

# Analisi di Immagini e Video (Computer Vision)

Giuseppe Manco

# Outline

- Preliminari
- Classificazione
  - Esempio: regressione logistica
- Quali features?

# Crediti

- Slides adattate da vari corsi e libri
  - Analisi di Immagini (F. Angiulli) – Unical
  - Intro to Computer Vision (J. Tompkin) – CS Brown Edu
  - Computer Vision (I. Gkioulekas) - CS CMU Edu
  - Computational Visual Recognition (V. Ordonez), CS Virginia Edu
  - Pattern Recognition and Machine Learning (C. Bishop, 2005)
  - Deep Learning (Bengio, Courville, Goodfellow, 2017)

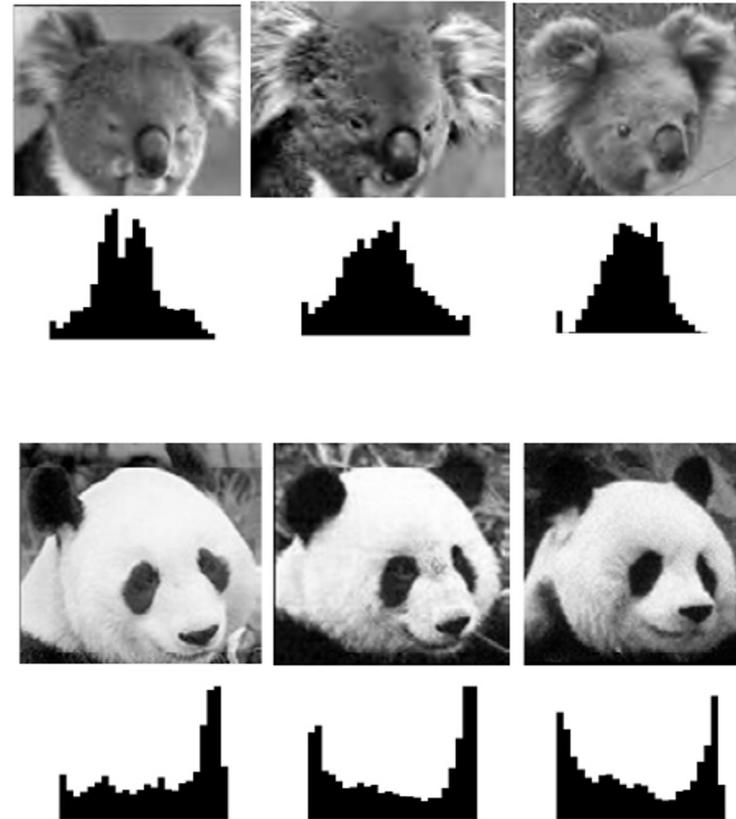
Quali features?

# Quali features?

- Global features
  - Colori
  - Istogrammi
- Local features
  - Edges
  - Corners
  - Histogram of Oriented Gradients (HOG)
  - Haar Cascades
  - Scale-Invariant Feature Transform (SIFT)
  - Speeded Up Robust Feature (SURF)
  - ...

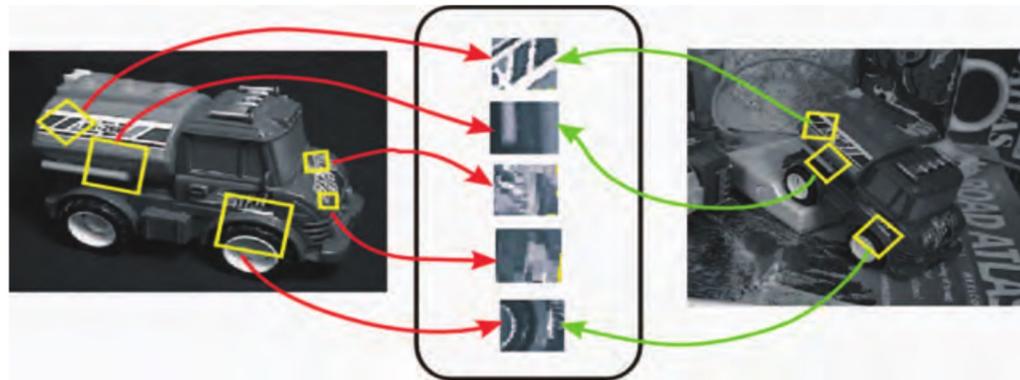
# Global Features

- Rappresentazione olistica dell'immagine
- Esempio: istogramma delle intensità dell'immagine
- Immagini simili hanno istogramma simile, ma in generale non è vero il viceversa
- Permettono di rappresentare la struttura globale dell'oggetto, ma non di gestire l'occlusione, il cambiamento di punto di vista e le altre variabilità



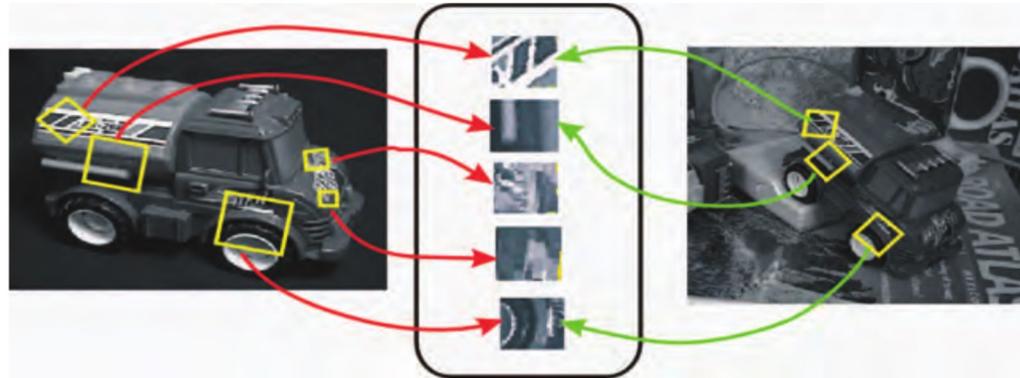
# Local Features

- Le local features rappresentano un *insieme sparso* di misurazioni locali che catturano l'essenza delle strutture all'interno dell'immagine
- Diverse proprietà richieste: precise, distintive, invarianti, numerose



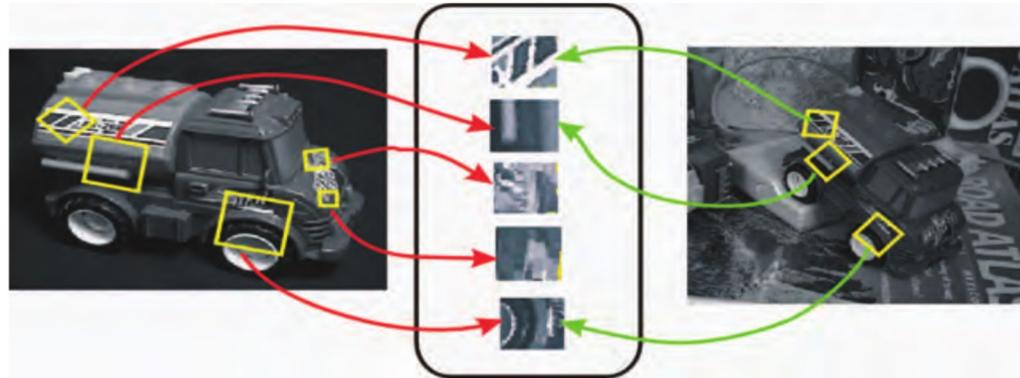
# Local Features

- Il processo di estrazione dev'essere *ripetibile* e *preciso*, in modo che le stesse features siano restituite da due immagini che riguardano lo stesso oggetto



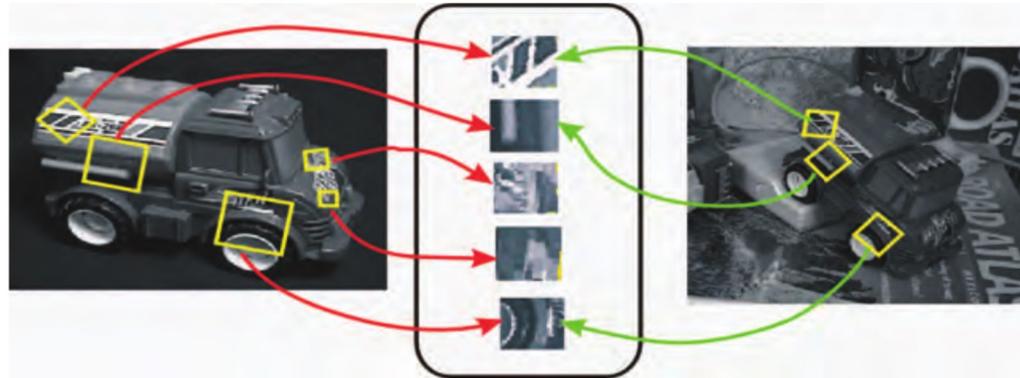
# Local Features

- Le features devono essere *distintive*, in modo che le diverse strutture presenti all'interno dell'immagine possano essere discriminate



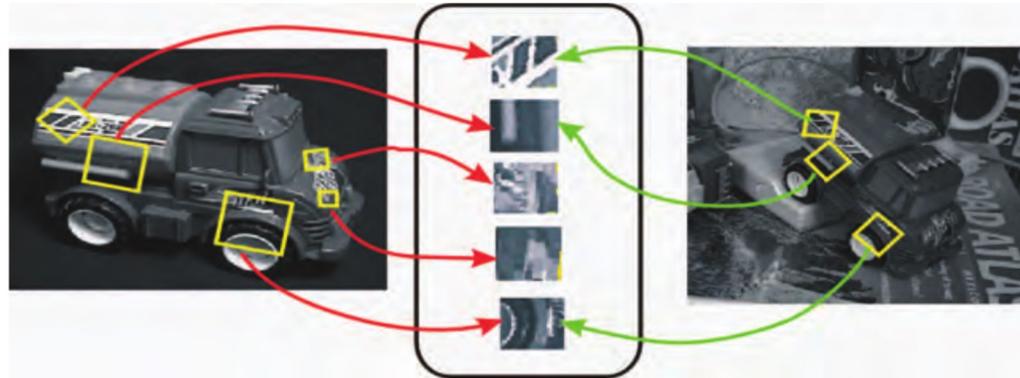
# Local Features

- Le local features dovrebbero essere *invarianti* rispetto a diverse trasformazioni applicate all'immagine
  - traslazioni, rotazioni, scalatura, ...

