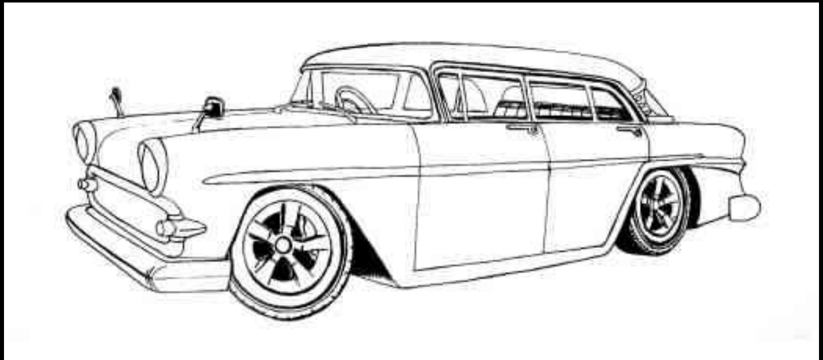
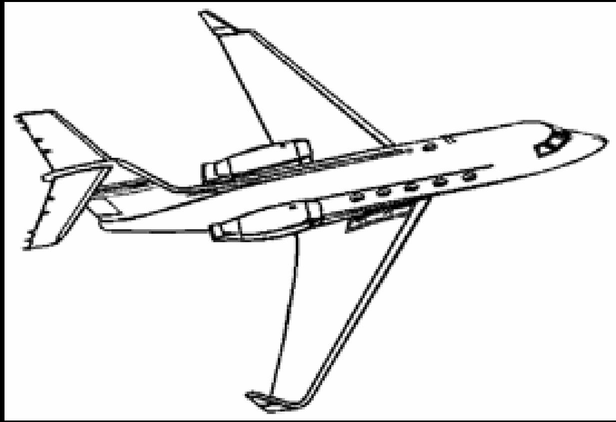
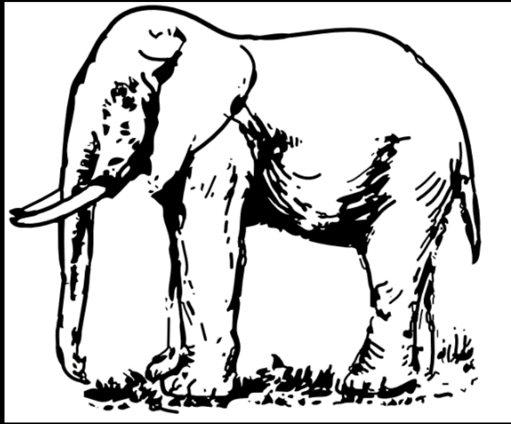
CS4495/6495
Introduction to Computer Vision

