Laplacian and Blob Detection

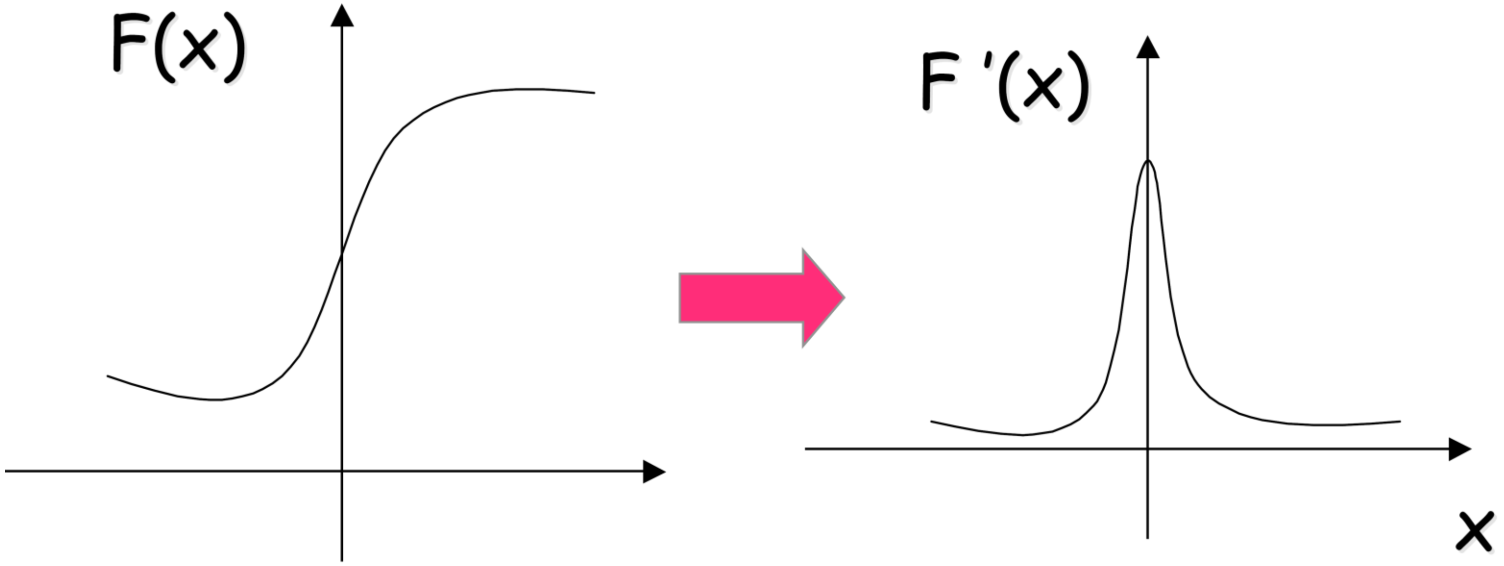
CS 4391 Introduction Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Recall: First Derivative Filters

- Sharp changes in gray level of the input correspond to “peaks or valleys” of the first-derivative of the input signal



