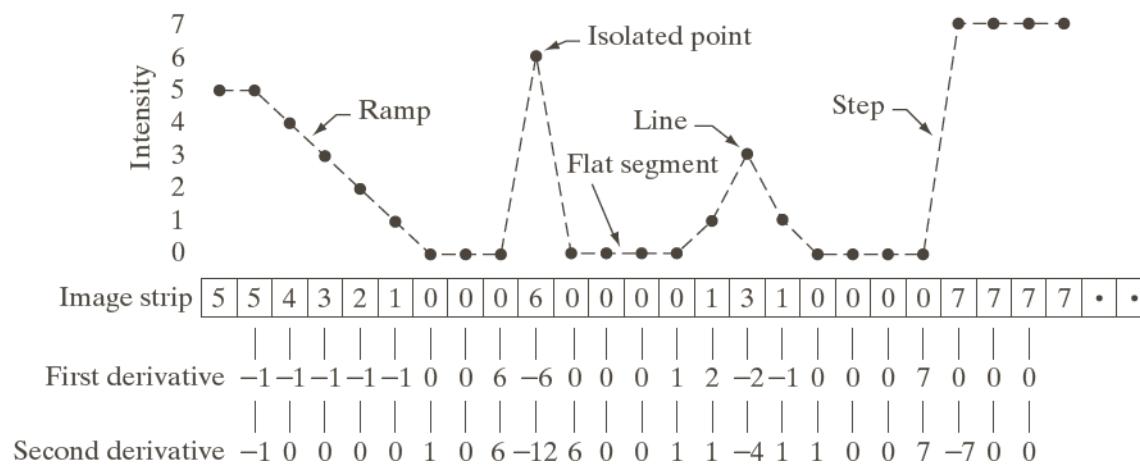
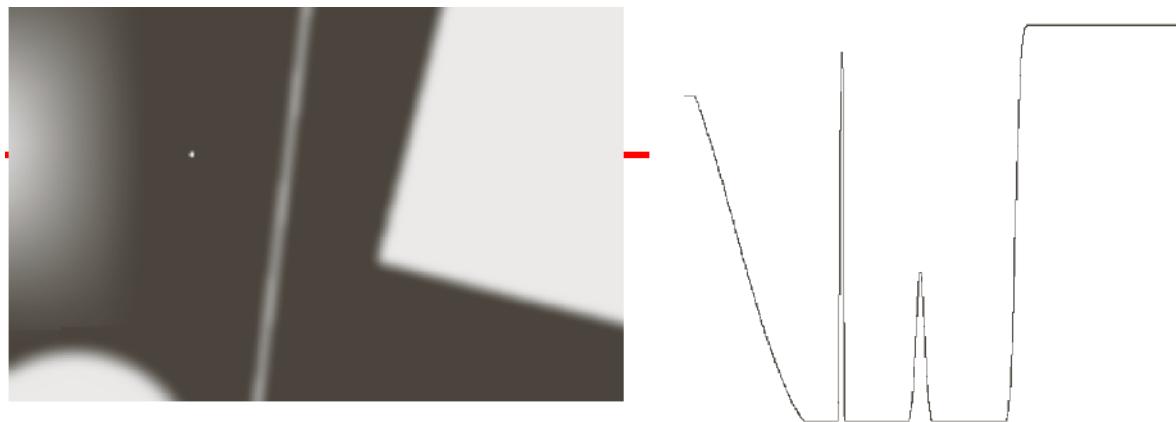




ELABORAZIONE DELLE IMMAGINI

SEGMENTAZIONE DELLE IMMAGINI: CANNY E HARRIS

CARATTERISTICHE DELLE DISCONITUITÀ



Elim Parte II – Prof. A. Ferone

CANNY EDGE DETECTOR

- L'algoritmo di Canny è uno dei migliori algoritmi per l'individuazione degli edge
- Tre obiettivi
 - **Basso tasso di errore:** minimizzare la probabilità di falsi positivi e falsi negativi 
 - **Punti di edge ben localizzati:** gli edge rilevati devono essere il più vicino possibile agli edge reali (la distanza tra il punto rilevato è quello reale deve essere minima) 
 - **Risposta puntuale per edge singolo:** restituire un solo punto per ogni punto di edge 
- La caratteristica di questo algoritmo è la rigida formulazione matematica

CANNY

- Canny **dimostrò** che una buona **approssimazione** per l'individuazione di edge a gradino è la **derivata prima della Gaussiana**

$$\frac{d}{dx} e^{-\frac{x^2}{2\sigma^2}} = \frac{-x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