2A-L5 *Edge detection: Gradients*

Reduced images

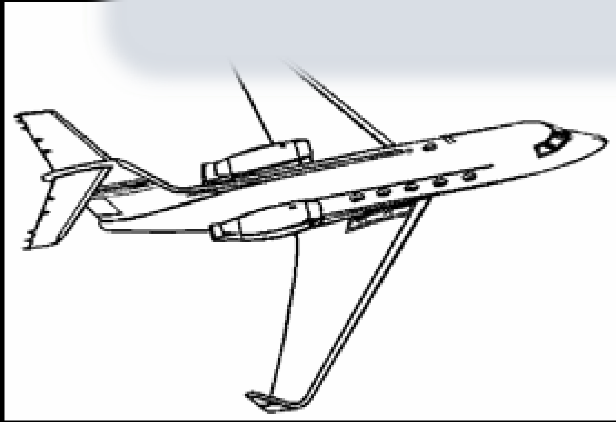


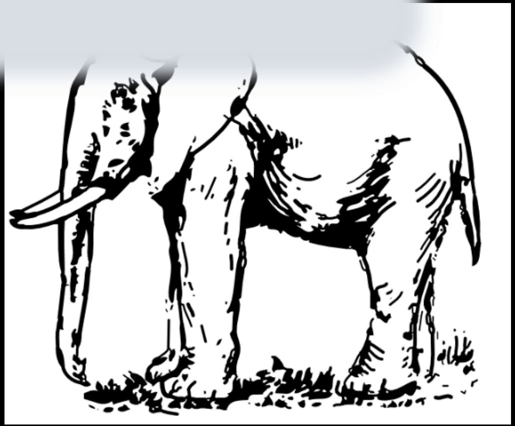




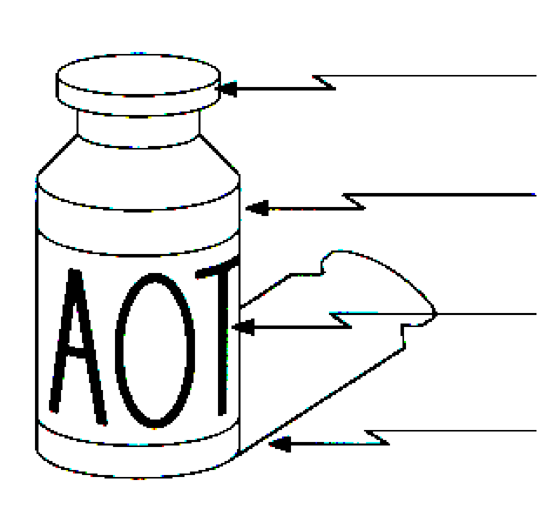
Reduced images

Edges seem to be important…





Origin of Edges



surface normal discontinuity

depth discontinuity

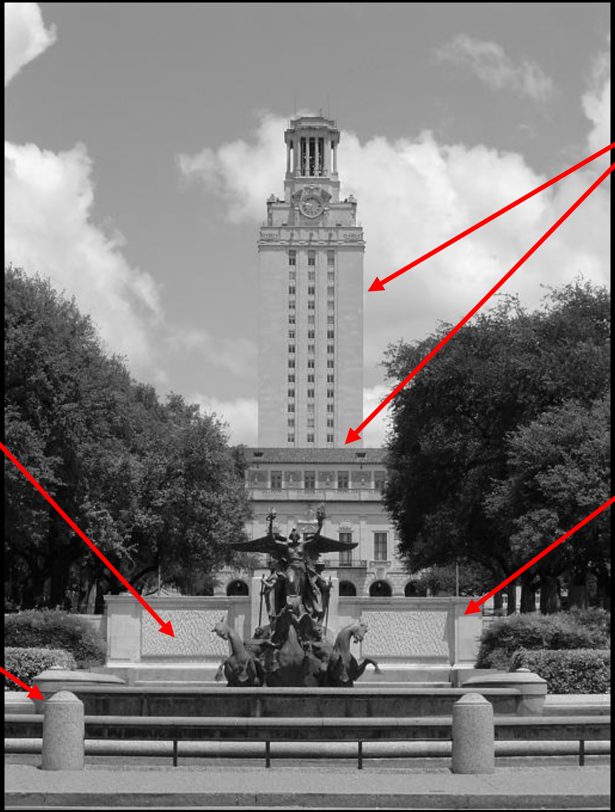
surface color discontinuity

illumination discontinuity

In a real image

Reflectance change: appearance information, texture

Discontinuous change in surface orientation



Depth discontinuity: object boundary

Cast shadows

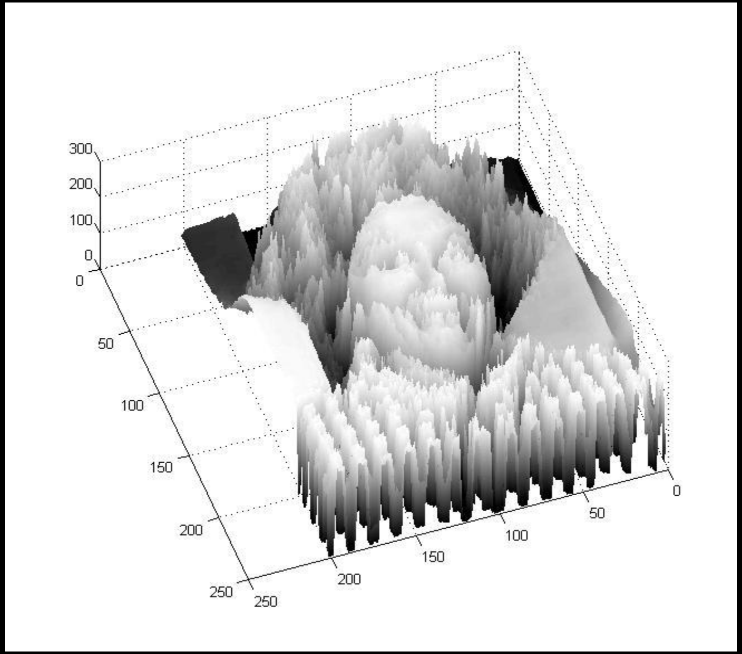
Edge detection

Quiz

Edges seem to occur "change boundaries" that are related to shape or illumination. Which is not such a boundary?

a) An occlusion between two people
b) A cast shadow on the sidewalk
c) A crease in paper
d) A stripe on a sign

Recall images as functions…



Edges look like steep cliffs

Edge Detection

Basic idea: look for a neighborhood with strong signs of change.

Problems:

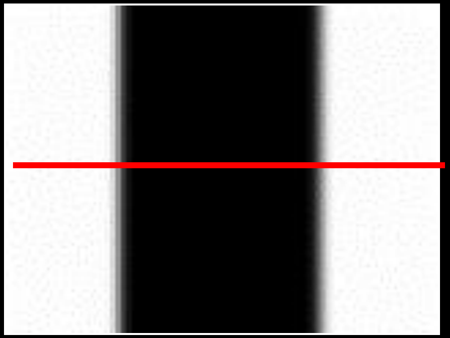
- neighborhood size
- how to detect change

81	82	26	24
82	33	25	25
81	82	26	24

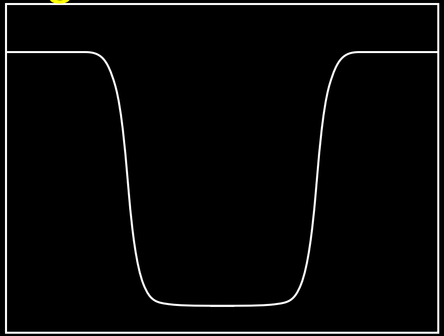
Derivatives and edges

An edge is a place of rapid change in the image intensity function.

