Introduction to Computer Vision

Michael J. Black

Sept 2009

Lecture 8:
Pyramids and image derivatives

Goals

- Images as functions
- Derivatives of images
- Edges and gradients
- Laplacian pyramids

Code for lecture

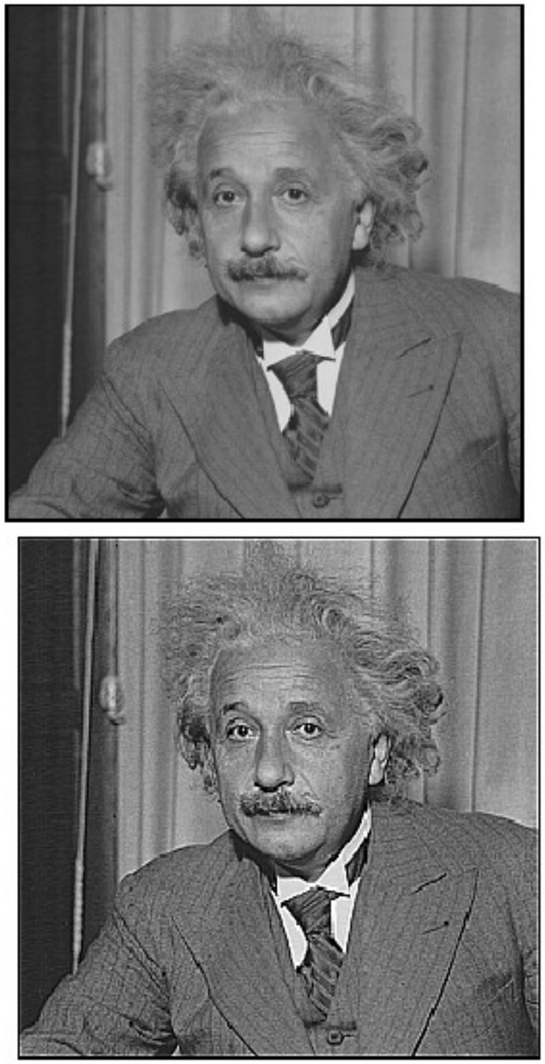
- http://www.cs.brown.edu/courses/cs143/Matlab/lecture5Script.m

- http://www.cs.brown.edu/courses/cs143/Matlab/lecture6featureScript.m

Next week

- I'm at ICCV in Japan
- Monday: Deqing Sun – features and correlation (assignment 1)
- Wednesday: data for assignment 2. important that you attend.
- Friday: Silvia Zuffi – color

Image Filtering

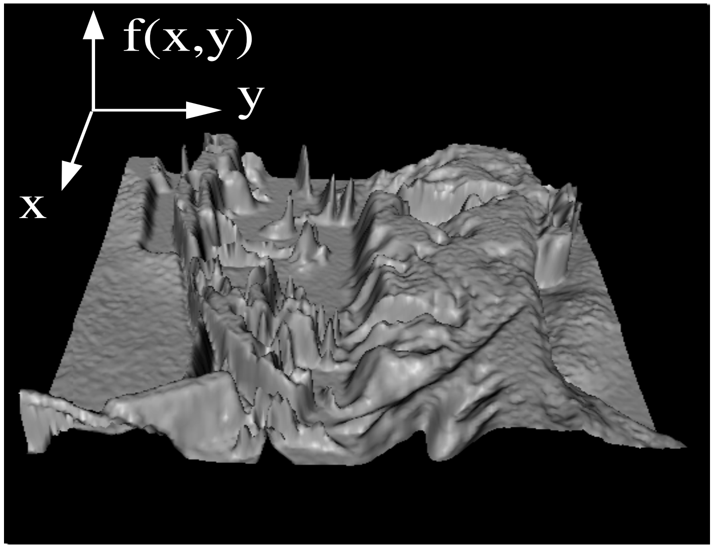


Smoothing and sharpening

Edge detection

Feature detection/search

Images as functions



- Image is a function, *f*, from R^2 to R:
  - $f(x, y)$ gives the image intensity at position (x, y)
  - Realistically, we expect the image only to be defined over a rectangle, with a finite range:
    - f: $[a,b]$ x $[c,d]$ → $[0, 1.0]$

Images as functions

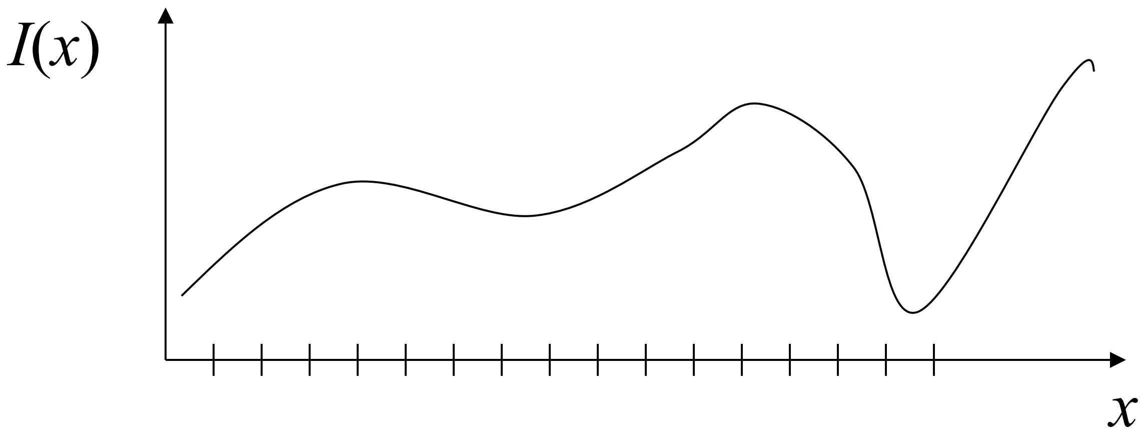
- A color image is just three functions pasted together. We can write this as a "vector-valued" function:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Source: S. Seitz

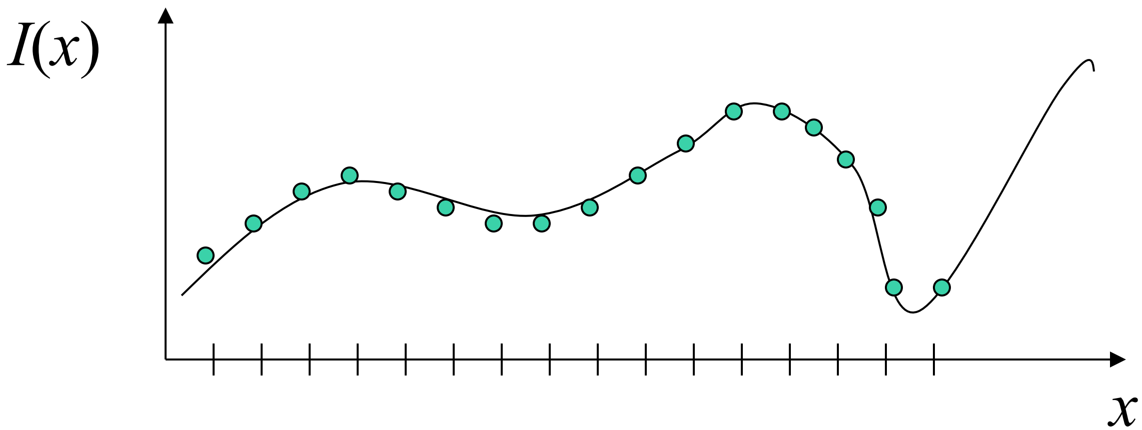
Images as Functions

- Images are a discretely sampled representation of a continuous signal



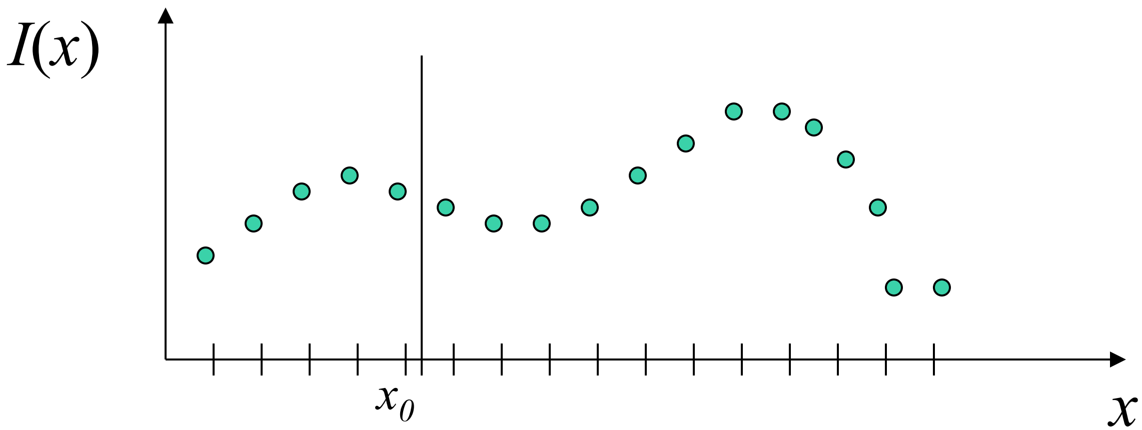
Images as Functions

- Images are a discretely sampled representation of a continuous signal



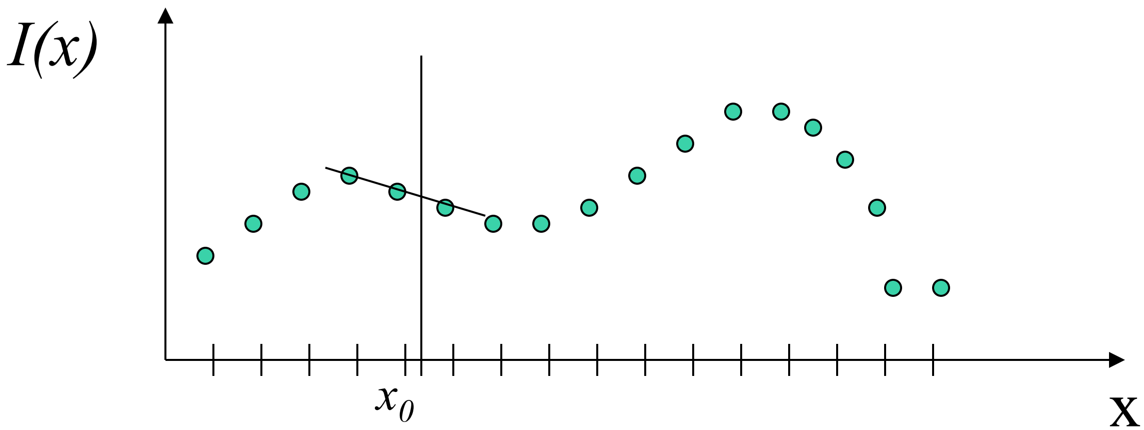
Images as Functions

What if I want to know $I(x_0+dx)$ for small $dx<1$?