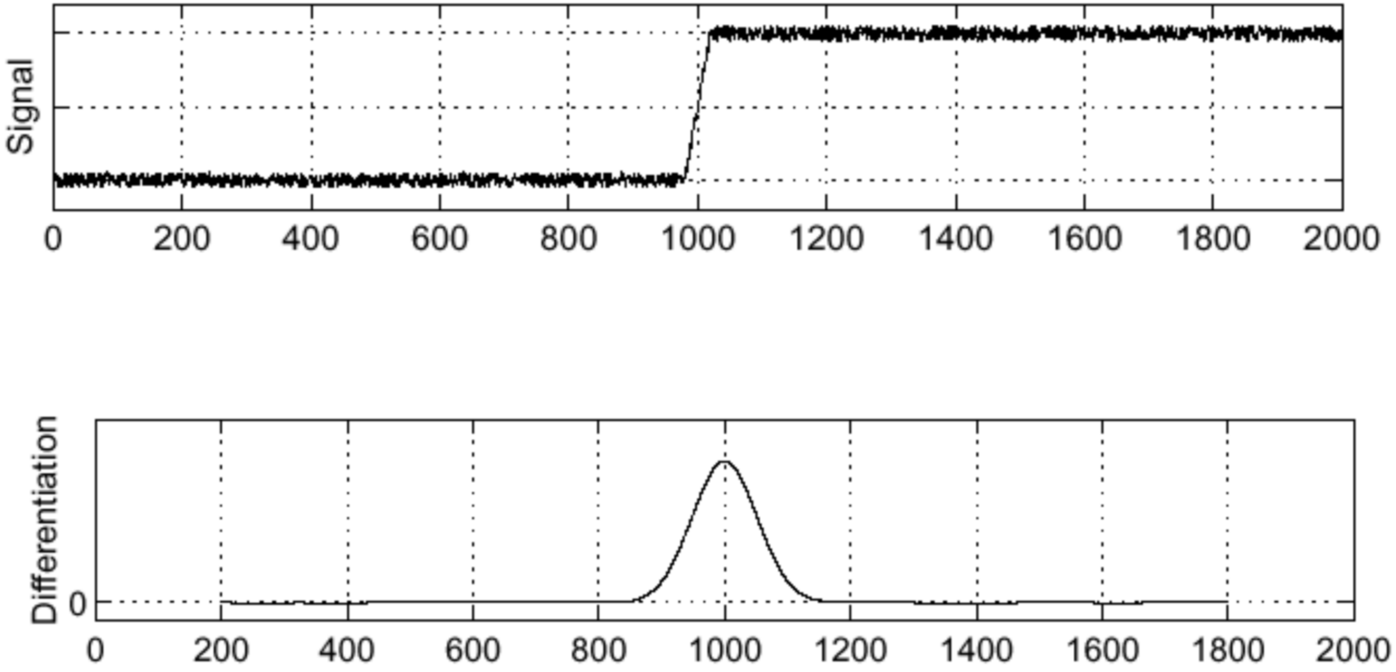
Recall: First Derivative Filters

- Central difference

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

-1	0	1

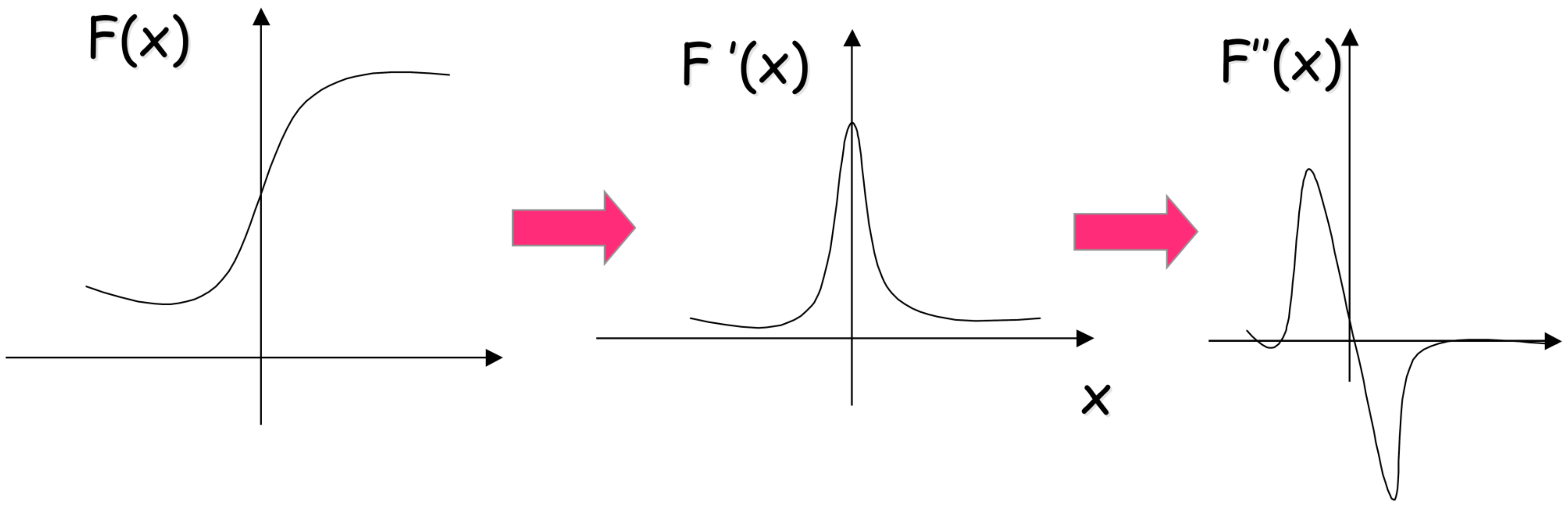
X derivative filter



Find edge

Second Derivative Filters

- Peaks or valleys of the first-derivative of the input signal, correspond to “zero-crossings” of the second-derivative of the input signal



Second Derivative Filters

- Taylor series expansion

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{3!}h^3 f'''(x) + O(h^4)$$

add

$$+ \left[f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{3!}h^3 f'''(x) + O(h^4) \right]$$

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + O(h^4)$$

$$\frac{f(x-h) - 2f(x) + f(x+h)}{h^2} = f''(x) + O(h^2)$$

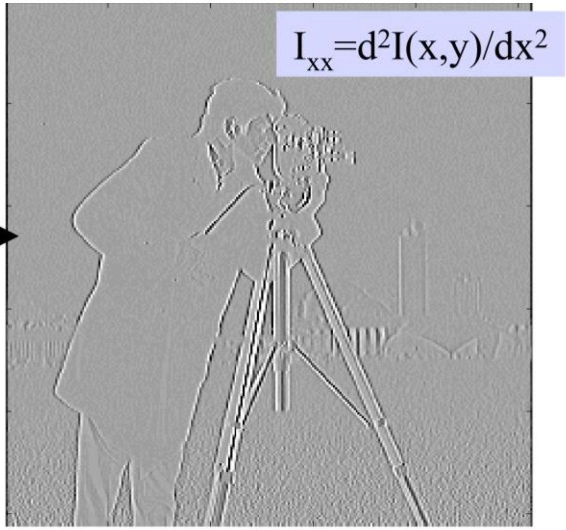
1	-2	1

Central difference approx to second derivative

Example: Second Derivatives

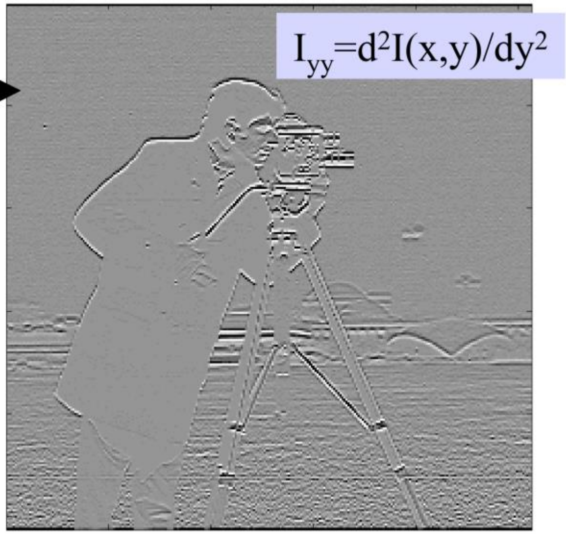
$[\ 1 \quad -2 \quad 1\ ]$

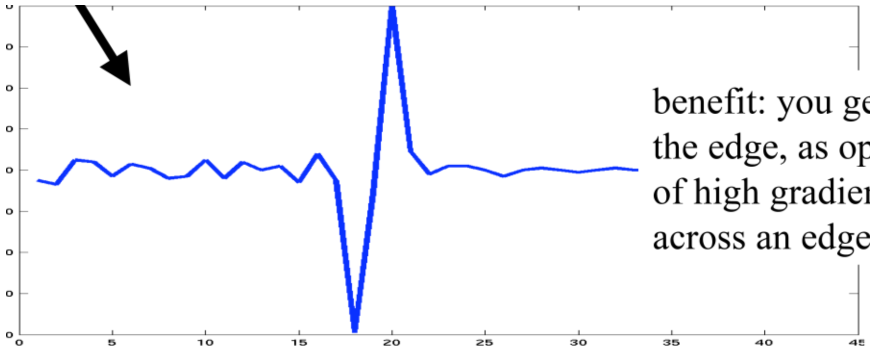
2nd Partial deriv wrt x



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

2nd Partial deriv wrt y





benefit: you get clear localization of the edge, as opposed to the "smear" of high gradient magnitude values across an edge

Edge Detection with Second Derivative Filters

- Find zero-crossings in second derivate
- In 1D, convolve with [1 -2 1] and look for pixels where response is (nearly) zero?
- In 1D, convolve with [1 -2 1] and look for pixels where response is nearly zero AND magnitude of first derivative is “large enough”.

Laplace Filter

first-order finite difference

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

Derivative filter

-1	0	1

second-order finite difference

$$f''(x) \approx \frac{\delta_h^2[f](x)}{h^2} = \frac{\frac{f(x+h)-f(x)}{h} - \frac{f(x)-f(x-h)}{h}}{h} = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

1D Laplacian filter

1	-2	1

Laplace Filter

- 2D

$$I_{xx} + I_{yy} = \left(\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \right) * I$$

$$= \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{Laplacian filter } \nabla^2 I(x,y)} * I$$

