Edge Detection

CS 4391 Introduction Computer Vision

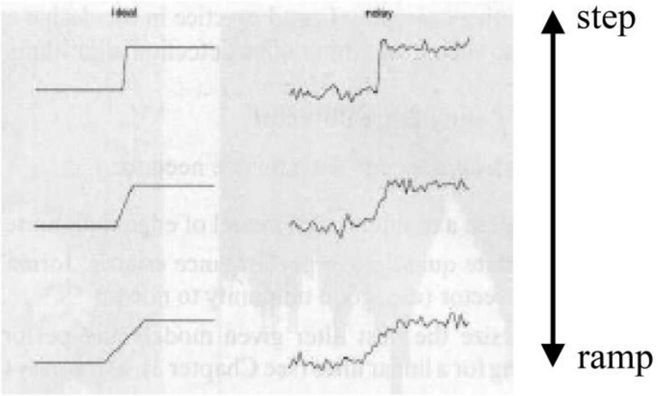
Professor Yu Xiang

The University of Texas at Dallas

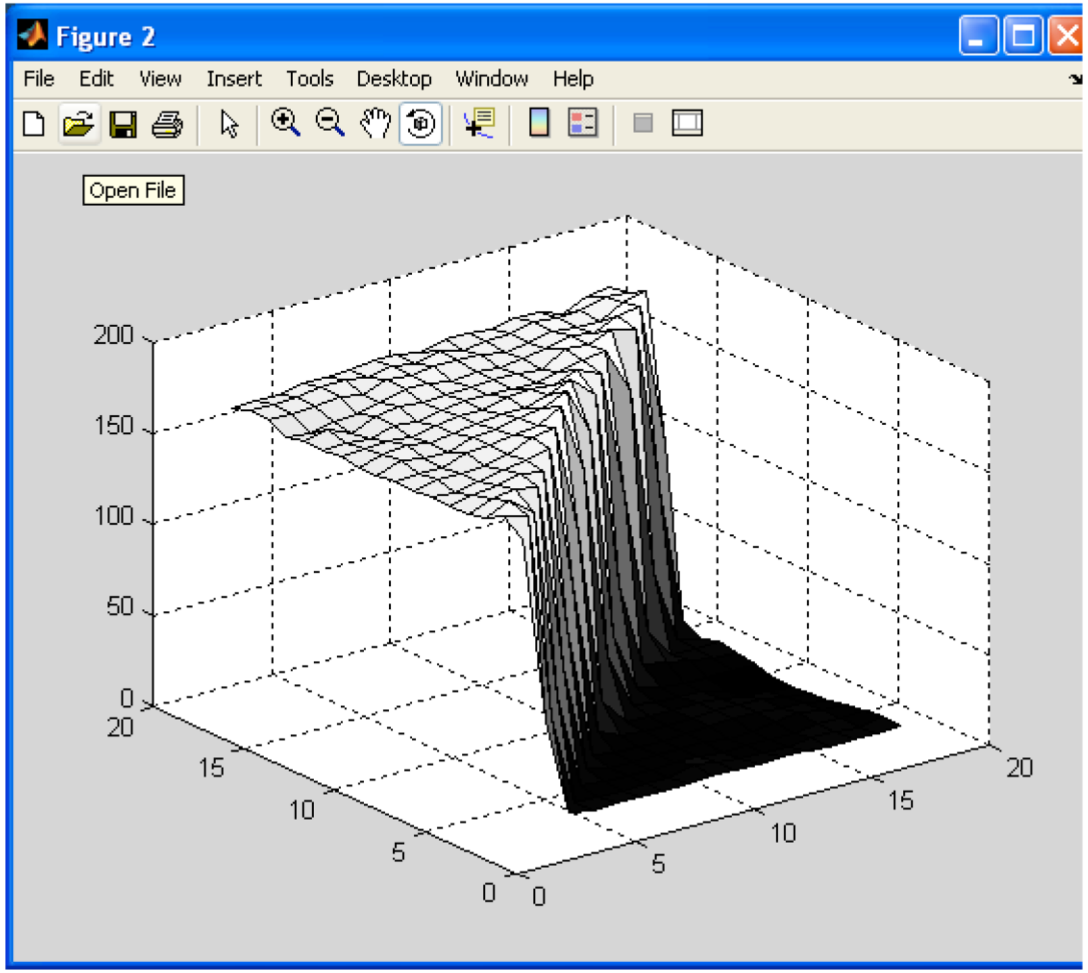
Edges

- Edges occur at boundaries between regions of different color, intensity or texture

