

# Analisi di Immagini e Video (Computer Vision)

Giuseppe Manco

# Outline

- Image Processing avanzato
  - Edge detection
  - Canny Edge Detection

# Crediti

- Slides adattate da vari corsi
  - Analisi di Immagini (F. Angiulli) – Unical
  - Intro to Computer Vision (J. Tompkin) – CS Brown Edu
  - Computer Vision (I. Gkioulekas), CS CMU Edu

# Sharpening

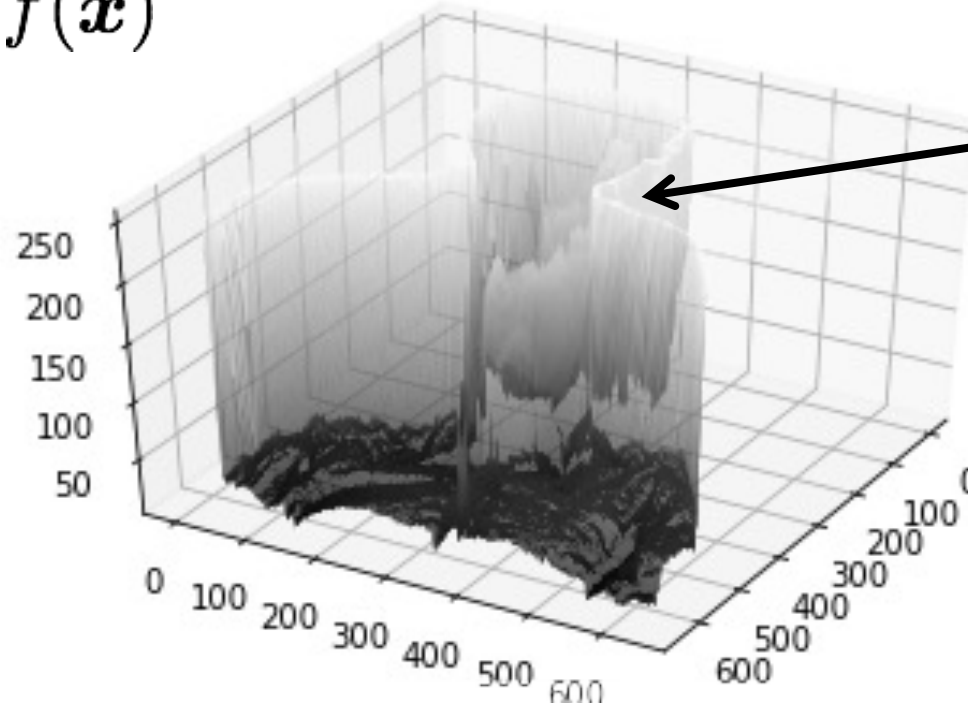
- Evidenziare transizioni d'intensità
  - Printing, medical imaging, industrial inspection, ecc
- Smoothing richiede averaging, analogo di integrazione spaziale
- Sharpening richiede derivazione spaziale

# Edge detection



Immagine

$f(\mathbf{x})$



Discontinuità  
molto ripide  
nell'intensità

dominio  $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

# Edge detection

- Come si possono identificare le discontinuità?

# Edge detection

- Come si possono identificare le discontinuità?
  - Si calcolano le derivate
    - Nei punti di discontinuità le derivate sono grandi

# Edge detection

- Come si possono identificare le discontinuità?
  - Si calcolano le derivate
    - Nei punti di discontinuità le derivate sono grandi
- Come si calcolano le derivate di un segnale discreto?



# Edge detection

- Come si possono identificare le discontinuità?
  - Si calcolano le derivate
    - Nei punti di discontinuità le derivate sono grandi
- Come si calcolano le derivate di un segnale discrete?
  - Calcolando le differenze finite

# Derivate di immagini digitali

- Le derivate di un'immagine digitale sono definite in termini di differenze
- Proprietà della **derivata prima**:
  - (1) pari a zero nelle aree di intensità costante
  - (2) diversa da zero in presenza di una rampa o gradino
  - (3) diversa da zero lungo le rampe
- Proprietà della **derivata seconda**:
  - (1) pari a zero nelle aree di intensità costante
  - (2) diversa da zero all'inizio ed alla fine di una rampa o gradino
  - (3) pari a zero lungo le rampe di pendenza costante

