Gaussian filter

$$f(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

Discretization on a 3x3 mask of a Gaussian with mean=0 and dev.standard =0.5

Derivative properties

f(x)	f'(x)	f''(x)
Constant trend	***= 0***	***= 0***
Discontinuity (begin-end)	***≠ 0***	***≠ 0***
Ramp	***≠ 0***	***= 0***

Objective:
Define 2 discrete operators f ' and f '' that verify these properties

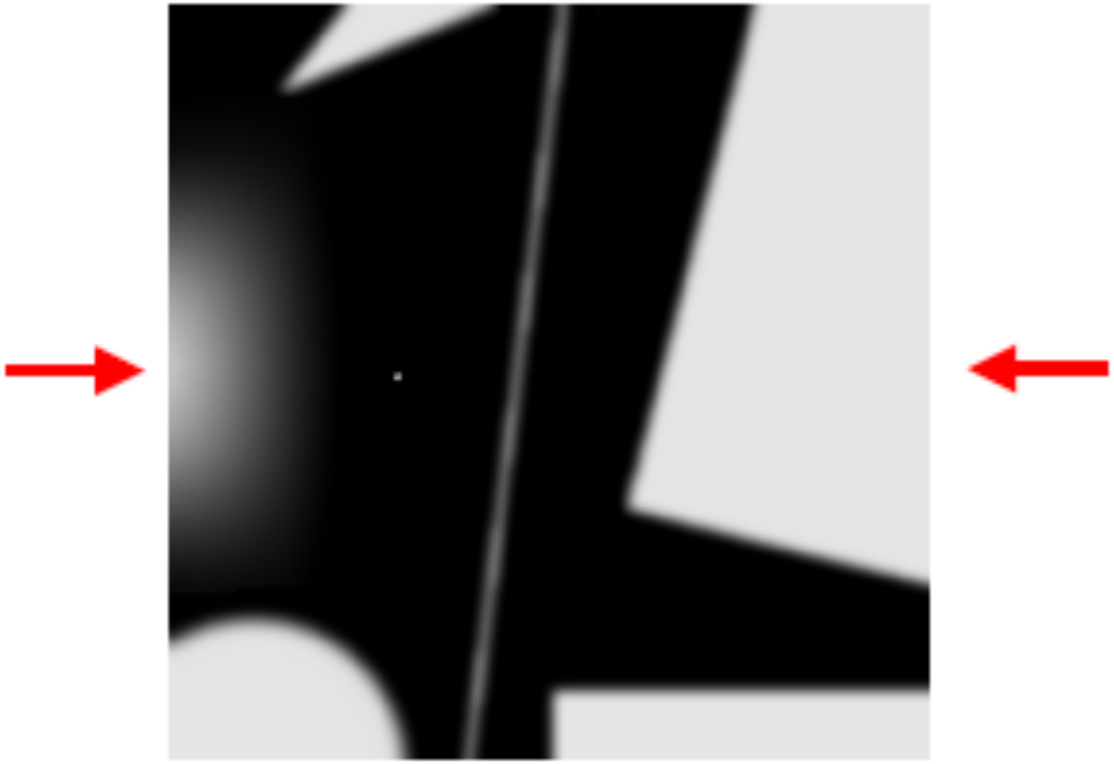
$f'(x) = f(x+1) - f(x)$

$f''(x) = \ldots$

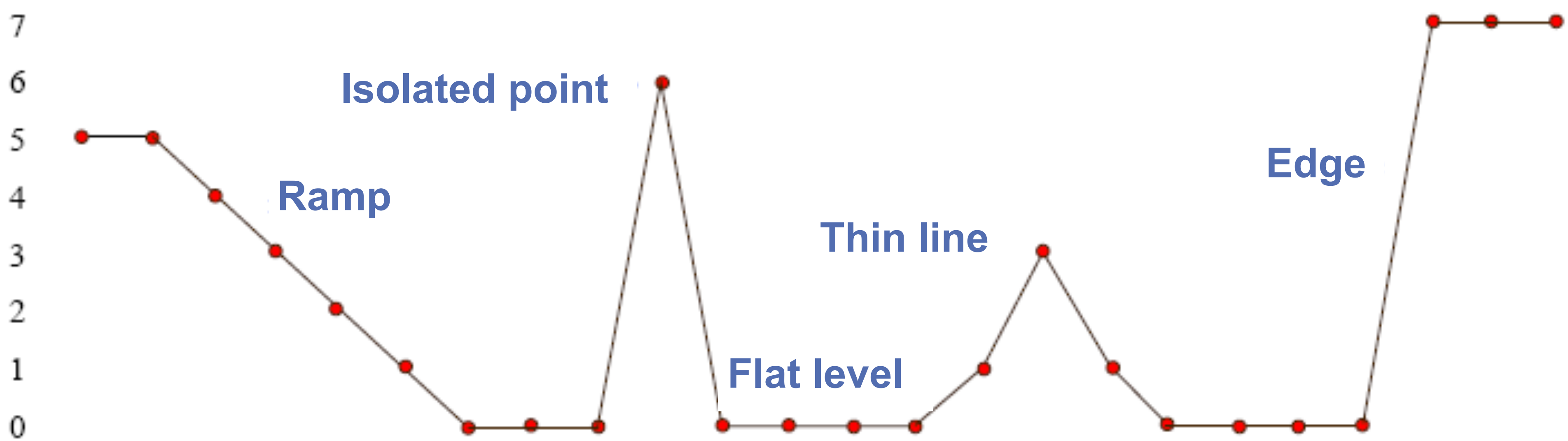
$= f(x+1) + f(x-1) - 2f(x)$

Example

- Let apply the derivative operators to the 1D line indicated by the arrows



Example (cont.)

