

Elaborazione dell'Immagine

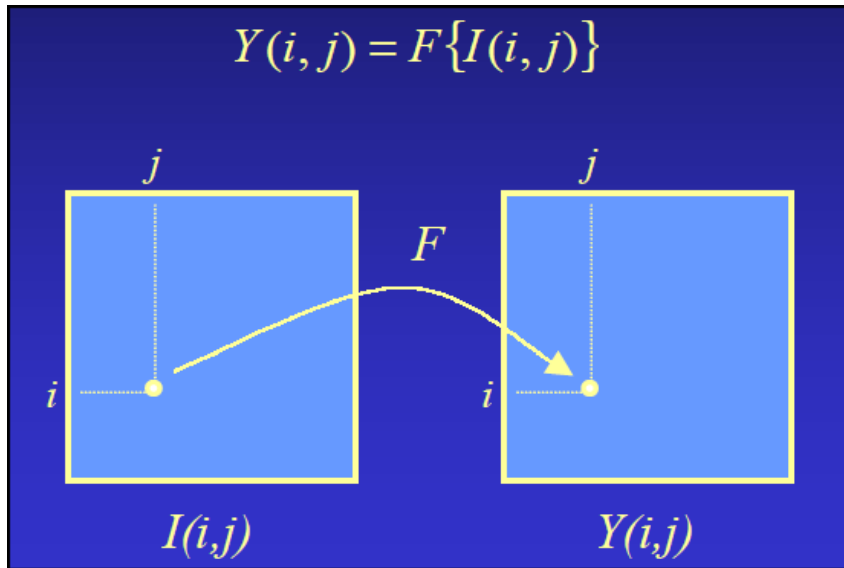
Alberto Alzati, Giovanni Naldi
Dipartimento di Matematica
“F. Enriques”

Ripasso Argomenti:

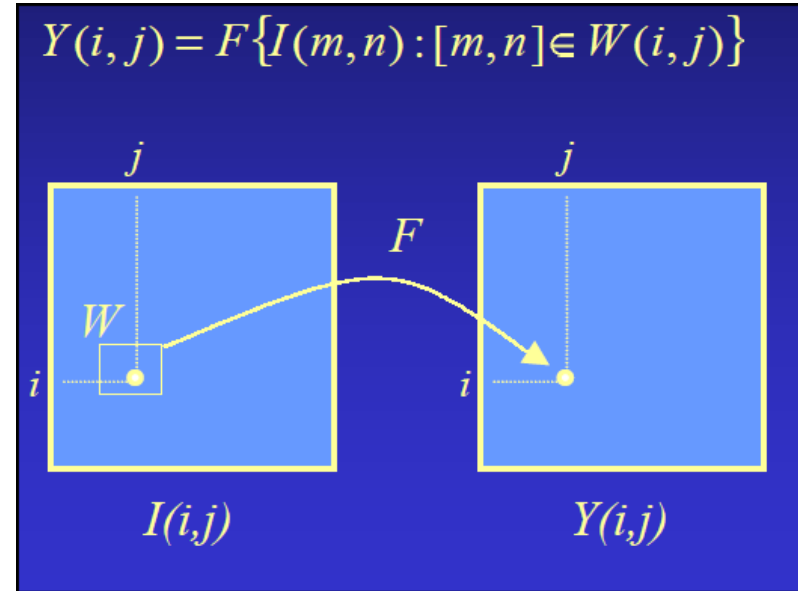
- Operatori locali
- Edge detection
- Convoluzione/Correlazione
- Operatori lineari/shift-invarianti

Operatori $F: I \rightarrow Y$, I e Y immagini

Operatori puntuali



Operatori locali



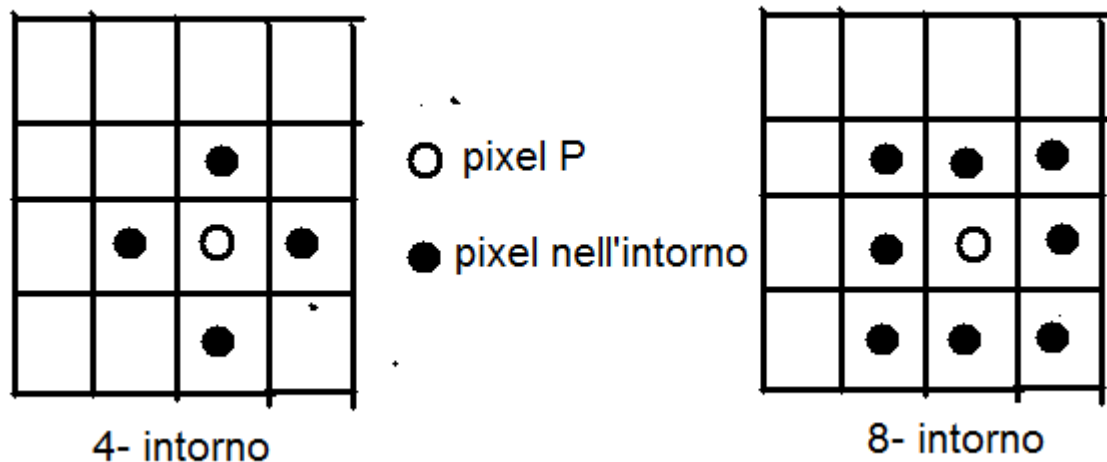
[Ci sono anche Operatori Globali]

OPERATORI LOCALI

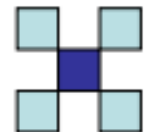
Nozione di intorno di un pixel p .

Tra i più utilizzati tipi di intorno di un pixel p di coordinate (x,y) :

- 4-intorno $N_4(P)$, pixel con coordinate $(x+1,y)$, $(x,y+1)$, $(x,y-1)$, $(x-1,y)$
- 8-intorno $N_8(P)$, pixel con coordinate come per il 4-intorno con anche i pixel diagonali, $(x-1,y-1)$, $(x-1,y+1)$, $(x+1,y-1)$, $(x+1,y+1)$.

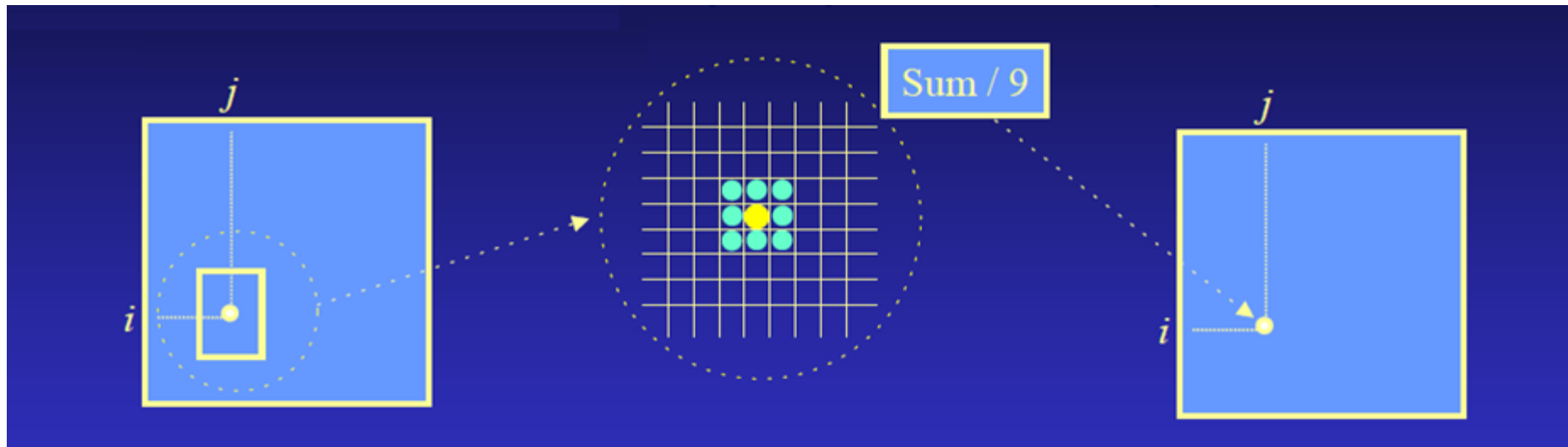


[Non sono gli unici intorni possibili], per esempio $N_D(P)$,



Esempio (operatore locale), «filtro media»

Il valore di ogni pixel viene aggiornato con la media pesata dei valori dei pixel all'interno di un opportuno intorno centrato nel pixel stesso. Caso più semplice: media aritmetica (argmin

