Lecture 6-1 Spatial filtering 2 (chapter 3.6)

Yuyao Zhang PhD

zhangyy8@shanghaitech.edu.cn

SIST Building 2 302-F

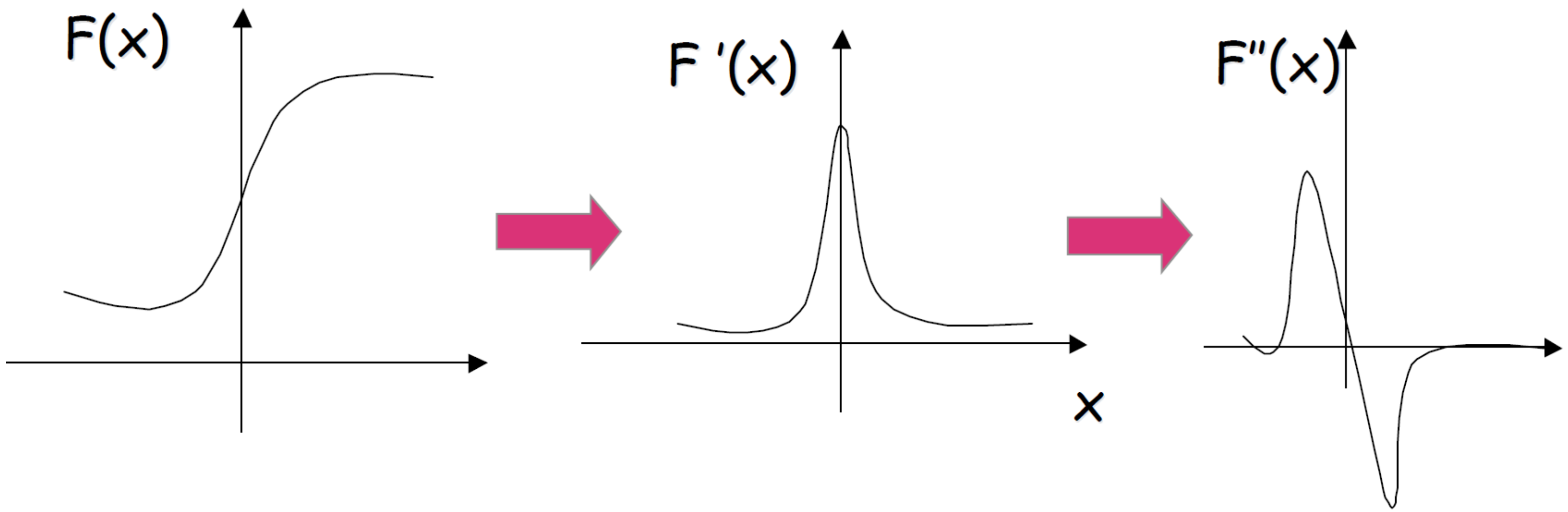
Outline

- **Sobel Filter**
- **Unsharpen Filter (非锐化掩蔽)**
- **LoG Filter**
  - useful for finding edges
  - also useful for finding blobs

Recall: First & Second-Derivative filters

- ➢Sharp changes in gray level of the input image corresponds to "peaks or valleys" of the first-derivative of the input signal.

- ➢Peaks or valleys of the first derivative of the input signal, correspond to "zero-crossings" of the second-derivative of the input signal.



Laplacian(拉普拉斯算子)

For an image function $f(x, y)$,

X direction： $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$

Y direction： $\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$

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

$$= f(x, y+1) + f(x, y-1) + f(x+1, y) + f(x-1, y) - 4f(x, y)$$

Laplacian Filter Masks

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y) - 4f(x,y)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Laplacian(拉普拉斯算子)

