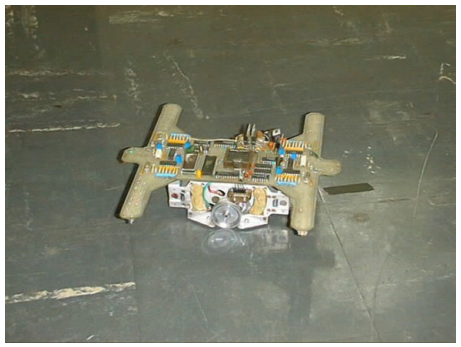
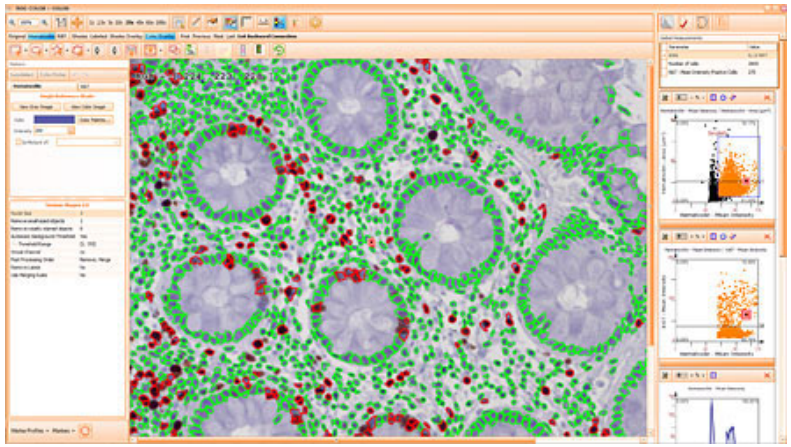
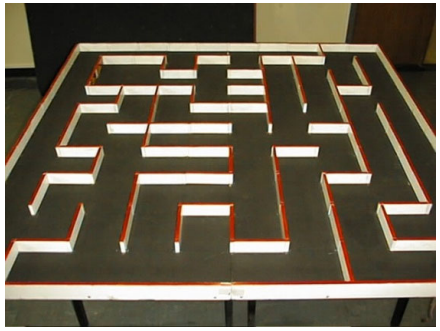
Image Analysis







Extracting Information From Images

Image Analysis

- The first step in image analysis is generally to segment the image.
- The level of segmentation depends on the problem to hand.
- For example, in an autonomous air-to-ground missile system we may segment the road from the image and then segment the road into objects that may correspond to vehicles. There is no point in segmenting below this scale or in looking at objects outside the boundary of the road.

Image Segmentation

- In general, autonomous segmentation is one of the most difficult tasks in image processing.
- Unfortunately, the performance of this difficult first step generally determines the overall success of the whole system.
- In fact, effective segmentation rarely fails to lead to a successful system.
- In industrial inspection applications, we often have some control over the environment which can greatly simplify the task of segmentation.

Vehicle Segmentation Example

Hand drawn. How do we do this with a computer?

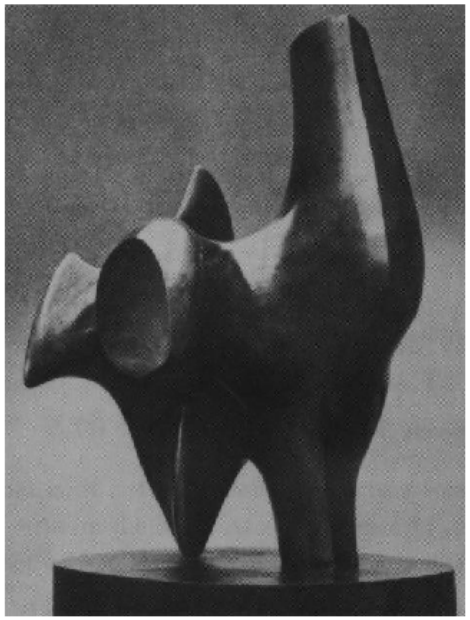
Controlling the Environment

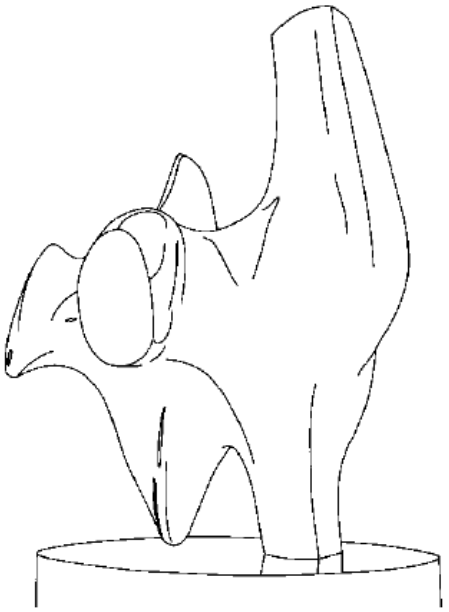
- If we can control some aspects of the imaging environment, we can often greatly simplify the task of segmentation. An experienced image analyst will never miss an opportunity to exploit this.
- Examples.
  - Control of lighting intensity, position of lights, colour of light.
  - Control of background field.
  - Control of camera type, focal length, position and aspect/orientation of object.
  - Use of segmentation aids such as staining of cell images.
  - Use of prior knowledge of expected objects under analysis.
  - Use of additional sensors (microwave).
  - Use of motion cues – temporal filtering.

