



# Image Filtering

**Michele Nappi, Ph.D**

**Università degli Studi di Salerno**

**mnappi@unisa.it**

**biplab.unisa.it**

**089-963334**

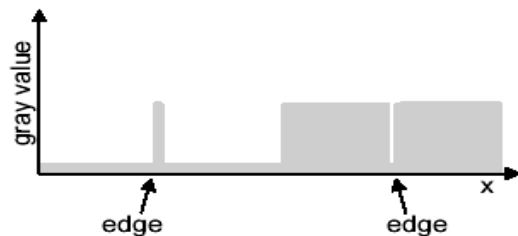


# Edge Detection

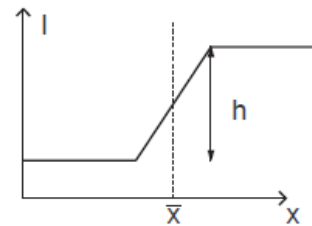
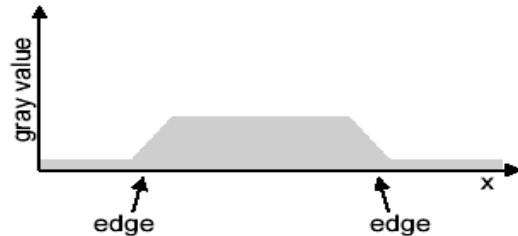
- Edge:
  - Aree di transizione tra regioni omogenee
- Punti Isolati
- Linee (Orizzontali, Verticali, Oblique)
- Contorni (insieme di punti con inizio e fine coincidenti)

# Edge Detection (cont.)

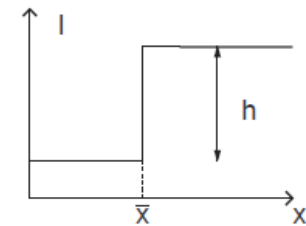
Line Edge



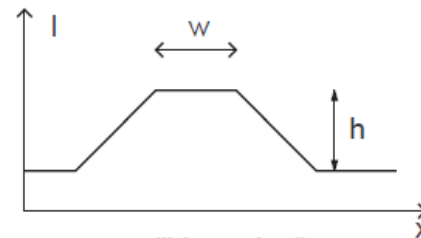
Step Edge



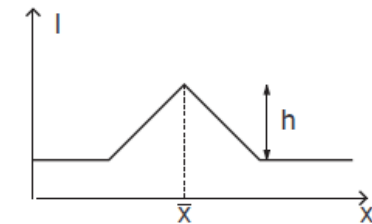
"Ramp Edge"



"Step Edge"



"Line edge"



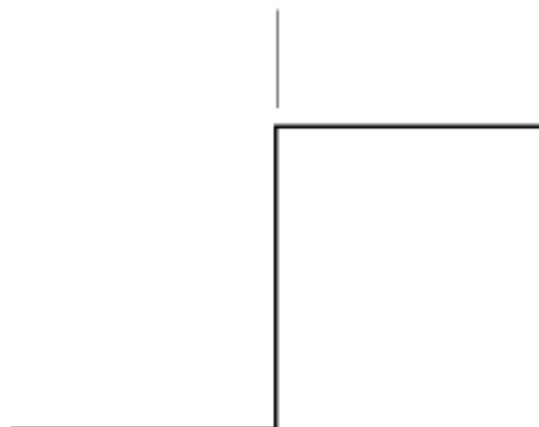
"Roof Edge"

Nel caso bidimensionale si assume che la discontinuità rimanga costante in un piccolo intervallo ortogonale al profilo del contorno. A destra un esempio di "step edge" bidimensionale discreto.

$$\begin{pmatrix} 3 & 3 & 3 & 3 & 3 & 7 \\ 3 & 3 & 3 & 3 & 7 & 7 \\ 3 & 3 & 3 & 7 & 7 & 7 \\ 3 & 3 & 7 & 7 & 7 & 7 \\ 3 & 7 & 7 & 7 & 7 & 7 \end{pmatrix}$$

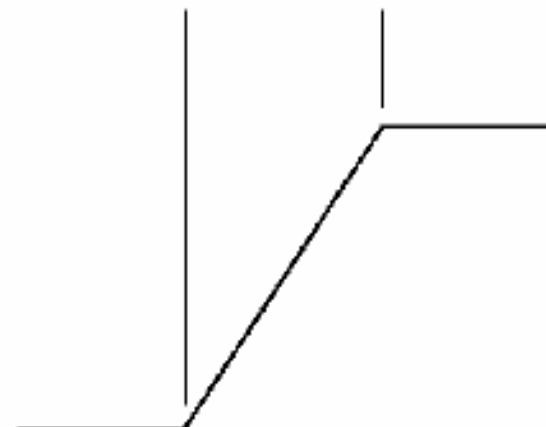
# Edge Detection (cont.)

