Corso di Visione Artificiale

Features

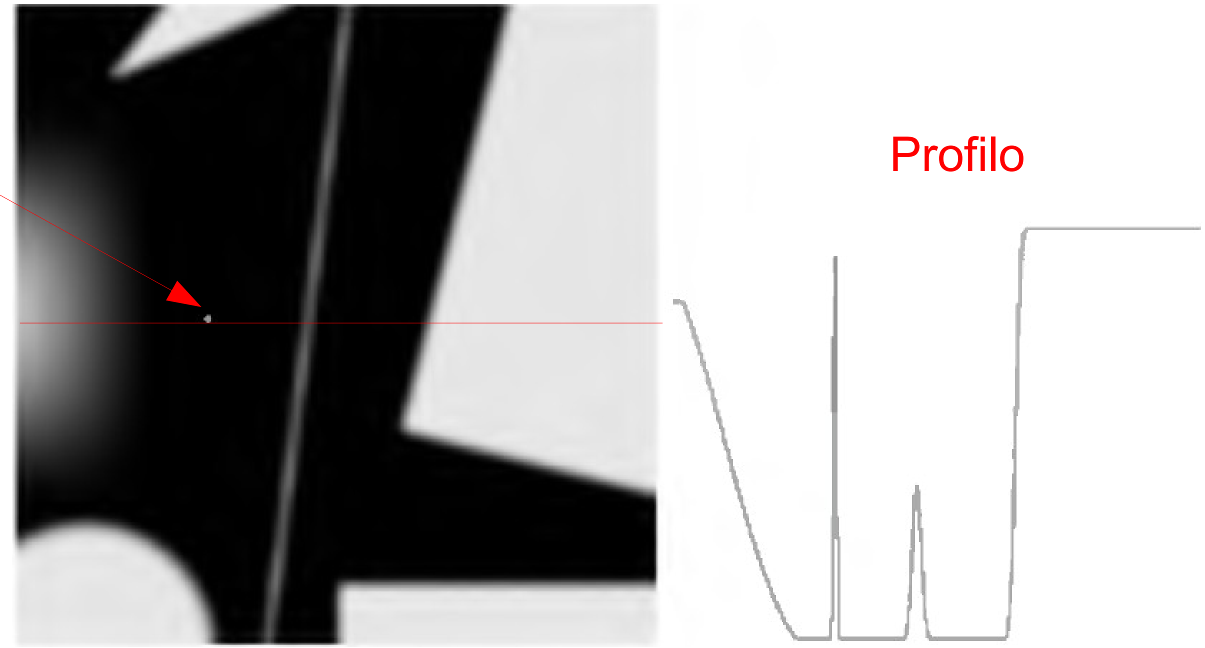
Samuel Rota Bulò

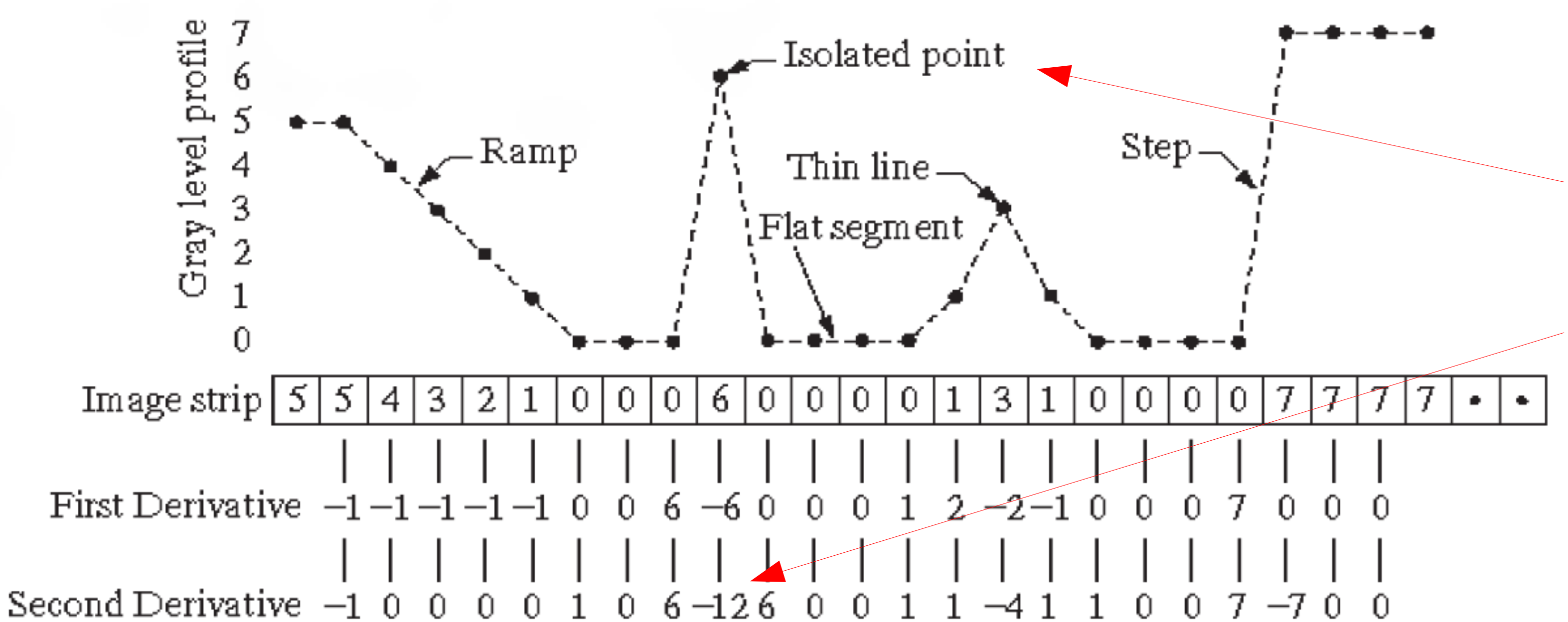
Features

- Le features (caratteristiche) sono parti di un'immagine che sono:
  - **locali**: caratteristica locale di un'immagine,
  - **significativi**: sono interessanti per il problema specifico,
  - **rilevabili**: esiste un modo per trovarle.
- es:. bordi (edges), linee, ellissi, angoli, textures, ...
- forniscono un'utile astrazione dell'immagine.
- la scelta di quali utilizzare dipende dal problema specifico che si affronta
- hanno spesso dei descrittori che forniscono informazioni sulla feature.

Rilevamento di punti isolati

- Per rilevare un punto isolato possiamo ricorrere ad una strategia basata sulla derivata seconda.





Picco (positivo o negativo) nella derivata seconda in corrispondenza di punti isolati

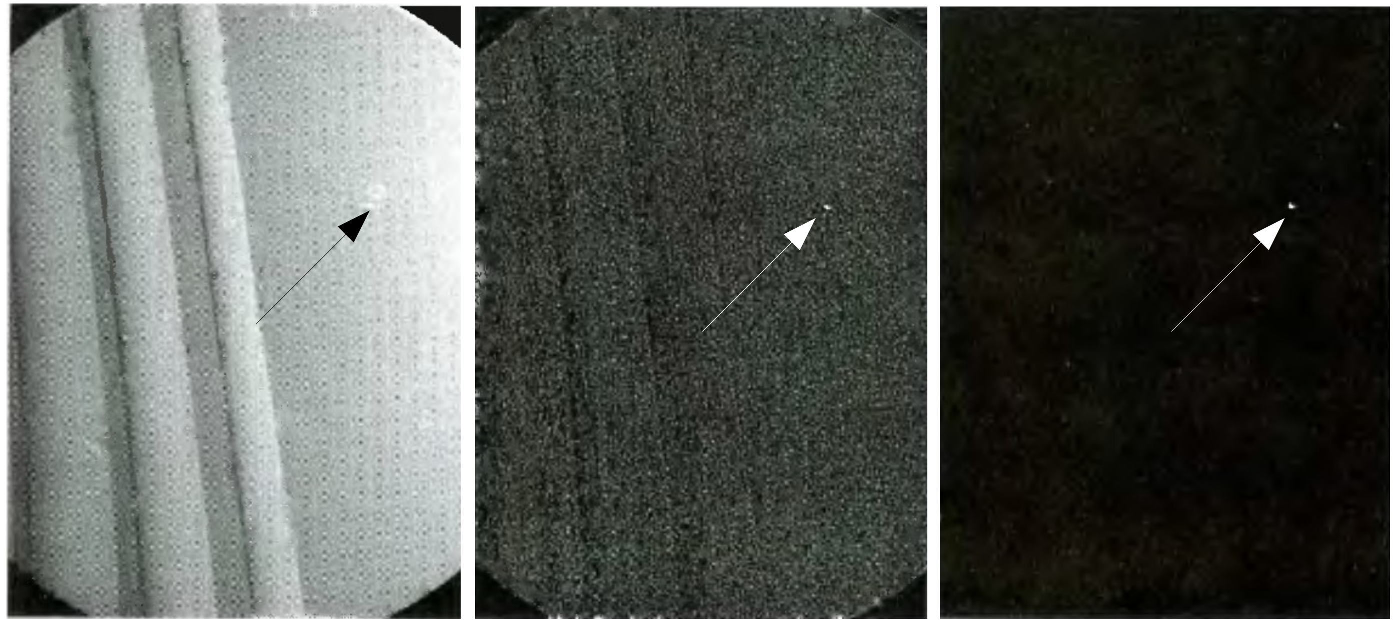
Rilevamento di punti isolati

FILTRO LAPLACIANO

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

RISPOSTE DEL FILTRO LAPLACIANO

$$\nabla^2 I = L * I$$



I $\quad$ $\nabla^2 I$ $\quad$ B

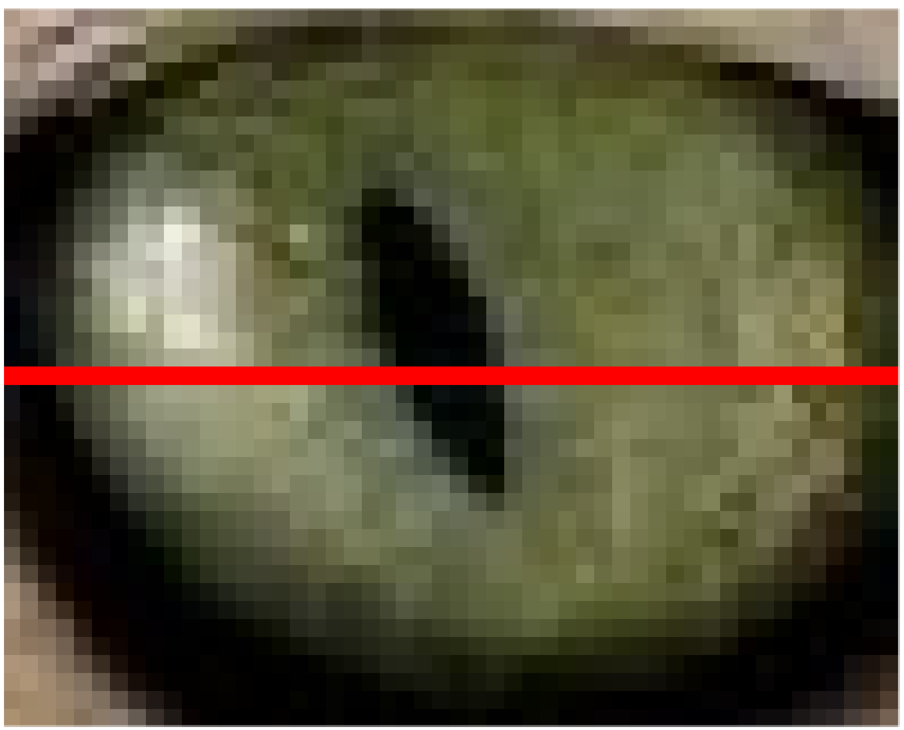
SOGLIATURA

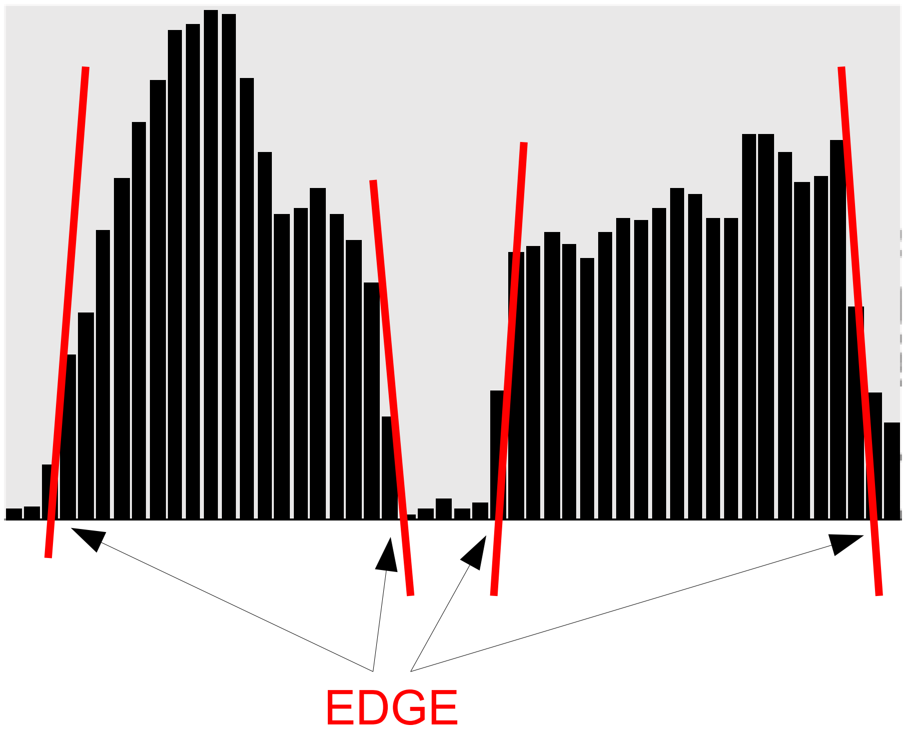
$$B[x,y] = \begin{cases} 1 & \text{se } |\nabla^2 I[x,y]| \geq \tau \\ 0 & \text{altrimenti} \end{cases}$$

- Abbiamo un punto isolato se la risposta del filtro Laplaciano è sufficientemente elevata.

Rilevamento di bordi

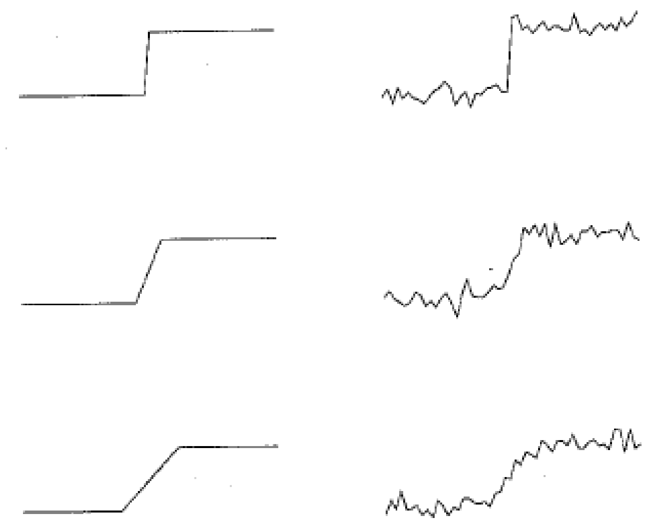
- Un **bordo** (o **edge)** è un punto dell'immagine attorno al quale troviamo una forte variazione di intensità.
- I bordi delineano oggetti, ombre ....
- Facilitano il rilevamento di linee, curve, contorni.