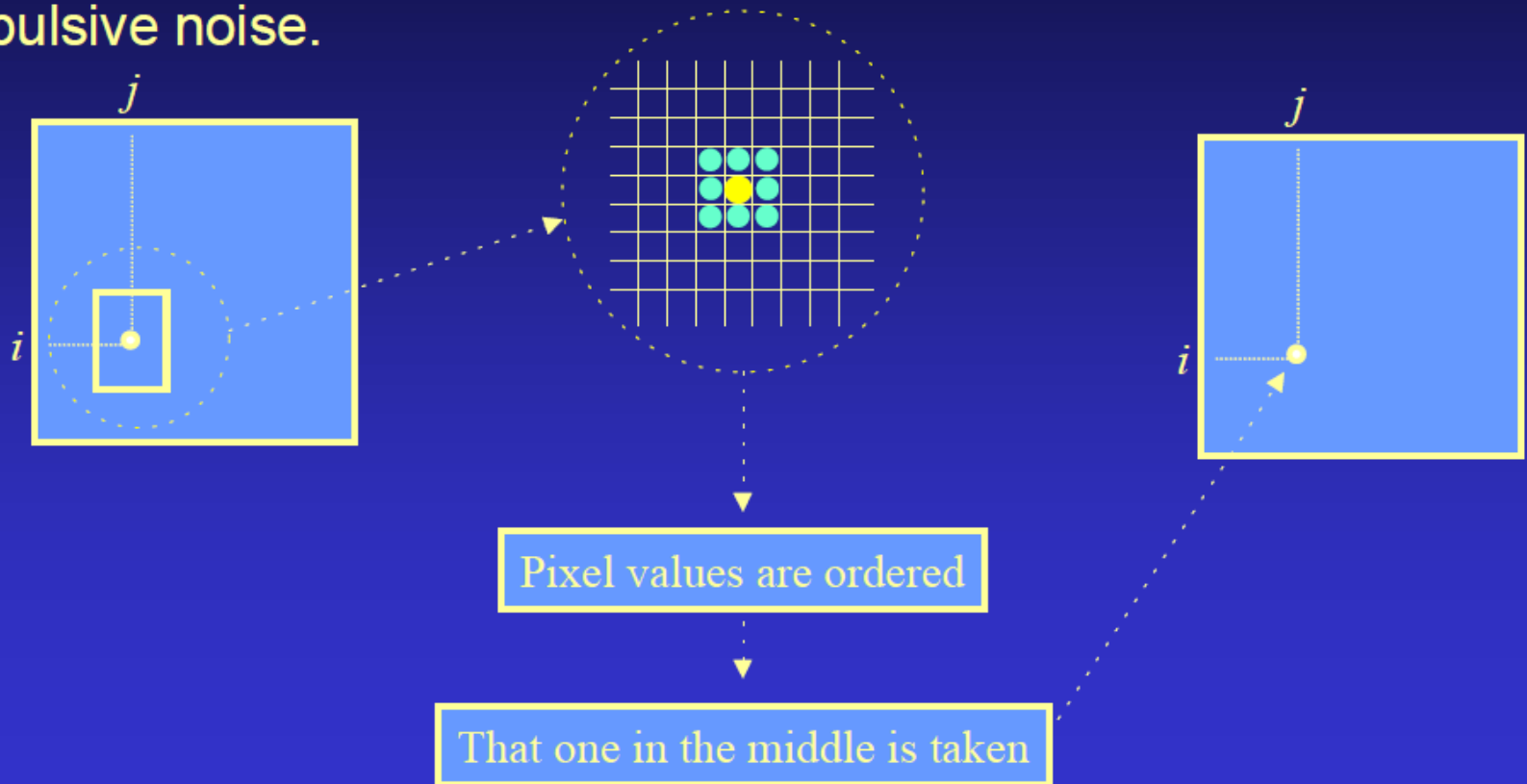


In generale avrò una media pesata con pesi $W_j \geq 0$, $\sum_j W_j > 0$ (fattore di normalizzazione) possiamo rappresentare i pesi con una matrice in corrispondenza dei pixel dell'intorno,

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \text{ media aritmetica} \quad \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix} \text{ «Gaussiana approssimata»}$$

Median Filter

Every pixel is changed with the median value computed inside a neighbourhood centred at the pixel position. Powerful for reducing impulsive noise.



Mean (Average) Filter



Added Gaussian noise with variance 100.



Mean filtered 5x5 window. The image is defocused.

Mean (Average) Filter



Added shot noise with 5% percentage.



Mean filtered 5x5 window. The image is defocused and disturbed.

Median Filter

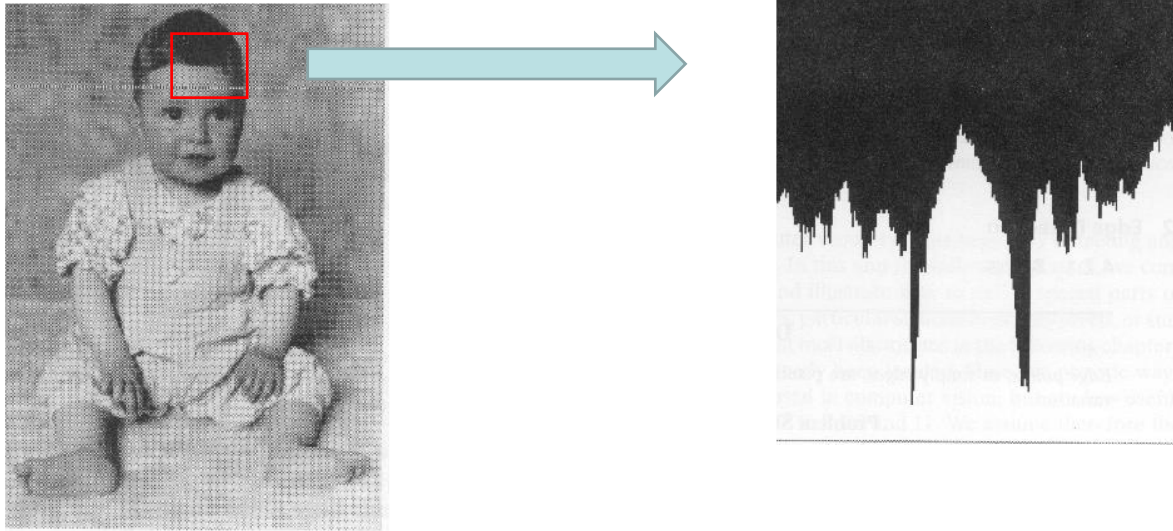


Added shot noise with 5% percentage.



Median filtered, 3x3 window. The image is slightly defocused.

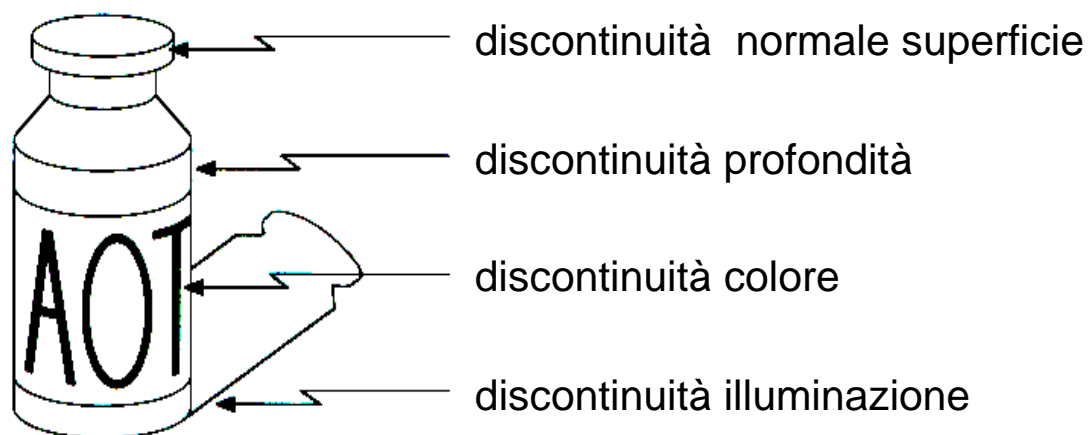
Edges are: cambiamenti locali significativi intensità dell'immagine.



Produrre una rappresentazione con line di una scena attraverso una immagine della scena.

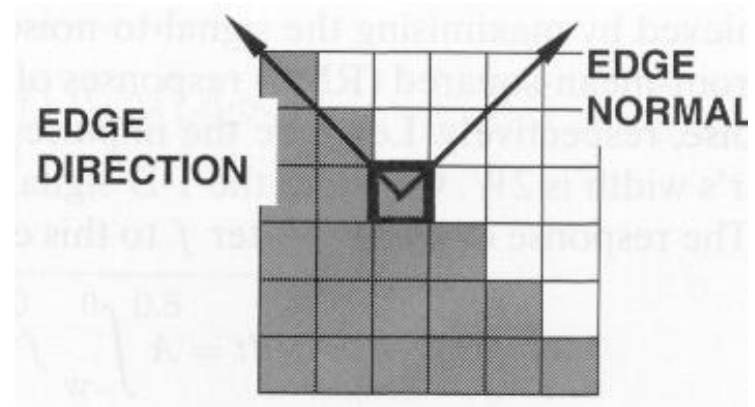
Cosa può cambiare il cambio di intensità?

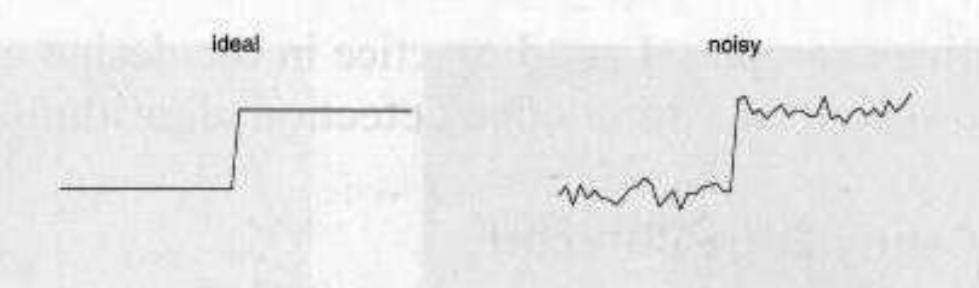
- Eventi geometrici
 - discontinuità orientazione superfici
 - discontinuità profondità
 - discontinuità colori e tessitura
- Eventi non geometrici
 - cambi illuminazione
 - ombre
 - riflessi



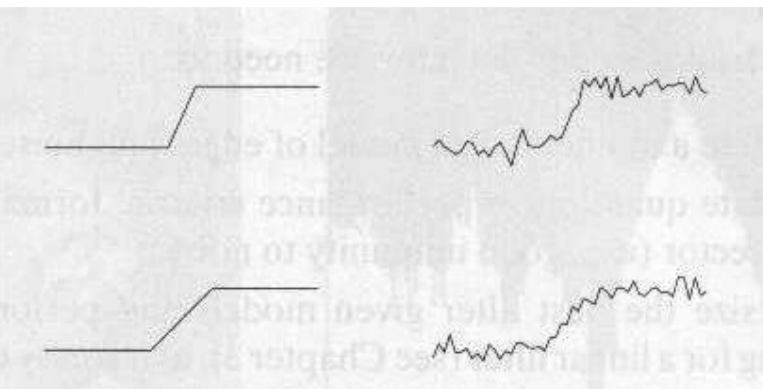
Descrizione Edge

- **direzione edge:**
perpendicolare alla direzione
del massimo cambiamento
intensità (cioè gradiente)
- **Edge strength:** relative al
contrasto locale
dell'immagine lungo la
direzione normale
- **Posizione edge:** posizione
nell'immagine dell'edge