Segmentation Techniques

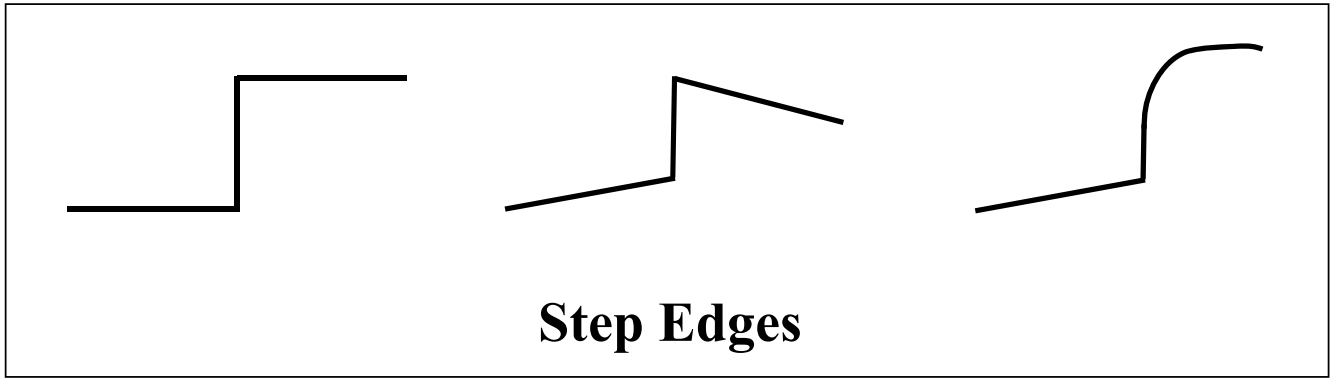
- Segmentation techniques for monochrome images generally are based on two basic properties of grayscale values
  - Discontinuity and Similarity
- Discontinuity based methods
  - detection of points, lines, and edges in an image
- Similarity Based Methods
  - Thresholding, region growing, region splitting and merging
- If we are examining dynamic (time-varying) images, then motion can give powerful cues to improve segmentation

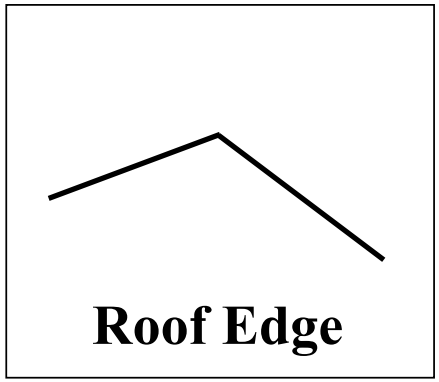
How can you tell that a pixel is on an edge?