Model of an ideal digital edge



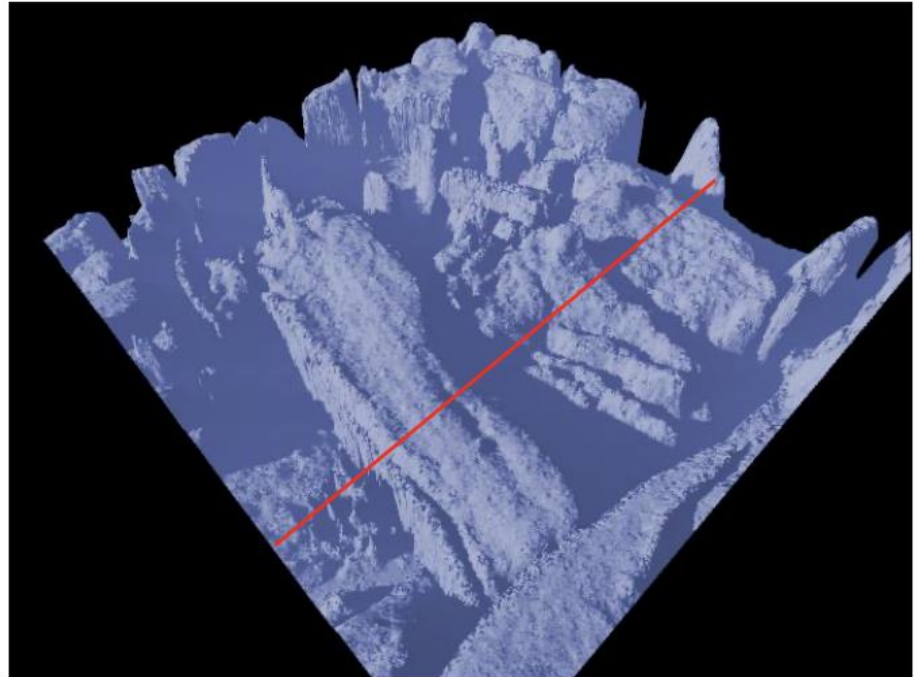
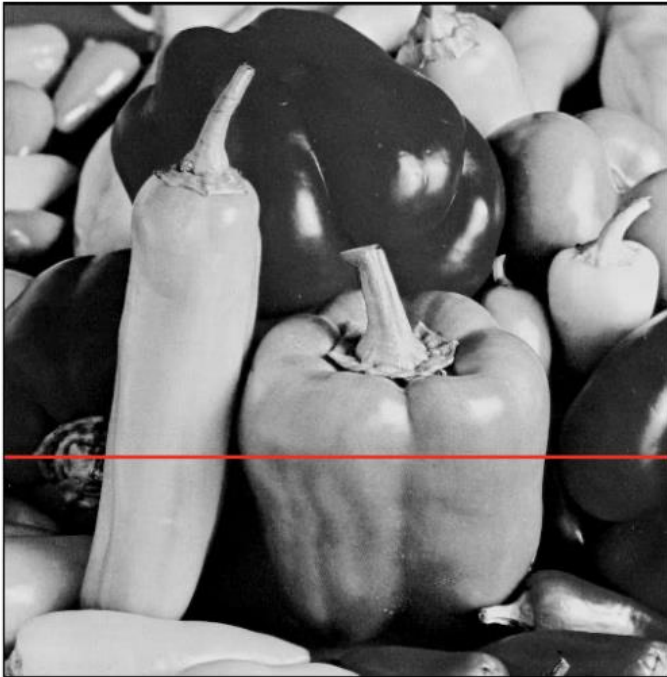
Gray-level profile  
of a horizontal line  
through the image

Model of a ramp digital edge

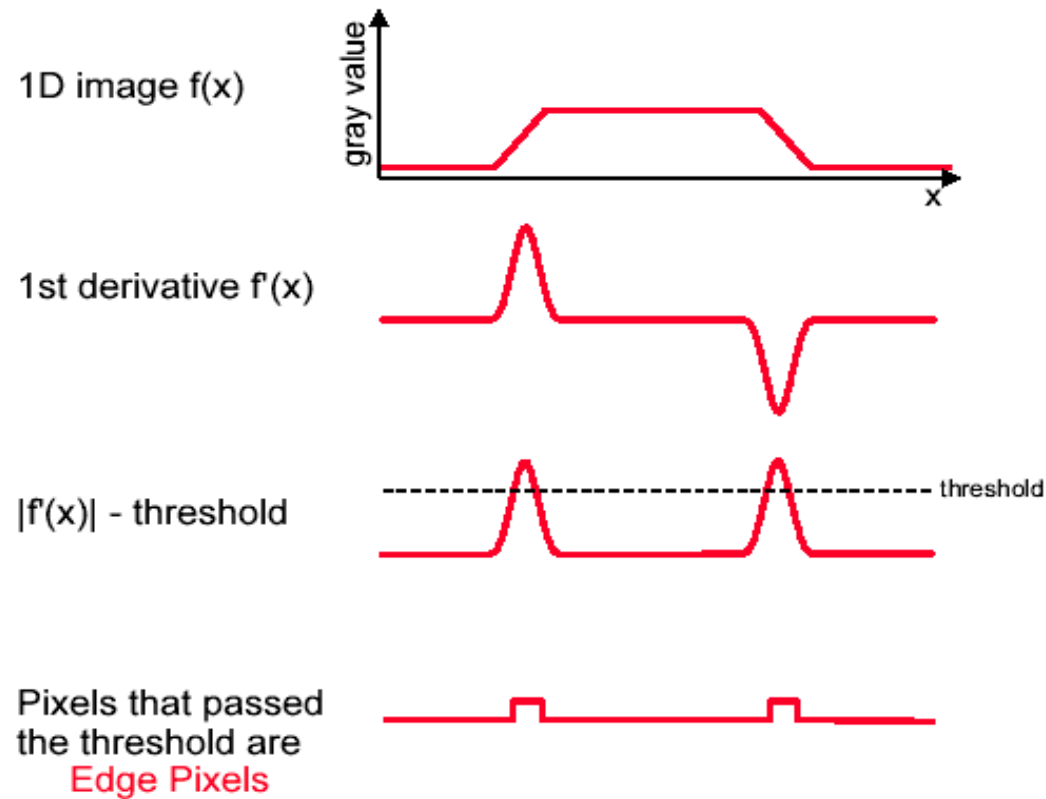


Gray-level profile  
of a horizontal line  
through the image

# Edge Detection (cont.)



# Edge Detection (cont.)



# Edge Detection: Punti Isolati



- o Le discontinuità probabilmente più semplici da rilevare sono i punti isolati
- o Il punto corrispondente alla posizione centrale della maschera può essere considerato un punto isolato quando, per esempio con riferimento alla maschera mostrata, si ha:

$$R > T$$

essendo  $T$  un opportuno valore di soglia (non negativo)

- o La motivazione è che il valore di grigio di un punto isolato è normalmente molto differente da quello dei suoi vicini

# Edge Detection: Punti Isolati (cont.)

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 100 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & -95 & -95 & -95 \\ 0 & 0 & -95 & 760 & -95 \\ 0 & 0 & -95 & -95 & -95 \end{bmatrix}$$

Large Positive values = **light** point on dark surround

Large Negative values = **dark** point on light surround



# Edge Detection: Linee

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

—      /      |      \

- La prima maschera risponde più intensamente alle linee orizzontali (spesse un pixel): con uno sfondo costante, il massimo di  $R$  si ha infatti quando la linea coincide con la riga centrale della maschera
- Analogamente, la seconda maschera risponde meglio a linee orientate a  $45^\circ$ , la terza a linee verticali, la quarta a linee orientate a  $45^\circ$
- Supponiamo di passare tutte e quattro le maschere su una immagine, e che in un certo punto si abbia che  $R_1$ ,  $R_2$ ,  $R_3$  e  $R_4$  le risposte delle quattro maschere.
- Se si verifica  $|R_i| > |R_j|$  per ogni  $i \neq j$  allora il punto in questione è associato alla maschera  $R_i$