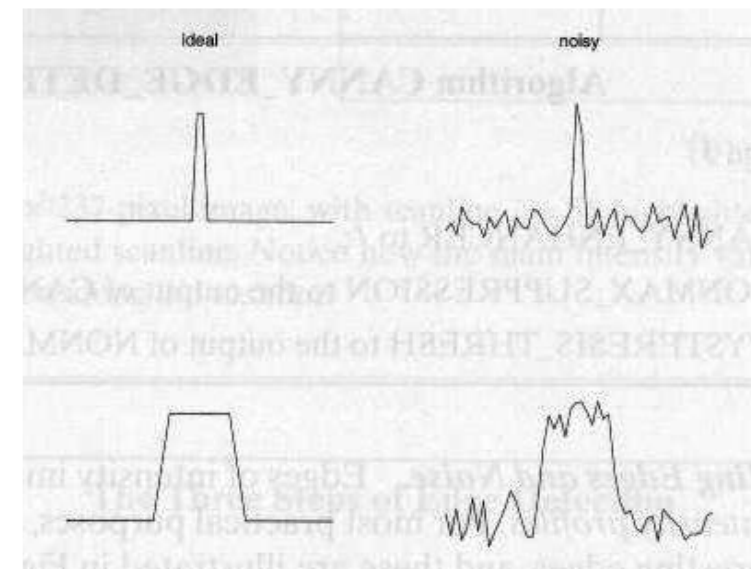




Step edge

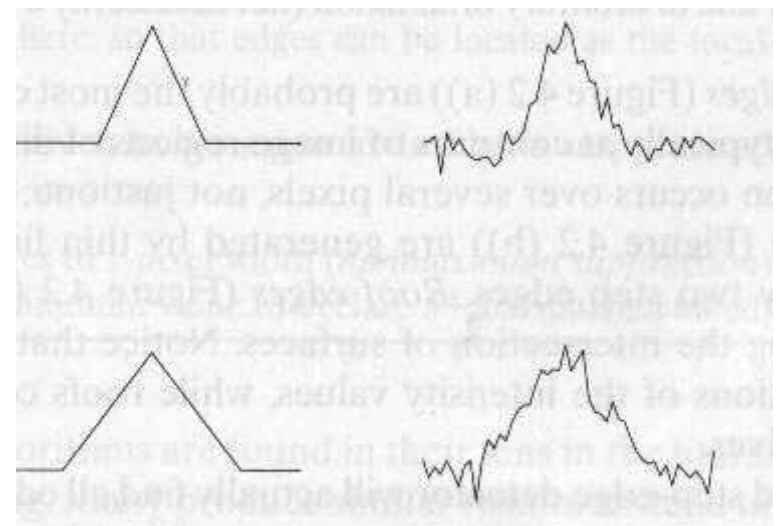


Ramp edge



Ridge edge

Roof edge



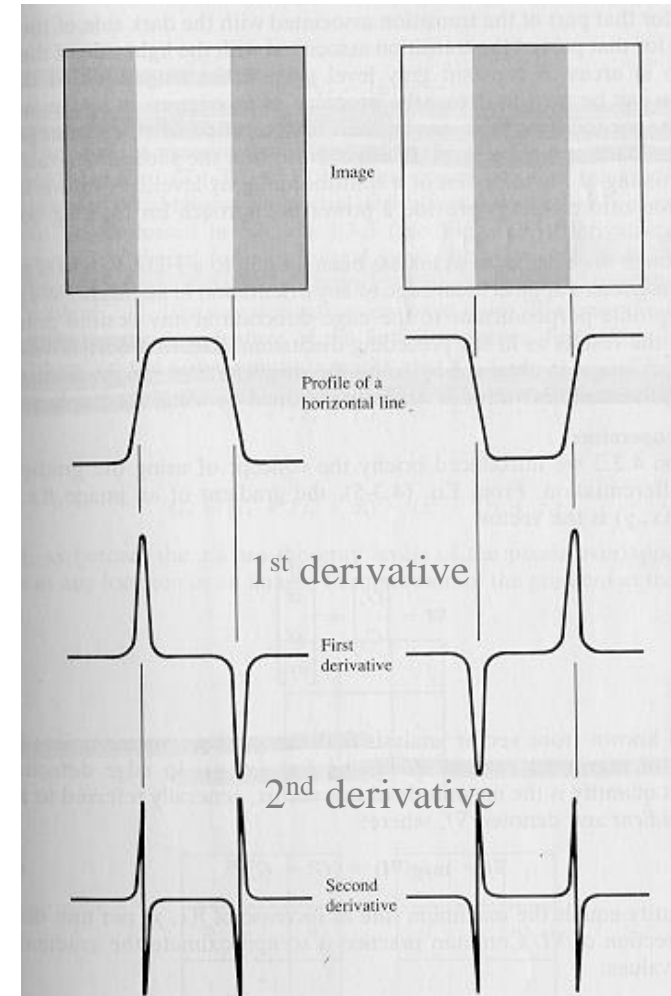
Esempio di come determinare i punti di bordo con le derivate

I pixel che stanno su un bordo (edge) sono individuati dai

massimi locali o i minimi locali della derivativa prima

Oppure

gli zero-crossings della derivata seconda



Derivate ed immagini

- Come calcolare la derivata di una immagine digitale?
 - **Opzione 1:** ricostruire una immagine continua, $f(x,y)$, poi calcolarne le derivate.
 - **Option 2:** Calcolare le derivate discrete (per esempio, differenze finite) dai valori dei pixel

Edge Detection con derivata prima

Segnale 1D

(non centrata in x)

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \approx f(x+1) - f(x) \quad (h=1) \quad \Rightarrow \quad \text{mask:} \quad [-1 \quad 1]$$

$$\text{mask } M = [-1, 0, 1] \quad (\text{centrata in } x)$$

(upward) step edge

S_1			12	12	12	12	12	24	24	24	24	24
S_1	\otimes	M	0	0	0	0	12	12	0	0	0	0

(downward) step edge

S_2			24	24	24	24	24	12	12	12	12	12
S_2	\otimes	M	0	0	0	0	-12	-12	0	0	0	0

ramp edge

S_3			12	12	12	12	15	18	21	24	24	24
S_3	\otimes	M	0	0	0	3	6	6	6	3	0	0

roof edge

S_4			12	12	12	12	24	12	12	12	12	12
S_4	\otimes	M	0	0	0	12	0	-12	0	0	0	0

Edge Detection usando derivata seconda

Segnale 1D

$$f''(x) = \lim_{h \rightarrow 0} \frac{f'(x+h) - f'(x)}{h} \approx f'(x+1) - f'(x) =$$

(centrato in $x+1$)

$$f(x+2) - 2f(x+1) + f(x) \quad (h=1)$$

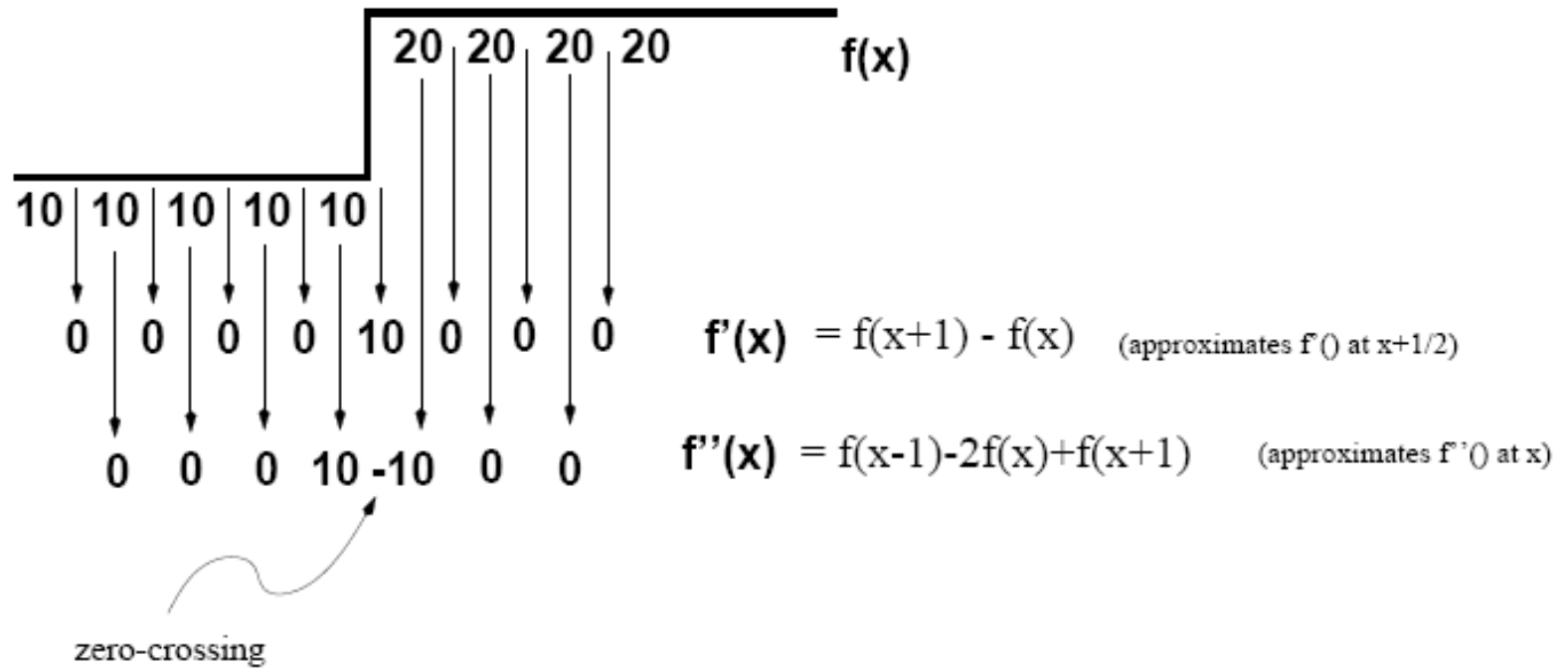
Sostituire $(x+1)$ con x (cioè centrato in x):

$$f''(x) \approx f(x+1) - 2f(x) + f(x-1)$$



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

Esempio.



(upward) step edge

S_1			12	12	12	12	12	24	24	24	24	24
S_1	\otimes	M	0	0	0	0	-12	12	0	0	0	0

(downward) step edge

S_2			24	24	24	24	24	12	12	12	12	12
S_2	\otimes	M	0	0	0	0	12	-12	0	0	0	0

roof edge

S_4			12	12	12	12	24	12	12	12	12	12
S_4	\otimes	M	0	0	0	-12	24	-12	0	0	0	0

ramp edge

S_3			12	12	12	12	15	18	21	24	24	24
S_3	\otimes	M	0	0	0	-3	0	0	0	3	0	0

Edge Detection con derivata seconda

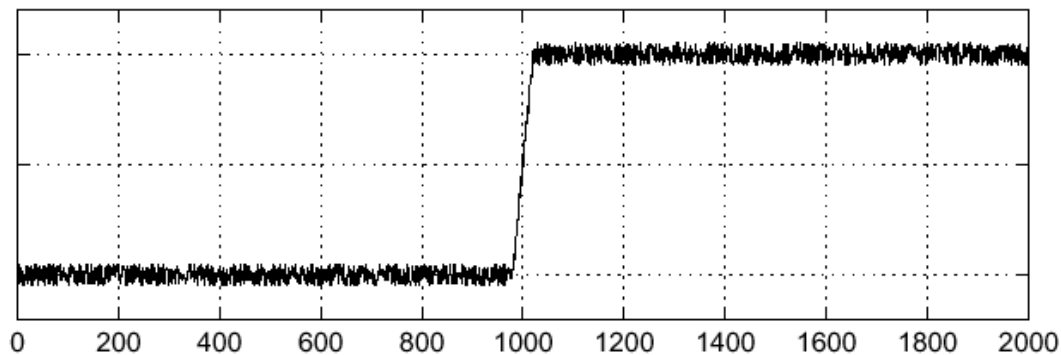
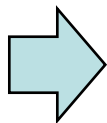
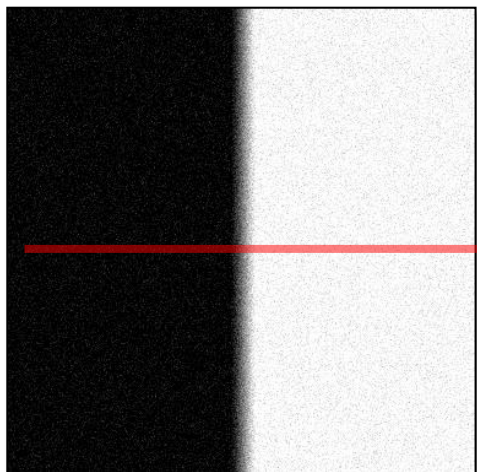
- Quattro casi di zero-crossings:

$$\{+,-\}, \{+,0,-\}, \{-,+\}, \{-,0,+\}$$

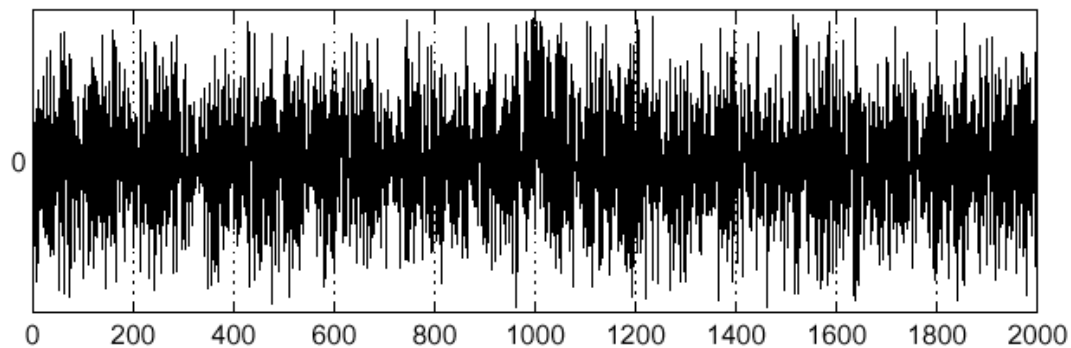
- **Pendenza** del zero-crossing $\{a, -b\}$ è $|a+b|$.
- Per determinare “strong” zero-crossing, si utilizza una soglia sulla pendenza.

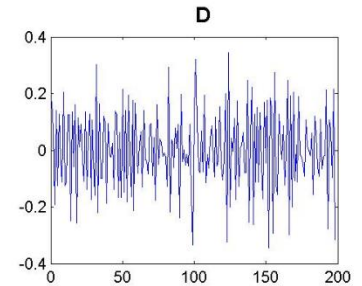
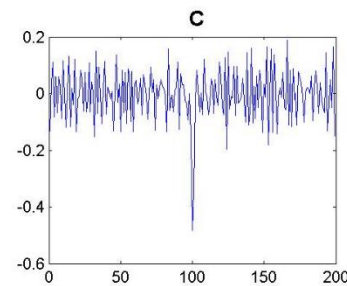
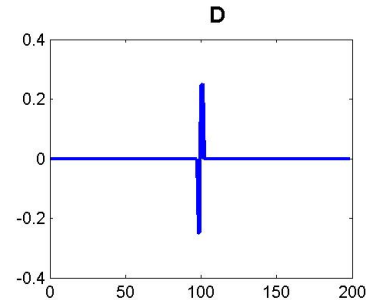
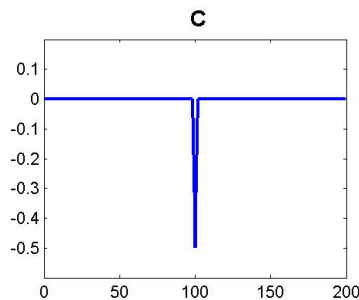
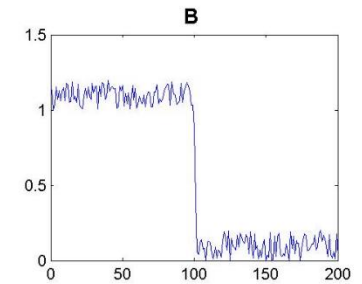
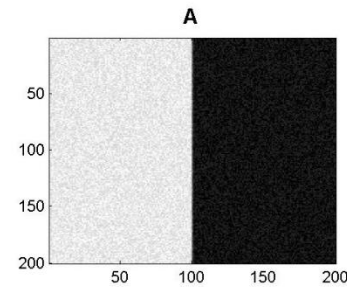
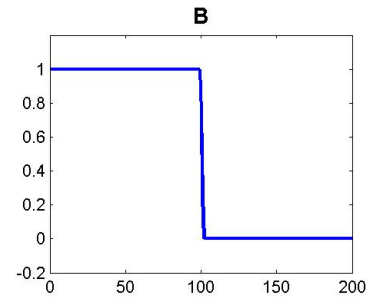
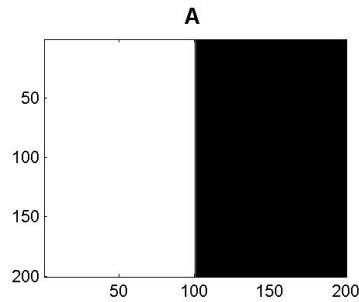
Effetto rumore sulla derivata

$$f(x)$$



$$\frac{d}{dx}f(x)$$





$$F_x \approx \frac{\partial f}{\partial x}, \quad F_y \approx \frac{\partial f}{\partial y}, \quad \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Derivata “discreta”, rapporti incrementali: occorre stimare la variazione.

$$F_x(i, j) = F(i+1, j) - F(i, j), \quad F_y(i, j) = F(i, j+1) - F(i, j)$$

Approssimazione gradiente

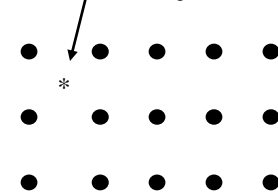
- Approssimazione derivate parziali

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

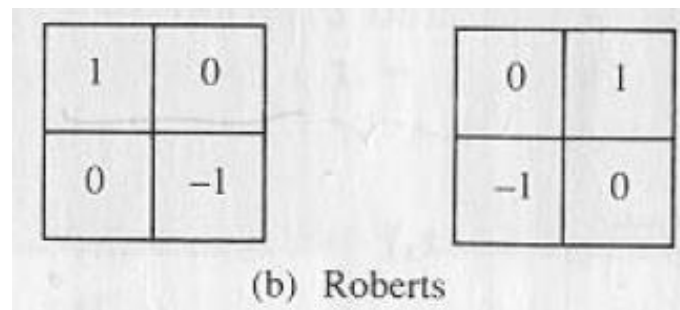
$$\frac{\partial f}{\partial y}(x, y) = f(x, y) - f(x, y + 1),$$

Buona approssimazione

$(x+1/2, y+1/2)$



- $\frac{\partial f}{\partial x}$ e $\frac{\partial f}{\partial y}$ possono essere implementati attraverso le maschere:



Un approccio più generale

- Consideriamo i pixels nell'8-intorno del pixel (i, j) :

$$\begin{array}{ccccc} & a_0 & a_1 & a_2 & \\ 3 \times 3 \text{ neighborhood:} & a_7 & [i, j] & a_3 & \\ & a_6 & a_5 & a_4 & \end{array}$$

- Le derivate $\frac{\partial f}{\partial x}$ $\frac{\partial f}{\partial y}$ possono essere approssimate come:

$$\begin{aligned} M_x &= (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \\ M_y &= (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2) \end{aligned}$$

- La costante c enfatizza i pixels nel 4-intorno.

Prewitt Operator

con $c = 1$, abbiamo l'operatore Prewitt:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

M_x and M_y are approximations at (i, j) .

Sobel Operator

con $c = 2$, abbiamo l'operatore di Sobel:

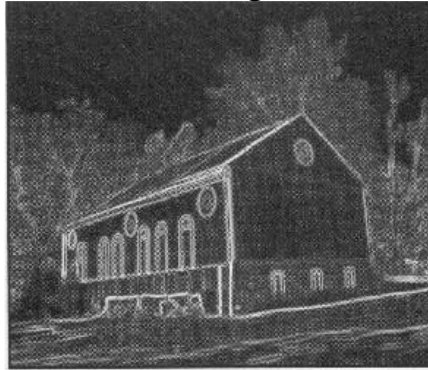
$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

M_x and M_y are approximations at (i, j) .

(la soglia è importante)



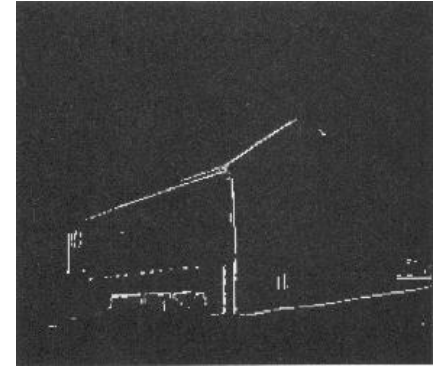
Norma del gradiente



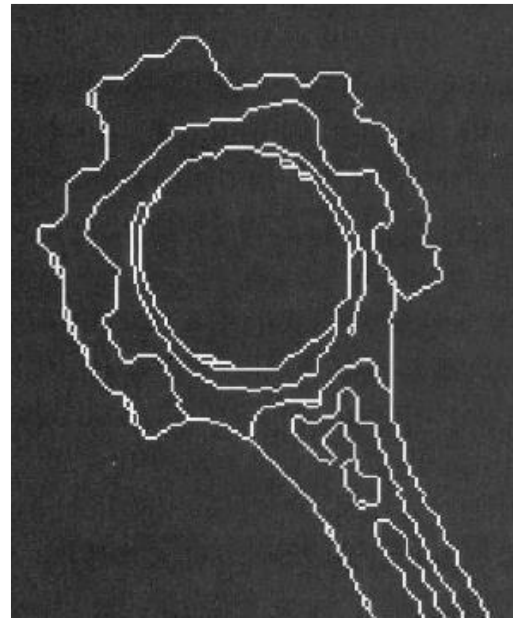
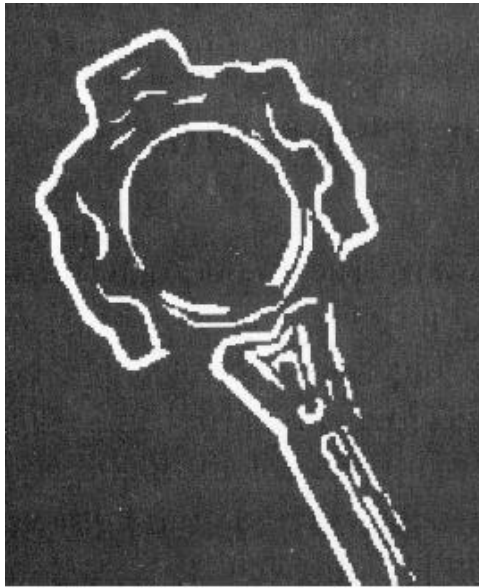
soglia bassa



soglia alta



(rendere gli Edge più sottili e collegare i punti di edge)