image



intensity function
(along horizontal scanline)

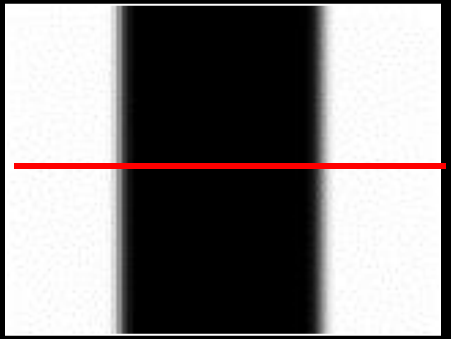


Source: S. Lazebnik

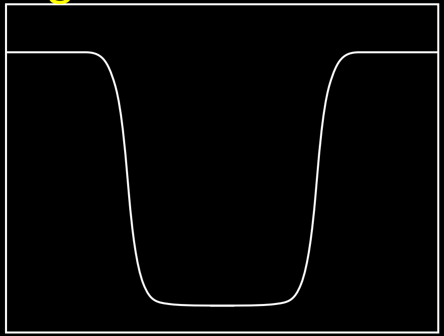
Derivatives and edges

An edge is a place of rapid change in the image intensity function.

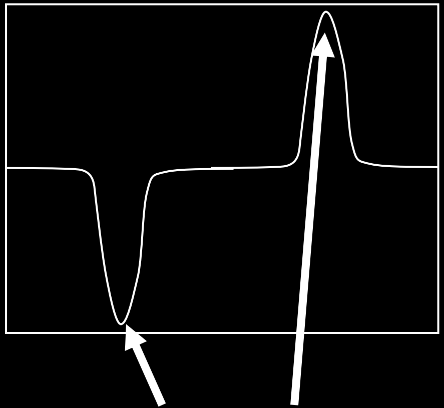
image



intensity function
(along horizontal scanline)



first derivative



edges correspond to
extrema of derivative

Source: S. Lazebnik

Differential Operators

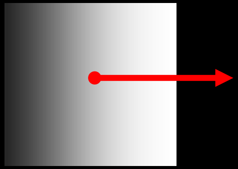
- Differential operators –when applied to the image returns some derivatives.
- Model these “operators” as masks/kernels that compute the image gradient function.
- Threshold the this gradient function to select the edge pixels.
- Which brings us to the question:

What’s a gradient?

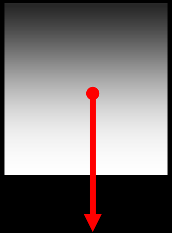
Image gradient

The gradient of an image:

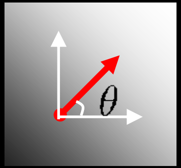
$$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$$



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



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



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

The gradient points in the direction of most rapid increase in intensity

Image gradient

The gradient of an image:

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

The gradient direction is given by:

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

The *edge strength* is given by the gradient magnitude:

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

Quiz

What does it mean when the magnitude of the image gradient is zero?

a) The image is constant over the entire neighborhood.
b) The underlying function $f(x,y)$ is at a maximum.
c) The underlying function $f(x,y)$ is at a minimum.
d) Either (a), (b), or (c).

words

- So that’s fine for calculus and other mathematics classes which you may now wish you had paid more attention. How do we compute these things on a computer with actual images.
- To do this we need to talk about discrete gradients.

Discrete gradient

For 2D function, f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

Discrete gradient

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x,y)}{1}$$

$$\approx f(x+1,y) - f(x,y)$$

"right derivative" But is it???

Finite differences

Source: D.A. Forsyth

Finite differences – x or y?

Source: D. Forsyth

Partial derivatives of an image

$$\frac{\partial f(x,y)}{\partial x}$$

$$\frac{\partial f(x,y)}{\partial y}$$

-1	1

(correlation filters)

Partial derivatives of an image

$$\frac{\partial f(x,y)}{\partial x}$$

-1	1

$$\frac{\partial f(x,y)}{\partial y}$$

-1
1

? or

1
-1

(correlation filters)

The discrete gradient

- We want an “operator” (mask/kernel) that we can apply to the image that implements:

$$\frac{\partial f(x, y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon}$$

How would you implement this as a cross-correlation?

The discrete gradient

0	0
-1	+1
0	0

H

Not symmetric around image point; which is "middle" pixel?

0	0	0
-1/2	0	+1/2
0	0	0

H

Average of "left" and "right" derivative. See?