- Poichè la **direzione** dell'**edge non** è **conosciuta** a priori, bisognerebbe applicare l'operatore in tutte le possibili direzioni
- Questo processo può essere **approssimato** effettuando la **convoluzione** dell'immagine con una **Gaussiana 2-D** e calcolando il **gradiente** del risultato
- **Magnitudo** e **orientazione** del gradiente possono essere utilizzati per calcolare l'**intensità** e l'**orientazione** dell'edge 

FUNZIONE GAUSSIANA

- Sia $f(x, y)$ l'immagine di input e $G(x, y)$ la funzione Gaussiana

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Si effettua la convoluzione delle due funzioni

$$f_s(x, y) = G(x, y) \star f(x, y)$$

- E successivamente si calcola la magnitudo e la direzione del gradiente

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x, y) = \tan^{-1} \left[\frac{g_y}{g_x} \right]$$

$$g_x = \frac{\partial f_s}{\partial x} \quad g_y = \frac{\partial f_s}{\partial y}$$

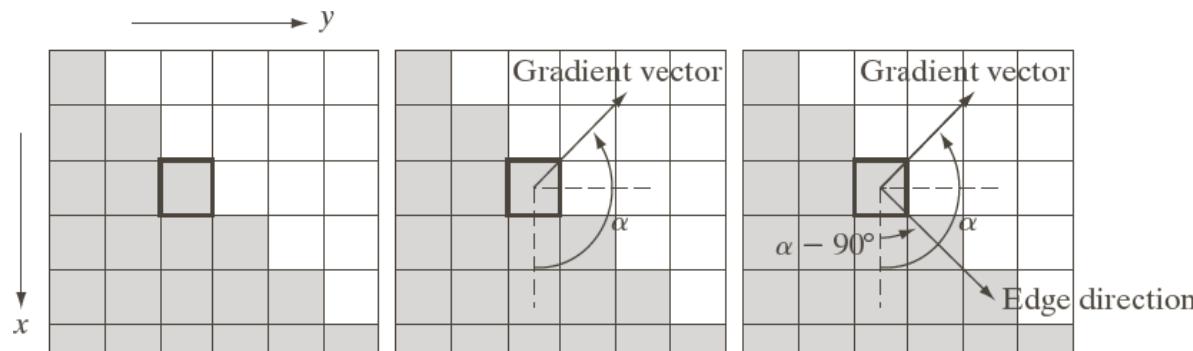
Elim Parte II – Prof. A. Ferone

INDIVDUAZIONE BASATA SU GRADIENTE

- La **direzione** del vettore **gradiente** è data dall'angolo rispetto all'asse x

$$\alpha(x, y) = \tan^{-1} \left[\frac{g_y}{g_x} \right]$$

- Un'**immagine delle direzioni** si può ottenere dividendo g_y e g_x
- Poichè il **gradiente punta** nella **direzione di massima variazione**, la direzione del vettore **gradiente** è **ortogonale** alla direzione dell'**edge**



Elim Parte II – Prof. A. Ferone

FUNZIONE GAUSSIANA

- La funzione **Gaussiana 2-D** si può separare in un **prodotto di due Gaussiane 1-D**
- La convoluzione può essere formulato come **convoluzioni 1-D** che operano sulle **righe** (colonne) dell'immagine e poi sulle **colonne** (righe) del risultato
- Utilizzano l'approssimazione delle differenze in avanti, si possono implementare i calcoli del gradiente come convoluzioni 1-D

FILTRO GAUSSIANO

- La dimensione del filtro **n** deve essere scelta pari al più piccolo intero dispari maggiore o uguale 6σ
- Es.: $\sigma = 4$ $n = 25$
- In OpenCV è possibile utilizzare la seguente funzione 

```
cv::Mat cv::getGaussianKernel(  
    int          ksize,           // Kernel size  
    double       sigma,          // Gaussian half-width  
    int          ktype = CV_32F   // Type for filter coefficients  
);  
  
filter2D(src,dst,CV_32F,getGaussianKernel(n,sigma));
```