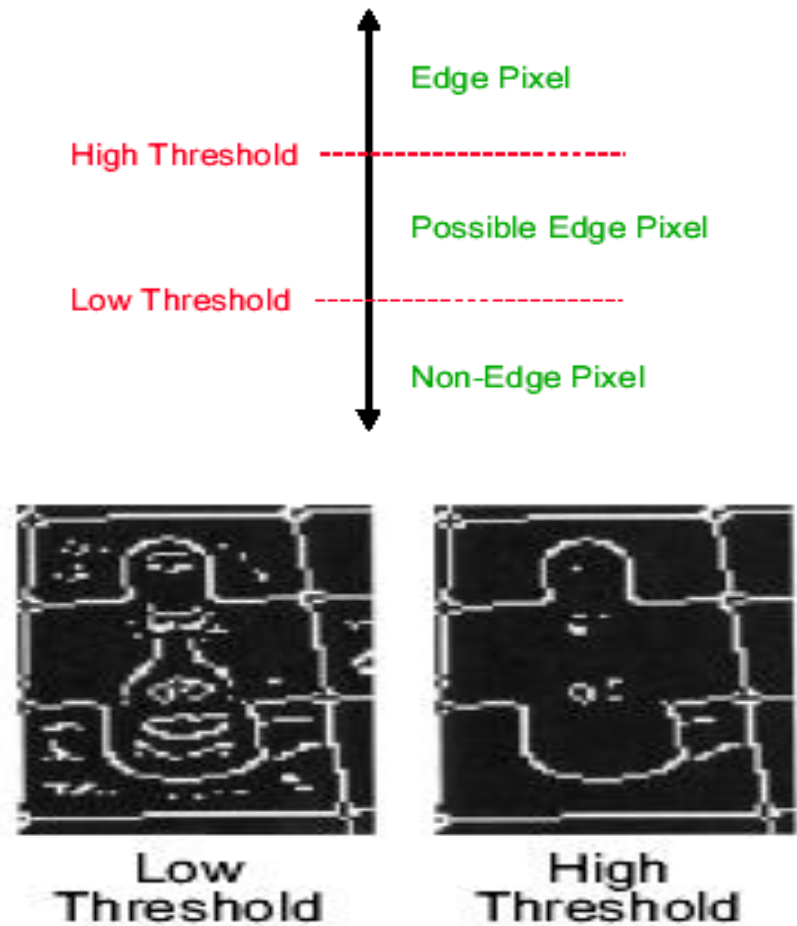
# Edge Detection: Contorni

- o L'estrazione dei contorni (edge) è sicuramente uno degli argomenti che hanno ricevuto più attenzione nella letteratura sull'image processing
- o Il contorno di un oggetto rappresenta infatti la separazione tra l'oggetto e lo sfondo o tra l'oggetto ed altri oggetti, per cui la sua estrazione è molto spesso il primo passo verso l'individuazione dell'oggetto
- o Un edge si presenta in una immagine come il confine tra due regioni caratterizzate da proprietà dei livelli di grigio in qualche modo distinguibili
- o Nel seguito ipotizzeremo che le regioni in questione siano sufficientemente omogenee, di modo che la determinazione della transizione tra le due regioni sia possibile sulla sola base della discontinuità dei valori di grigio
- o Le prime tecniche di edge detection che analizziamo sono basate sull'applicazione di un operatore *locale* di derivata.

# Edge Detection: Contorni (cont.)



Original



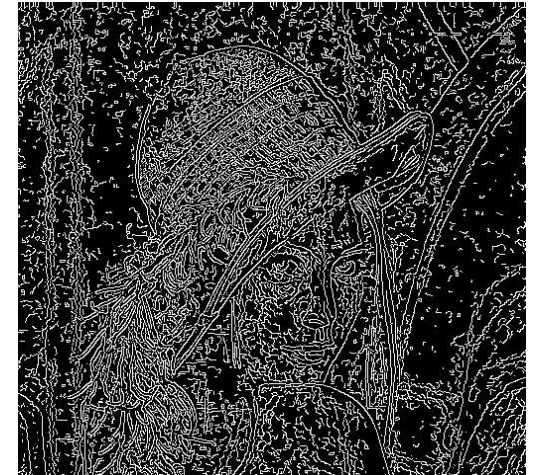
# Edge Detection: Contorni (cont.)



High threshold



Good threshold

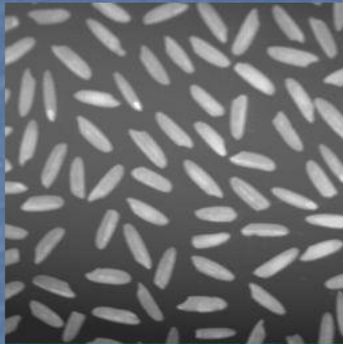


Low threshold

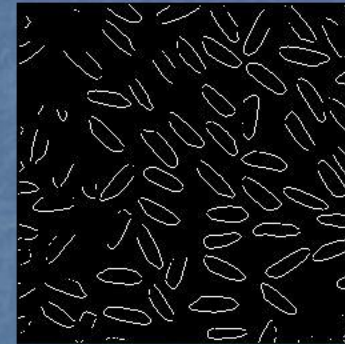
# Edge Detection: Contorni (cont.)