For an image function $f(x, y)$,

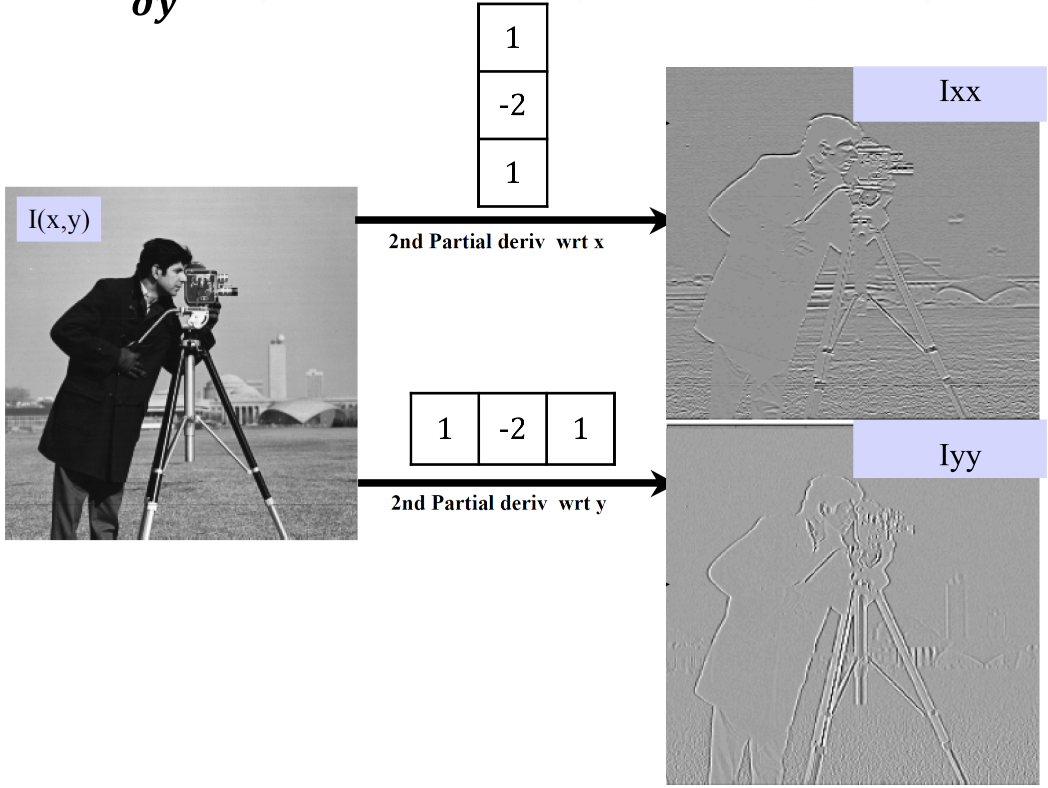
X direction： $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$