NON MAXIMA SUPPRESSION

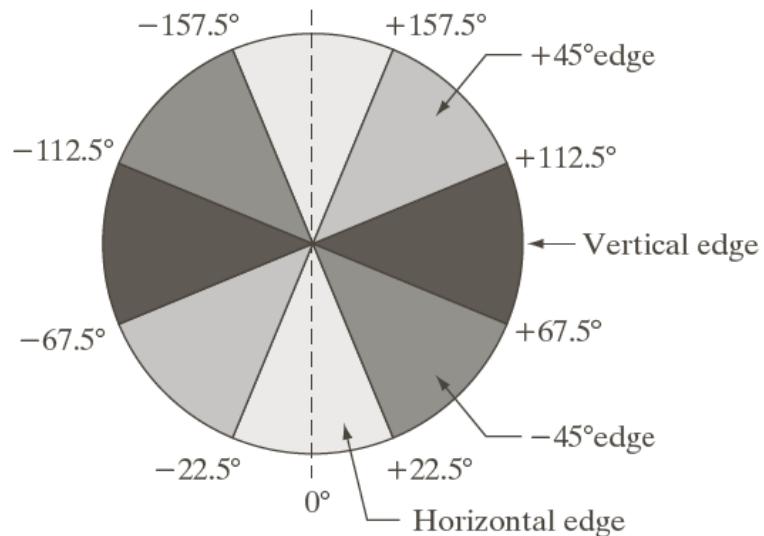


- L'immagine **magnitudo** $M(x,y)$ presenta **molti picchi** intorno ai massimi locali per cui è necessario **sopprimere** tali valori
- A tal fine si considera un **numero finito di orientazioni**, ad es. 4 orientazione in un **intorno 3x3**
 - Orrizzontale
 - Verticale
 - $+45^\circ$
 - -45°



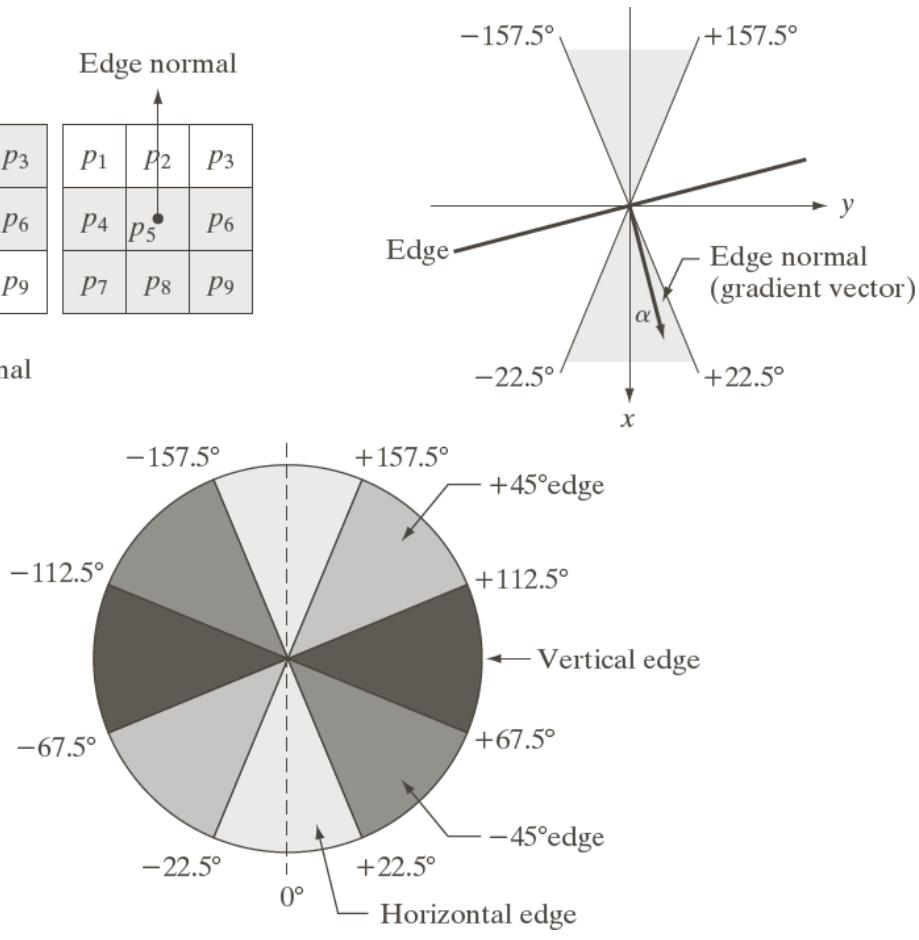
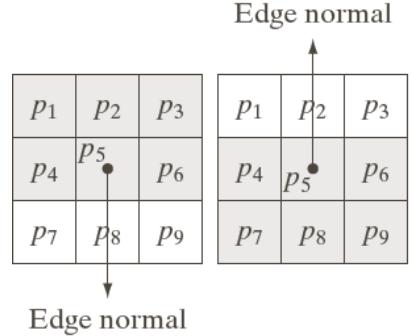
NON MAXIMA SUPPRESSION