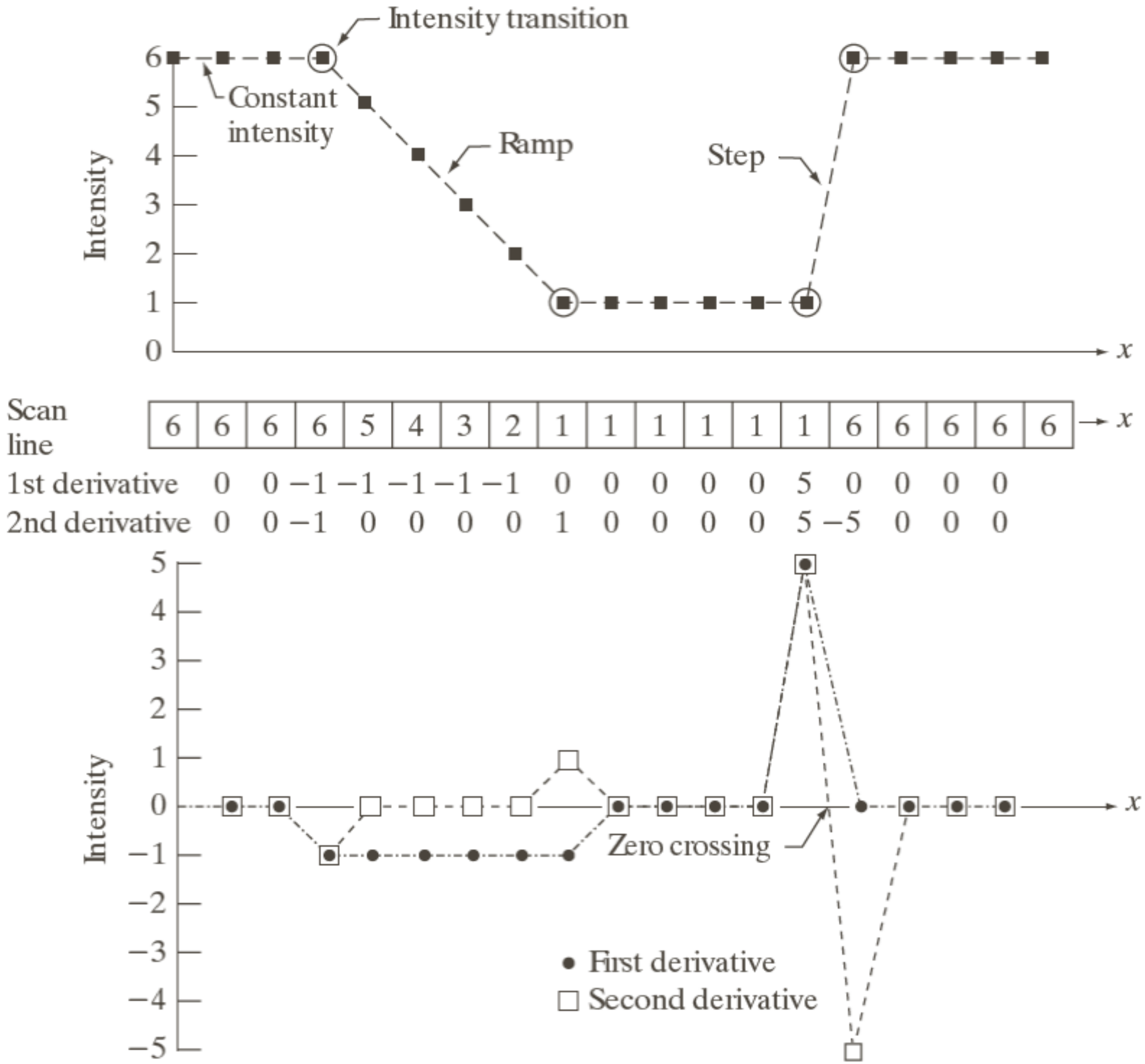


f	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	--
f'	--	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	--	--
f''	--	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	--	--

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

$$f''(x) = f(x+1) + f(x-1) - 2f(x)$$

Another example



Considerations

- Properties verified!

- Moreover:
  - f ' :
    - Ramp → Thick contours
    - Step → Strong response in correspondence to steep edges
  - f '' :
    - Point → Stronger response to thin details
    - Edges → Double response to edges with the zero-crossing

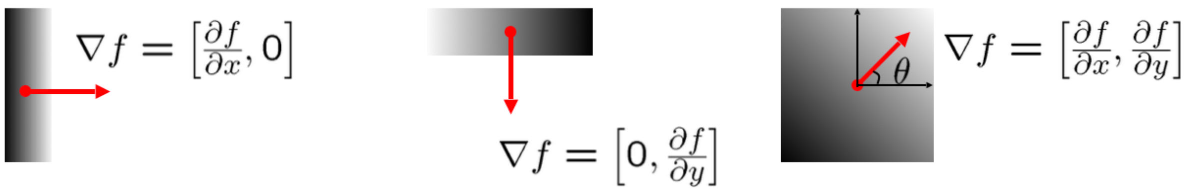
- Conclusion:
  - f '' ideal for sharpening
  - f ' e f '' ideal for edge extraction

Gradient

- The image gradient is defined as:

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

- The gradient is a vector oriented toward the direction of maximum intensity variation



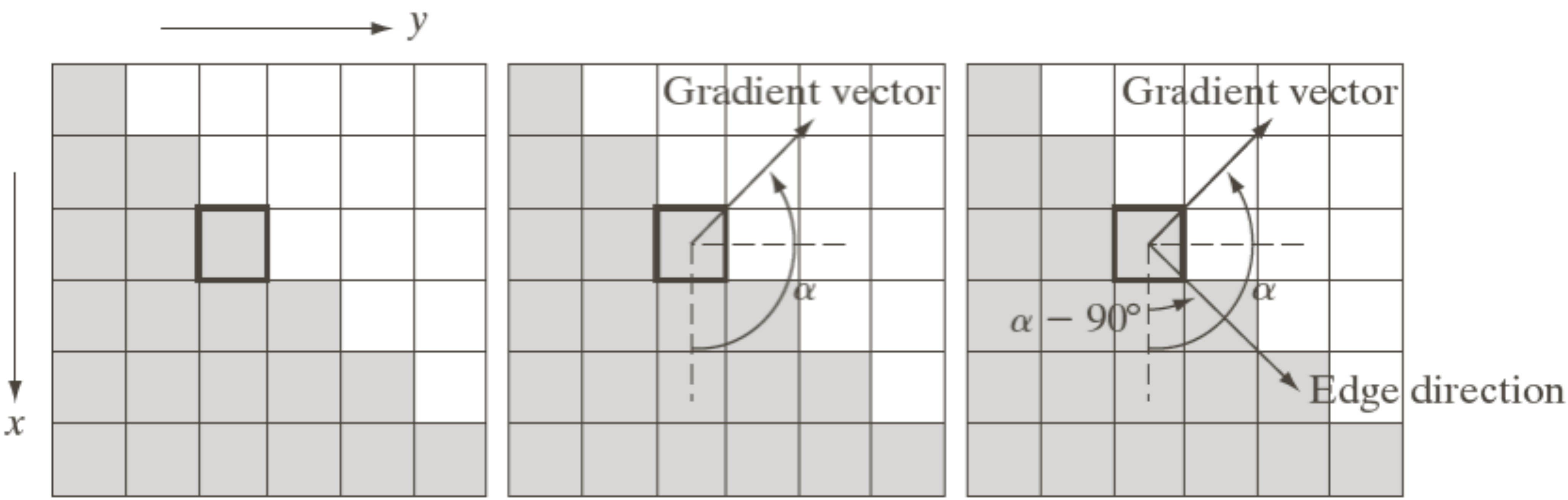
- The direction of the gradient is given by:

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

  - What can we conclude about the edge direction?
- The edge intensity is given by the gradient magnitude