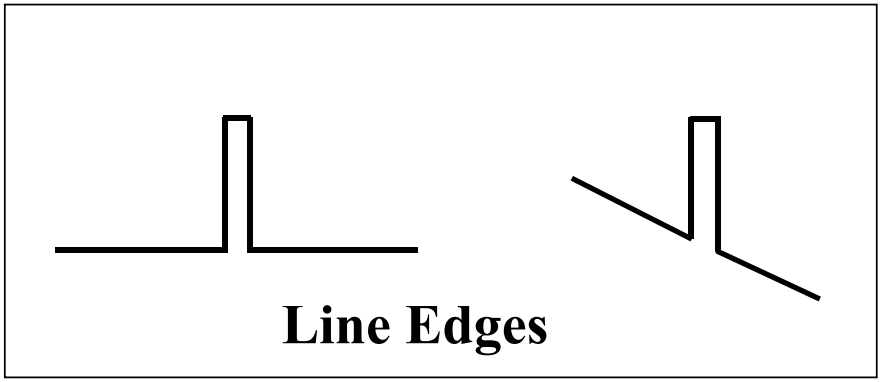




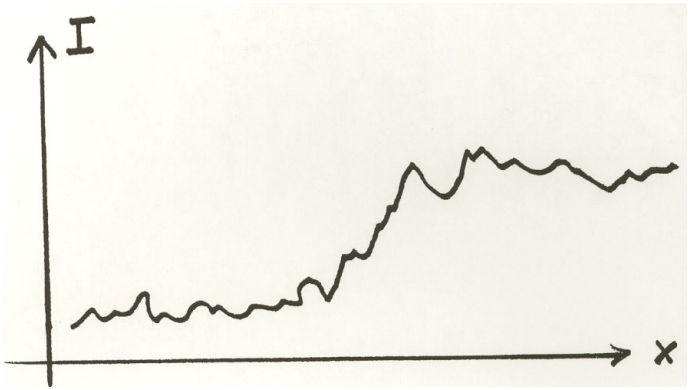
Edge Types







Real Edges



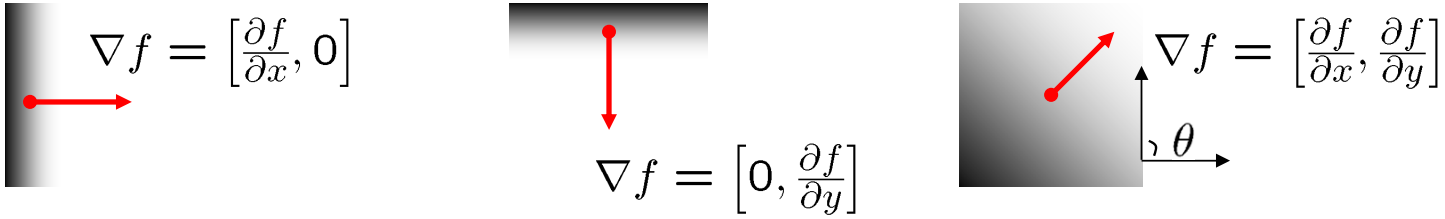
Noisy and Discrete!

We want an Edge Operator that produces:

- Edge Magnitude
- Edge Orientation
- High Detection Rate and Good Localization

Gradient

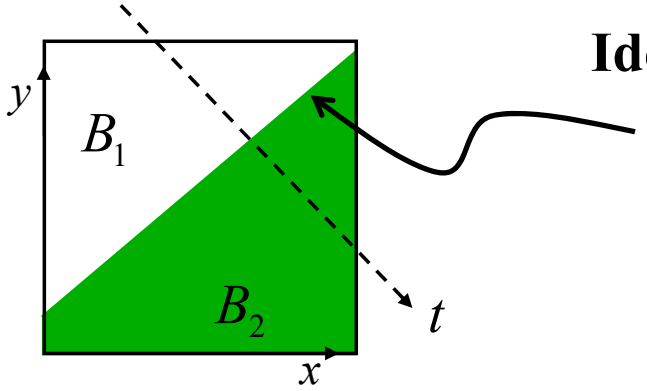
- Gradient equation: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- Represents direction of most rapid change in intensity



- **Gradient direction:** $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$
- **The *edge strength* is given by the gradient magnitude**

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

Theory of Edge Detection



Ideal edge

$$L(x,y) = x\sin\theta - y\cos\theta + \rho = 0$$

$$B_1 : L(x,y) < 0$$

$$B_2 : L(x,y) > 0$$

Unit step function:

$$u(t) = \begin{cases} 1 & \text{for } t > 0 \\ 1/2 & \text{for } t = 0 \\ 0 & \text{for } t < 0 \end{cases} \qquad u(t) = \int_{-\infty}^{t} \delta(s)\,ds$$

Image intensity (brightness):

$$I(x,y) = B_1 + (B_2 - B_1)\,u(x\sin\theta - y\cos\theta + \rho)$$

Theory of Edge Detection

- **Image intensity (brightness):**

$$I(x,y) = B_1 + (B_2 - B_1)u(x\sin\theta - y\cos\theta + \rho)$$

- **Partial derivatives (gradients):**

$$\frac{\partial I}{\partial x} = +\sin\theta(B_2 - B_1)\delta(x\sin\theta - y\cos\theta + \rho)$$

$$\frac{\partial I}{\partial y} = -\cos\theta(B_2 - B_1)\delta(x\sin\theta - y\cos\theta + \rho)$$

- **Squared gradient:**

$$s(x,y) = \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 = [(B_2 - B_1)\delta(x\sin\theta - y\cos\theta + \rho)]^2$$

Edge Magnitude: $\sqrt{s(x,y)}$ **Rotationally symmetric, non-linear operator**

Edge Orientation: $\arctan\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$ **(normal of the edge)**