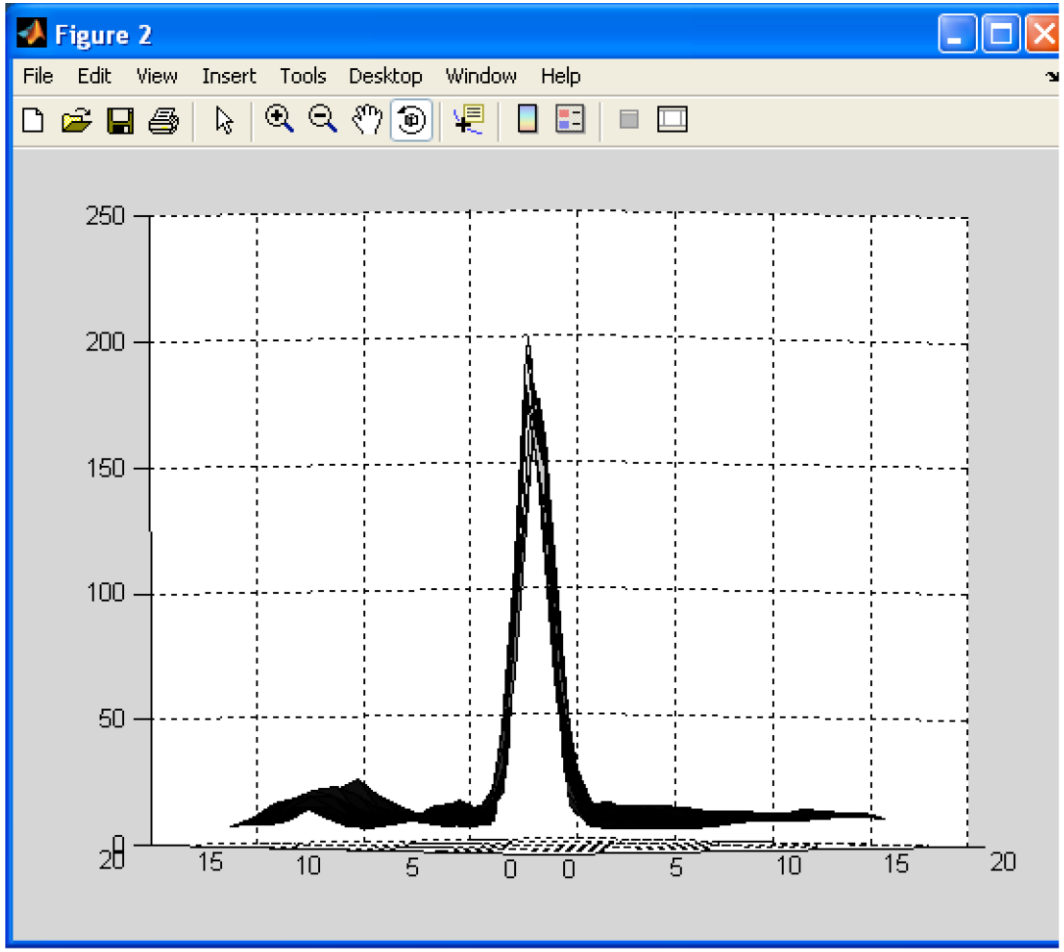
Step Edge and Ramp Edge



Step Edge, Ramp Edge

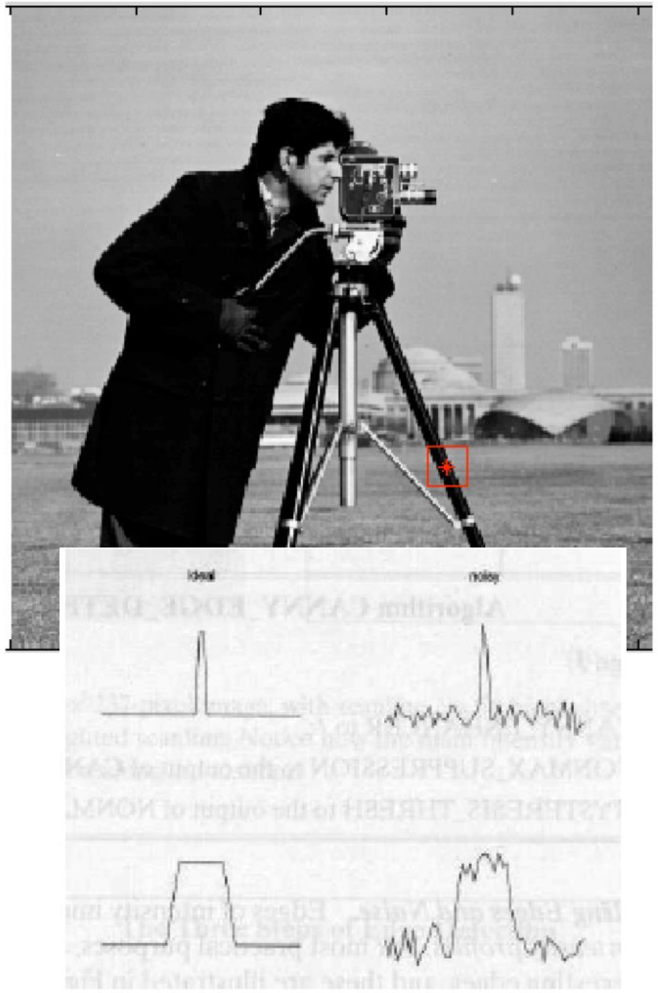


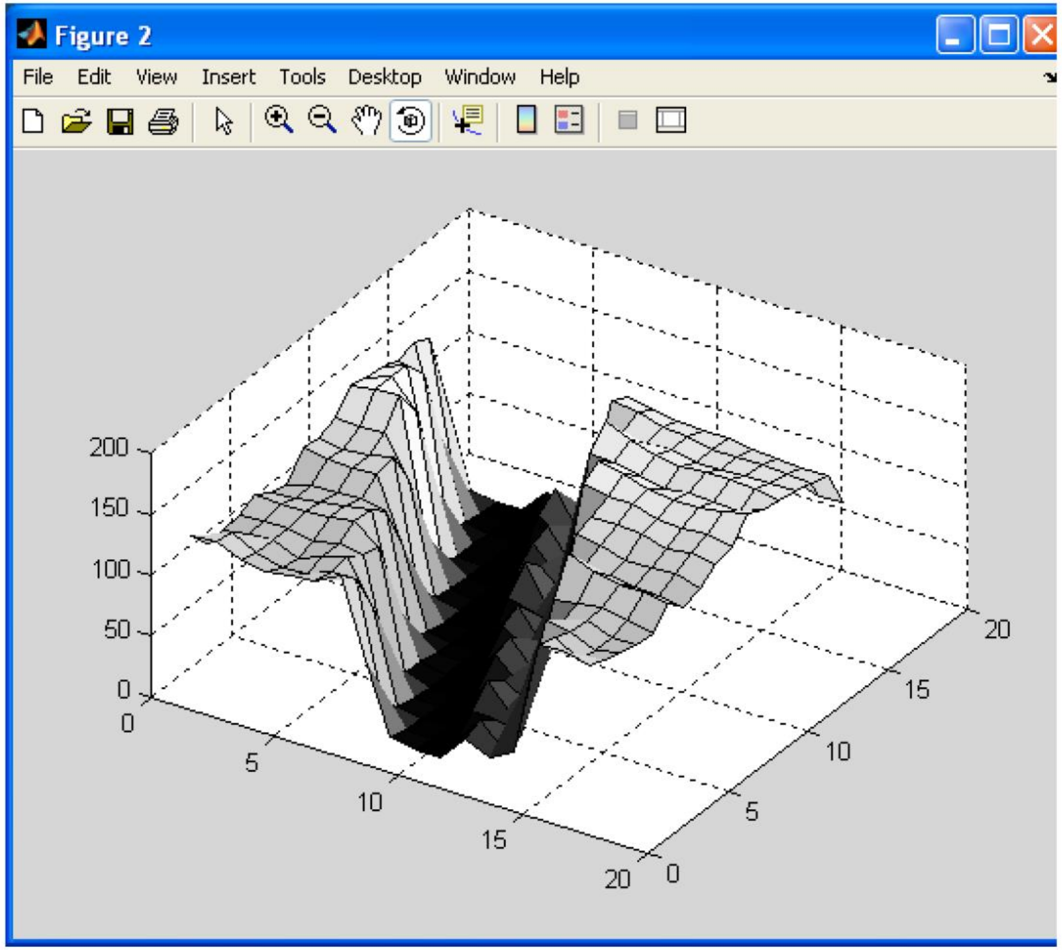