Y direction： $\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$

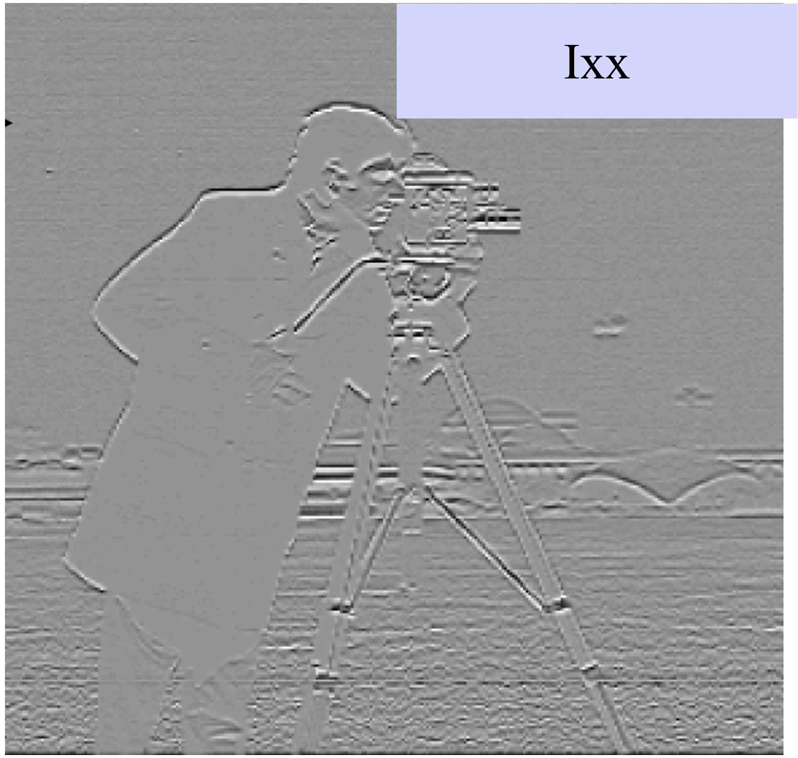
I_{yy}

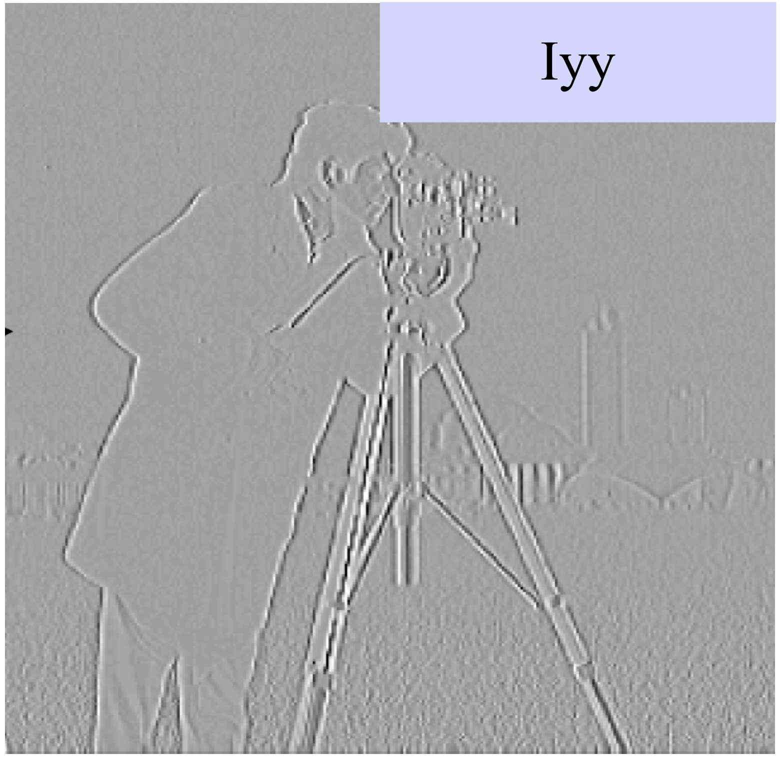
1	-2	1

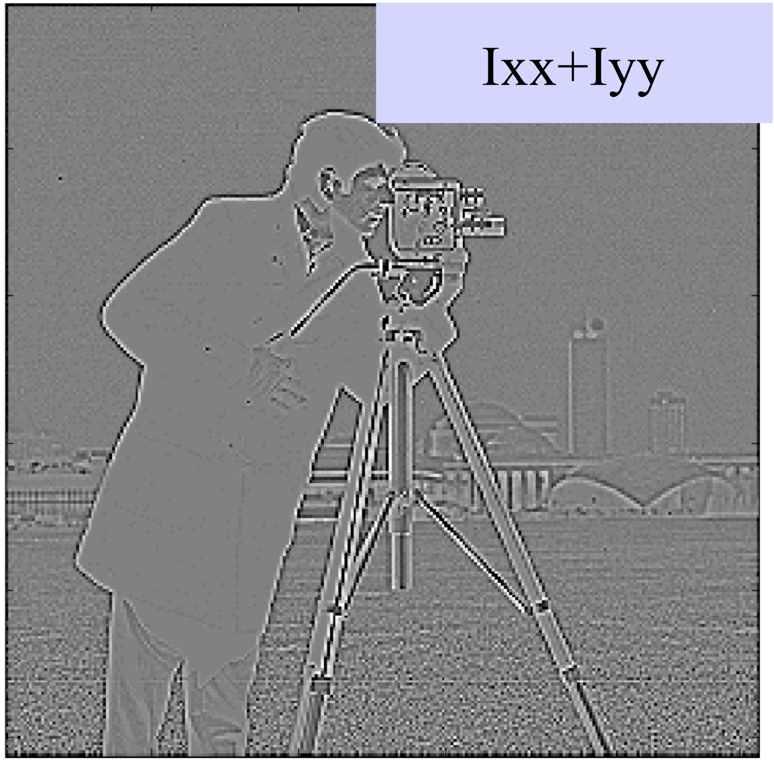
1	
-2	I_{xx}
1	



Laplacian(拉普拉斯算子)







Gradient(梯度)

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

The first-order derivative of $f(x,y)$: $\nabla \boldsymbol{f} \equiv \mathbf{grad}(\boldsymbol{f}) \equiv \begin{cases} \boldsymbol{g}_x \\ \boldsymbol{g}_y \end{cases} = \begin{cases} \dfrac{\partial f}{\partial x} = \\ \dfrac{\partial f}{\partial y} \end{cases}$

-1	1

*
1

The amplitude： $\boldsymbol{M}(\boldsymbol{x},\boldsymbol{y}) = \mathbf{mag}(\nabla \boldsymbol{f}) = \sqrt{\boldsymbol{g}_x^2 + \boldsymbol{g}_y^2}$

$$\boldsymbol{M}(\boldsymbol{x},\boldsymbol{y}) \approx |\boldsymbol{g}_x| + |\boldsymbol{g}_y|$$

Gradient(梯度)

➢ Roberts cross-gradient operator（罗伯特交叉梯度算子）

$$M(x,y) \approx |g_x| + |g_y|$$

$$= |z_9 - z_5| + |z_8 - z_6|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0
0	1

0	-1
1	0

Gradient(梯度)

➢ **Sobel operator （Sobel算子）**

$f(x+1) - f(x-1)$

$$\boldsymbol{M(x,y) = |(z_7+2z_8+z_9)-(z_1+2z_2+z_3)| + |(z_3+2z_6+z_9)-(z_1+2z_4+z_7)|}$$