- **Image intensity (brightness):**

$$I(x,y) = B_1 + (B_2 - B_1)u(x\sin\theta - y\cos\theta + \rho)$$

- **Partial derivatives (gradients):**

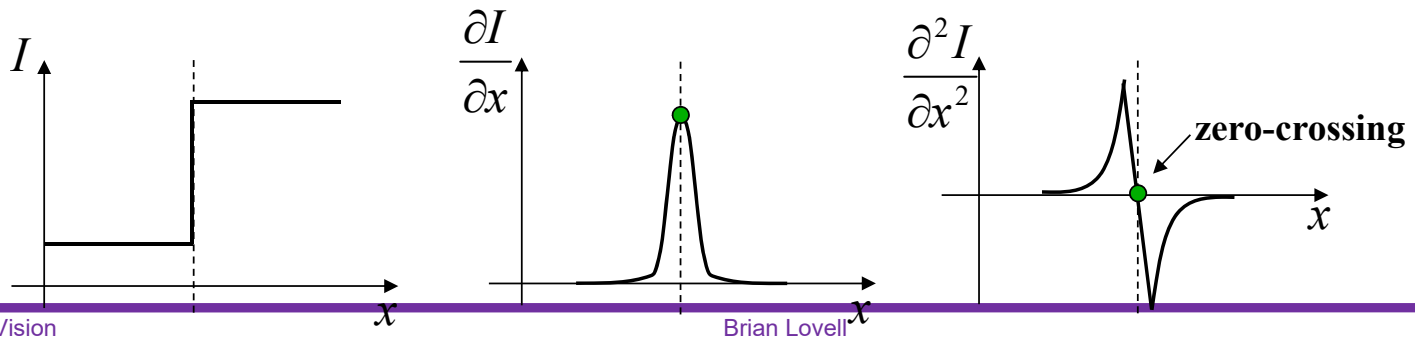
$$\frac{\partial I}{\partial x} = +\sin\theta(B_2 - B_1)\delta(x\sin\theta - y\cos\theta + \rho)$$

$$\frac{\partial I}{\partial y} = -\cos\theta(B_2 - B_1)\delta(x\sin\theta - y\cos\theta + \rho)$$

- **Laplacian:**

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} = (B_2 - B_1)\delta'(x\sin\theta - y\cos\theta + \rho)$$

Rotationally symmetric, linear operator



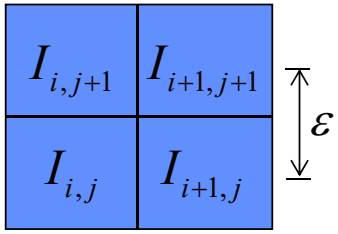
Discrete Edge Operators

- How can we differentiate a ***discrete*** image?

Finite difference approximations:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon}\left(\left(I_{i+1,j+1} - I_{i,j+1}\right) + \left(I_{i+1,j} - I_{i,j}\right)\right)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon}\left(\left(I_{i+1,j+1} - I_{i+1,j}\right) + \left(I_{i,j+1} - I_{i,j}\right)\right)$$



Convolution masks :

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Discrete Edge Operators

- **Second order partial derivatives:**

$$\frac{\partial^2 I}{\partial x^2} \approx \frac{1}{\varepsilon^2}\left(I_{i-1,j} - 2I_{i,j} + I_{i+1,j}\right)$$

$$\frac{\partial^2 I}{\partial y^2} \approx \frac{1}{\varepsilon^2}\left(I_{i,j-1} - 2I_{i,j} + I_{i,j+1}\right)$$

$I_{i-1,j+1}$	$I_{i,j+1}$	$I_{i+1,j+1}$
$I_{i-1,j}$	$I_{i,j}$	$I_{i+1,j}$
$I_{i-1,j-1}$	$I_{i,j-1}$	$I_{i+1,j-1}$

- **Laplacian :**

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Convolution masks :

$$\nabla^2 I \approx \frac{1}{\varepsilon^2}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array} \quad \textbf{or} \quad \frac{1}{6\varepsilon^2}\begin{array}{|c|c|c|}\hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline\end{array} \quad \textbf{(more accurate)}$$

The Sobel Operators

- Better approximations of the gradients exist
  - The *Sobel* operators below are commonly used

$$\frac{1}{8}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad s_x$$

$$\frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad s_y$$

Gradient:

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

Roberts (2 x 2):

0	1
-1	0

1	0
0	-1

Sobel (3 x 3):

-1	0	1
-1	0	1
-1	0	1

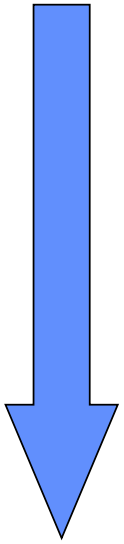
1	1	1
0	0	0
-1	-1	1

Sobel (5 x 5):