Lemma di rappresentazione per operatori lineari (invarianti per traslazione). Caso 1-D (non è limitativo), spazio dei segnali (supporto compatto) reali (l'estensione al caso complesso è immediata)

$$\mathcal{S} = \{ x : Z \rightarrow R, \text{ t.c. } \exists m \geq 0 \text{ tale che } |x_i| = 0 \text{ } |i| \geq m \}$$

Notazioni

X_i componente i -esima del segnale X ;

$X^{(\tau)}$ con τ intero, è il segnale traslato definito come

$$X^{(\tau)}_i = X_{i-\tau}$$

$\delta^{(k)}$, k intero, successione delta definita come

$$\begin{aligned} \delta^{(k)}_i &= 0 & \text{se } k \neq i, \\ &= 1 & \text{se } k = i \end{aligned}$$

Lemma (**di rappresentazione**)

Sia $T : S \rightarrow S$ un operatore

- *Lineare*
- *Invariante per traslazioni*

Allora, posto $h=T(\delta^{(0)})$, per ogni segnale X in S ,

$$T(X)_i = \sum_k X_k h_{i-k}$$

Notazione: $T(X)=X*h$

prodotto di convoluzione discreto

h è detta anche mask in questo ambito

Dimostrazione. (vedi lucidi)

Caso 2D, considero spazio segnali a supporto compatto(+), sia A un segnale 2D, $A_{i,j}$ la componente di posizione (i,j) , T un operatore lineare, invariante per traslazioni, $h = T(\delta^{(0,0)})$, con $\delta^{(0,0)}$ segnale uguale ad 1 solo quando per indici $i=0, j=0$, allora

$$T(A)_{i,j} = \sum_l \sum_k A_{l,k} h_{i-l, j-k}$$

Nota. Quando è chiaro il contesto indicheremo con $A(l,k)$ la componente $A_{l,k}$ di un segnale 2D (analogamente per il caso 1D).

Nota. In laboratorio MATLAB, a causa delle convenzioni di rappresentazione delle immagini, usualmente $A(l,k)$ indica la componente di riga k e colonna l : primo indice “conta” le colonne, il secondo le righe.

(+) in questo modo si evitano questioni di convergenza di serie

Convoluzione Spaziale bidimensionale f (input) $\rightarrow g$ (output)

$f(i,j)$ valore pixel (i,j) dell'immagine f , $g(i,j)$ valore pixel (i,j) dell'immagine g .

Solitamente, per convenienza, l'immagine di input e di output sono scelte con identiche dimensioni, mentre la risposta impulsiva bidimensionale normalmente ha dimensioni limitate, a partire da finestre $(2n+1) \times (2n+1)$

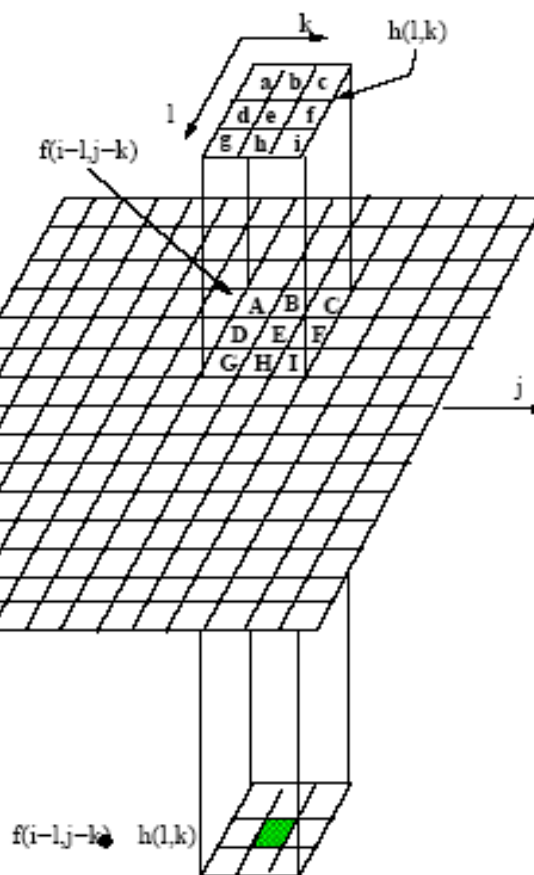
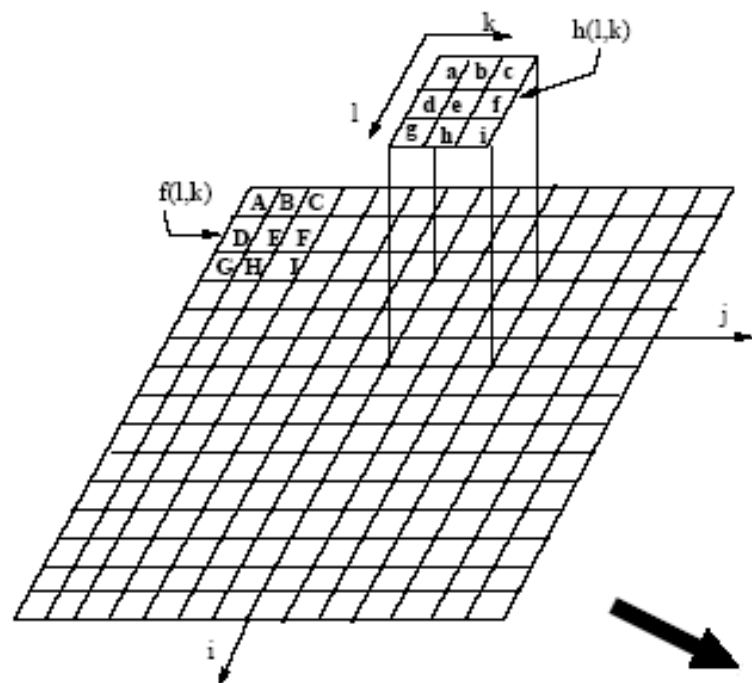
Sempre per convenienza, il risultato della convoluzione nell'immagine di output $g(i,j)$, si fa corrispondere con il pixel centrale della finestra quadrata in cui è definita la risposta impulsiva h localizzata nel pixel (i,j)

Il valore della convoluzione in corrispondenza del pixel (i,j) nell'immagine di output é ottenuta dalla somma dei «prodotti pixel per pixel» tra la funzione di input $f(l,k)$ e la risposta impulsiva $h(i-l,j-k)$, derivata da $h(l,k)$ eseguendo prima una rotazione di 180° rispetto all'origine ottenendo $h(0-l,0-k)$ e successivamente una traslazione dall'origine alla posizione (i,j)

Convoluzione Spaziale bidimensionale

$$g(i, j) = \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} f(l, k) \cdot h(i-l, j-k)$$

considerando l'immagine di input $f(i, j)$ e l'immagine di output $g(i, j)$ di dimensioni $N \times N$, mentre la risposta impulsiva h è definita con una finestra quadrata di dimensioni limitate $L \times L$



La convoluzione (maschera h)

- La convoluzione nel discreto avviene su un supporto limitato nello spazio
- Può essere considerata una finestra che si muove sull'immagine

1	2	3	3	2	3
3	2	5	2	7	6
1	3	6	7	8	8
1	2	8	9	6	7
2	3	7	7	6	8
3	3	8	9	8	8

	26	33	43	46	
	31	44	58	60	
	33	52	64	66	
	37	50	68	68	

Convoluzione Spaziale bidimensionale

Per ragioni implementative è conveniente utilizzare operatori di convoluzione con equazioni della forma

$$g(i, j) = \sum_{l=-r}^r \sum_{k=-r}^r f(i-l, j-k) \cdot h(l, k)$$

con la finestra relativa alla risposta impulsiva $h(l, k)$

Indici l e k riferiti rispetto al pixel centrale $h(0,0)$.

Le dimensioni $L \times L$ della finestra hanno valore dispari e con $r = (L-1)/2$

Esempio di Convoluzione Spaziale bidimensionale

Per semplicità consideriamo un esempio di convoluzione spaziale dove la risposta impulsiva é rappresentata da una finestra 3×3 con i pixel contenenti valori discreti di una funzione Gaussiana

In questo caso il pixel centrale $h(0,0)$ coincide con il picco della gaussiana che essendo simmetrica rispetto all'origine l'eventuale rotazione della funzione di input non avrebbe nessun effetto

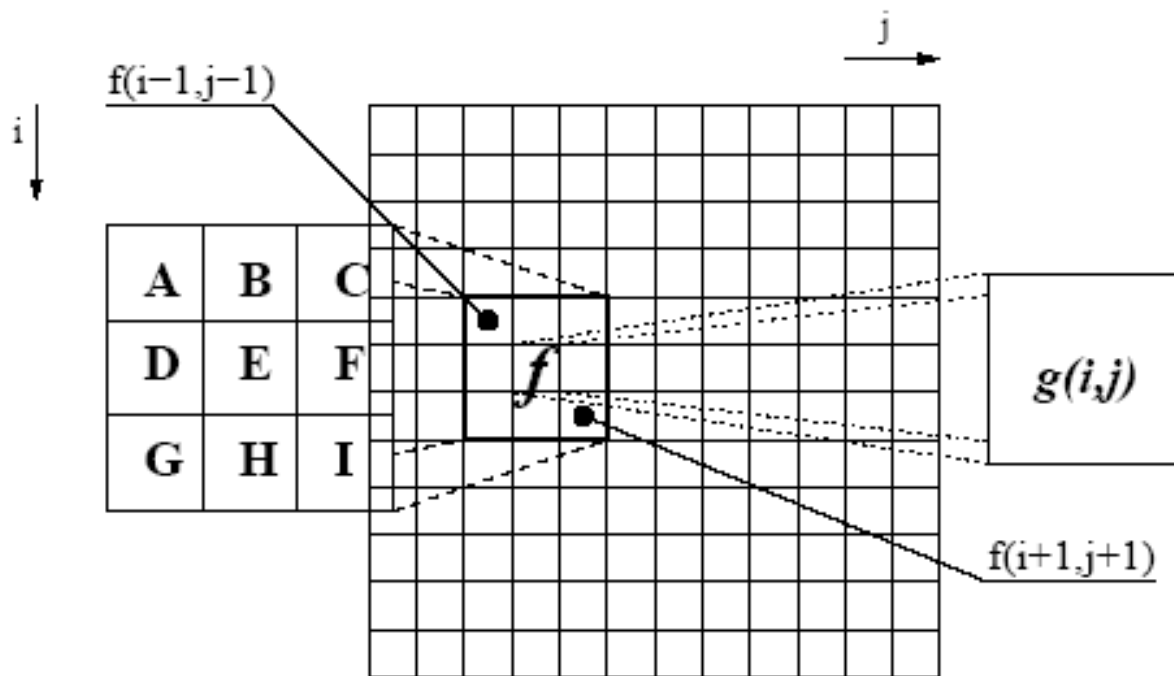
Esempio di Convoluzione Spaziale bidimensionale