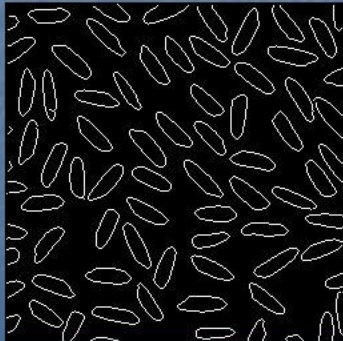
## Edge Detection With Hysteresis



Low threshold



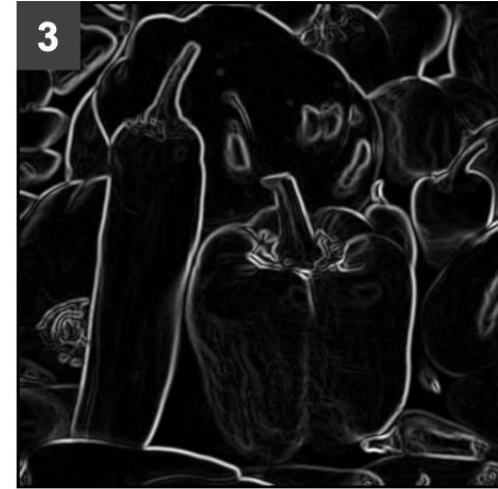
High threshold



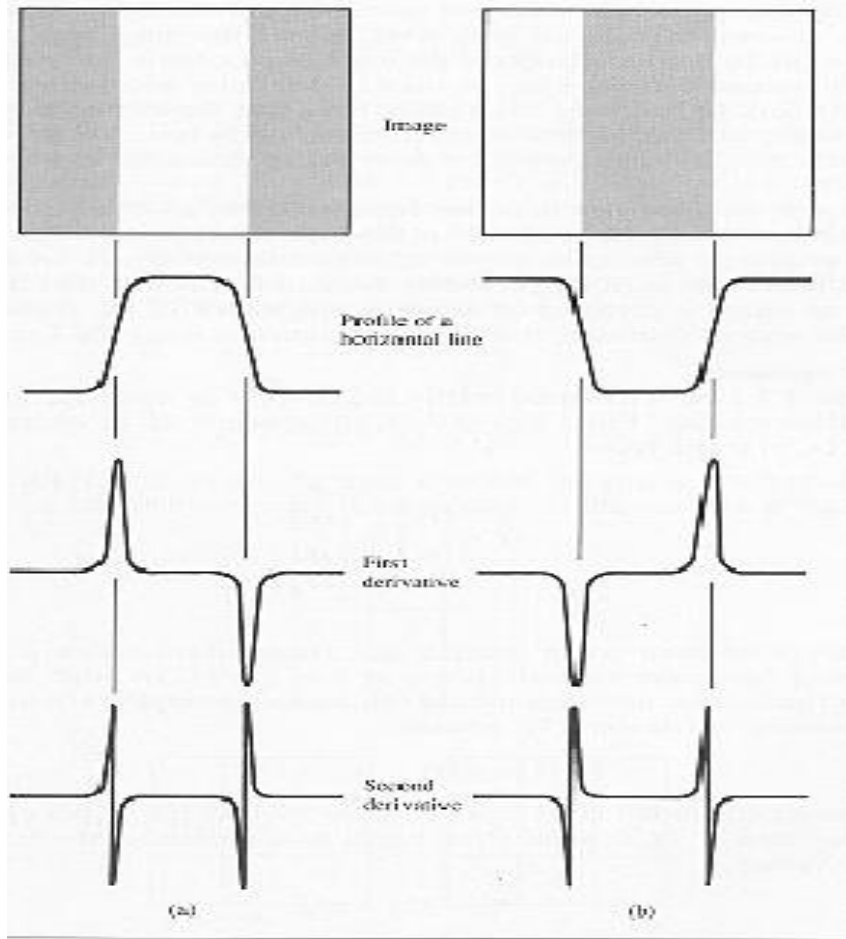
Hysteresis (high and low threshold)



# Edge Detection: Contorni (cont.)



# Edge Detection: Contorni (cont.)



- Il fatto che la derivata prima e la derivata seconda del profilo siano significativamente diverse da 0 *soltanto* in corrispondenza alle transizioni costituisce la motivazione dell'uso di operatori derivativi per l'estrazione dei contorni
- La derivata prima del profilo è positiva in corrispondenza ad una transizione scuro-chiaro, negativa in corrispondenza ad una transizione chiaro-scuro, nulla nelle zone a livello di grigio costante
- La derivata seconda è positiva in prossimità di un contorno, dalla parte *scuro* del contorno stesso, negativa dalla parte *chiaro* del contorno, nulla nelle zone a livello di grigio costante, ed esibisce un passaggio per lo zero o zero crossing *esattamente* in corrispondenza alle transizioni

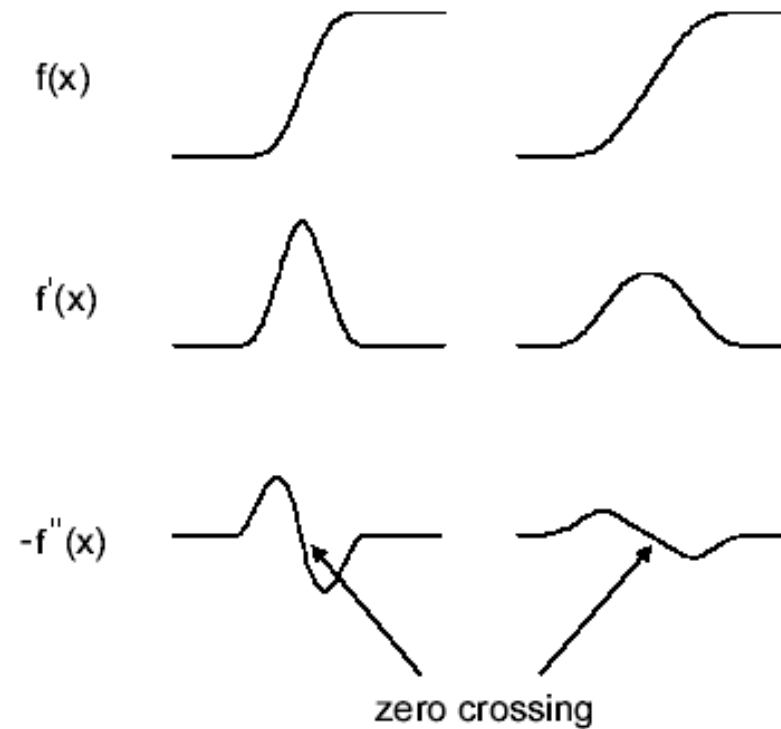
# Edge Detection: Contorni (cont.)