-1	-2	0	2	1
-2	-3	0	3	2
-3	-5	0	5	3
-2	-3	0	3	2
-1	-2	0	2	1

1	2	3	2	1
2	3	5	3	2
0	0	0	0	0
-2	-3	-5	-3	-2
-1	-2	-3	-2	-1

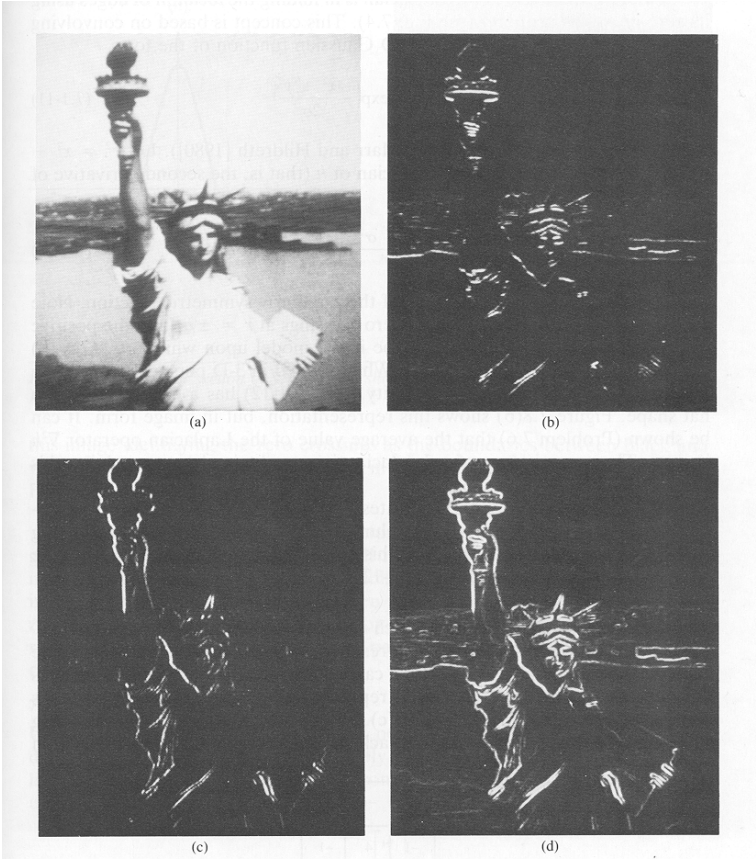
Good Localization
Noise Sensitive
Poor Detection



Poor Localization
Less Noise Sensitive
Good Detection

Sobel Edge Detection

Original



Gx

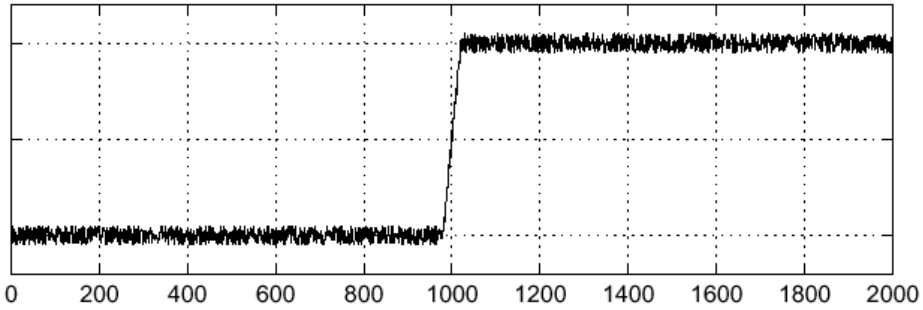
Gy

Combined

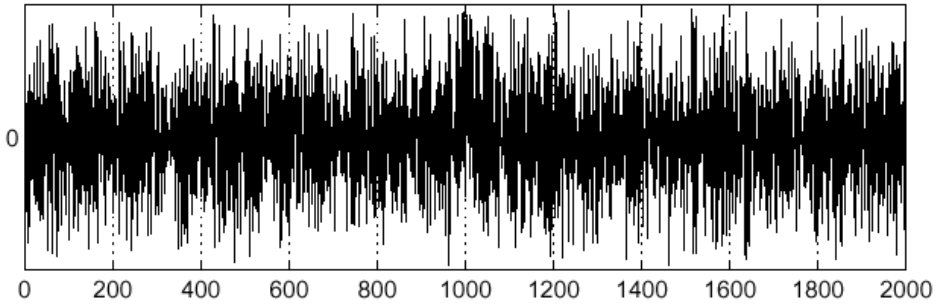
Effects of Noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