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

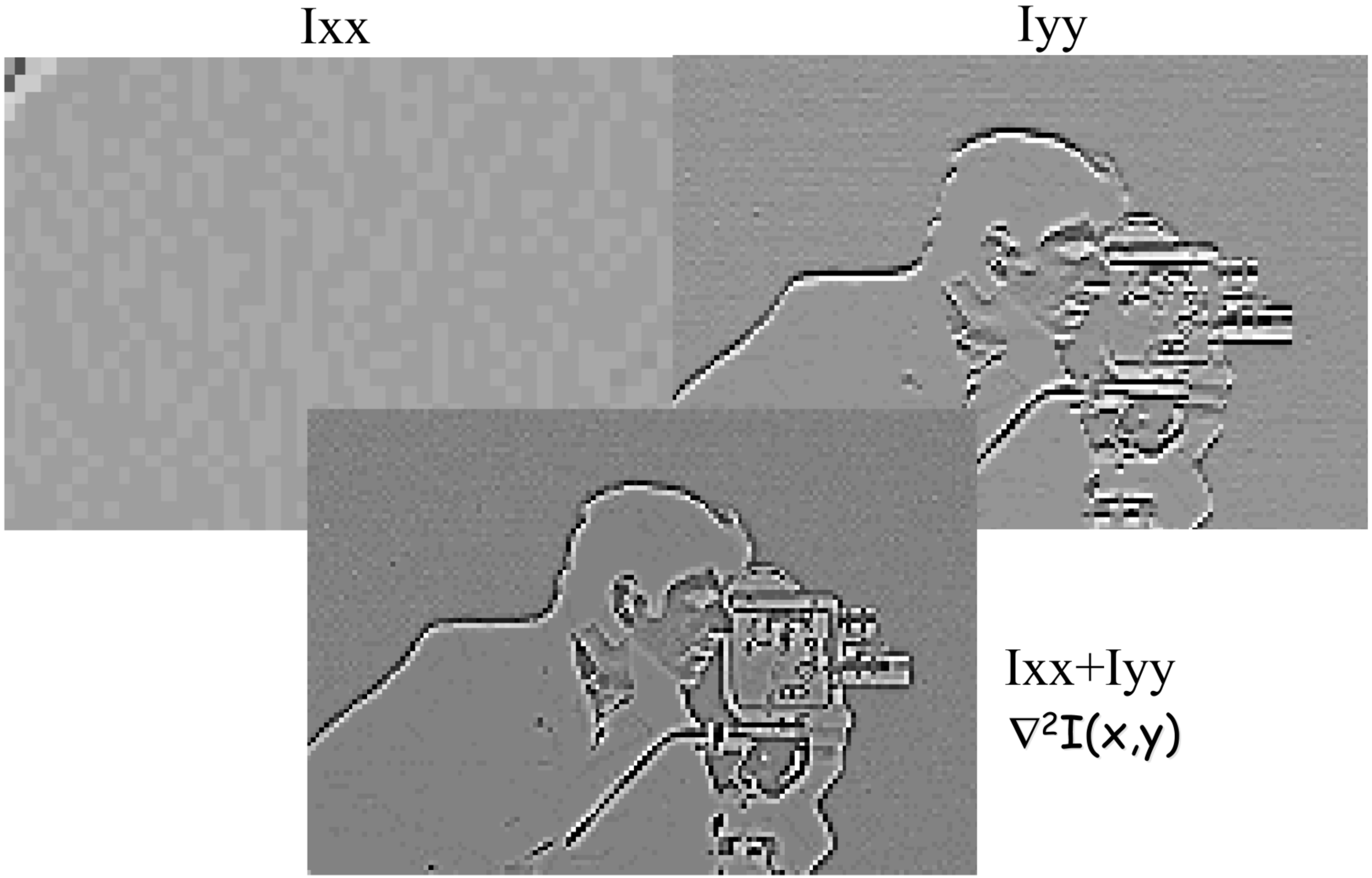
0	1	0
1	-4	1
0	1	0

2D Laplace filter

Example: Laplacian

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * I$$

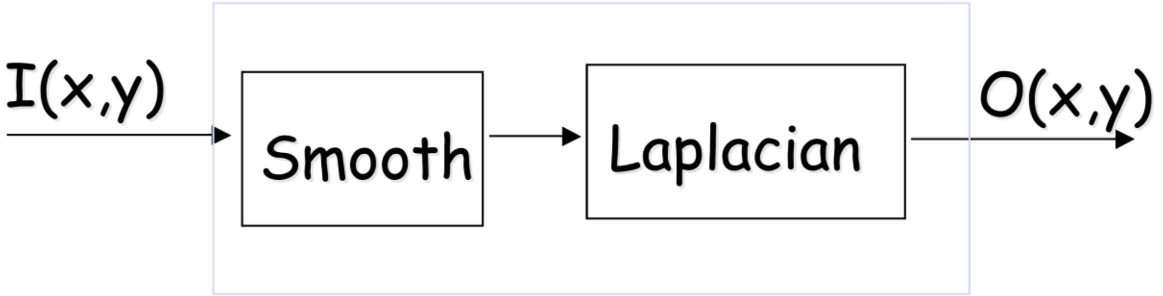
Example: Laplacian



More about Laplacian

- $\nabla^2 I(x,y)$ is a SCALAR
  - ↑ Can be found using a SINGLE mask
  - ↓ Orientation information is lost

- $\nabla^2 I(x,y)$ is the sum of SECOND-order derivatives
  - But taking derivatives increases noise
  - Very noise sensitive!



Laplacian of Gaussian (LoG) Filter

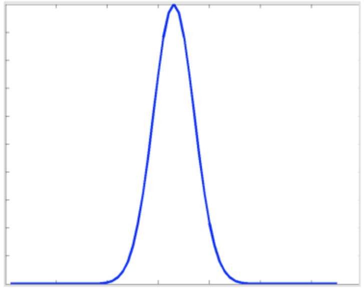
- First smooth with a Gaussian filter
- Then apply the Laplacian filter

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

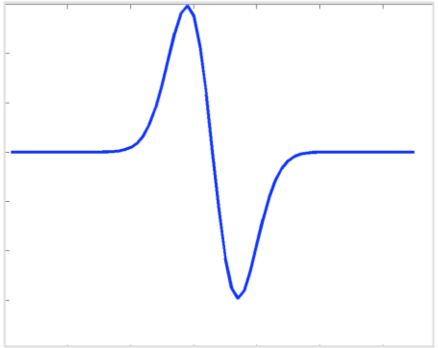
$$\underbrace{\nabla^2(f(x,y) \otimes G(x,y))}_{\text{Laplacian of Gaussian-filtered image}} = \underbrace{\nabla^2 G(x,y) \otimes f(x,y)}_{\text{Laplacian of Gaussian (LoG)-filtered image}}$$

1D Gaussian and Derivatives

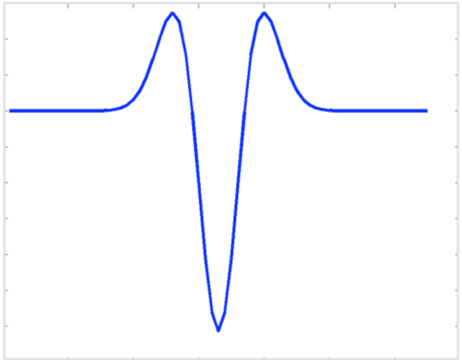
$$g(x) = e^{-\frac{x^2}{2\sigma^2}}$$



$$g'(x) = -\frac{1}{2\sigma^2} 2xe^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