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

Gradient



Gradient approximation

Roberts

1	0
0	-1

0	1
-1	0

Prewitt

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

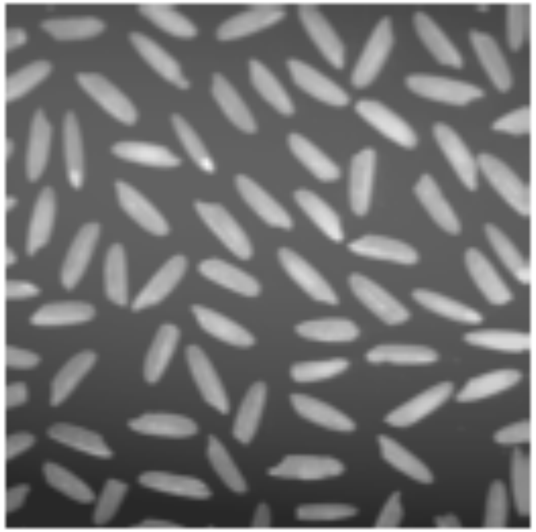
-1	0	1
-1	0	1
-1	0	1

Sobel

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

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

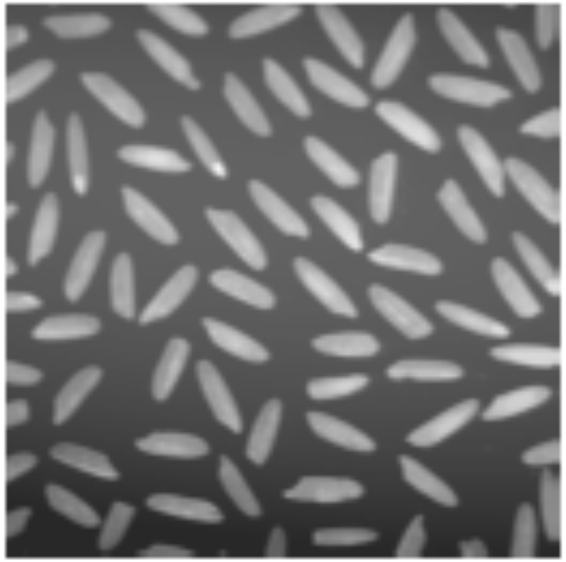
Roberts operator

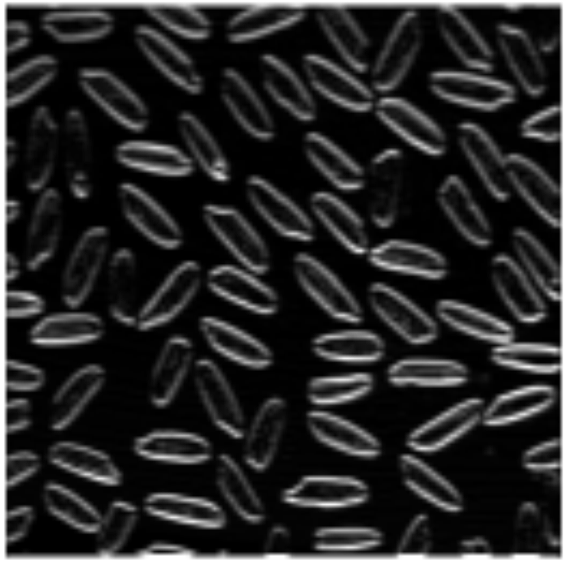


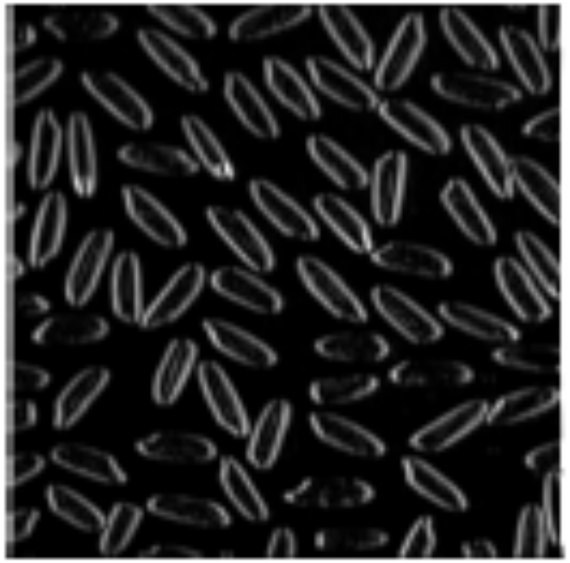
1	0
0	-1

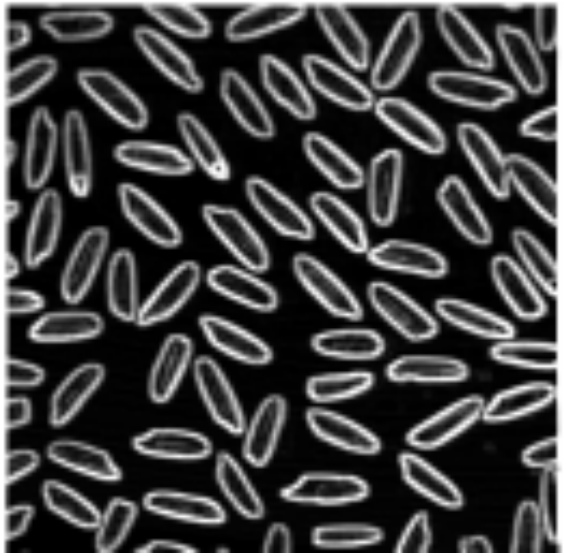
0	1
-1	0

Prewitt operator





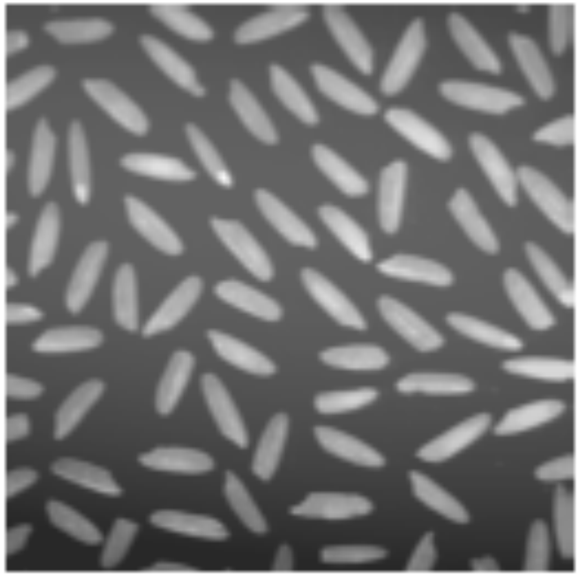


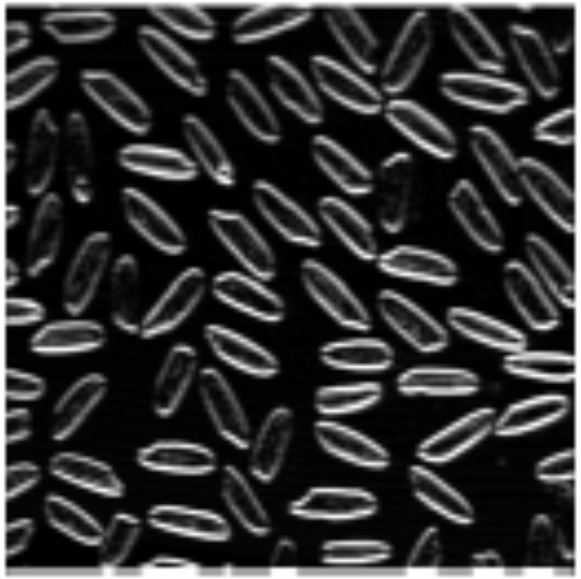


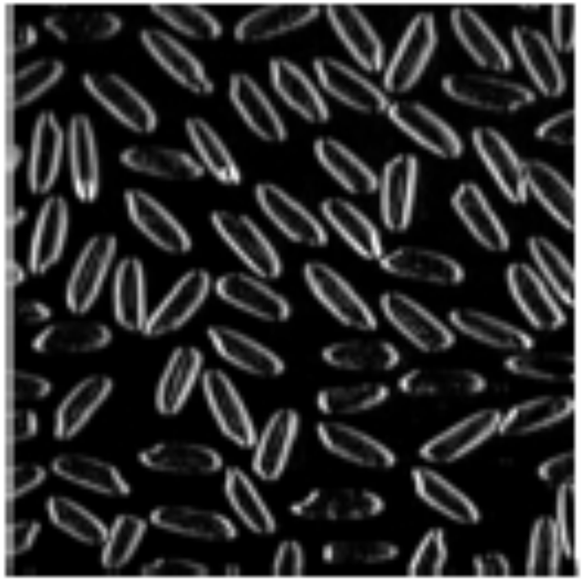
-1	-1	-1
0	0	0
1	1	1

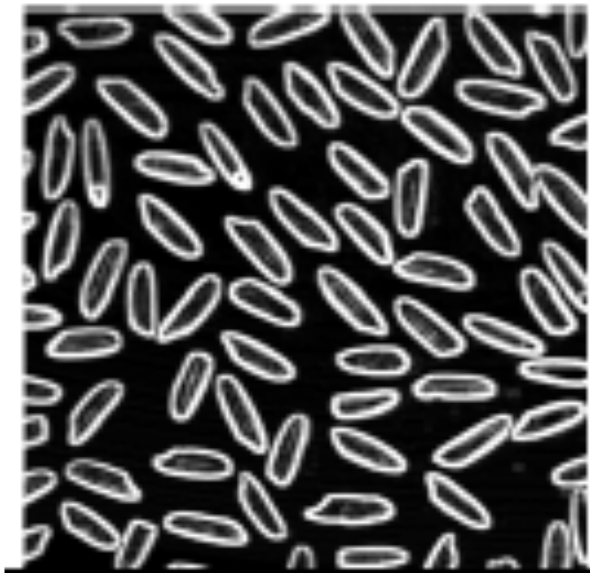
-1	0	1
-1	0	1
-1	0	1

Sobel operator