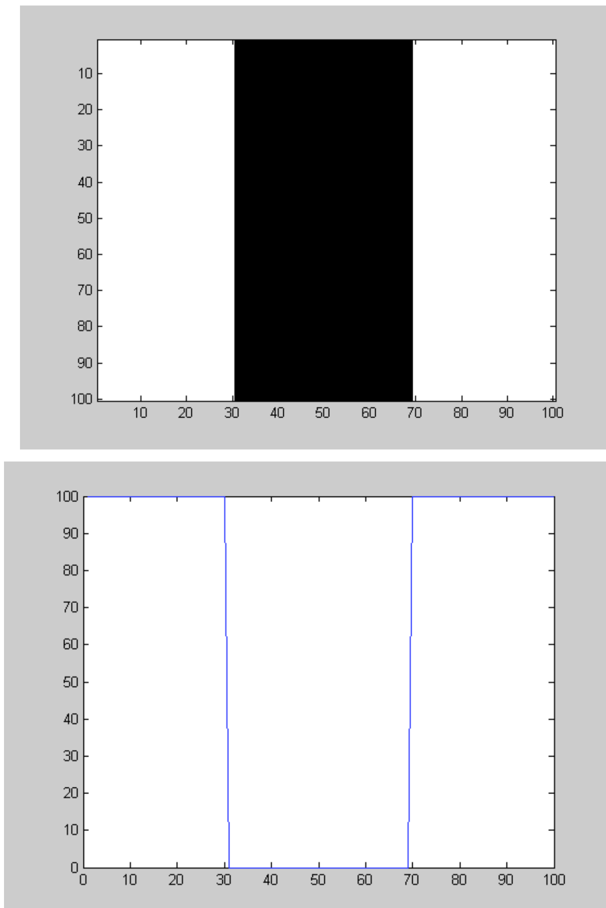
Taylor Series Approximation

$$I(x_0 + dx) \approx I(x_0) + dx \frac{\mathrm{d}}{\mathrm{d}x} I(x_0) + \varepsilon$$

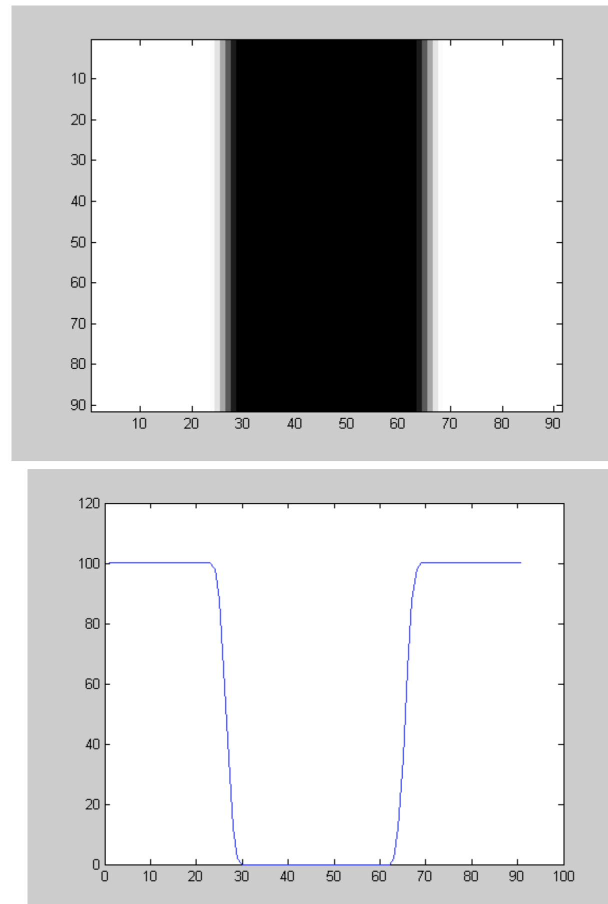


Locally linear approximation to the function using an estimate of the slope.
How do we compute the partial derivatives of an image?