Il valore della convoluzione $g(i,j)$, applicata al pixel (i,j) è ottenuta posizionando il pixel centrale della maschera di convoluzione in (i,j) , e la seguente espressione viene eseguita:

$$g(i,j) = A \cdot f(i-1,j-1) + B \cdot f(i-1,j) + C \cdot f(i-1,j+1) + \\ D \cdot f(i,j-1) + E \cdot f(i,j) + F \cdot f(i,j+1) + \\ G \cdot f(i+1,j-1) + H \cdot f(i+1,j) + I \cdot f(i+1,j+1)$$

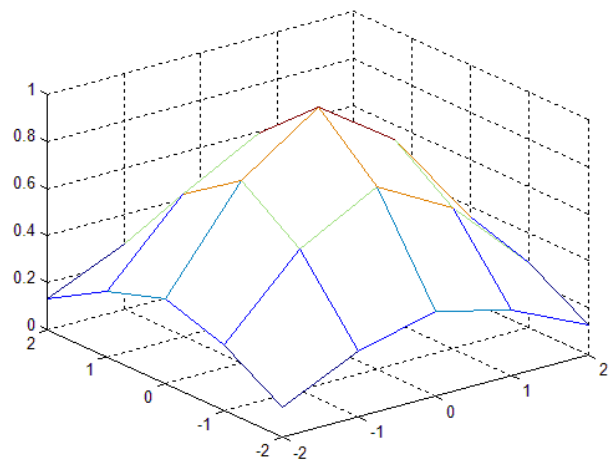
dove $f(i,j)$ è l'immagine di input, le costanti da "A" ad "I" indicano i coefficienti della maschera ossia i valori discreti risultati dal campionamento della risposta impulsiva $h(i,j)$ dell'operatore stesso, e $g(i,j)$ è l'immagine ottenuta applicando il processo di convoluzione

Attenzione: in questo caso la maschera è Simmetrica, altrimenti **VA RUOTATA**



Convoluzione Spaziale bidimensionale (Gaussiana)

$$h(l,k) = e^{\frac{-(l^2+k^2)}{2\sigma^2}}$$



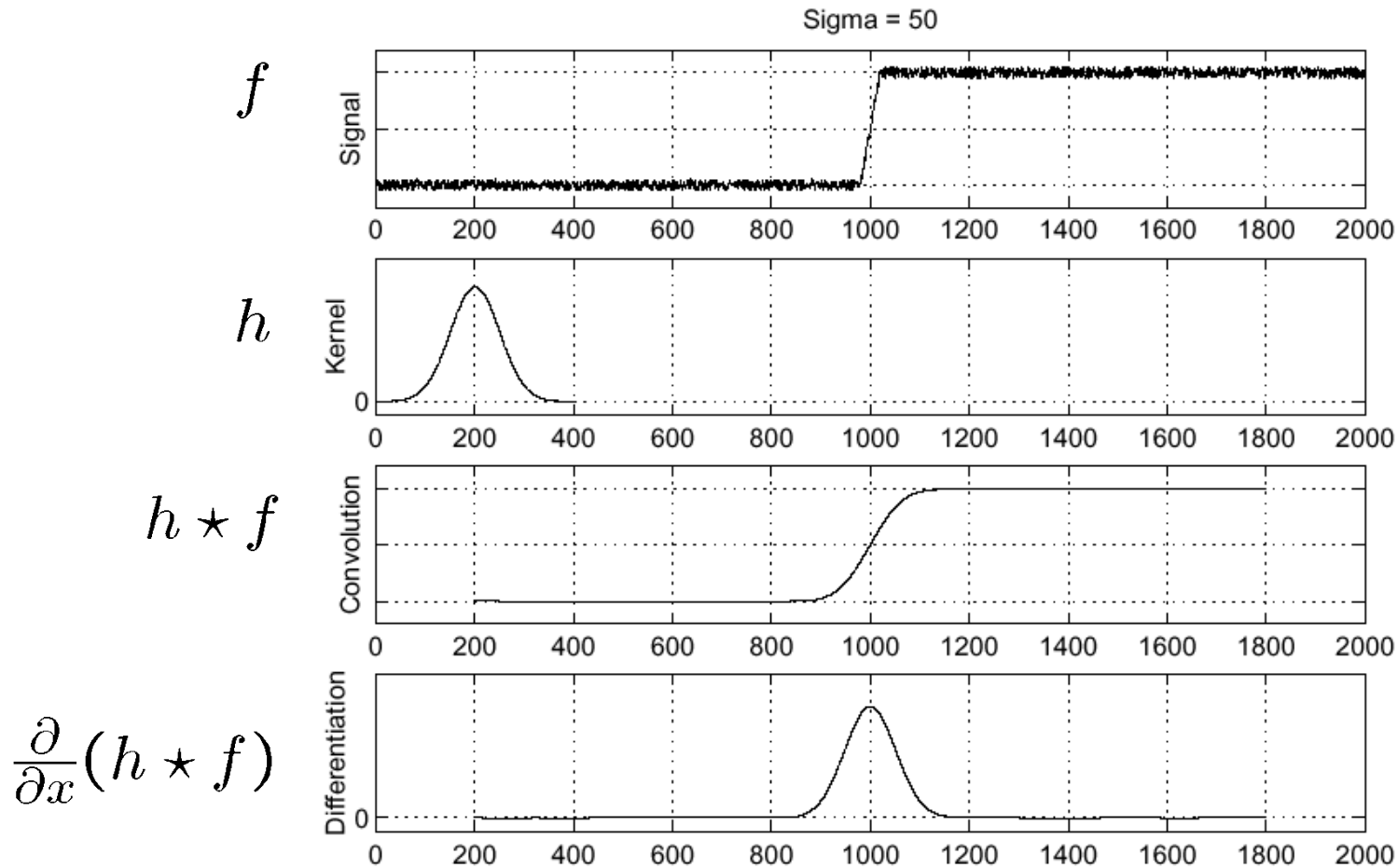
$h_0=1$
 $h_1=0.78$
 $h_2=0.61$
 $h_3=0.37$
 $h_4=0.28$
 $h_5=0.13$

$\sigma = 1.4$
 $L = 5$
 $r = 2$

-2	h_5	h_4	h_3	h_4	h_5
-1	h_4	h_2	h_1	h_2	h_4
0	h_3	h_1	h_0	h_1	h_3
1	h_4	h_2	h_1	h_2	h_4
2	h_5	h_4	h_3	h_4	h_5
	-2	-1	0	1	2

Edge detection

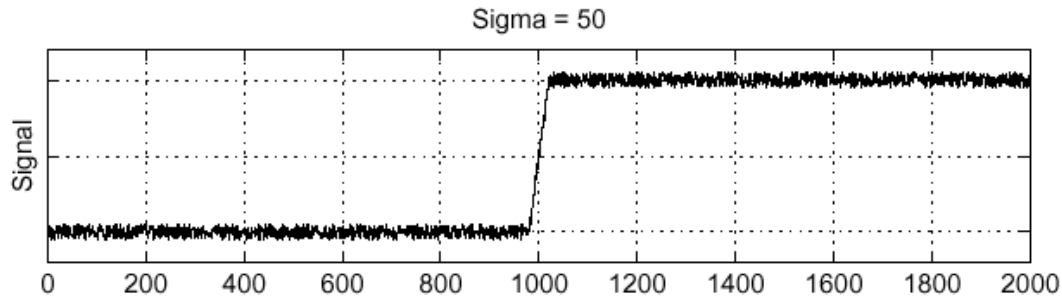
Effetto regolarizzazione



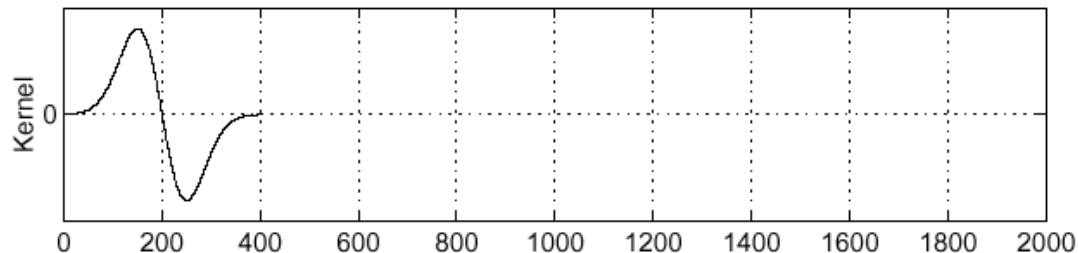
Risparmiamo operazioni dalle proprietà della convoluzione

$$\frac{\partial}{\partial x}(h \star f) = \left(\frac{\partial}{\partial x}h\right) \star f \quad (\text{calcolo un solo prodotto})$$

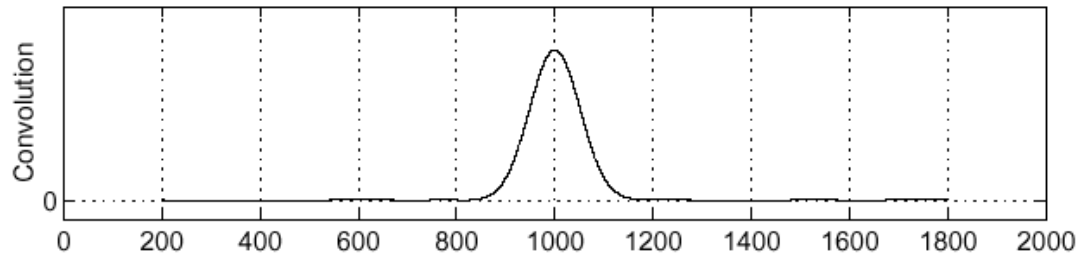
f



$\frac{\partial}{\partial x}h$



$\left(\frac{\partial}{\partial x}h\right) \star f$



Edge Detection Steps Using Gradient

(1) Smooth the input image ($\hat{f}(x, y) = f(x, y) * G(x, y)$)

(2) $\hat{f}_x = \hat{f}(x, y) * M_x(x, y) \longrightarrow$

(3) $\hat{f}_y = \hat{f}(x, y) * M_y(x, y) \longrightarrow$

(4) $magn(x, y) = |\hat{f}_x| + |\hat{f}_y|$ (i.e., `sqrt` is costly!)