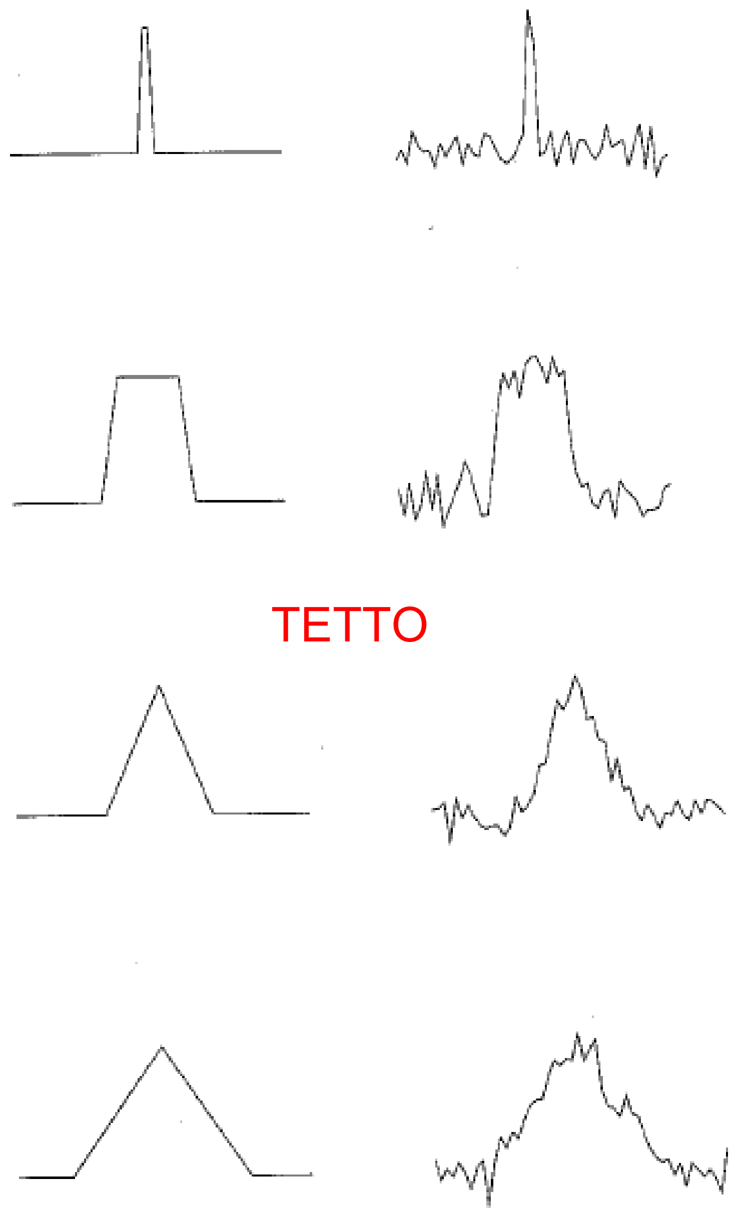
Classificazione di bordi

GRADINO



RAMPA

CRESTA



TETTO

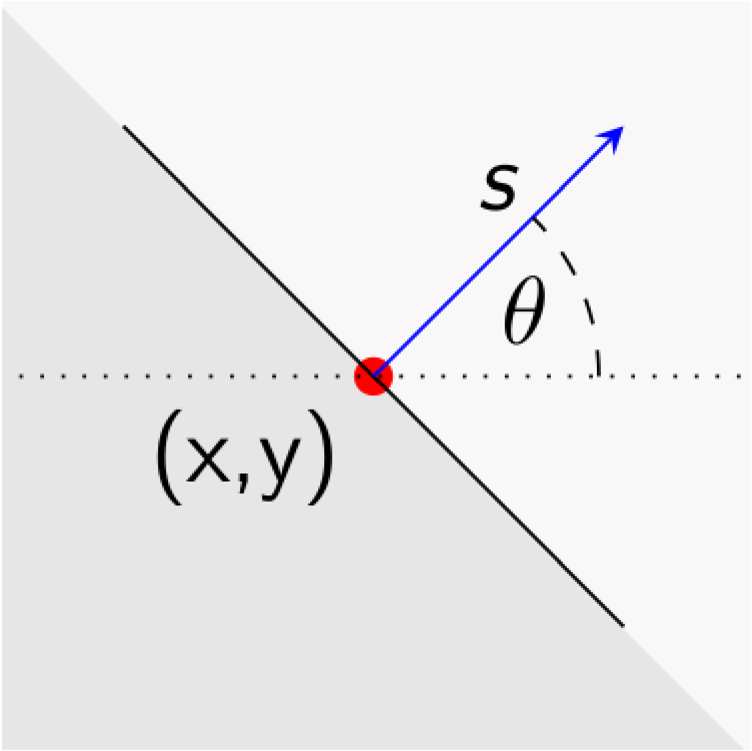
- **bordi a gradino / rampa** sono comuni e delineano i contorni di regioni con intensità differenti

- **bordi a cresta** sono generati da linee spesse. Corrispondono a 2 step-edges.

- **bordi a tetto** sono generati da linee sottili.

Modello per bordo a gradino

- Un **bordo** può essere rappresentato da una tupla (x,y, θ, s).
  - (x,y) posizione del pixel
  - θ direzione di massima variazione di intensità
  - s intensità della variazione di intensità



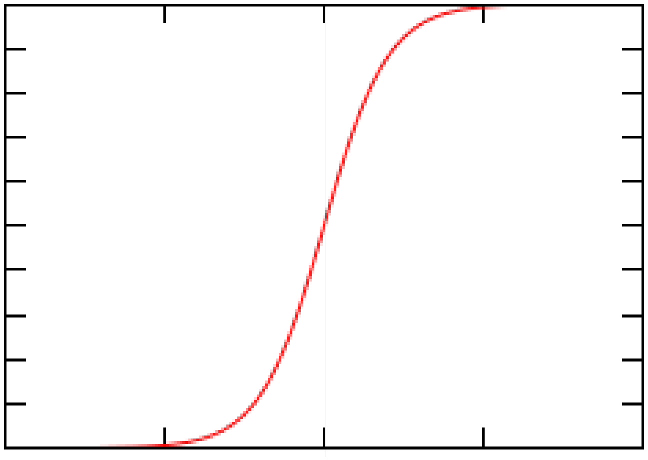
Problema del rilevamento di bordi

- Data un'immagine corrotta da rumore di acquisizione, cercare i bordi che sono generati da elementi della scena, evitando quelli generati da rumore.

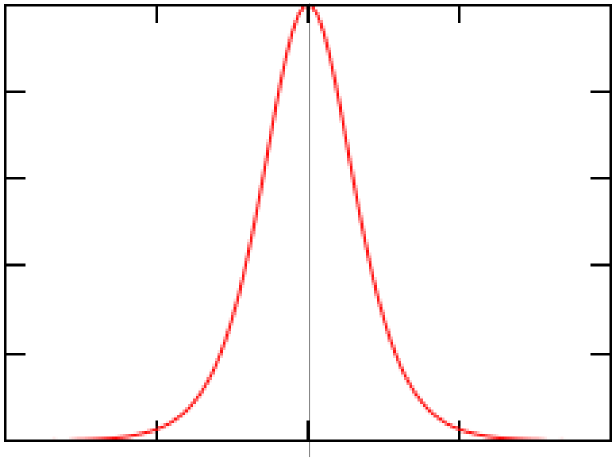
Strategie

- Strategie di **primo ordine**
  - trovare il massimo della derivata prima
- Strategie di **secondo ordine**
  - trovare lo zero-crossing della derivata seconda

Smooth step edge

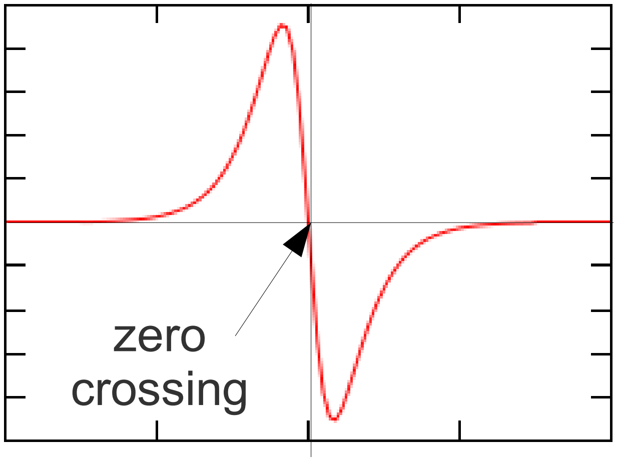


Derivata prima



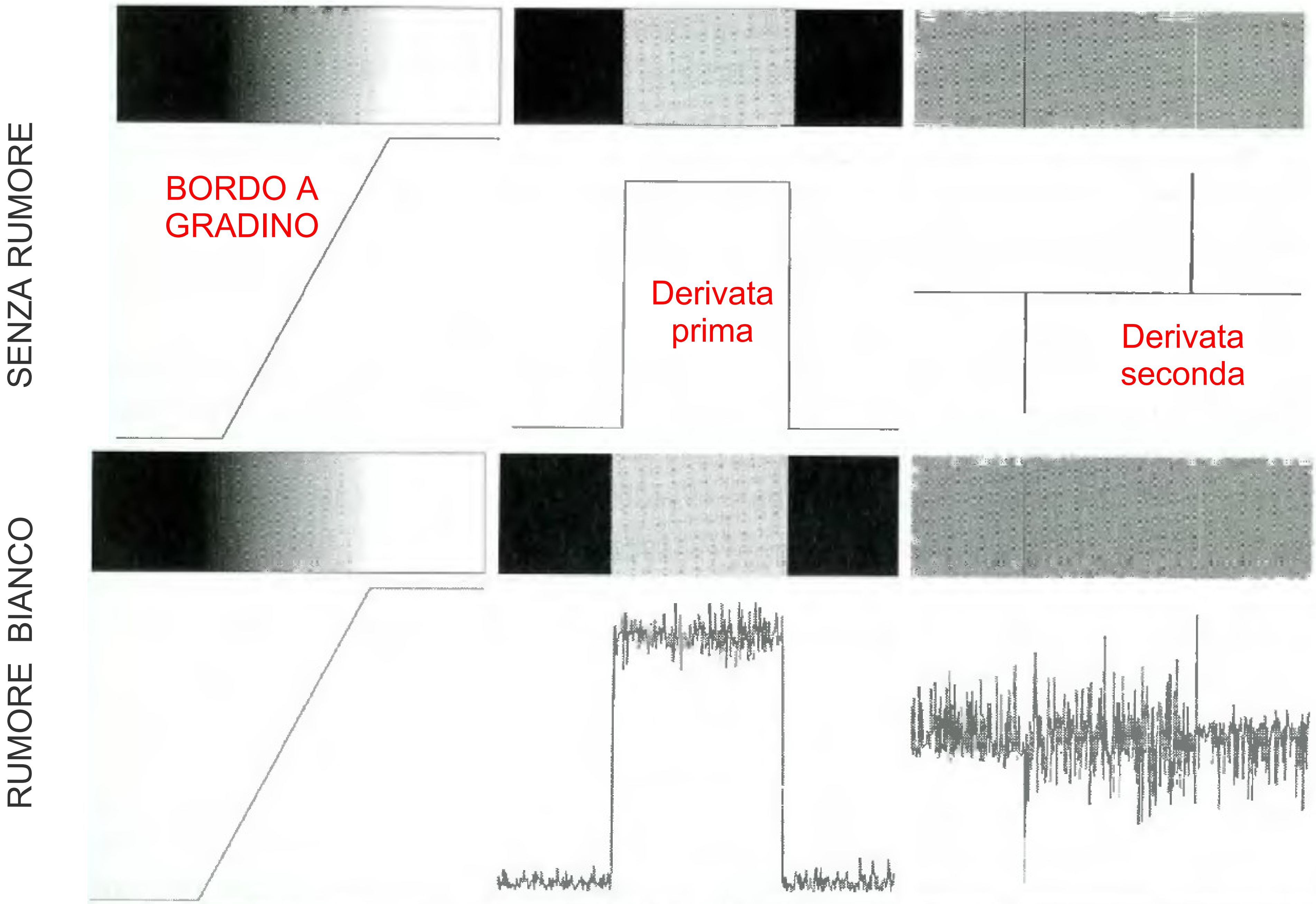
rilevamento in base a magnitudo della derivata

Derivata seconda



rilevamento in base a variazione di segno della derivata

Bordi e rumore

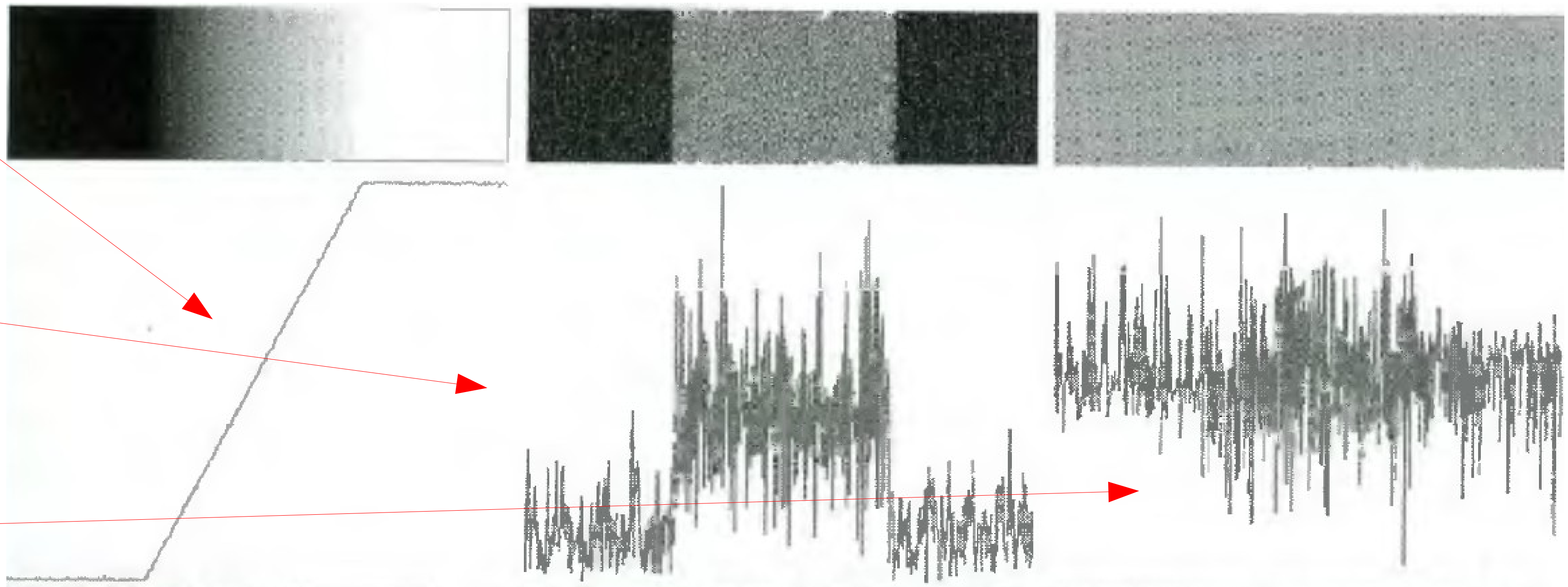


Bordi e rumore

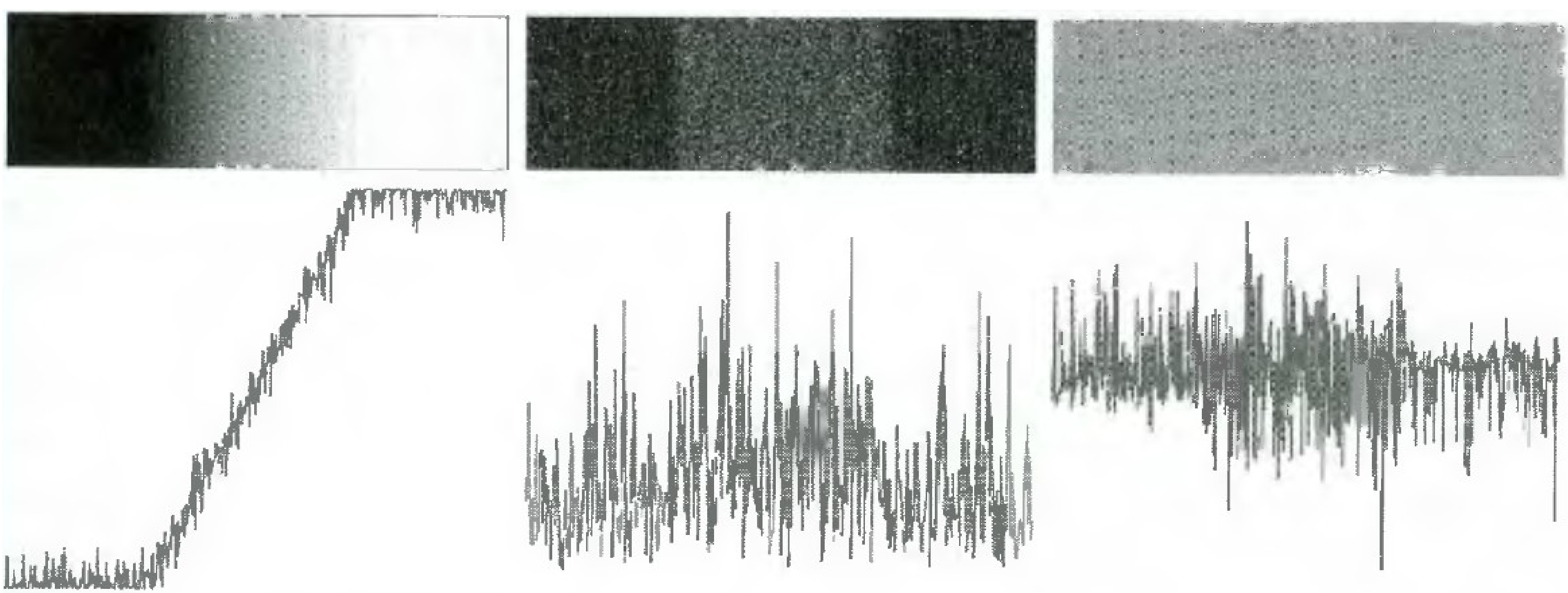
Il rumore è impercettibile.