Ridge Edge



Ridge Edge

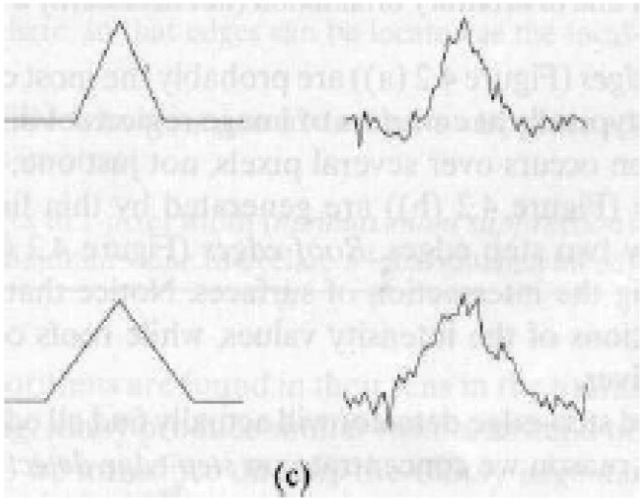
Ridge Edge

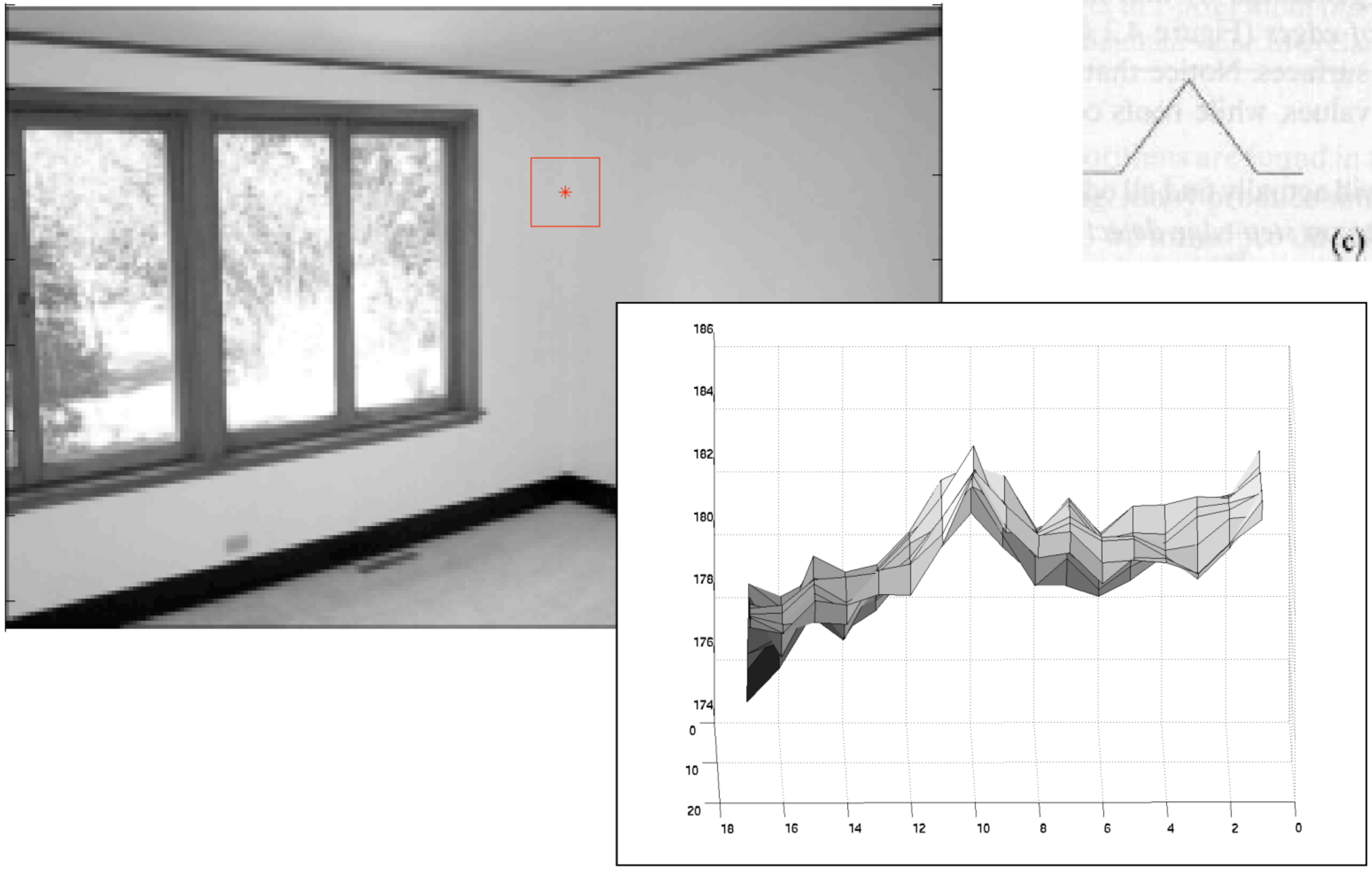




Ridge Edge

Roof Edge





Roof Edge

How to Detect Edges?