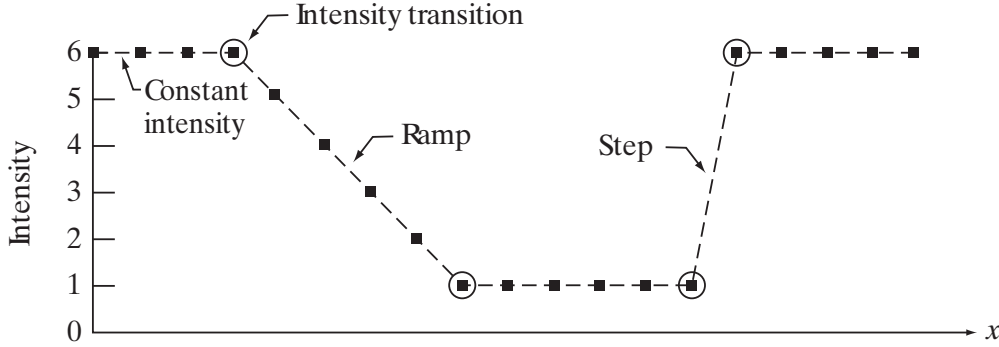
# Derivate di immagini digitali

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

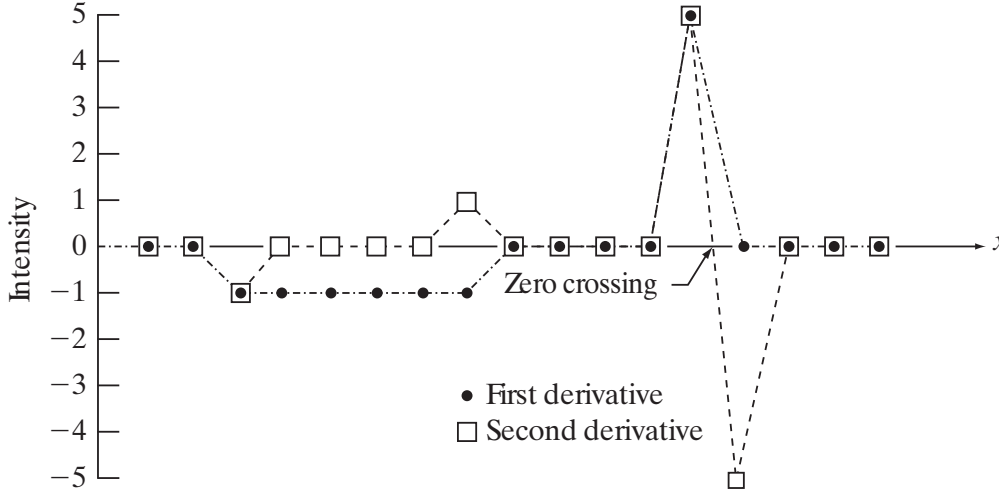
$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial f'(x)}{\partial x} = f'(x+1) - f'(x) = \\ &= f(x+2) - f(x+1) - [f(x+1) - f(x)] = \\ &= f(x+2) - 2f(x+1) + f(x)\end{aligned}$$

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

# Derivate di immagini digitali



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0	0	0
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	5	-5	0	0	0	0



$$\frac{\partial f}{\partial x} = f(x + 1) - f(x)$$

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

# Sobel filter

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

Sobel filter

=

1
2
1

Che cos'è?

\*

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

Derivata prima

# Sobel filter

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

Sobel filter

=

1
2
1

Blurring

\*

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

Derivata prima

Qual'è il responso su un'immagine 2D?

# Sobel filter

Orizzontale

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

Com'è fatto il filtro verticale?

# The Sobel filter

Orizzontale

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

Verticale

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$



# Filtri basati su derivate

Sobel

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

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

Scharr

3	0	-3
10	0	-10
3	0	-3

3	10	3
0	0	0
-3	-10	-3

Prewitt

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

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

Roberts

0	1
-1	0

1	0
0	-1

- Come si ottiene un filtro di dimensione più grande?

# Gradient-based filtering

1. Scegli la derivata

$$\mathbf{S}_x = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$\mathbf{S}_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

# Gradient-based filtering

1. Scegli la derivata

$$\mathbf{S}_x = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$\mathbf{S}_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

2. Convolvi con l'immagine

$$\frac{\partial f}{\partial x} = \mathbf{S}_x * f$$

$$\frac{\partial f}{\partial y} = \mathbf{S}_y * f$$

# Gradient-based filtering

1. Select your favorite derivative filters.

$$\mathbf{S}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\mathbf{S}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