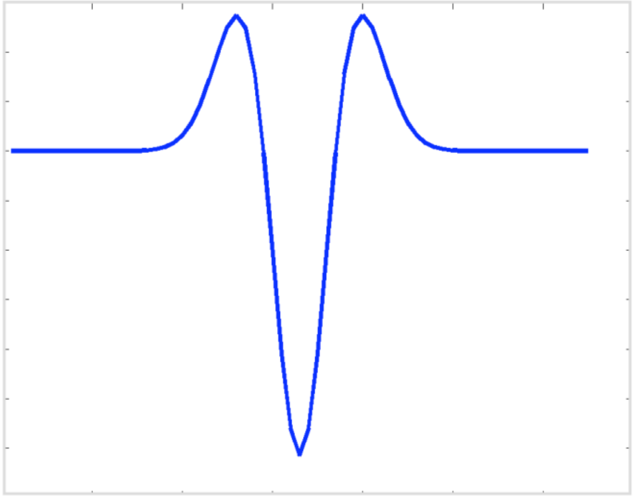


$$g''(x) = (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}$$



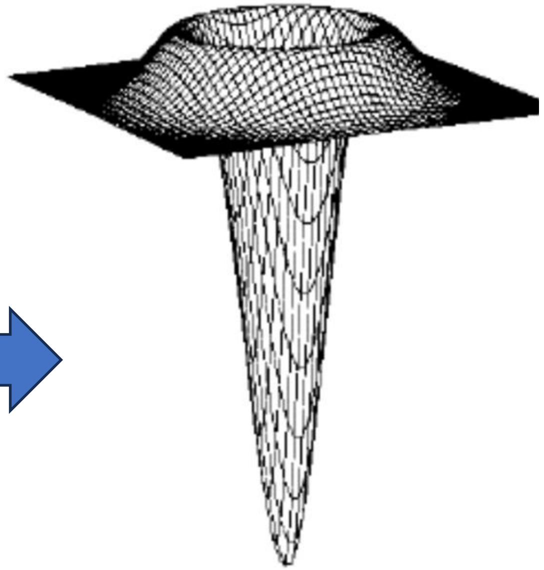
Second Derivate of Gaussian

$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right)e^{-\frac{x^2}{2\sigma^2}}$$



2D

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



Laplacian of Gaussian

Mexican Hat Function

Laplacian of Gaussian Filter

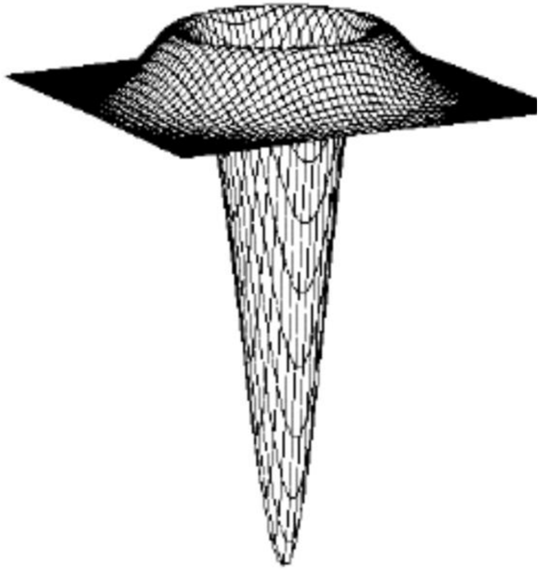
$$\nabla^2 \mathbf{I} = \frac{\partial^2 \mathbf{I}}{\partial x^2} + \frac{\partial^2 \mathbf{I}}{\partial y^2}$$