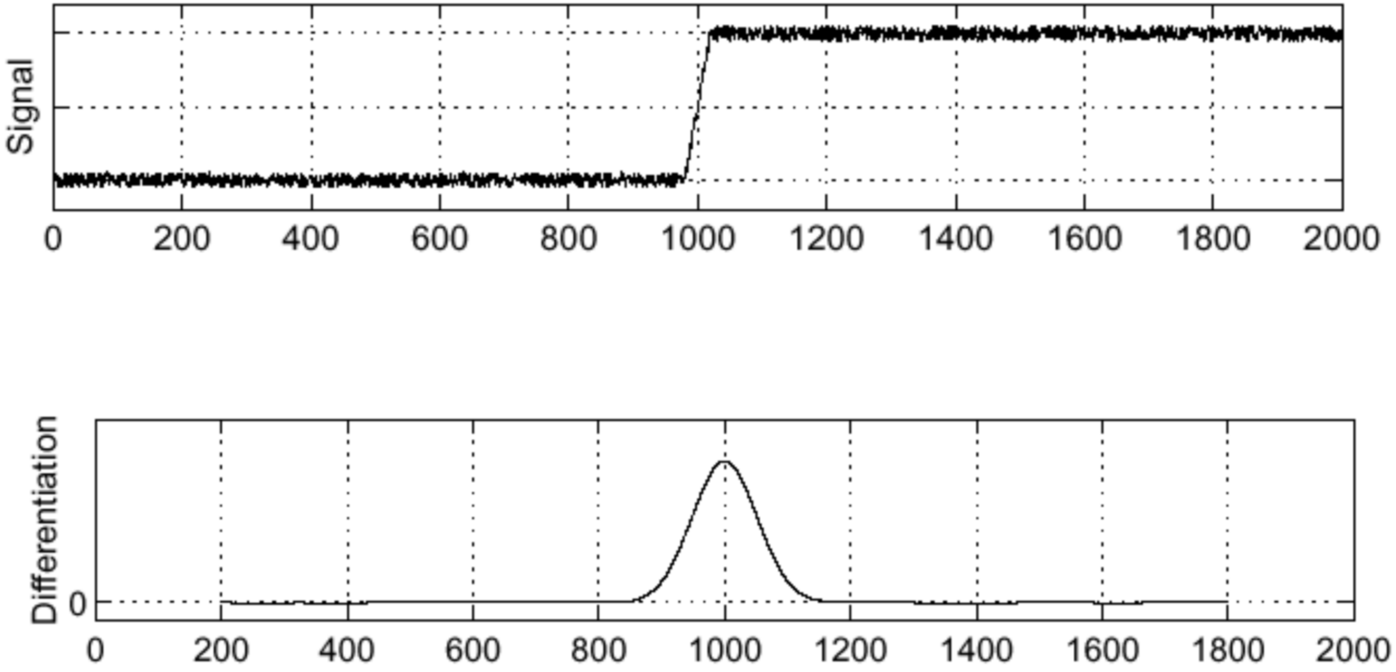
- Recall Derivative Filter

Central difference

$$f'(x) = \lim_{h \to 0} \frac{f(x+0.5h) - f(x-0.5h)}{h}$$

-1	0	1

X derivative filter

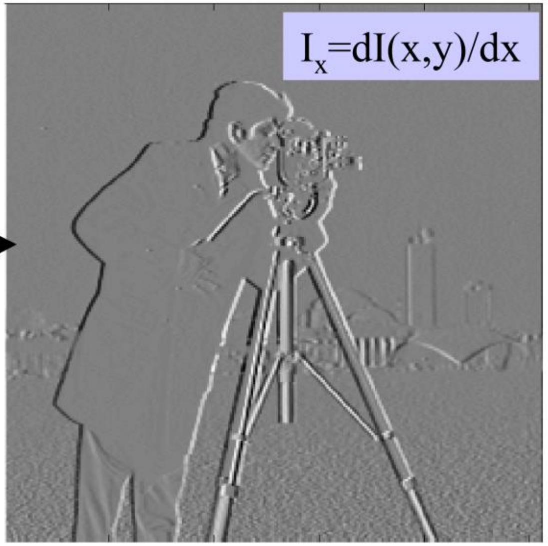


Find edge

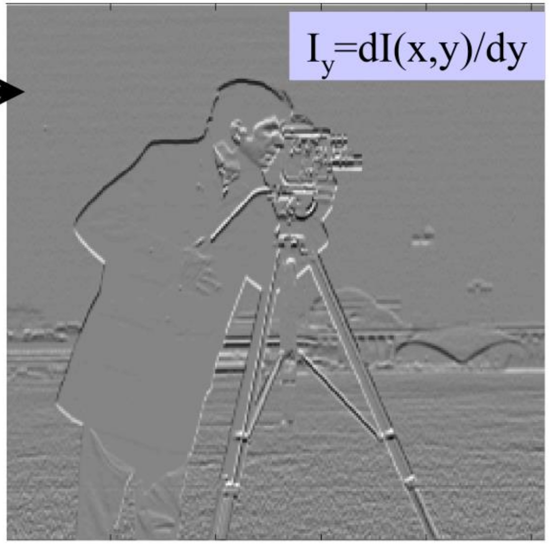
Recall Derivative Convolution

$$I_x = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * I$$

Partial derivative wrt x



Partial derivative wrt y



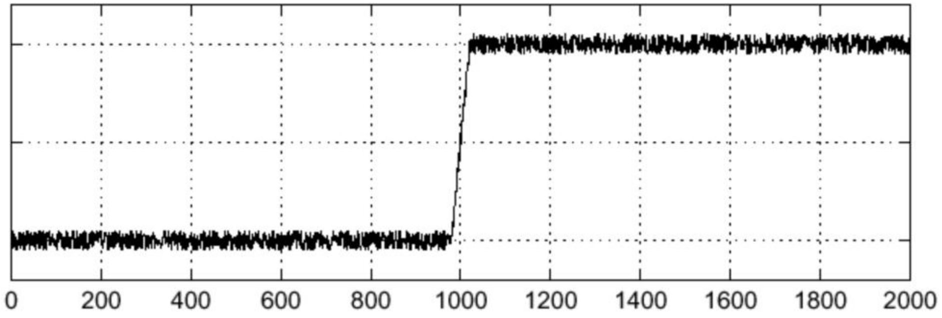
$$I_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * I$$

Note that there is a difference between convolving with a 1xn row filter and an nx1 col filter.

Signal Noises

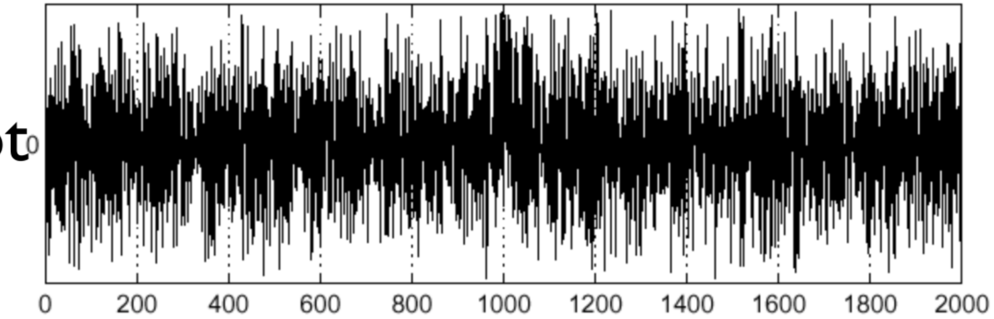
- Derivative filters are sensitive to noises

Intensity plot



How to deal with noises?