2. Convolvi con l'immagine

$$\frac{\partial f}{\partial x} = \mathbf{S}_x * f$$

$$\frac{\partial f}{\partial y} = \mathbf{S}_y * f$$

3. Calcola direzione e ampiezza del gradiente.

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

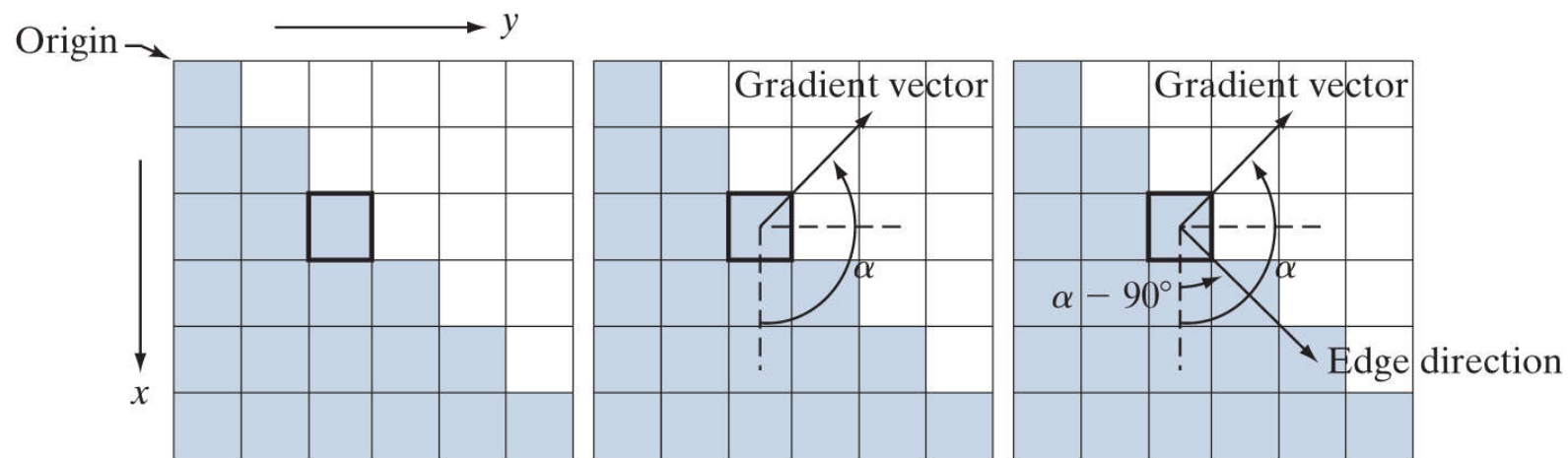
gradient

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

direction

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

amplitude



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

gradient

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

direction

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

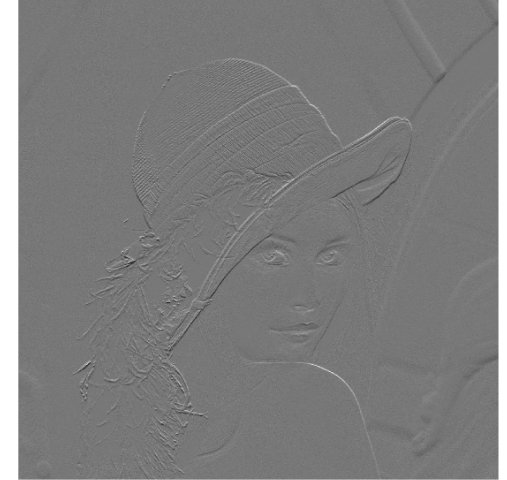
amplitude

# Esempio

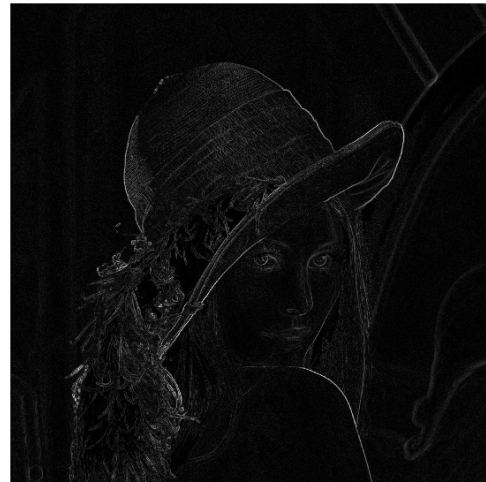
original



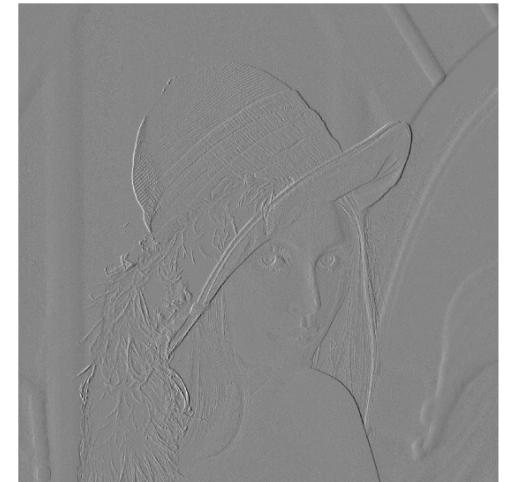
Derivata  
verticale



Ampiezza



Derivata  
orizzontale



# Laplaciano: derivata seconda di immagini digitali

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

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

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

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

# Laplaciano di immagini digitali

- **Filtro isotropico:** invariante rispetto alle rotazioni
- Laplaciano può essere utilizzato per image sharpening
  - Una proprietà desiderabile è che la risposta del filtro sia indipendente dalla direzione delle discontinuità (isotropia)

$$w = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$w = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad w = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