$$\nabla^2 \mathbf{I} \circ g = \nabla^2 g \circ \mathbf{I}$$

$$\nabla^2 g = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} g(x, y)$$

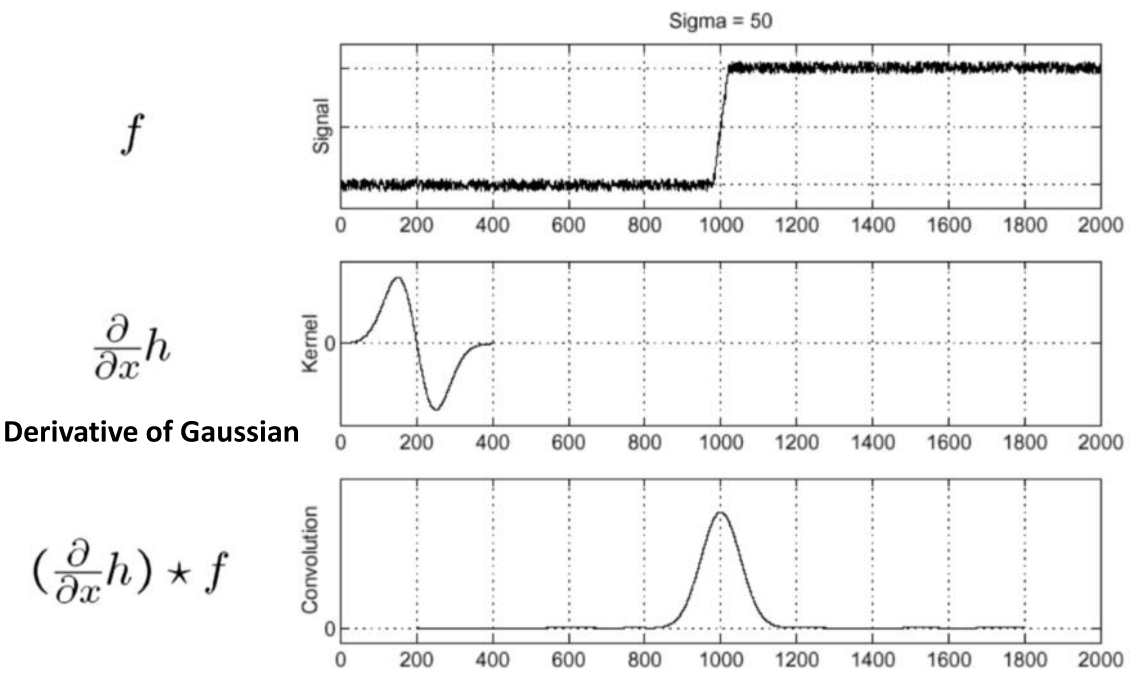
Smoothing and second derivative

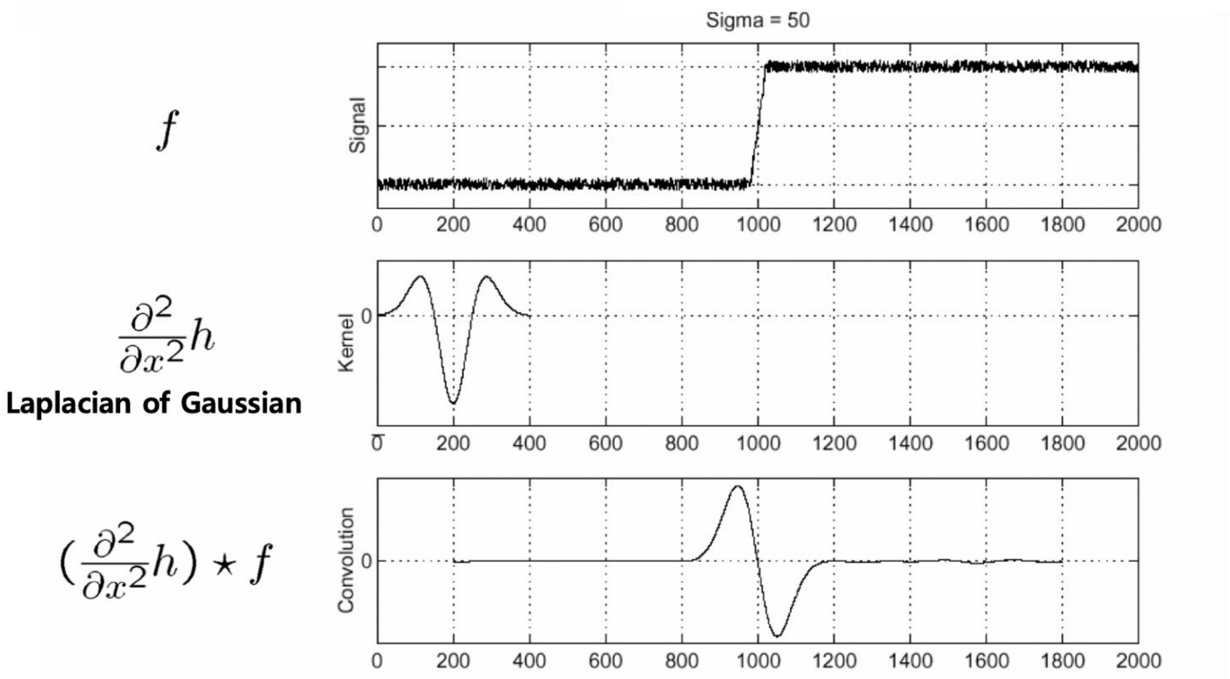
$\nabla^2 h_\sigma(u, v)$

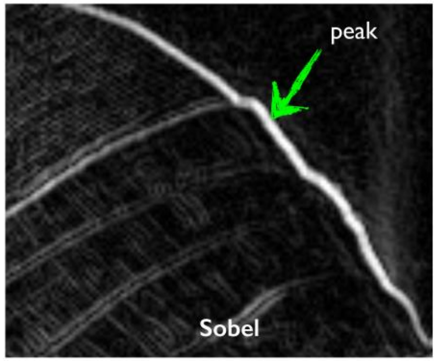


Laplacian of Gaussian

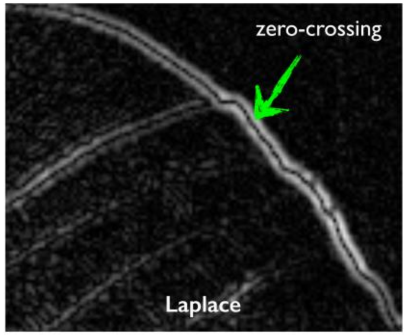