Example: Sobel operator

$$\frac{1}{8} * \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad \frac{1}{8} * \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

$$s_x \qquad\qquad s_y$$

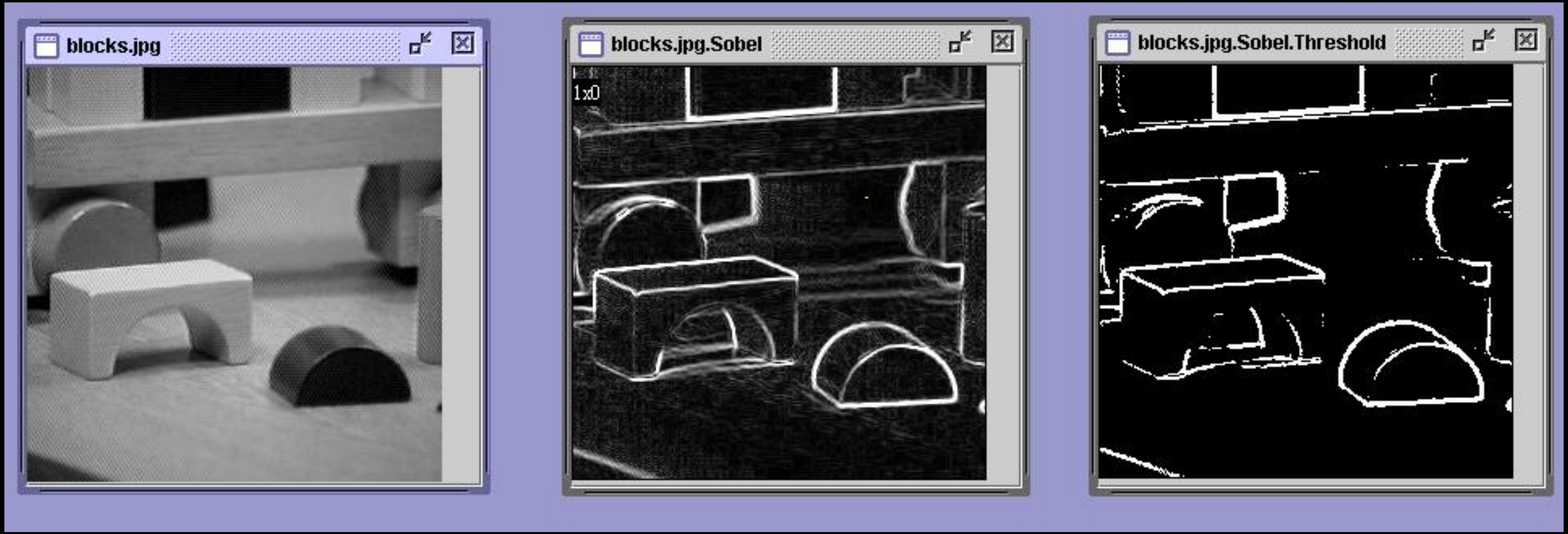
(here positive y is up)

(Sobel) Gradient is $\nabla \mathbf{I} = [\mathbf{g_x} \ \ \mathbf{g_y}]^{\mathrm{T}}$

$g = (g_x^{\,2} + g_y^{\,2})^{1/2}$ is the gradient magnitude.

$\theta = \mathrm{atan2}(g_y, g_x)$ is the gradient direction.

Sobel Operator on Blocks Image



original image

gradient magnitude

thresholded gradient magnitude

Some Well-Known Gradients Masks

- Sobel:

	Sx				Sy	
-1	0	1		1	2	1
-2	0	2		0	0	0
-1	0	1		-1	-2	-1

- Prewitt:

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

- Roberts:

0	1		1	0
-1	0		0	-1

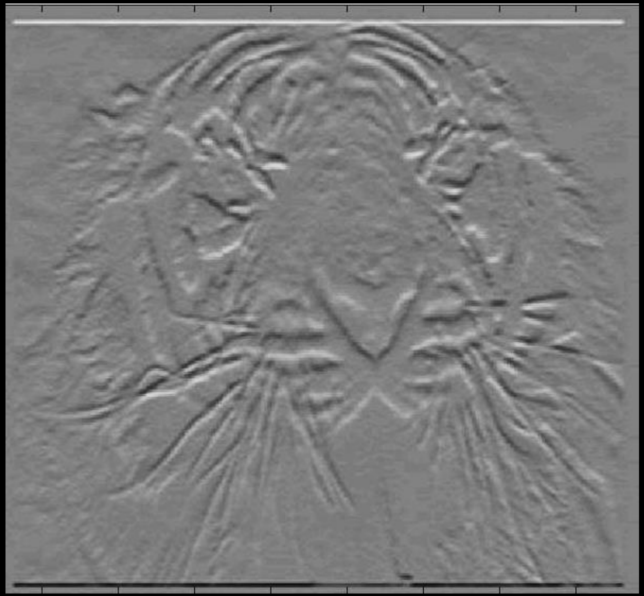
Matlab does gradients

```
filt = fspecial('sobel')

filt =

     1     2     1
     0     0     0
    -1    -2    -1

outim = imfilter(double(im),filt);
imagesc(outim);
colormap gray;
```



Quiz

It is better to compute gradients using:

a) ***Convolution*** since that's the right way to model filtering so you don't get flipped results.

b) ***Correlation*** because it's easier to know which way the derivatives are being computed.

c) Doesn't matter.