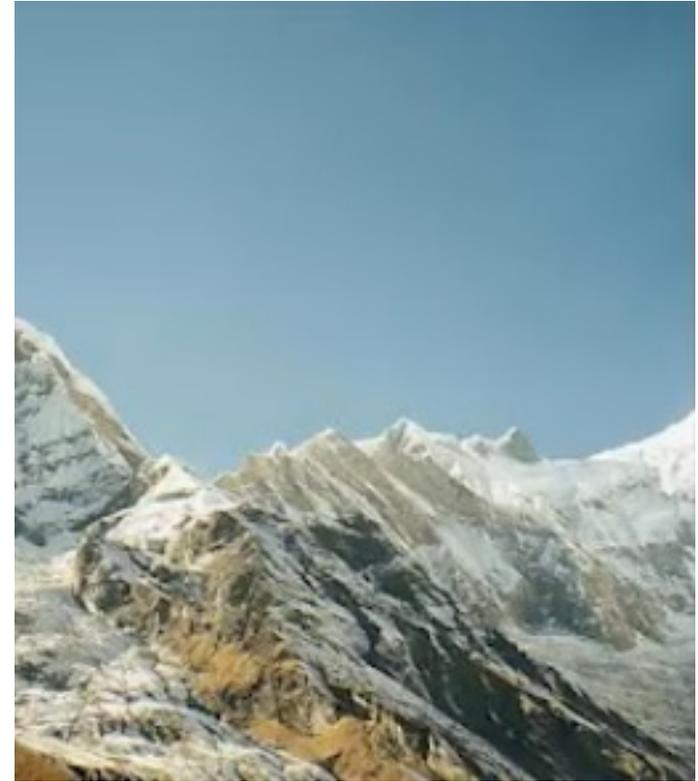


# Local Features

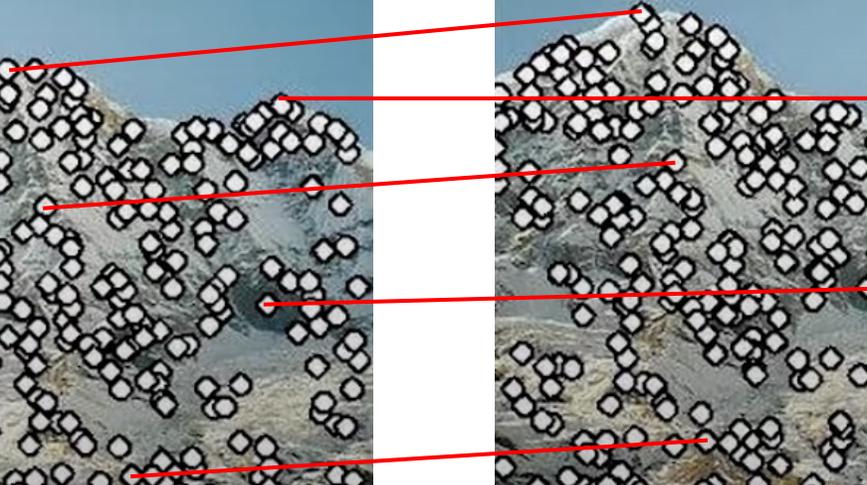
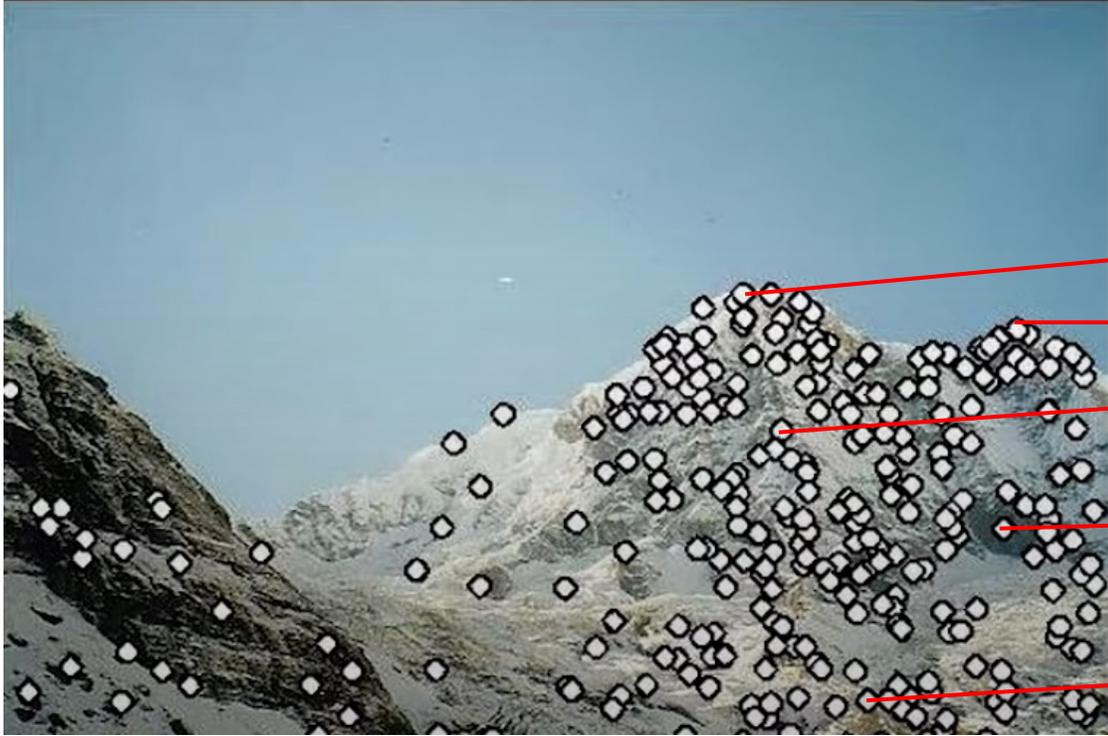
- Le features estratte devono essere abbastanza *numerose* da assicurare una buona copertura dell'immagine
  - ad esempio, per permettere il riconoscimento di oggetti anche parzialmente occlusi





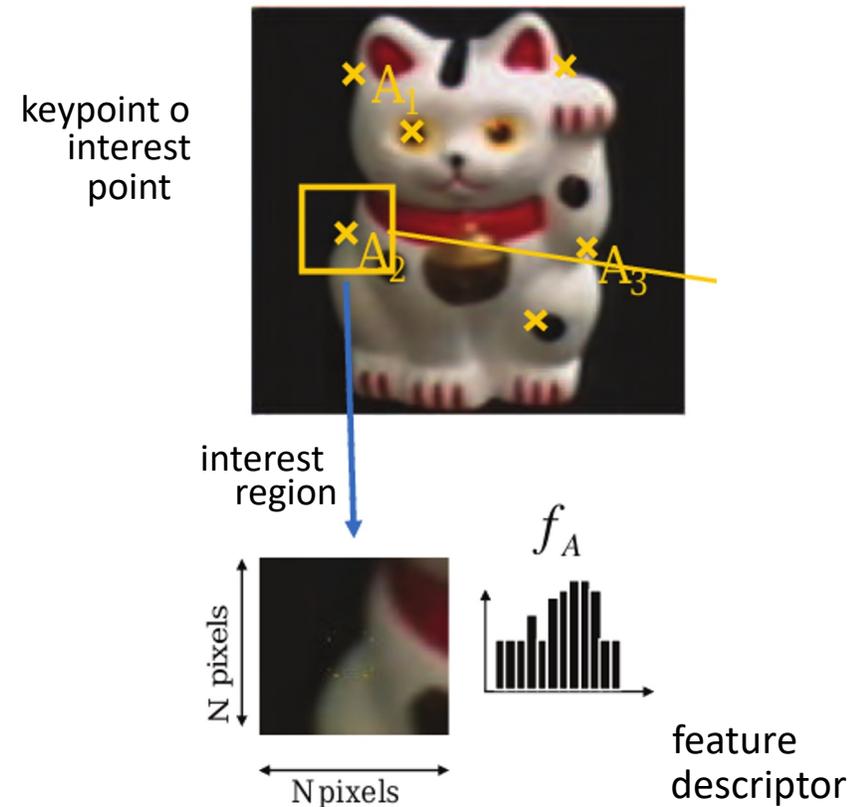






# Feature Extraction

- Passi del processo di *Feature Extraction*:
  1. Individua un insieme di *keypoints* o *interest points* (*Keypoint Localization* o *Feature Detection*)
  2. Determina una *interest region* intorno ad ogni *keypoint*
  3. Estrai e *normalizza* il contenuto della regione e rappresentalo mediante un *feature descriptor*



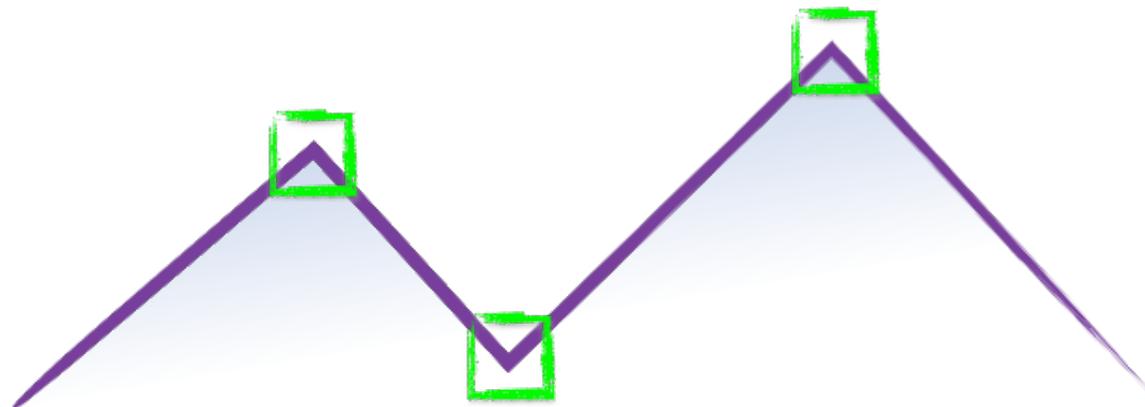
# Keypoints, features and feature descriptors

