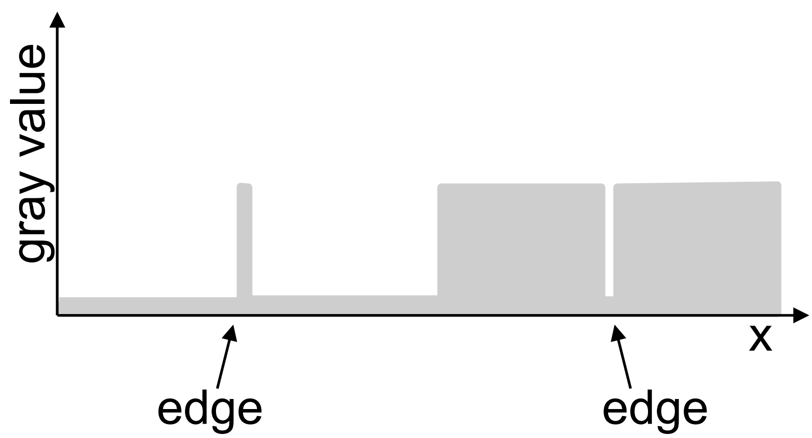
Example:

$$\begin{matrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 100 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{matrix} \quad * \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

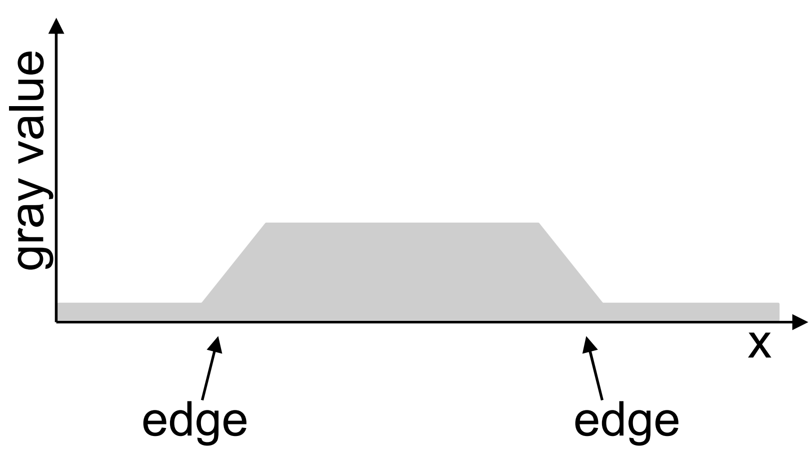
$$= \quad \begin{matrix} 0 & 0 & -95 & -95 & -95 \\ 0 & 0 & -95 & 760 & -95 \\ 0 & 0 & -95 & -95 & -95 \end{matrix}$$

Edge Definition

Line Edge



Step Edge



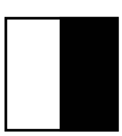
Edge Detection

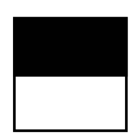
Line Edge Detectors

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

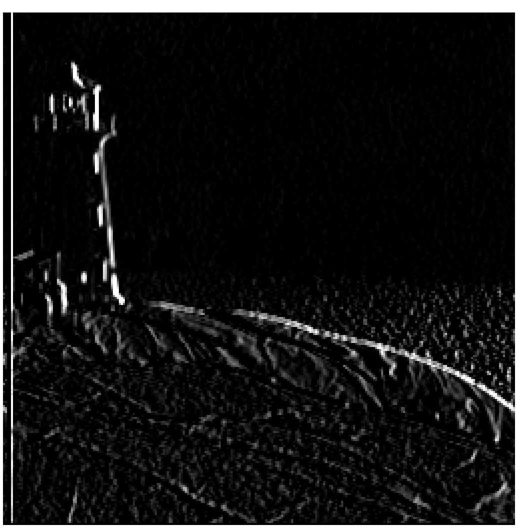
Step Edge Detectors

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





Example



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

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

Edge Detection by Differentiation

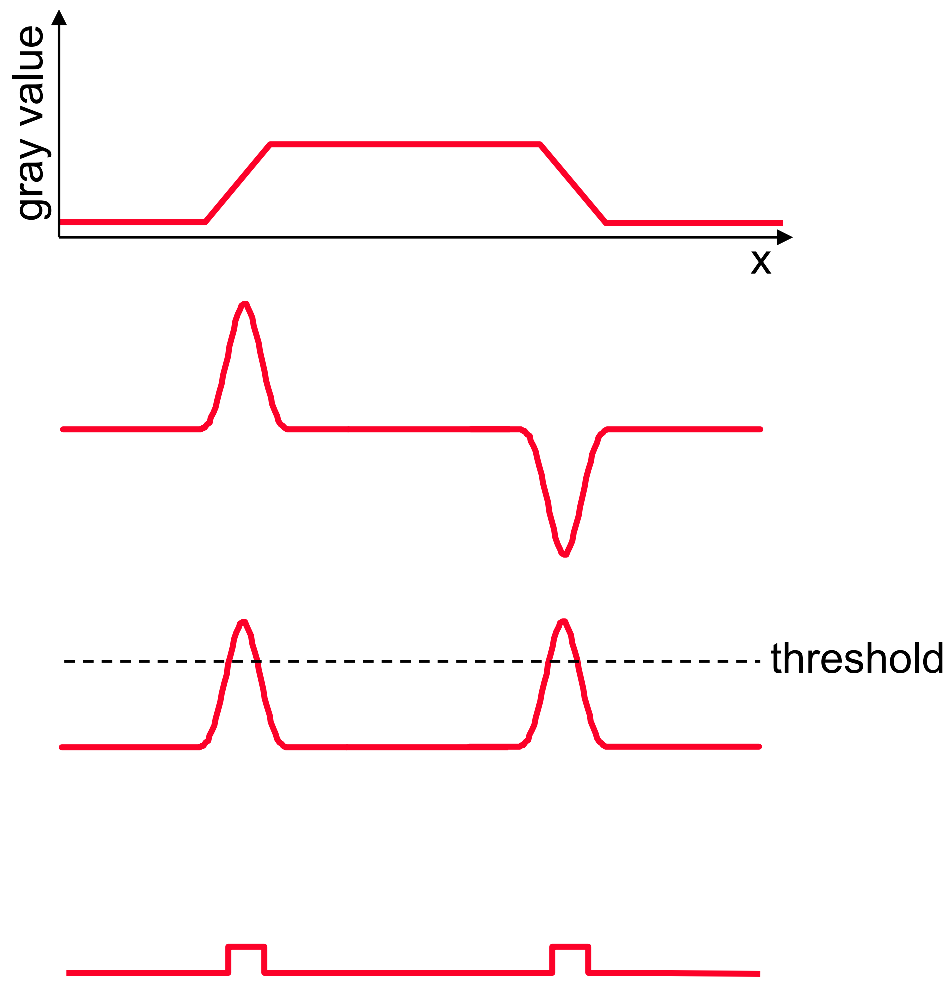
<u>Step Edge detection by differentiation:</u>

1D image f(x)

1st derivative f'(x)

|f'(x)| - threshold

Pixels that passed the threshold are
Edge Pixels

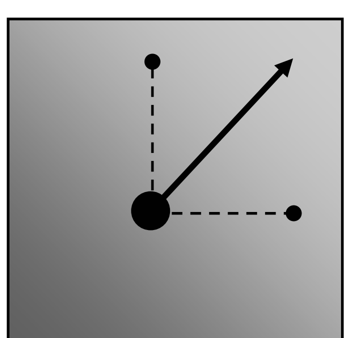


Gradient Edge Detection

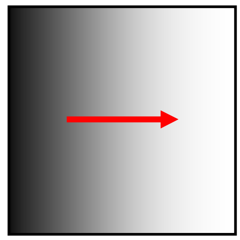
Gradient $\nabla f((x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

Gradient Magnitude $\sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$

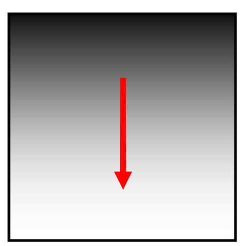
Gradient Direction $tg^{-1}\,(\frac{\partial f}{\partial y} \,/\, \frac{\partial f}{\partial x})$



Gradient Edge - Examples



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



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

Differentiation in Digital Images

horizontal - differentiation approximation:

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

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

vertical - differentiation approximation:

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

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

Gradient (F_A , F_B)

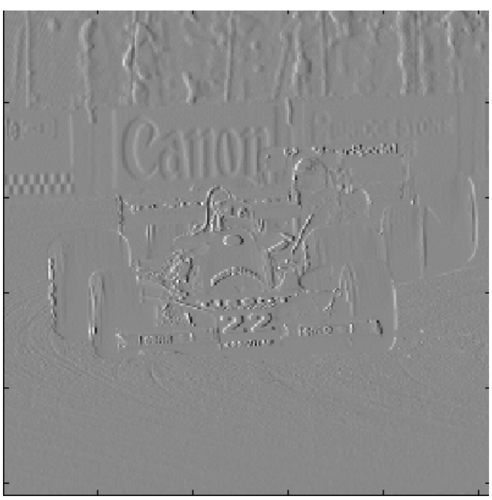
Magnitude $((F_A)^2 + (F_B)^2)^{1/2}$

Approx. Magnitude $|F_A| + |F_B|$

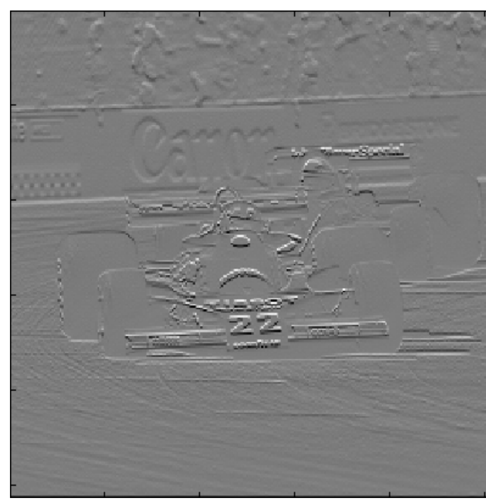
Example Edge

Original

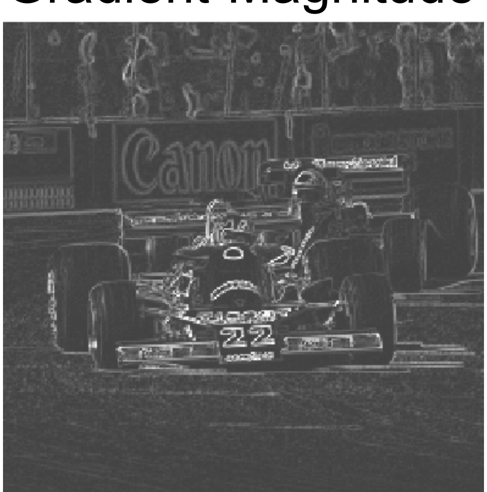
Gradient-X



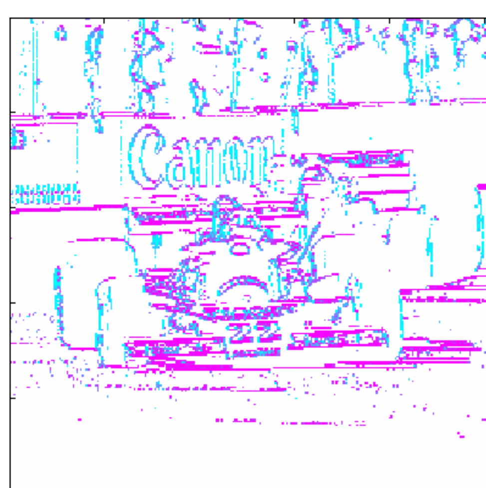
Gradient-Y



Gradient-Magnitude



Gradient-Direction



Roberts Edge Detector

$$F_A = f(x,y) - f(x-1,y-1)$$

$$F_B = f(x-1,y) - f(x,y-1)$$

$$A = \begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix}$$

Roberts and other 2x2 operators are sensitive to noise.