$f(x)$

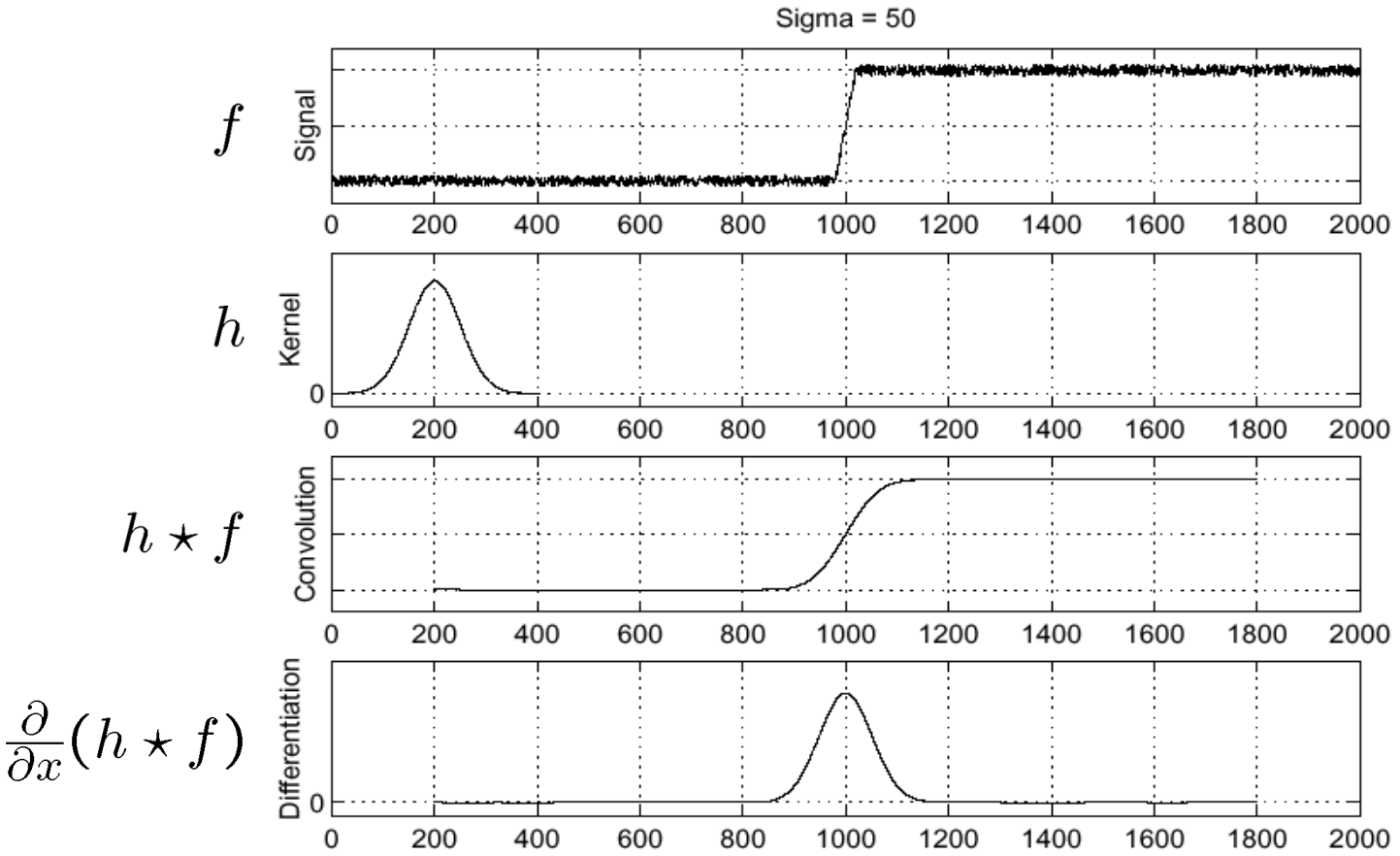


$\frac{d}{dx}f(x)$



Where is the edge??

Solution: Smooth First



Where is the edge?

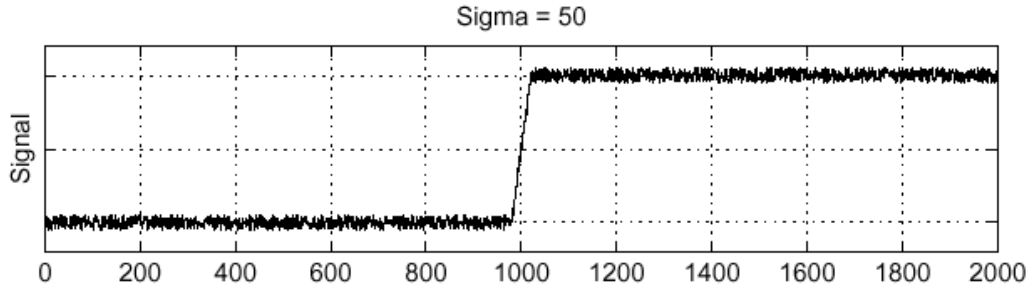
Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative Theorem of Convolution

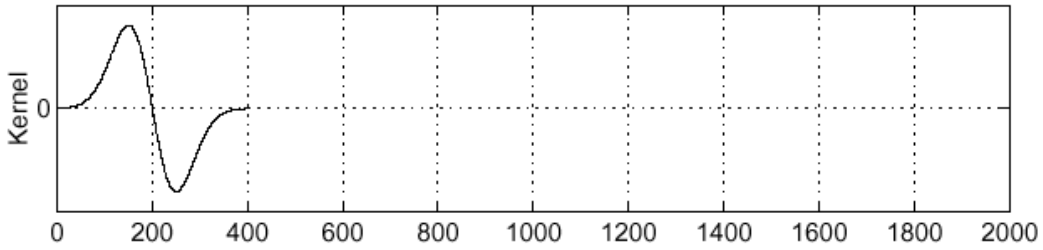
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

…saves us one operation.