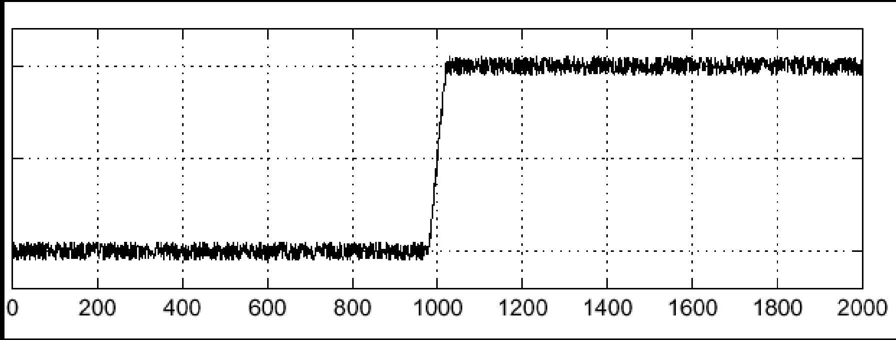
d) Neither since I can just write a for-loop to computer the derivatives.

But in the real world…

Consider a single row or column of the image

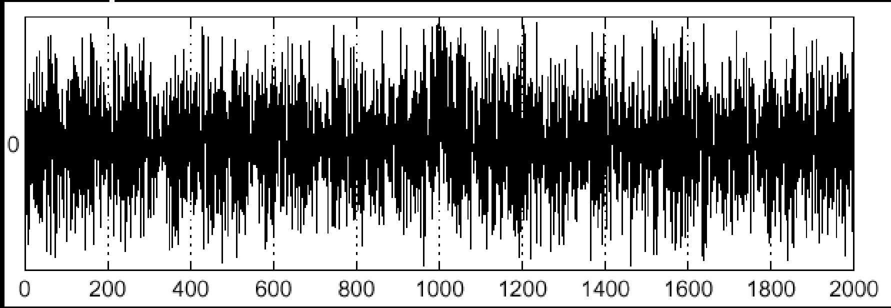
(plotting intensity as a function of *x*)

$f(x)$



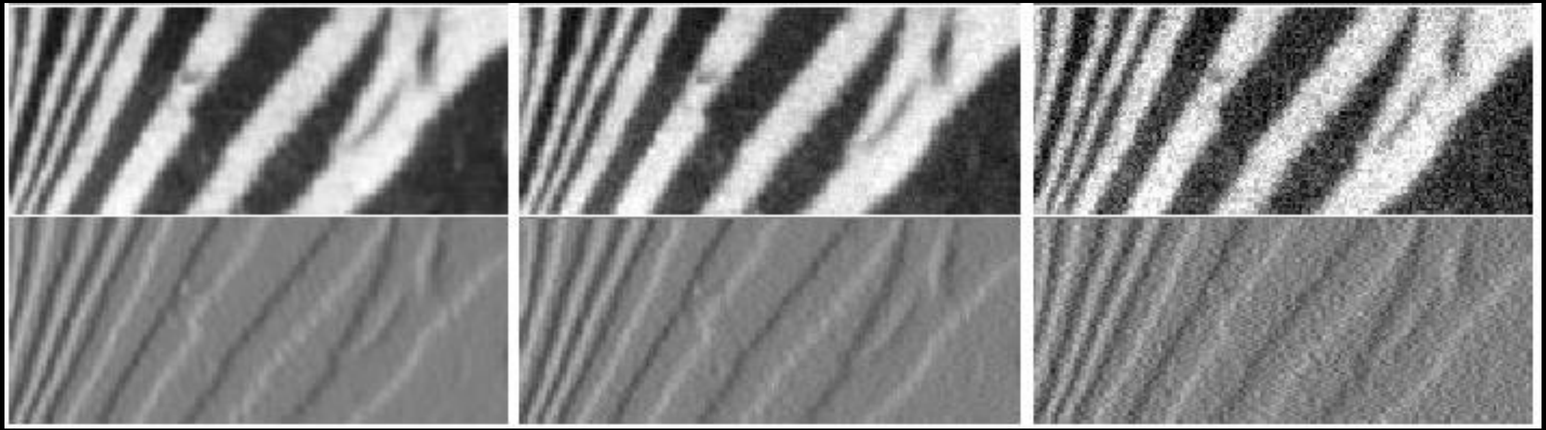
Apply derivative operator….

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



Uh, where's the edge?

Finite differences responding to noise



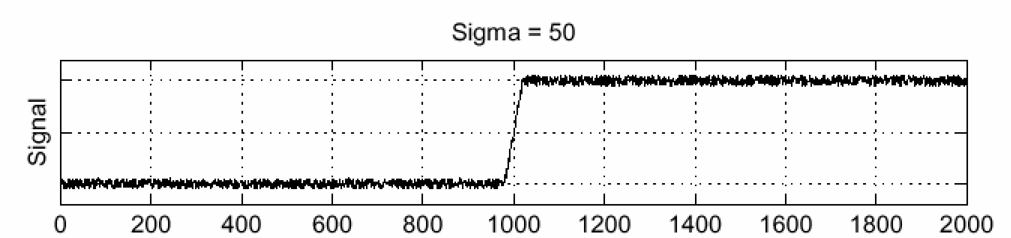
Increasing noise

(this is zero mean additive Gaussian noise)