- Si estrae un insieme di features da ogni immagine del training set
  - Ogni feature ha una rappresentazione multidimensionale
- Si trovano i K raggruppamenti di tutte le features
  - Ad esempio, usando il K-Means
  - Ogni feature è associata ad un cluster
- Per ogni immagine, si calcola l'istogramma delle features sui K gruppi
  - Ogni feature contribuisce al cluster a cui è associata
- L'istogramma (normalizzato) rappresenta la rappresentazione K-dimensionale dell'immagine

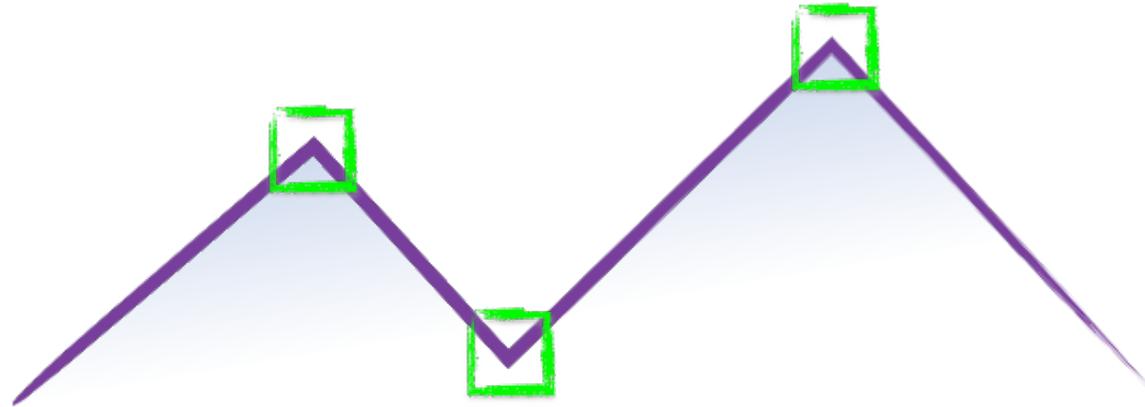
# Keypoint Localization

- I **keypoint** (o **interest point**) sono punti *distintivi* che possono essere localizzati anche in presenza di variazioni nell'immagine, quali traslazioni o rotazioni, cambiamenti di punti di vista e presenza di rumore
- Escludiamo dai punti candidati ad essere distintivi quelli per cui non si manifestano cambiamenti in qualche direzione:
  - Non possiamo determinare il moto dei punti appartenenti ad una regione uniforme
  - Per i punti appartenenti ad una linea, possiamo misurare solo il moto perpendicolarmente alla linea

Come identificare un angolo?



# Come identificare un angolo?

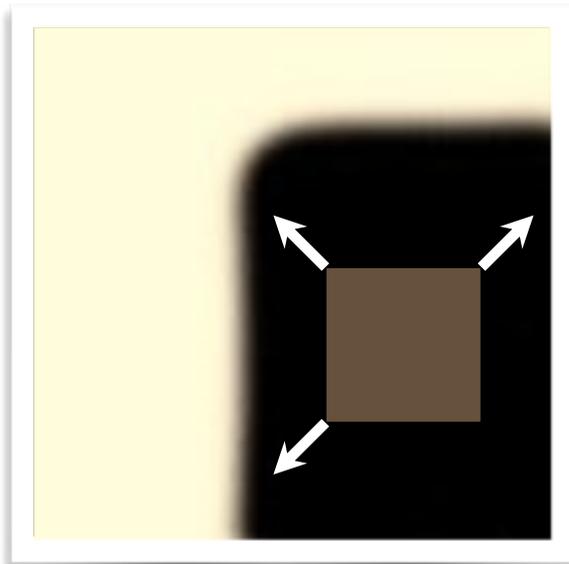


Si analizza una finestra

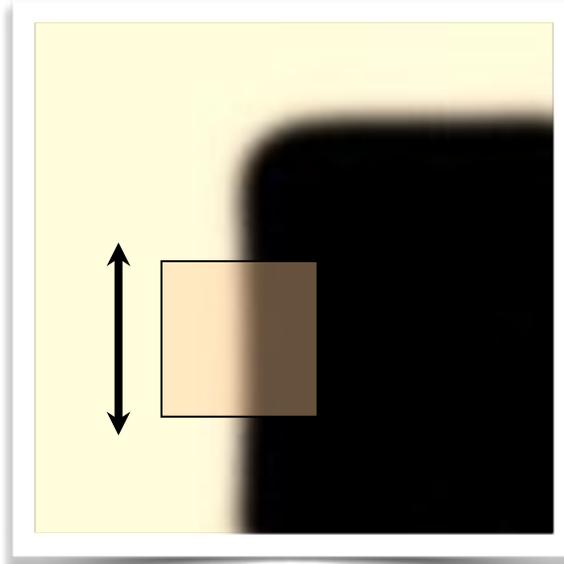
Muovendo la finestra si osserva un significativo cambiamento di intensità

Si analizza una finestra

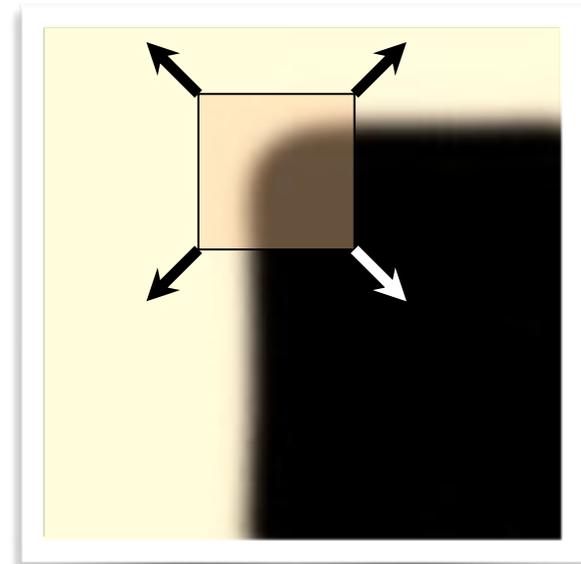
Muovendo la finestra si osserva un significativo cambio di intensità



“flat” nessun  
cambiamento

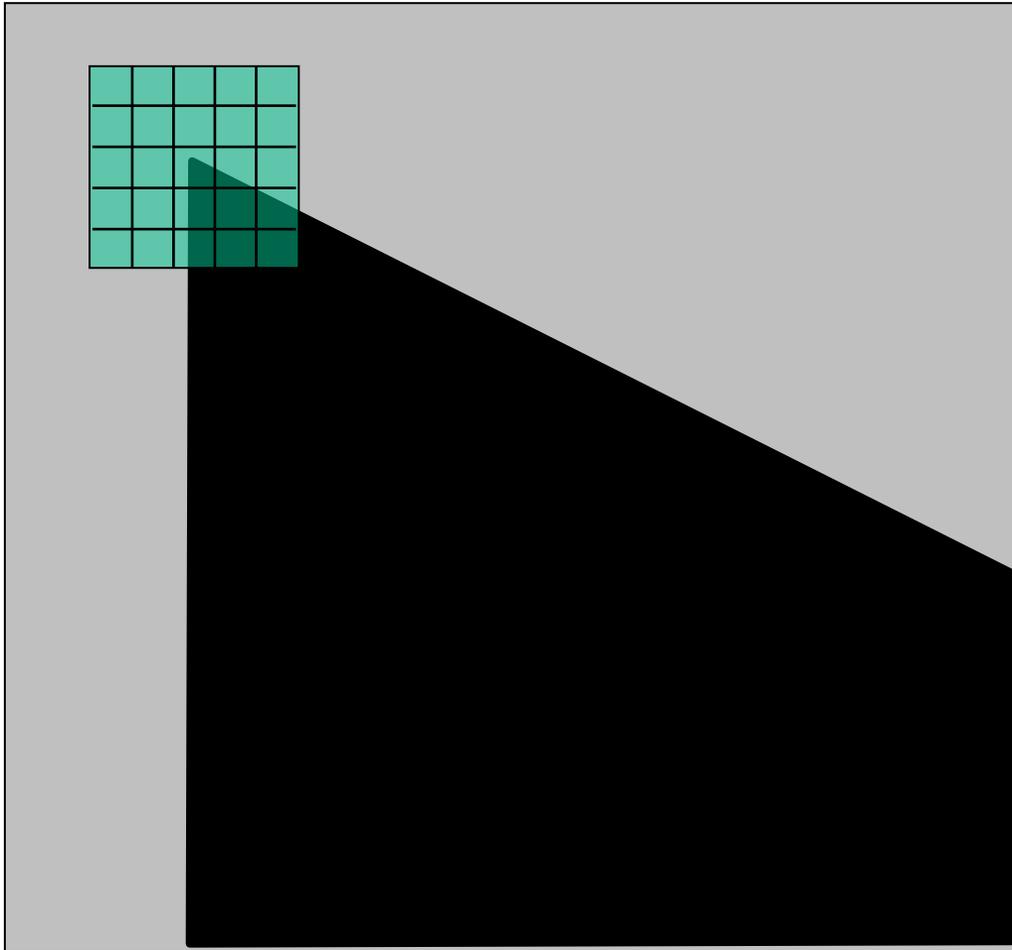


“edge”:  
cambiamento lungo una  
direzione

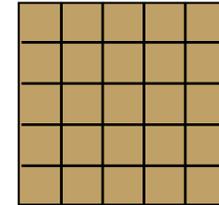


“corner”:  
Cambiamento lungo  
entrambe le direzioni

# Il gradiente su un punto

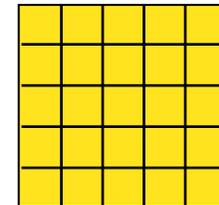


$$I_x = \frac{\partial I}{\partial x}$$

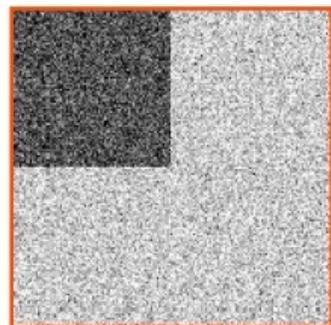
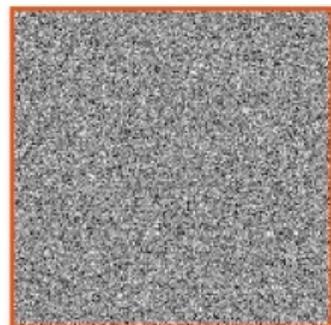
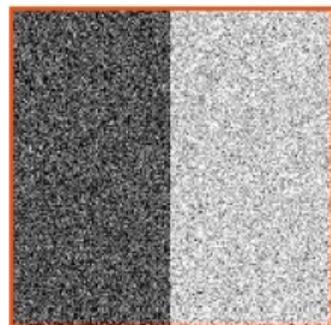