Derivative plot

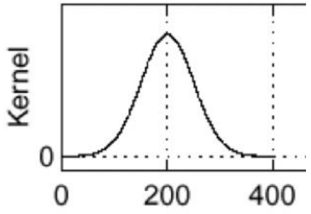


Gaussian Filter

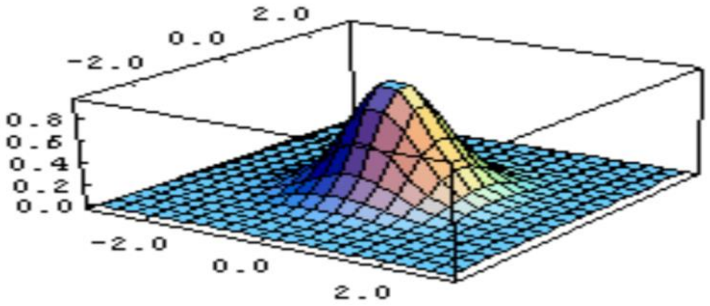
- Smoothing

1D $$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

Gaussian Filter h



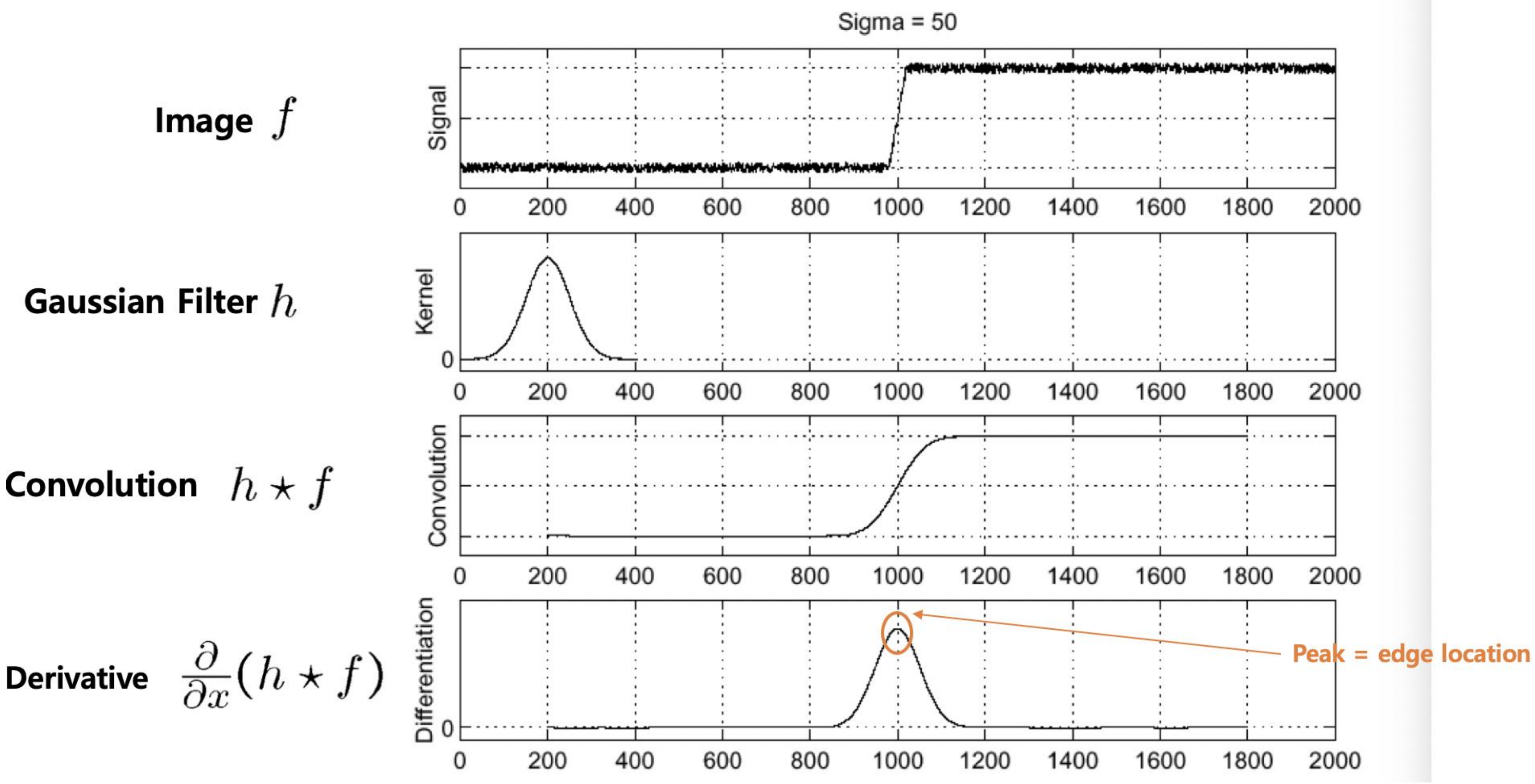
2D $$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

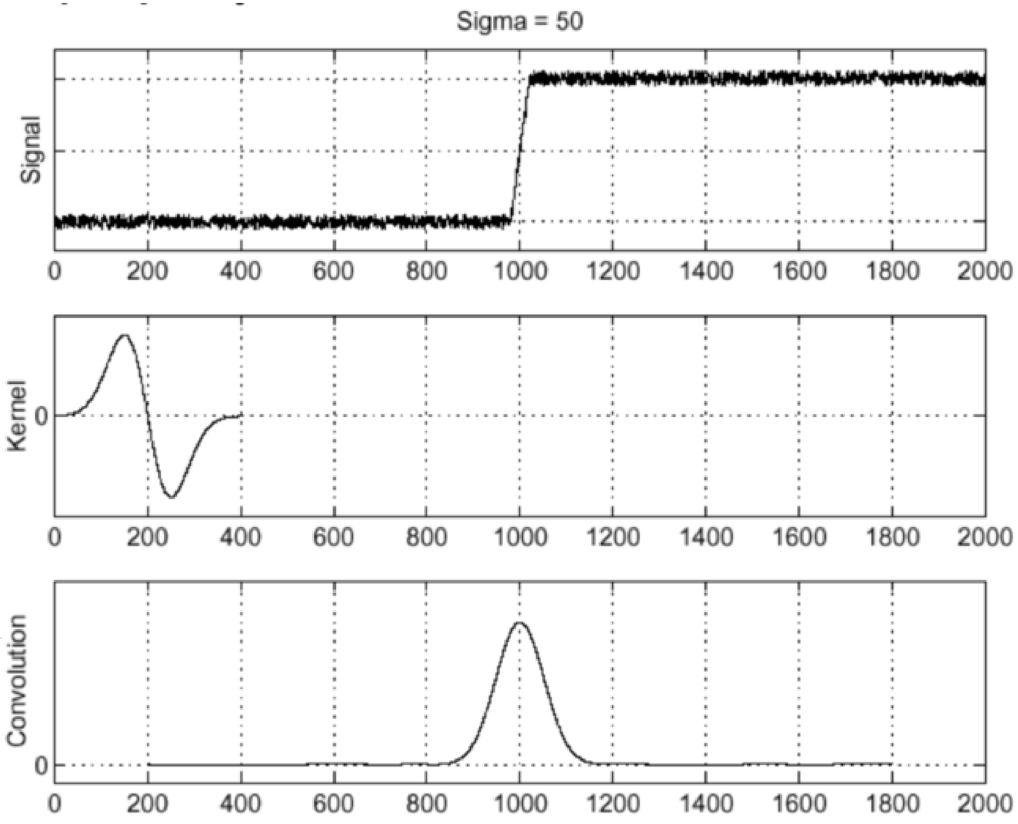
Gaussian Filter



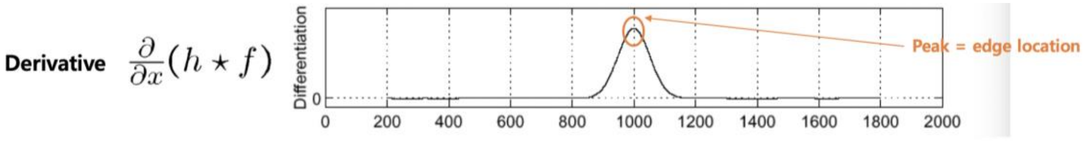
Derivative of Gaussian Filter

- Derivative Theorem of Convolution $\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$

f



Smoothing and derivative $\frac{\partial}{\partial x}h$

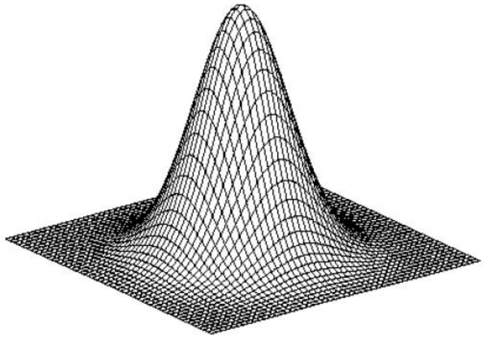


$(\frac{\partial}{\partial x}h) \star f$

Derivative of Gaussian Filter

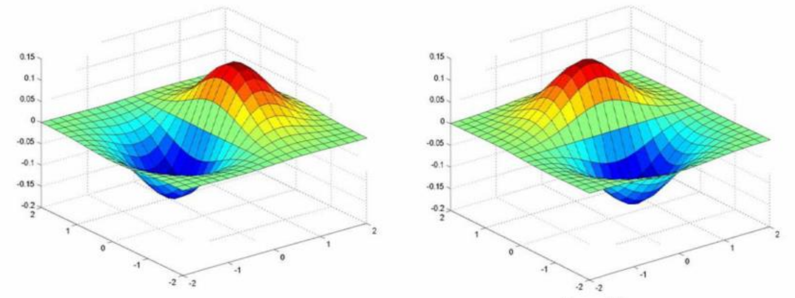
- Derivative Theorem of Convolution $\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$

$$g_x(x,y) = \frac{\partial g(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad g_y(x,y) = \frac{\partial g(x,y)}{\partial y} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