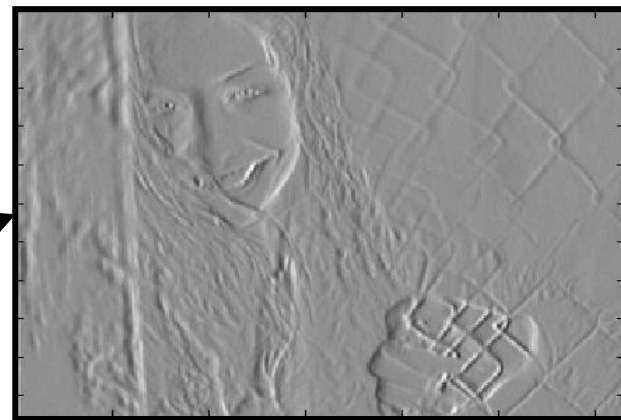
(5) $dir(x, y) = \tan^{-1}(\hat{f}_y / \hat{f}_x)$

(6) If $magn(x, y) > T$, then possible edge point

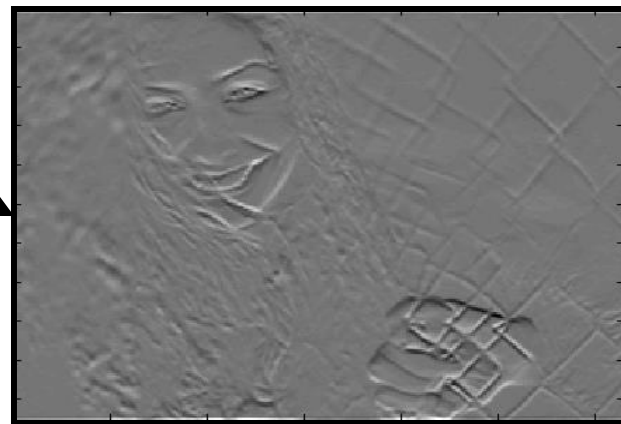
Esempio



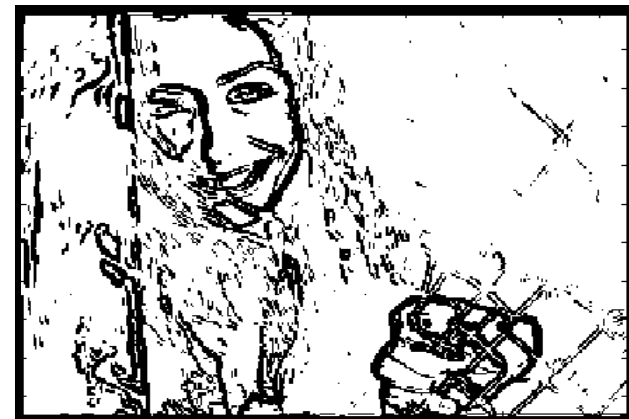
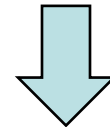
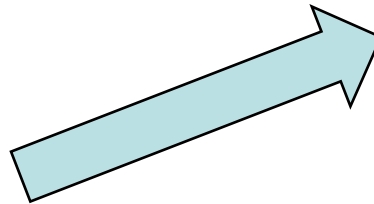
$$\frac{\partial}{\partial x} I$$



$$\frac{\partial}{\partial y} I$$



$$\|\nabla I\|_2 = \sqrt{\left(\frac{\partial}{\partial x} I\right)^2 + \left(\frac{\partial}{\partial y} I\right)^2}$$

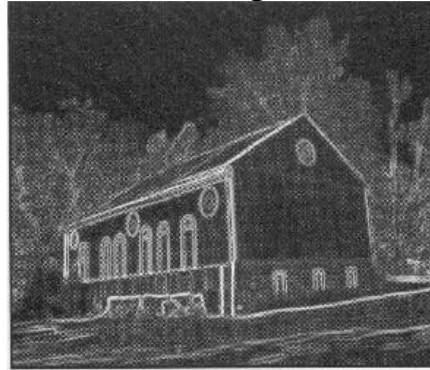


$$\|\nabla I\|_2 \geq \textit{soglia} (=100)$$

(la soglia è importante)



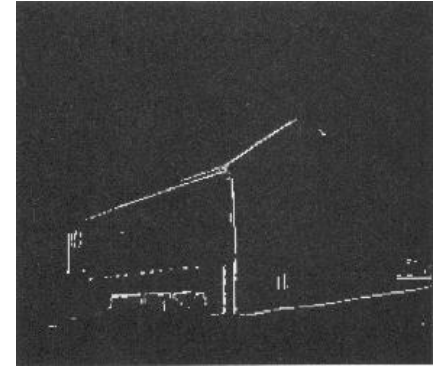
Norma del gradiente



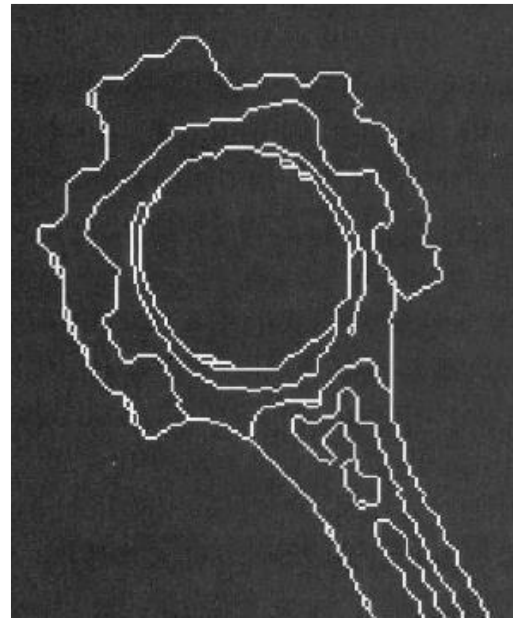
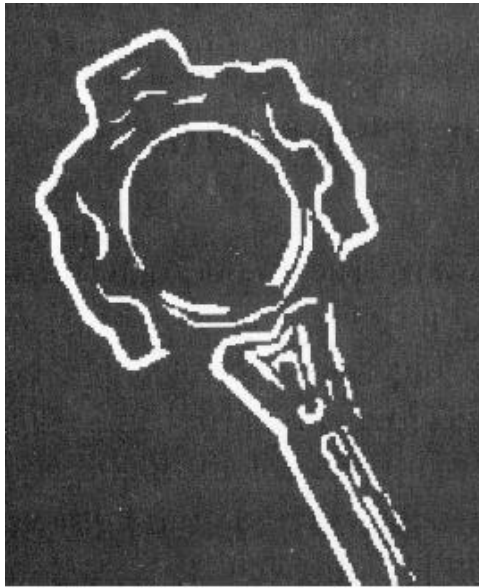
soglia bassa



soglia alta

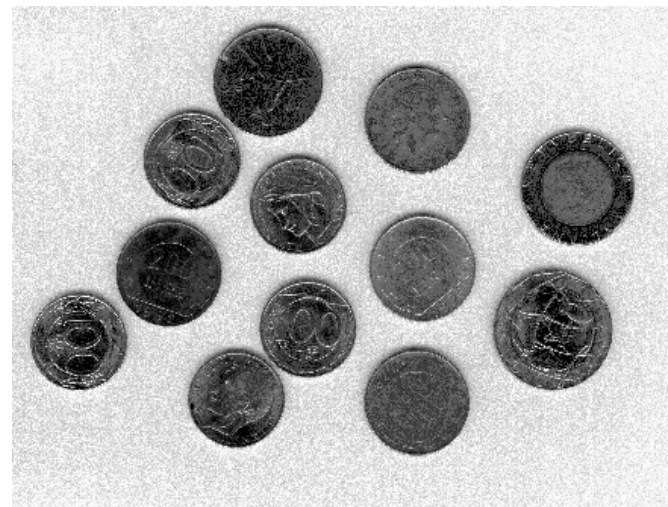


(rendere gli Edge più sottili e collegare i punti di edge)



Riconoscimento di forme

- Gli oggetti sono spesso *occlusi*
- Le acquisizioni sono rumorose



Trasformata di Hough

- Una delle soluzioni più efficaci per il riconoscimento è la **trasformata di Hough (HT)**
 - La HT nella sua formulazione originaria (primi anni 60) permette di individuare forme descritte in forma analitica (prima è stata introdotta per le rette, poi per i cerchi, per le parabole, ecc.)
 - La HT è parzialmente insensibile alle occlusioni (e al rumore)
 - La HT trasforma il problema della ricerca di una curva (spesso) nella semplice ricerca di massimi

Trasformata di Hough

- In generale una curva piana è definita in forma analitica tramite un insieme (limitato se è semplice) di parametri.
- Una equazione lega i parametri alle coordinate cartesiane

$$f((x,y), (a_1, a_2, \dots, a_n))=0 \quad (x,y) \text{ è un punto della curva}$$

- (a_1, a_2, \dots, a_n) è una n-upla di valori che individuano un punto nello spazio dei parametri
- Un punto nello spazio dei parametri individua in modo univoco una curva analitica

HT: proprietà

- In generale ad ogni punto P' nello spazio immagine (SI) corrisponde una ipersuperficie nello spazio dei parametri (SP) in cui ogni punto rappresenta una curva passante per P' .
- n punti nello SI appartenenti ad una stessa curva generano n superfici che si intersecano in uno stesso punto in SP, corrispondente ai parametri della curva.
- Più ricca è l'evidenza nello spazio immagine (es. oltre al passaggio, anche la direzione della curva, oppure la curvatura nel punto), più limitato è l'insieme delle compatibilità in SP.