Prewitt Edge Detector

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Smoothed operators

Sobel Edge Detector

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

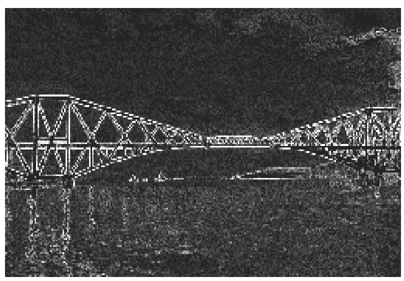
Isotropic Edge Detector

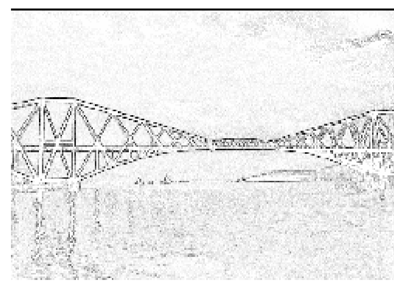
$$A = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & \boxed{0} & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & \boxed{0} & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

Edge Detector - Comparison

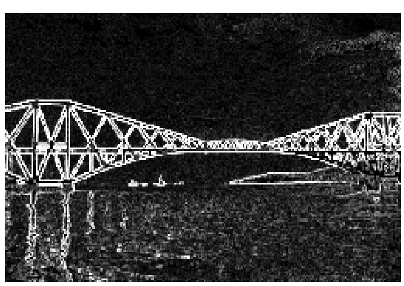
Original

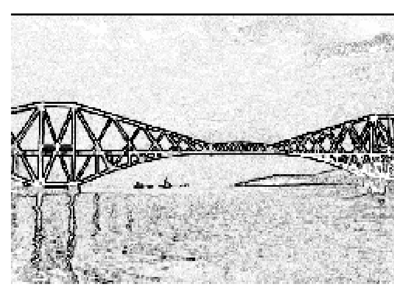
Roberts



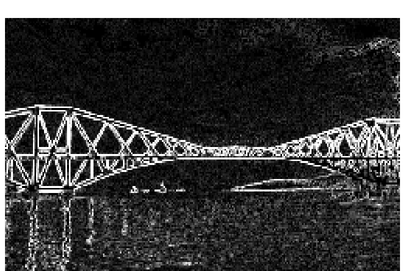


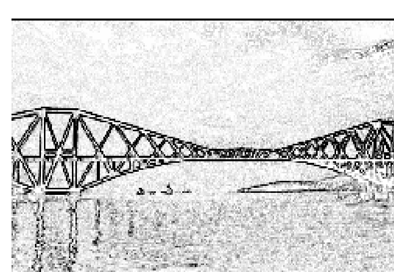
Sobel





Prewitt





Magnitude

Negative Magnitude

Compass Operators

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

↑ N   ↖ NW   ← W   ↙ SW

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

↓ S   ↘ SE   → E   ↗ NE

Given k operators, $g_k(x,y)$ is the image obtained by convolving $f(x,y)$ with the k^{th} operator.

The gradient is defined as:

$$g(x,y) = \max_k g_k(x,y)$$

k defines the edge direction

Various Compass Operators:

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

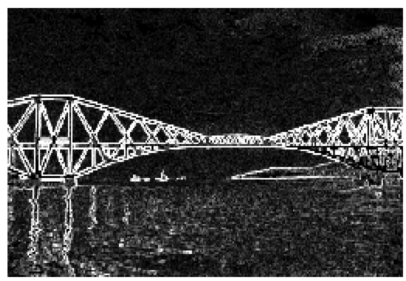
$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & \boxed{0} & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

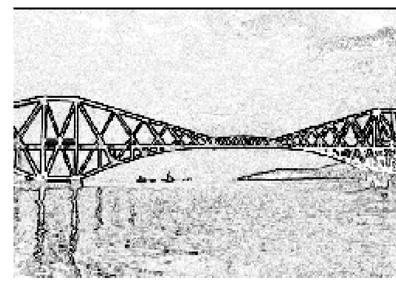
Kirsch Edge Detector

Edge Detector - Comparison

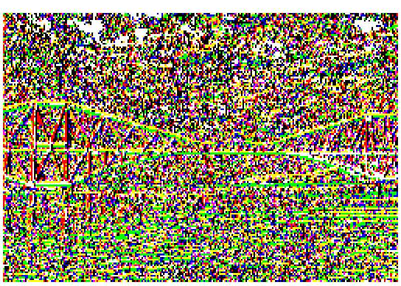
Original

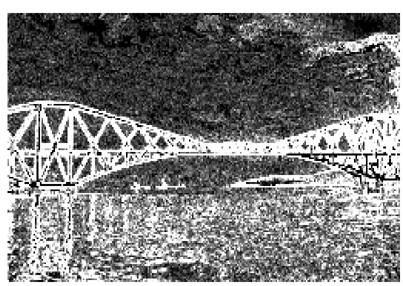
Sobel

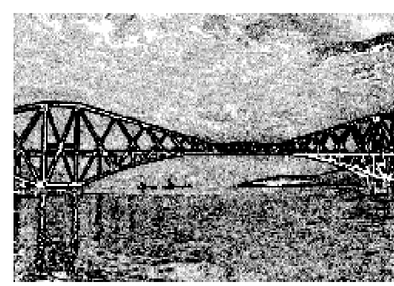




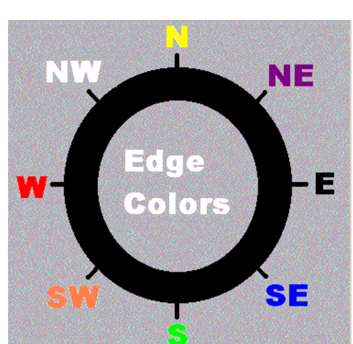
Kirsch (Compass)



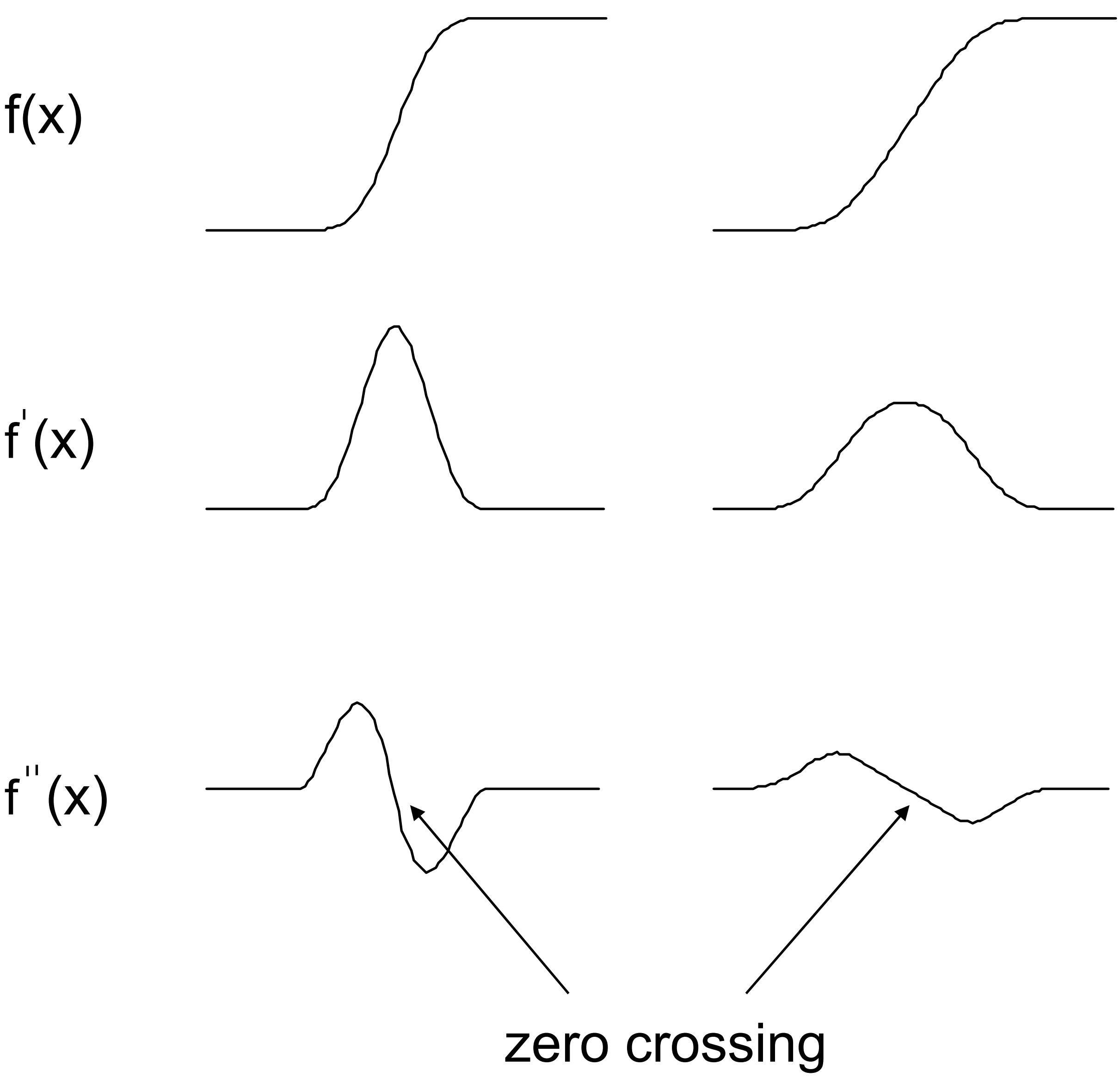




Direction Map | Magnitude | Negative Magnitude



Derivatives



Laplacian Operators

Approximation of second derivative (horizontal):

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

$$= [f(x+1,y) - f(x,y)] - [f(x,y) - f(x-1,y)]$$

$$= f(x+1,y) - 2f(x,y) + f(x-1,y)$$

convolution with: $[\,1 \;\; -2 \;\; 1\,]$

Approximation of second derivative (vertical):

convolution with: $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

Laplacian Operator

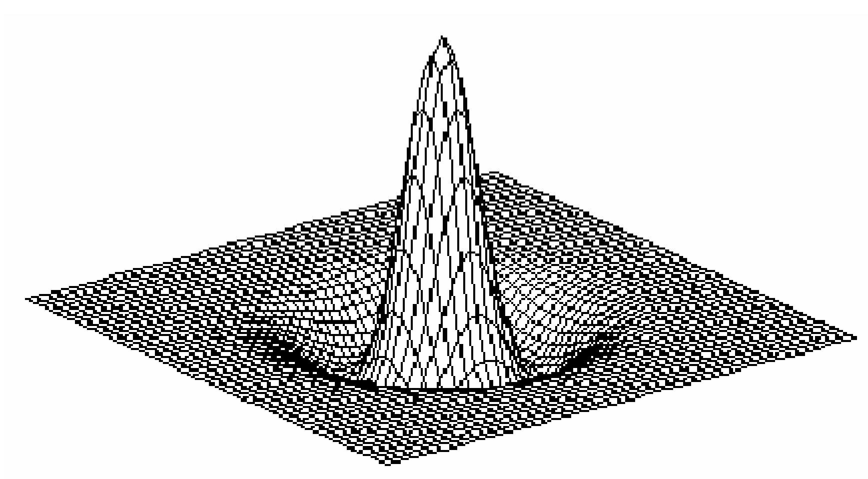
$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$$

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

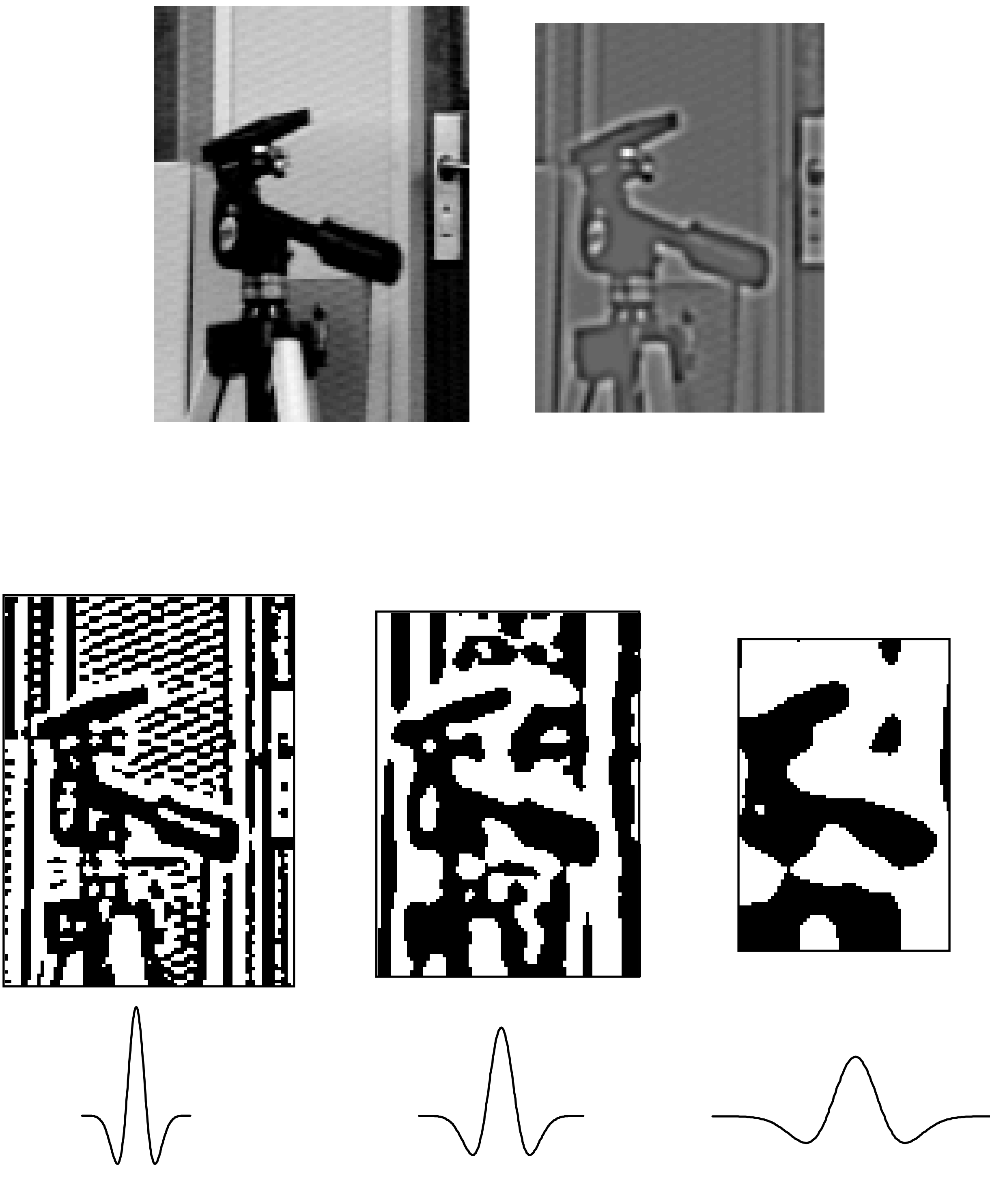
Variations on Laplace Operators:

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

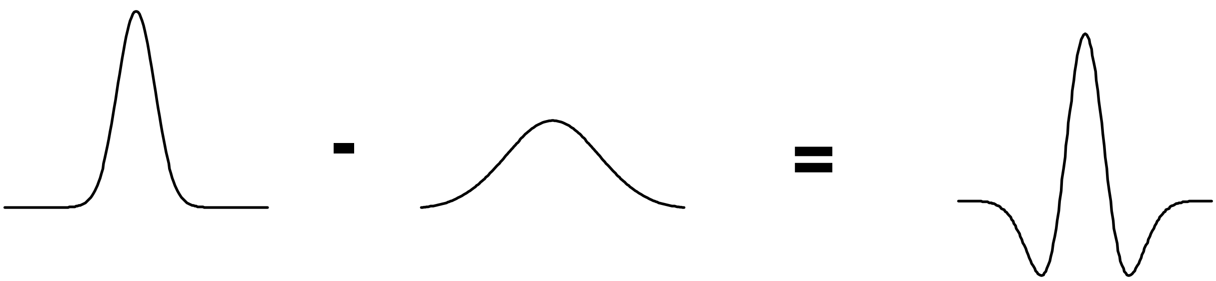
All are approximations of:



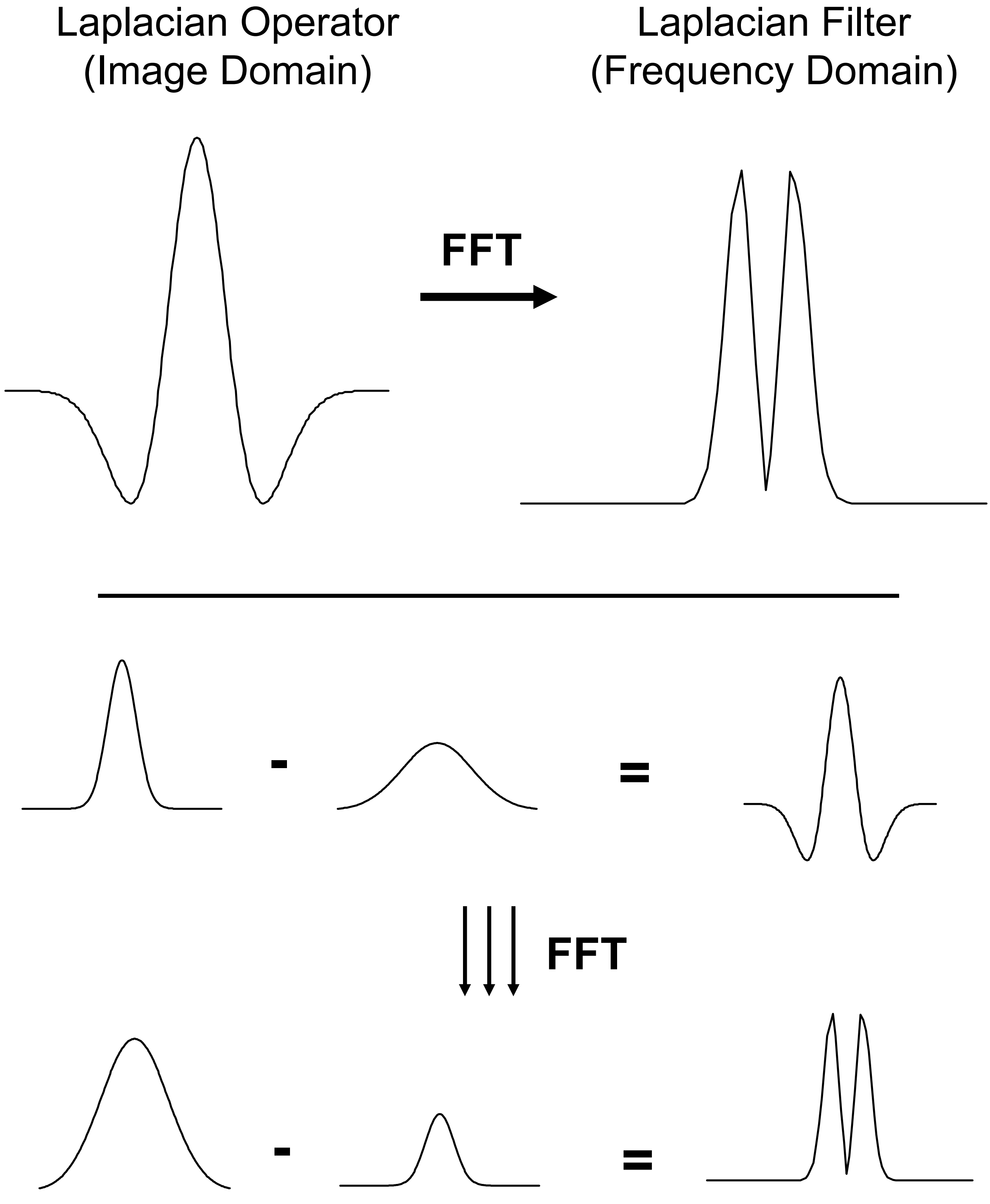
Example of Laplacian Edge Detection



Laplacian ~ Difference of gaussians

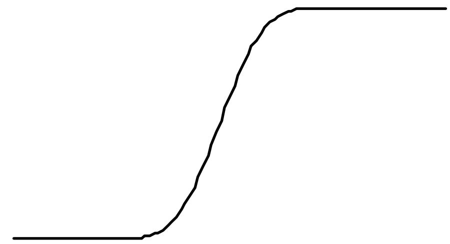


DOG = Difference of Gaussians

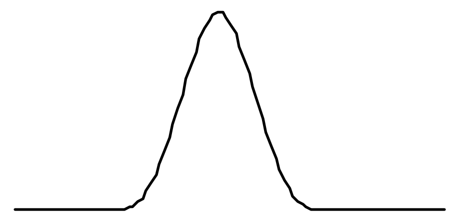

Laplacian Operator
(Image Domain)
Laplacian Filter
(Frequency Domain)
FFT
-
=
FFT
-
=

Enhancement Using the Laplacian Operator

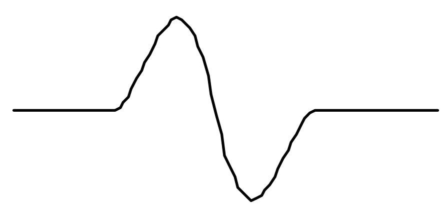
Edge Image $f(x)$



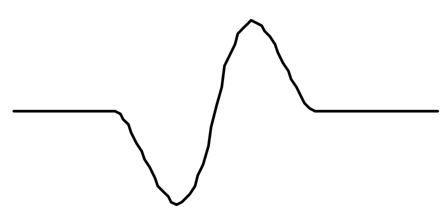
1^{st} derivative $\frac{\partial f}{\partial x}$



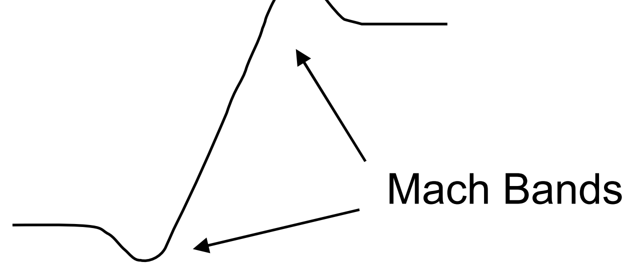
2^{nd} derivative (Laplacian) $\frac{\partial^2 f}{\partial x^2}$