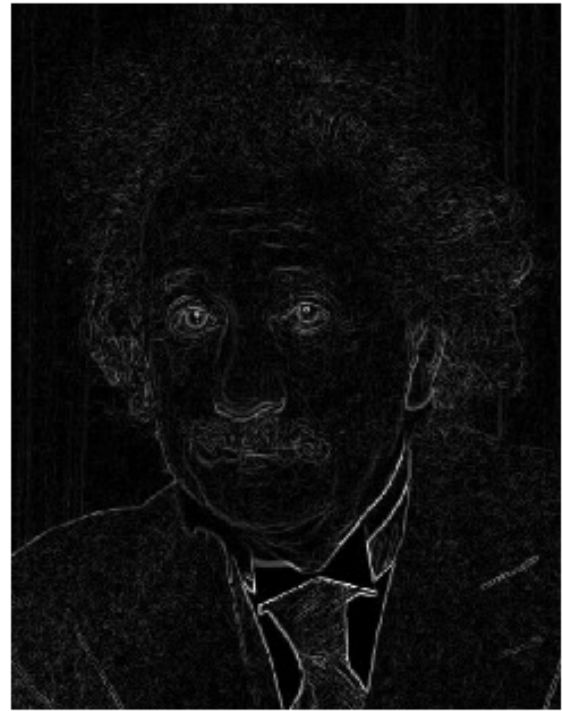
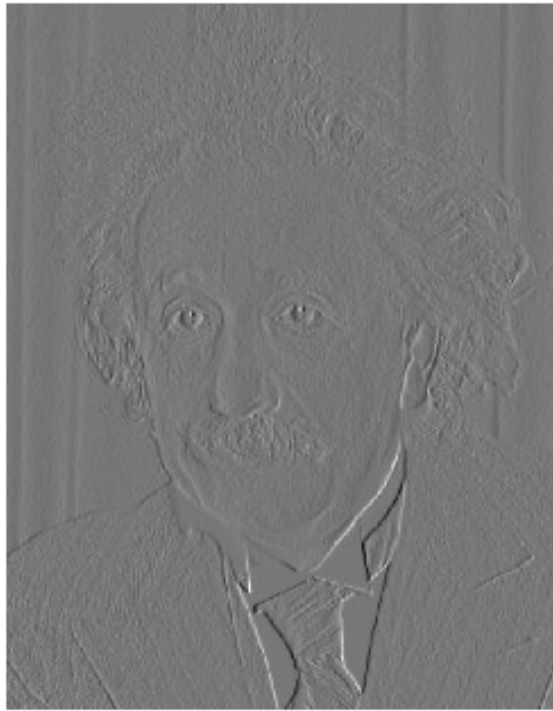
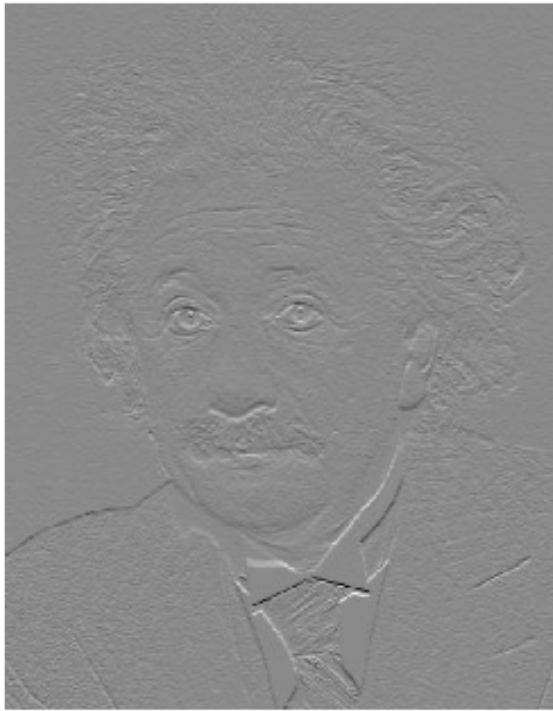
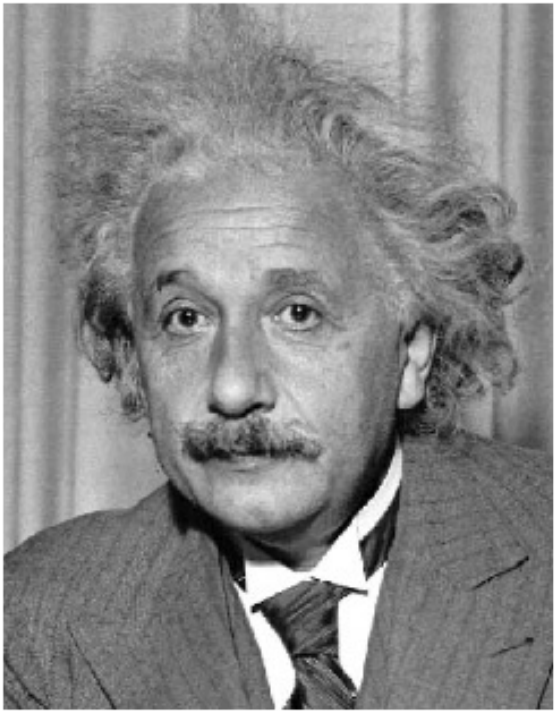


# Sharpening utilizzando il Laplaciano

- Gli operatori del secondo ordine sono più sensibili alle variazioni d'intensità
- Il laplaciano evidenzia le discontinuità di intensità; per recuperare lo sfondo sommiamo l'input

$$g(x, y) = f(x, y) + c [\nabla^2 f(x, y)]$$

- dove  $c=-1$  se il centro è negativo e  $c=+1$  se è positivo

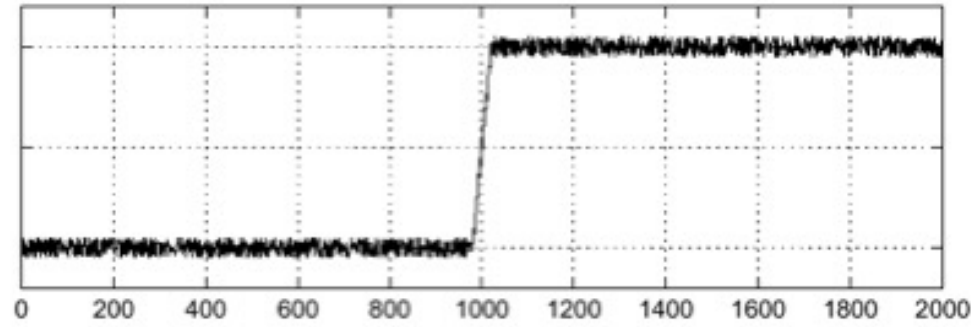


# Problema

- Le derivate sono calcolate sulla scala determinata dalla risoluzione dell'immagine
- Le immagini rumorose comportano un'alta instabilità della derivata