La derivata prima ha accentuato il rumore.

La derivata seconda lo ha accentuato ulteriormente



ULTERIORE RUMORE



Fasi di un rilevamento di bordi

1. **Smoothing dell'immagine:** per ridurre il rumore (abbiamo visto nelle slides precedenti il motivo !).
2. **Rilevamento di bordi:** adottando un metodo di primo o secondo ordine estraiamo un insieme di potenziali candidati.
3. **Localizzazione di bordi**: filtrare i falsi positivi (punti che non sono un bordo) dall'insieme di candidati bordo.

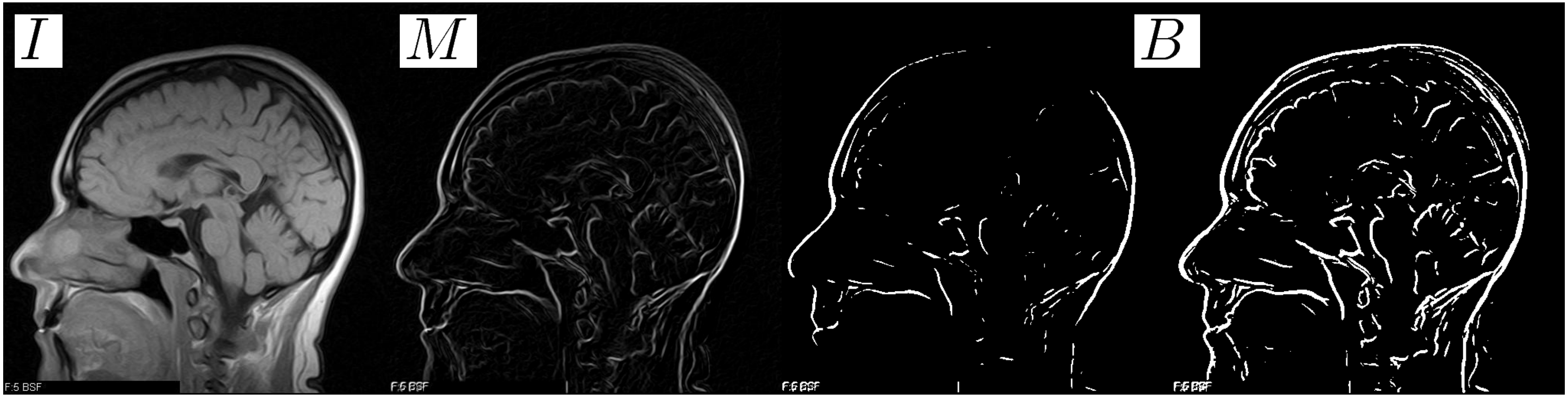
Rilevamento per sogliatura

- Abbiamo visto che in corrispondenza di bordi troviamo un picco della derivata prima.
- Un metodo semplice di estrazione di bordi consiste nel sogliare il magnitudo del gradiente dell'immagine.

$$M(x,y) = \|\nabla I(x,y)\| = \sqrt{I_x^2(x,y) + I_y^2(x,y)}$$

$$B(x,y) = \begin{cases} 1 & \text{se } M(x,y) > \tau \\ 0 & \text{altrimenti} \end{cases}$$

SOGLIA



Algoritmo di Marr-Hildreth

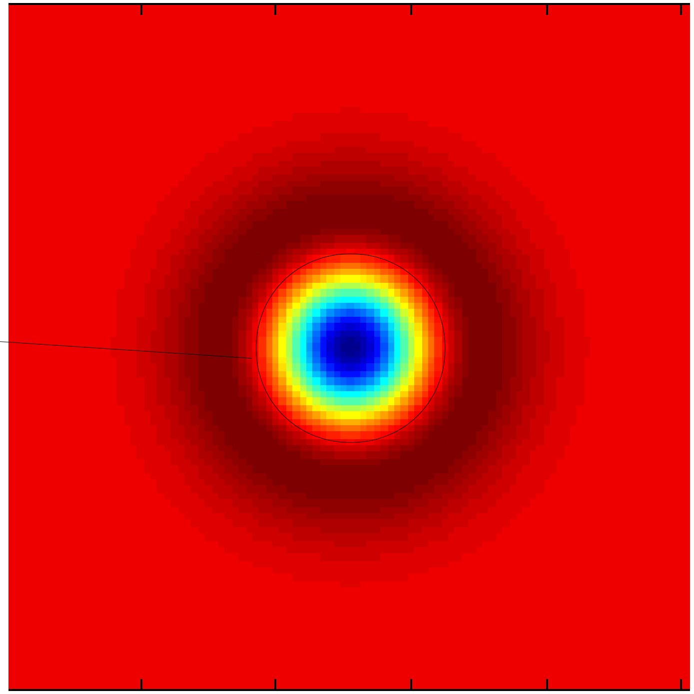
- Basato sull'idea che
  1. variazione di intensità dipendente dalla scala
  2. variazioni brusche di intensità danno vita ad un picco nella derivata prima e un zero-crossing della derivata seconda.
- L'operatore utilizzato per rilevare bordi deve essere differenziale per calcolare un'approssimazione della derivata prima o seconda e deve essere adattabile a scale diverse.
- Marr e Hildreth (1980) optarono per il filtro chiamato Laplacian of a Gaussian (LoG)

$$\nabla^2 G_\sigma(x, y) = \left[G_\sigma^x(x, y)\right]^2 + \left[G_\sigma^y(x, y)\right]^2$$

- LoG essendo isotropico risponde allo stesso modo a variazioni di intensità con direzioni diverse.
- Si può quindi evitare l'utilizzo di diversi filtri differenziali direzionati (uno per ogni direzione).

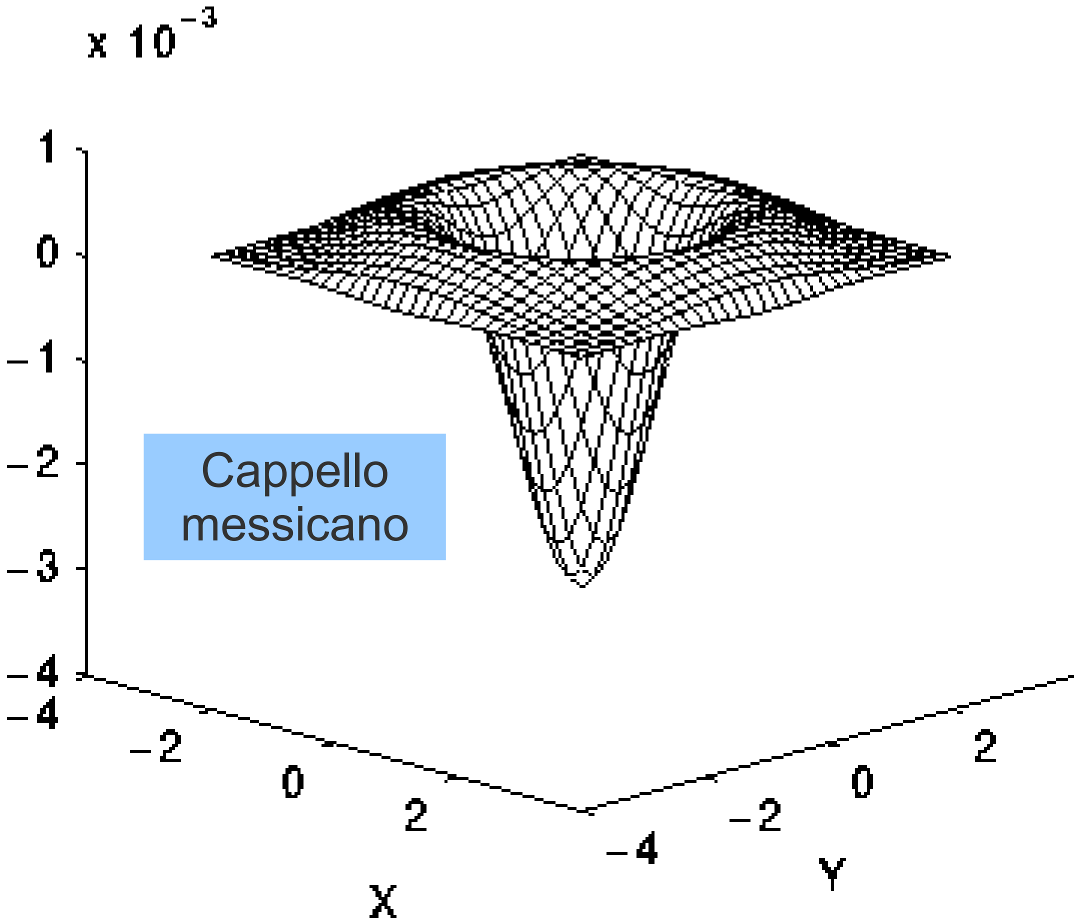
LoG

$$\nabla^2 G_\sigma(x,y) \propto \left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



zero crossing

$$x^2+y^2=2\sigma^2$$



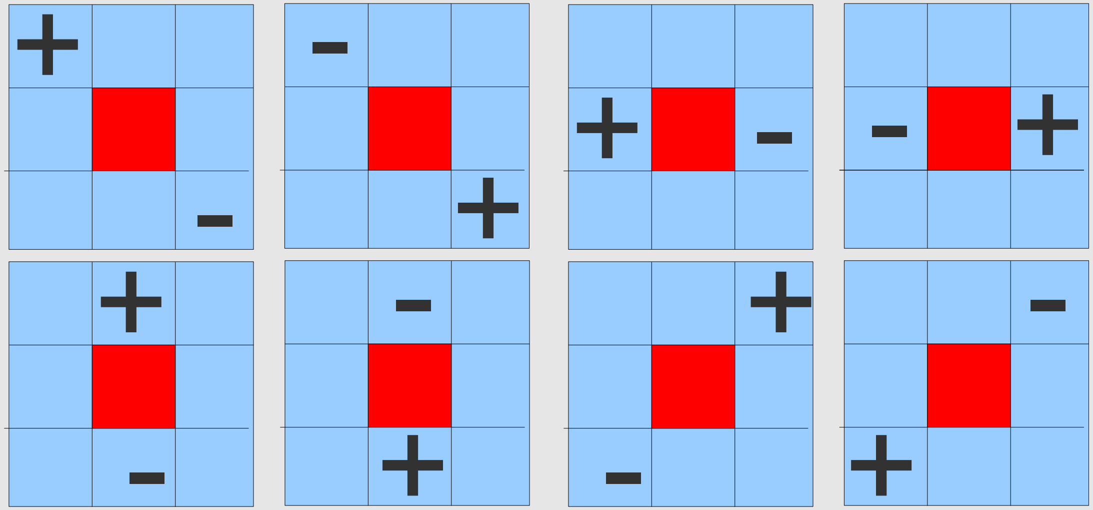
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Algoritmo di Marr-Hildreth

1. Applicare un filtro LoG all'immagine

$$J = \nabla^2 G_\sigma * I$$

2. Trovare gli zero-crossing di J

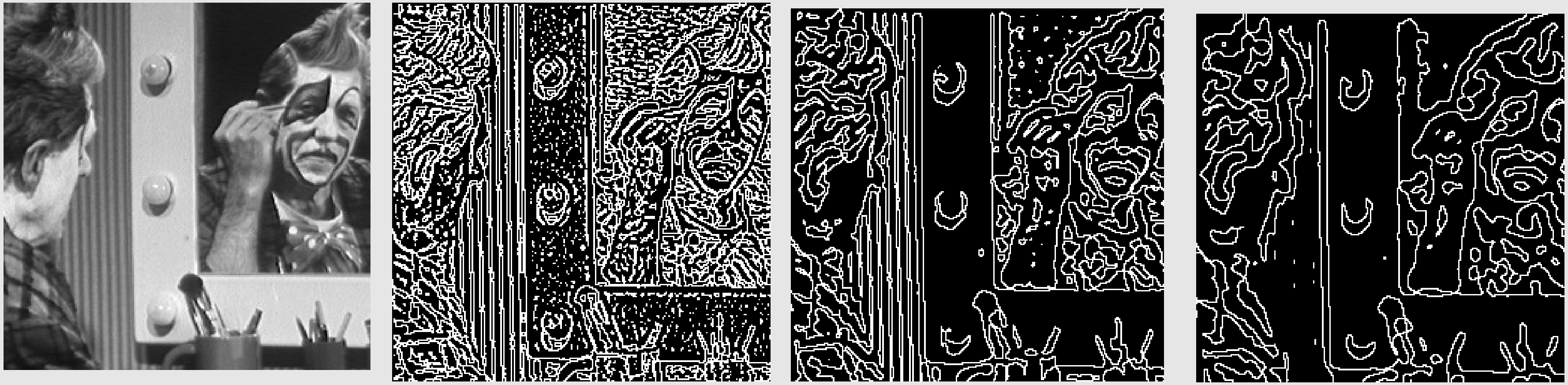


- Possiamo ottenere un filtro LoG dalla convoluzione di una gaussiana e un filtro Laplaciano L

$$\nabla^2 G_\sigma = L * G_\sigma$$

Algoritmo di Marr-Hildreth

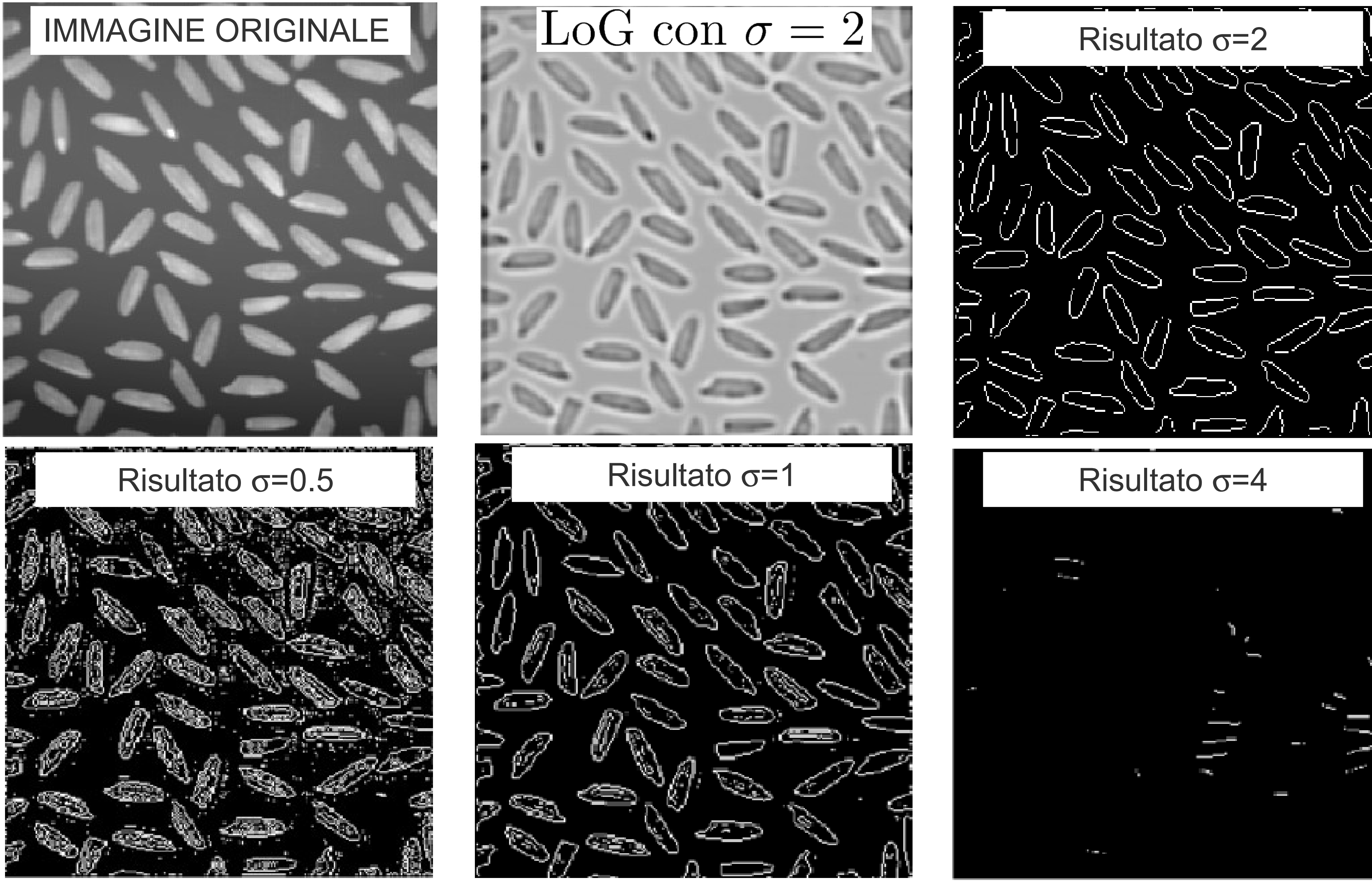
- La scelta del fattore di scala σ influisce sul tipo di dettaglio evidenziato
- Per ottenere risultati più affidabili Marr e Hildreth suggerirono di filtrare l'immagine con filtri LoG a varie scale (σ diversi), individuare gli zero-crossing per ciascun filtro e combinare i risultati.