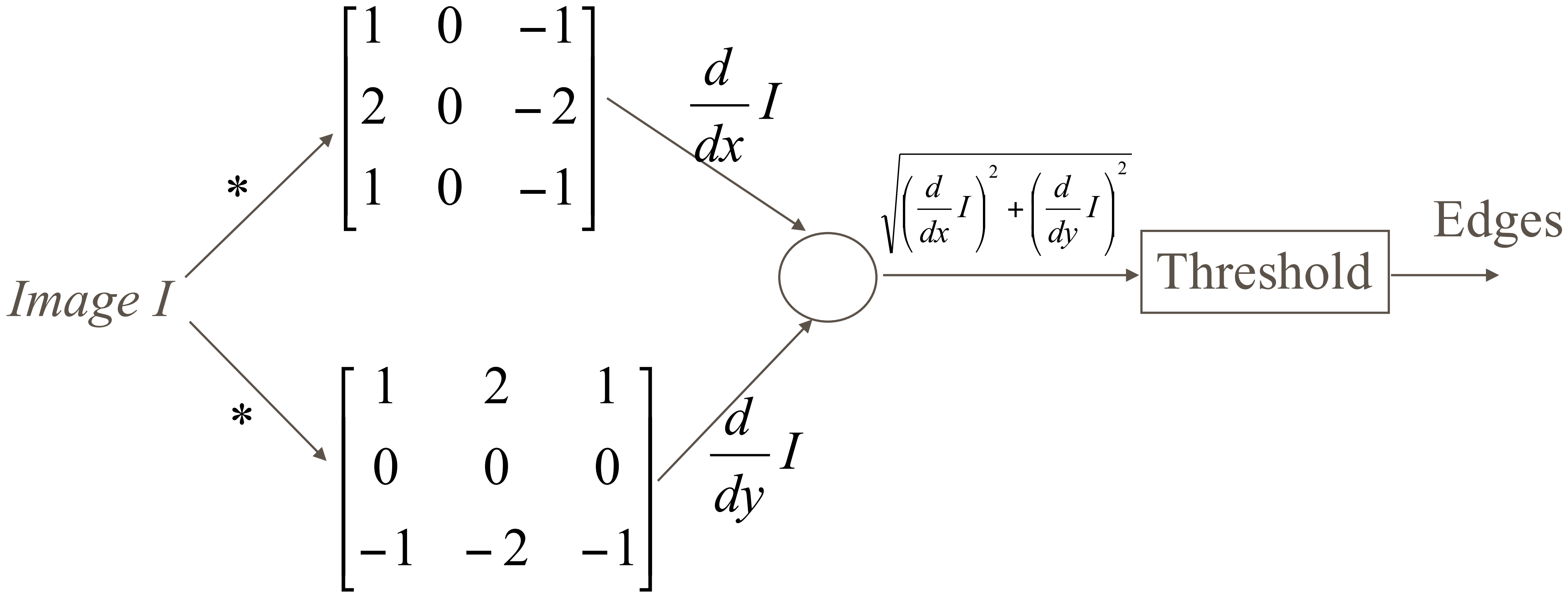


-1	-2	-1
0	0	0
1	2	1

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

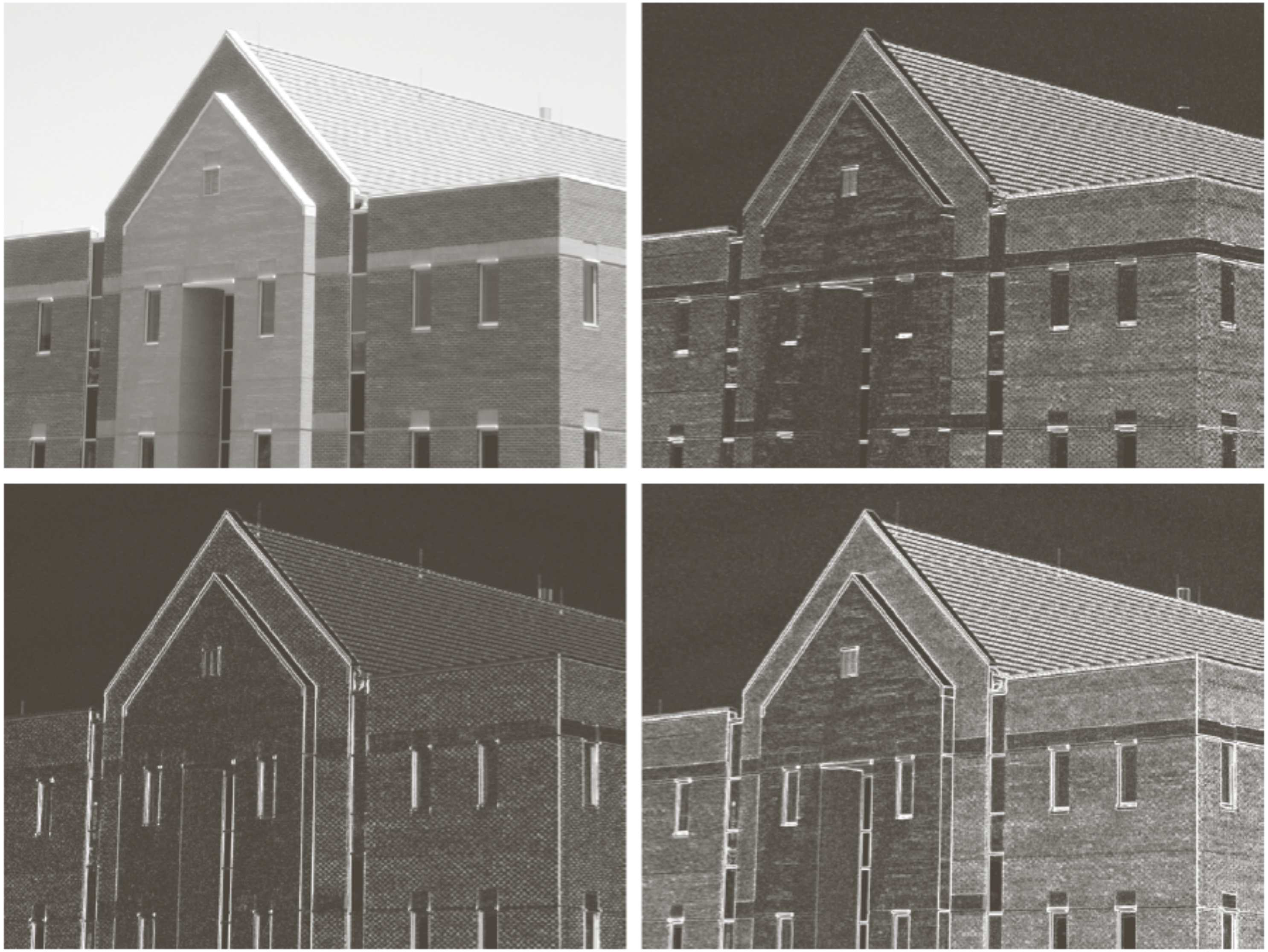
Sobel Edge detector

- Sobel Edge Detector



Sobel application

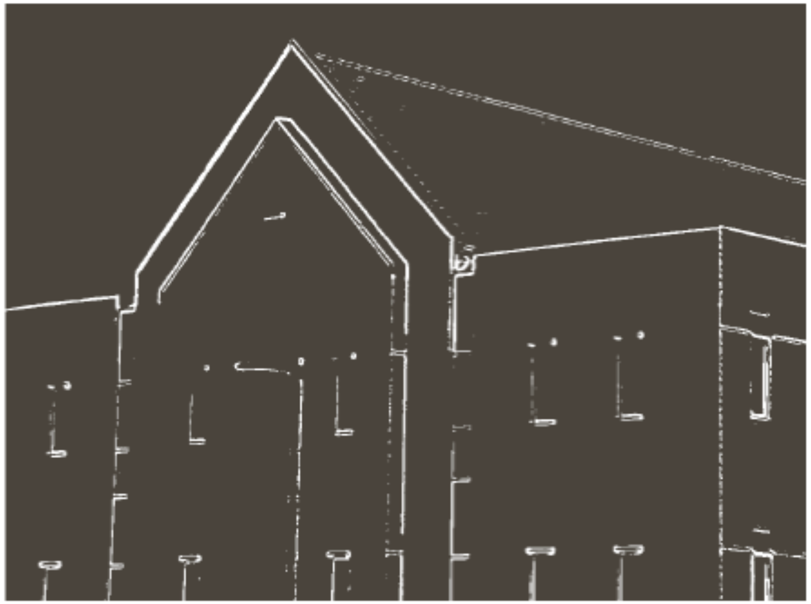
Original image, partial derivatives, and magnitude



Alternative: Gradient magnitude Thresholding

Let maintain the 33% of the edges with the highest magnitude

Starting from the original image (NOT BLURRED)



Starting from a BLURRED version of the original image

Laplacian

- def: $L(x, y) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$

We have already derived:

$$\frac{d^2 f}{dx^2} = \mathrm{f(x + 1, y) - 2f(x, y) + f(x - 1, y)}$$

Analogously we have:

$$\frac{d^2 f}{dy^2} = \mathrm{f(x, y + 1) - 2f(x, y) + f(x, y - 1)}$$

Laplacian, cont.