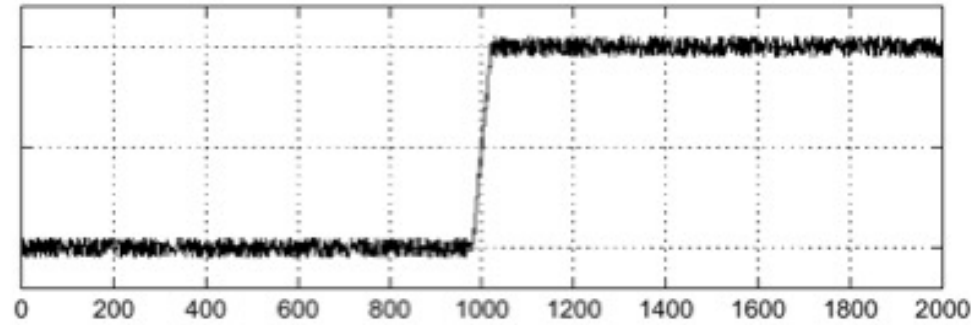
# Segnali instabili

intensità

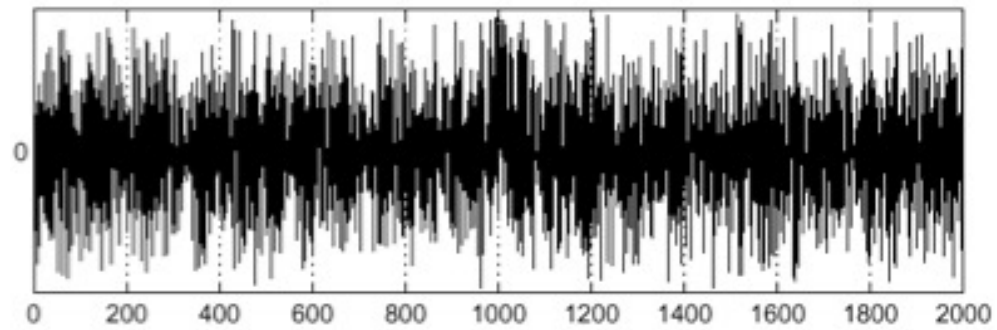


# Segnali instabili

intensità



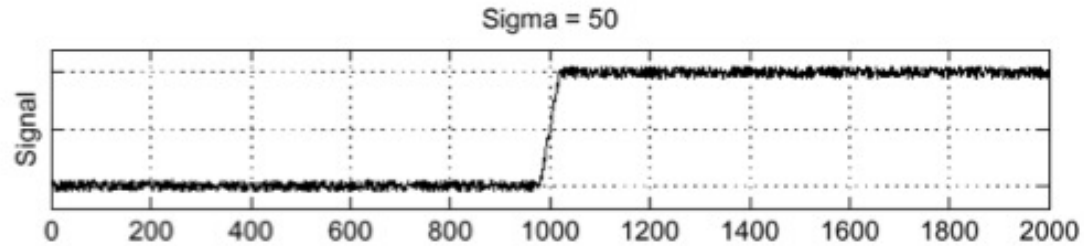
derivata



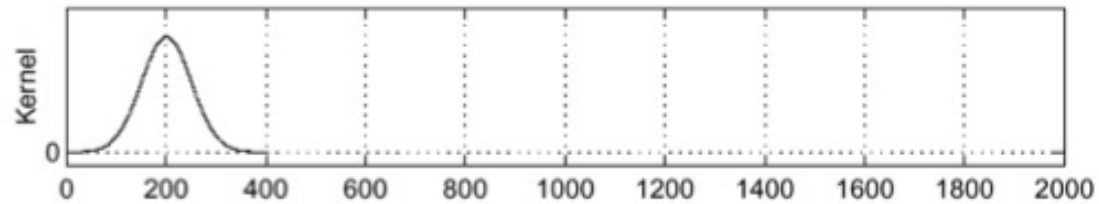
# Segnali instabili

Usiamo il filtro di blurring

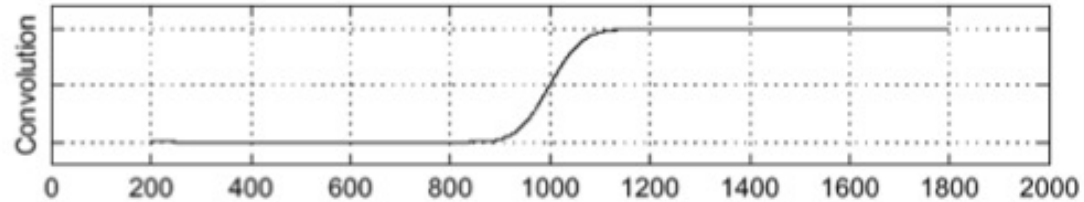
input



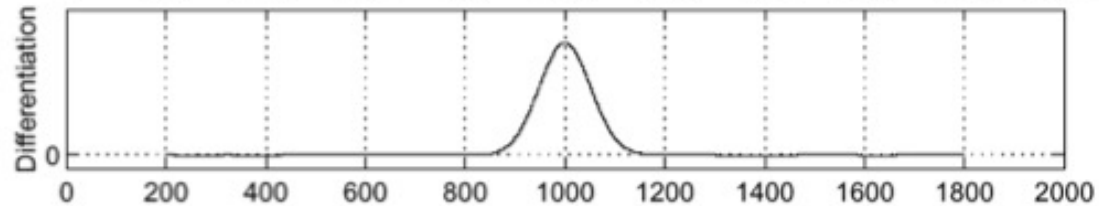
Gaussian



blurred