- Poichè è necessario **quantizzare** tutte le **possibili direzioni** in **4 settori**, bisogna definire degli **opportuni intervalli** 
- La **direzione** dell'**edge** viene ricava dalla **direzione** del **gradiente** (normale all'edge) 



Elim Parte II – Prof. A. Ferone

NON MAXIMA SUPPRESSION



Elim Parte II – Prof. A. Ferone

NON MAXIMA SUPPRESSION

- Siano d_1, d_2, d_3, d_4 le **quattro direzioni** orizzontale, verticale, $+45^\circ$ e -45°
- Lo schema per un **intorno 3x3** prevede i seguenti **passi**
 - Trovare la direzione d_k più **vicina** ad $\alpha(x, y)$
 - Se $M(x, y)$ è **minore** di **almeno uno** dei **due vicini** lungo la **direzione** d_k ,
 $g_n(x, y) = \mathbf{0}$ (soppressione) altrimenti $g_n(x, y) = M(x, y)$
- L'operazione successiva consiste nel **sogliare** l'immagine $g_n(x, y)$ per **ridurre i falsi positivi**
- Con una **soglia troppo bassa** otteniamo falsi positivi
- Con una **soglia troppo alta** otteniamo falsi negativi



THRESHOLDING CON ISTERESI

- L'**isteresi** utilizza **due soglie**: una soglia bassa T_L ed una soglia alta T_H
- È possibile visualizzare l'operazione di **thresholding** creando **due immagini** ausiliarie

$$g_{NH}(x, y) = g_N(x, y) \geq T_H$$

$$g_{NL}(x, y) = g_N(x, y) \geq T_L$$

- $g_{NH}(x, y)$ avrà meno pixel non nulli di $g_{NL}(x, y)$ ma tutti i pixel non nulli di $g_{NH}(x, y)$ saranno in $g_{NL}(x, y)$, per cui rimuoviamo quelli già in $g_{NH}(x, y)$

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$$

- I pixel di edge in $g_{NH}(x, y)$ sono pixel di edge "**forti**"
- I pixel di edge in $g_{NL}(x, y)$ sono pixel di edge "**deboli**"

THRESHOLDING CON ISTERESI

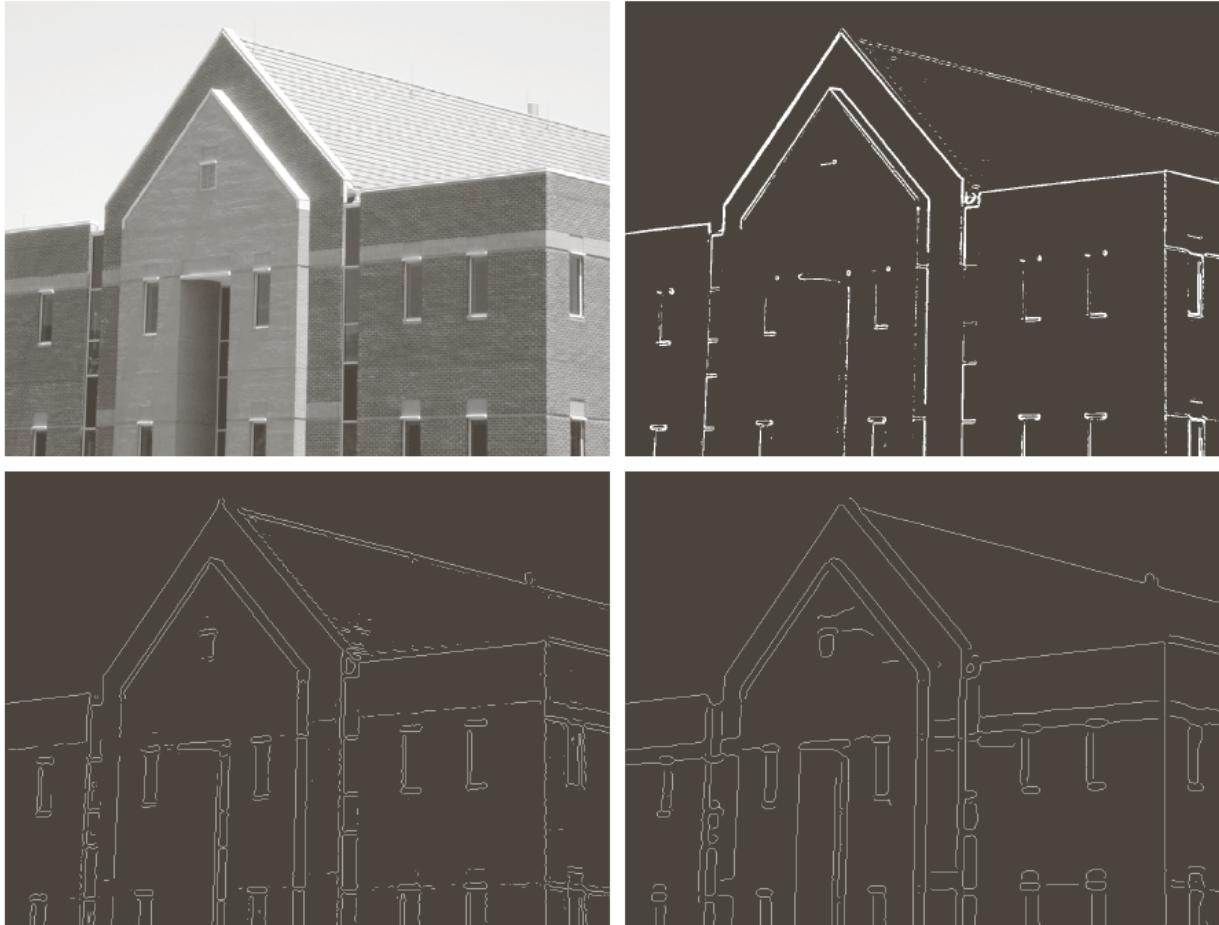
- Per **riempire** i "**vuoti**" negli edge si usa la seguente procedura:
 1. Localizzare il prossimo pixel di edge p in $g_{NH}(x, y)$
 2. Etichettare come pixel di edge "forti" tutti i pixel di edge di $g_{NL}(x, y)$ in $N_8(p)$
 3. Ripetere il passo 1 finchè tutti i pixel in $g_{NH}(x, y)$ non sono stati visitati
- Dal punto di vista **implementativo**, **non** è **necessario** creare le **due immagini** ausiliarie
- È possibile eseguire il thresholding con isteresi direttamente sull'immagine $g_N(x, y)$