$f_x = f(x+1) - f(x)$

$f_x = f(x) - f(x+1)$

$2f_x =$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

f_x

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operator

More Sobel operators

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

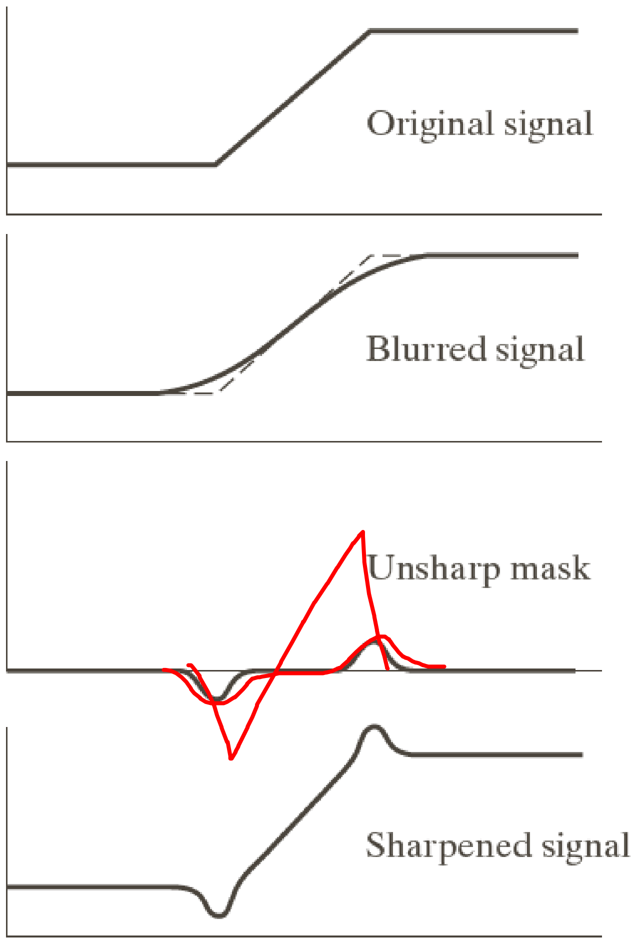
The Notes about the Laplacian

- $\nabla^2 I(x, y)$ is a SCALAR
  - $\uparrow$ Can be found using a SINGLE mask
  - $\downarrow$ Orientation information is lost
- $\nabla^2 I(x, y)$ is the sum of SECOND-order derivatives
  - But taking derivatives increases noise.
  - Very noise sensitive!
- It is always combined with a smoothing operation.

Unsharpen Mask(非锐化掩蔽)

$$g_{\text{mask}}(x, y) = f(x, y) - \overline{f(x, y)}$$

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$



Laplacian of Gaussian (LoG) Filter

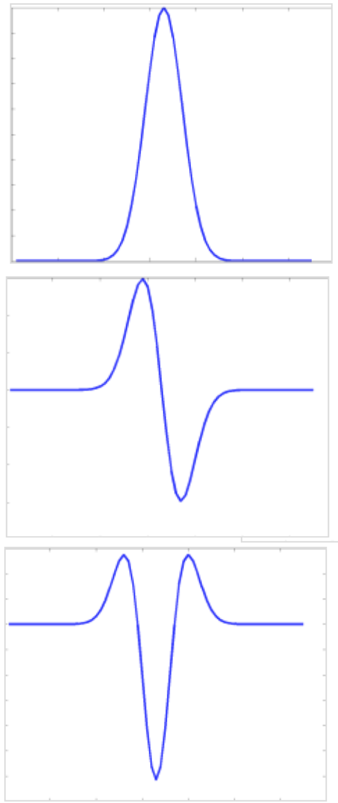
- First smooth (Gaussian filter),
- Then, find zero-crossings (Laplacian filter):

$$\nabla^2 \left(G(x,y)\right)$$

$$G(x,y) = e^{-\frac{x^2}{2\sigma^2}}$$

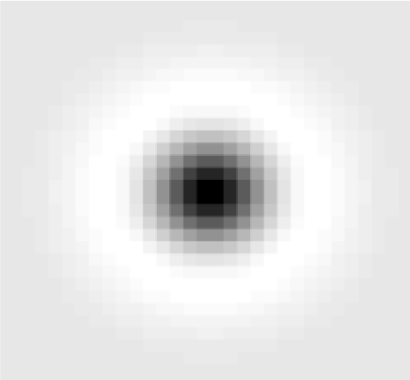
$$G'(x,y) = -\frac{1}{2\sigma^2} 2x e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