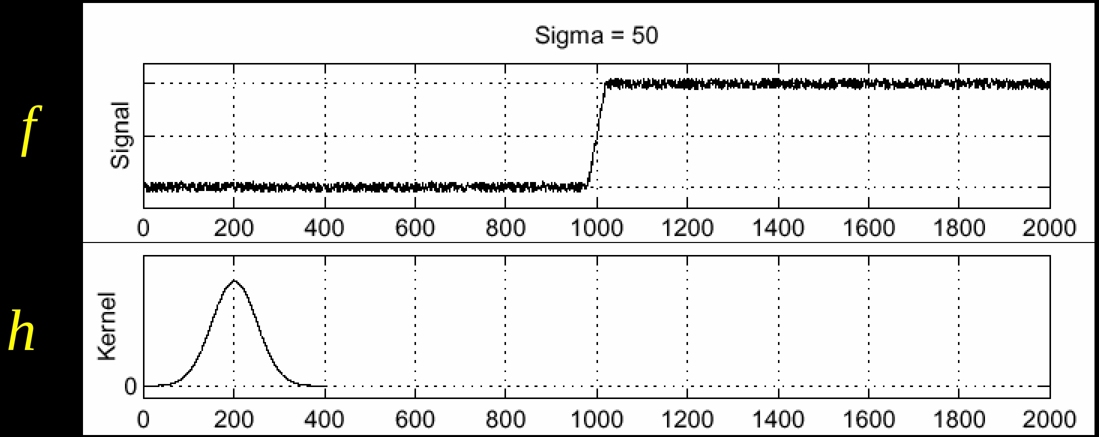
Source: D. Forsyth

Solution: smooth first

f



Solution: smooth first

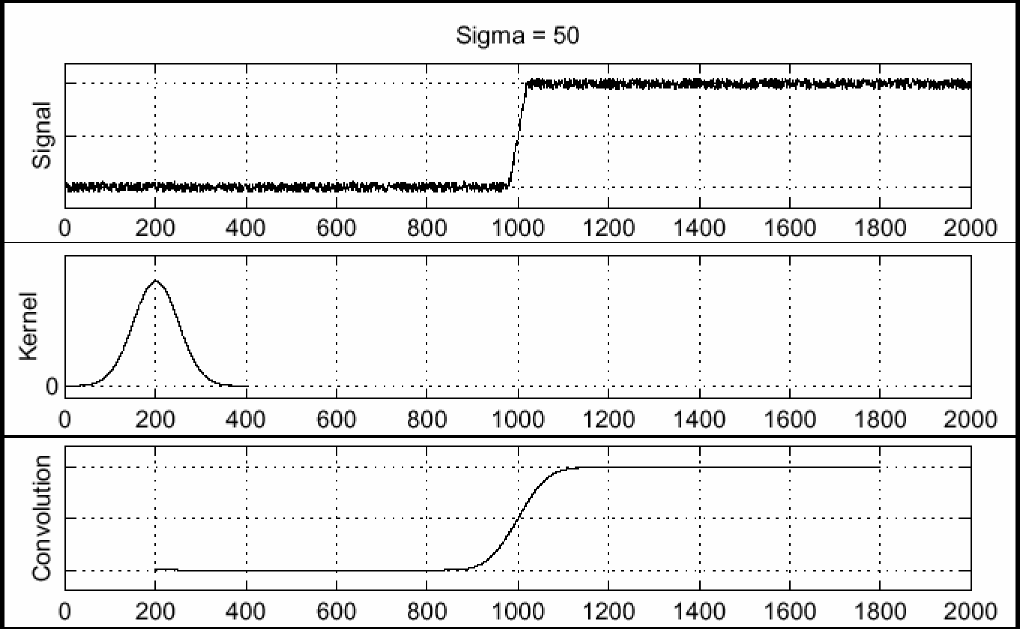


Solution: smooth first

f

h

$h * f$



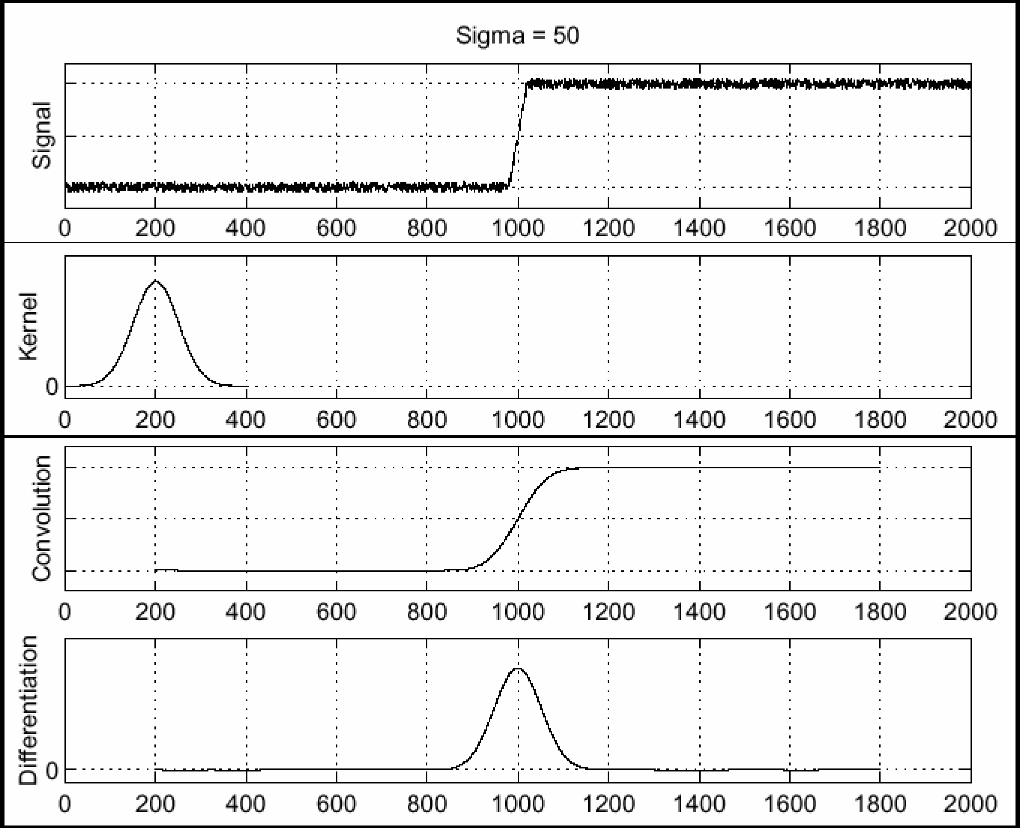