$$\frac{d^2 f}{dx^2} = \mathrm{f(x+1,y) - 2f(x,y) + f(x-1,y)}$$

$$\frac{d^2 f}{dy^2} = \mathrm{f(x,y+1) - 2f(x,y) + f(x,y-1)}$$

Adding the two components we have:

$$L(x,y) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} =$$
$$= f(x+1,y) + f(x-1,y) +$$
$$+f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Masks:

0	1	0
1	-4	1
0	1	0

Isotropic to 90° rotations

1	1	1
1	-8	1
1	1	1

Isotropic to 45° rotations

Masks, cont.:

0	-1	0
-1	4	-1
0	-1	0

Sign changes

-1	-1	-1
-1	8	-1
-1	-1	-1

Observation

- Coefficient sum = 0 => constant areas =0
- To maintain the image information:

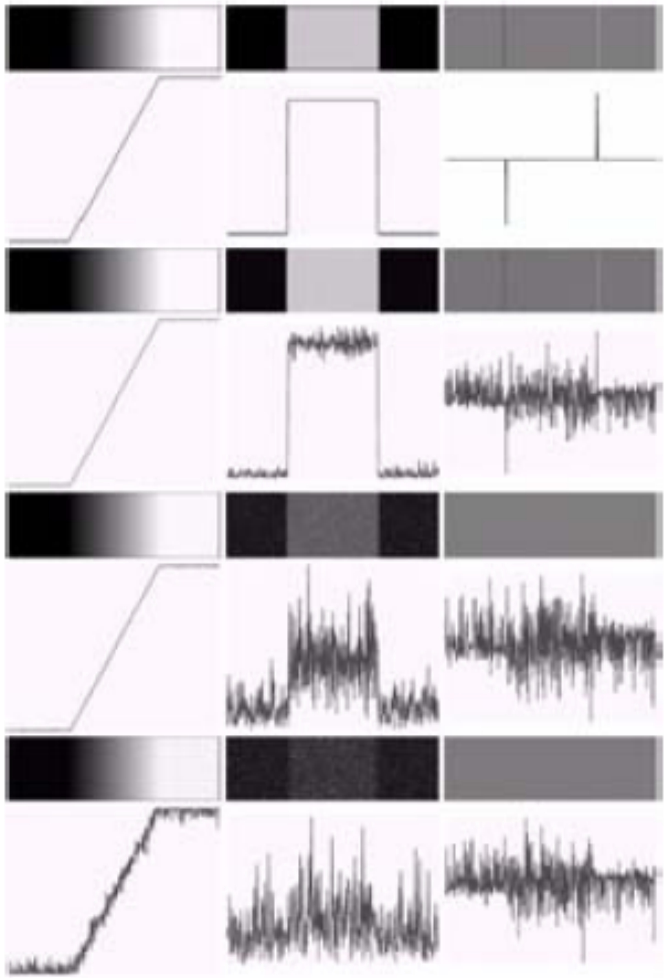
$$g(x,y) = \begin{cases} f(x,y) - L(x,y) & \text{se coeff centrale} < 0 \\ f(x,y) + L(x,y) & \text{se coeff centrale} > 0 \end{cases}$$

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

➔ Sharpening!