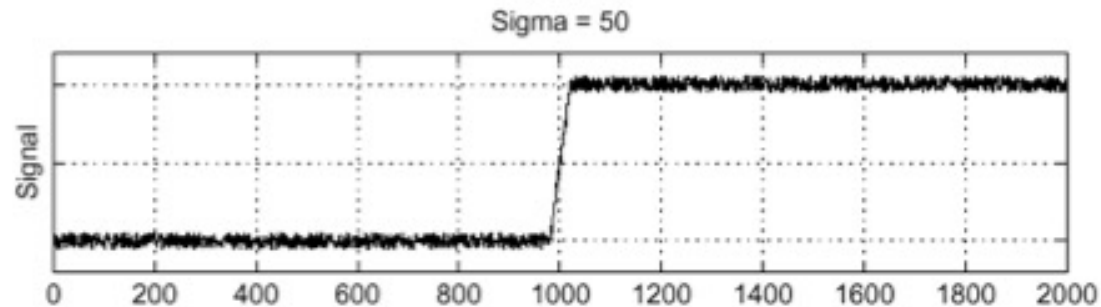
Derivata



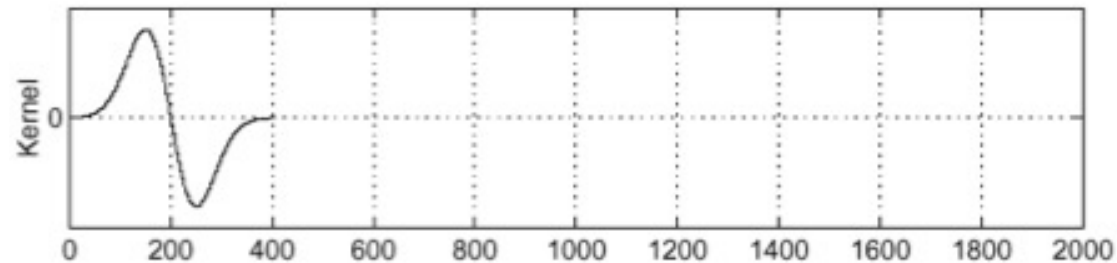
# Derivata del Filtro Gaussiano (DoG Filter)

Derivata della convoluzione:  $\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$

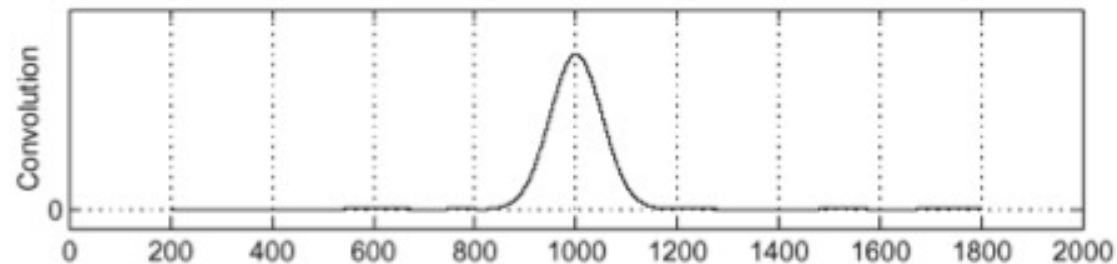
input



Derivata della  
gaussiana



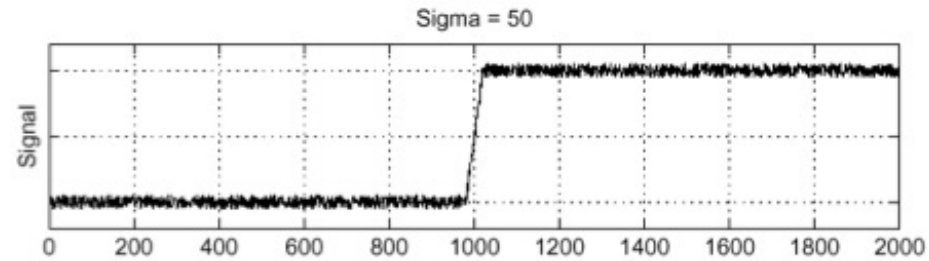
output



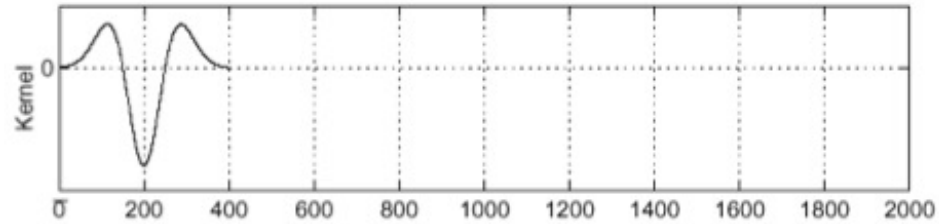
# Laplaciano del filtro Gaussiano (LoG filter)