array of y gradients

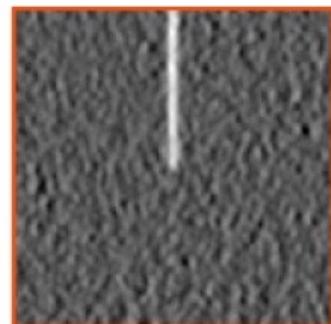
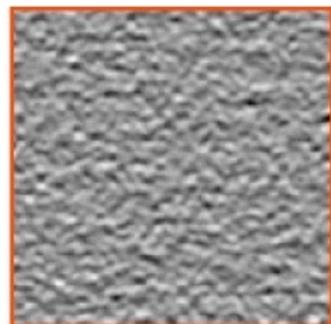
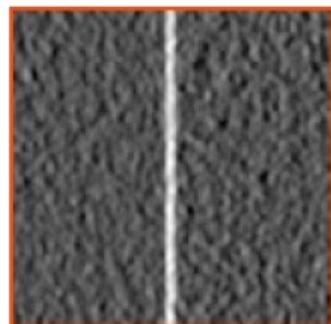
$$I_y = \frac{\partial I}{\partial y}$$



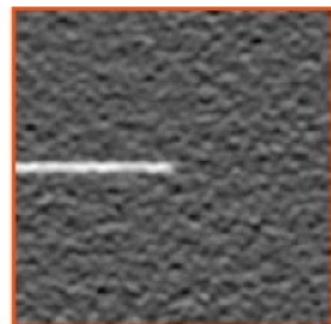
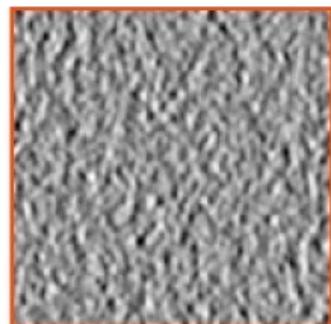
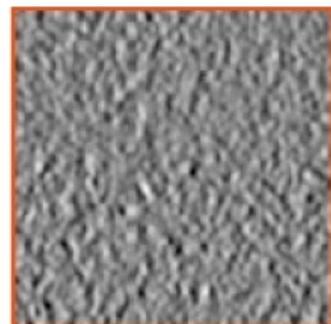
Immagine



Gradiente su X



Gradiente su Y



# La matematica del corner detection

Cambio di intensità intorno allo shift  $[u, v]$ :

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

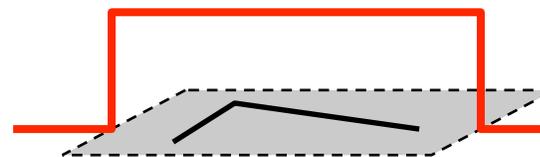
Error  
function

Window  
function

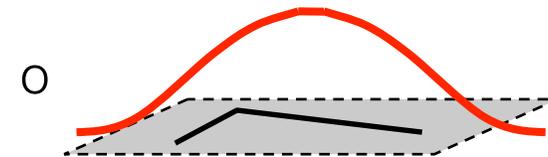
Shifted  
intensity

Intensity

Window function  $w(x, y) =$

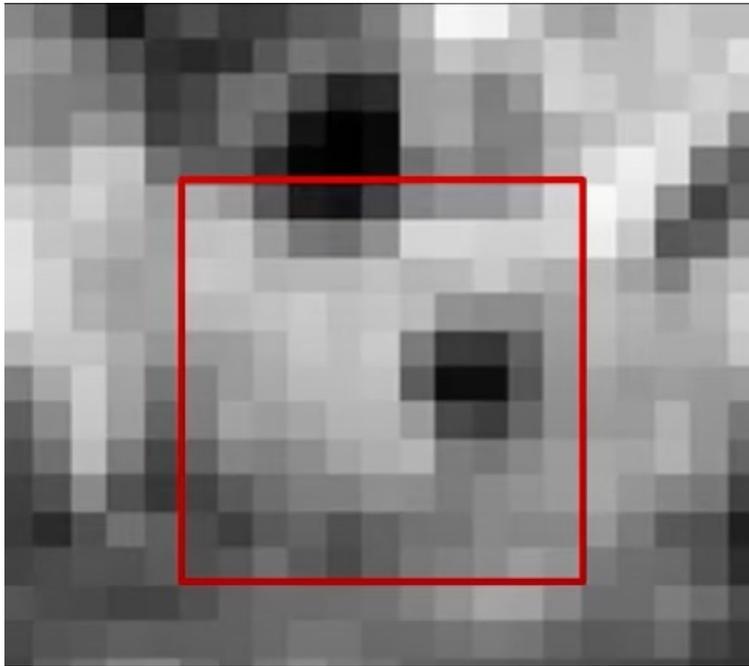


1 in window, 0 outside

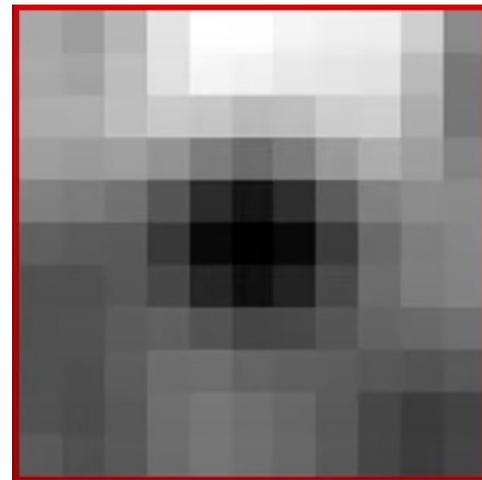


Gaussian

$I(x, y)$



$E(u, v)$



# La matematica del corner detection

Cambio di intensità intorno allo shift  $[u, v]$ :

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

- Come cambia  $E(u, v)$  per piccoli cambiamenti di  $u, v$ ?
  - Approssimazione di Taylor al secondo ordine!

$$F(\delta x) \approx F(0) + \delta x \frac{dF(0)}{dx} + \frac{1}{2} \delta x^2 \frac{d^2F(0)}{dx^2}$$

# La matematica del corner detection

Cambio di intensità intorno allo shift  $[u, v]$ :

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

$$F(\delta x) \approx F(0) + \delta x \frac{dF(0)}{dx} + \frac{1}{2} \delta x^2 \frac{d^2F(0)}{dx^2}$$

$$E(u, v) \approx E(0,0) + [u \quad v] \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} [u \quad v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{vu}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

# La matematica del corner detection

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

$$E(u, v) \approx E(0,0) + [u \quad v] \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} [u \quad v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{vu}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E_u(0,0) = E(0,0) = E_v(0,0) = 0$$

$$E_{uu}(0,0) = \sum_{xy} 2w(x, y) I_x(x, y) I_x(x, y)$$

$$E_{uv}(0,0) = \sum_{xy} 2w(x, y) I_x(x, y) I_y(x, y)$$

# La matematica del corner detection

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

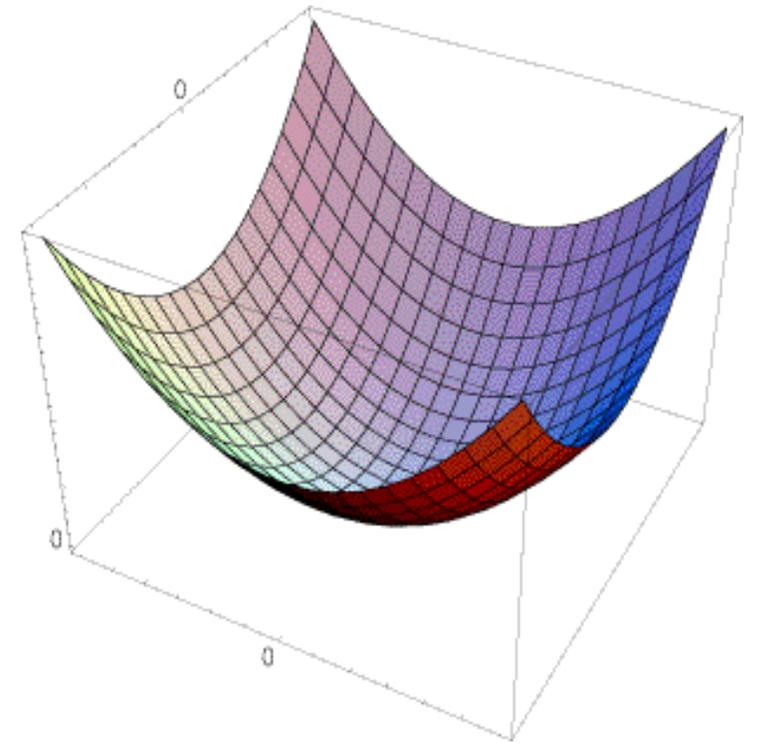
$$E(u, v) \approx \frac{1}{2} \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} \sum_{xy} 2w(x, y) I_x^2(x, y) & \sum_{xy} 2w(x, y) I_x(x, y) I_y(x, y) \\ \sum_{xy} 2w(x, y) I_x(x, y) I_y(x, y) & \sum_{xy} 2w(x, y) I_y^2(x, y) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