Noise effect



OBS:
Noise obstruct the correct behavior of the derivative filters

➔ Remove noise BEFORE applying the derivative filters

Marr model

- **Marr approach:**
  Studied the vision mechanisms to reproduce them into the computational systems

  - ***Primal Sketch:*** scene *lowest level* description given by the vision system

  - **Obs**: the *edges* are a component of the primal sketch

Marr model

- **Marr conclusions**
  1. **Characteristics at different details levels**
     → operators at different scales
  2. Smoothing to remove non interesting details
     → Gaussian filter optimal for smoothing
  3. Edge = zero-crossing of the second derivative
  4. Laplacian optimal to this end (isotropic)

Marr-Hildreth Algorithm (1980)

- Image smoothing by means of a Gaussian filter
- Edge extraction by means of the Laplacian filter
- Localization of the zero-crossing points

Marr-Hildreth Algorithm

- Let call I the input image and G(x,y) the bidimensional Gaussian filter
- Following the steps, we should proceed as:

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

  - where $\nabla^2 = \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2}$ is the Laplacian,

  - and $I * G(x,y)$ applies the Gaussian filtering

Marr-Hildreth Algorithm

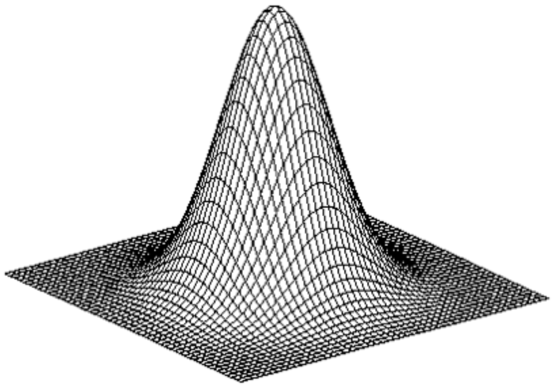
- Given the linearity, we can invert the operators order:

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

- That is, we construct the *Laplacian of the Gaussian* (**LoG**) and apply it only once to the image.
- LoG:

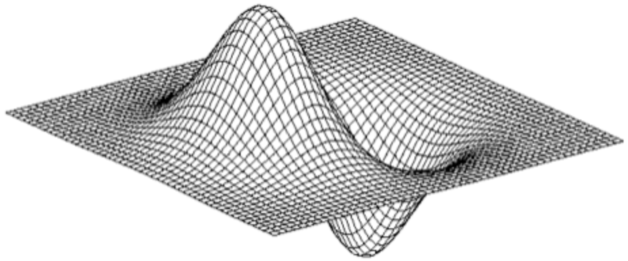
$$\nabla^2 G(x,y) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

LoG filter



Gaussian

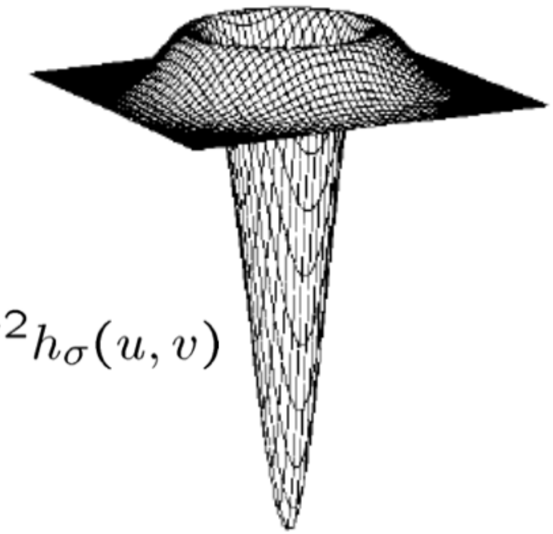
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

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

Laplacian of Gaussian

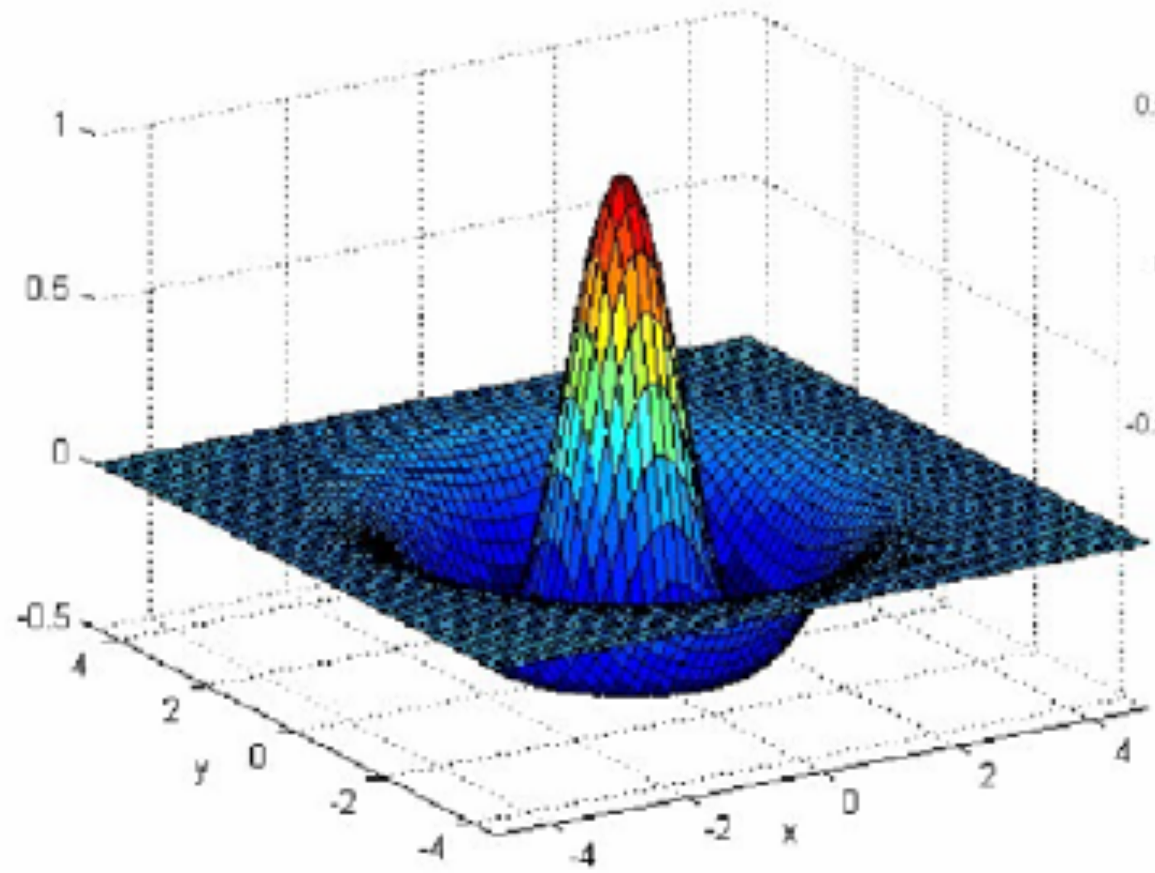


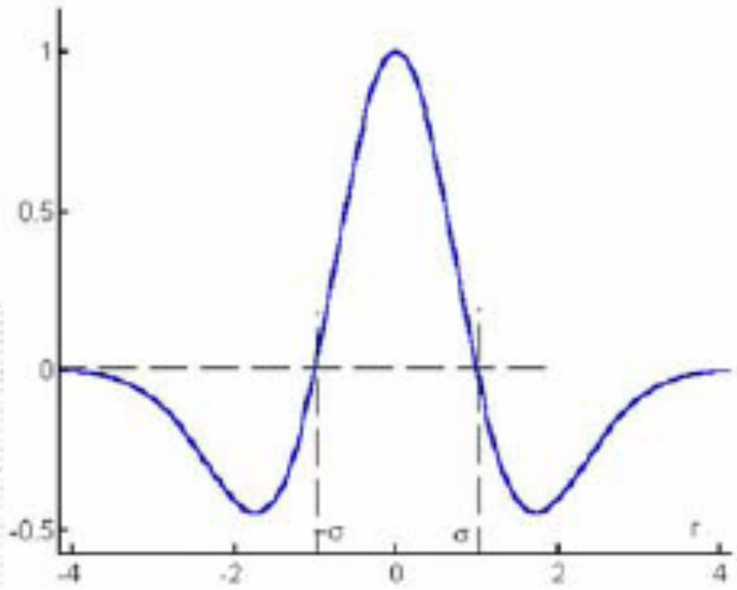
$$\nabla^2 h_\sigma(u,v)$$

∇^2 is the **Laplacian** operator:

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

LoG filter





LoG filter is also called *Mexican Hat* for its shape

Mask example

Example of a 5x5 mask approximating the LoG function shape:
central positive value, surrounded by negative values augmenting going far away from the center till becoming 0

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

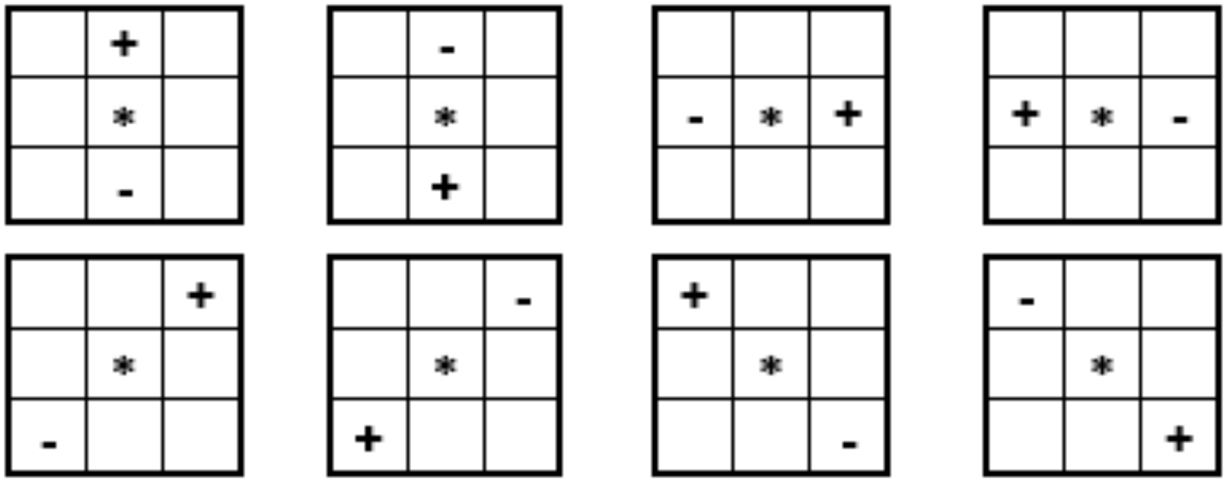
Obs: to produce masks of arbitrary dimensions:

- sample the Log eq.
- Or sample the G(x,y) and then apply the Laplacian

Zero crossing

- **Zero crossing search**: we have to take into account the fact that the edge could be in any of the 8 possible directions in the digital plane

- This corresponds to one of these **configurations**:

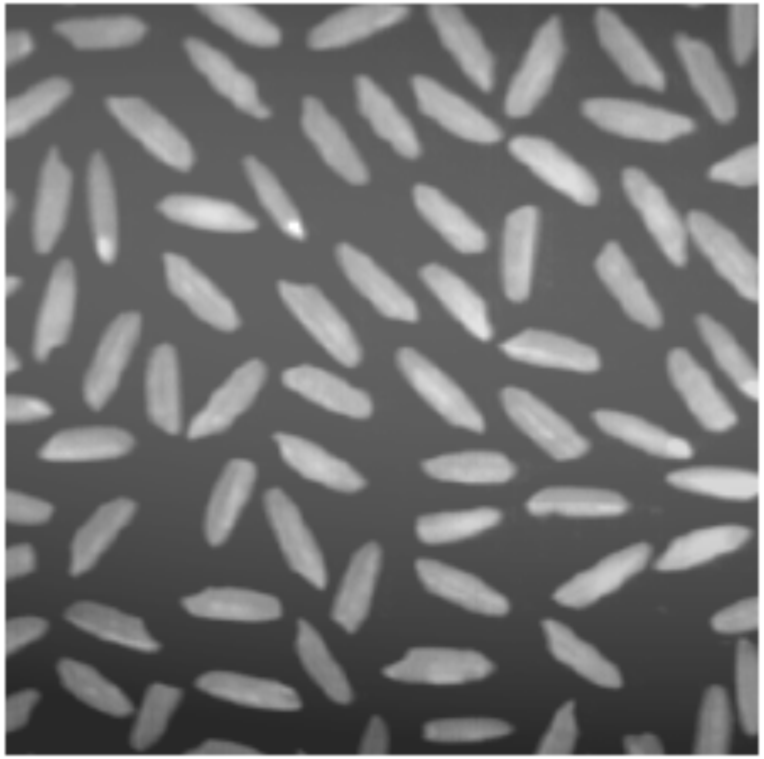


Further considerations

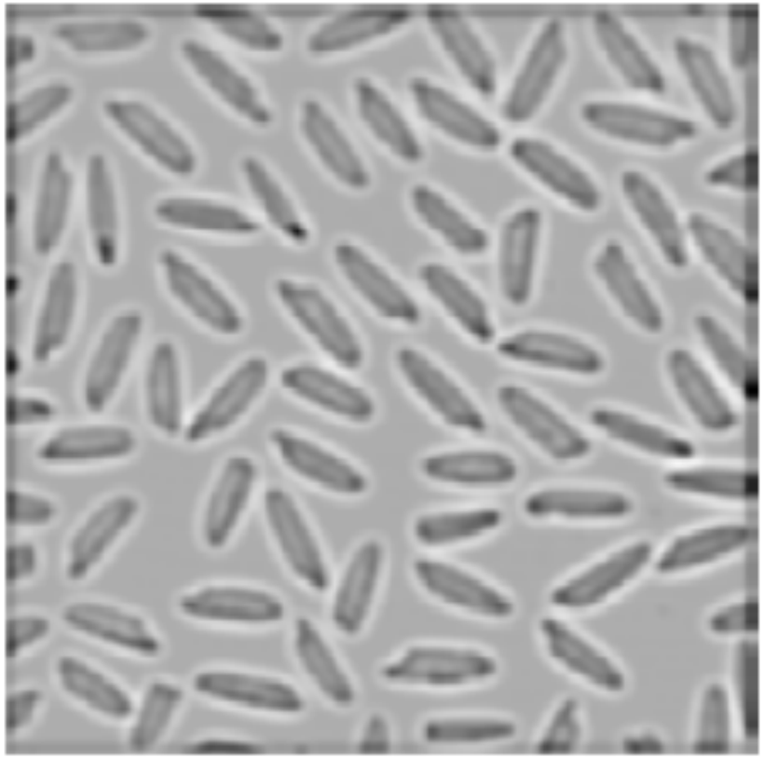
- The value of σ determines the highlighted level of detail
- To have more robust results, we could evaluate the LoG filter at different scales (different σ), finding the different zero crossing, and combining the results

- To describe the whole LoG shape, the mask should have dimensions WxW, with W>3c, where $c = 2\sqrt{2}\sigma$ is the centrale lobe amplitude. Alternatively, we could adopt $W = \lceil 3\sigma \rceil * 2 + 1$