Applichiamo lo stesso principio

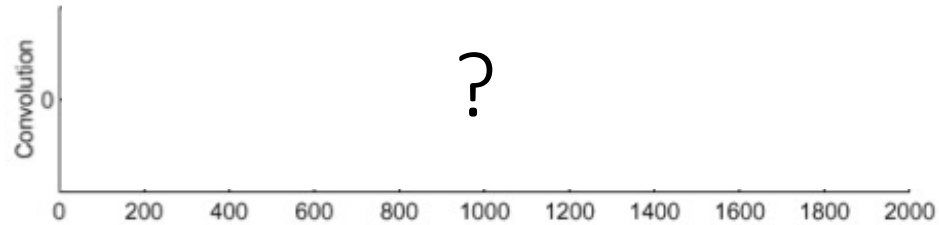
input



Laplacian of  
Gaussian



output

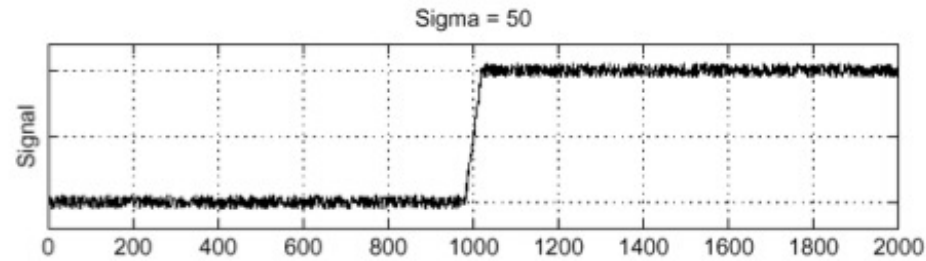




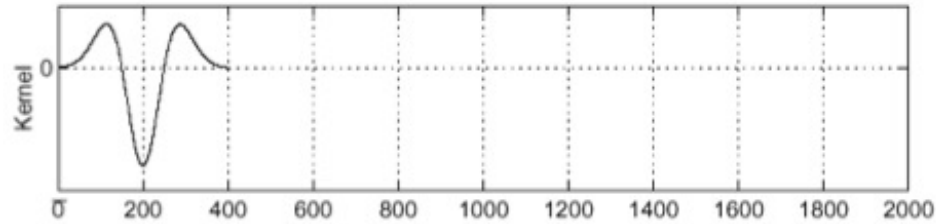
# Laplacian of Gaussian (LoG) filter

Applichiamo lo stesso principio

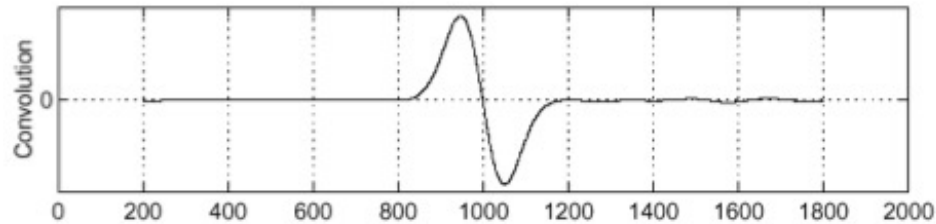
input



Laplacian of  
Gaussian



output



“zero crossings”

# Zero crossing

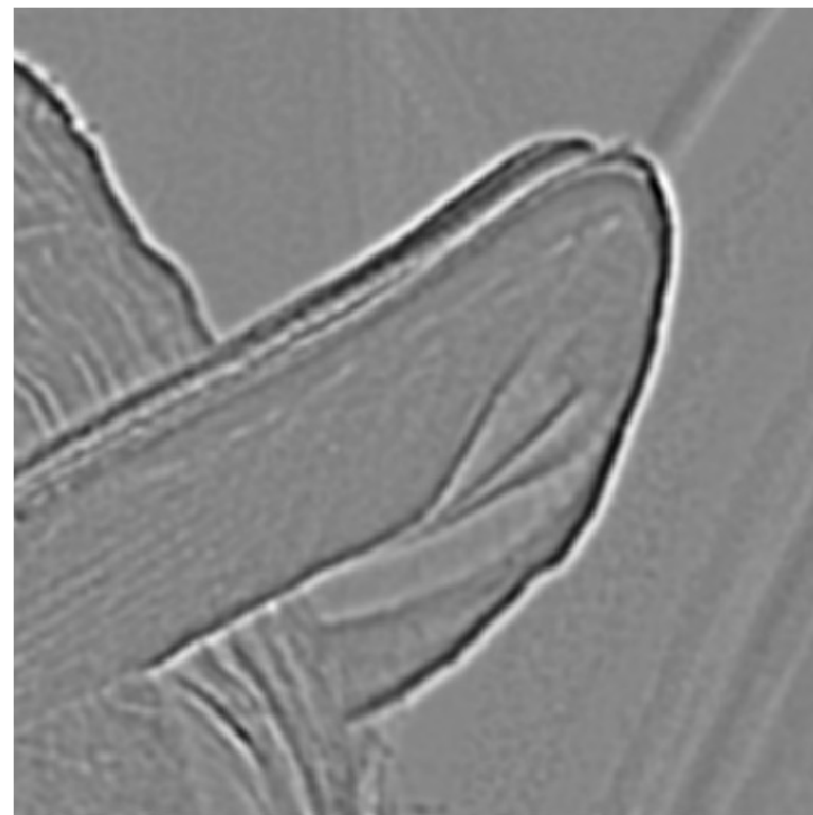
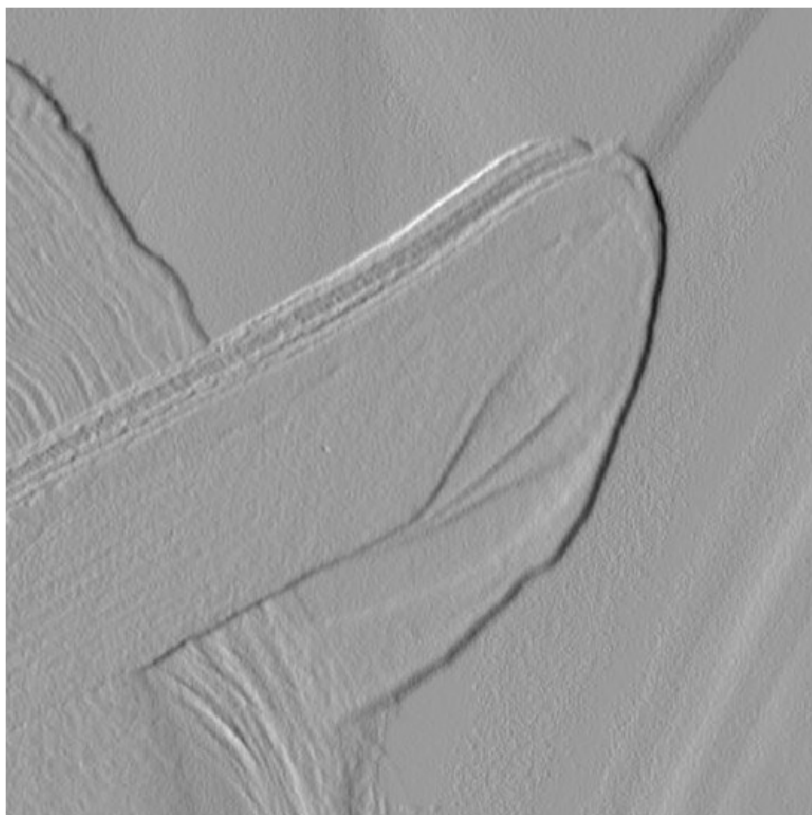
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	200	200	200	200	200	200	50	50
50	50	200	200	200	200	200	200	50	50
50	50	200	200	200	200	200	200	50	50
50	50	200	200	200	200	200	200	50	50
50	50	50	50	200	200	200	200	50	50
50	50	50	50	200	200	200	200	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50