$\sigma = 1$ $\sigma = 2$ $\sigma = 3$

- In pratica si sceglie il fattore di scala più idoneo all'applicazione specifica.

Effetto della scala



DoG

- Marr e Hildreth notarono che è possibile approssimare il filtro LoG con una differenza di gaussiane (DoG):

$$\mathrm{DoG} = G_{\sigma_1} - G_{\sigma_2}$$

dove $\sigma_1 > \sigma_2$.

- Una buona approssimazione della LoG (a meno di un fattore di scala) si ottiene scegliendo

$$\frac{\sigma_1}{\sigma_2} \approx 1.6$$

- LoG è in generale preferito a DoG, tuttavia risultati sperimentali suggeriscono che certi recettori del sistema visivo umano sono selettivi rispetto ad orientamento e frequenza e sono modellabili con DoG.

Algoritmo di Canny

- L'approccio di Canny (1986) si basa su 3 obiettivi
  - **tasso d'errore basso**: tutti i bordi devono essere trovati con alta probabilità evitando risposte spurie (falsi positivi);
  - **buona localizzazione**: la distanza tra il bordo rilevato e il bordo vero deve essere minima;
  - **unicità della risposta per bordo**: ogni bordo reale deve generare un'unica risposta.
- L'essenza del lavoro di Canny è stato quello di formalizzare matematicamente i 3 criteri elencati per poi cercare di trovare una soluzione ottimale.
- La soluzione è unica ma non in forma chiusa.
- Approssimabile con un **filtro differenziale Gaussiano**.

Algoritmo di Canny

- La prima fase consiste quindi nel calcolare il gradiente dell'immagine utilizzando il filtro differenziale Gaussiano.

$$\nabla I = \begin{bmatrix} G_\sigma^x * I \\ G_\sigma^y * I \end{bmatrix} = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

- Calcoliamo poi il magnitudo del gradiente e la direzione:

$$M(x,y) = \|\nabla I(x,y)\| = \sqrt{I_x(x,y)^2 + I_y(x,y)^2}$$

Ci interessa solo la direzione non il verso.

$$\alpha(x,y) = \tan^{-1}\left(\frac{I_y(x,y)}{I_x(x,y)}\right)$$

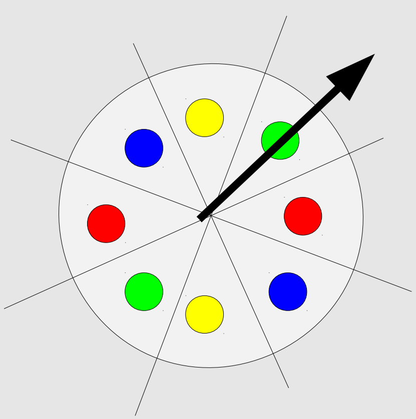
- Il magnitudo del gradiente presenta delle ampie rampe intorno ai massimi locali che sono in corrispondenza di bordi. Una semplice sogliatura quindi non è sufficiente

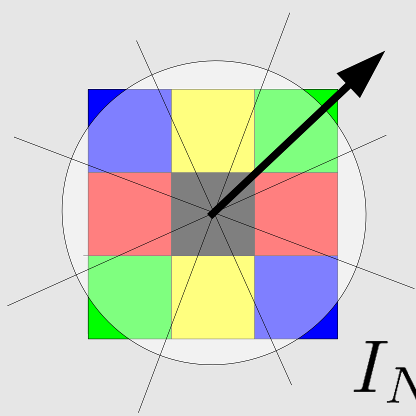
Soppressione dei non-massimi

- La strategia adottata per identificare i massimi locali, consiste nel sopprimere i non-massimi basandosi sull'idea che ogni pixel centrato in un bordo avrà un intensità superiore rispetto ai 2 pixels vicini nella direzione del gradiente.

- Dividiamo l'intorno di un pixel in 4 regioni (giallo, verde, rosso e blu) corrispondenti a 4 direzioni. Verifichiamo in che regione cade il vettore gradiente del pixel in centro. Se l'intensità del pixel centrale I(**p**) è inferiore all'intensità di almeno uno dei 2 pixel vicini lungo la direzione del gradiente, il pixel viene soppresso. Facendolo per ogni pixel otteniamo una nuova immagine I_N.

$$\begin{array}{c|c|c} * & * & M(\mathbf{q}_2) \\ \hline * & M(\mathbf{p}) & * \\ \hline M(\mathbf{q}_1) & * & * \end{array}$$





$$I_N(\mathbf{p}) = \begin{cases} M(\mathbf{p}) & \text{se } M(\mathbf{p}) \geq M(\mathbf{q}_{1,2}) \\ 0 & \text{altrimenti} \end{cases}$$

Sogliatura con isteresi

- L'ultima operazione consiste nel sogliare I_N per ridurre i falsi bordi.
- Abbiamo già visto che usare una sola soglia ha il problema di lasciare falsi bordi se troppo bassa e rimuovere bordi reali se troppo elevata.
- L'algoritmo di Canny utilizza 2 soglie $\tau_L < \tau_H$. Canny suggerì di scegliere le due soglie in modo da avere

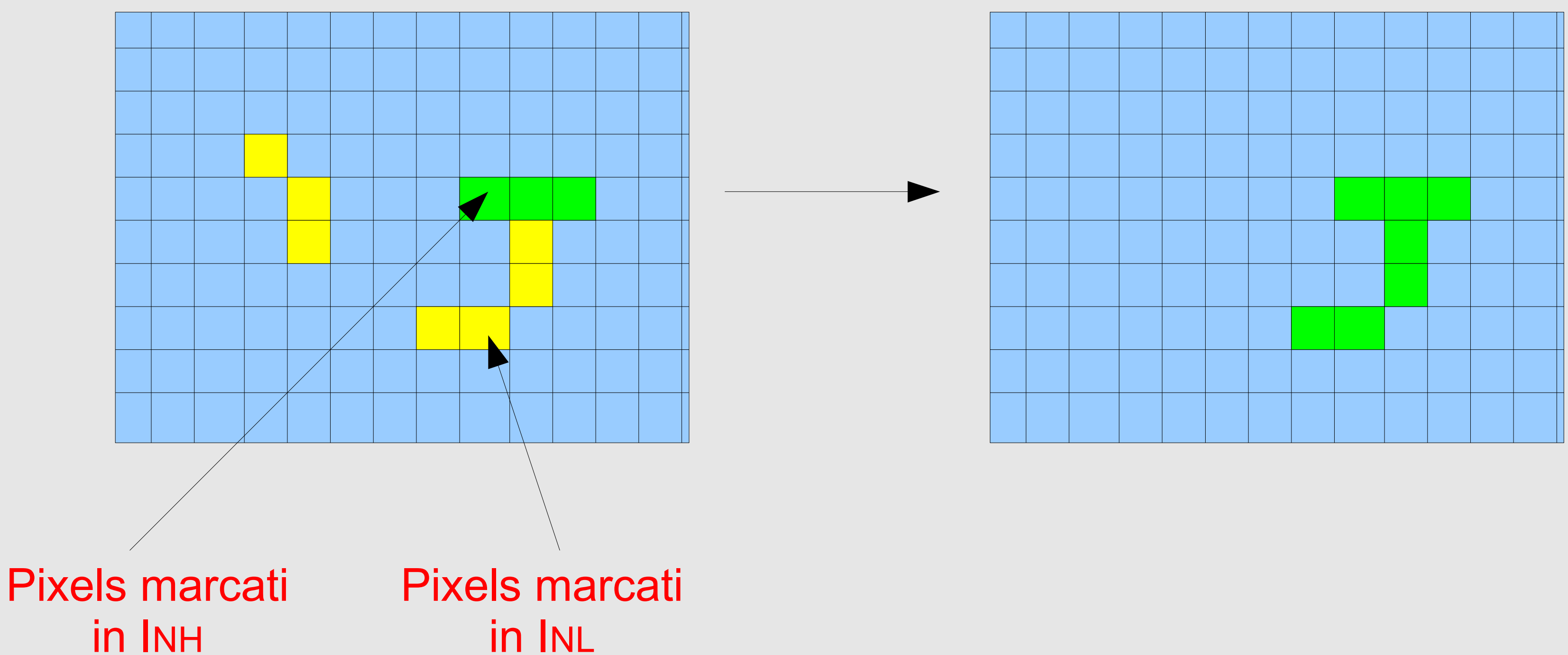
$$2 \leq \frac{\tau_H}{\tau_L} \leq 3$$

- Calcoliamo 2 immagini di bordi “forti” ed bordi “deboli”

$$I_{NH}(x,y) = I_N(x,y) \geq \tau_H$$
$$I_{NL}(x,y) = \tau_L \leq I_N(x,y) < \tau_H$$

Sogliatura con isteresi

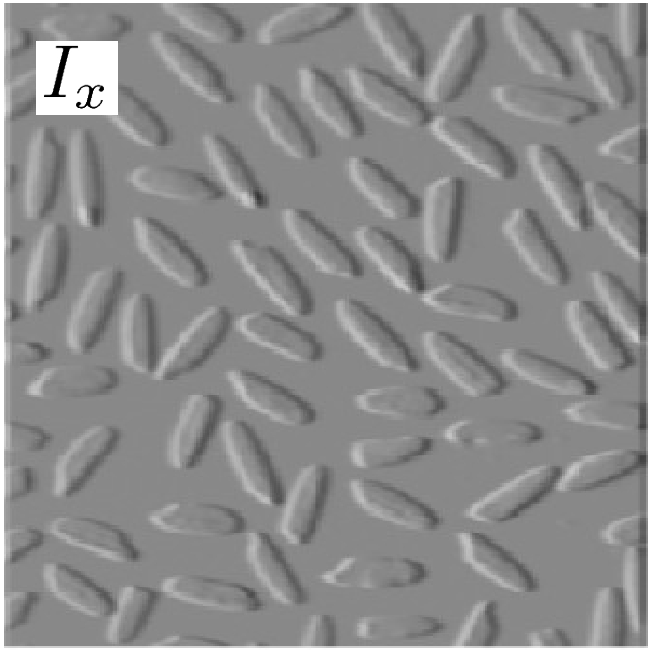
- I pixel marcati in I_{NH} sono considerati validi.
- A seconda del valore della soglia T_H i bordi presentano dei buchi. Per ottenere bordi più lunghi si considerano bordi validi anche tutti quei pixels marcati in I_{NL} che sono connessi direttamente o tramite una catena a un pixel marcato in I_{NH}.

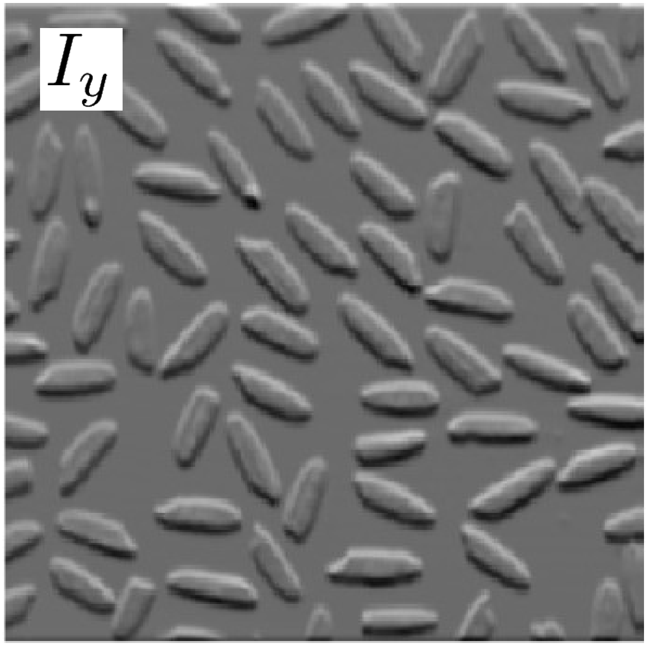


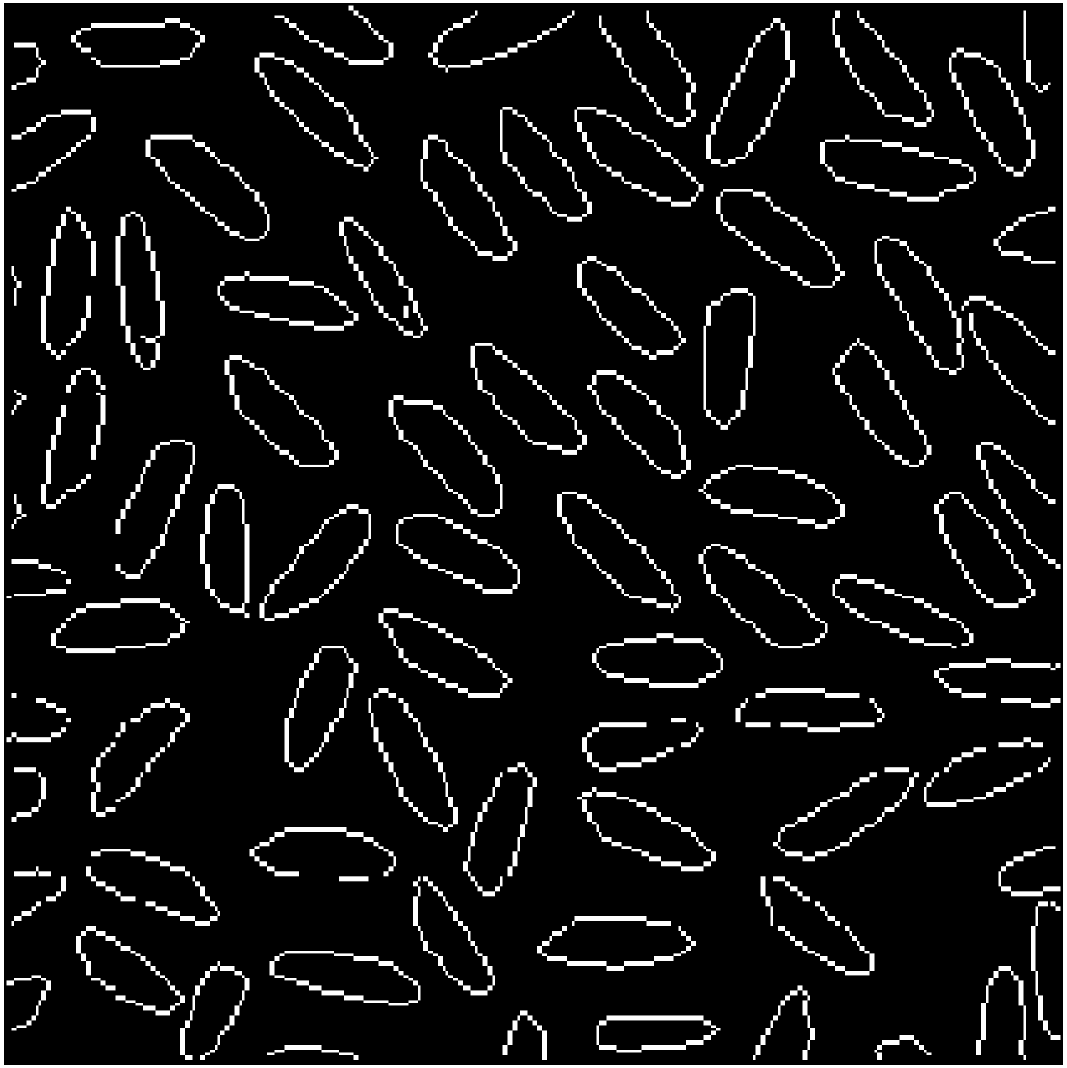
Algoritmo di Canny

1. Smussare l'immagine e calcolarne le derivate (o in alternativa utilizzare filtri differenziali smooth)
2. Calcolare il magnitudo e angolo del gradiente per ogni pixel.
3. Applicare la soppressione dei non massimi.
4. Usare la sogliatura con isteresi e l'analisi della connettività per rilevare e connettere i bordi.