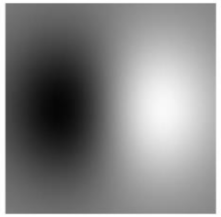
Gaussian

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



x-direction *y*-direction

Smoothing and derivative



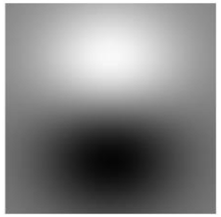
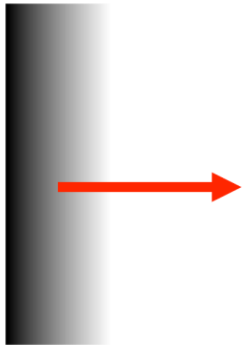


Image Gradient

Gradient in x only

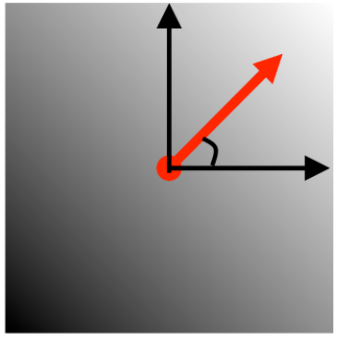


$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$

Gradient in y only

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

Gradient in both x and y



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

Gradient direction

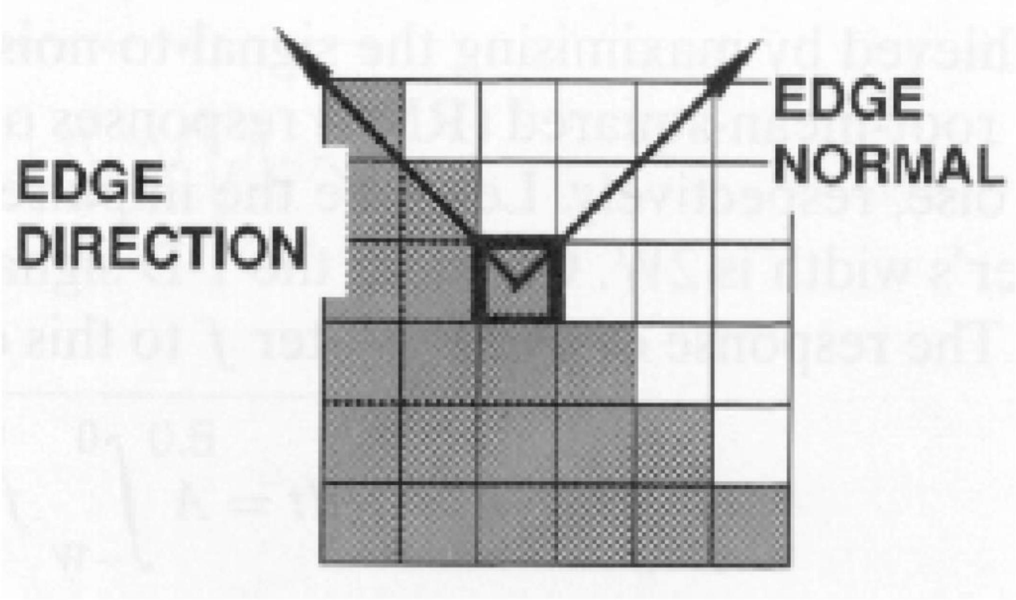
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Gradient magnitude

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

Edge Normal and Edge Direction

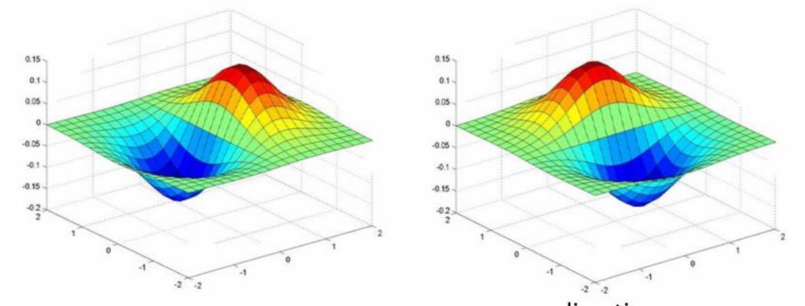
- Edge normal
  - Unit vector in the direction of maximum intensity change
  - Gradient direction
- Edge direction
  - Unit vector along edge (perpendicular to edge normal)



Edge Detection

- A simple edge detector using gradient magnitude

1. Compute gradient vector at each pixel by convolving image with horizontal and vertical derivative filters



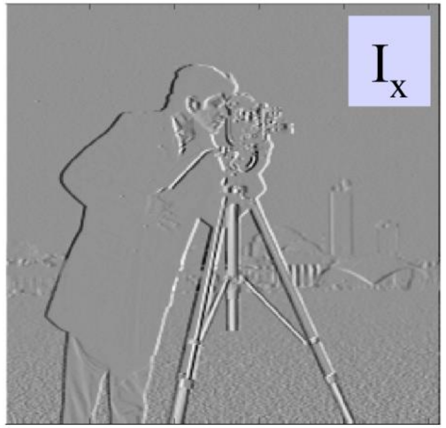
x-direction *y*-direction

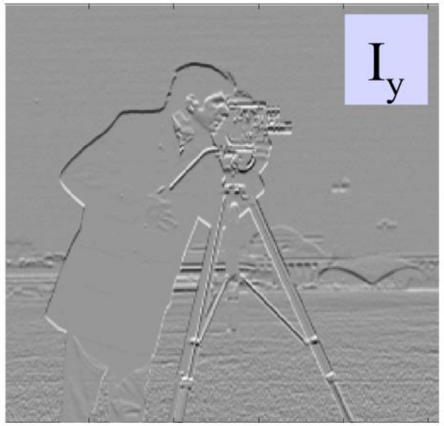
2. Compute gradient magnitude at each pixel