Marr-Hildreth Algorithm Examples

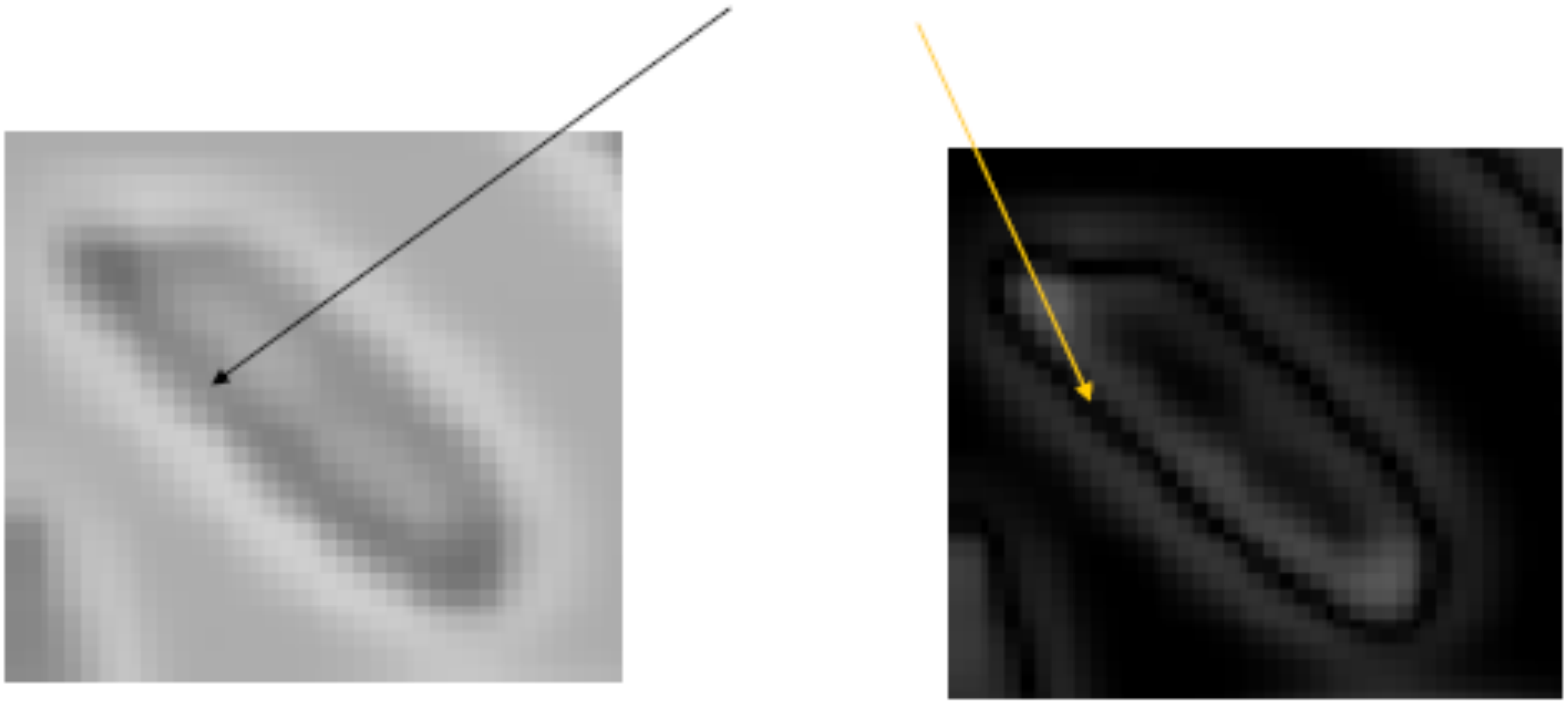


- Image filtered setting:
  $\sigma = 2, \quad W = 13$
- Gray scales rescaled



Marr-Hildreth Algorithm Examples

zero crossing

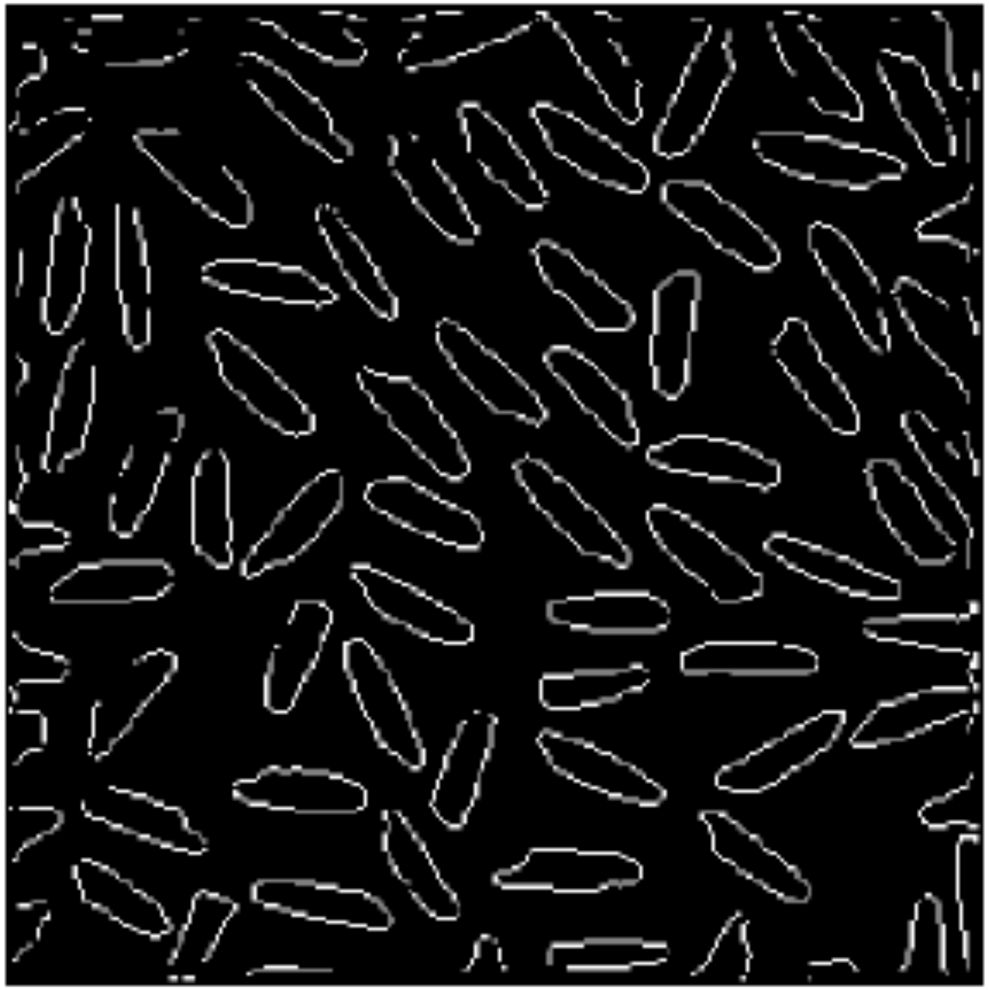


LoG　　　　abs(LoG)

Marr-Hildreth Algorithm Examples

Edge image

$\sigma=2$



Marr-Hildreth Algorithm, sigma dependency

σ=0.5

σ=1

Marr-Hildreth Algorithm, sigma dependency

$\sigma=2$

$\sigma=4$