Algoritmo di Canny

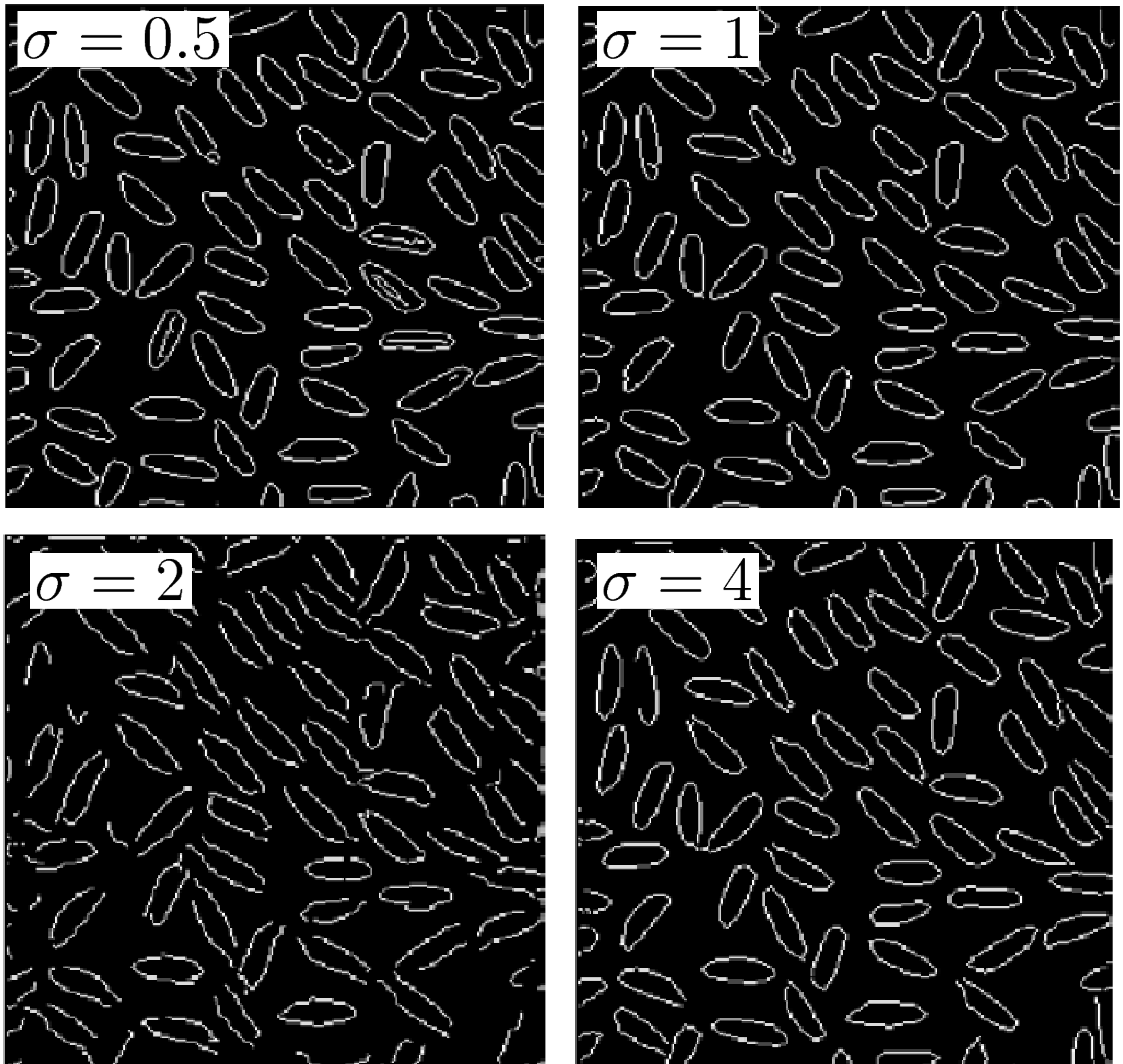






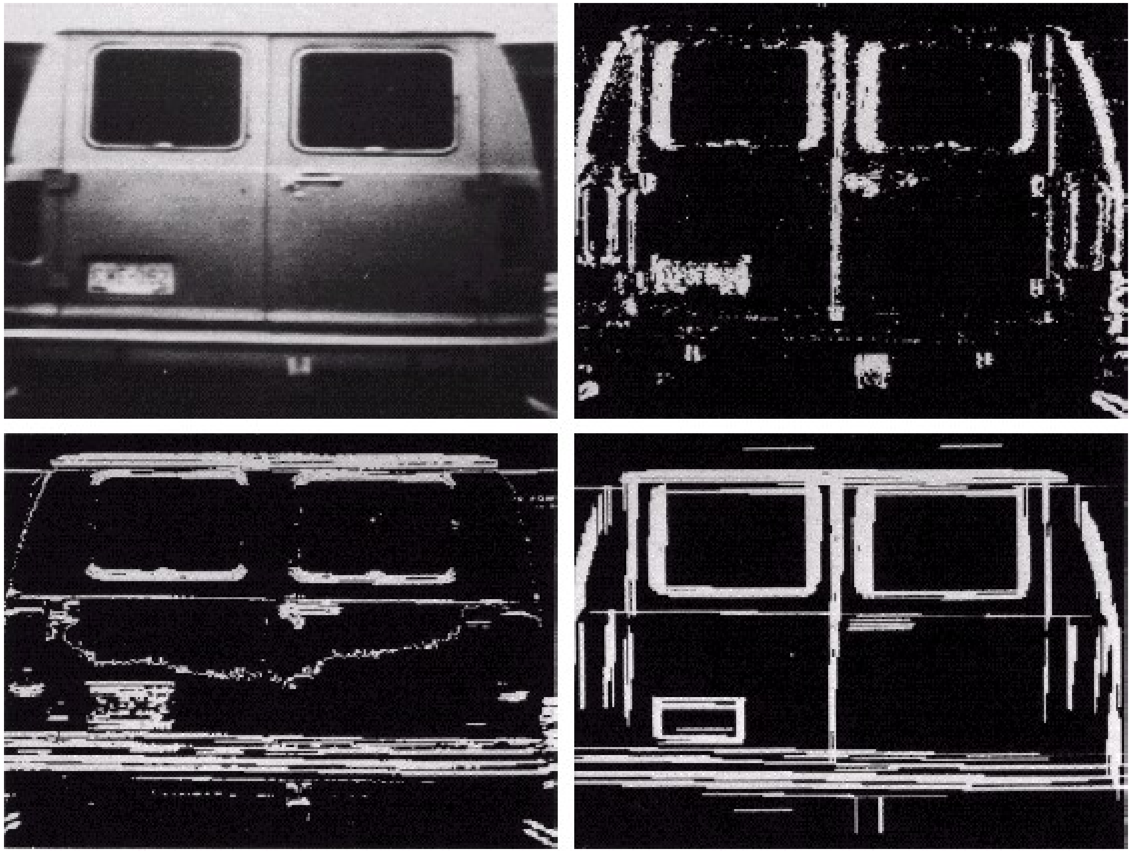
$\sigma = 1 \qquad \tau_H = 0.1875 \qquad \tau_L = 0.075$

Algoritmo di Canny



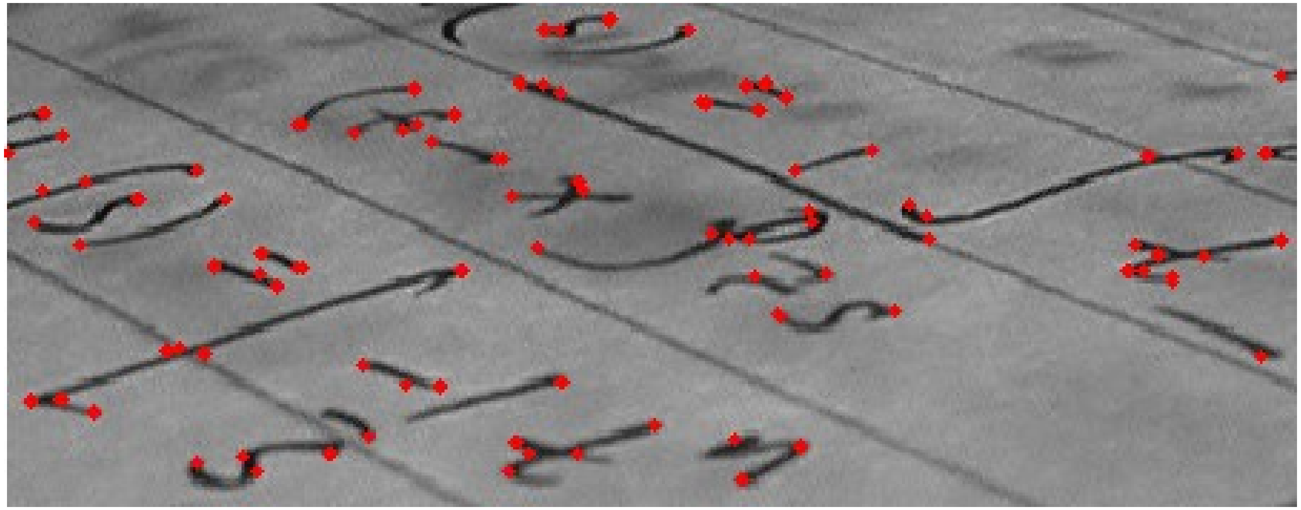
Connessione di Bordi

- I bordi che tipicamente vengono rilevati con un algoritmo di rilevamento bordi presentano delle discontinuità dovute a rumore o condizioni di illuminazioni variabile che impediscono di avere linee ben definite
- Per ovviare a questo si adotta una tecnica di connessione di edge.
- Distinguiamo 2 tipologie di tecniche: locali e globali.



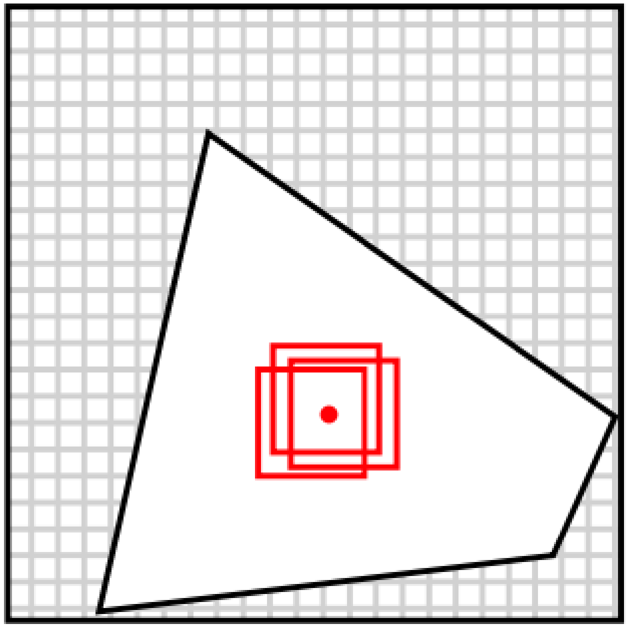
Rilevamento di punti salienti

- Un **punto saliente** è un punto dell'immagine che ha
  - una chiara definizione matematica
  - una posizione ben definita
  - ricco di informazione locale
  - stabile sotto perturbazioni locali/globali (alto grado di riproducibilità)

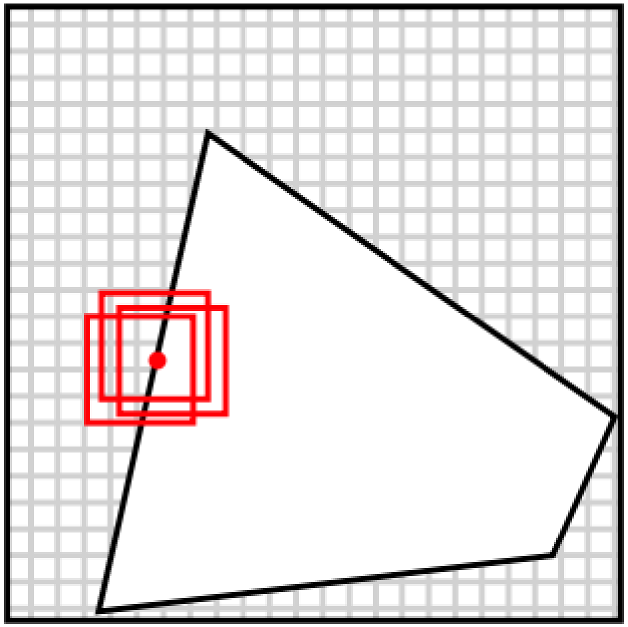


Auto-correlazione

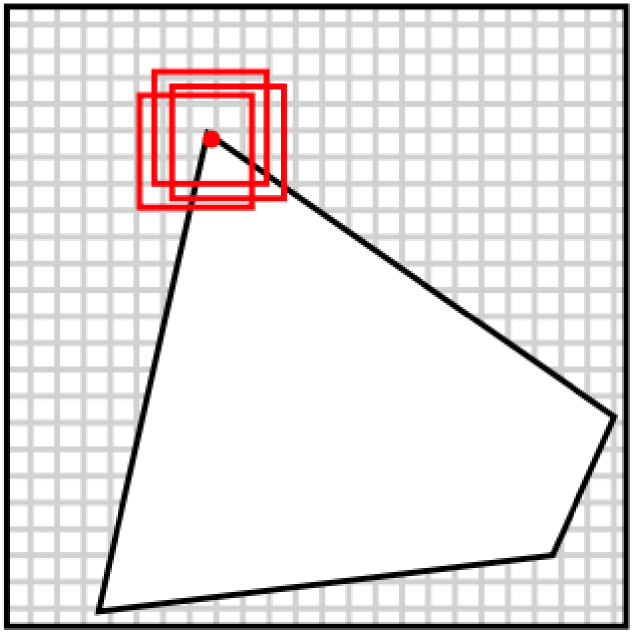
- Idea: utilizziamo l'auto-correlazione per capire se un punto ha un contesto locale ricco di informazione



REGIONE PIATTA
nessun cambiamento locale



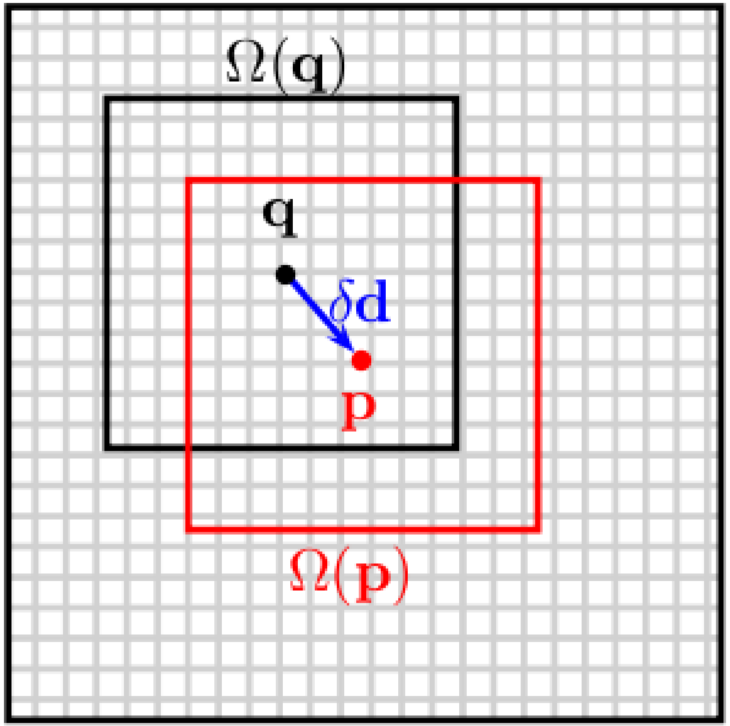
BORDO
nessun cambiamento lungo il bordo



PUNTO SALIENTE
cambiamento in ogni direzione

Rilevatore di Harris

- Consideriamo un pixel **q** e un suo intorno locale Ω(**q**).
- Confrontiamo Ω(**q**) e una patch centrata in **q**+δ**d** (assumiamo **d** un vettore di norma unitaria) con δ infinitesimale.