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

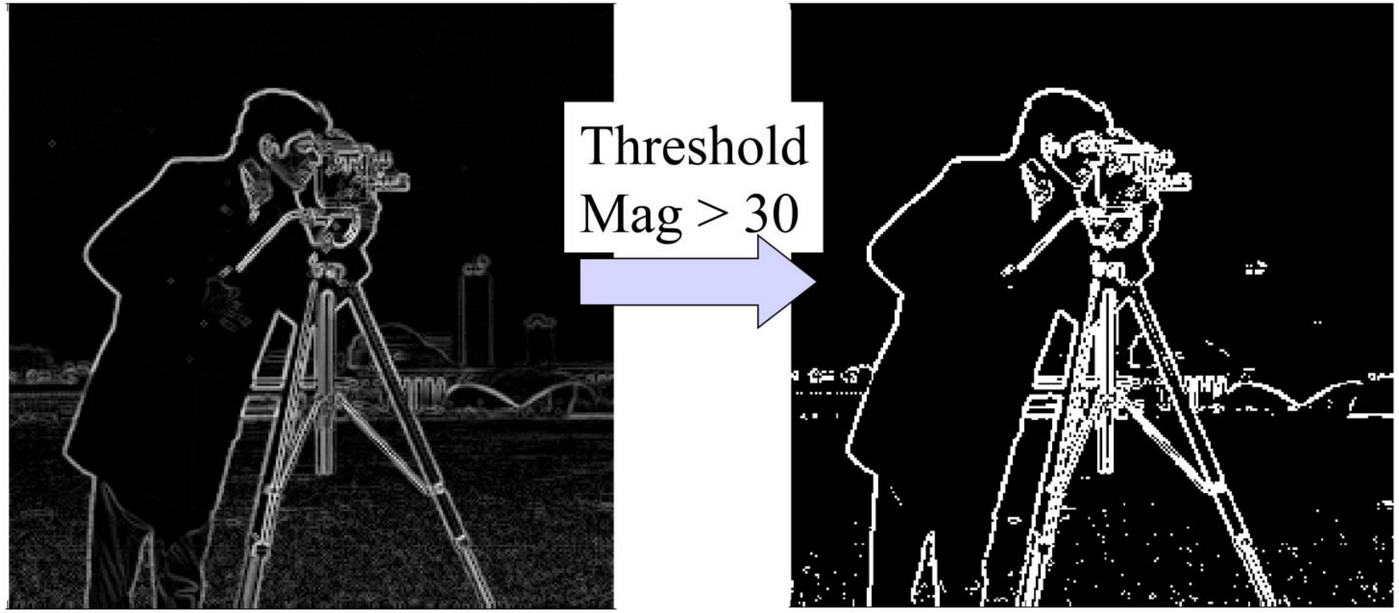
3. If magnitude at a pixel exceeds a threshold, report a possible edge point

Edge Detection





Magnitude of gradients



Edge Detection

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

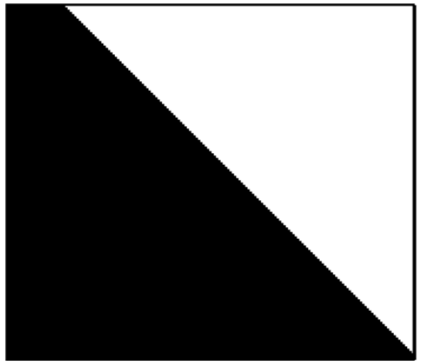
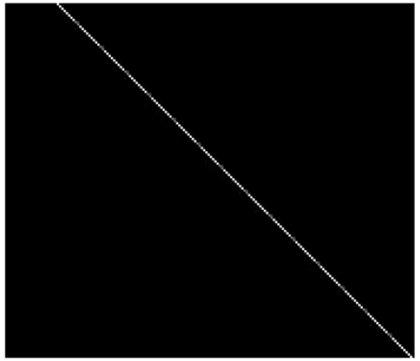


Image with Edge


Edge Location

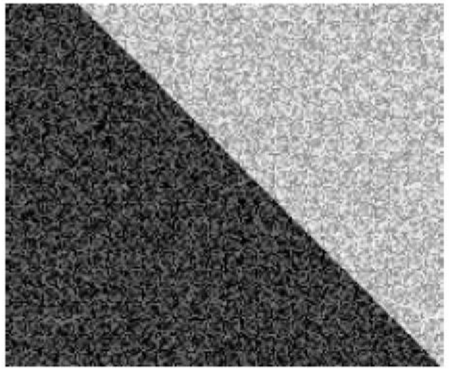
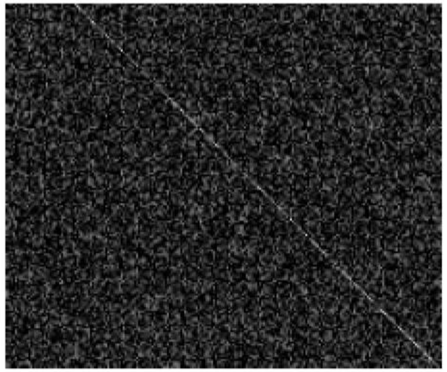
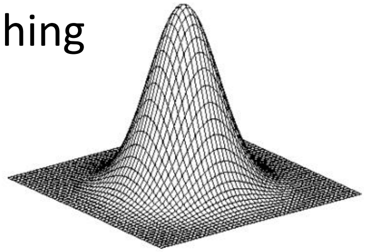


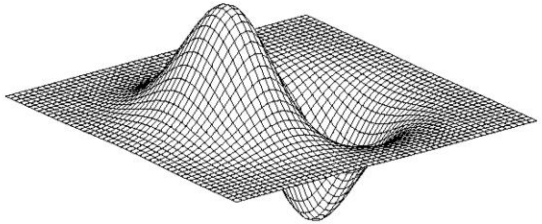
Image + Noise


Derivatives detect
edge *and* noise

Smoothing


Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

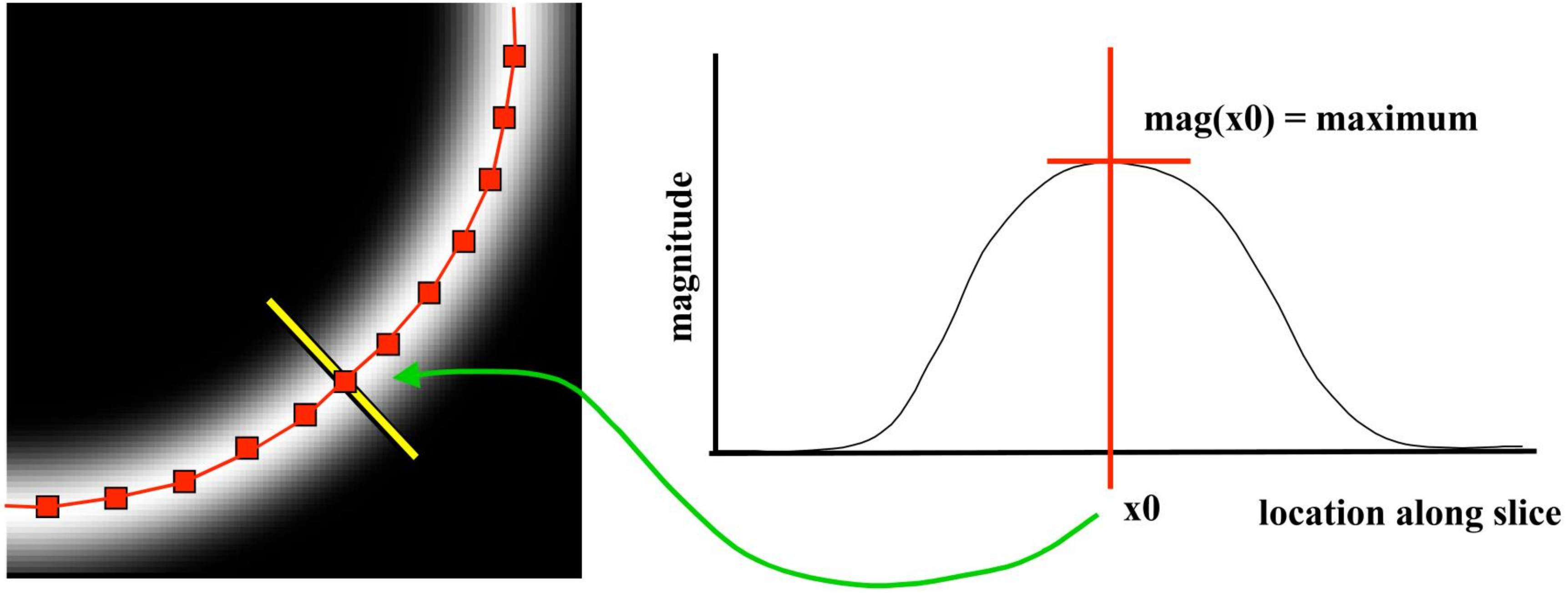

derivative of Gaussian (x)