Laplacian of Gaussian Filter



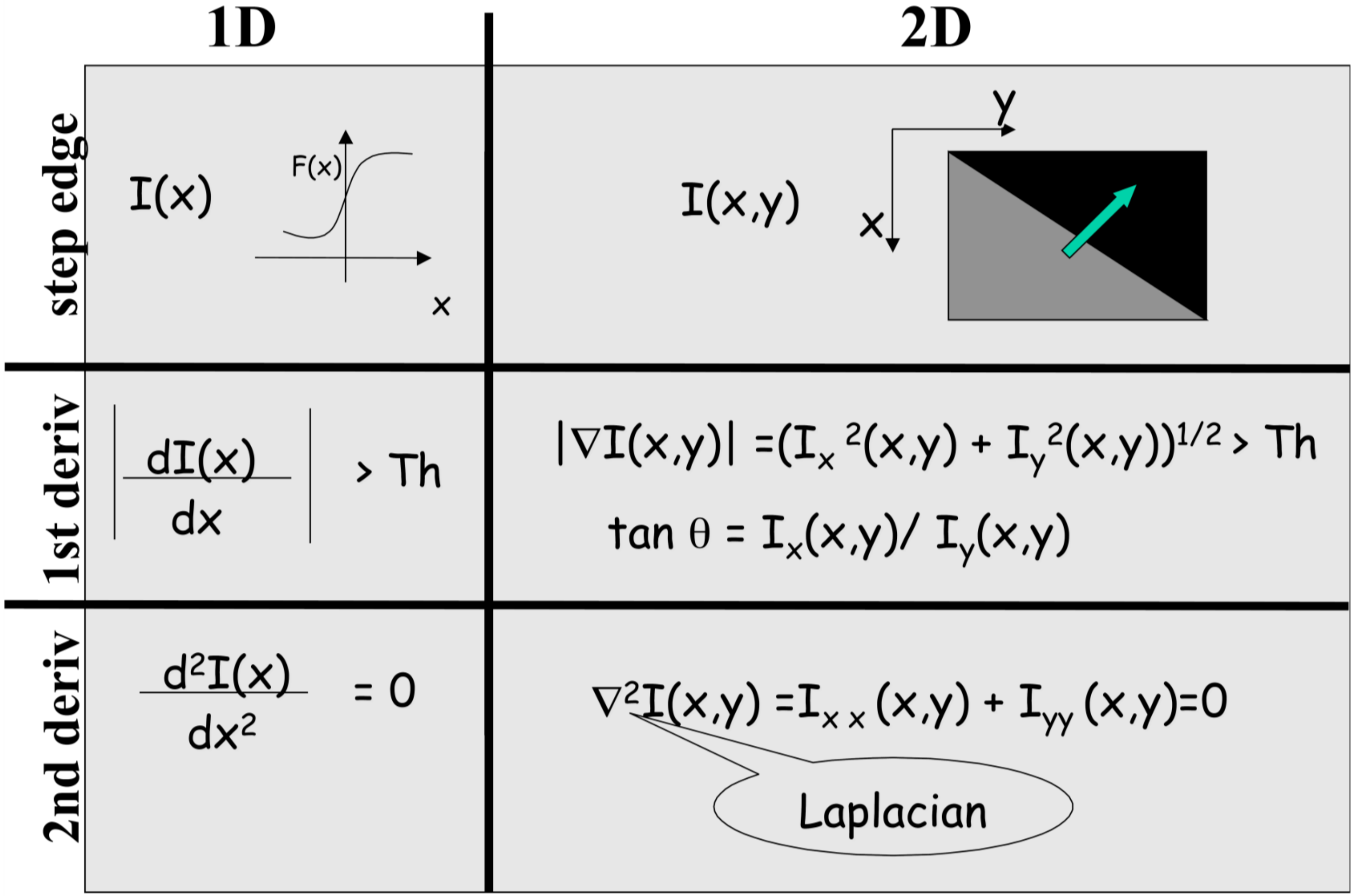




Zero crossings



Edge Detection with LoG



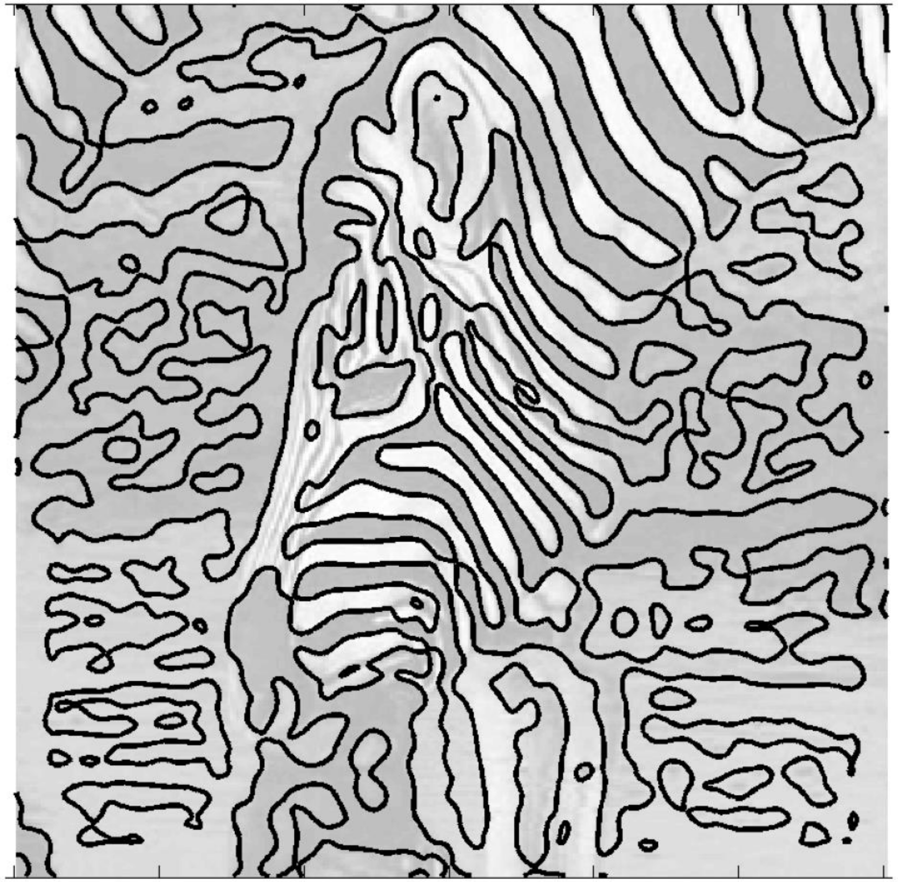
Zero-Crossing as an Edge Detector

Raw zero-crossings (no contrast thresholding)

LoG sigma = 2, zero-crossing

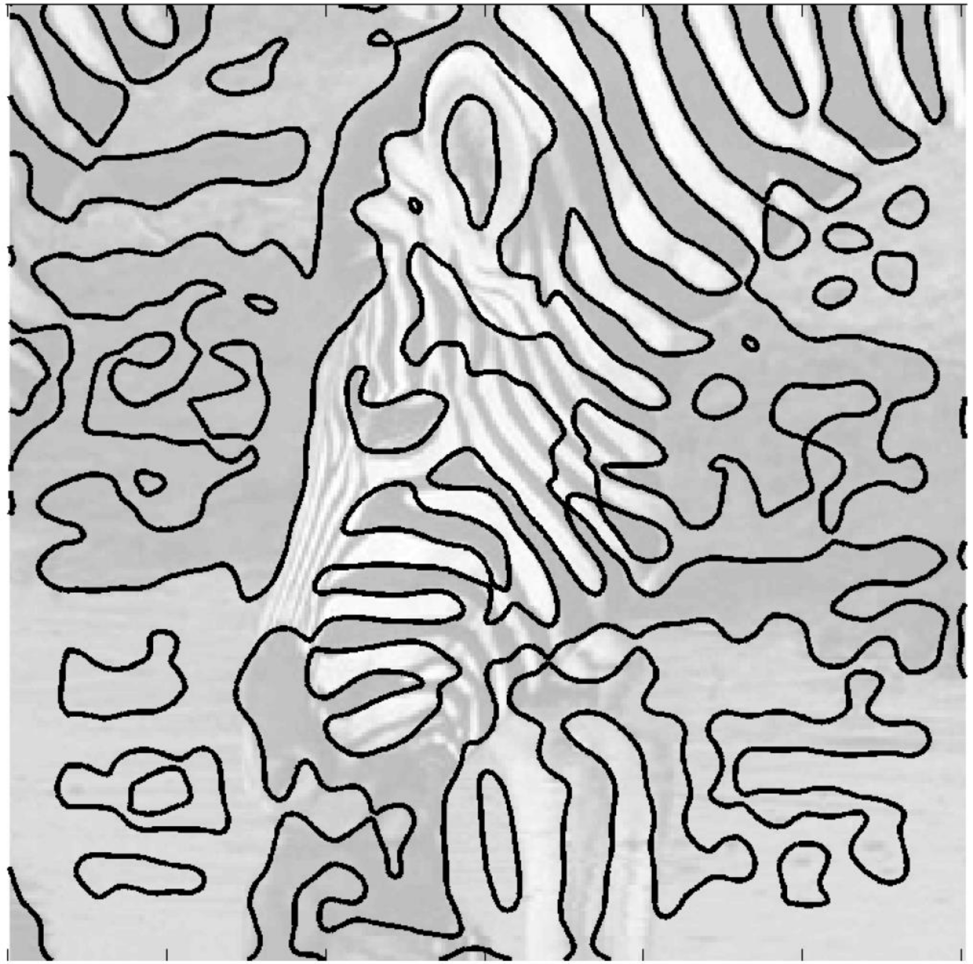
Zero-Crossing as an Edge Detector

Raw zero-crossings (no contrast thresholding)