$-\frac{\partial^2 f}{\partial x^2}$

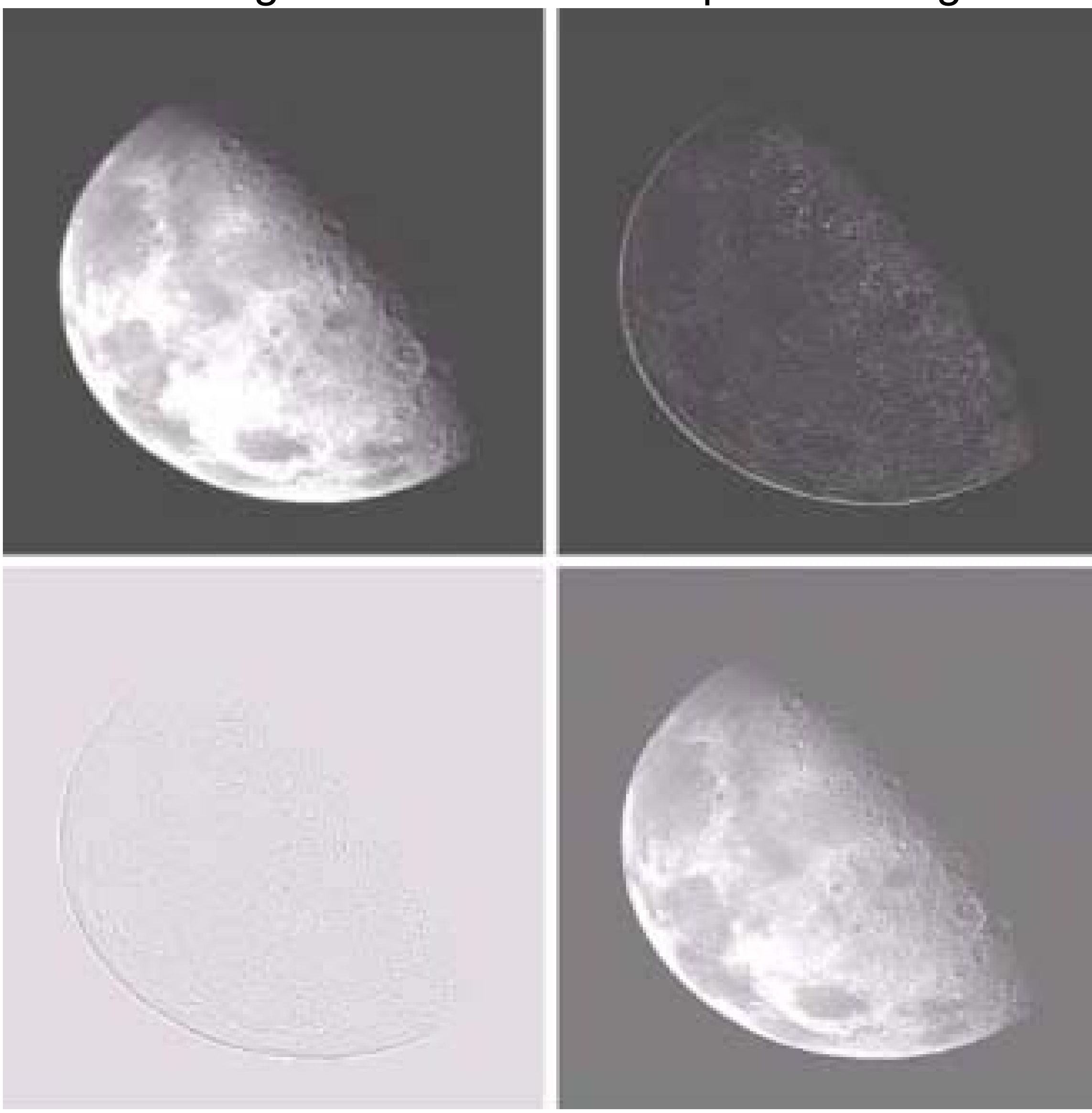


$f(x) - \frac{\partial^2 f}{\partial x^2}$



Examples of enhanced images

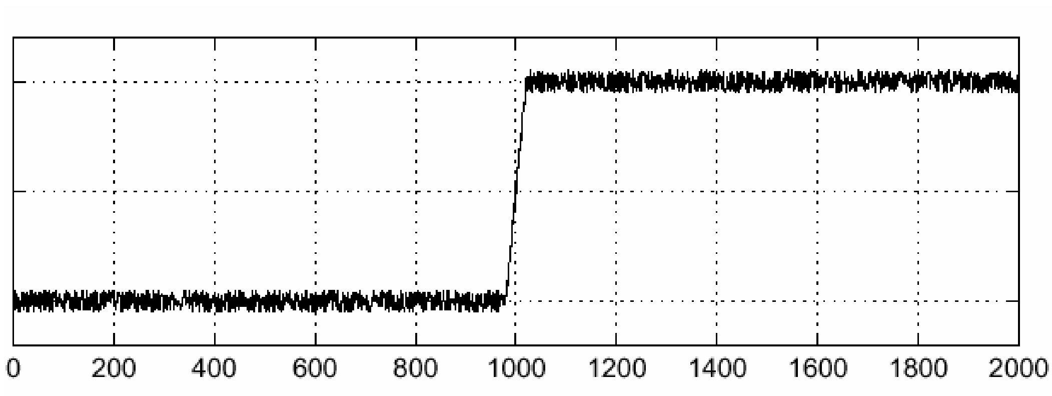
Original | Laplacian Image



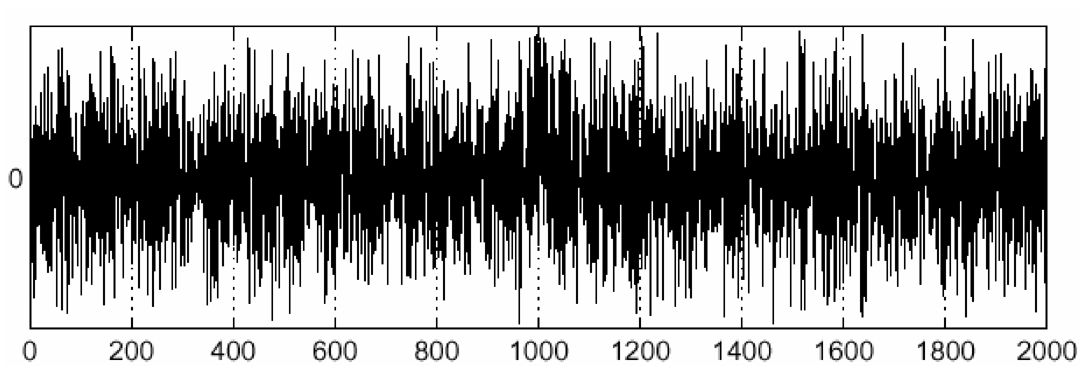
Scaled Laplacian Image | Enhanced Image

Edge Detection - Noise Issues

$f(x)$



$\frac{d}{dx} f(x)$



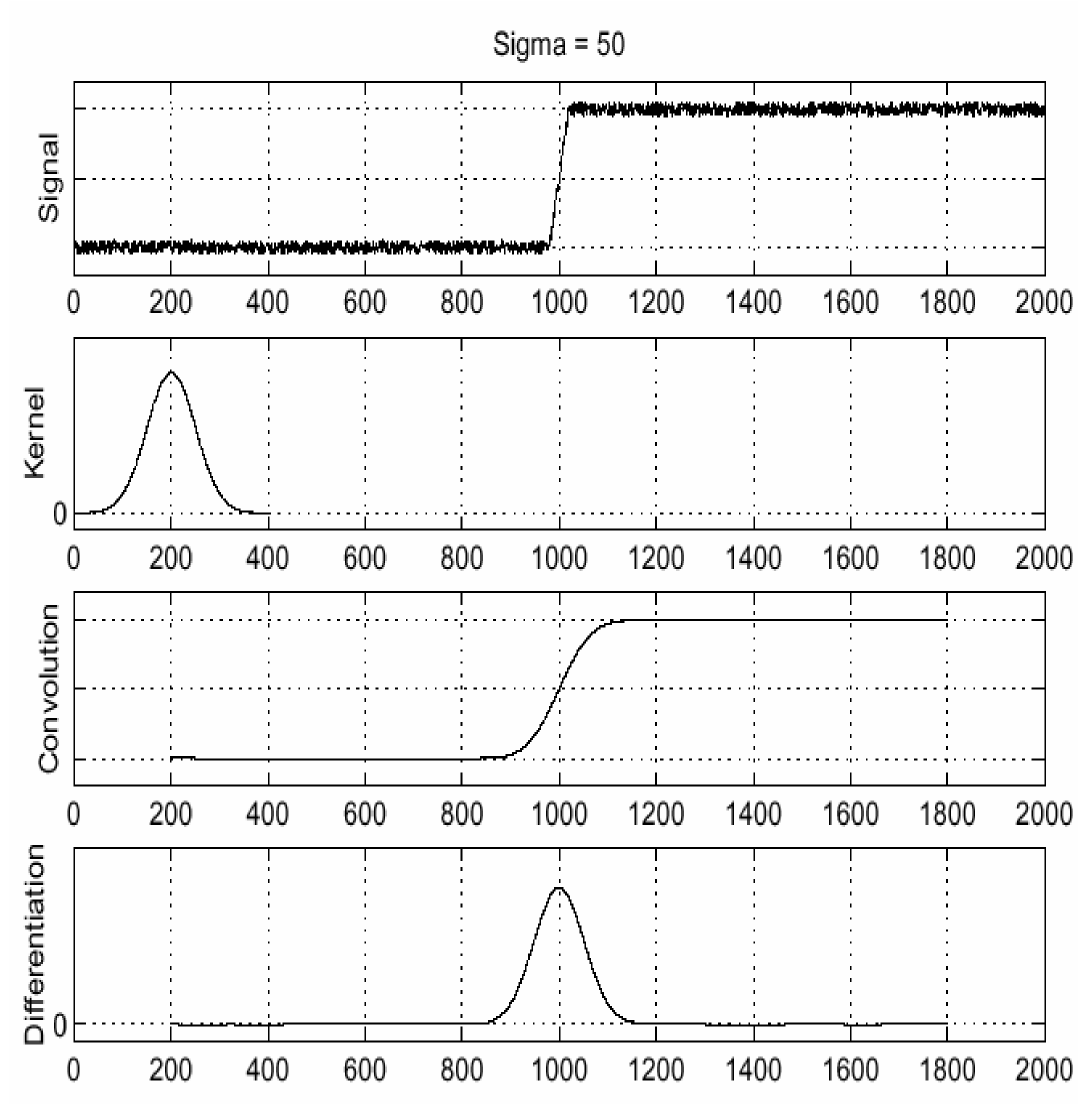
Edge Detection - Noise Issues

f

g

$g * f$

$\frac{\partial}{\partial x}(g * f)$



Peaks in $\frac{\partial}{\partial x}(g * f)$ mark the edge

Edge Detection - Noise Issues

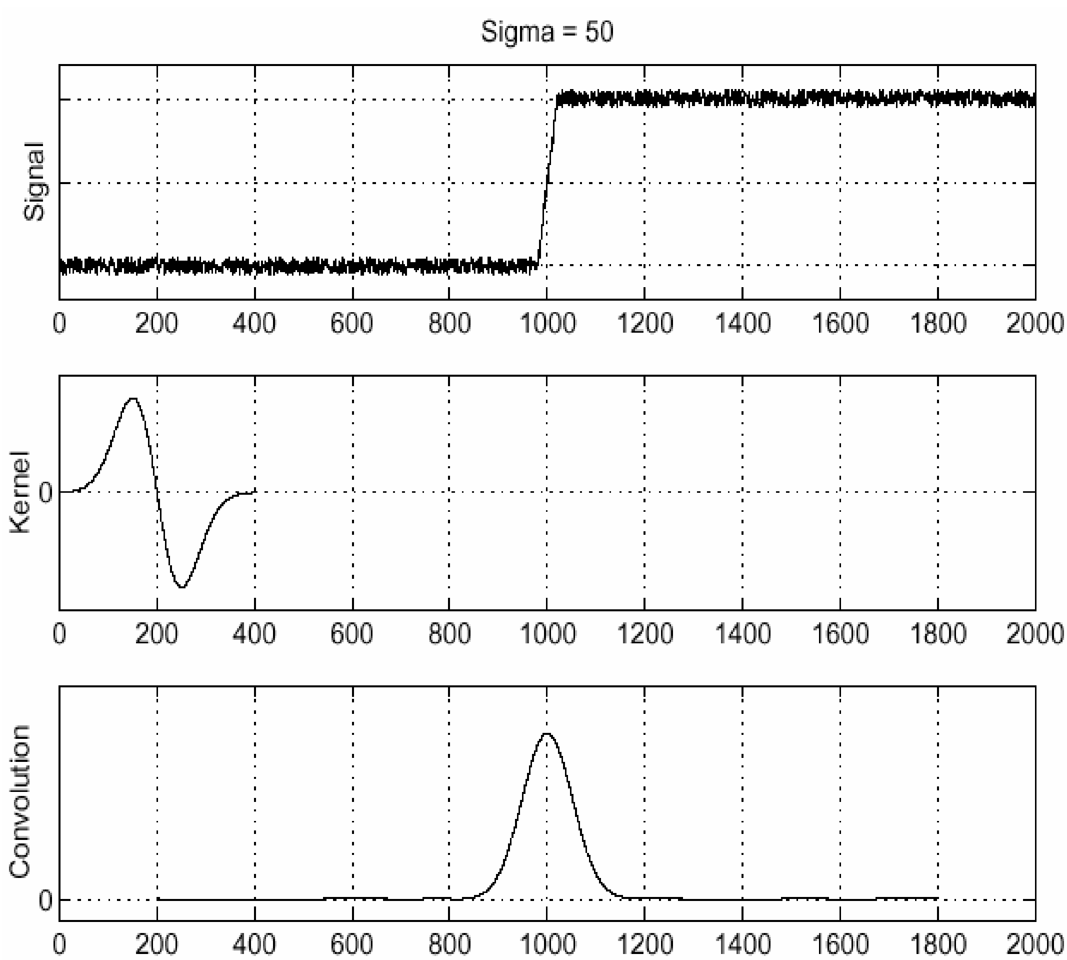
From the convolution theorem we have:

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x}h\right) * f$$

f

$\frac{\partial}{\partial x}h$

$\left(\frac{\partial}{\partial x}h\right) * f$

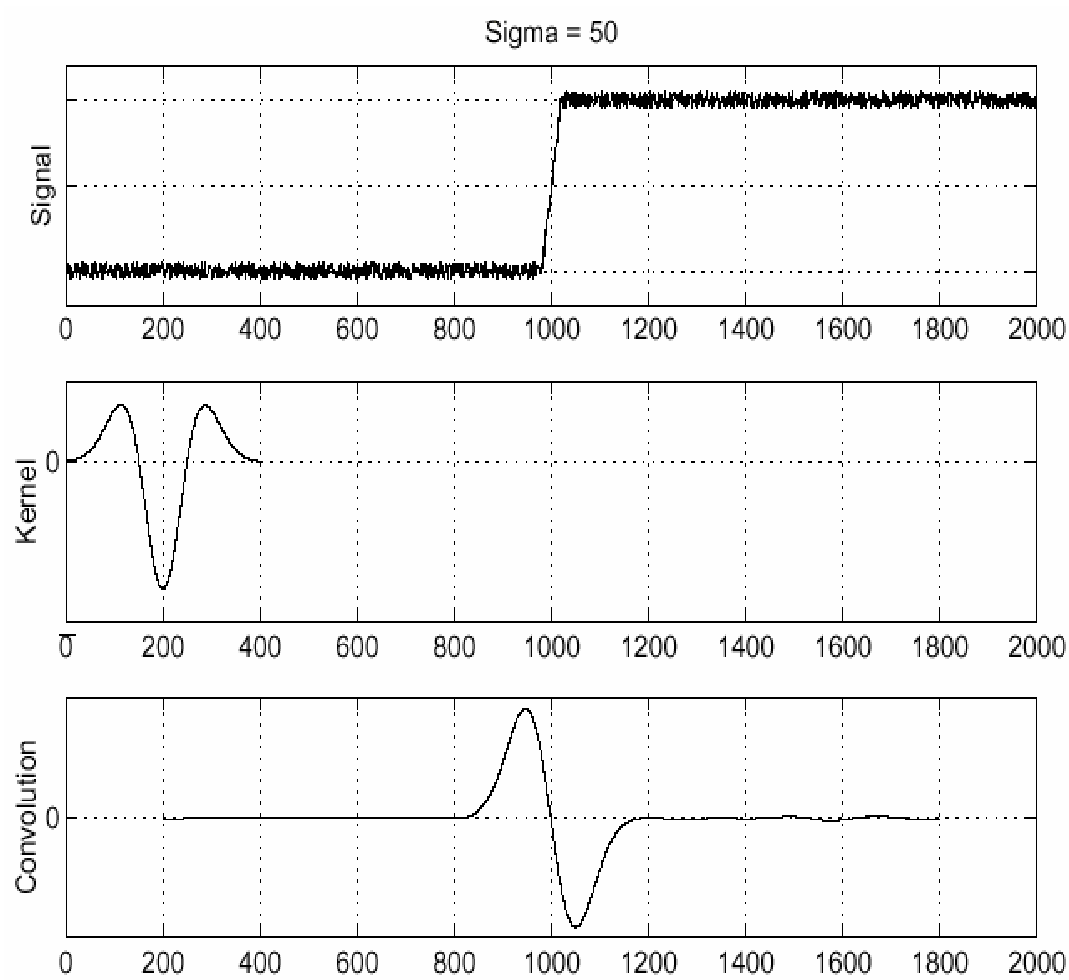


Edge Detection - Noise Issues

f

$$\frac{\partial^2}{\partial x^2} g$$

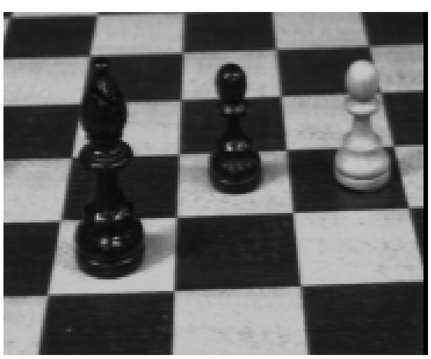
$$\left(\frac{\partial^2}{\partial x^2} g \right) * f$$