- Riassumendo, il valore della derivata prima può essere utilizzato per determinare la presenza di contorni in una immagine
- Gli zero crossing della derivata seconda ne possono consentire la precisa localizzazione
- Il segno della derivata seconda permette di stabilire l'appartenenza di un pixel al versante scuro o al versante chiaro di un contorno
- L'applicazione dei concetti precedentemente illustrati necessita tuttavia di alcune cautele, essenzialmente legate alla natura digitale delle immagini

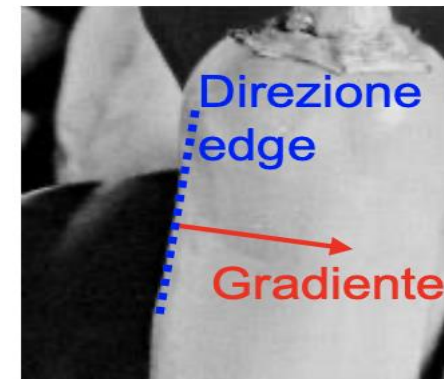


# Localizzazione dei Punti di Edge



# Edge Detection: Il Gradiente

Gradient  $\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

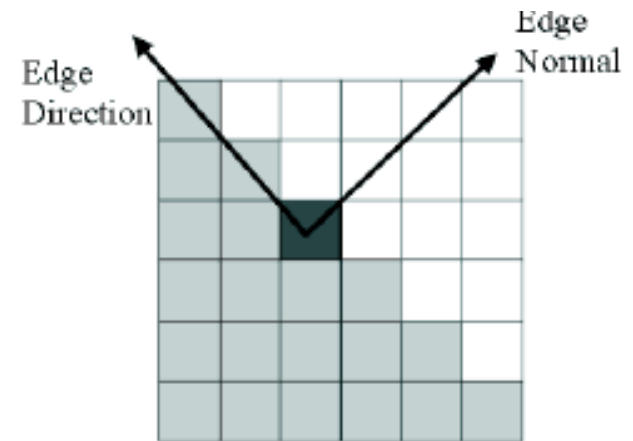


Gradient Magnitude

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

Gradient Direction

$$\text{tg}^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$



# Edge Detection: II Gradiente (cont.)

horizontal - differentiation approximation:

$$F_A = \frac{\partial f(x,y)}{\partial x} = f(x,y) - f(x-1,y)$$

convolution with  $\begin{bmatrix} 1 & -1 \end{bmatrix}$

vertical - differentiation approximation:

$$F_B = \frac{\partial f(x,y)}{\partial y} = f(x,y) - f(x,y-1)$$

convolution with  $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

---

Gradient	$(F_A, F_B)$
Magnitude	$((F_A)^2 + (F_B)^2)^{1/2}$
Approx. Magnitude	$ F_A  +  F_B $



# Edge Detection: Roberts e Prewitt

- Molto Sensibili al Rumore

## Roberts Edge Detector

$$F_A = f(x,y) - f(x-1,y-1)$$

$$F_B = f(x-1,y) - f(x,y-1)$$

$$A = \begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix}$$

## Prewitt Edge Detector

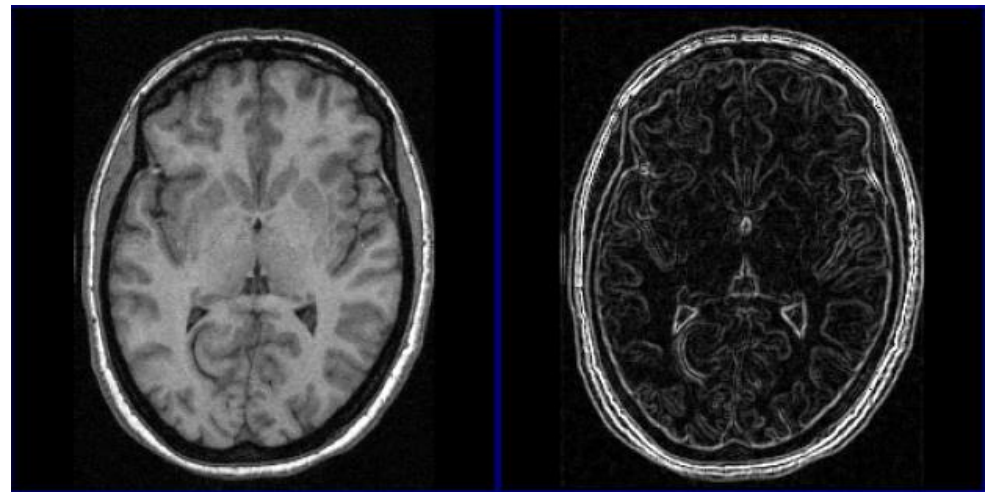
$$A = \begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

# Edge Detection: Sobel

## Sobel Edge Detector

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

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



# Edge Detection: Roberts, Sobel, etc.



Roberts Operators

$$\begin{array}{cc} \boxed{0} & 1 & \boxed{1} & 0 \\ -1 & 0 & 0 & -1 \end{array}$$

↖      ↗

Sobel Operators

$$\begin{array}{ccc} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{array} \quad \begin{array}{ccc} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{array}$$

→      ↓

Prewitt Operators

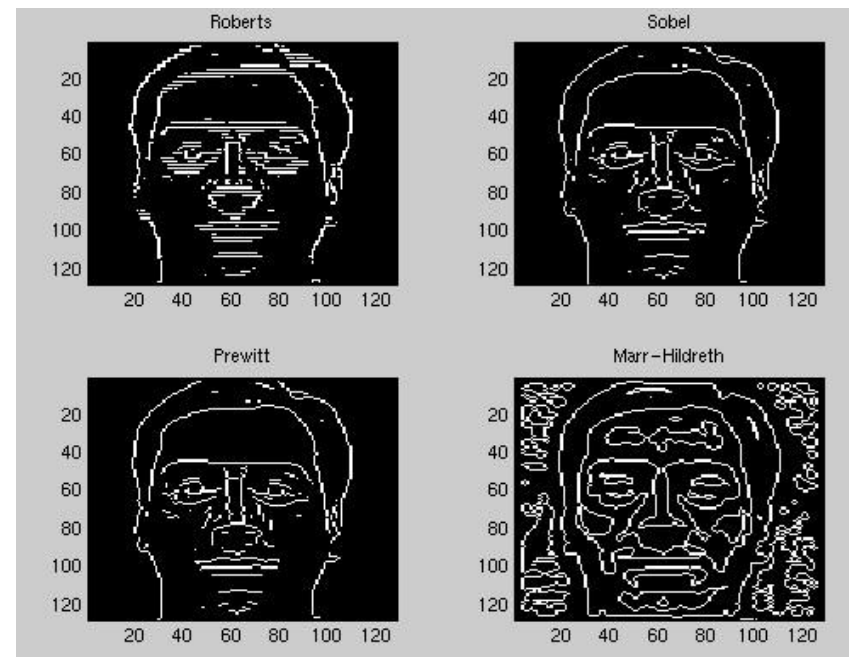
$$\begin{array}{ccc} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{array} \quad \begin{array}{ccc} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{array}$$