STEP ALGORITMO DI CANNY

- Convolvere l'immagini di input con un filtro Gaussiano
- Calcolare la magnitudo e l'orientazione del gradiente
- Applicare la non maxima suppression
- Applicare il thresholding con isteresi



LOG VS CANNY

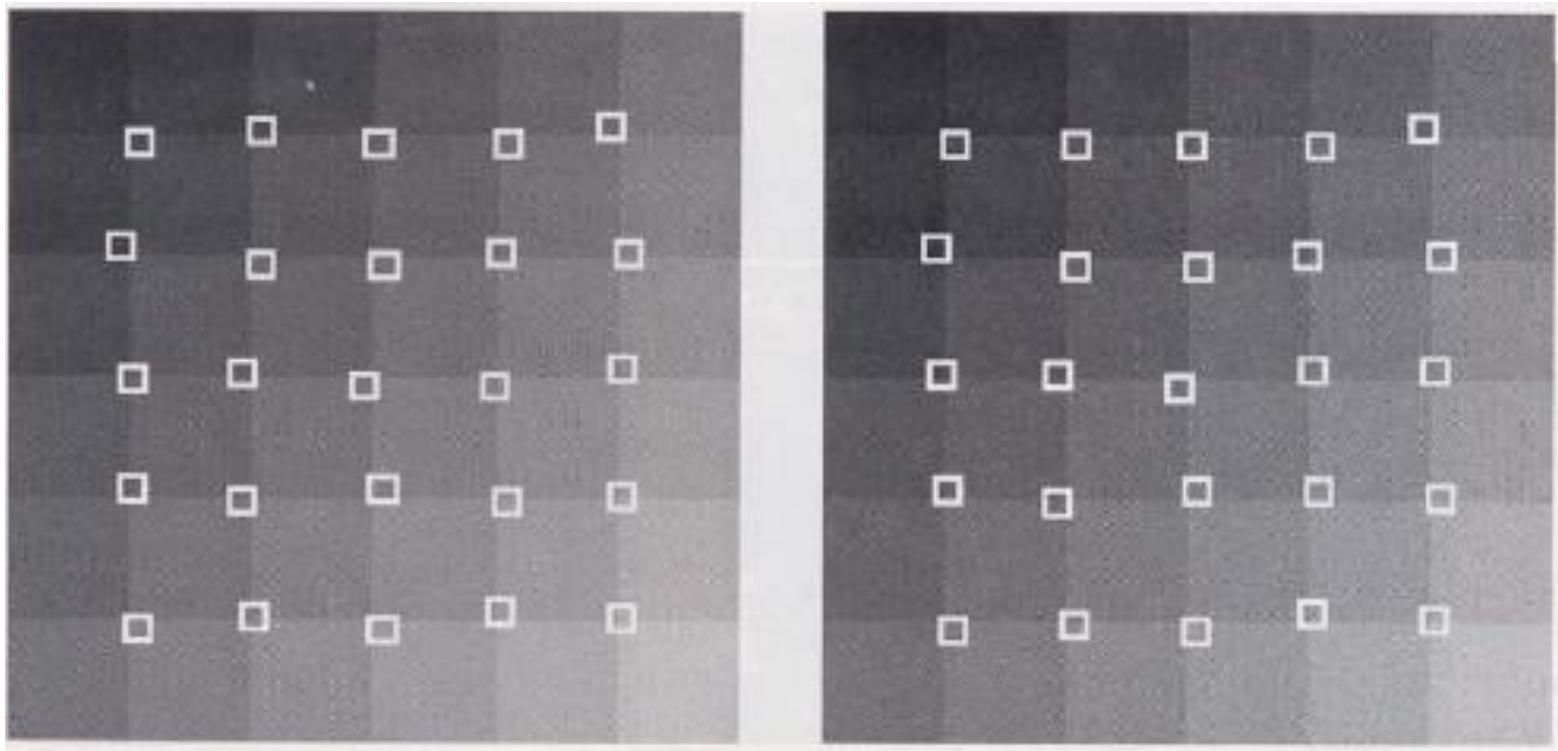


Elim Parte II – Prof. A. Ferone

HARRIS CORNER DETECTOR

- Altre **caratteristiche** molto importanti che possono essere rilevate in un'immagine sono i **corner (angoli)**
- Un **corner** rappresenta: l'**intersezione** di **linee** all'interno dell'immagine o **strutture in pattern di intensità**