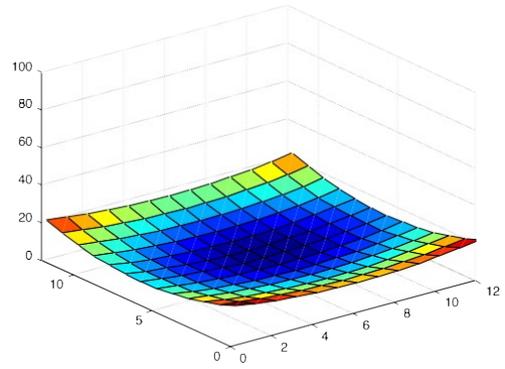
# La matematica del corner detection

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

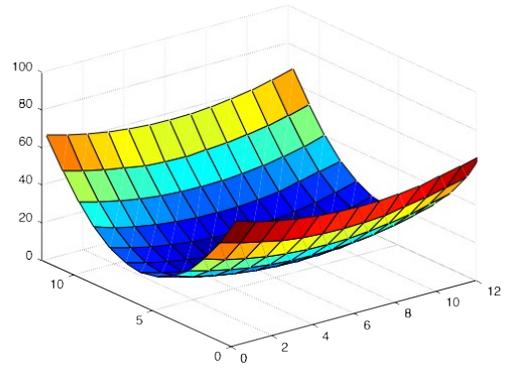
$$E(u, v) \approx [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_{xy} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

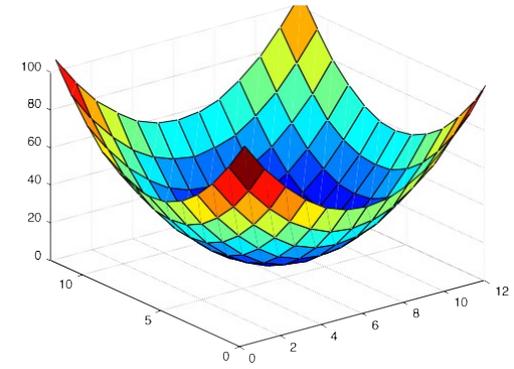




flat



edge  
'line'



corner  
'dot'

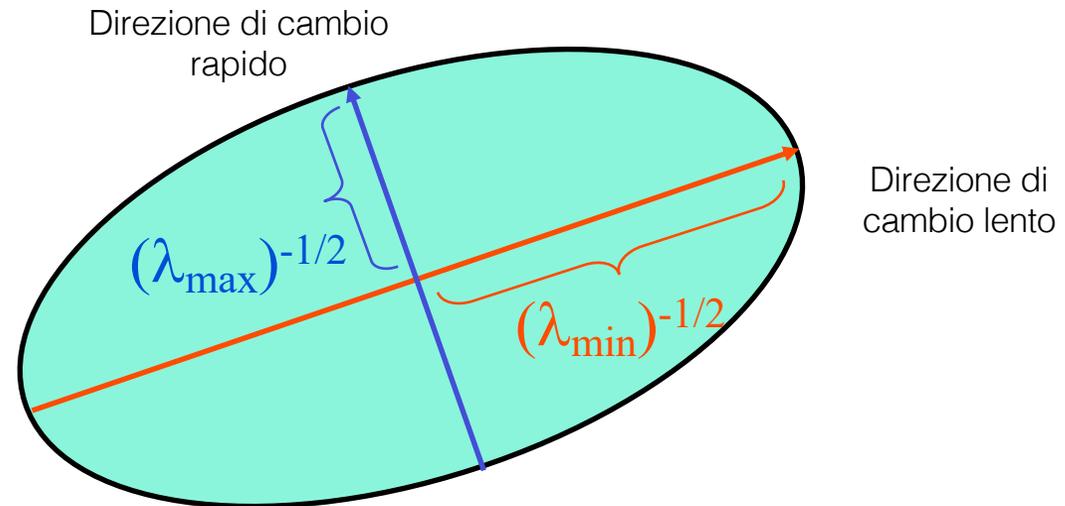
# Visualizzazione

Poiché  $M$  è simmetrica,

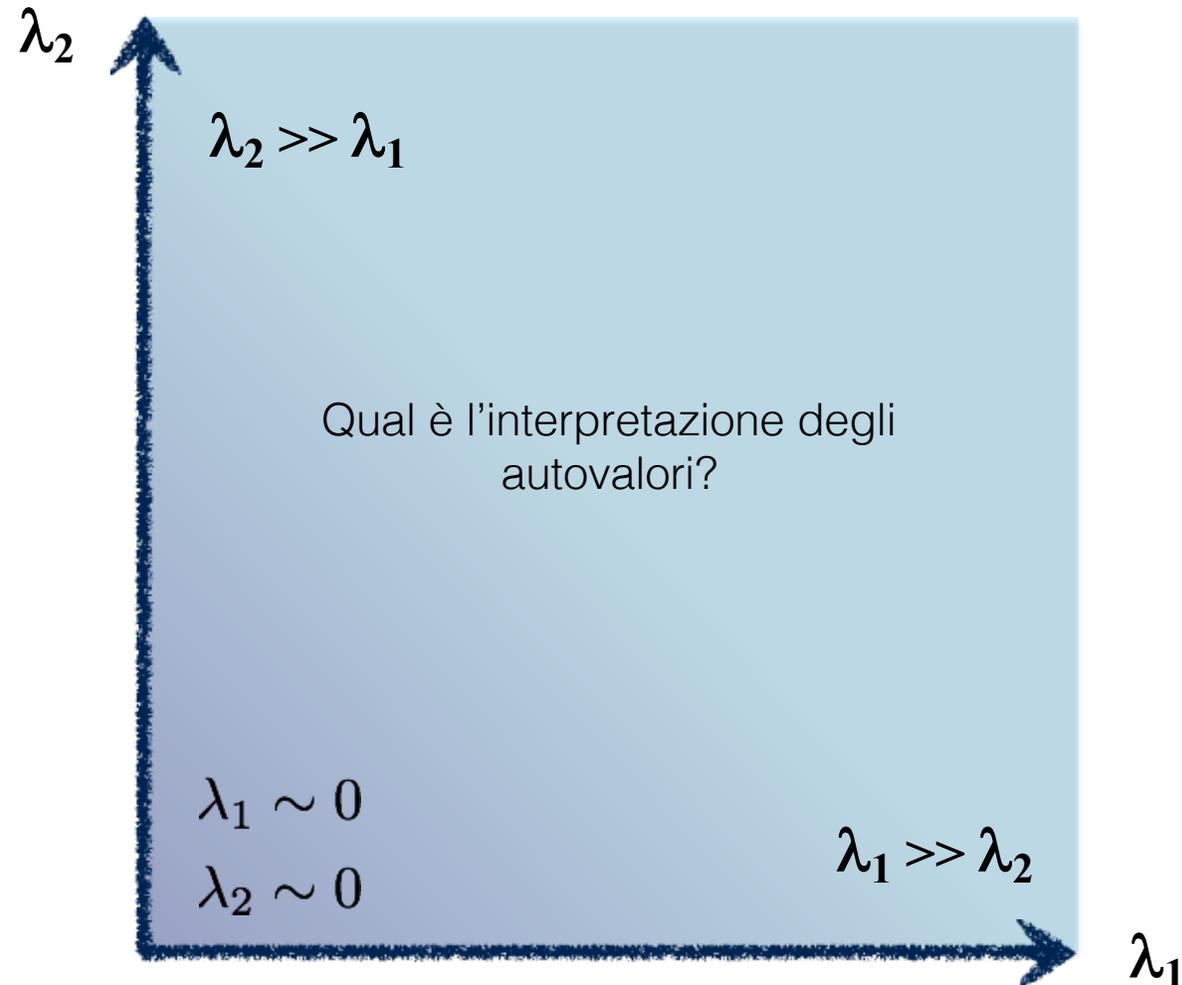
$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Equazione dell'ellisse