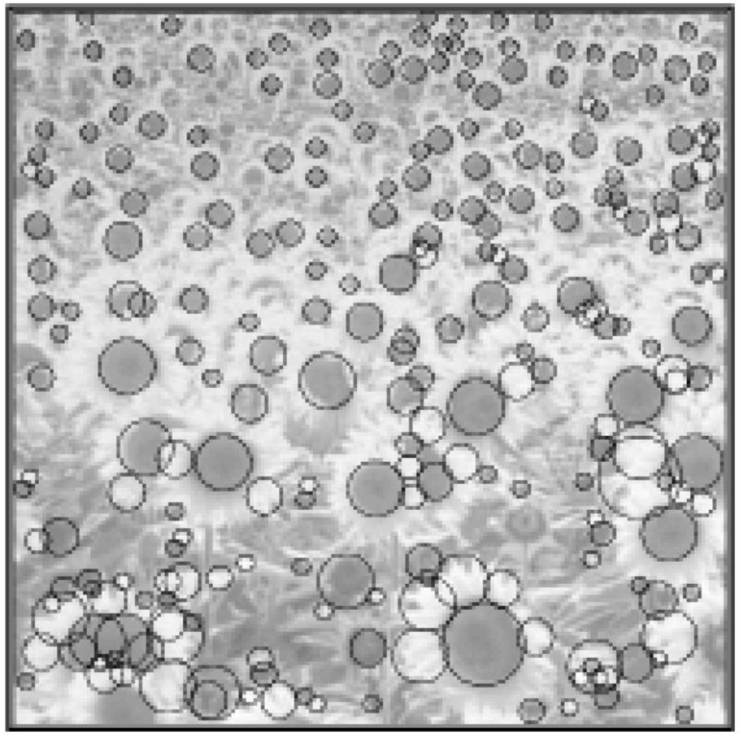
LoG sigma = 4, zero-crossing

Zero-Crossing as an Edge Detector



LoG sigma = 8, zero-crossing

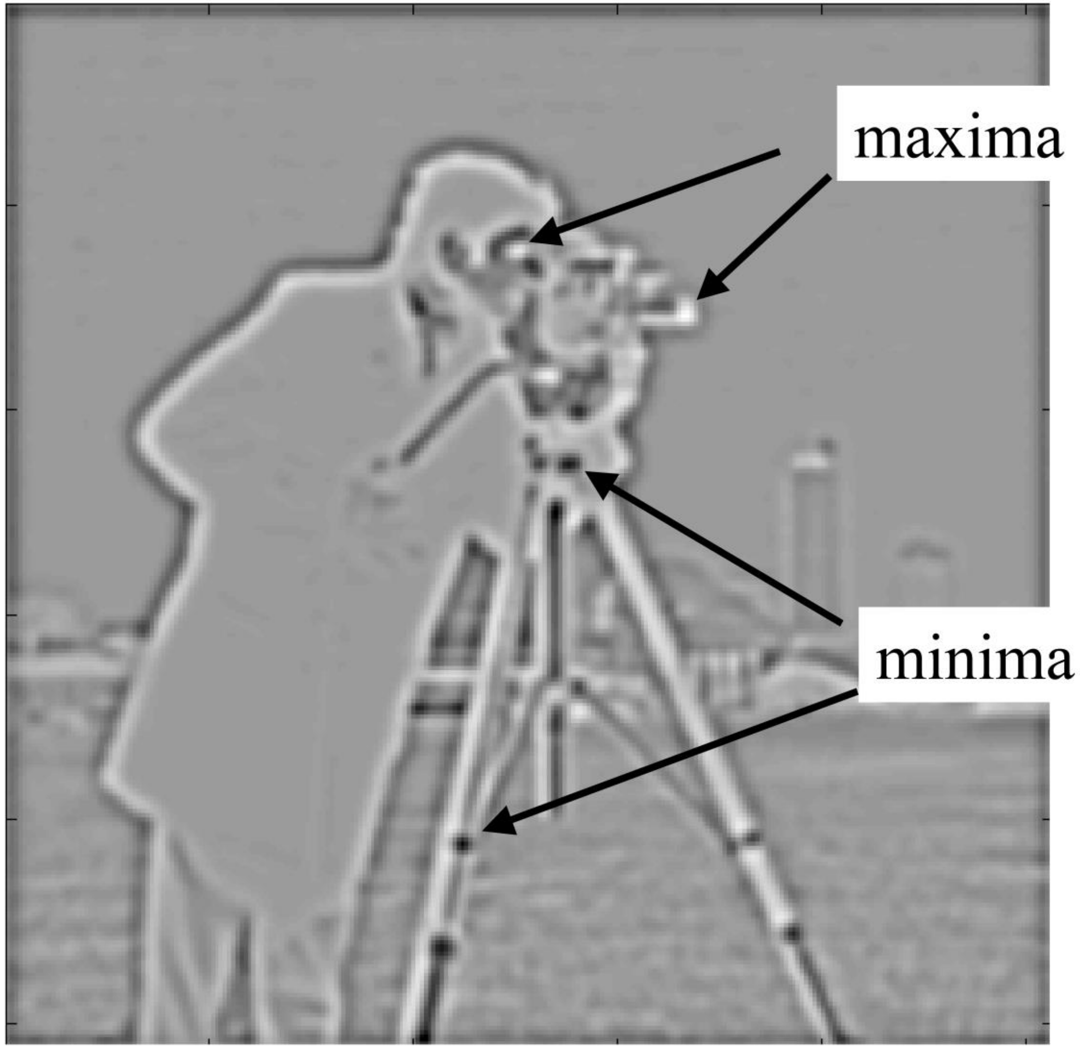
Blob Detection with LoG



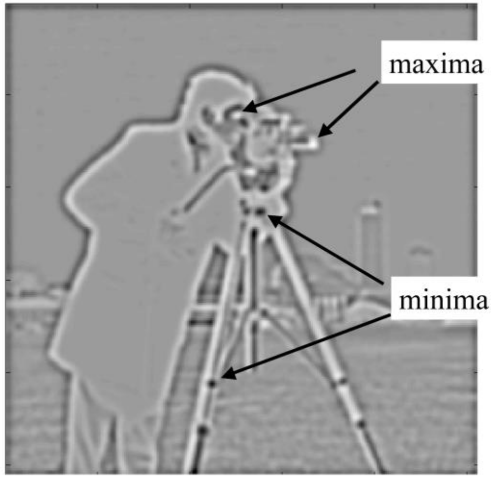
Lindeberg: ``Feature detection with automatic scale selection''. International Journal of Computer Vision, vol 30, number 2, pp. 77--116, 1998.