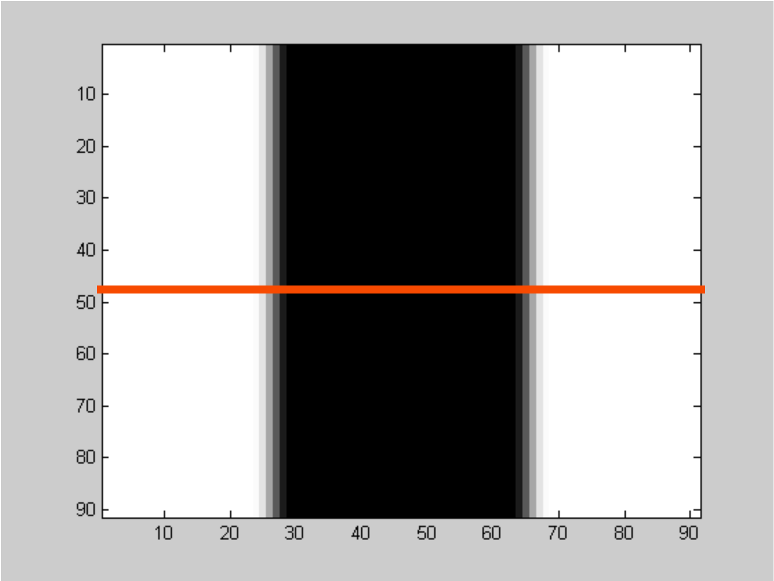
Simple Signal

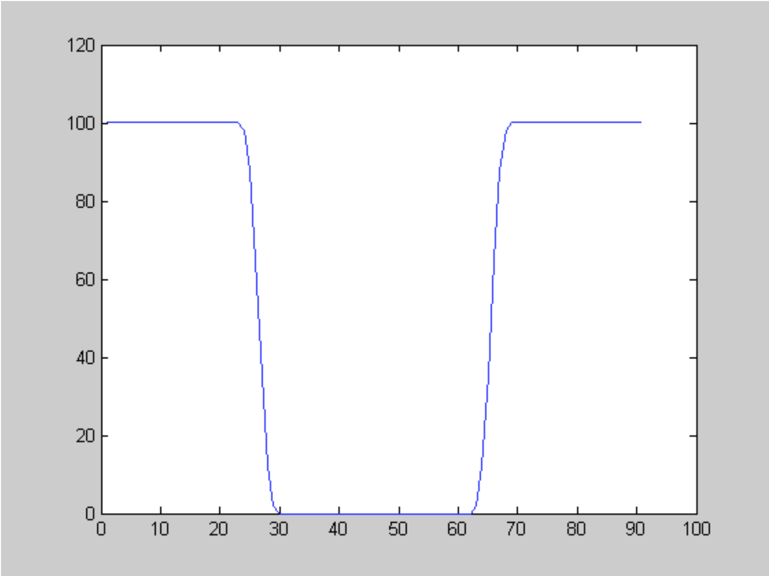


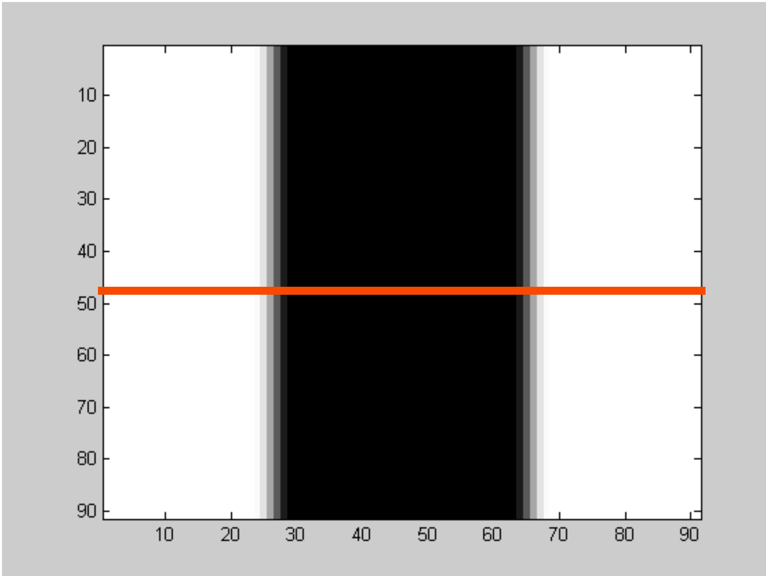
Discontinuous

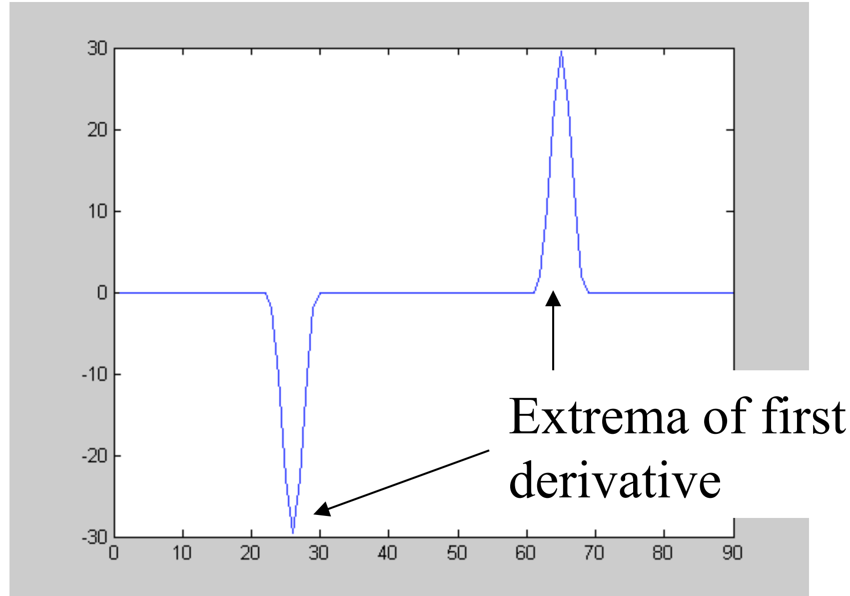


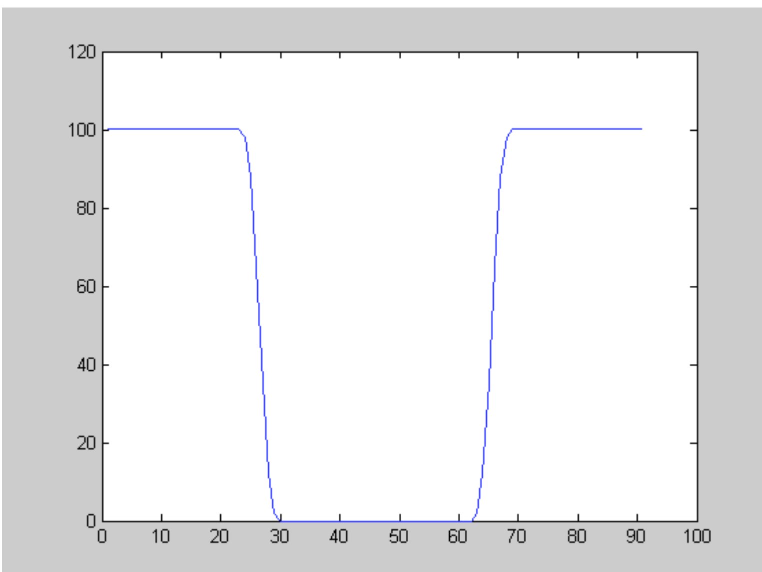
Smoothed with Gaussian


10
20
30
40
50
60
70
80
90
10
20
30
40
50
60
70
80
90


120
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
90
100

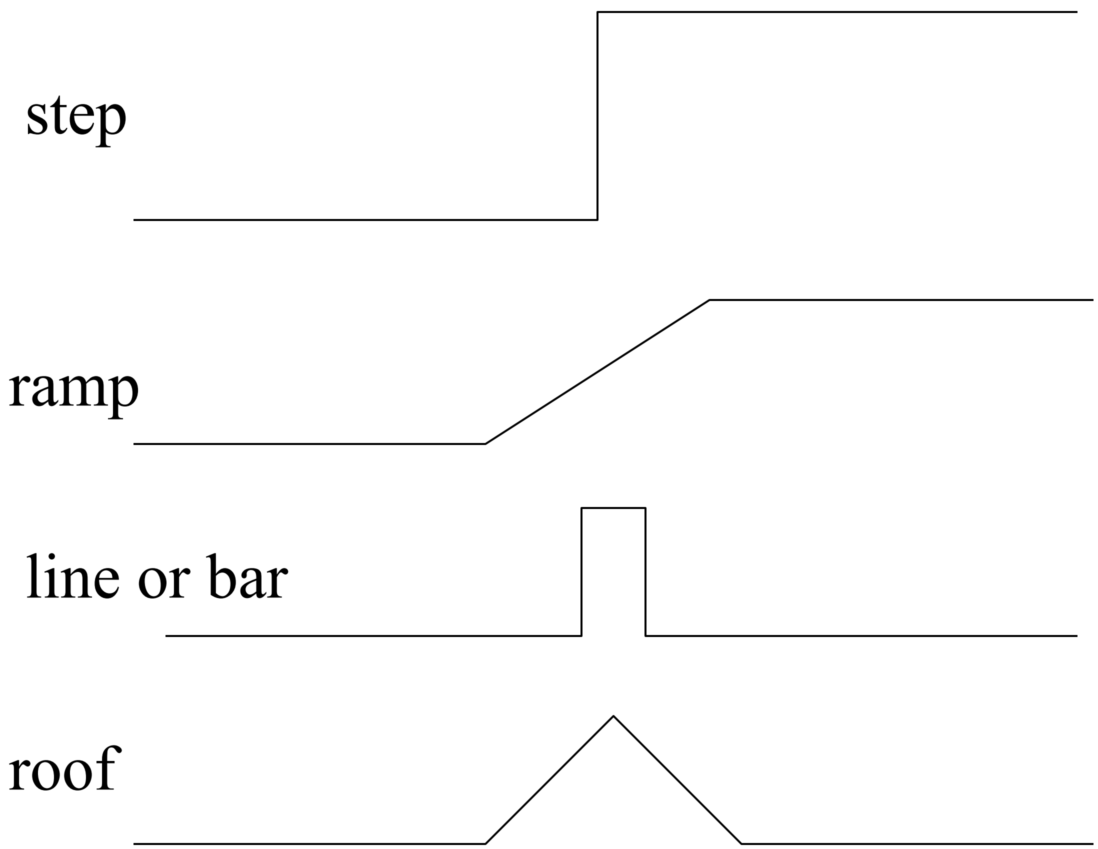



Extrema of first derivative

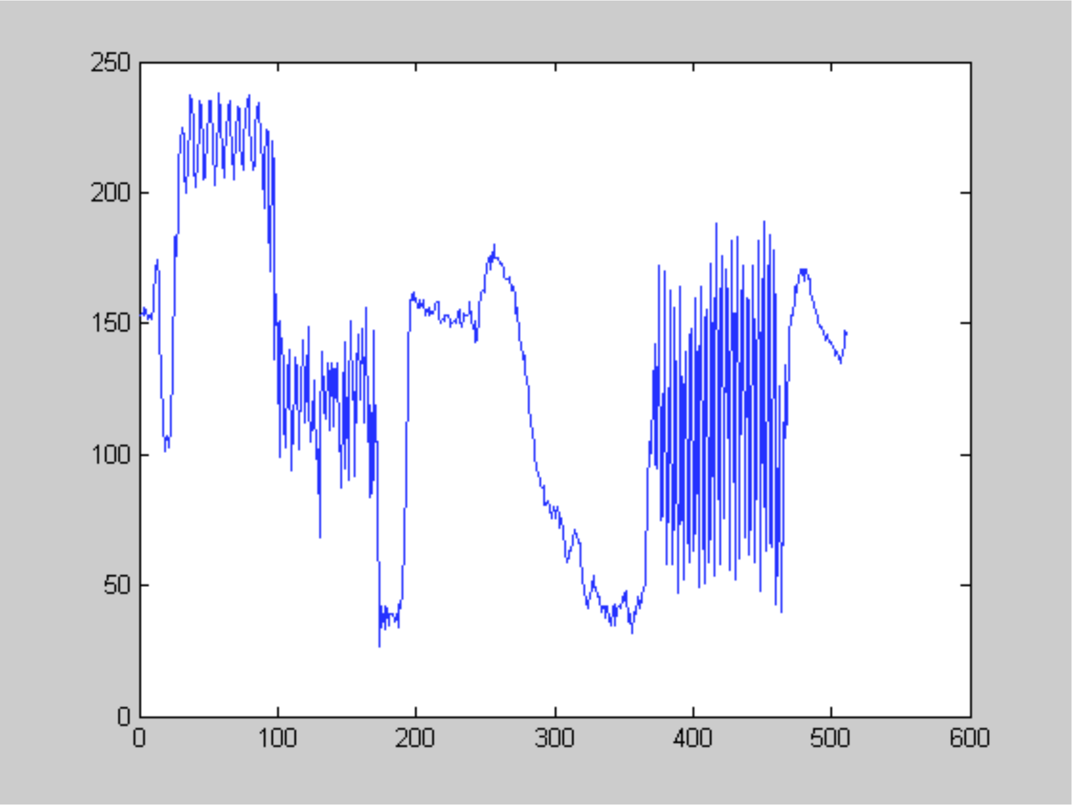


What are “edges” (1D)

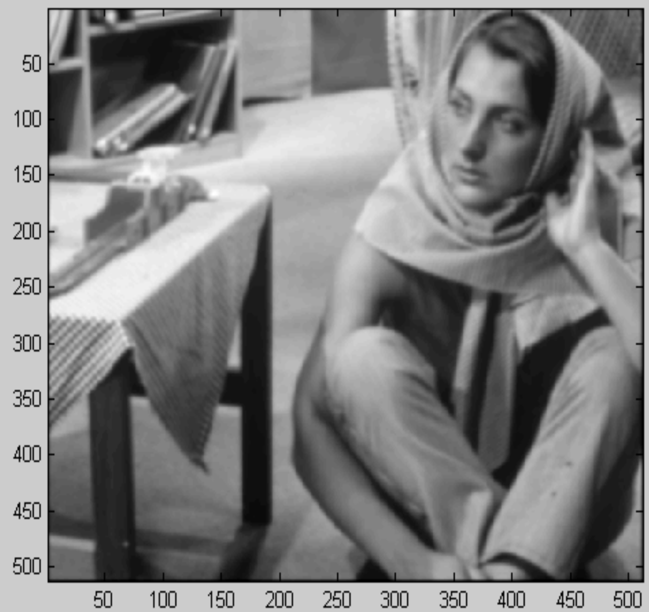
Idealized:

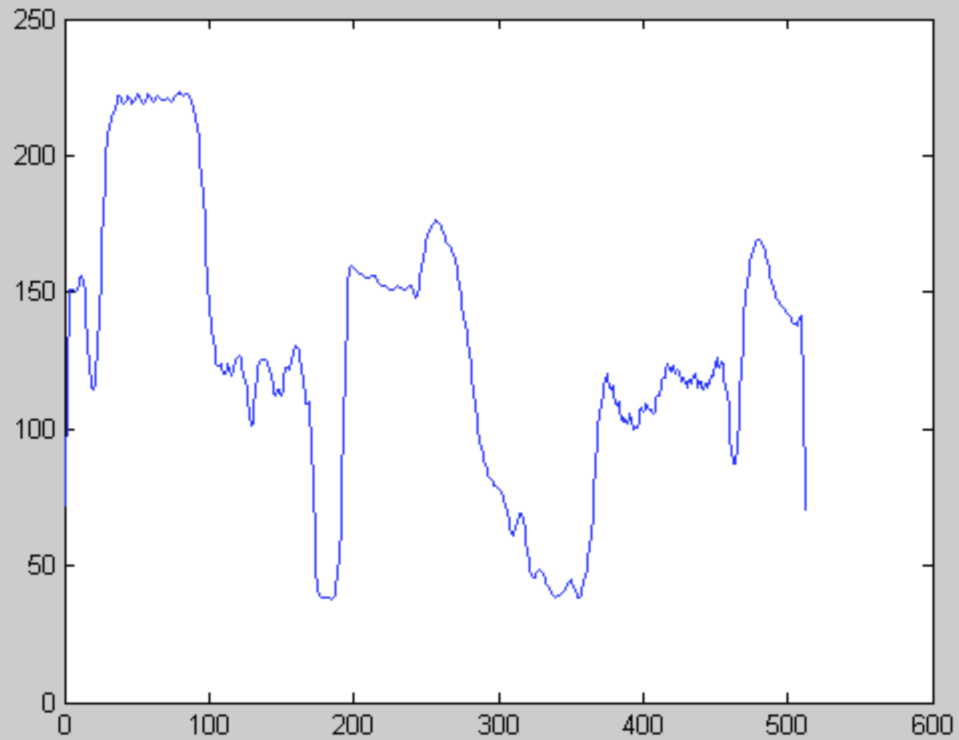


Actual 1D profile



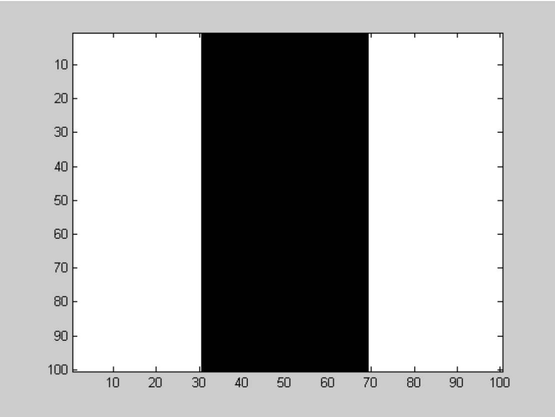
Smoothed with a Gaussian

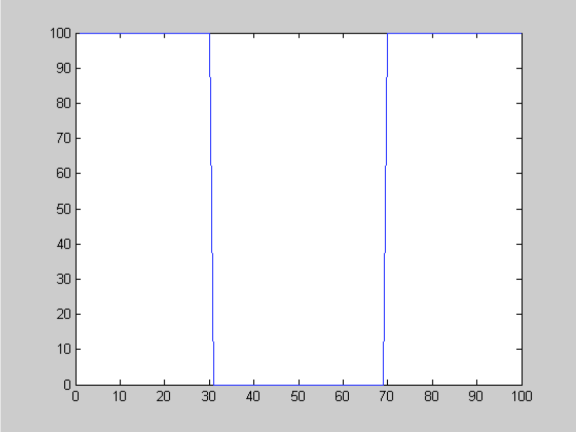




Edges

- Correspond to fast changes
  – Where the magnitude of the derivative is large

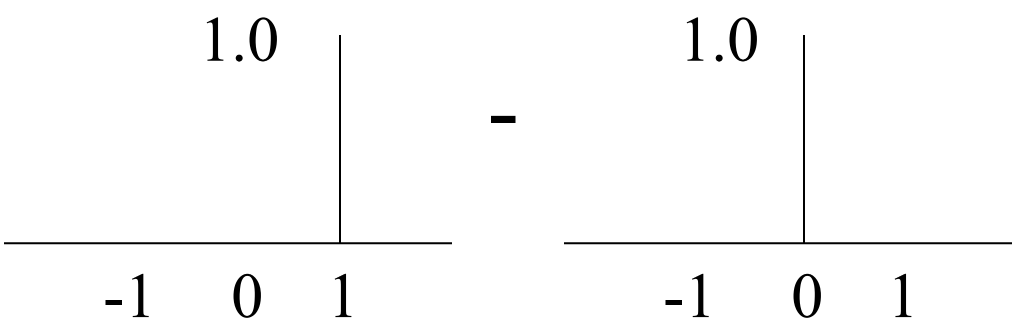




Compute Derivatives

$$I_x(x) = \lim_{dx \to 0} \frac{I(x+dx) - I(x)}{dx} \approx I(x+1) - I(x)$$

We can implement this as a linear filter:



[-1 1]

Or [-1 0 1]

symmetric

Partial Derivatives

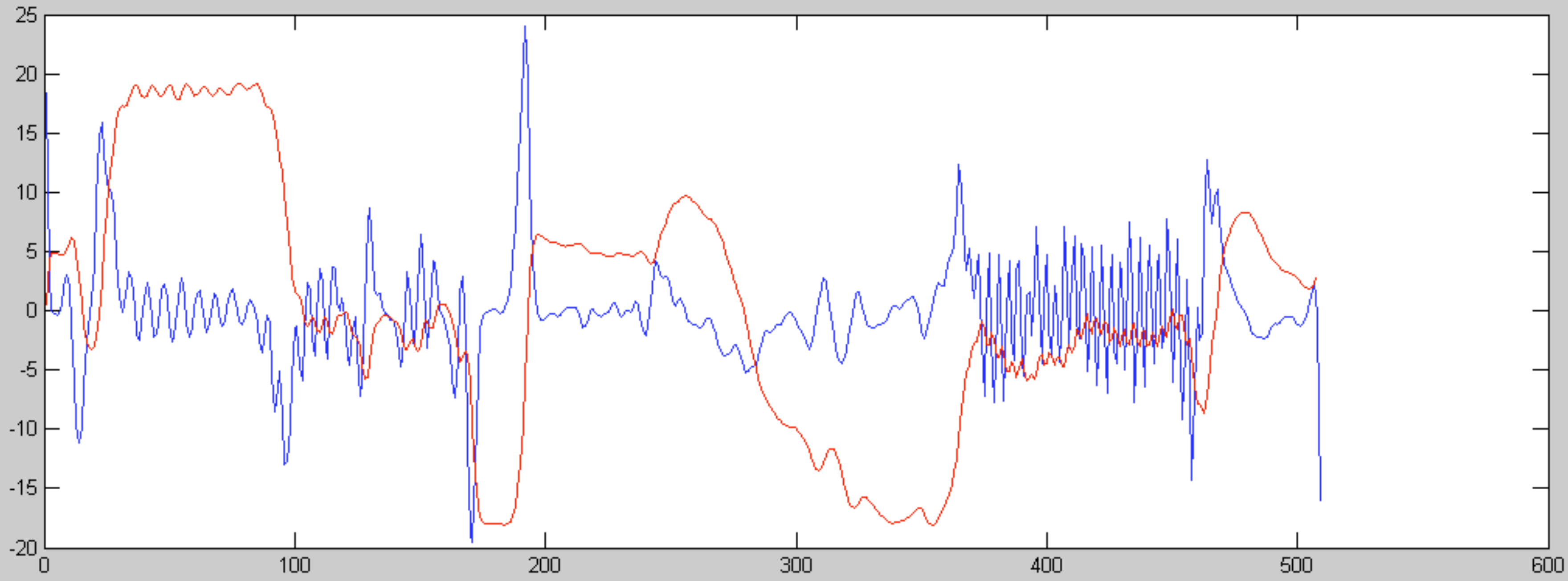
$$\frac{\partial}{\partial x} I(x,y) = I_x \approx I \otimes D_x, \quad \frac{\partial}{\partial y} I(x,y) = I_y \approx I \otimes D_y$$

- Often approximated with simple filters:

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

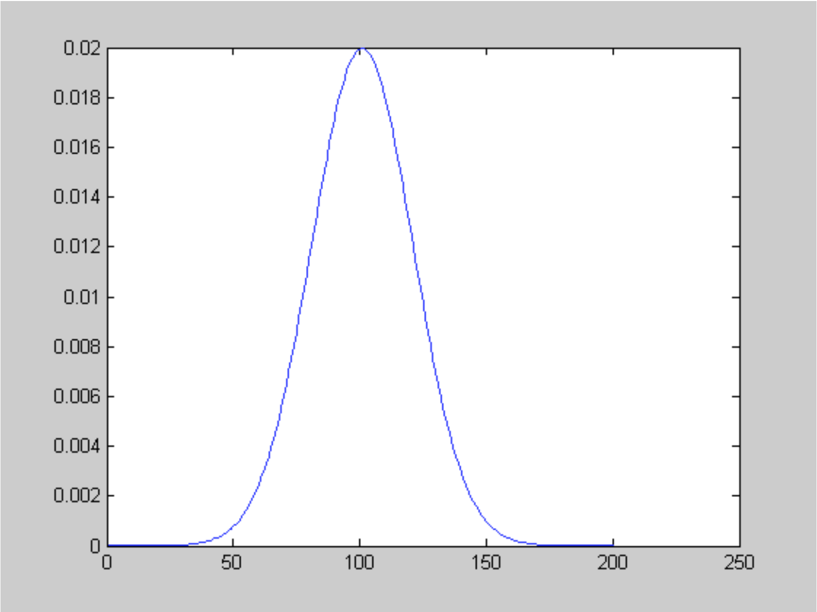
Finite differences

Barbara signal and derivatives



Derivatives and Smoothing

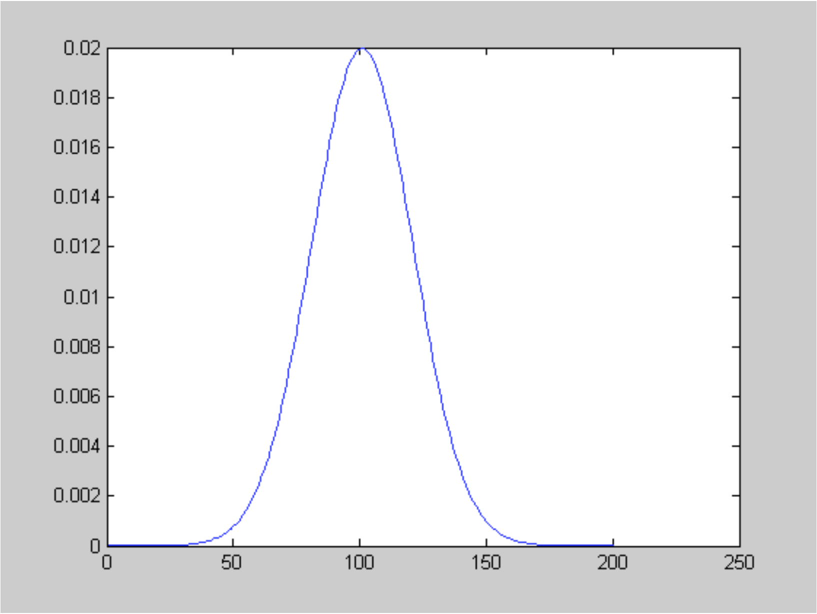
$$D_x \otimes (G \otimes I) = \qquad ?$$



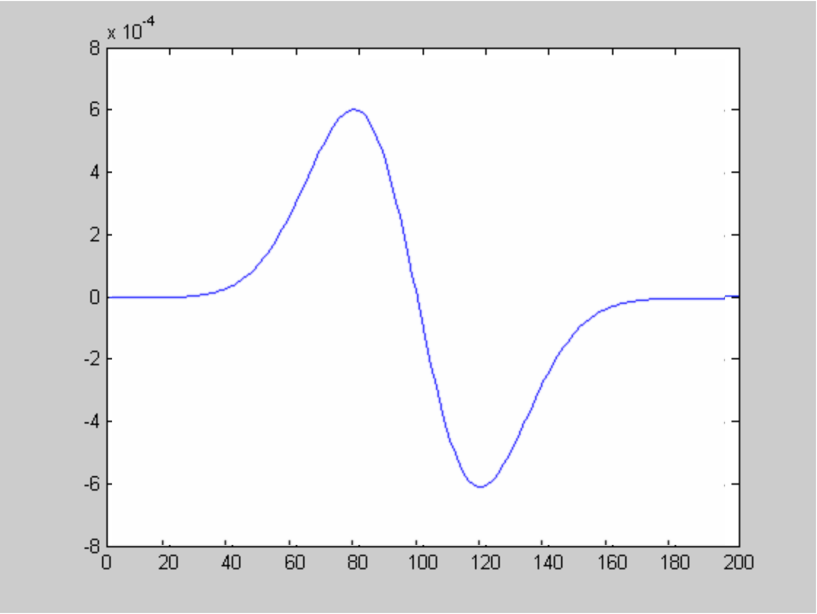
G

Derivatives and Smoothing

$$D_x \otimes (G \otimes I) = (D_x \otimes G) \otimes I$$



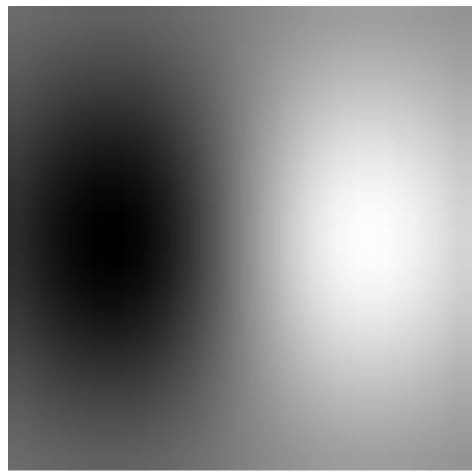
G

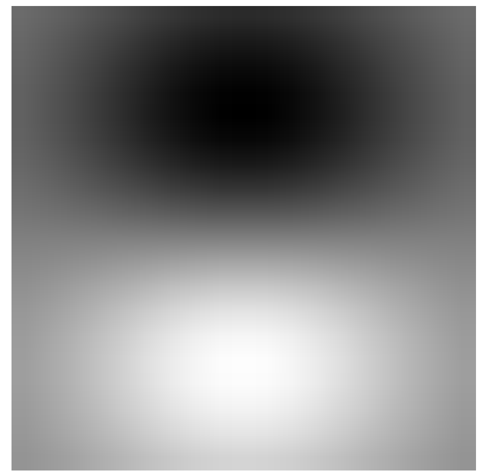


dG

In 2D

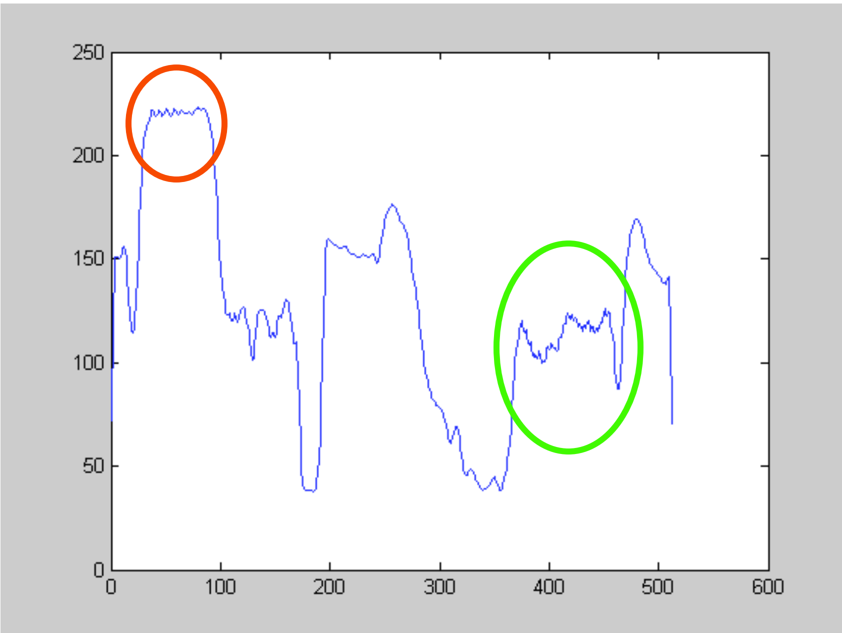
$$D_x \otimes (G \otimes I) = (D_x \otimes G) \otimes I$$