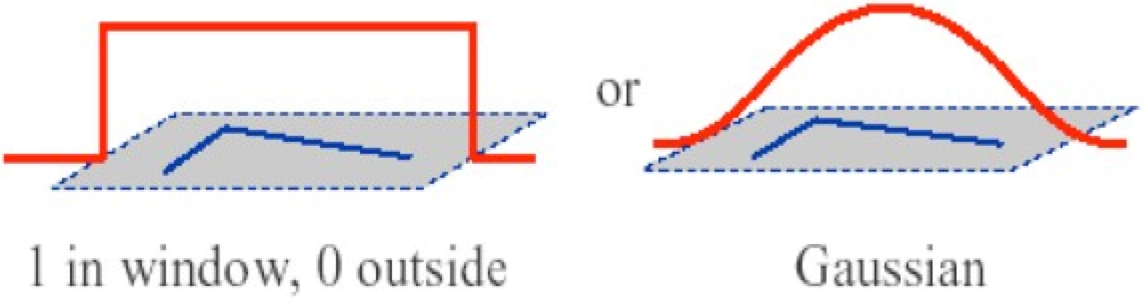
$$D_{\mathbf{q}}(\mathbf{d}) = \sum_{\mathbf{r}\in\Omega(\mathbf{q})} \left[\mathbf{d}^\top \nabla I(\mathbf{r})\right]^2$$

misura il contenuto informativo dell'immagine I in un punto **q** nella direzione **d**, rispetto all'intorno Ω(**q**).

Rilevatore di Harris

$$D_{\mathbf{q}}(\mathbf{d}) = \sum_{\mathbf{r}\in\Omega(\mathbf{q})} w(\mathbf{r}) \left[\mathbf{d}^\top \nabla I(\mathbf{r})\right]^2 \qquad \nabla I(\mathbf{r}) = \begin{bmatrix} I_x(\mathbf{r}) \\ I_y(\mathbf{r}) \end{bmatrix}$$

potremmo non voler dare a tutti i punti nell'intorno lo stesso peso. Per esempio W potrebbe essere una gaussiana centrata in **q**.



In forma matriciale

$$D_{\mathbf{q}}(\mathbf{d}) = \mathbf{d}^\top \begin{pmatrix} \sum_{\Omega(\mathbf{q})} w I_x^2 & \sum_{\Omega(\mathbf{q})} w I_x I_y \\ \sum_{\Omega(\mathbf{q})} w I_y I_x & \sum_{\Omega(\mathbf{q})} w I_y^2 \end{pmatrix} \mathbf{d} = \mathbf{d}^\top C \mathbf{d}$$

- La matrice C è detta matrice di auto-correlazione

Rilevatore di Harris

- Abbiamo un punto saliente in **q** se in tutte le direzioni **d** il contenuto informativo $D_q(d)$ è significativo.
- In altre parole, se esiste un (significativo) τ positivo per cui

$$\min \left\{ \mathbf{d}^\top C \mathbf{d} \,:\, \mathbf{d} \in \mathbb{R}^2,\ \|\mathbf{d}\| = 1 \right\} > \tau$$

- questo è equivalente a

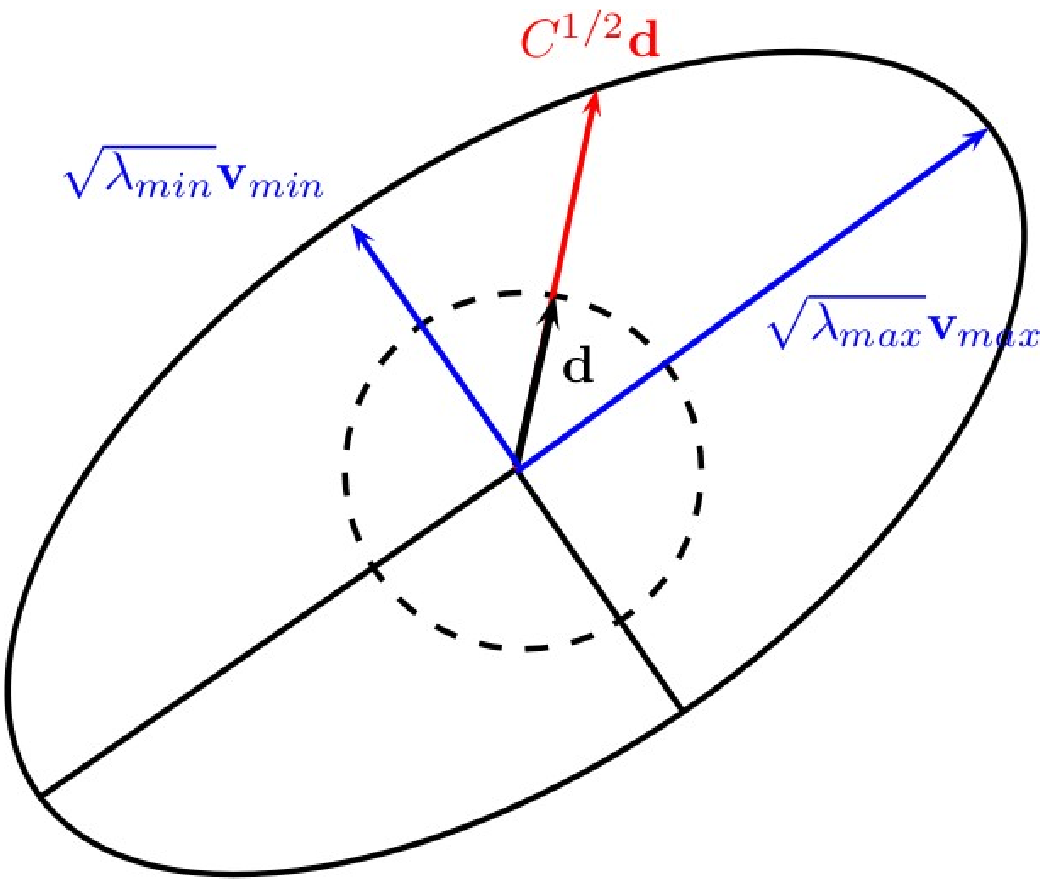
$$\lambda_{\min}(C) > \tau$$

- Notare che entrambi gli autovalori di C sono positivi perchè

$$\mathbf{d}^\top C \mathbf{d} \geq 0 \quad \text{per ogni } \mathbf{d} \in \mathbb{R}^2$$

Interpretazione geometrica di C

$$C = V \Lambda V^\top = \begin{pmatrix} \mathbf{v}_{max} \mathbf{v}_{min} \end{pmatrix} \begin{pmatrix} \lambda_{max} & 0 \\ 0 & \lambda_{min} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{max}^\top \\ \mathbf{v}_{min}^\top \end{pmatrix}$$



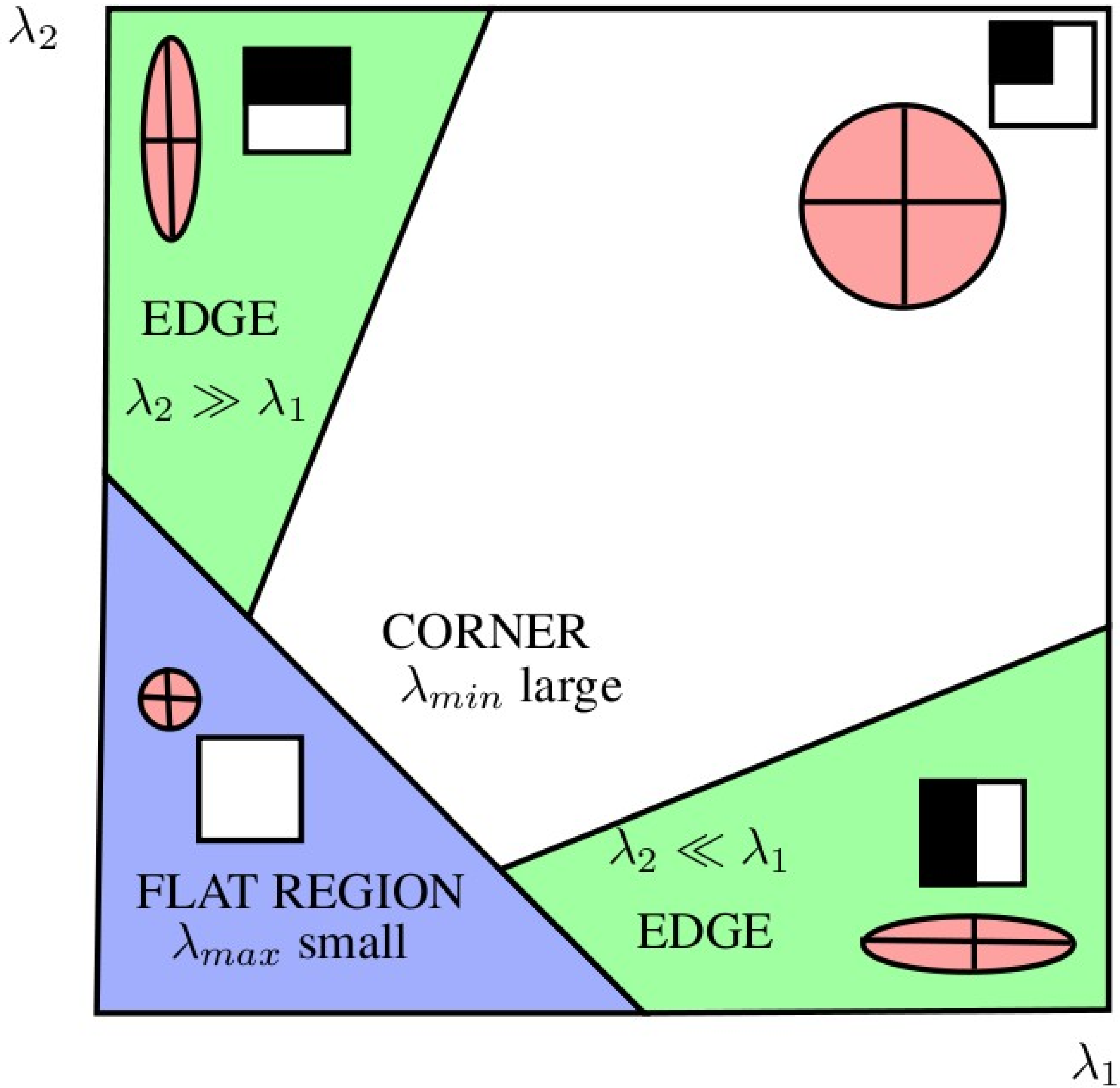
- $\mathbf{v}_{min}$ e $\mathbf{v}_{max}$ sono gli autovettori di C relativi all'autovalore minimo λ_{min} e massimo λ_{max}.
- Assumendo l'ellisse centrata sull'origine abbiamo che ogni punto del perimetro soddisfa

$$\mathbf{x}^\top C^{-1} \mathbf{x} = 1$$

- I vettori relativi agli assi principali sono

$$\sqrt{\lambda_{\max}}\mathbf{v}_{\max} \qquad \sqrt{\lambda_{\min}}\mathbf{v}_{\min}$$

Classificazione basata sull'ellisse



Rilevamento di punti salienti

- Abbiamo un punto saliente quando abbiamo un alto contenuto informativo in ogni direzione, ovvero quando l'autovalore minimo di C supera una certa soglia significativa.
- Un modo approssimato ma veloce per capire se abbiamo un punto saliente consiste nel calcolare la seguente misura:

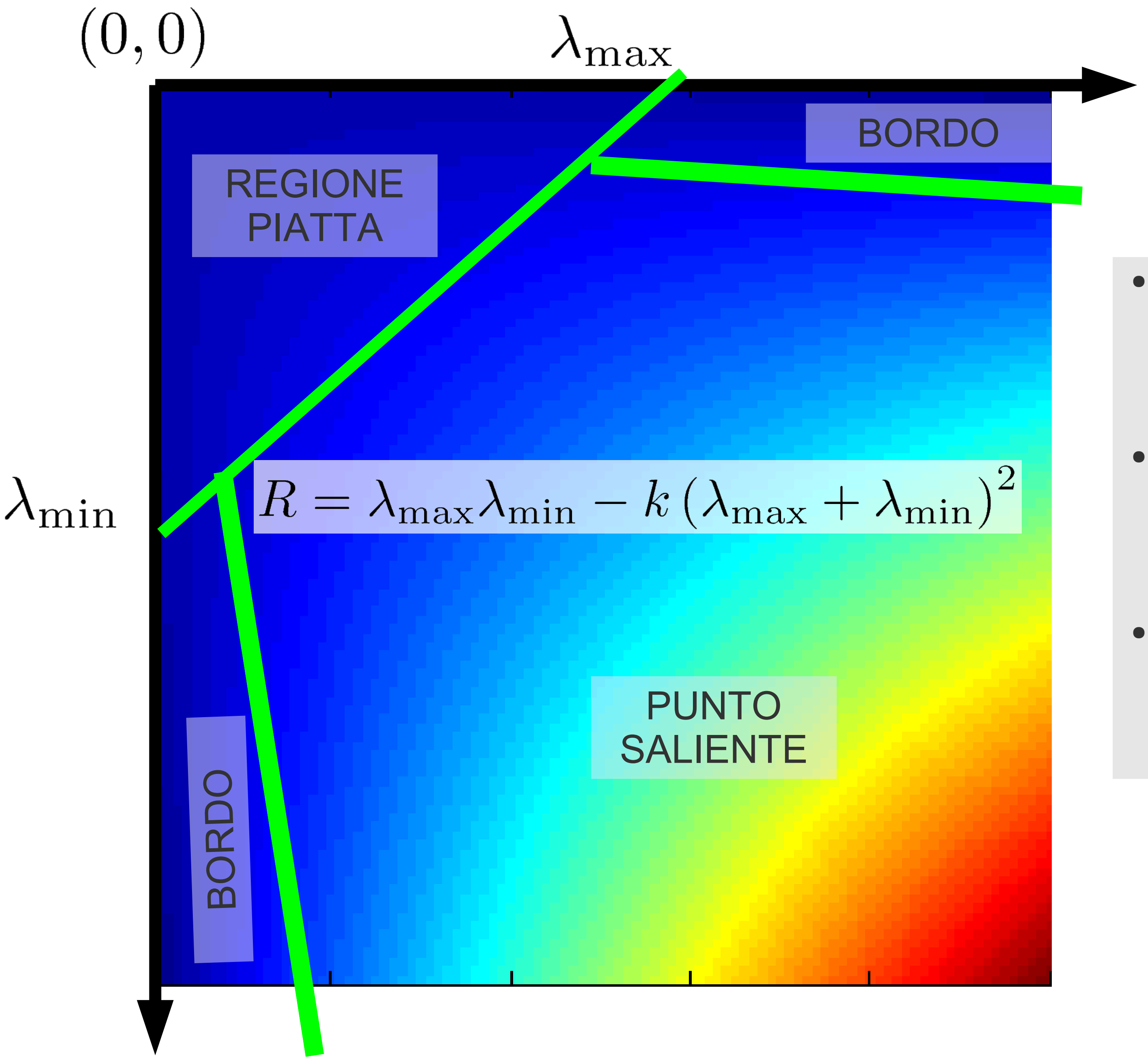
$$R = \det(C) - k\left[\mathrm{trace}(C)\right]^2$$

dove det(C) è il determinante di C, trace(C) è la traccia di C e k è una costante da determinare empiricamente (in genere k=0.04-0.06)

$$\det(C) = c_{11}c_{22} - c_{12}c_{21} = \lambda_{\min}\lambda_{\max}$$
$$\mathrm{trace}(C) = c_{11} + c_{22} = \lambda_{\min} + \lambda_{\max}$$

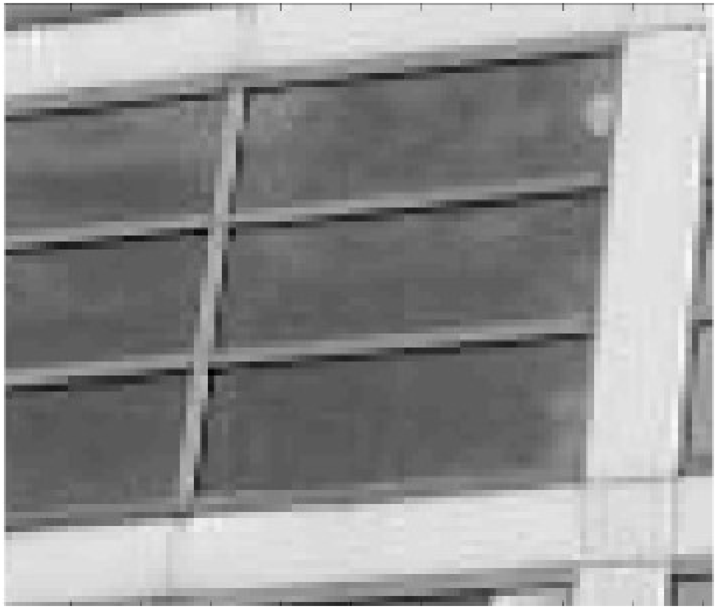
$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