→      ↓

Isotropic Operators

$$\begin{array}{ccc} -1 & 0 & 1 \\ -\sqrt{2} & \boxed{0} & \sqrt{2} \\ -1 & 0 & 1 \end{array} \quad \begin{array}{ccc} -1 & -\sqrt{2} & -1 \\ 0 & \boxed{0} & 0 \\ 1 & \sqrt{2} & 1 \end{array}$$

→      ↓





# Edge Detection: Laplaciano

Approximation of second derivative (horizontal):

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

convolution with:  $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

Approximation of second derivative (vertical):

convolution with:  $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

---

Laplacian Operator

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

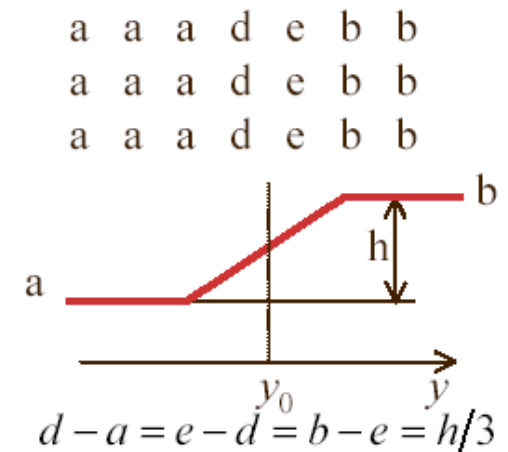
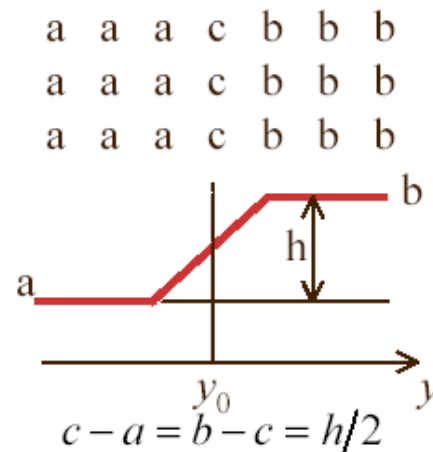
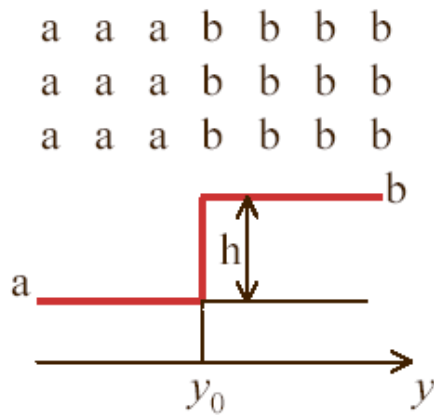
convolution with:  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

# Edge Detection: Laplaciano (cont.)

- ✓ Una versione (normalizzata) del laplaciano riferito agli 8-vicini, quindi con le differenze delle derivate prime mediate su tre righe e tre colonne, è:

$$\mathbf{H} = \frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

- ✓ Applichiamo questo operatore ai modelli di edge visti in precedenza:





# Edge Detection: Laplaciano (cont.)

## Operatori basati sul laplaciano

✓ Si ottiene, con riferimento alla riga centrale dei tre modelli:

grigi	a	a	a	b	b	b	b
edge	0	0	$-\frac{3}{8}h$	$\frac{3}{8}h$	0	0	0

gradino

a	a	a	c	b	b	b
0	0	$-\frac{3}{16}h$	0	$\frac{3}{16}h$	0	0

rampa più ripida  $c - a = b - c = h/2$

grigi	a	a	a	d	e	b	b
edge	0	0	$-\frac{h}{8}$	0	0	$\frac{h}{8}$	0

rampa meno ripida  
 $d - a = e - d = b - e = h/3$

Per l'edge a rampa più ripida, il laplaciano rileva correttamente il contorno in corrispondenza allo ZC

Per l'edge a gradino, lo ZC si trova fra i due pixel a sinistra e a destra del contorno, e per convenzione l'edge è collocato sul pixel a destra dello ZC

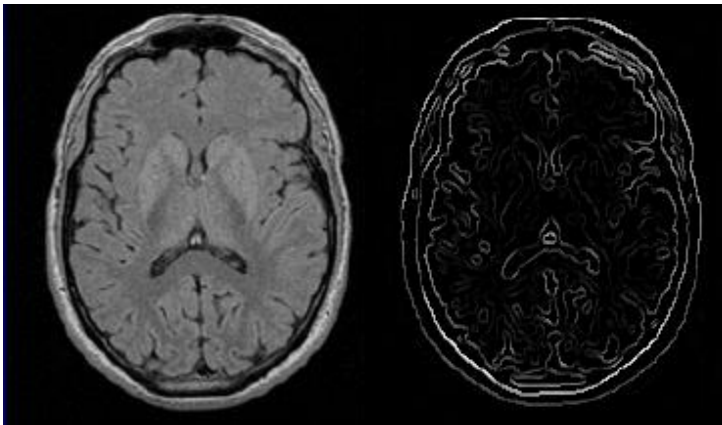
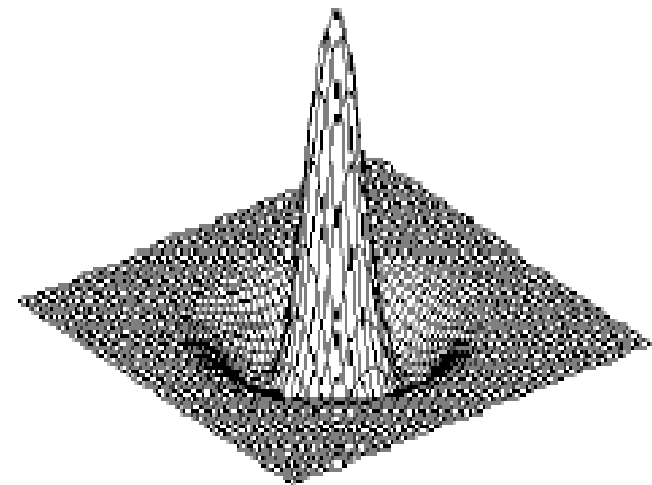
Neanche nell'ultimo caso lo ZC è localizzato esattamente su un pixel, anzi si ha un edge *doppio*

# Edge Detection: Laplaciano (cont.)

Variations on Laplace Operators:

All are approximations of:

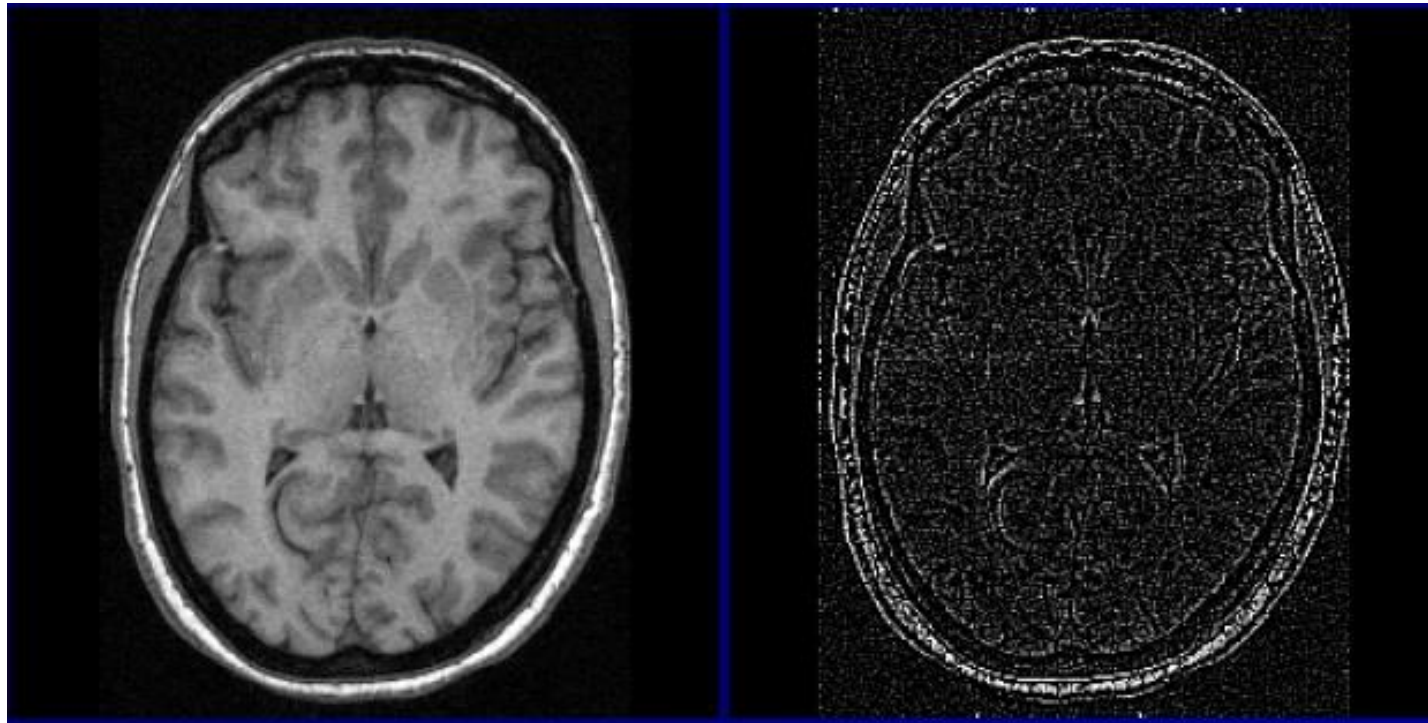
$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



# Edge Detection: Laplaciano (cont.)



# Edge Detection: Laplaciano (cont.)



# Sharpening

- Filtri Highpass
  - Evidenziano i dettagli fini

$$\frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

# Sharpening



$$M_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

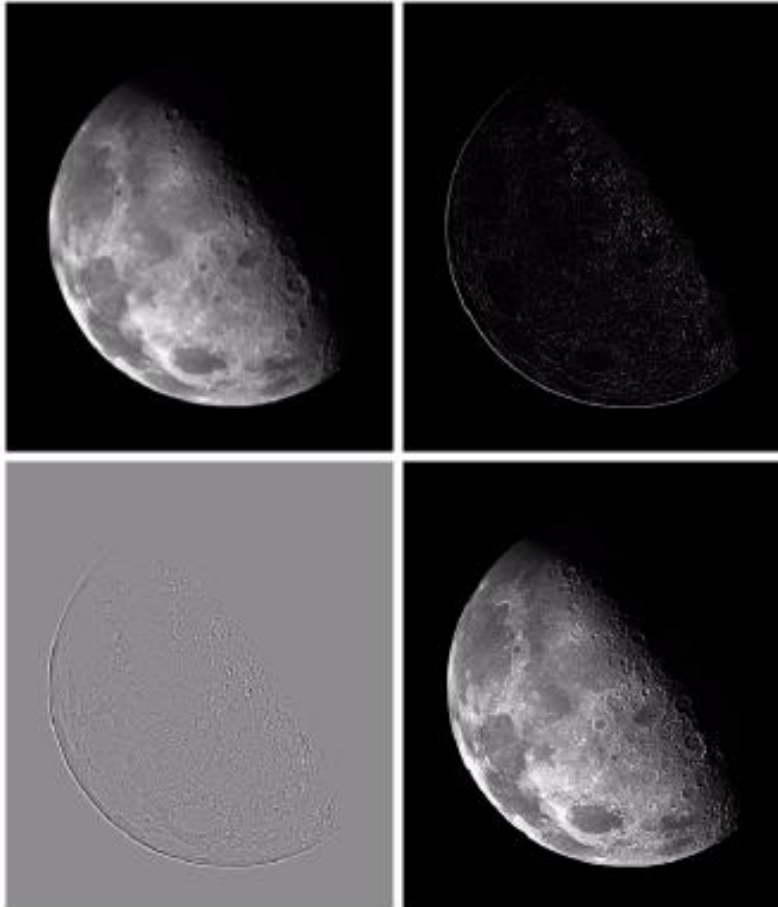
$$M_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

## Example of Sharpening

a b  
c d

**FIGURE 3.40**

(a) Image of the North Pole of the moon.  
(b) Laplacian-filtered image.  
(c) Laplacian image scaled for display purposes.  
(d) Image enhanced by using Eq. (3.7-5).  
(Original image courtesy of NASA.)



$$H_h = \frac{1}{4} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$H_s = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 16 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