- Una caratteristica importante dei **corner** è che si dimostrano **stabili** in **sequenze di immagini (video)** (es.: possono essere utilizzati per tracciare gli oggetti)
- Il concetto di **corner** è strettamente **legato** al concetto di **edge**

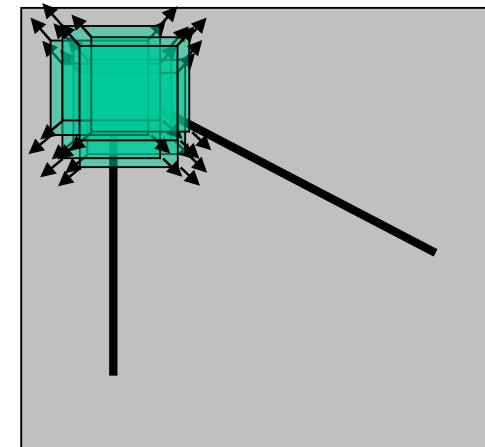
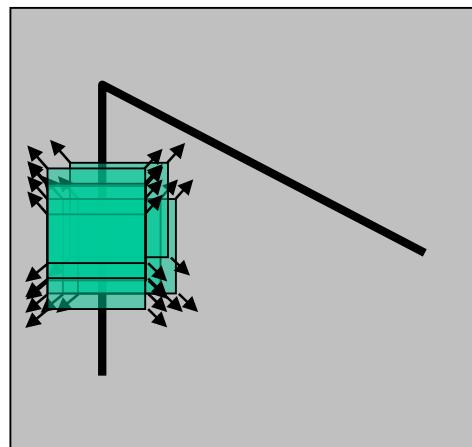
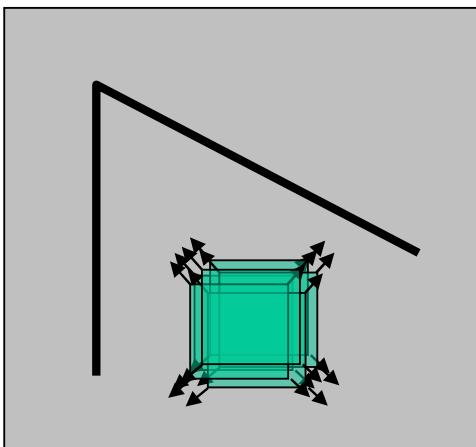
CORNER



Elim Parte II – Prof. A. Ferone

IDEA

- In aree di **intensità uniforme** ci saranno **variazioni nulle**
- In corrispondenza di un **edge** non ci saranno **variazioni** lungo la **direzione ortogonale** all'**edge**
- In corrispondenza di un **corner** ci saranno **variazioni** significative in **più direzioni**