Solution: smooth first

f

h

$h * f$

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



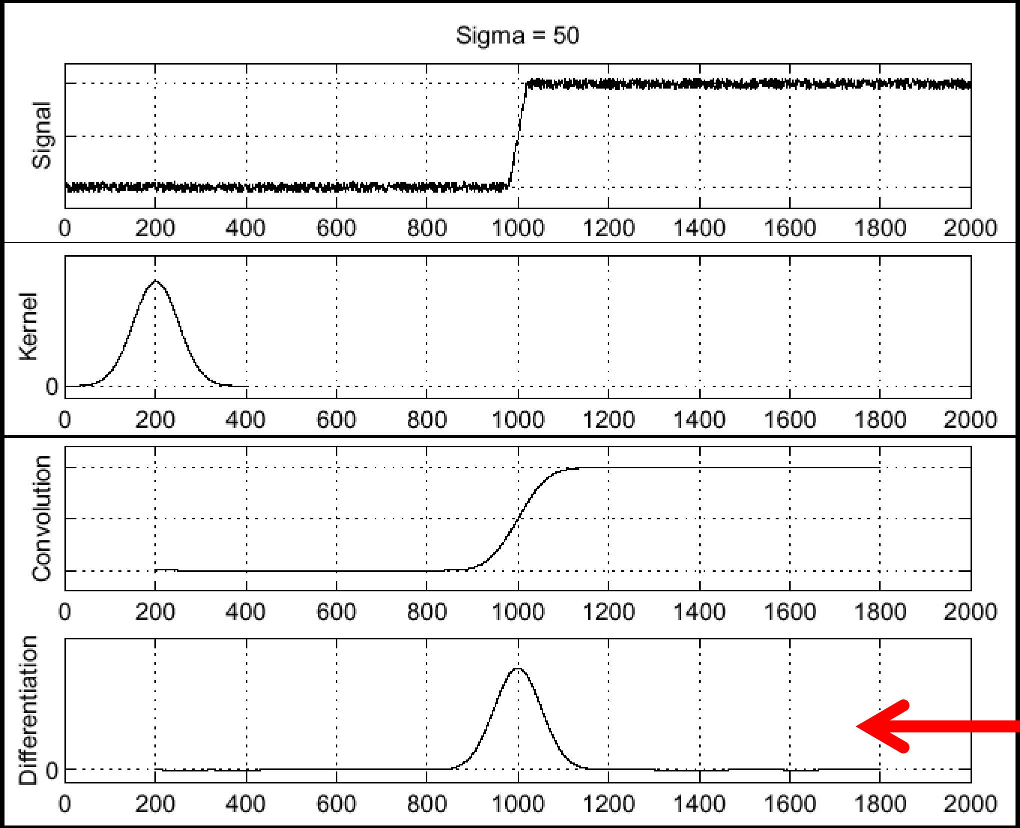
Solution: smooth first

f

h

$h * f$

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



Where is the edge?