HT: esempio rette

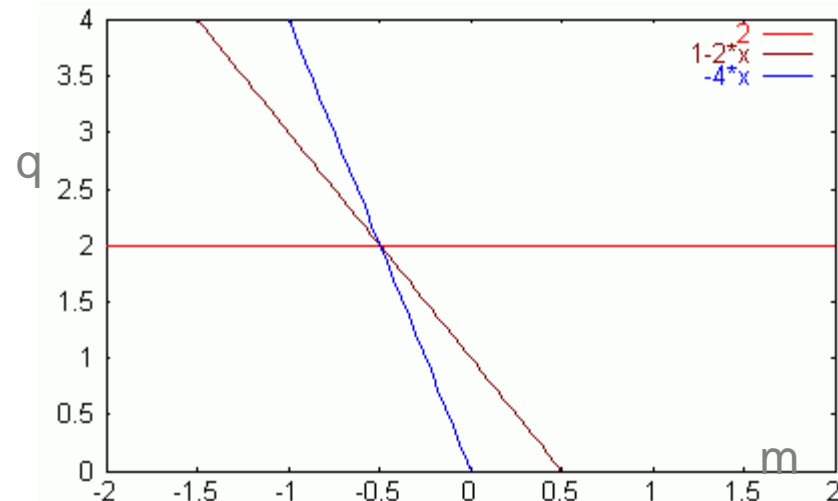
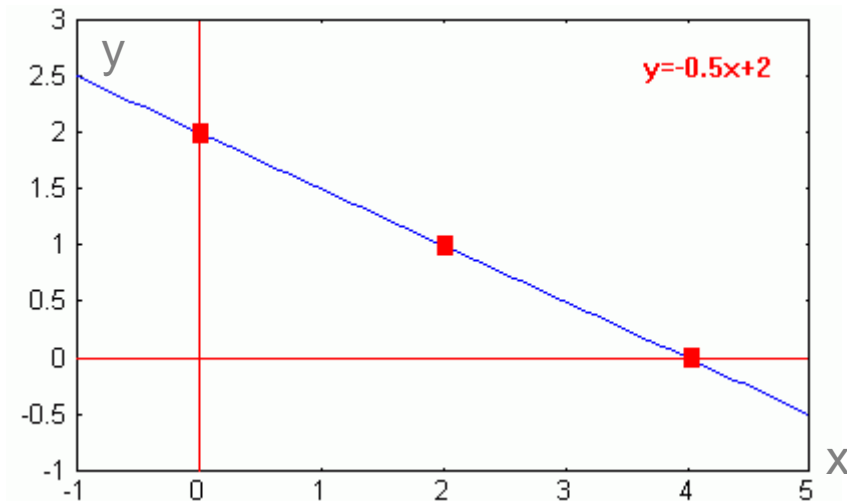
- Equazione classica della retta

$$y = mx + q$$

– può essere riscritta come

$$f((x,y), (m,q)) = y - mx - q = 0$$

- Fissato un punto (x_i, y_i) nello SI l'equazione $q = y_i - mx_i$ descrive la curva (che rimane ancora una retta)



HT: Ricerca di rette

- L'equazione classica della retta presenta dei problemi perché:

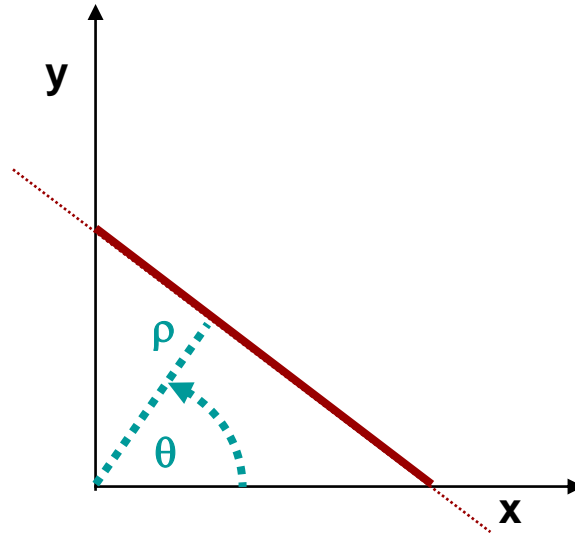
$$-\infty < m, q < +\infty$$

- Lo spazio immagine per immagini reali è ovviamente limitato (per esempio 256x256)
- Allo stesso modo sarebbe auspicabile che anche lo spazio dei parametri fosse limitato

HT: Ricerca di rette

- Per ovviare al problema descritto si usa una diversa rappresentazione della retta:

$$\rho = x \cos(\theta) + y \sin(\theta)$$



Esempio di rette

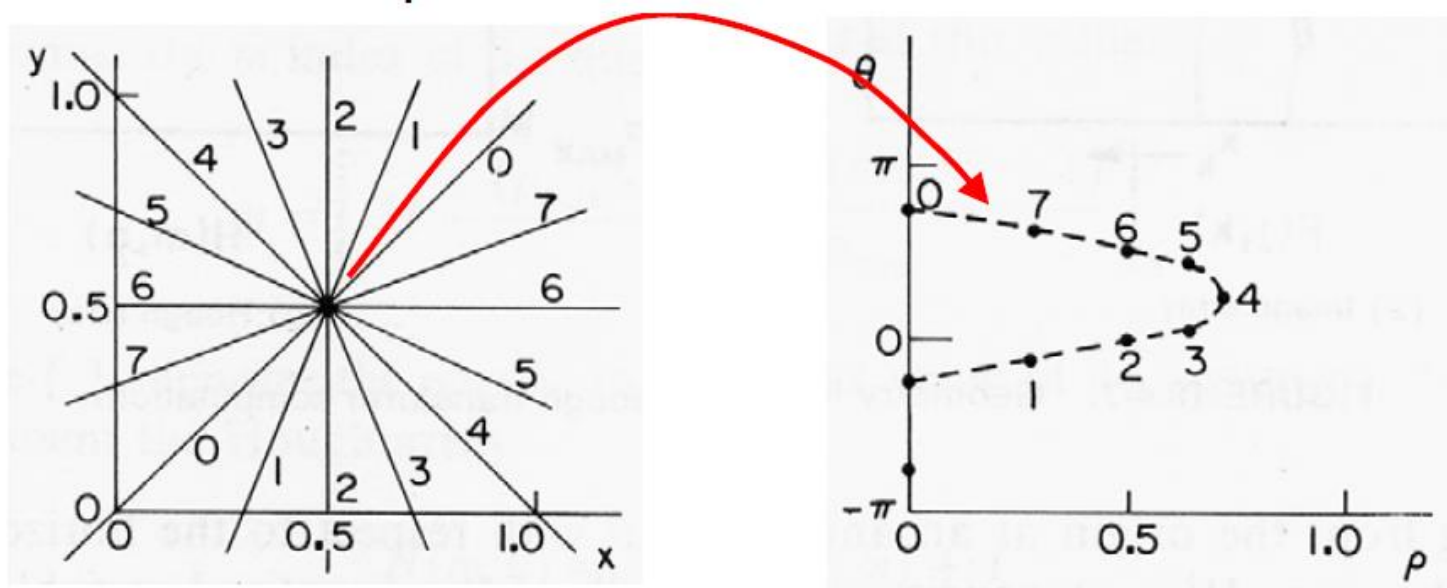
- Attraverso operatori locali (per esempio differenziali) si producono informazioni sul gradiente (locale)
- Decido una rappresentazione parametrica, per esempio (forma normale di Hesse)

$$\rho = x \cos(\theta) + y \sin(\theta).$$

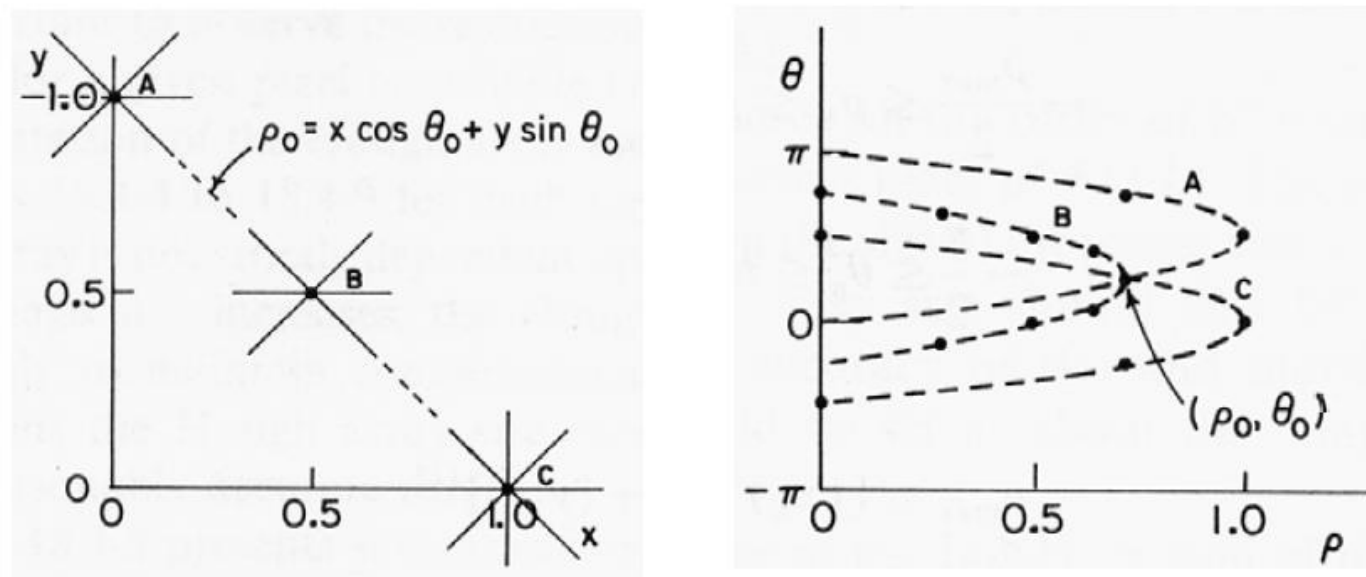
In questo caso i parametri risultano limitati, per esempio:

$$0 < \rho < L\sqrt{2}; -\pi \leq \theta \leq \pi.$$

Trasformata di un punto



Individuazione retta nello spazio dei parametri



Il processo di voto

- Da un punto di vista implementativo occorre discretizzare lo SP in celle (n-dimensionali)
- La dimensione delle celle dipende dalla precisione massima richiesta
- Ogni cella corrisponde ad una edizione (quantizzata) della curva
- Per ogni evidenza nello SI si incrementa ogni cella compatibile dello SP (**processo di voto**)
- Ogni cella misura perciò il numero di contributi al riconoscimento della curva corrispondente

HT per ricerca di segmenti rettilinei

- Definire un accumulatore $A(\theta, \rho)$ di dimensioni opportune
- Inizializzare $A(\theta, \rho)$ a 0
- Per ogni evidenza dell'immagine si incrementano tutti i punti dell'accumulatore compatibili con una regola di mappatura e pesi predefiniti.
- Seleziona i massimi (magari con un'analisi locale) nell'accumulatore: corrispondono alle edizioni più votate della curva cercata.

Esempio di operatore globale: trasformata di Hough