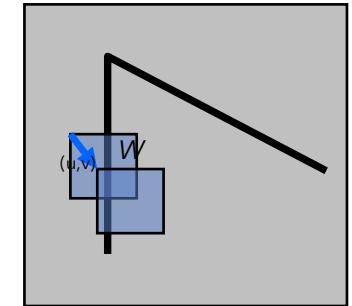


Elim Parte II – Prof. A. Ferone

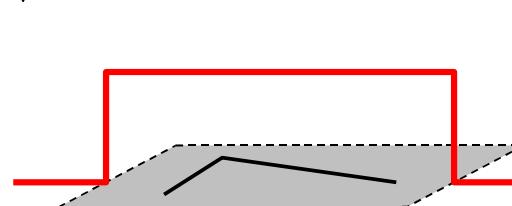
CARATTERIZZAZIONE DI UN CORNER

- Considerando un piccolo spostamento $[u,v]$

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



- La funzione **w()** è utilizzata per **assegnare** un **peso** ad ogni pixel dell'immagine



1 in window, 0 outside



Gaussian

Elim Parte II – Prof. A. Ferone

CARATTERIZZAZIONE DI UN CORNER

- Per un piccolo spostamento, il termine al quadrato viene **approssimato** dal **gradiente**

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



$$E(u, v) = \sum_{x,y} w(x, y)[\nabla I(u, v)\nabla I(u, v)^T], \text{ dove } \nabla I(u, v) = [I_u \ I_v]$$

$$E(u, v) = \sum_{x,y} w(x, y) \begin{bmatrix} I_u^2 & I_u * I_v \\ I_u * I_v & I_v^2 \end{bmatrix}$$

$$E(u, v) = \begin{bmatrix} \sum_{x,y} w(x, y) I_u^2 & \sum_{x,y} w(x, y) I_u * I_v \\ \sum_{x,y} w(x, y) I_u * I_v & \sum_{x,y} w(x, y) I_v^2 \end{bmatrix}$$

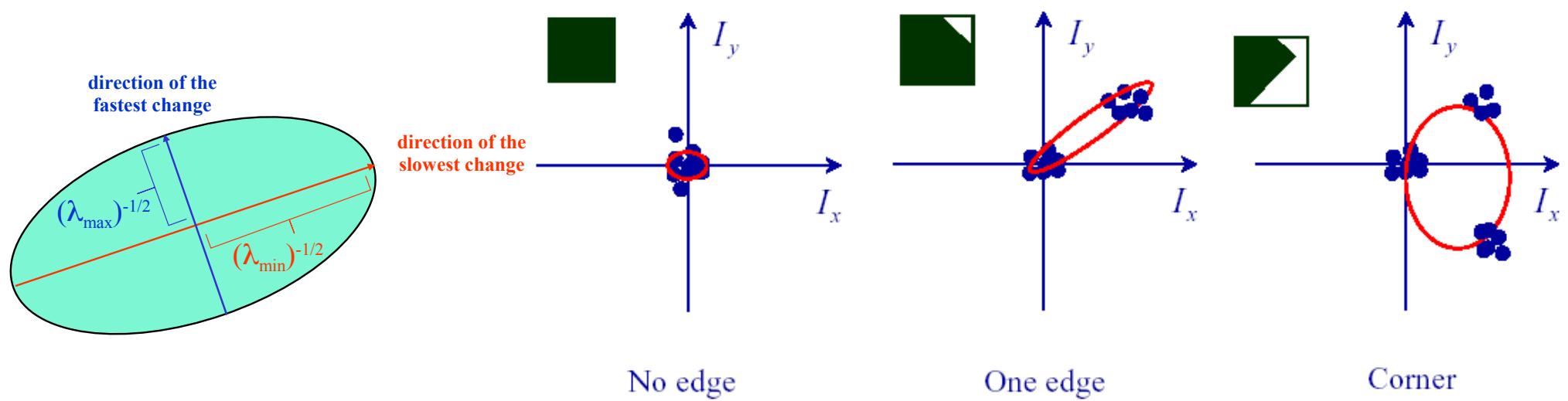
CARATTERIZZAZIONE DI UN CORNER

- La matrice E è **simmetrica**, ovvero $E=E^T$ per cui può essere decomposta tramite **SVD** in