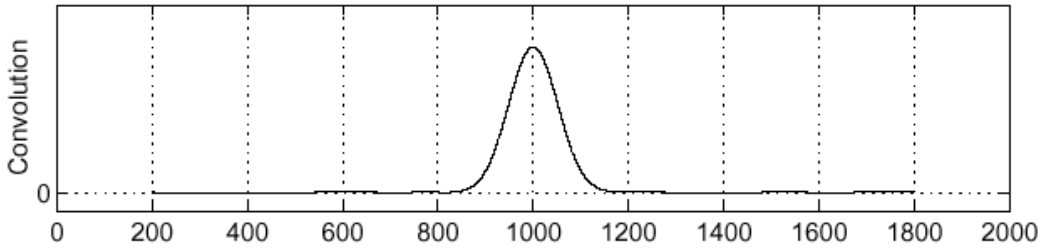
f



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



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



Laplacian of Gaussian (LoG)

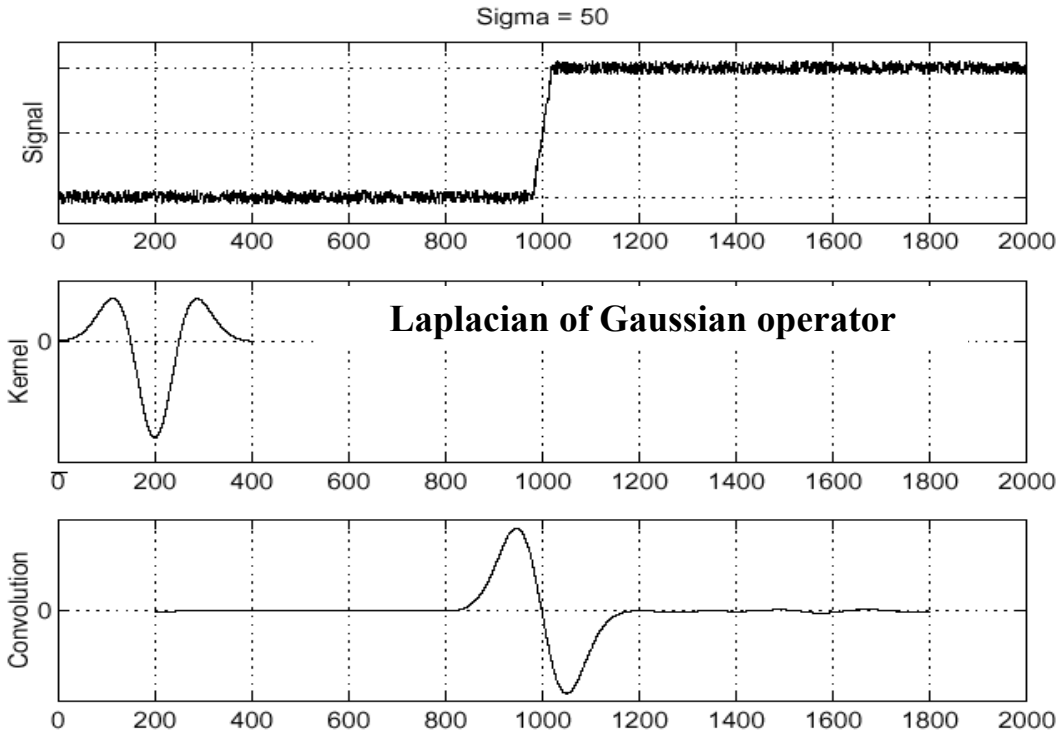
$$\frac{\partial^2}{\partial x^2}(h * f) = \left(\frac{\partial^2}{\partial x^2} h\right) * f$$

Laplacian of Gaussian

f

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

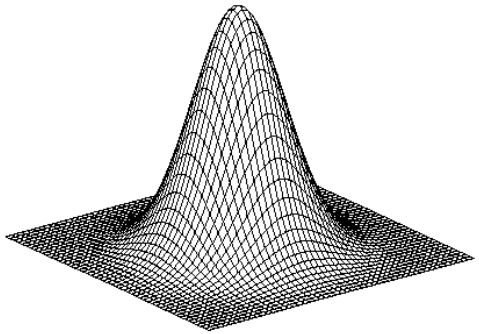
$(\frac{\partial^2}{\partial x^2} h) \star f$



Where is the edge?

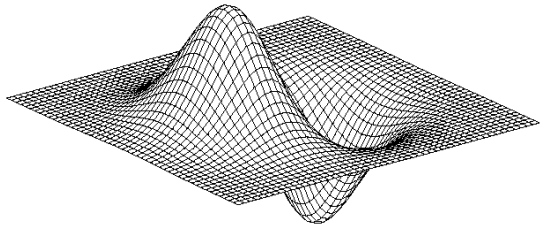
Zero-crossings of bottom graph !