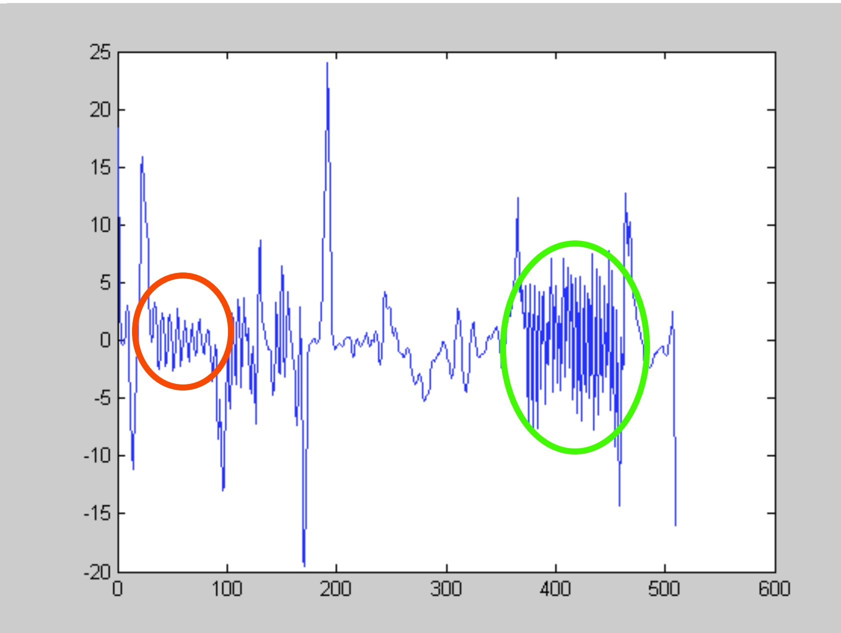


Compare with [-1 0 1] filters.

1D Barbara signal



Smoothed Signal



First Derivative

Note the “amplification” of small variations.

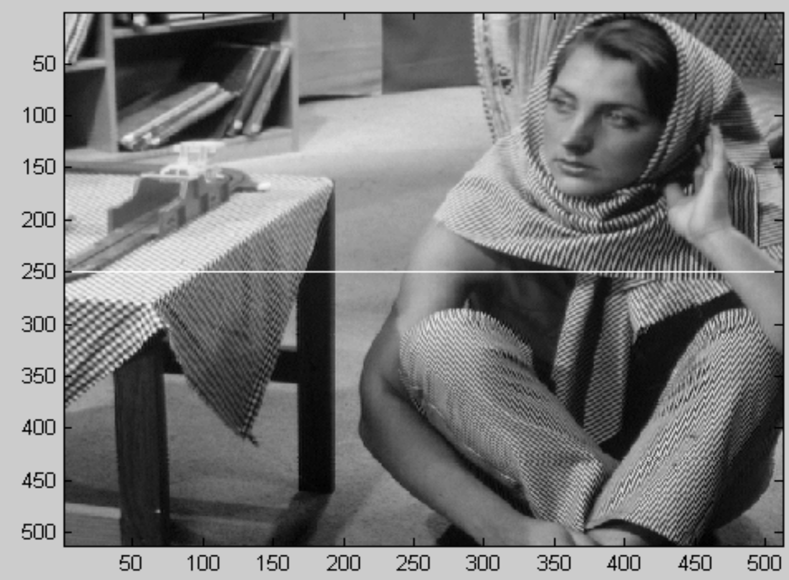
How can we “detect” edges?

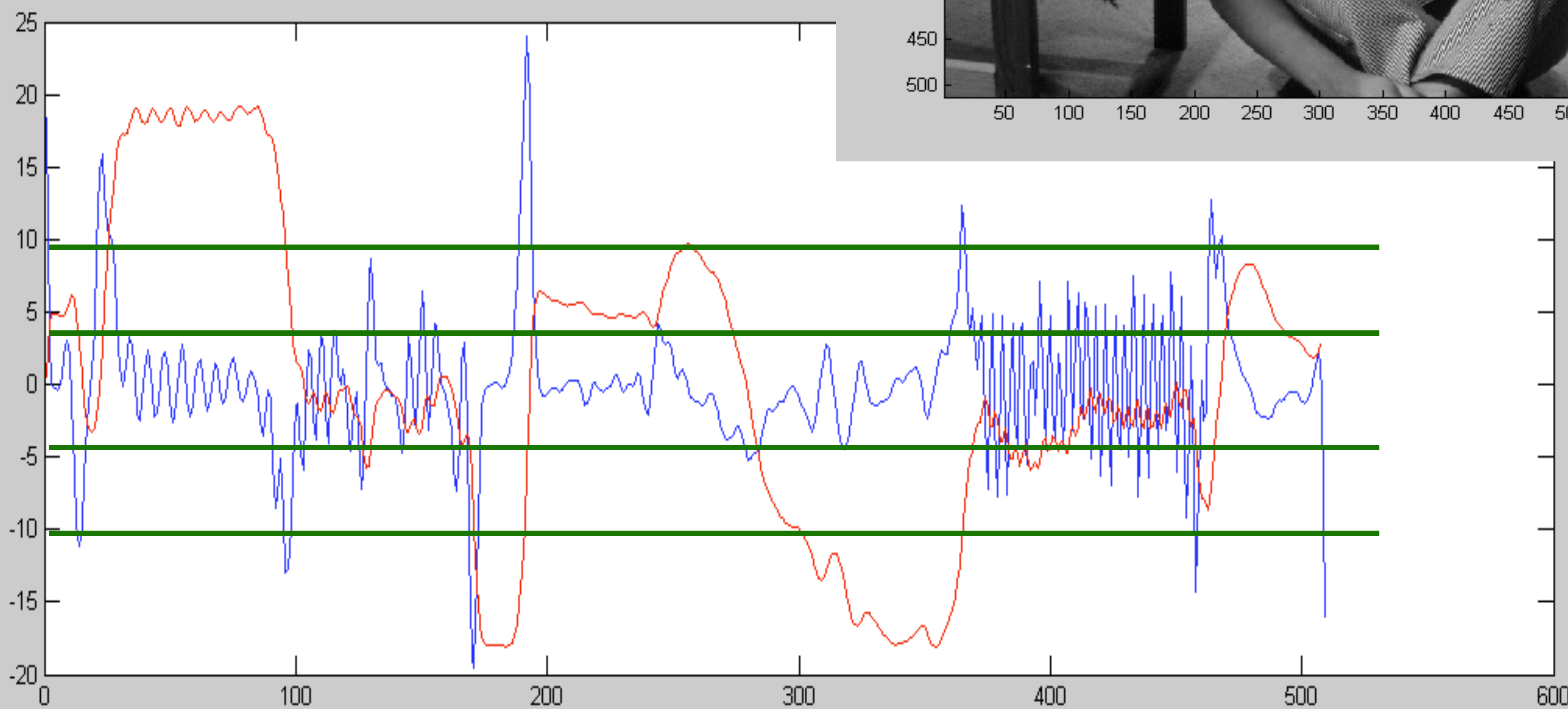
Find the peak in the derivative.

Two issues:

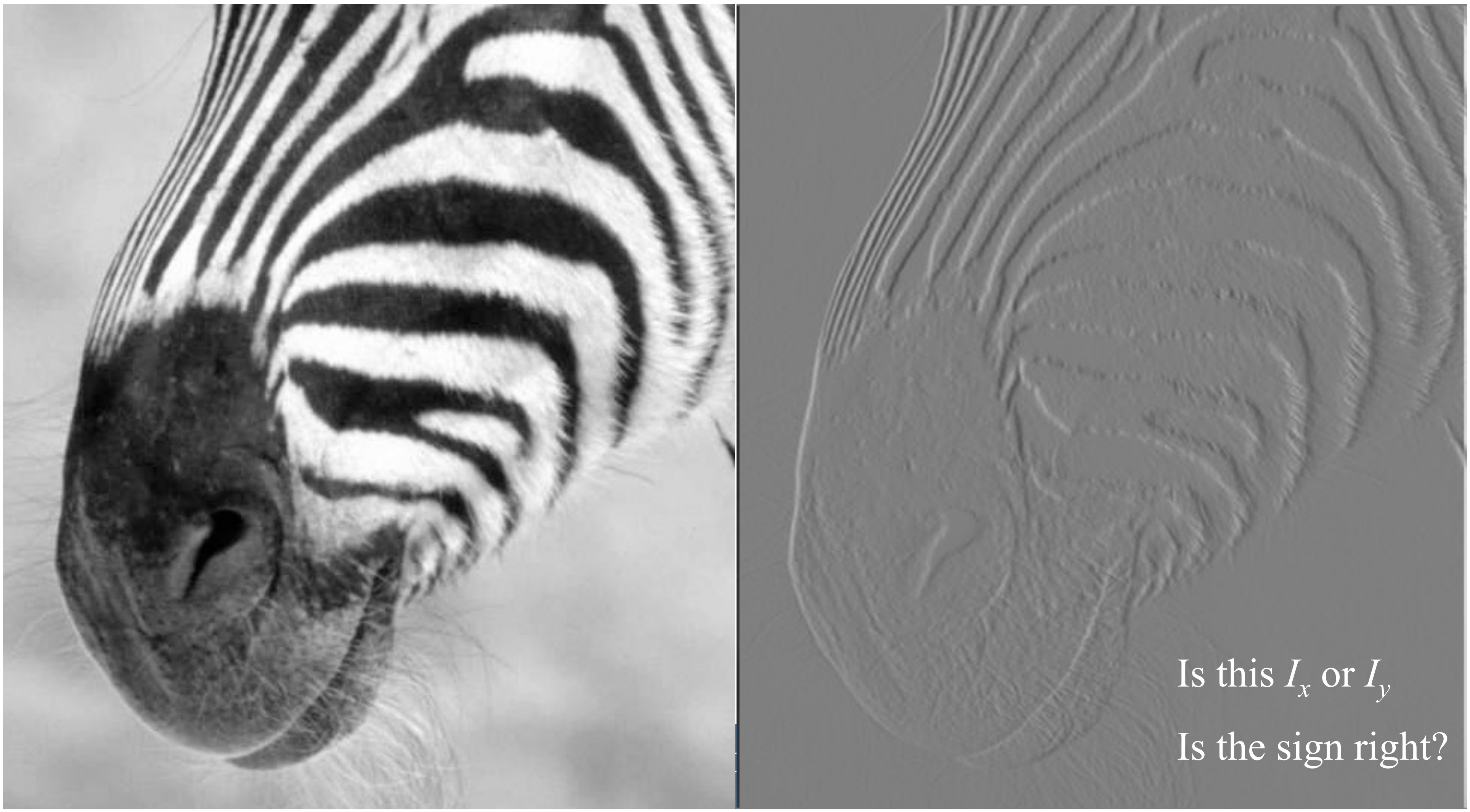
- Should be a local maximum.
- Should be “sufficiently” high.

Thresholding the Derivative?