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$


Smoothed derivative removes
noise, but blurs edge

Edge Detection

- Thinning

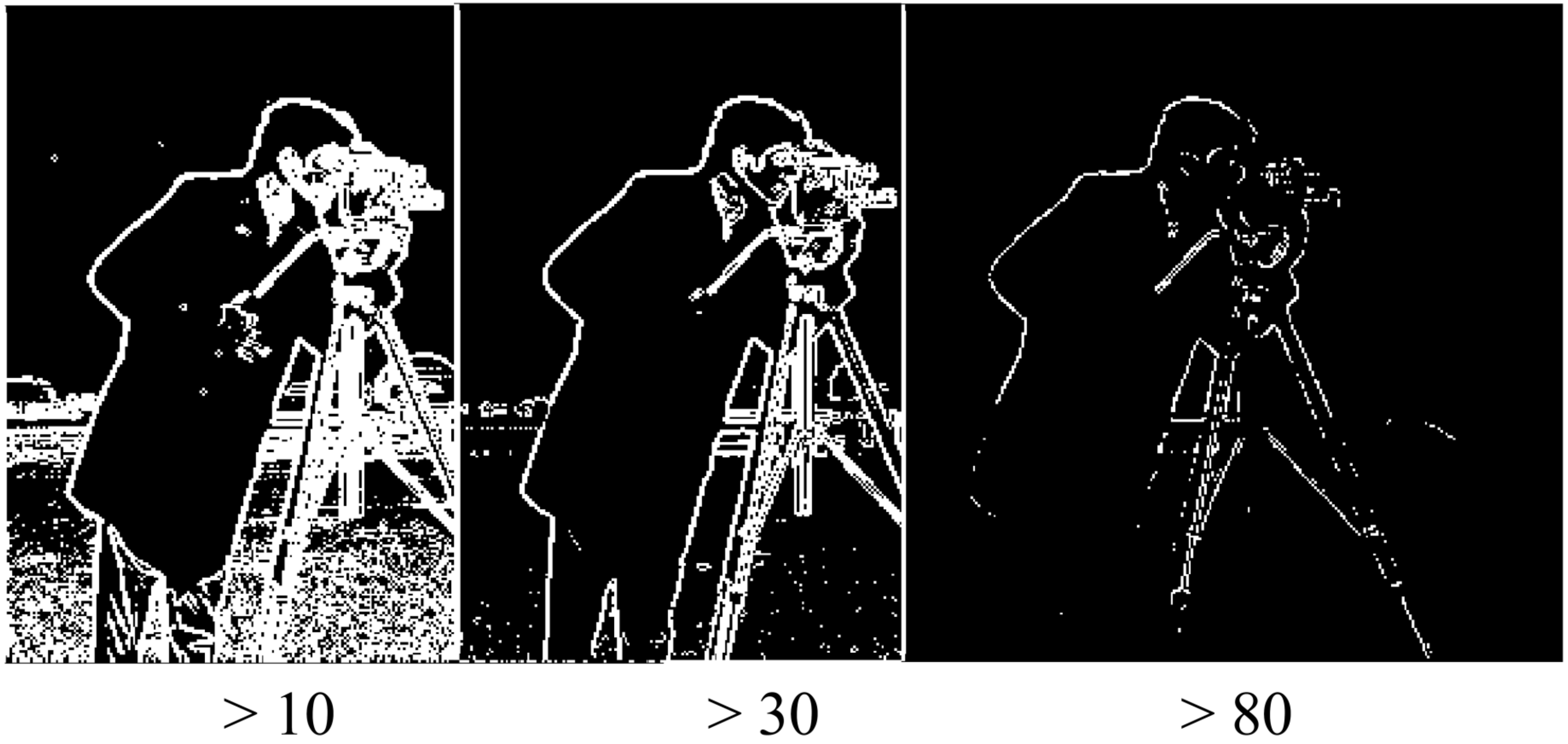


Along a 1D intensity slice normal to the curve (non-maximum suppression)

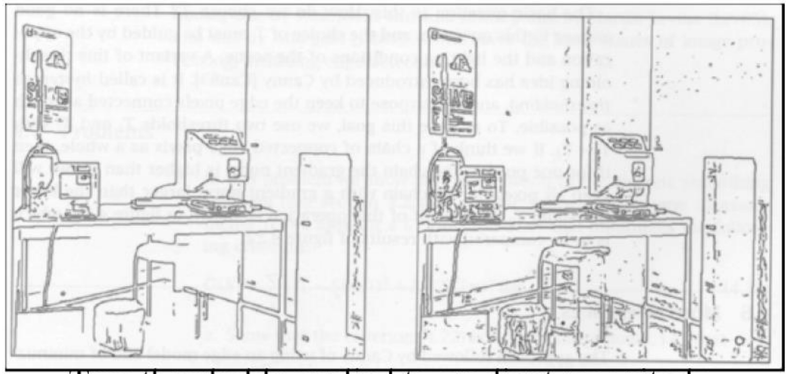
- Direction of gradient

Edge Detection

- How to chose the threshold?



Edge Detection

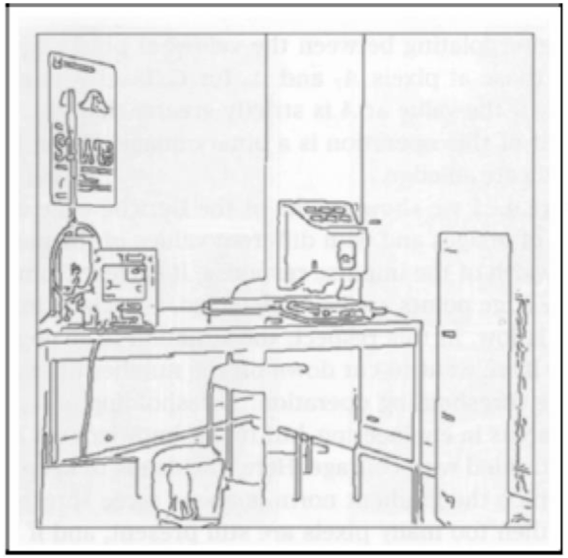


Two thresholds applied to gradient magnitude

T = 15 T = 5

- How to chose the threshold?

- Hysteresis thresholding
  - Keep a high threshed H and a low threshold L
  - Any edge with strength < L is discarded
  - Any edge with strength > H is kept
  - An edge P with strength between L and H is kept only if there is a path of edges with strength > L connecting P to an edge of strength > H



Hysteresis

T_h=15 T_l = 5

Canny Edge Detector

J. Canny A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986

Further Reading

- Chapter 7.2, Richard Szeliski
- J. Canny. A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986