# Canny edge detector

- Il metodo di Canny produce edge **connessi** che possono essere efficacemente utilizzati per le successive fasi di elaborazione.
- L'approccio prevede le seguenti fasi:
  - 1) Smoothing gaussiano dell'immagine
  - 2) Calcolo del gradiente
  - 3) Soppressione dei non-massimi in direzione ortogonale all'edge
  - 4) Selezione degli edge significativi mediante isteresi
- I risultati dipendono da alcuni parametri:
  - $\sigma$  – ampiezza della gaussiana nella prima fase
  - Dimensione del filtro nella prima fase
  - T1 e T2 – soglie per l'isteresi nell'ultima fase



$\sigma = 2$ , Filtro 5x5



$\sigma = 4$ , Filtro 9x9



$\sigma = 5$ , Filtro 15x15

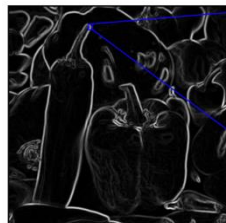
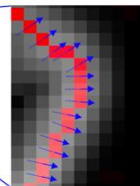
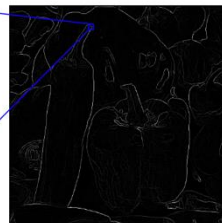


Immagine iniziale

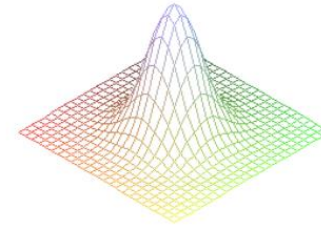


Solo i massimi locali sono mantenuti



Risultato

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



$$G_{1D}(t, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

$$G_{2D}(x, y, \sigma) = G_{1D}(x, \sigma) \cdot G_{1D}(y, \sigma)$$

## SOGLIATURA CON ISTERESI (Doppia Soglia)

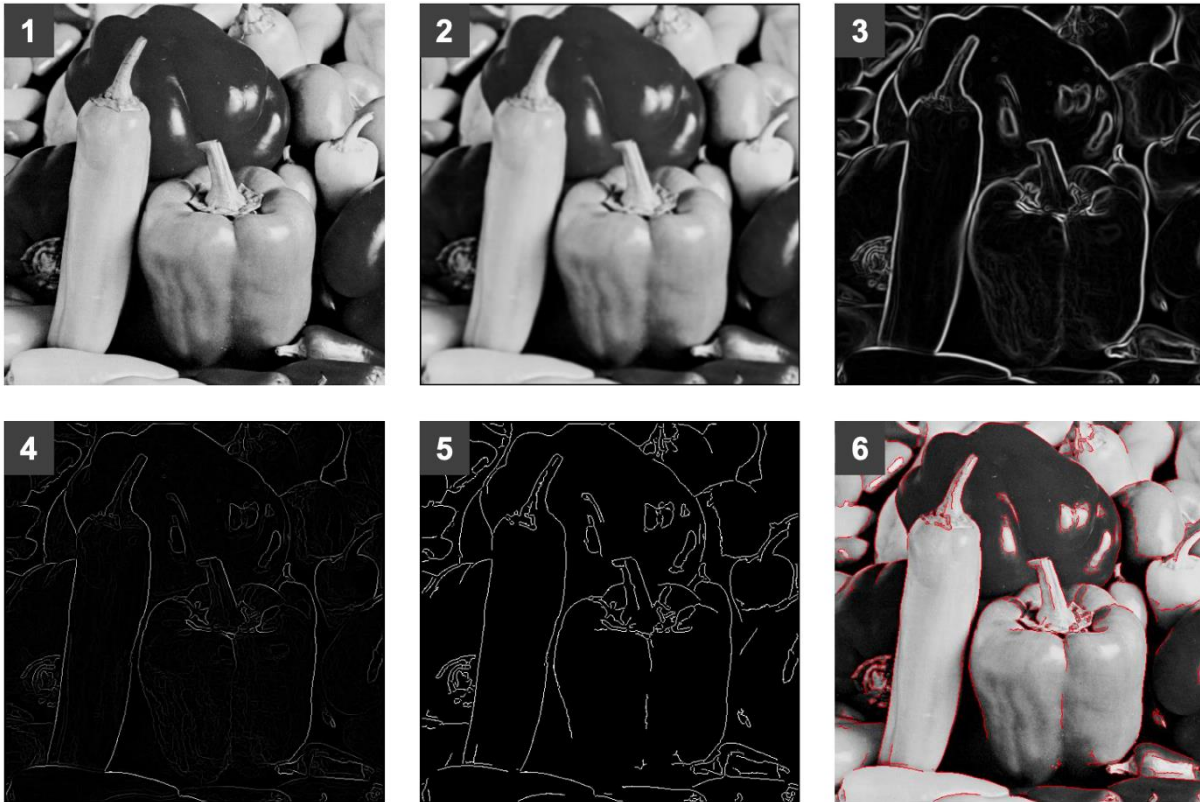
Due soglie, una bassa ed una alta, che vengono confrontate con il gradiente in ciascun punto.

- Valore del gradiente:
  - **Inferiore** alla soglia bassa, il punto è scartato;
  - **Superiore** alla soglia alta, il punto è accettato come parte di un contorno;
  - **Compreso** fra le due soglie, il punto è accettato solamente se *contiguo* ad un punto già precedentemente accettato.

**Output:** Immagine binaria dove ciascun pixel è marcato come contorno o no



# Canny edge detector – Esempi

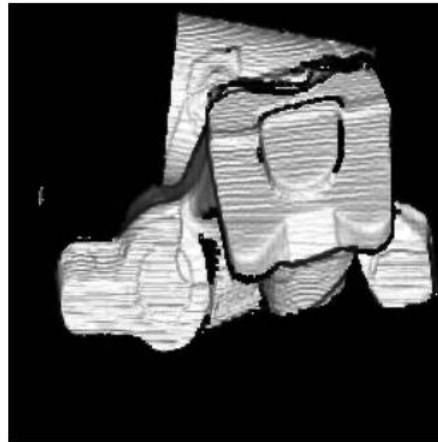


1) Immagine originale – 2) Smoothing gaussiano – 3) Modulo del gradiente  
4) Soppressione non-massimi – 5) Selezione edge – 6) Edge sovrapposti a immagine originale

# Ancora Esempi di Canny Edge Detector



$\sigma=2$  F:5x5  $T_1=0.3$   $T_2=0.15$



$\sigma=2$  F:5x5  $T_1=0.4$   $T_2=0.2$



$\sigma=1$  F:3x3  $T_1=0.3$   $T_2=0.15$