Classificazione in base a R

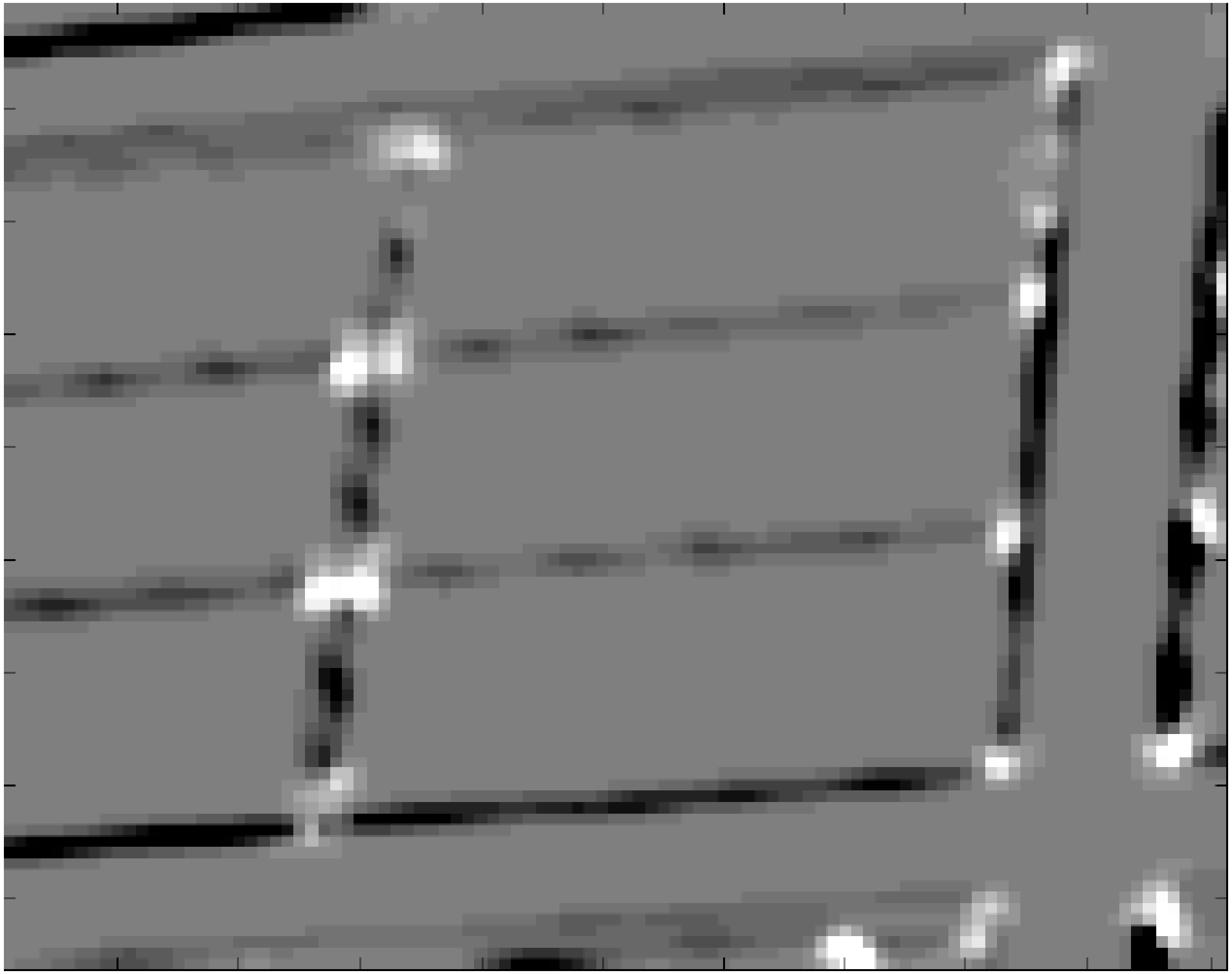


- R ha positivi valori grandi se il punto è saliente
- R è negativo di magnitudo elevato in presenza di un edge
- R ha magnitudo basso in presenza di regioni piatte.