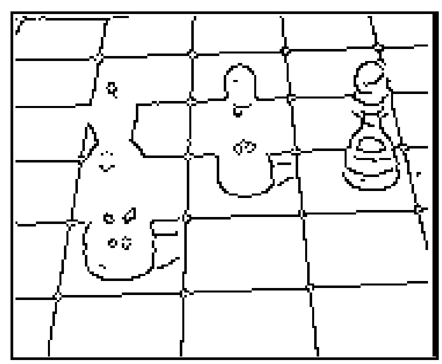


Zero Crossings in $\frac{\partial^2}{\partial x^2}(g * f)$ mark the edge

Canny Edge Detector

1) Convolve image with derivative of a Gaussian
2) Perform NonMaximum suppression.
3) Perform Hysteresis Thresholding.





Canny Edge Detector - Step 1

Convolve image with derivative of a Gaussian

$$f(x,y) * G'(x,y) \cong f(x,y) * G'_x(x,y) + f(x,y) * G'_y(x,y)$$

$$F_a = f(x,y) * G_x(x,y) * D_x(x,y)$$

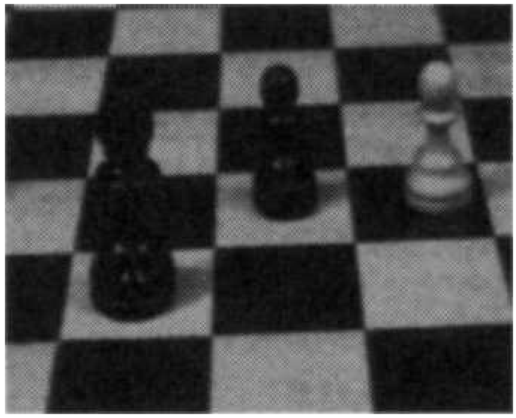
$$= f(x,y) * \begin{bmatrix} 1 & 2 & 4 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$F_b = f(x,y) * G_y(x,y) * D_y(x,y)$$

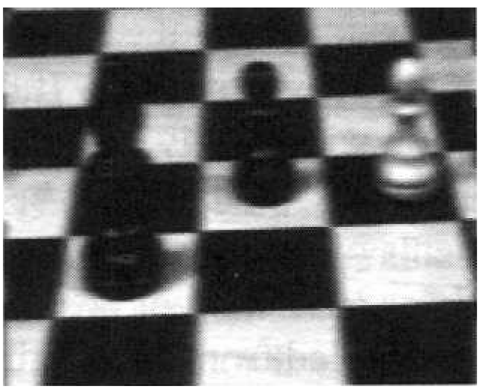
$$= f(x,y) * \begin{bmatrix} 1 \\ 2 \\ 4 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Magnitude $((F_A)^2 + (F_B)^2)^{1/2}$

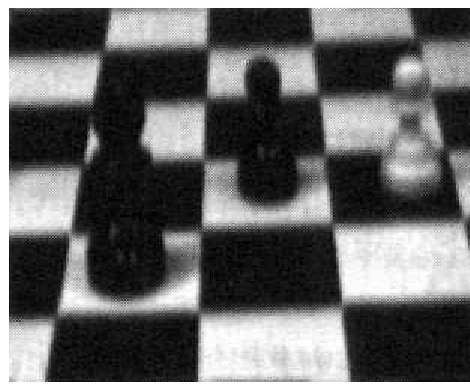
Canny Edge Detector - Step 1



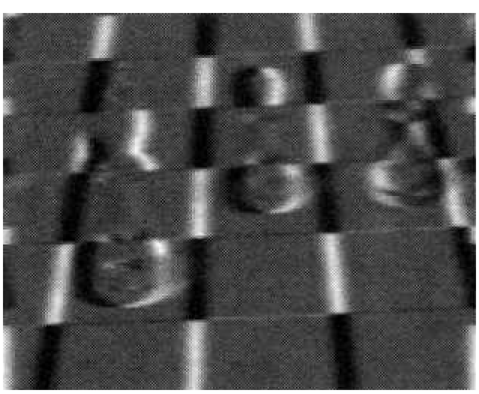
Original



$f * G_x$

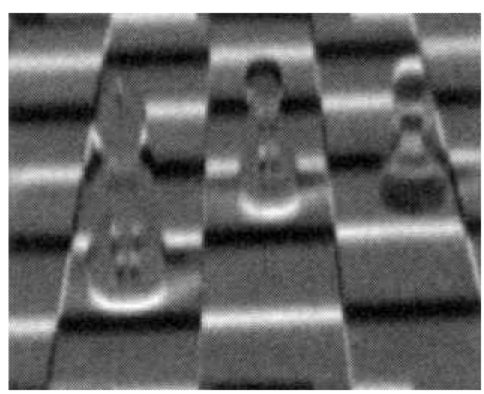


$f * G_y$



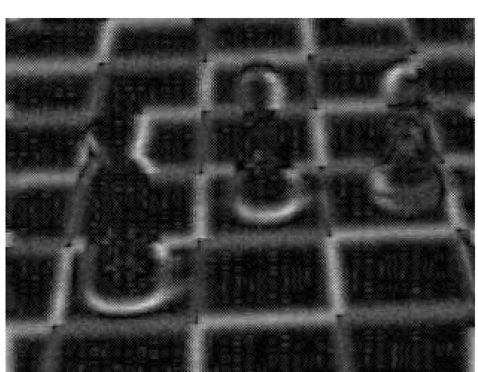
$F_a =$

$f*G_x*D_x$



$F_b =$

$f*G_y*D_y$



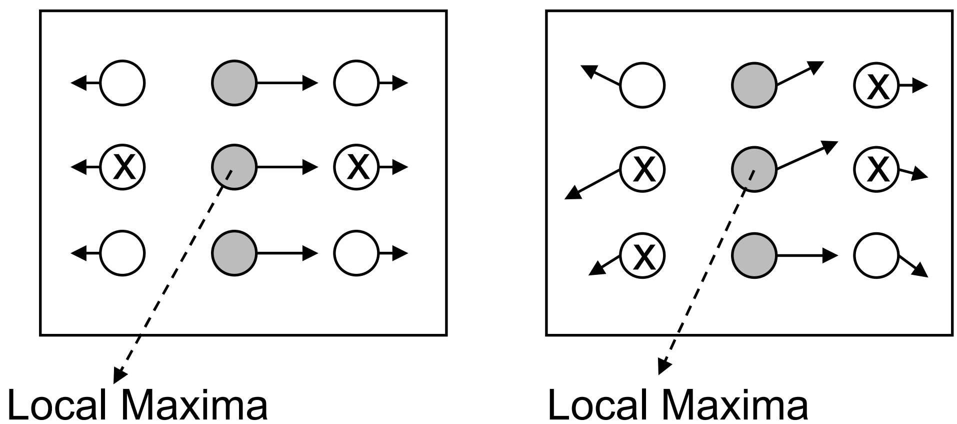
Magnitude

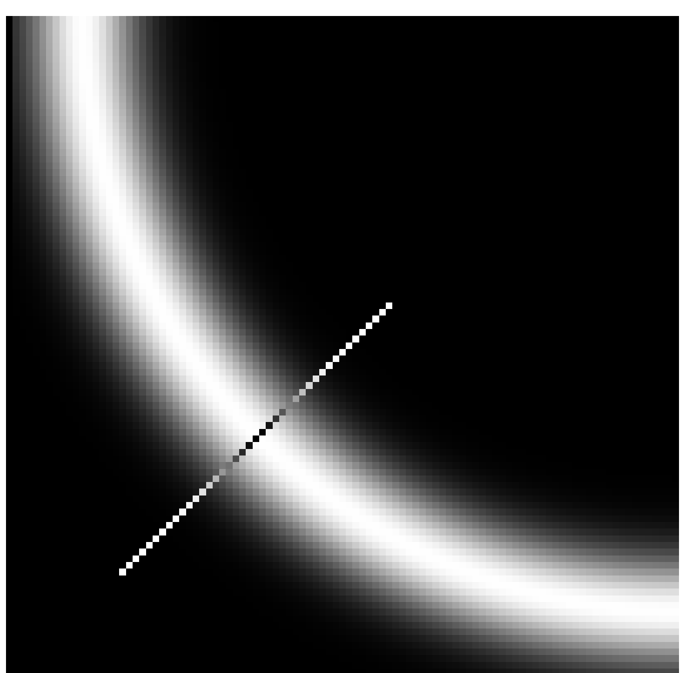
$((F_A)^2 + (F_B)^2)^{1/2}$

Canny Edge Detector - Step 2

NonMaximum suppression.

Remove edges that are NOT local Maxima in the gradient direction.

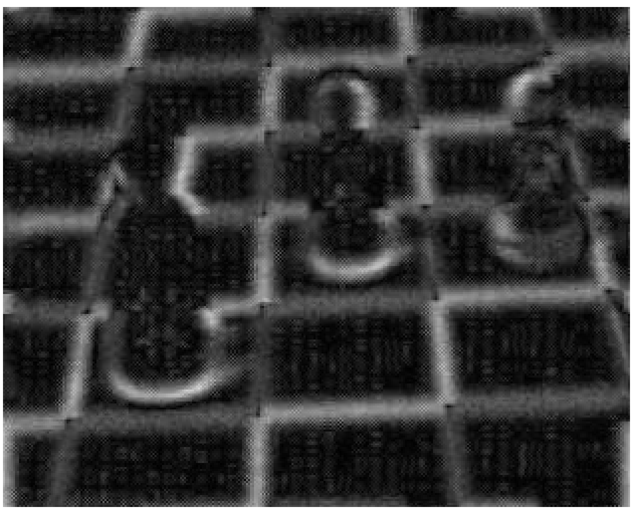




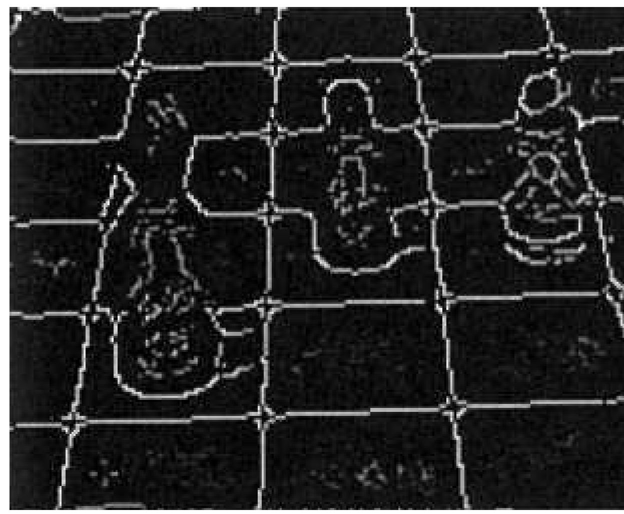
Canny Edge Detector - Step 2

NonMaximum Suppression.