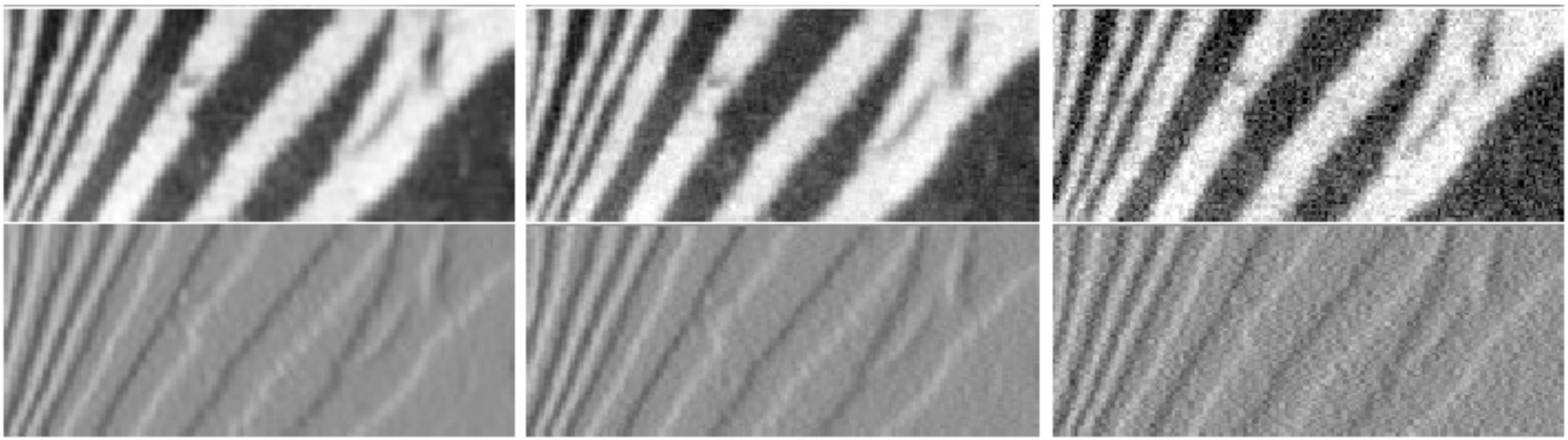




Finite differences

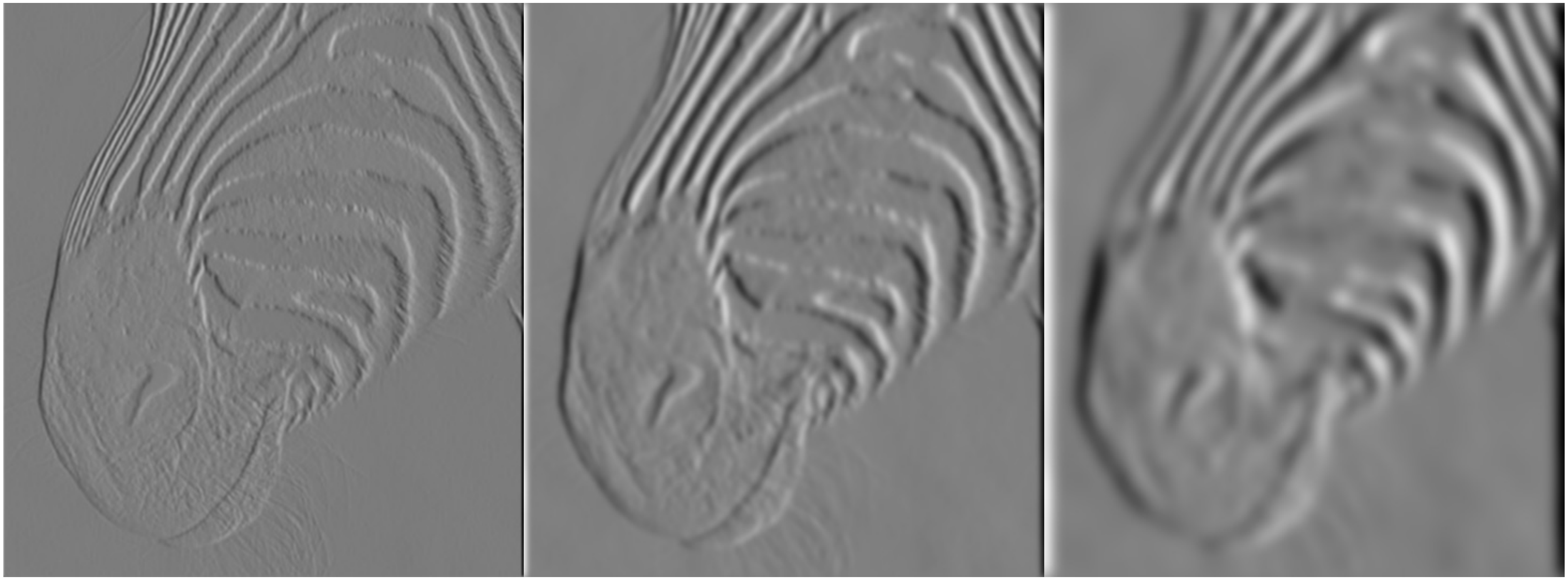


Finite differences responding to noise



Increasing noise →
(this is zero mean additive Gaussian noise)

Ponce & Forsyth



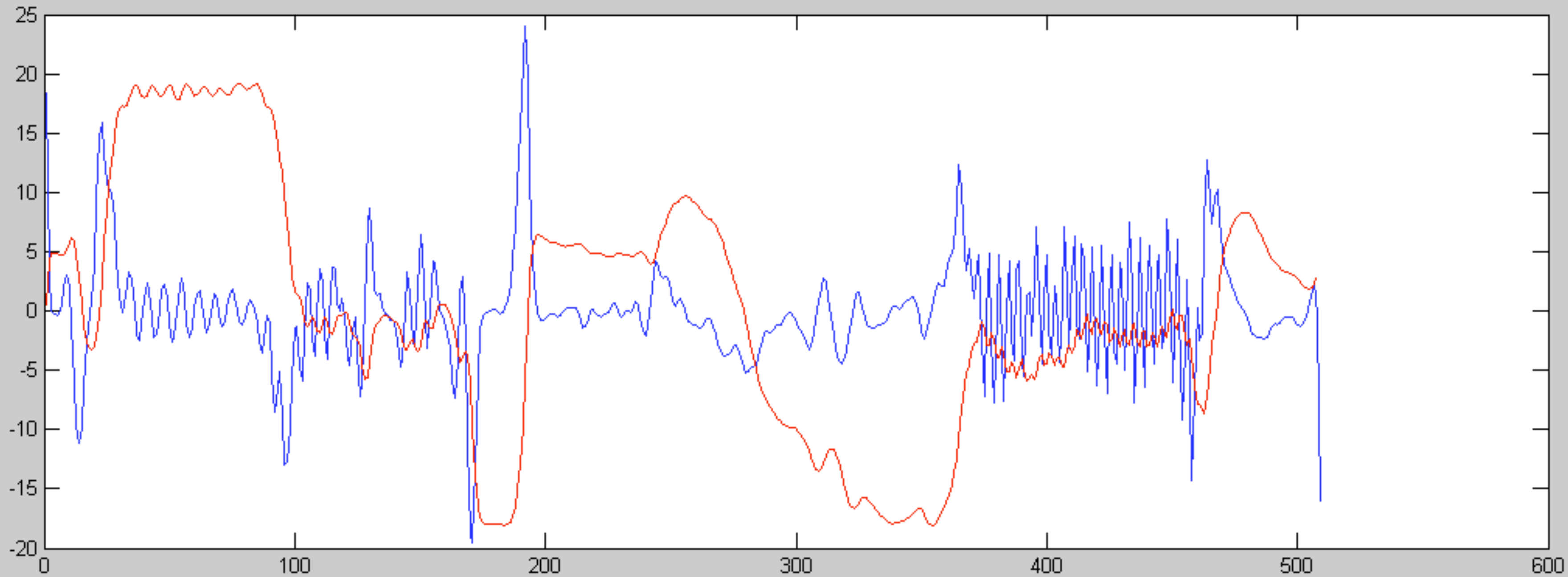
1 pixel 3 pixels 7 pixels

The scale of the smoothing filter affects derivative estimates, and also the semantics of the edges recovered.

Note: strong edges persist across scales.

Ponce & Forsyth

Barbara signal and derivatives



What happens if we now take the derivative of the derivative?