(a) A simple image

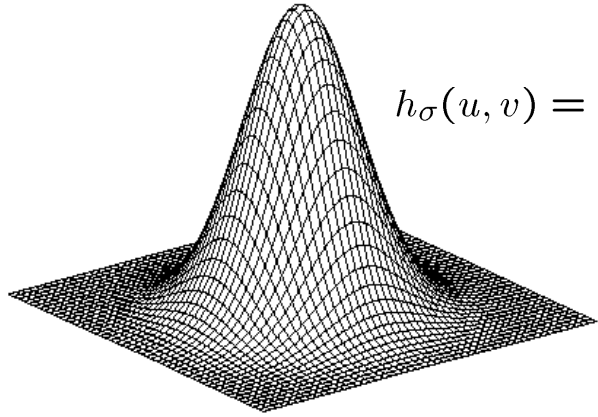
-100	-50	-50	-50	-50	-50	-50	-50	-50	-100
-50	0	150	150	150	150	150	150	0	-50
-50	150	-300	-150	-150	-150	-150	-300	150	-50
-50	150	-150	0	0	0	0	-150	150	-50
-50	150	-150	0	0	0	0	-150	150	-50
-50	150	-300	-150	0	0	0	-150	150	-50
-50	0	150	300	-150	0	0	-150	150	-50
-50	0	0	150	-300	-150	-150	-300	150	-50
-50	0	0	0	150	150	150	150	0	-50
-100	-50	-50	-50	-50	-50	-50	-50	-50	-100

(b) After laplace filtering

# Zero crossing

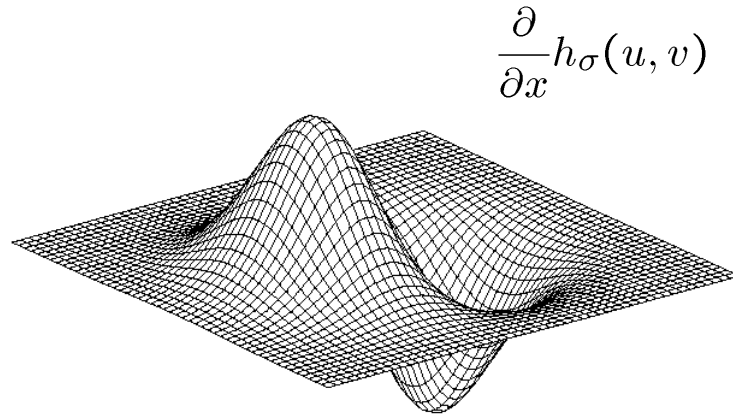


# Filtri gaussiani



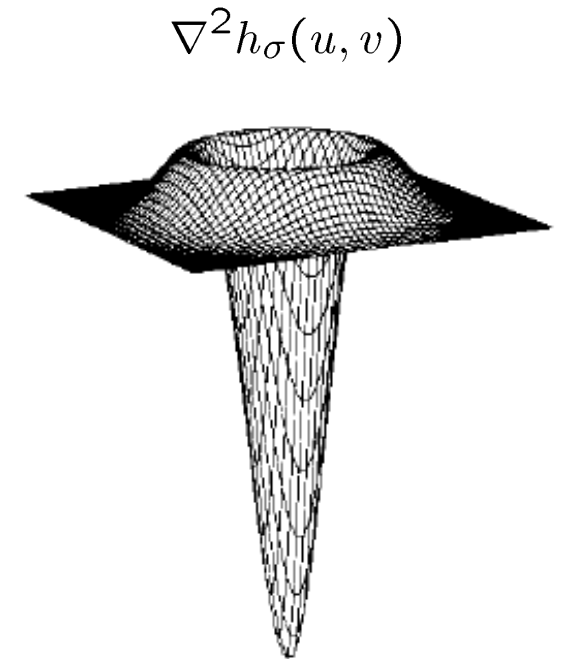
$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Gaussian



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

DoG



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

LoG