Magnitude

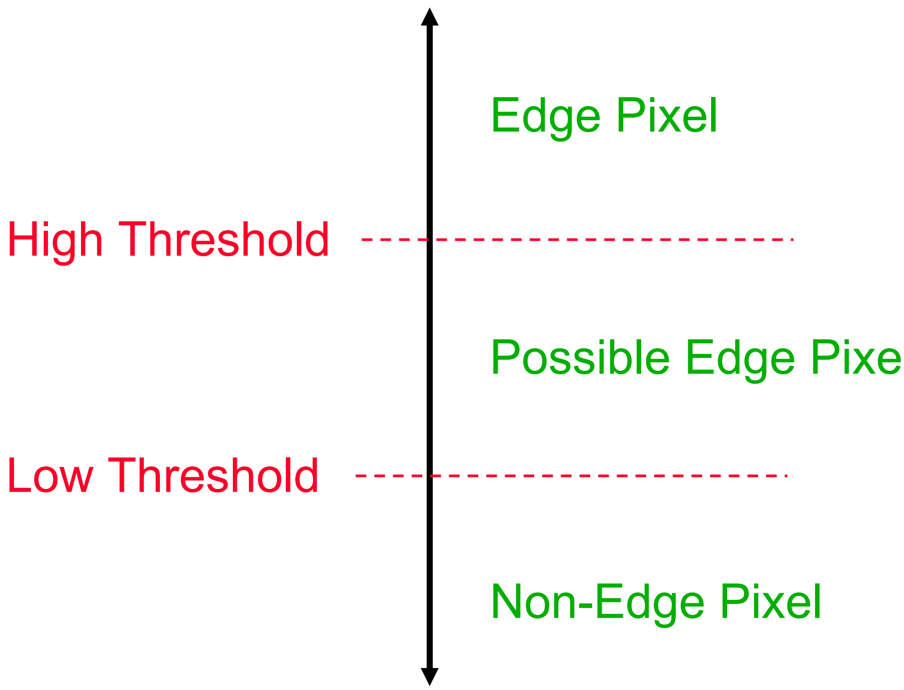


NonMaximum Supression

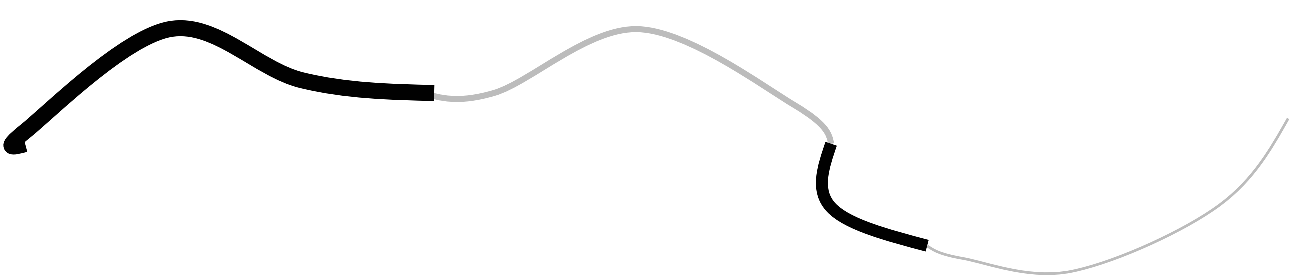


Canny Edge Detector - Step 3

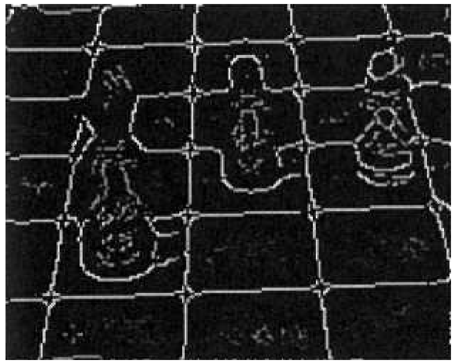
Hysteresis Thresholding.



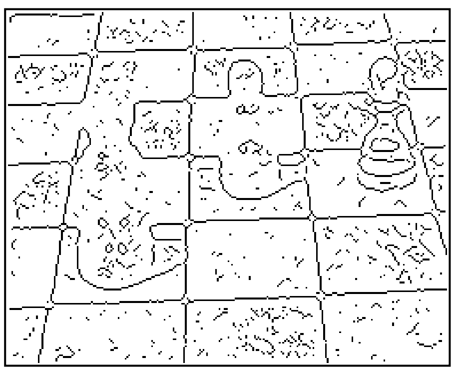
Include possible edge pixels that are adjacent to edge pixels.



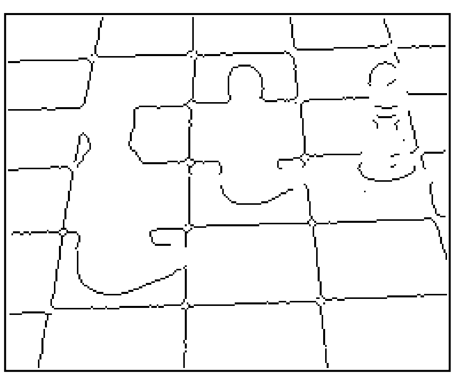
Canny Edge Detector - Step 3



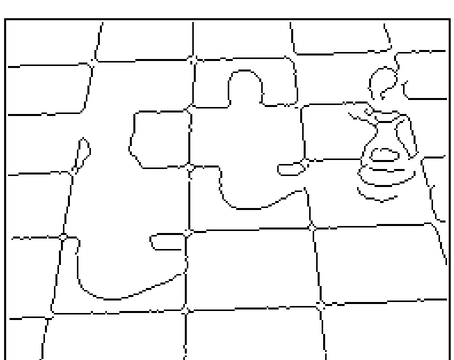
Magnitude
(NonMax
Suppressed)



Low
Threshold



High
Threshold

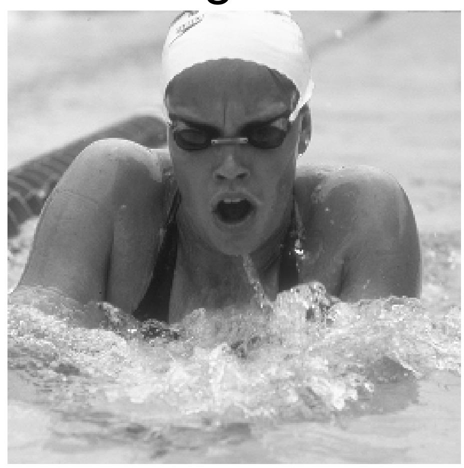


Hysteresis
Threshold

Effect of Scale

Effect of σ (Gaussian kernel size)

original

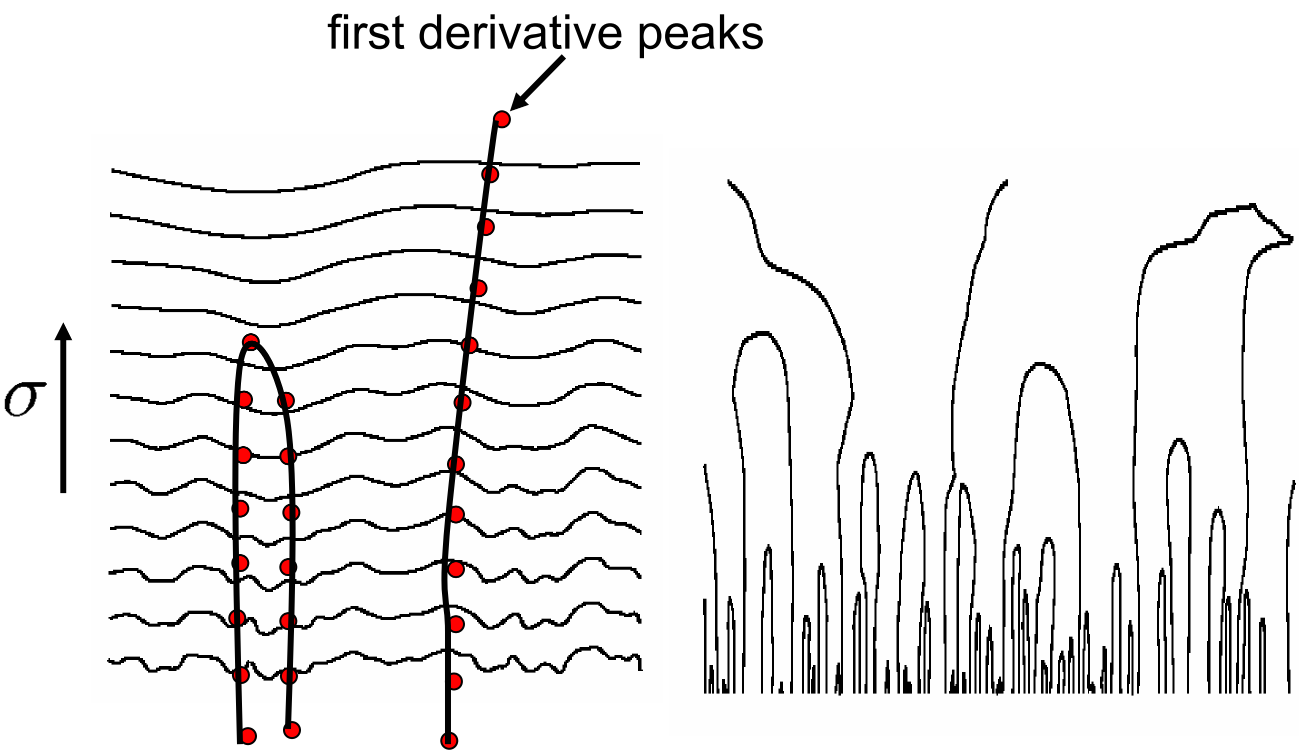


Canny with $\sigma = 1$

Canny with $\sigma = 2$

Scale Space

Witkin 83

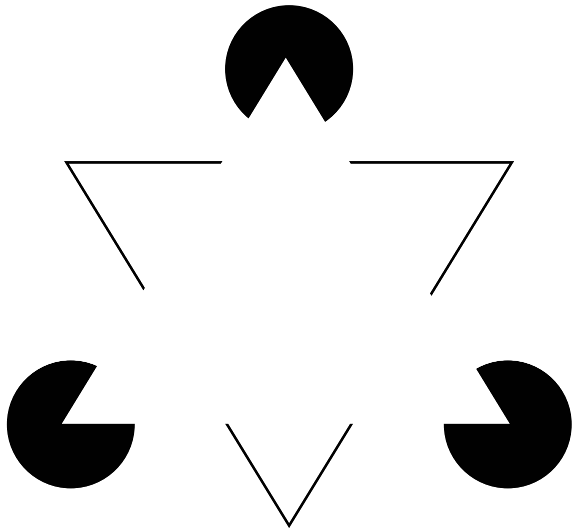


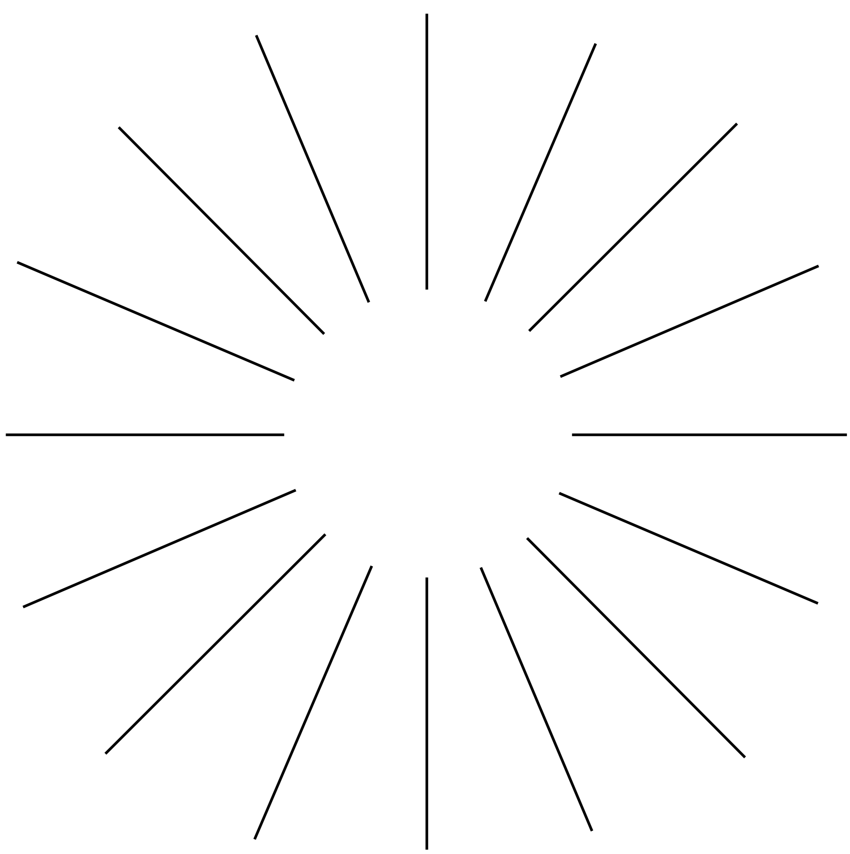
Gaussian filtered signal

Properties of Scale Space:

- Position of edges may change with scale.
- Edges may merge with increase in scale.
- Edges do NOT split with increase in scale.