2D Gaussian Edge Operators



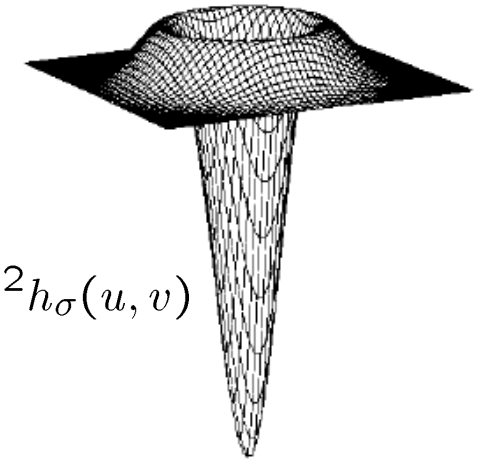
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Gaussian



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

Derivative of Gaussian (DoG)



$$\nabla^2 h_\sigma(u,v)$$

Laplacian of Gaussian

Mexican Hat (Sombrero)

- ∇^2 **is the Laplacian operator:**

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Canny Edge Operator

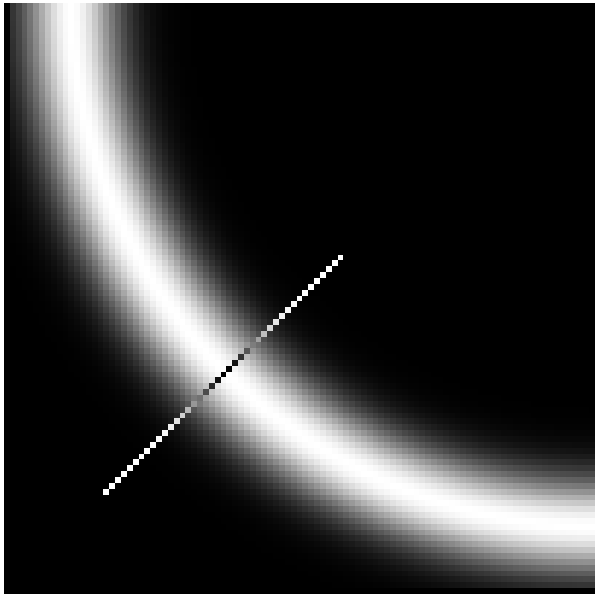
- Smooth image *I* with 2D Gaussian: $G * I$
- Find local edge normal directions for each pixel

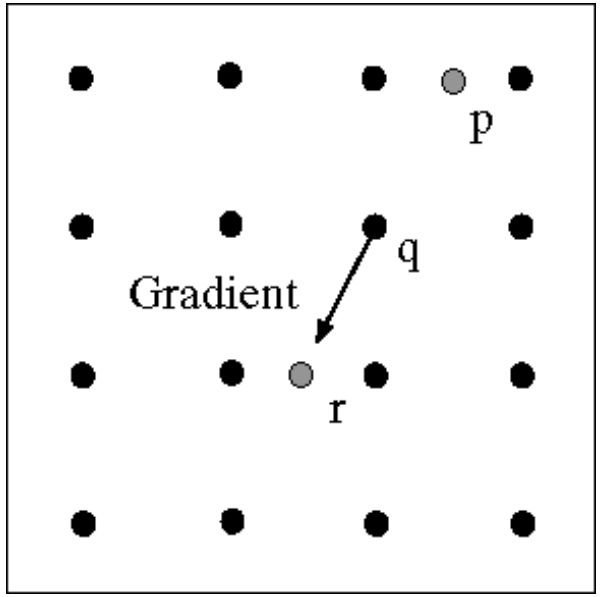
$$\overline{\mathbf{n}} = \frac{\nabla(G * I)}{|\nabla(G * I)|}$$

- Compute edge magnitudes $|\nabla(G * I)|$
- Locate edges by finding zero-crossings along the edge normal directions (**non-maximum suppression**)

$$\frac{\partial^2(G * I)}{\partial \overline{\mathbf{n}}^2} = 0$$

Non-maximum Suppression





- Check if pixel is local maximum along gradient direction
  - requires checking interpolated pixels p and r
  - Also uses hysteresis thresholding to follow weak edges below primary threshold

The Canny Edge Detector

original image (Lena)

The Canny Edge Detector

magnitude of the gradient

The Canny Edge Detector

After non-maximum suppression

Canny Edge Operator

original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$

- The choice of σ depends on desired behavior
  - large σ detects large scale edges
  - small σ detects fine features

Canny Edge Detector (and Linker)

Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

Processing Steps

- Apply Gaussian filter to smooth the image in order to remove the noise
- Find the intensity gradients of the image
- Apply non-maximum suppression to get rid of spurious response to edge detection (ridge following)
- Apply double threshold to determine potential edges
- Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Canny Edge Detector

The preferred method for many edge detection tasks