Look for peaks

Derivative theorem of convolution

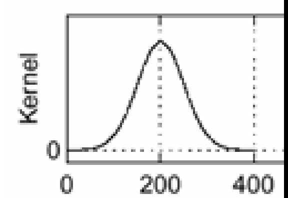
This saves us one operation: $\frac{\partial}{\partial x}(h * f) = (\frac{\partial}{\partial x} h) * f$

Derivative theorem of convolution

This saves us one operation: $\frac{\partial}{\partial x}(h * f) = (\frac{\partial}{\partial x} h) * f$

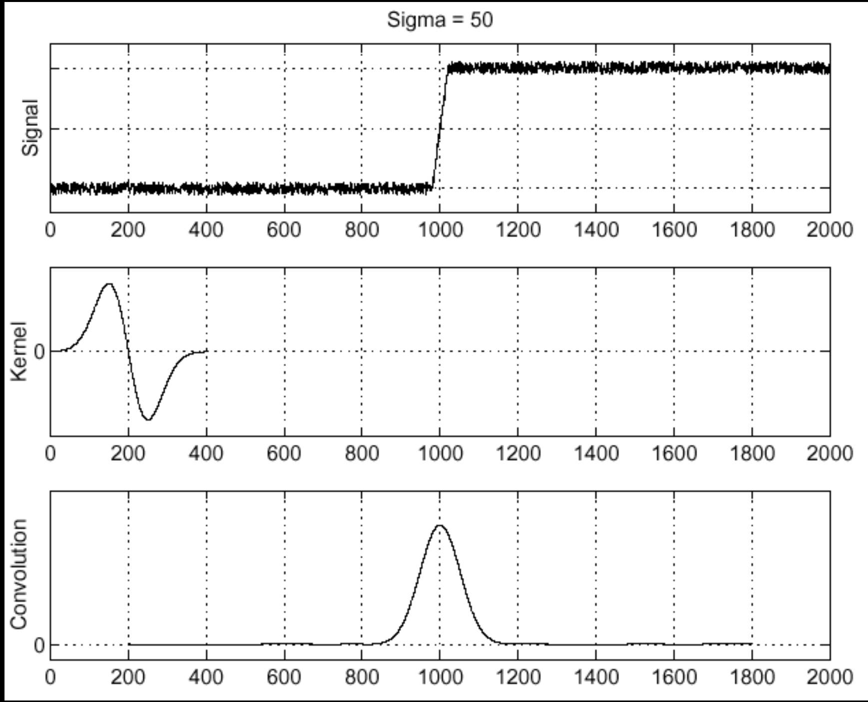
f

h



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

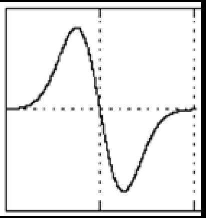
$(\frac{\partial}{\partial x} h) * f$



2^{nd} derivative of Gaussian

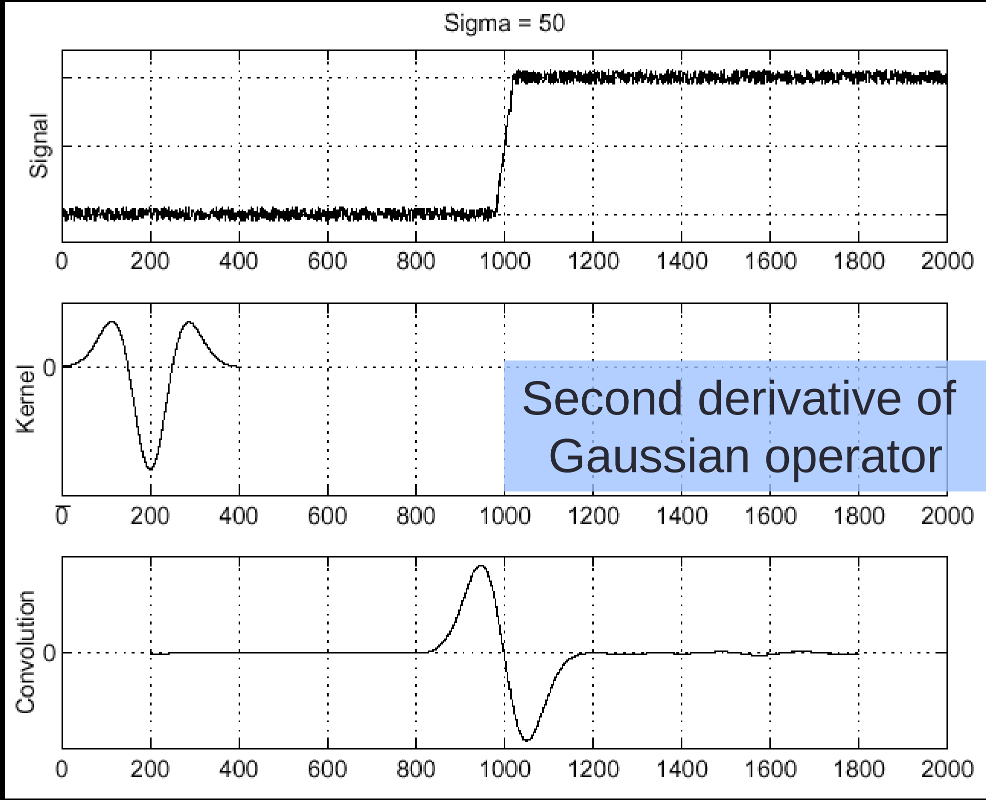
Consider $\frac{\partial^2}{\partial x^2}(h * f)$

f

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

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

Where is the edge? $(\frac{\partial^2}{\partial x^2} h) * f$



Quiz

Which linearity property did we take advantage of to first take the derivative of the kernel and then apply that?

a) associative

b) commutative

c) differentiation

d) (a) and (c)