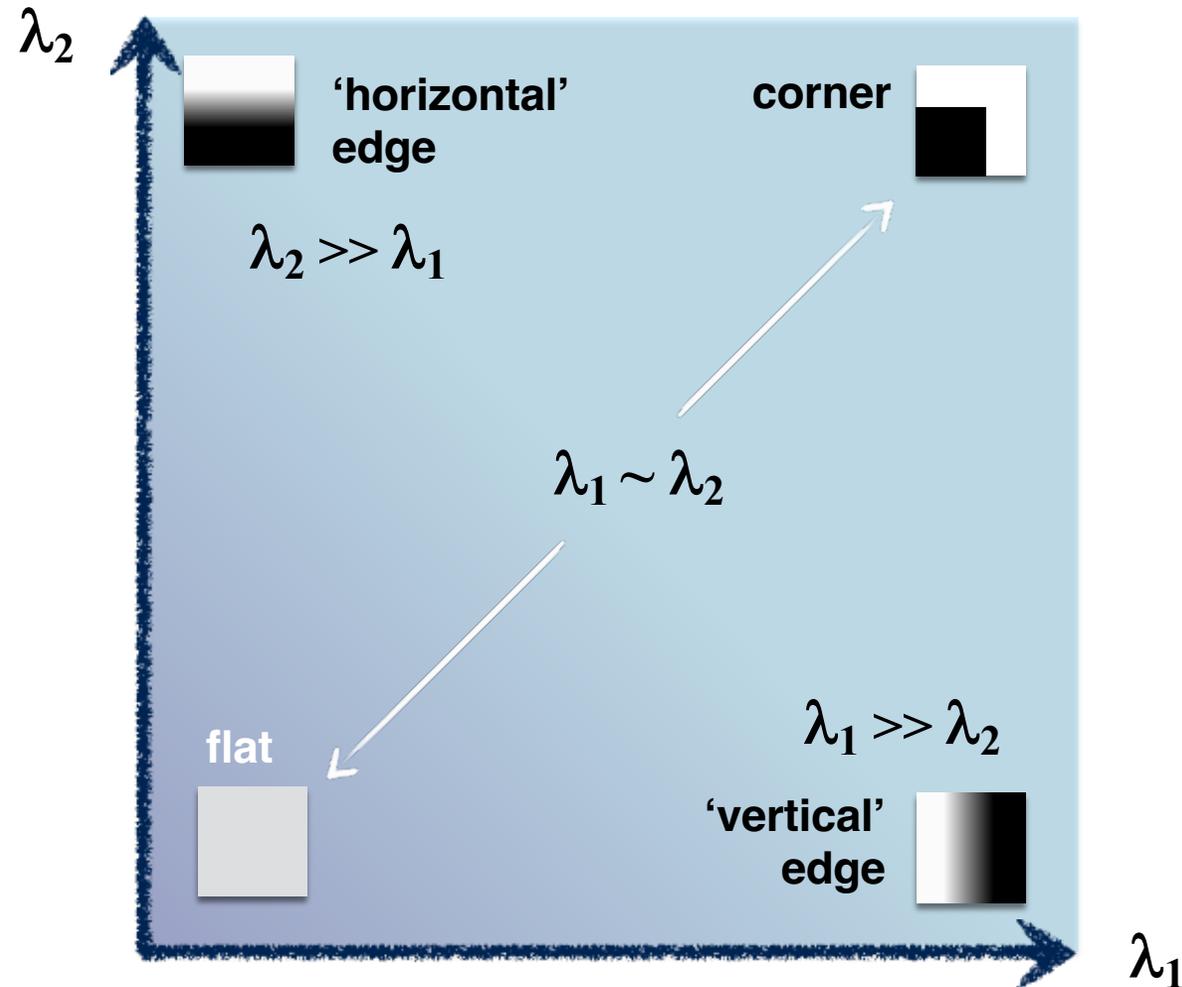
$$[u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$