Esempio

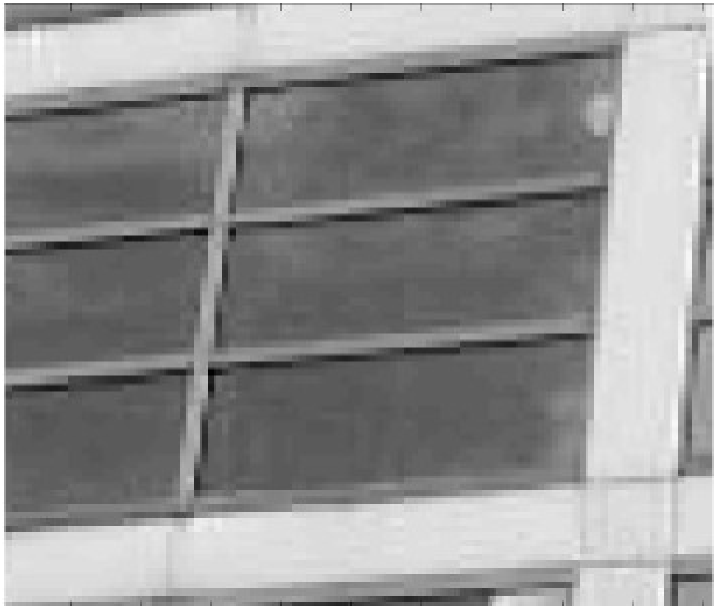


Valori di R per ogni punto dell'immagine

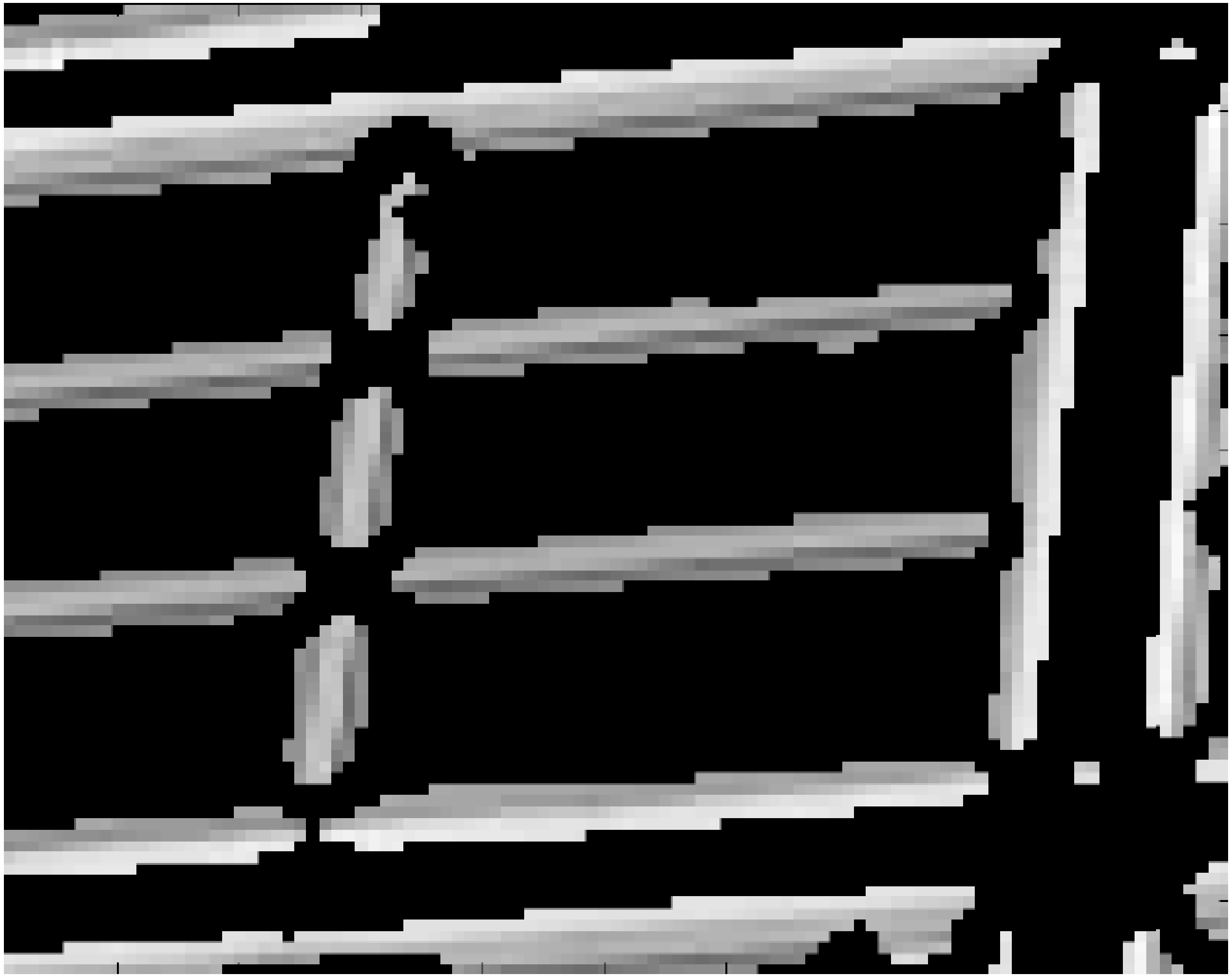


- Derivate calcolate con filtri di Sobel.
- Funzione peso w gaussiana con σ=1

Esempio

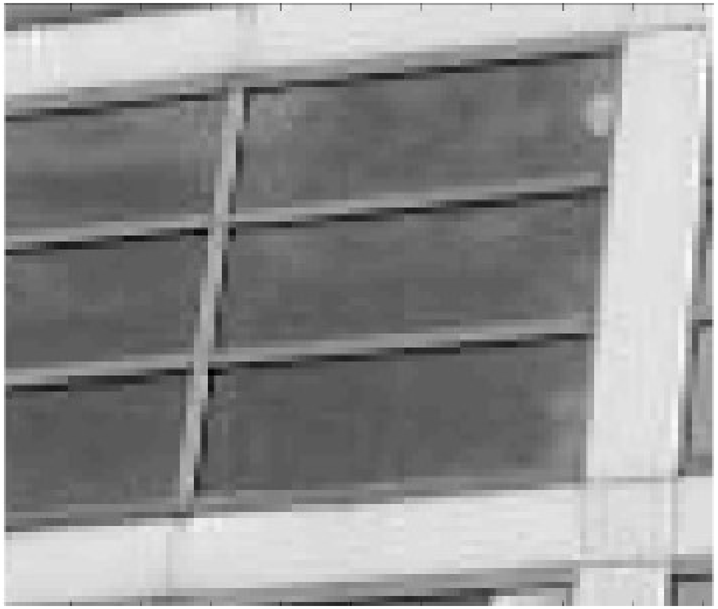


R<10000

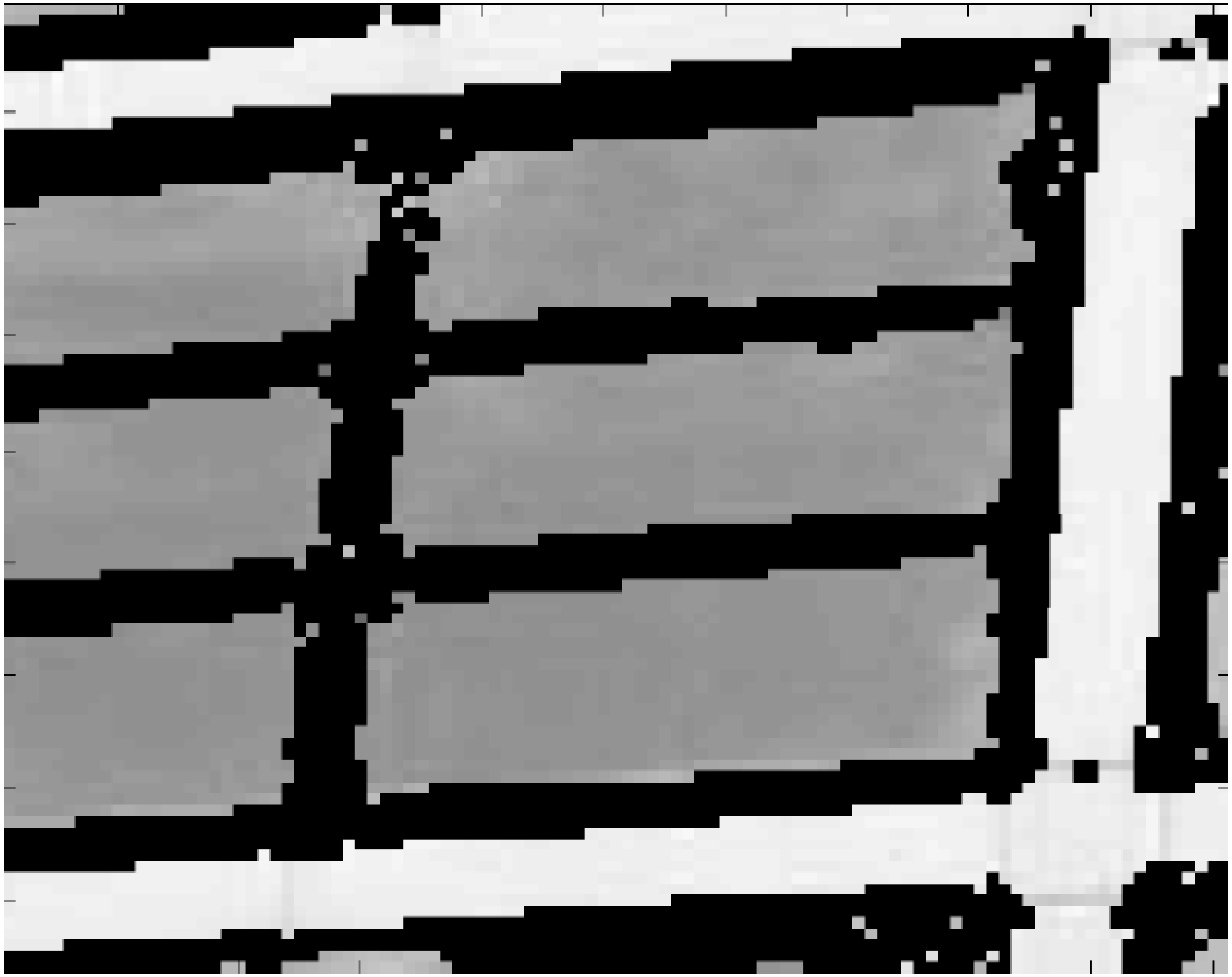


BORDI

Esempio

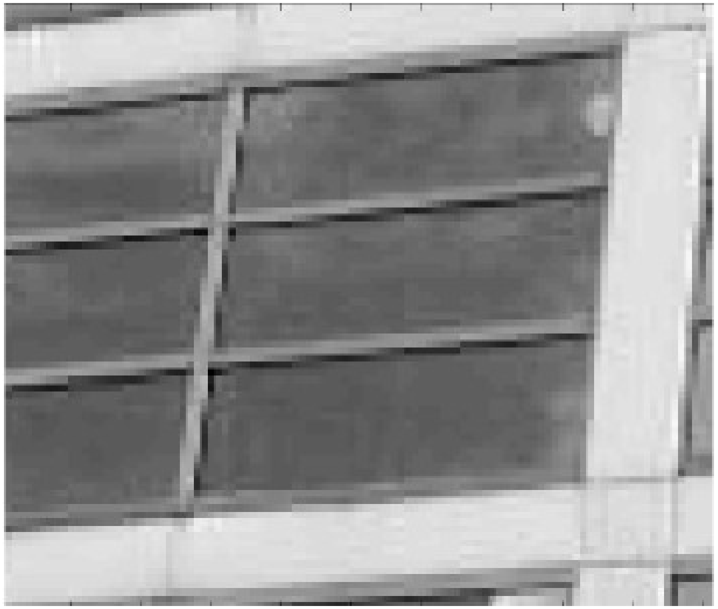


|R|<10000

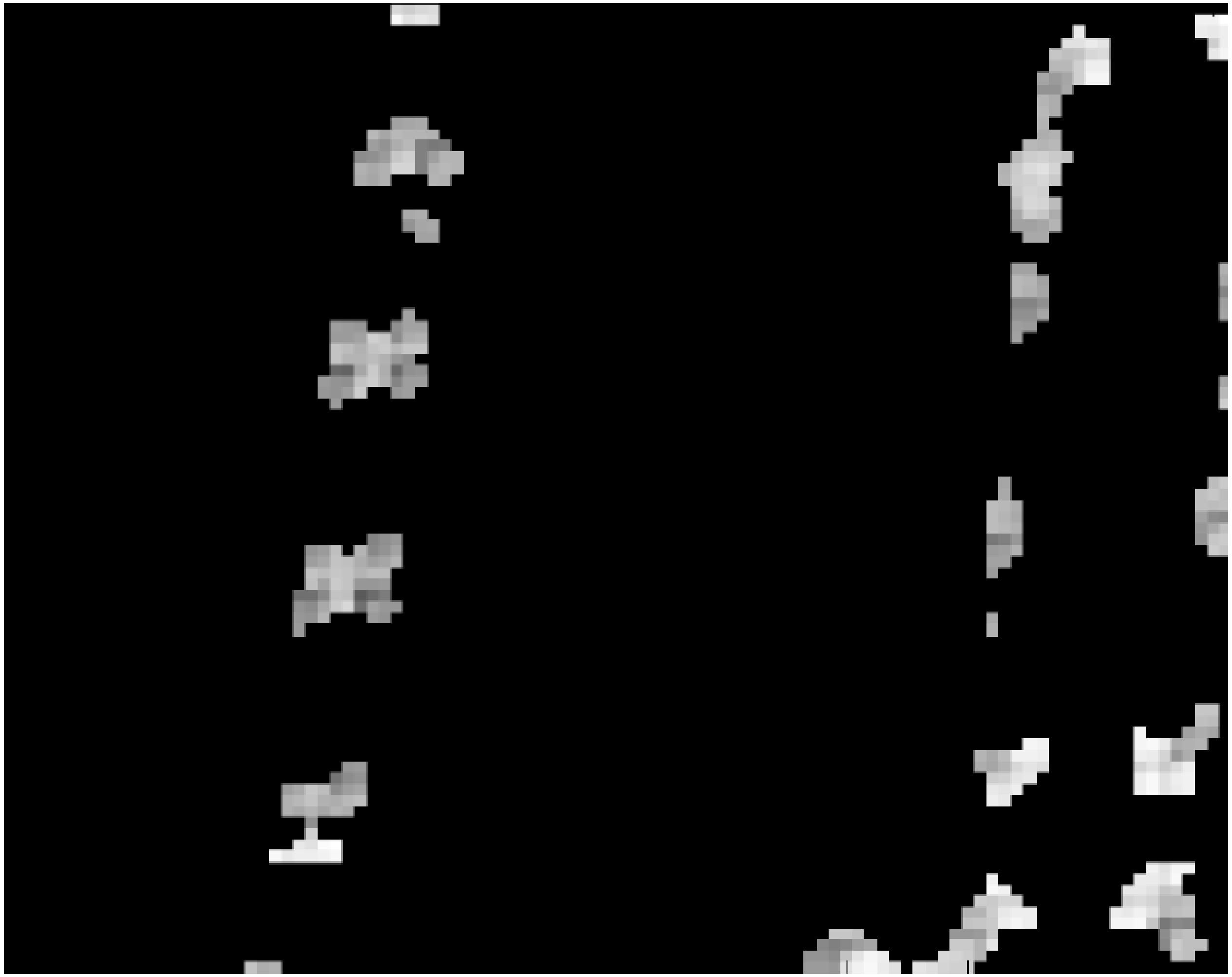


REGIONI PIATTE

Esempio

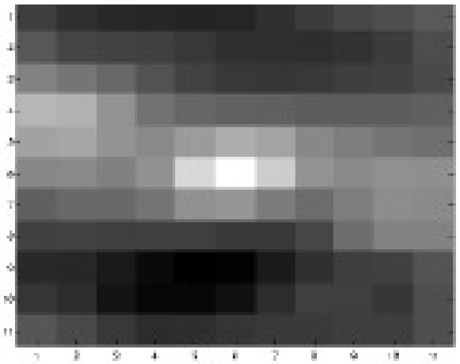


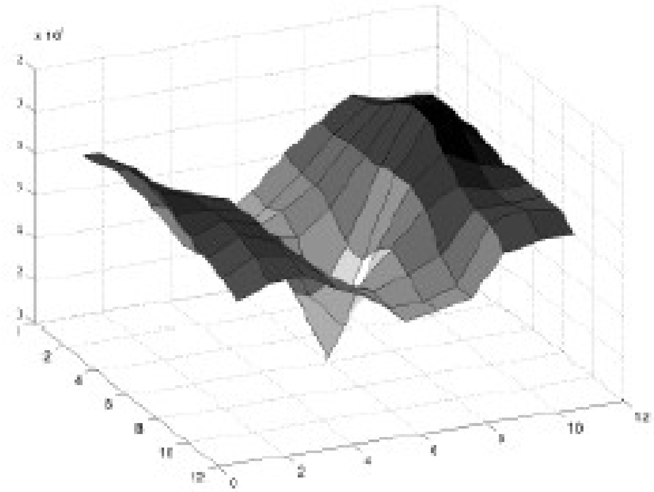
R>10000



PUNTI SALIENTI

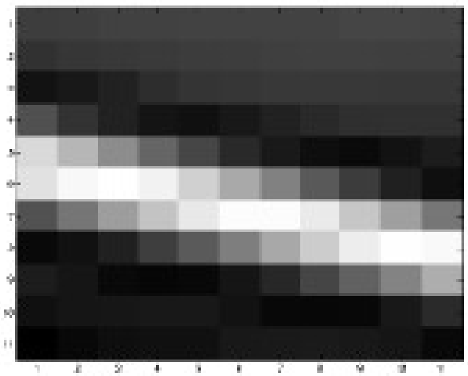
Esempio

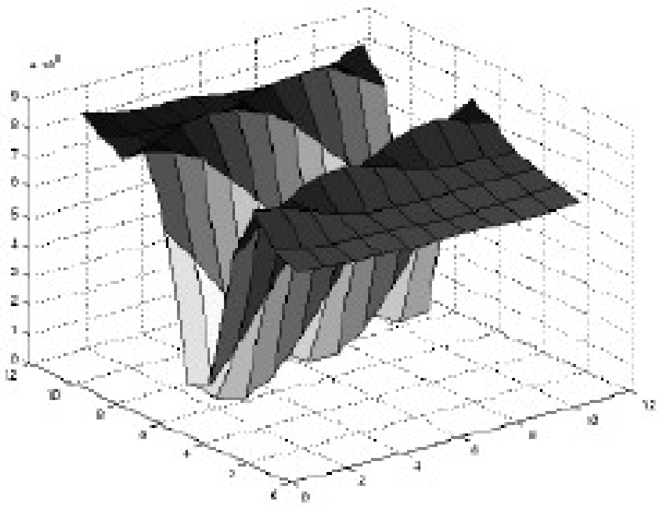




λ_1 and λ_2 are large

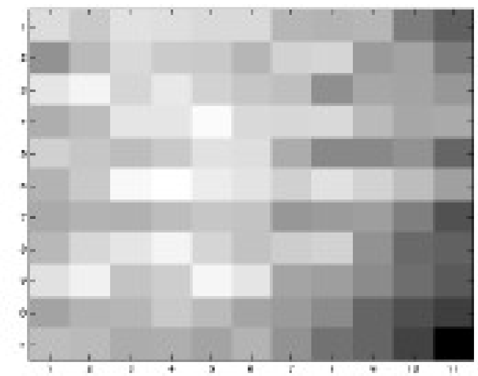
Esempio

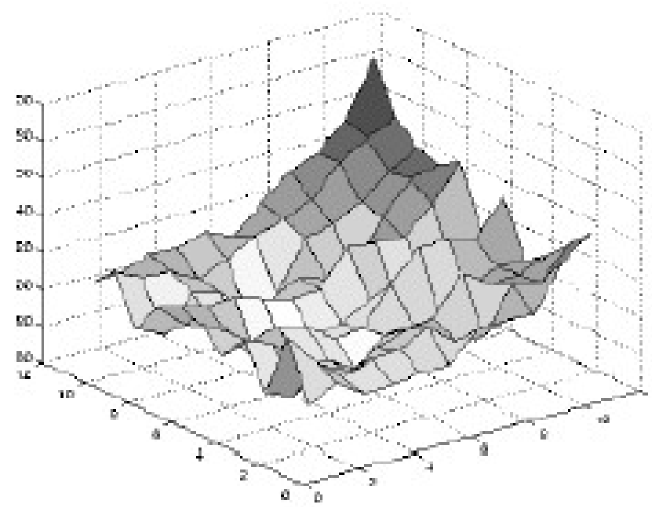




large λ_1, small λ_2

Esempio





small λ_1, small λ_2

Algoritmo di Harris

1. Calcolare le derivate x e y dell'immagine

$$I_x = G_\sigma^x * I \qquad I_y = G_\sigma^y * I$$

2. Calcolare prodotti delle derivate:

$$I_{xx}[i,j] = I_x[i,j]^2 \qquad I_{yy}[i,j] = I_y[i,j]^2$$

$$I_{xy}[i,j] = I_x[i,j] I_y[i,j]$$

3. Calcolare le somme pesate dei prodotti delle derivate per ogni pixel:

$$S_{xx} = G_\sigma * I_{xx} \qquad S_{xy} = G_\sigma * I_{xy} \qquad S_{yy} = G_\sigma * I_{yy}$$

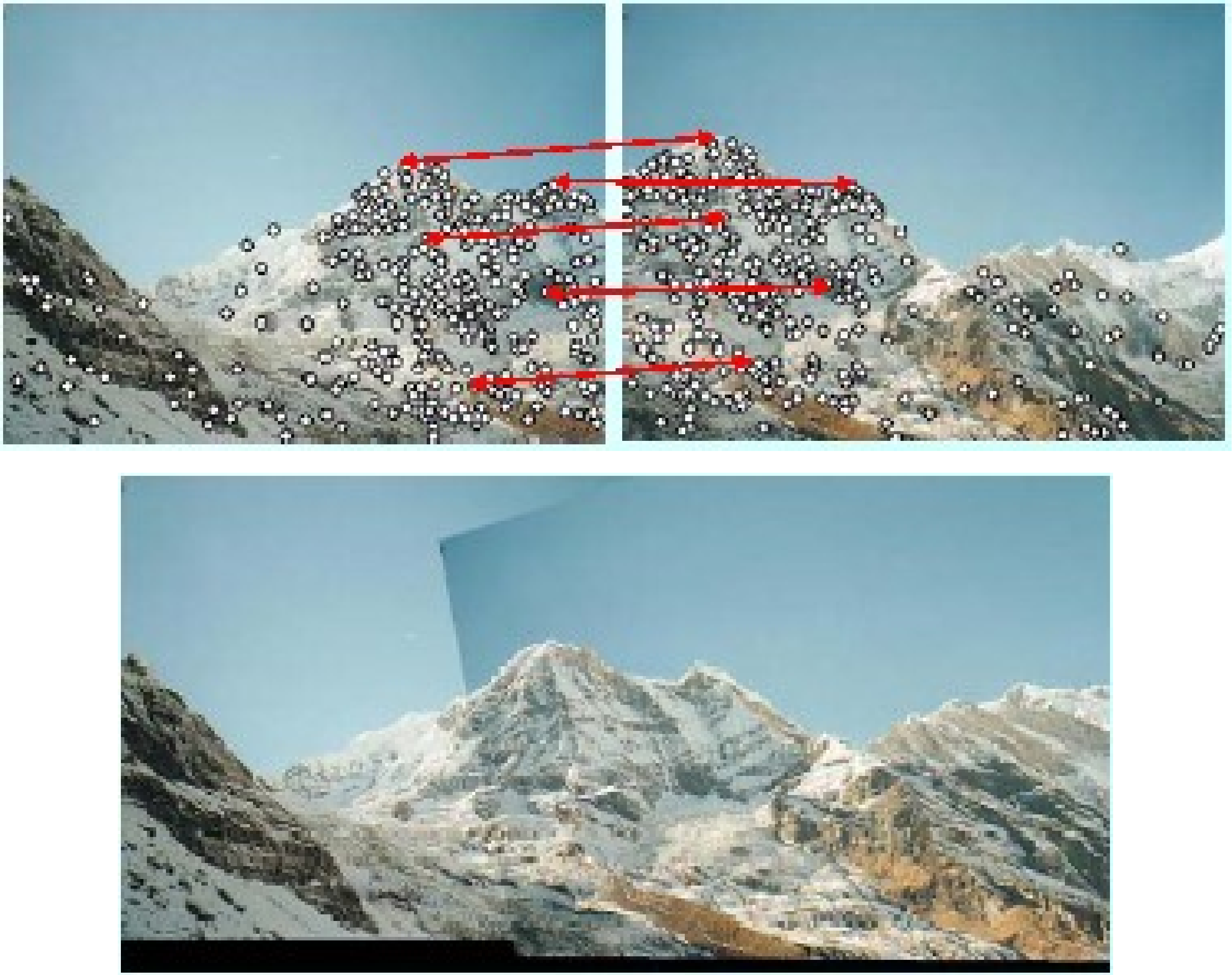
4. Calcolare R per ogni pixel

$$R[i.j] = S_{xx}[i,j] S_{yy}[i,j] - S_{xy}[i,j]^2 - k(S_{xx}[i.j] + S_{yy}[i,j])^2$$

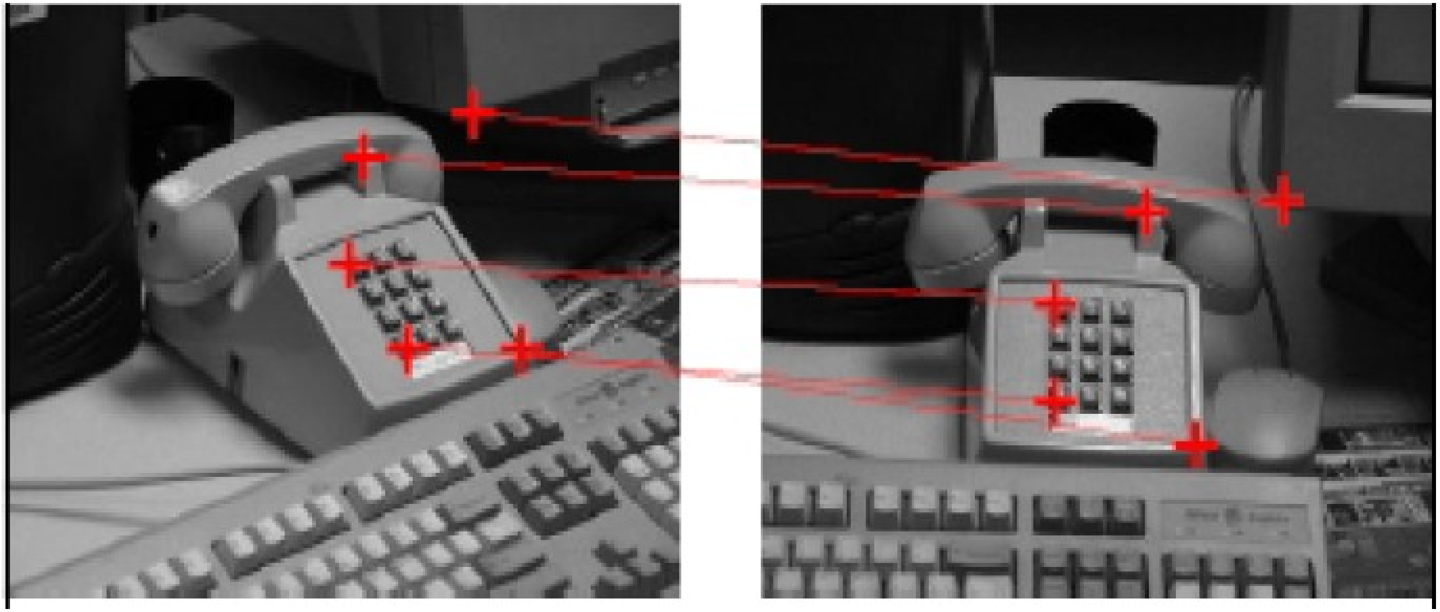
5. Infine sogliare opportunamente R e fare nonmaximal suppression.

Esempio

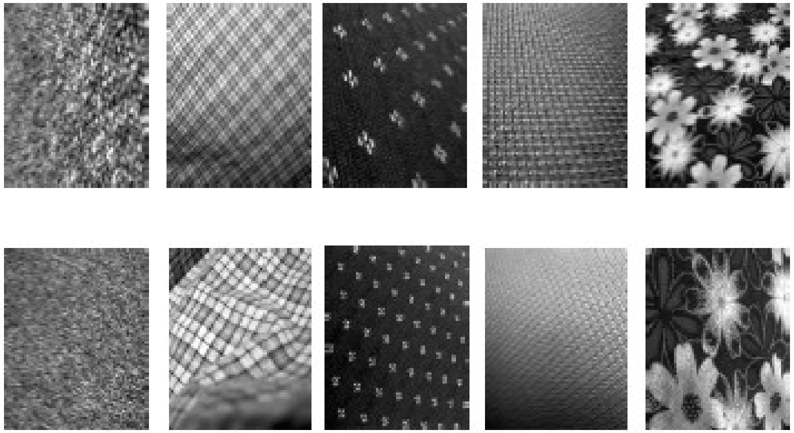
Esempio



Esempio



Esempio



texture recognition

car detection