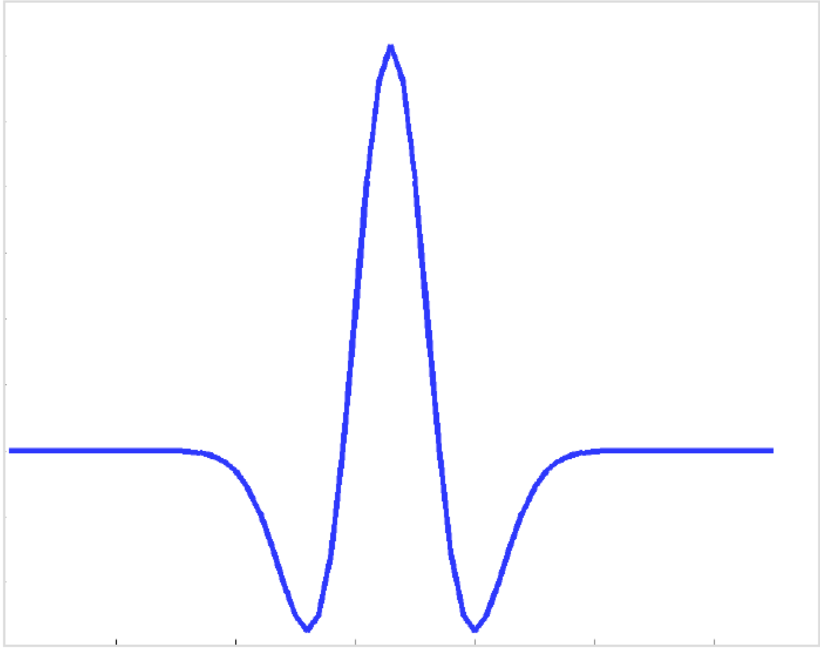
$$G''(x,y) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{-\frac{x^2}{2\sigma^2}}$$



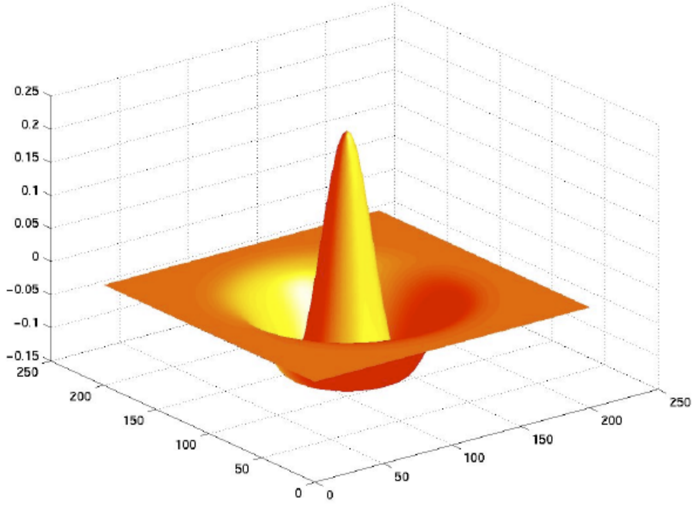
Second derivative of a Gaussian

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





2D analog



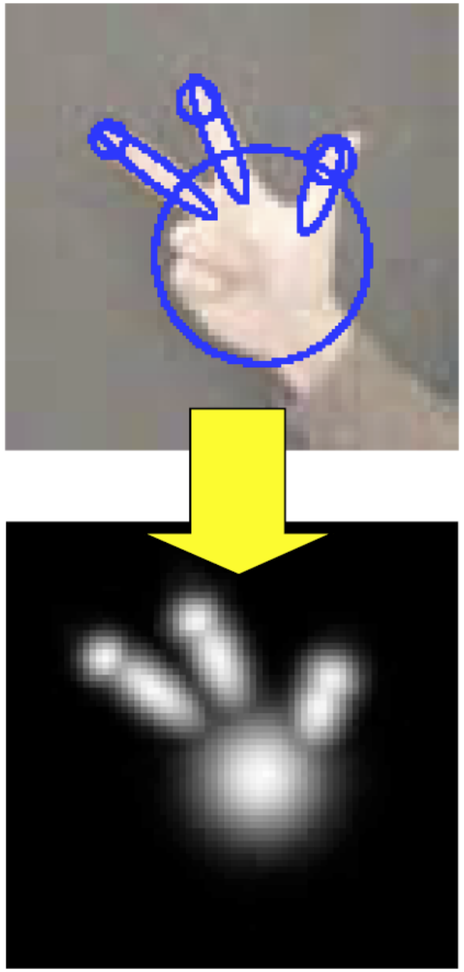
LoG "Mexican Hat"