Example: LoG Extrema

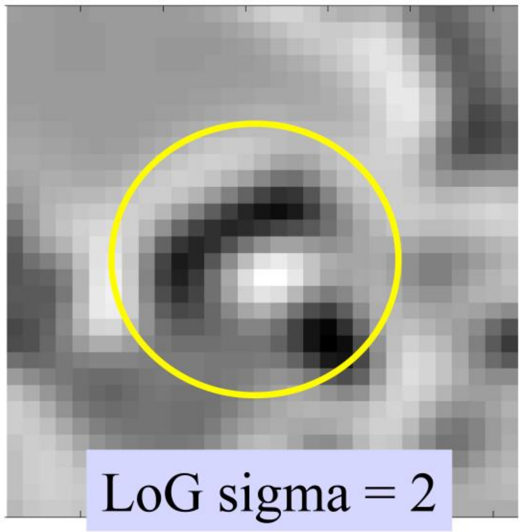
LoG
sigma = 2

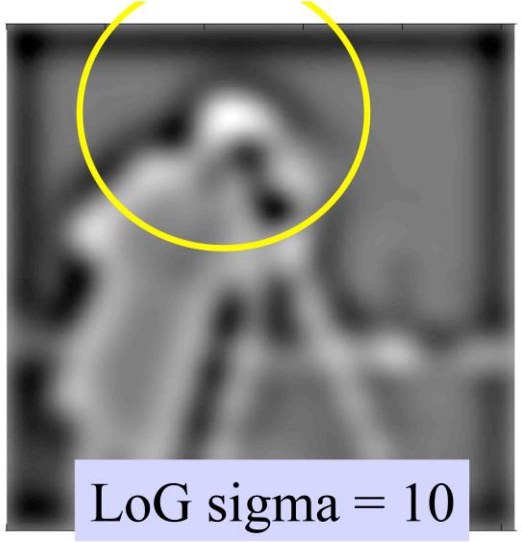


LoG Blob Detection



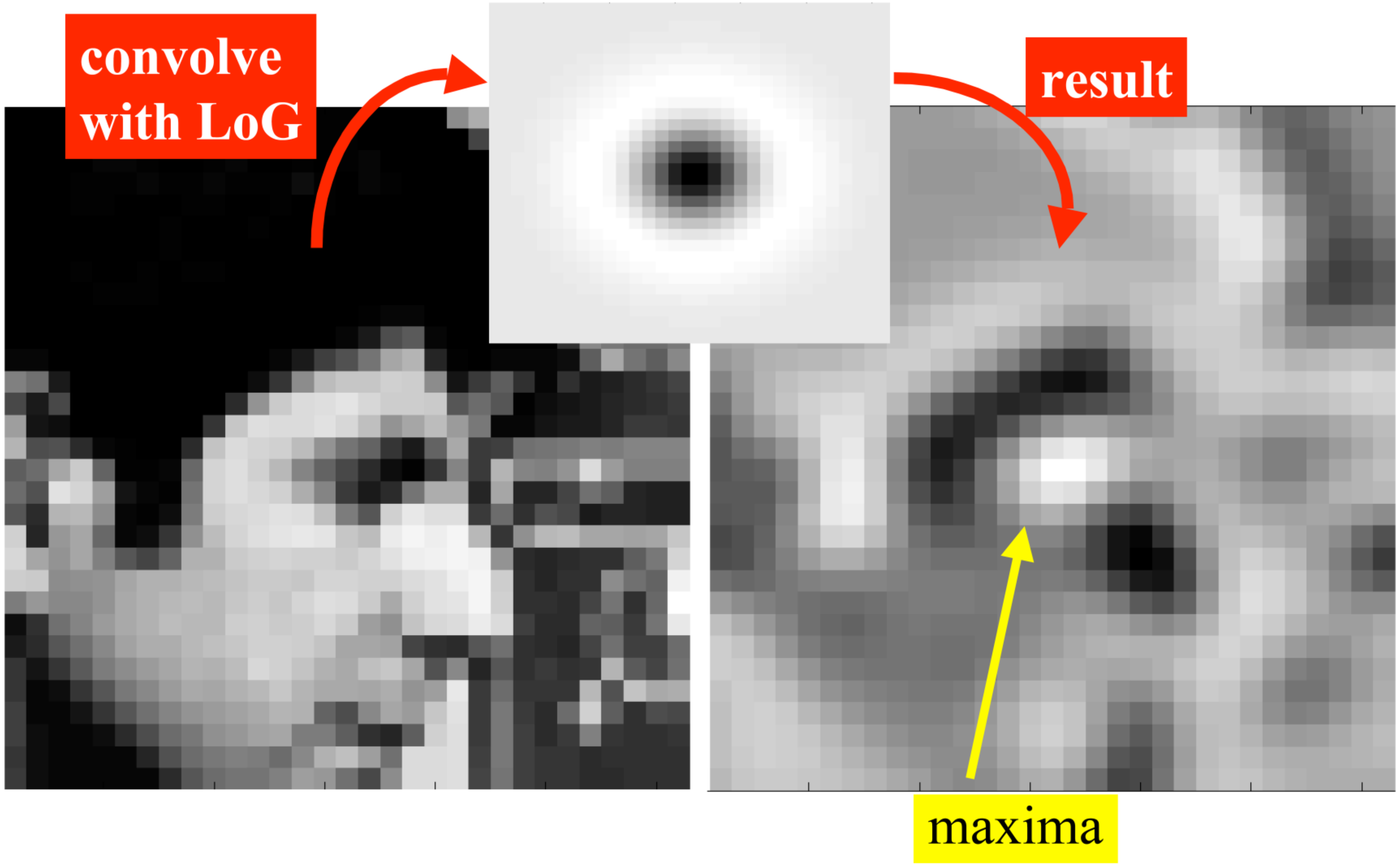
- LoG filter extrema locates “blobs”
  - Maxima: dark blobs on light background
  - Minima: light blobs on dark background
- Scale of blob (size ; radius in pixels) is determined by the sigma parameter of the LoG filter



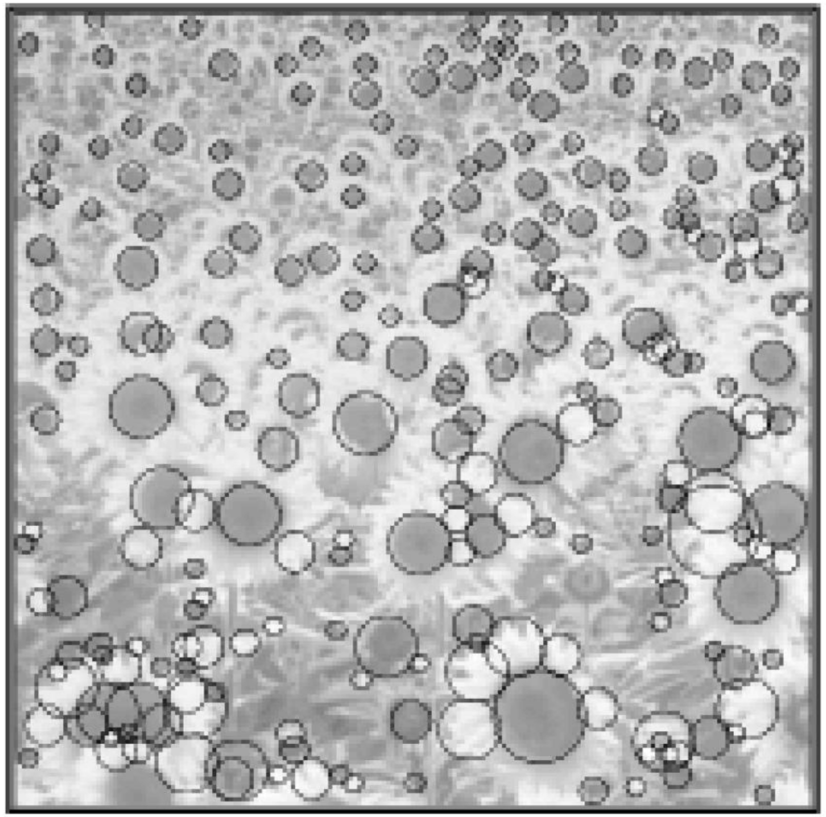


LoG Blob Detection

Convolution (and cross correlation) with a filter can be viewed as comparing a little “picture” of what you want to find against all local regions in the mage.



LoG Blob Detection



Lindeberg: blobs are detected as local extrema in space and scale, within the LoG scale-space volume.

Further Reading

- Tony Lindeberg, Feature Detection with Automatic Scale Selection, https://people.kth.se/~tony/papers/cvap198.pdf