Edge Completion





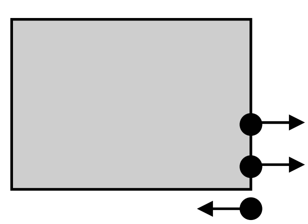
Edge Linking

(x,y) is an edge pixel.
Search for neighboring edge pixels that are "similar".

Similarity:

Similarity in Edge Orientation
Similarity in Edge strength (Gradient Amplitude)

Perform Edge Following along similar edge pixels.
(as in Contour Following in binary images).

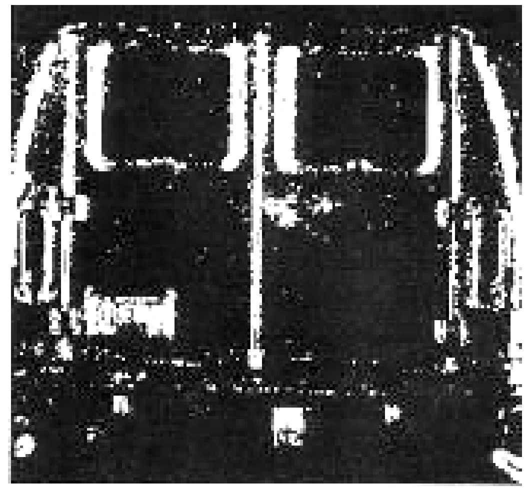


Examples of edge linking

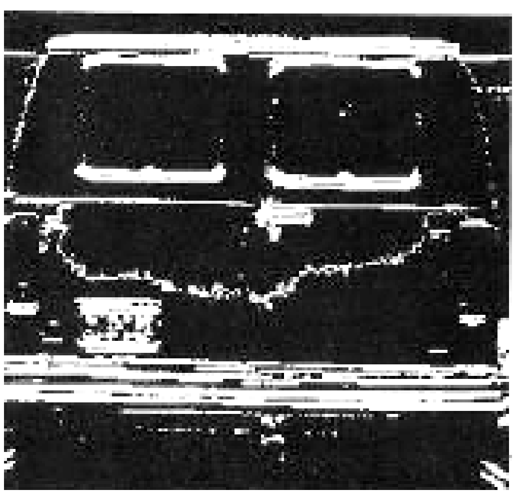
Original

(a)

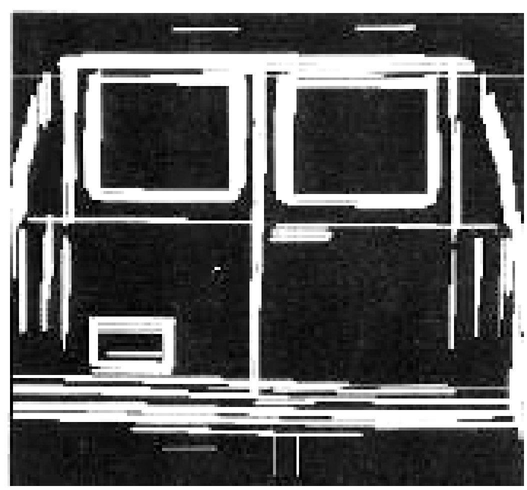
Sobel Vertical



(b)



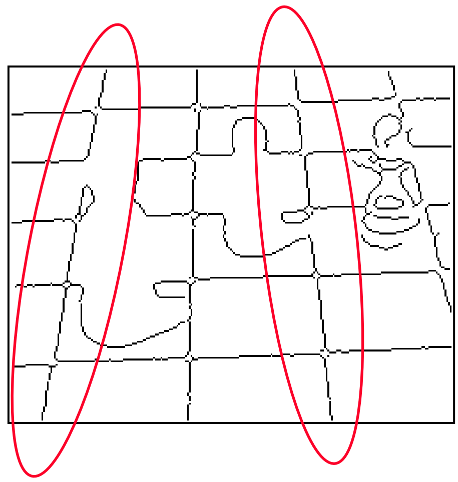
Sobel Horizontal



Linked Edges

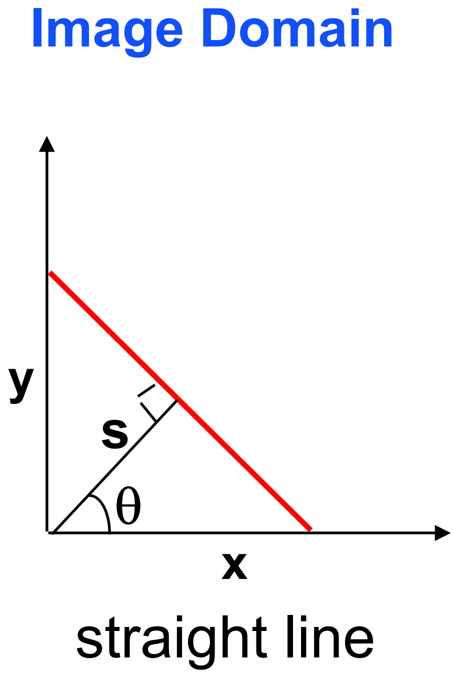
Edge Points linked: Gradient Value > 25
Gradient direction within 15%

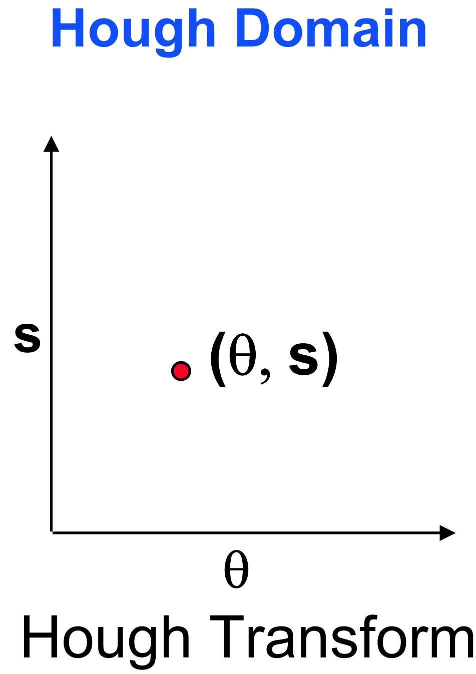
Problem:
Edges are not lines even when linked.



Edge pixels are not ellipses even when linked.

Hough Transform





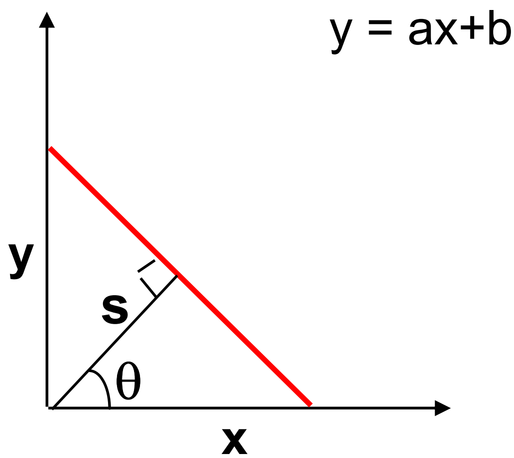
$$y = ax+b$$

↓

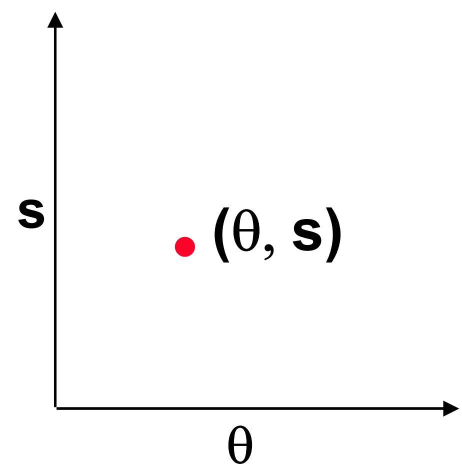
$$s = x\cos(\theta)+y\sin(\theta)$$

Hough Transform

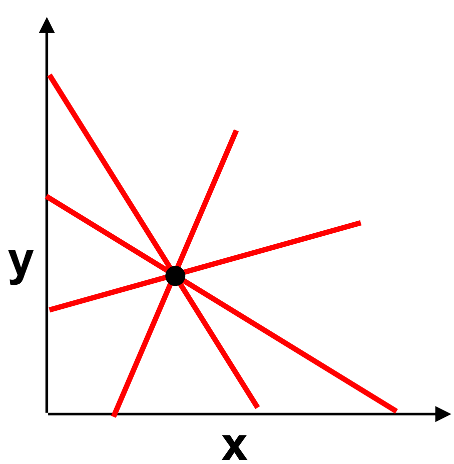
Image Domain | **Hough Domain**

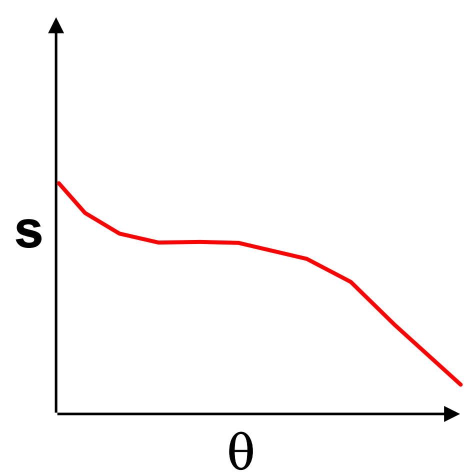


straight line



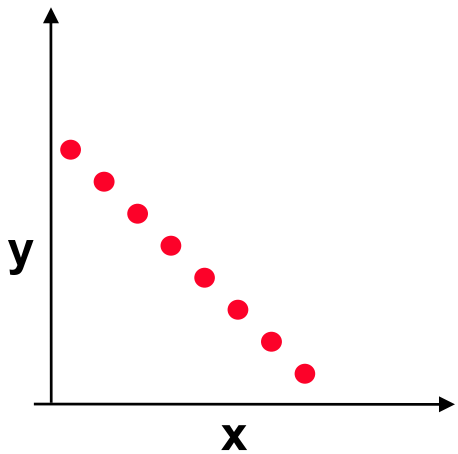
Hough Transform

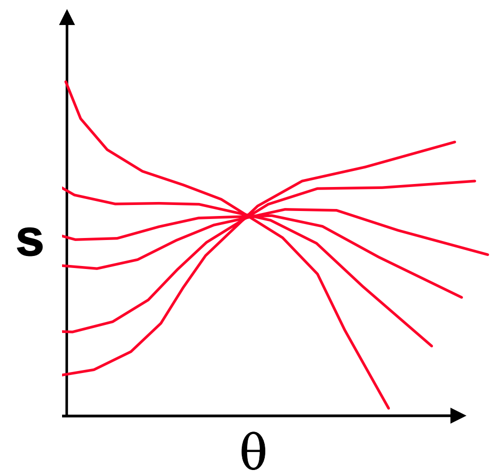




single point = many possible lines

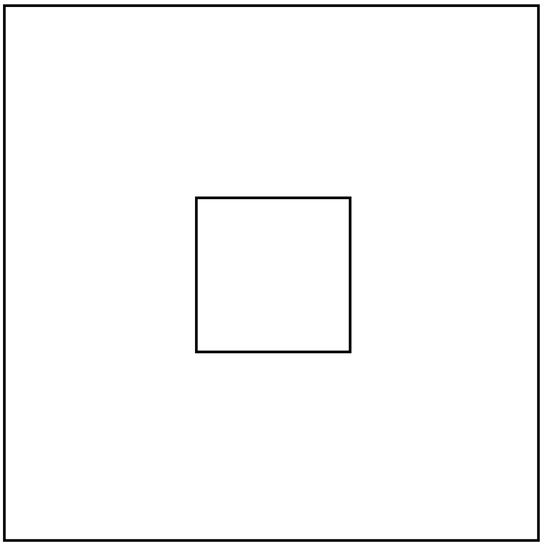
Hough Transform



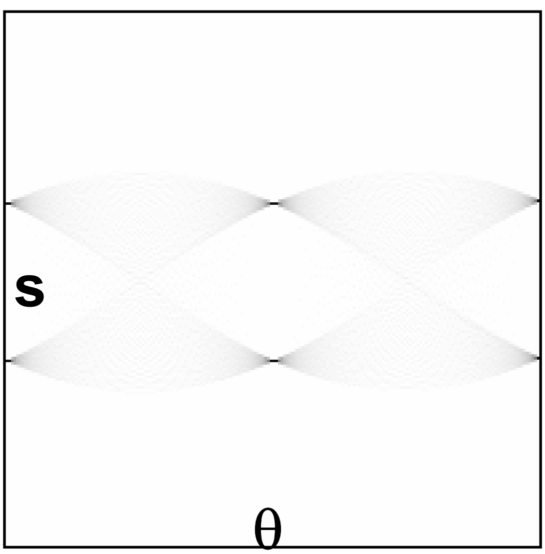


many points on a line =
many lines in the Hough transform space which intersect at 1 point.