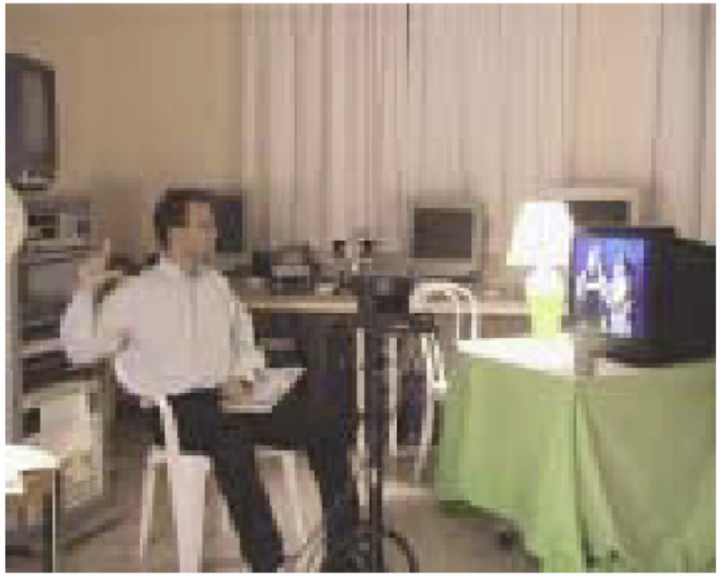
Effect of LoG Filter

Sigma = 1 Sigma = 4 Sigma = 10

Band-Pass Filter (suppresses both high and low frequencies)

Application of LoG Filter



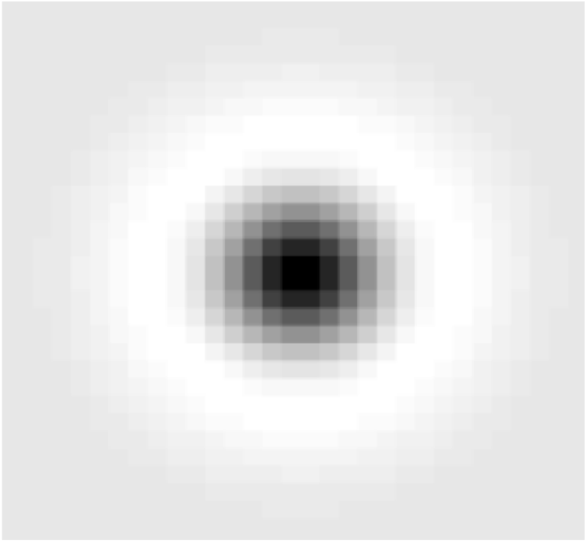


Gesture recognition for the ultimate couch potato

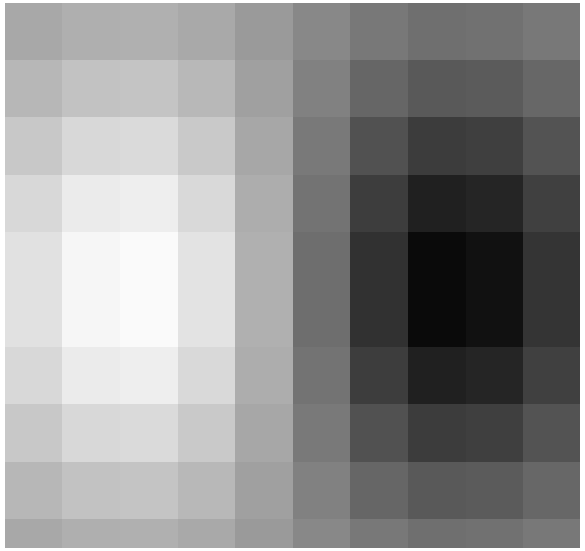
Take home message

- Key idea: 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 image.

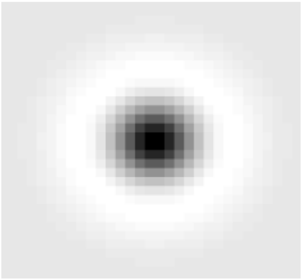
LoG



Derivative of Gaussian



*



=