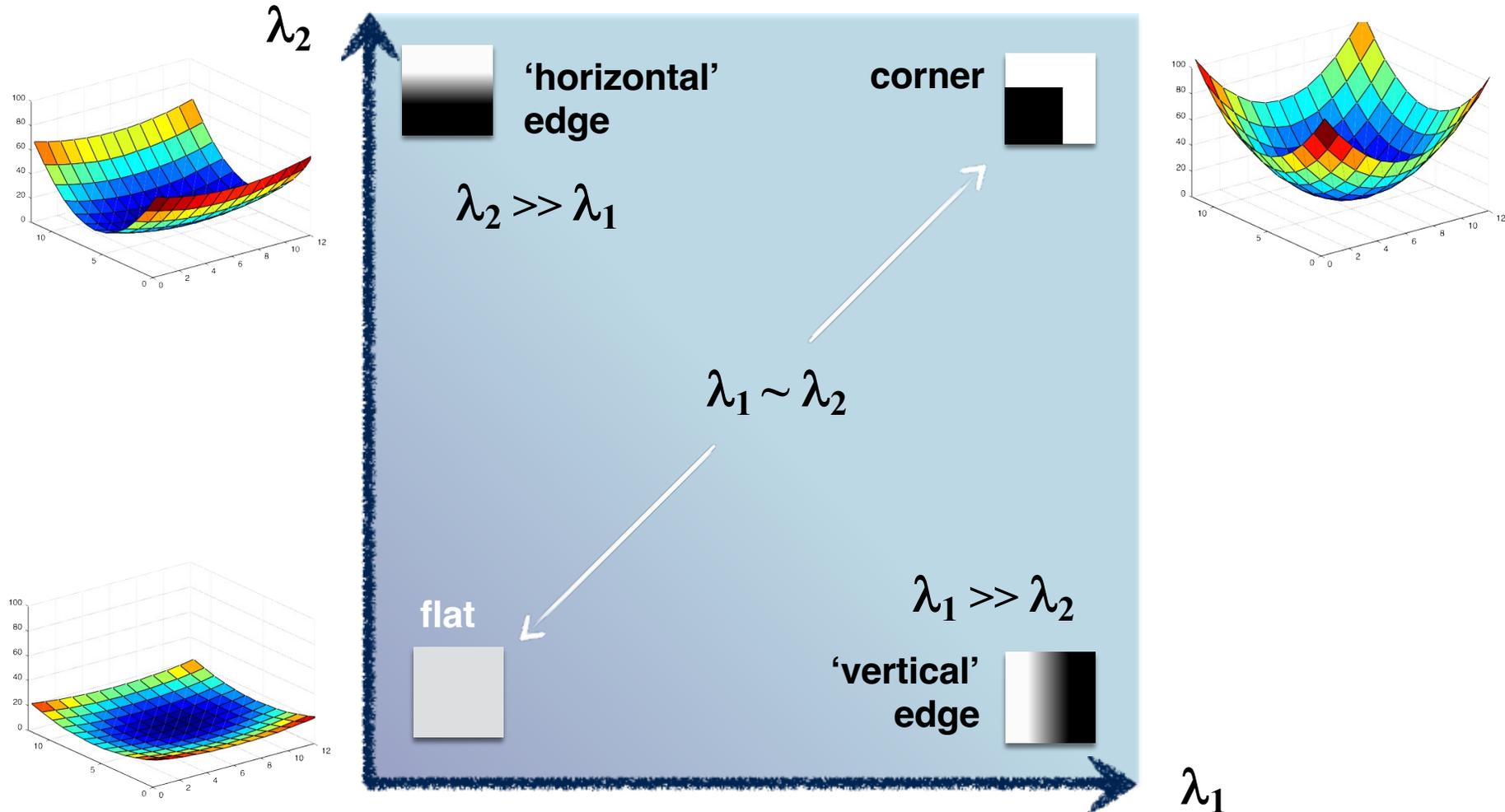
# Gli autovalori



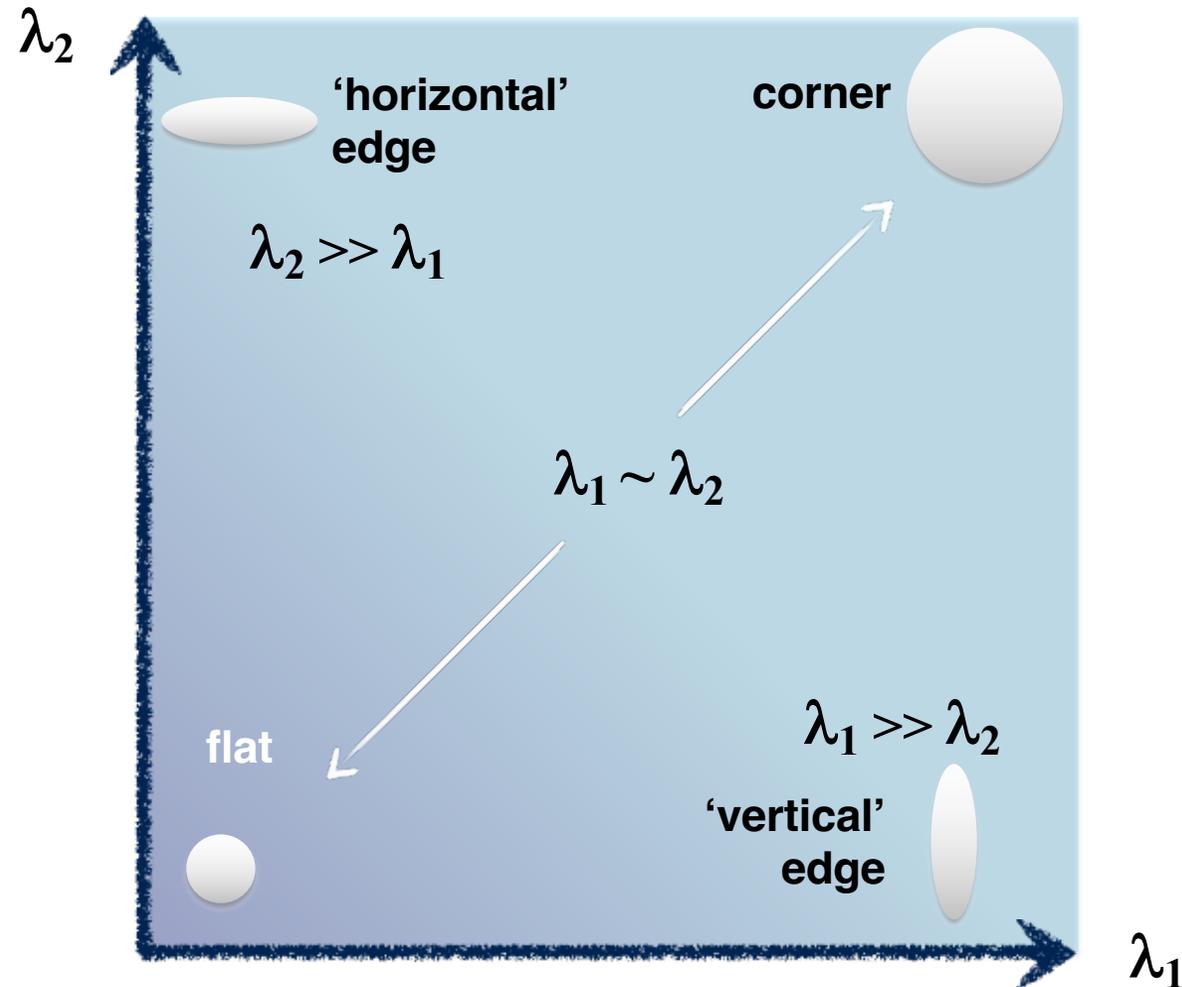
# Gli autovalori



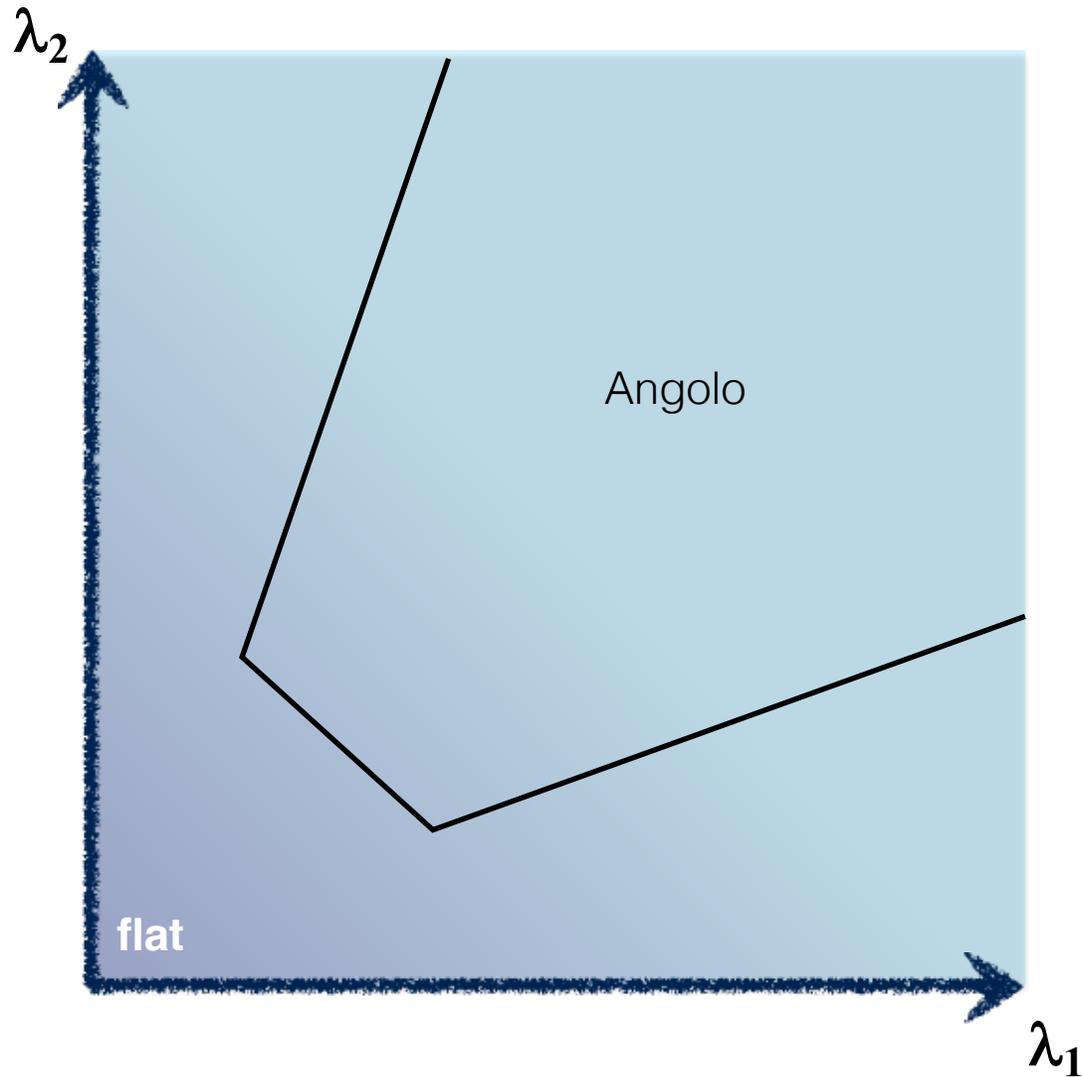
# Gli autovalori



# Gli autovalori

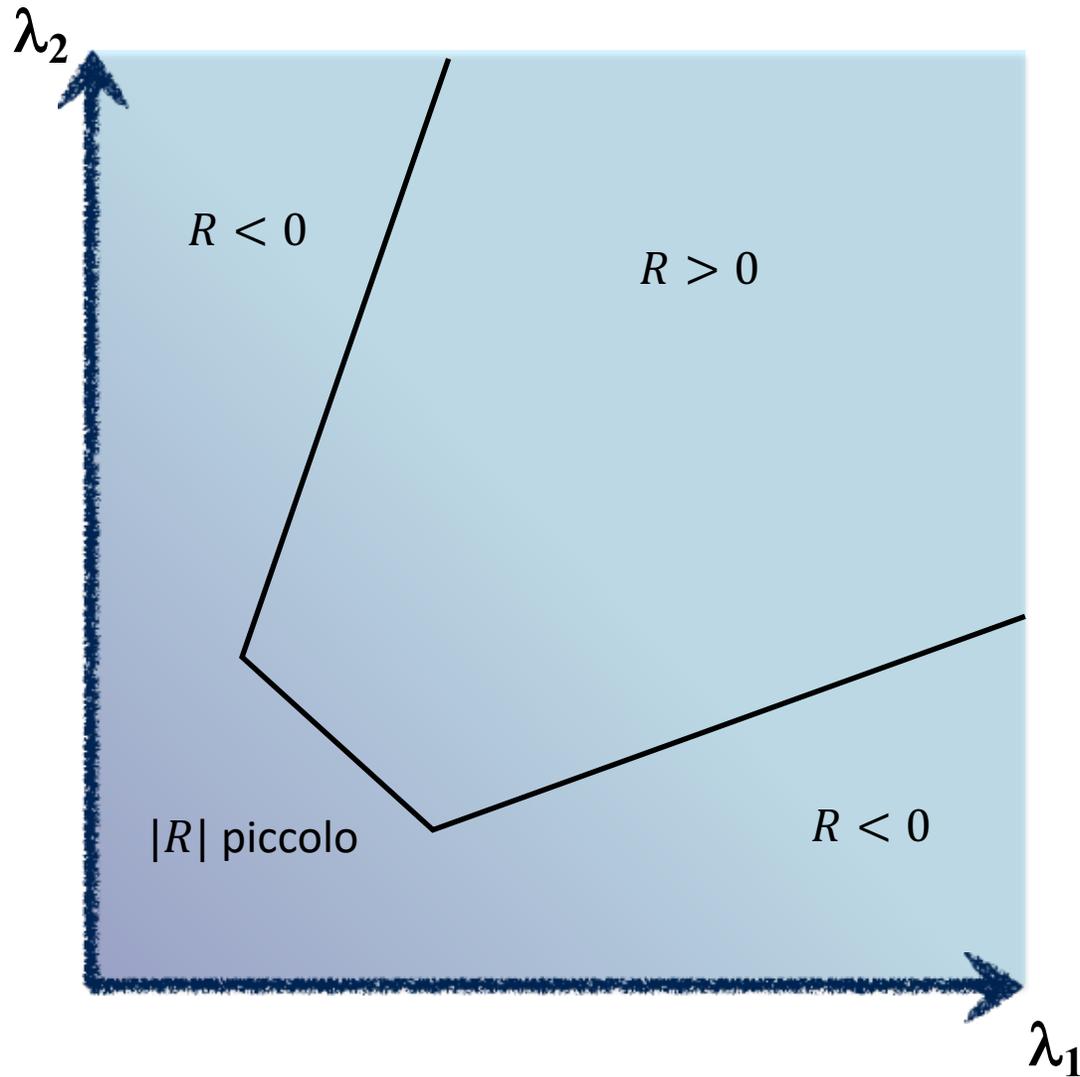


# Gli autovalori



# Gli autovalori

$$R = \det(M) - \alpha \text{Tr}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$



# Harris Detector

1. Calcola le derivate di un'immagine

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

2. Calcolo i prodotti per ogni pixel

$$I_x^2 = I_x I_x \quad I_y^2 = I_y I_y \quad I_{xy} = I_x I_y$$

3. Calcola la somma dei prodotti ad ogni pixel

$$S_x^2 = G_{\sigma'} * I_x^2 \quad S_y^2 = G_{\sigma'} * I_y^2 \quad S_{xy} = G_{\sigma'} * I_{xy}$$