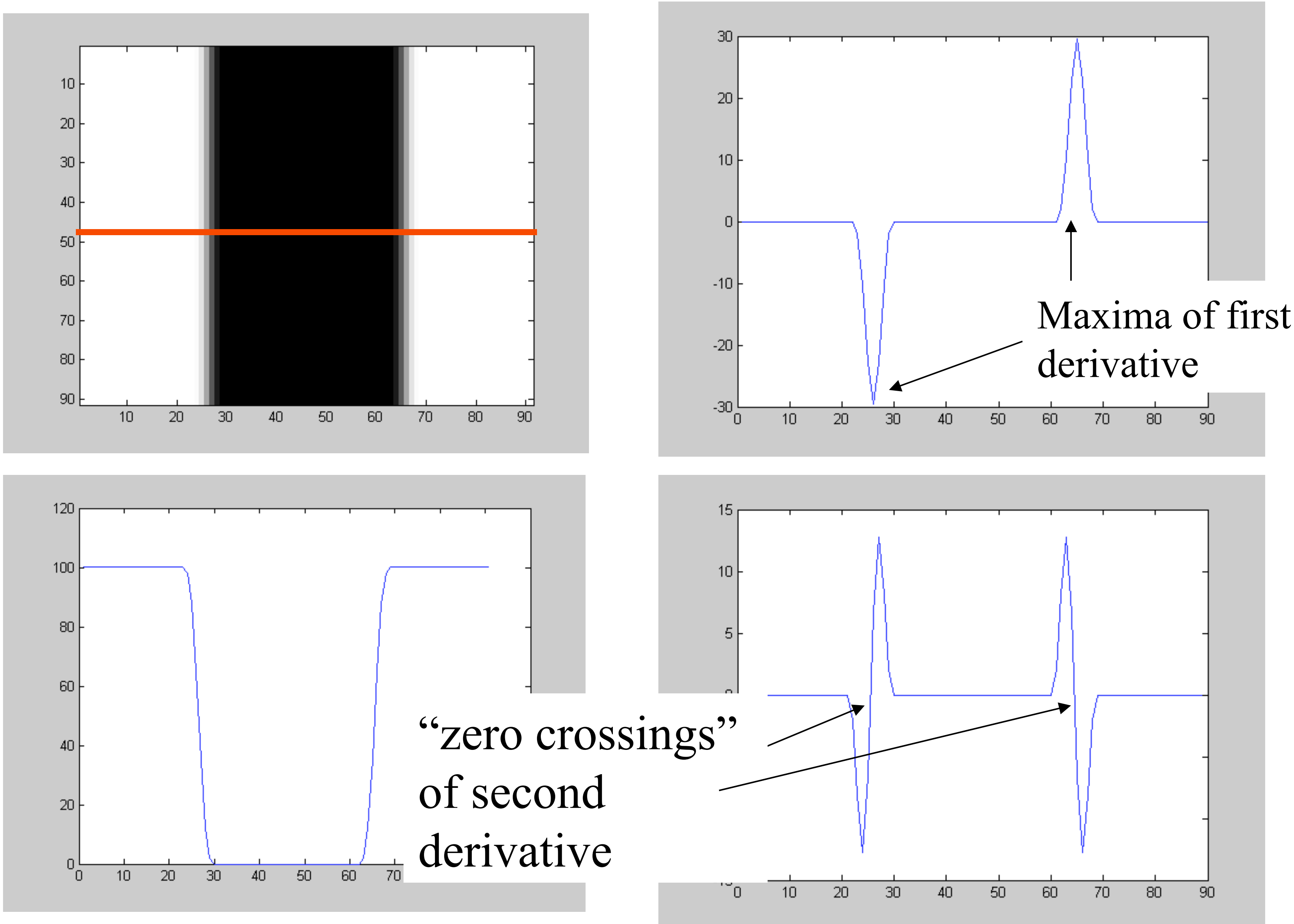
Second Derivative

$$I''(x) = \lim_{dx \to 0} \frac{I'(x+dx) - I'(x)}{dx} \approx I'(x+1) - I'(x)$$

$$= I(x+2) - 2I(x+1) + I(x)$$

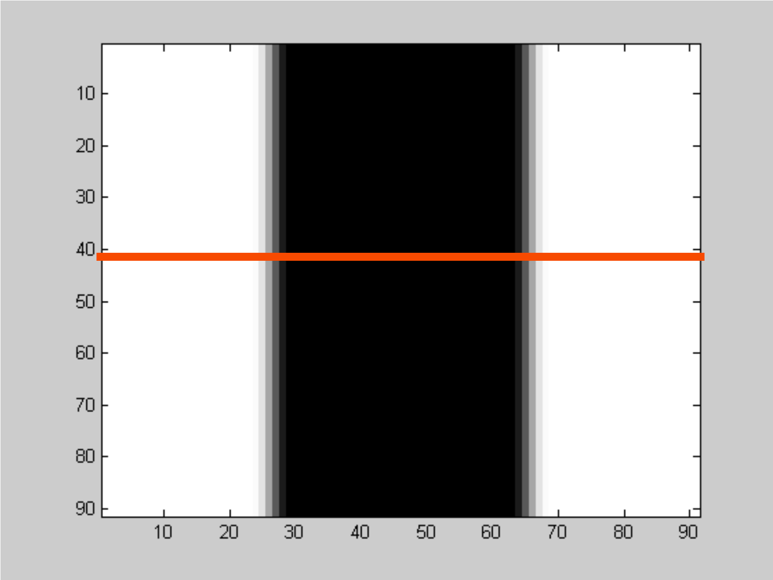
Filter kernel?

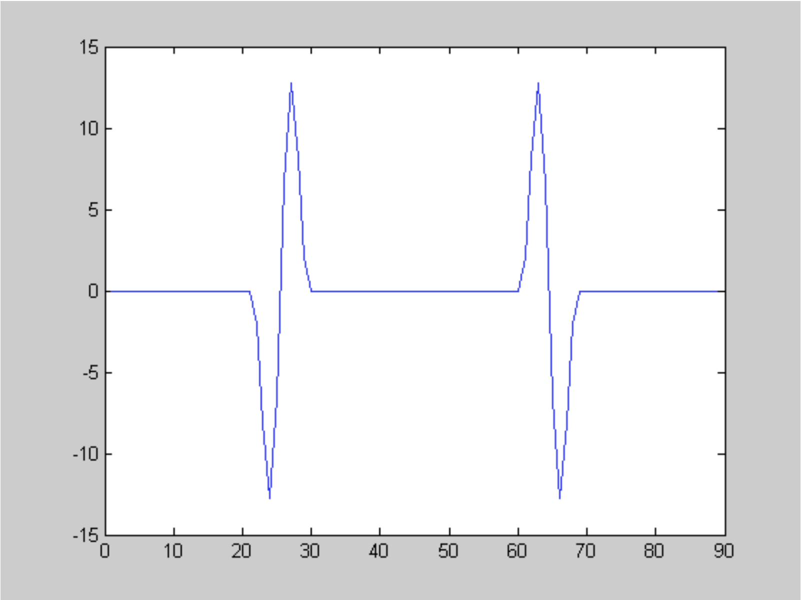
[1 -2 1]



Maxima of first derivative
"zero crossings" of second derivative

Step Edge





The zero-crossings of the second derivative tell us the location of edges.

The Laplacian

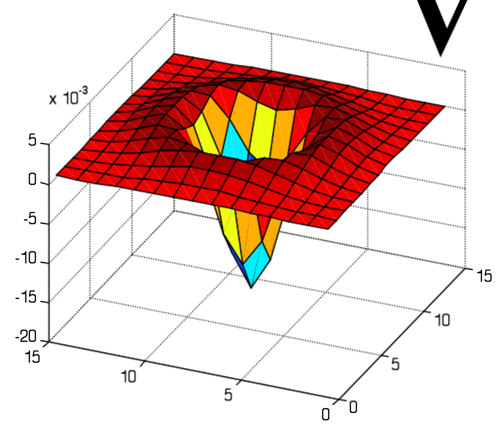
$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

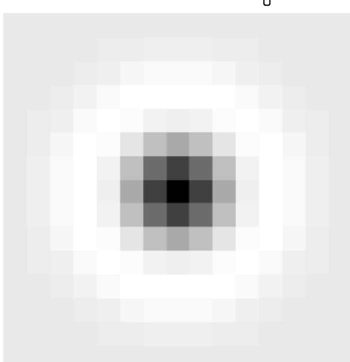
Just another linear filter.

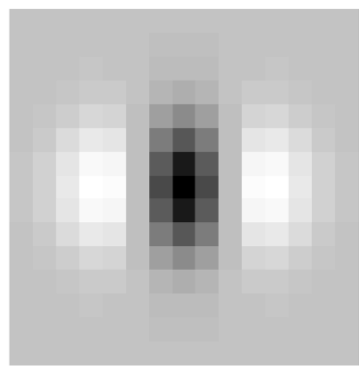
$$\nabla^2 (I(x,y) \otimes G(x,y)) = \nabla^2 G(x,y) \otimes I(x,y)$$

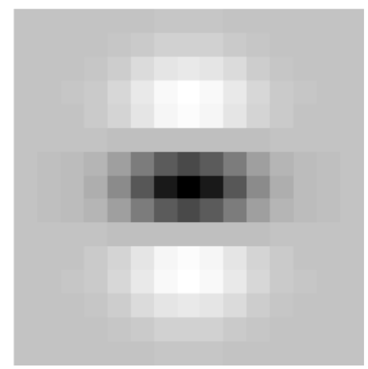
The Laplacian

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





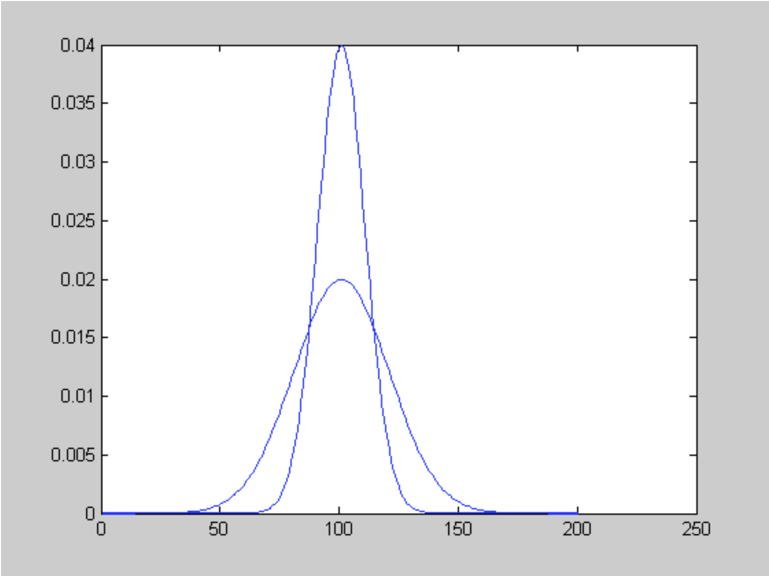


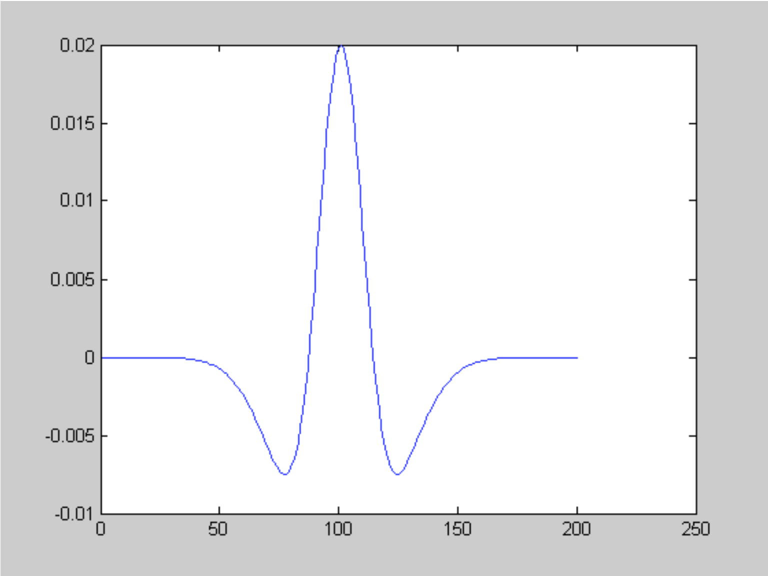


“center-surround” “Mexican hat”

Approximating the Laplacian

- Difference of Gaussians at different scales.

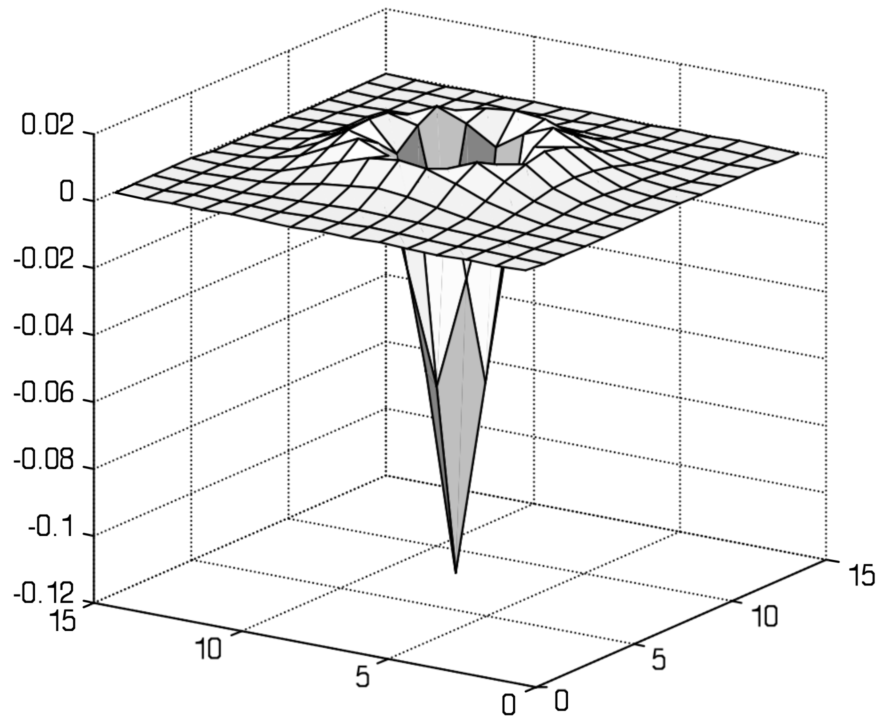




Approximating the Laplacian

- Difference of Gaussians at different scales.