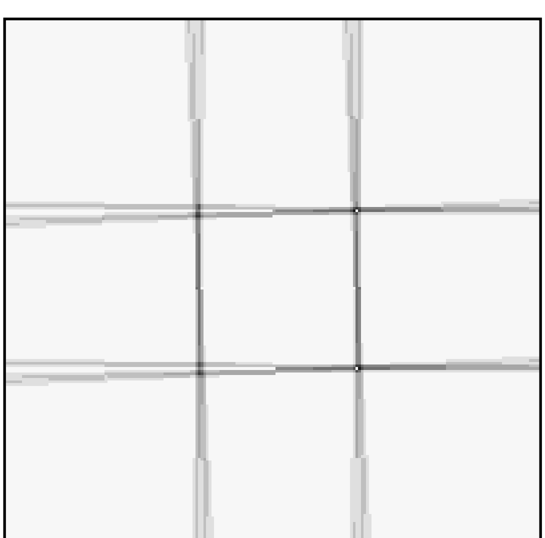
Hough Transform Example



Original square image



Hough Transform (s,θ) space

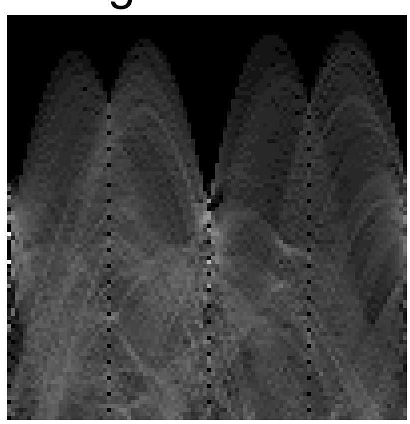


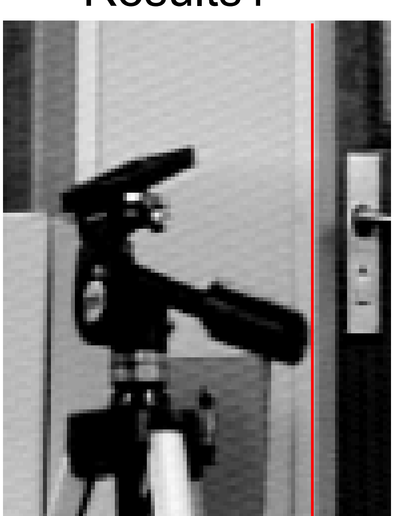
Reconstructed line segments

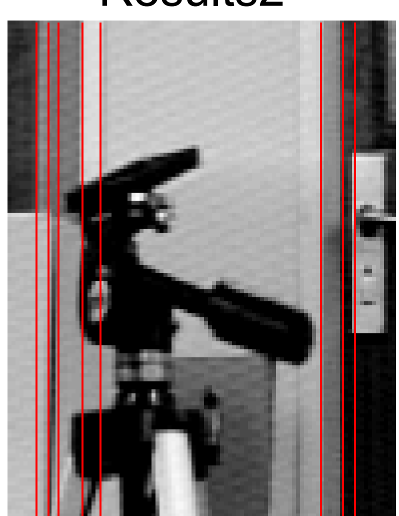
Hough Transform Example

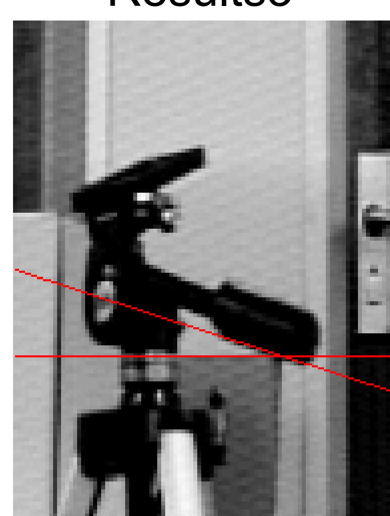

Original


Edges

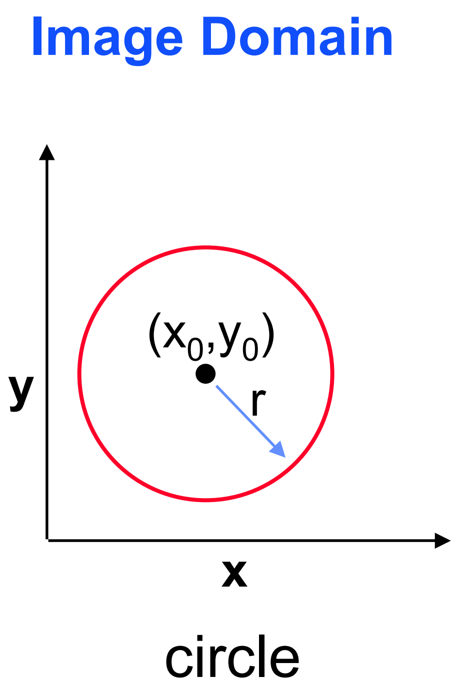

Hough Transform

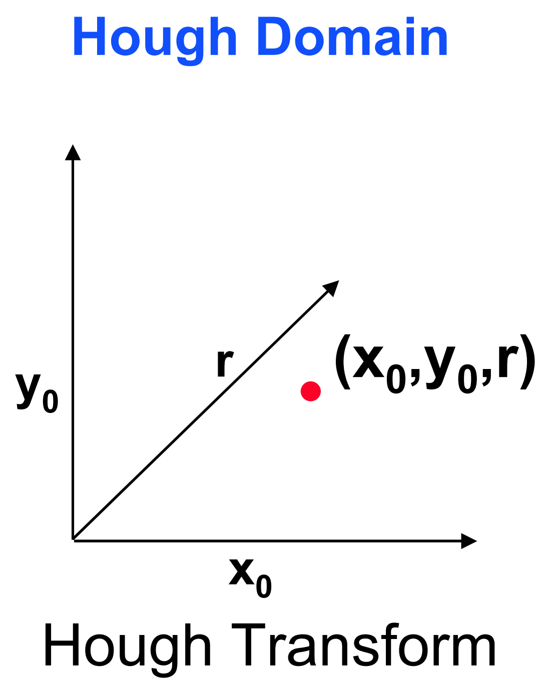

Results1


Results2


Results3

Hough Transform for Circles

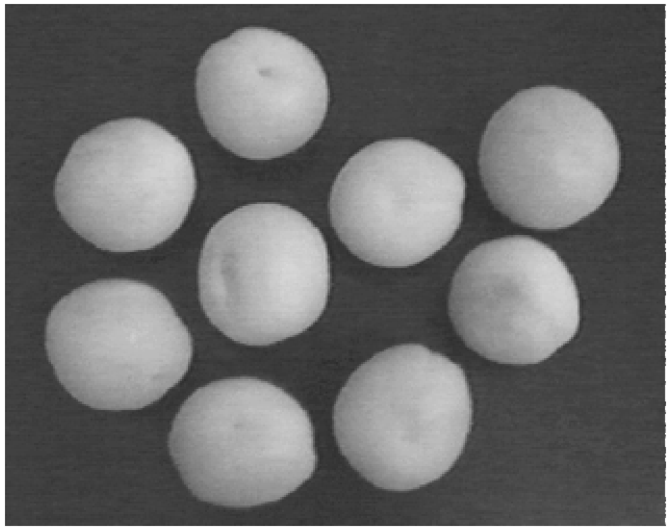




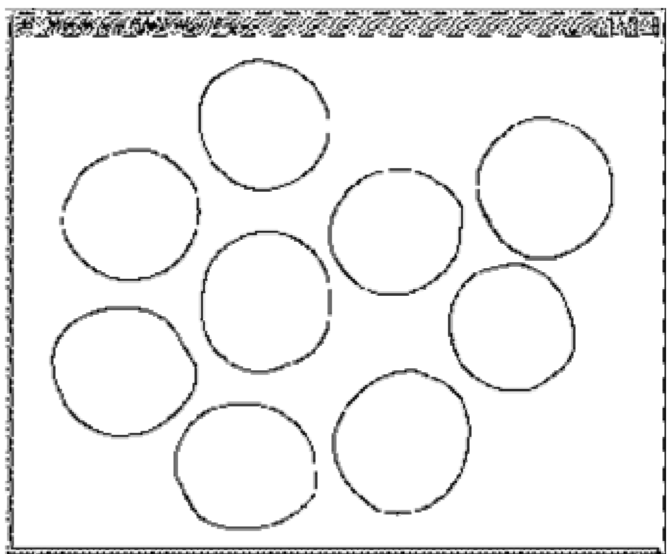
$$r^2 = (x-x_0)^2+(y-y_0)^2$$

Hough Transform Example

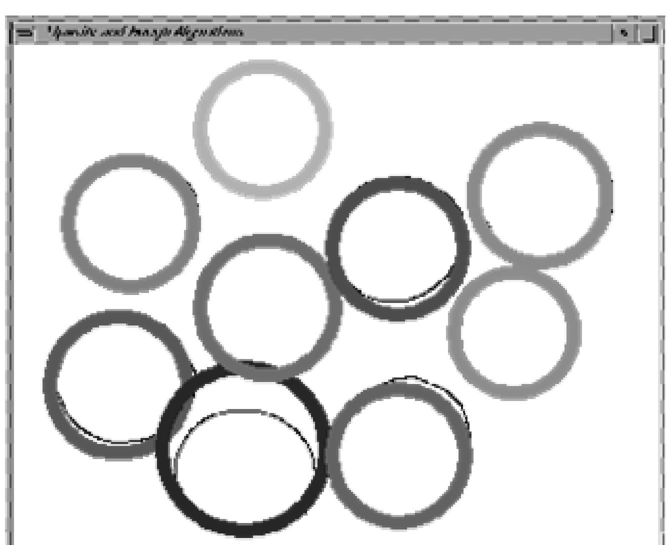
Original



Edges

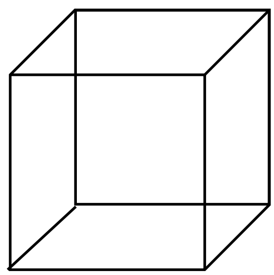


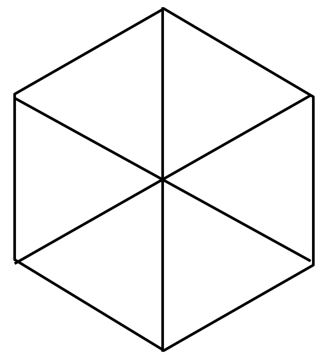
Result

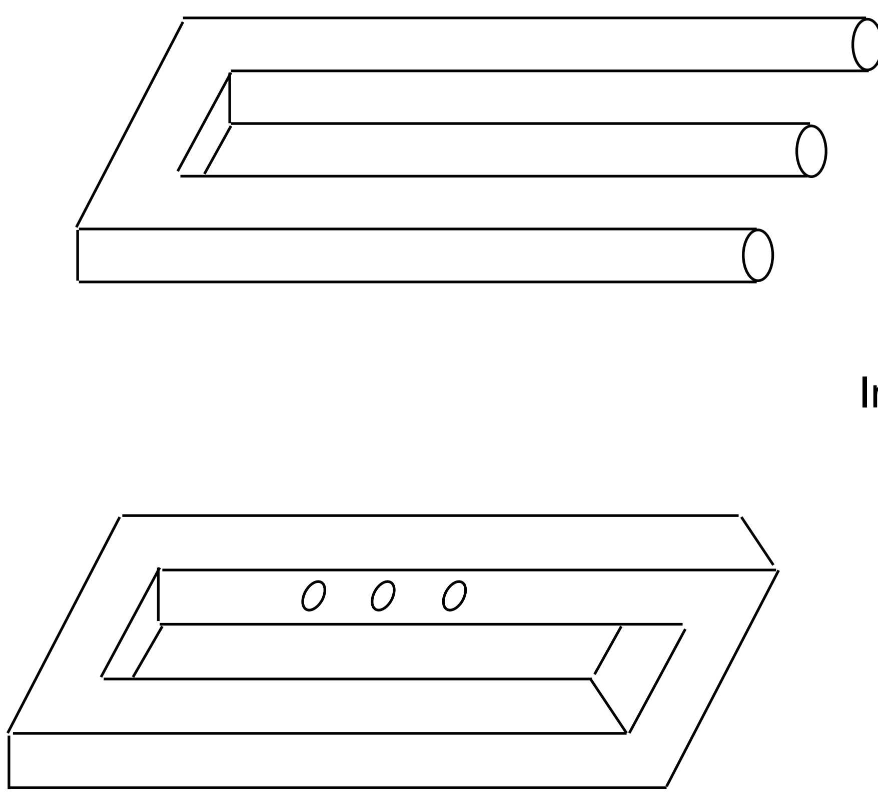


Nice demo: http://www.markschulze.net/java/hough/index.html

3D Perception - Depth Perception







Impossible Figures
(Escher)