# Harris Detector

4. Calcola la matrice per ogni pixel

$$M = \begin{bmatrix} S_x^2 & S_{xy} \\ S_{xy} & S_y^2 \end{bmatrix}$$

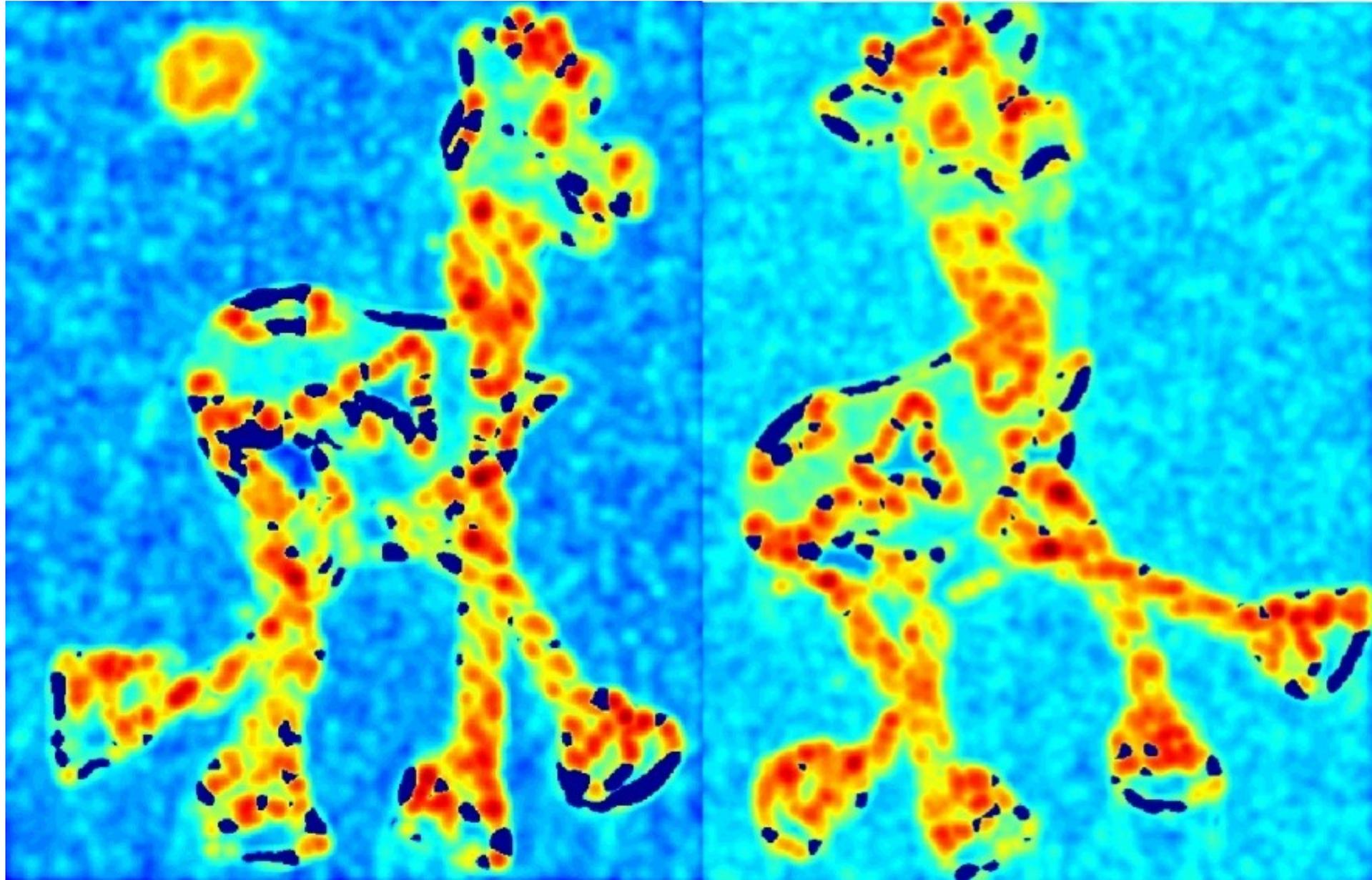
5. Calcola il responso

$$R = \det(M) - \alpha \text{Tr}(M)^2$$

6. Applica una soglia a  $R$
7. Calcola la non-maximal suppression

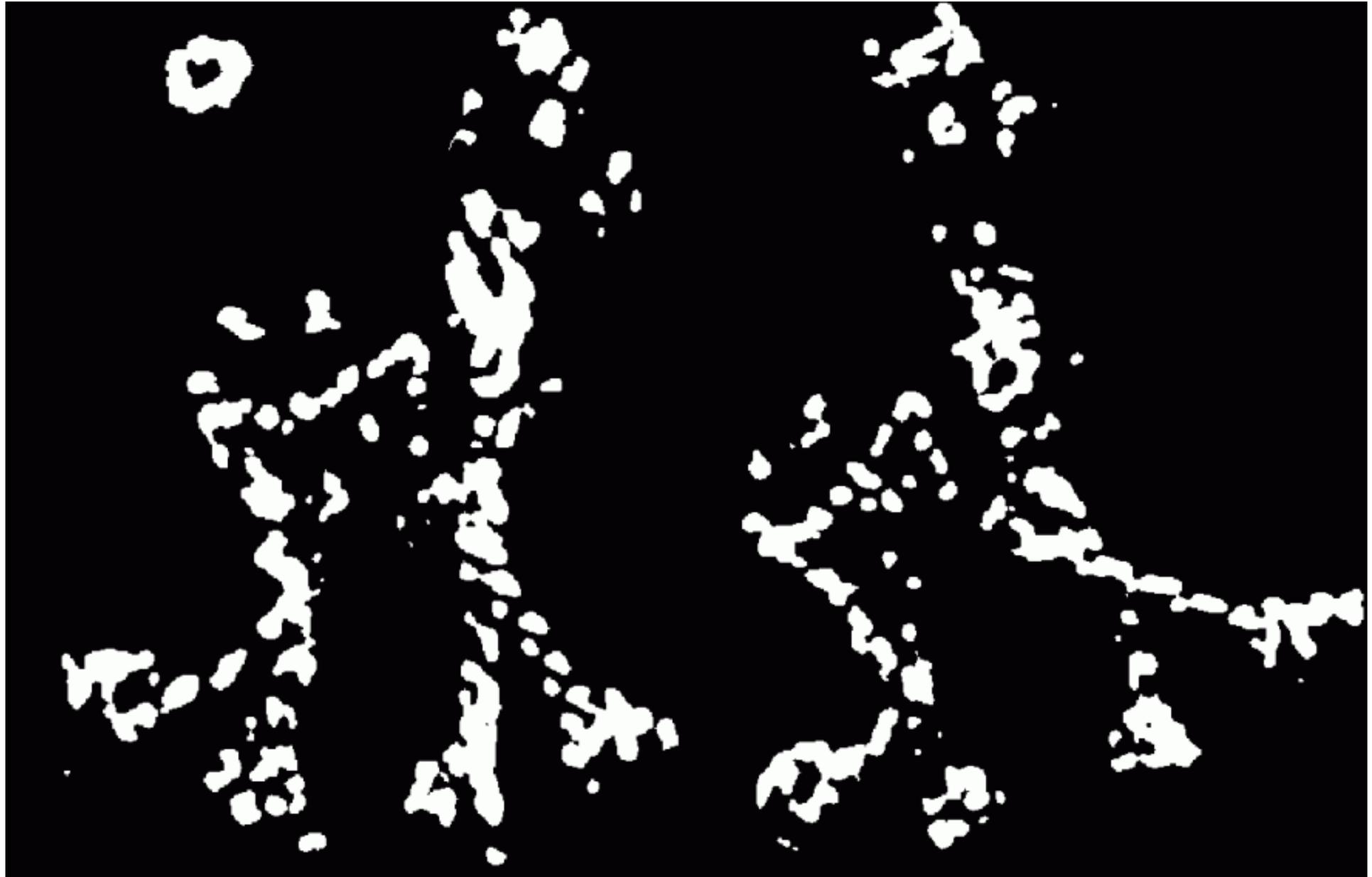


Calcolo di R

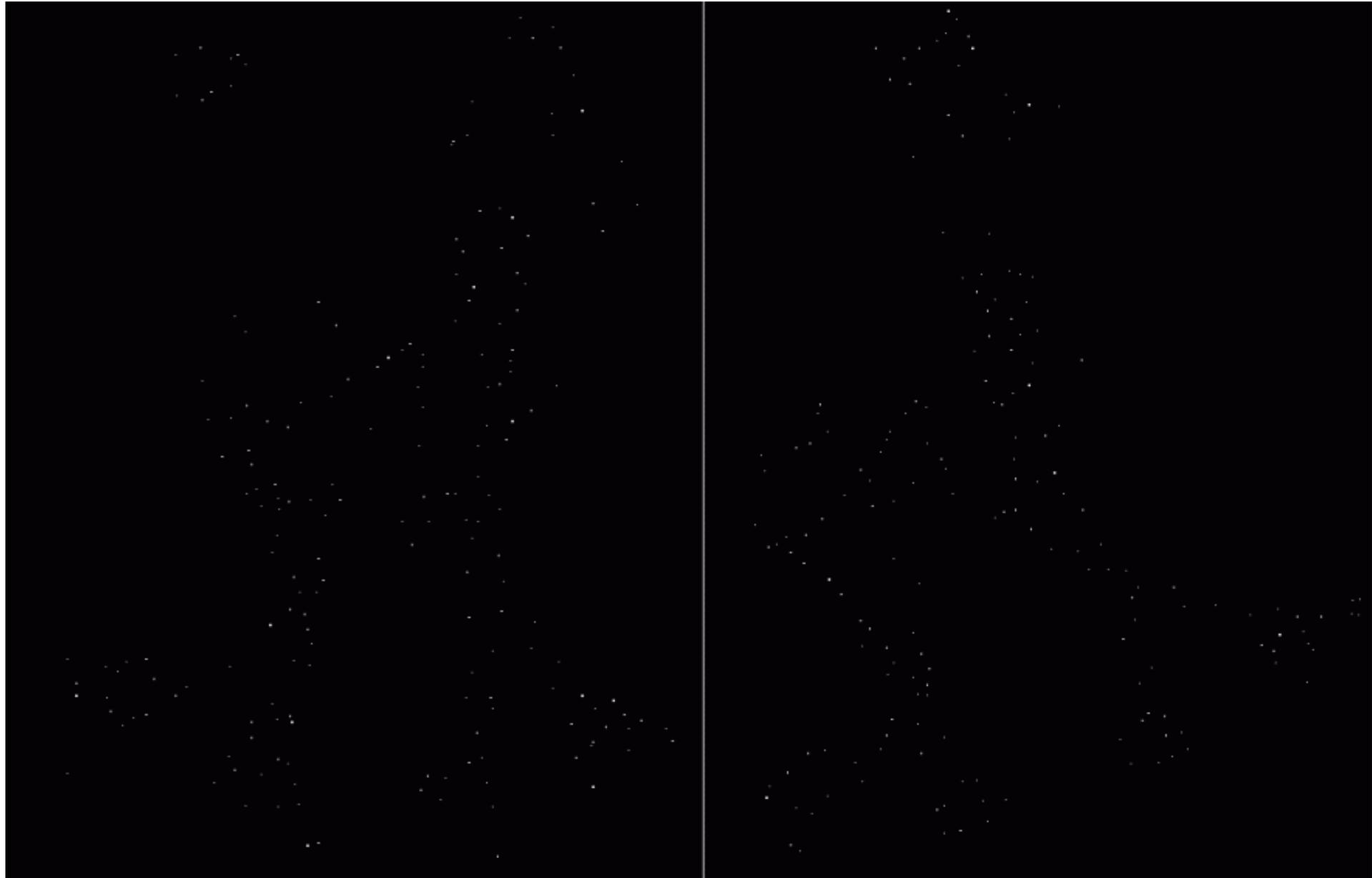




# Thresholding



# Non-maximal suppression





# Qual è la qualità?

- Invarianza
  - Rotazione?
  - Variazioni di intensità?
  - Scala?