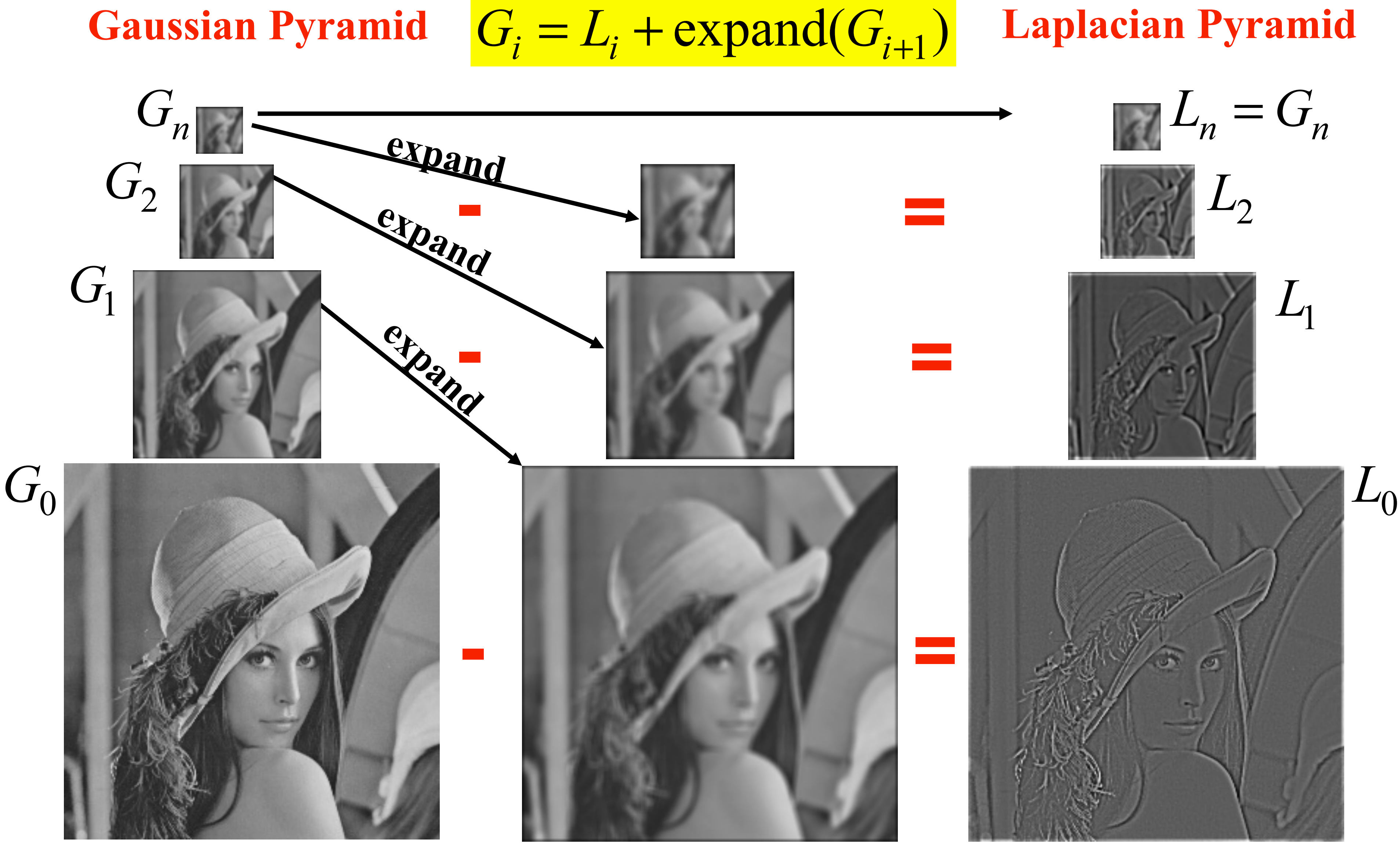
```
DoG=fspecial('gaussian',15,2)-…
        fspecial('gaussian',15,1);
surf(DoG)
```



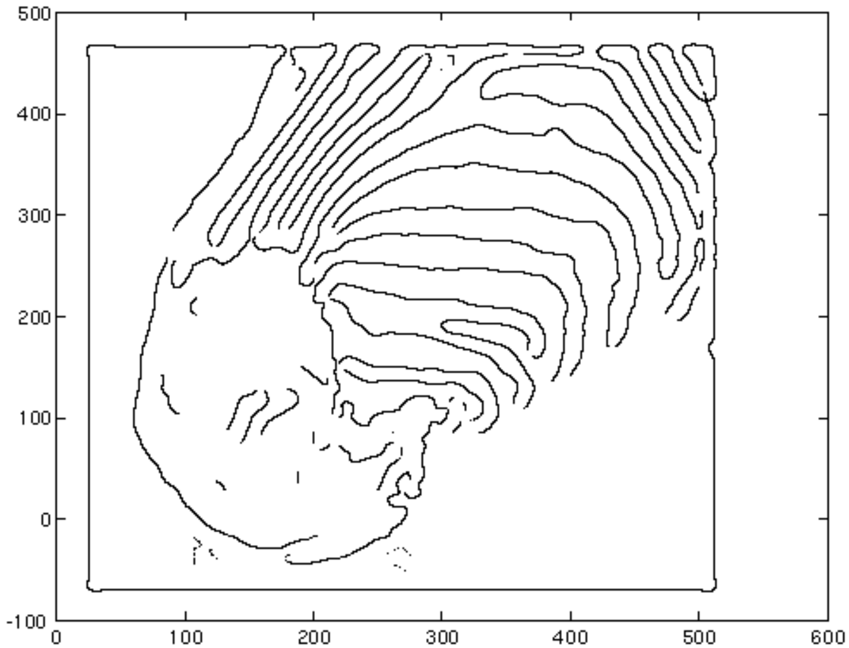
The Laplacian Pyramid

$$L_i = G_i - \text{expand}(G_{i+1})$$

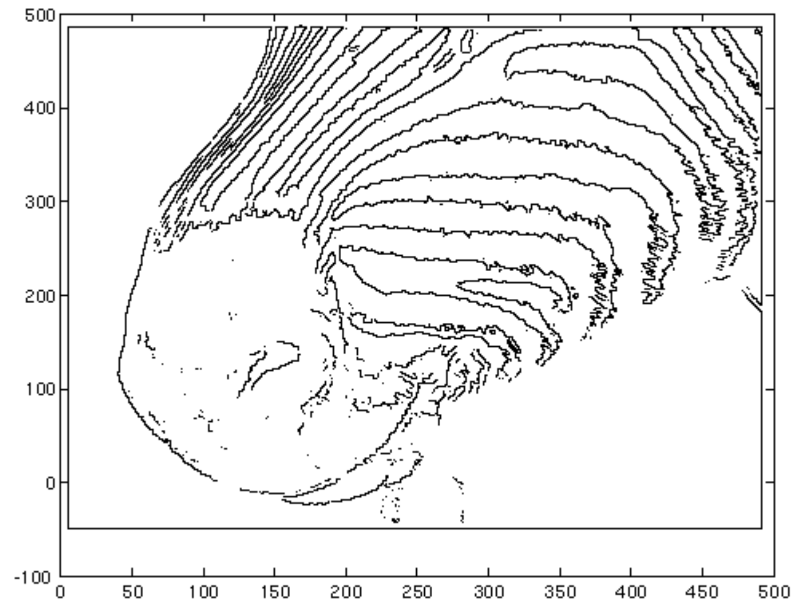
$$G_i = L_i + \text{expand}(G_{i+1})$$



LoG zero crossings
(where the filter response changes sign)



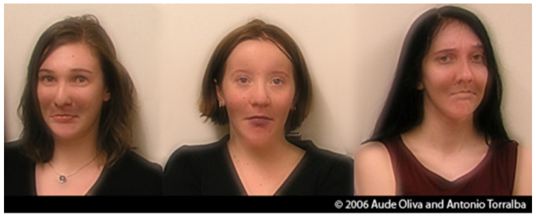
sigma=4



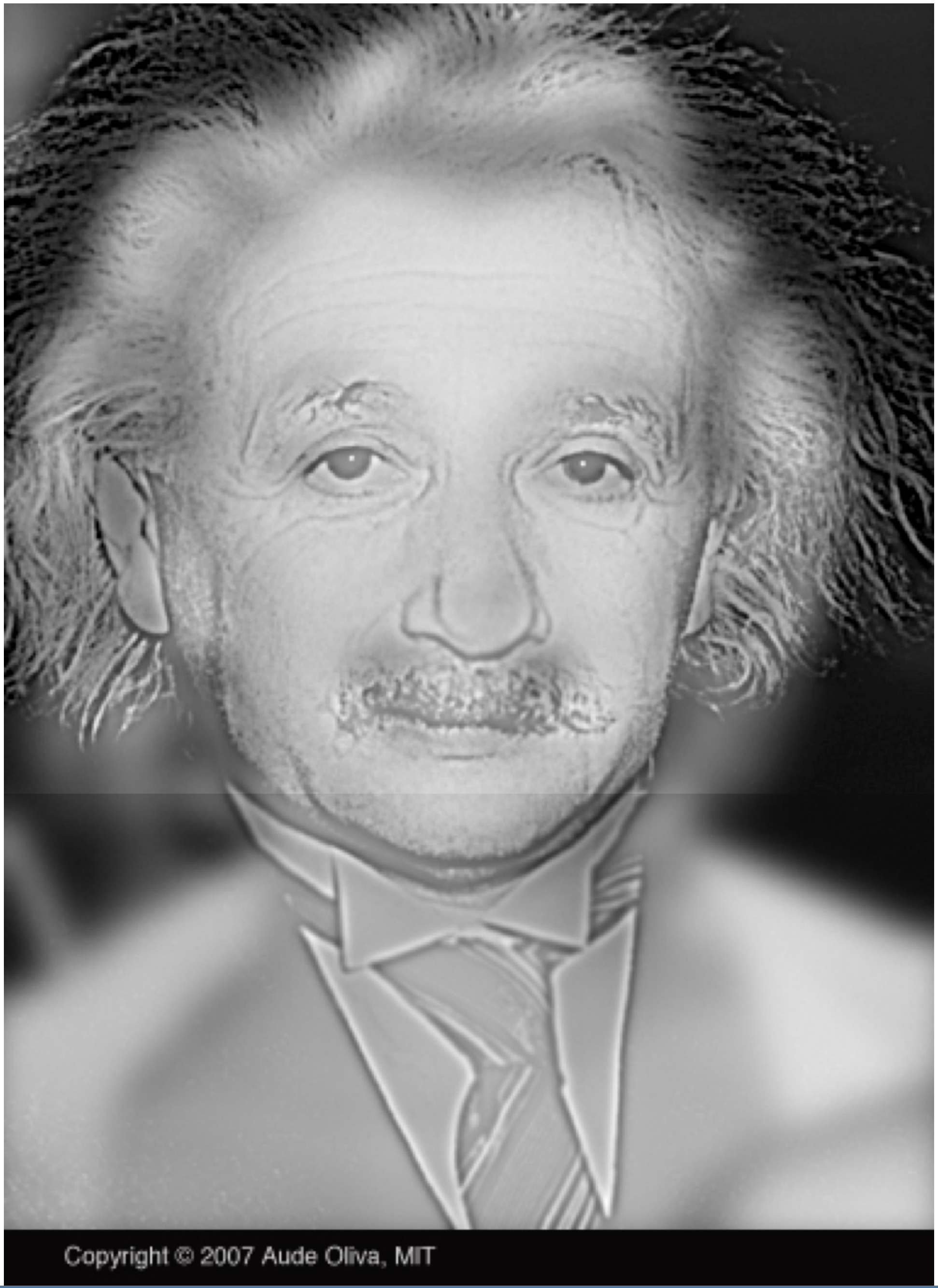
sigma=2

Ponce & Forsyth

http://cvcl.mit.edu/hybridimage.htm



http://cvcl.mit.edu/hybridimage.htm


Copyright © 2007 Aude Oliva, MIT