$$E = USU^T = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T$$

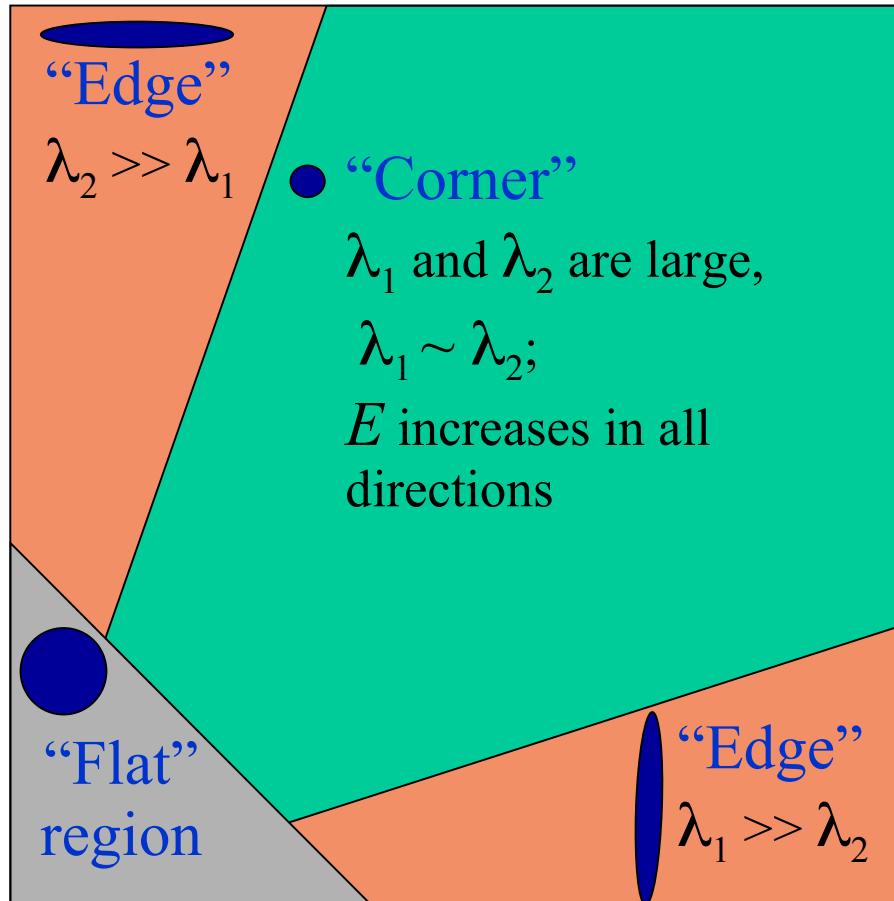
- Dall'**interpretazione geometrica** degli **autovalori** λ_1 e λ_2
 - $\lambda_1 \gg 0$ e $\lambda_2 \approx 0$: nell'**intorno** del punto considerato c'è **una grossa variazione** di **intensità** nella **direzione** dell'**autovettore** associato a λ_1 (l'edge è ortogonale all'autovettore)
 - $\lambda_1 \gg 0$ e $\lambda_2 \gg 0$: nell'**intorno** del punto considerato c'è **una grossa variazione** di **intensità** sia nella **direzione** dell'autovettore associato a λ_1 che nella **direzione** dell'autovettore associato a λ_2 (corner)
 - $\lambda_1 \approx 0$ e $\lambda_2 \approx 0$: zona di **intensità uniforme**

INTERPRETAZIONE GEOMETRICA



Elim Parte II – Prof. A. Ferone

INTERPRETAZIONE GEOMETRICA



Elim Parte II – Prof. A. Ferone

OTTIMIZZAZIONE

- Il calcolo della **SVD** andrebbe ripetuto **per ogni punto** dell'immagine
- Per **ridurre il costo computazionale** è possibile utilizzare degli indici più veloci da calcolare

$$R(u, v) = \det(C(u, v)) - k \operatorname{trace}^2(C(u, v))$$

- Infatti

$$\operatorname{trace}(C(u, v)) = \lambda_1 + \lambda_2$$

$$\det(C(u, v)) = \lambda_1 * \lambda_2$$

- Se $R \gg 0$ nell'intorno è presente un **corner** 

OTTIMIZZAZIONE

- Per calcolare l'indice R

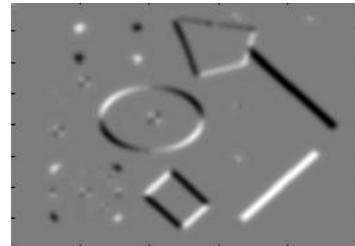
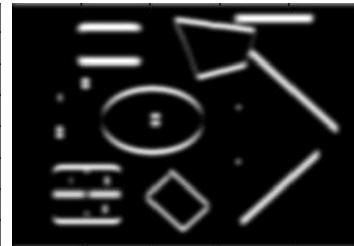
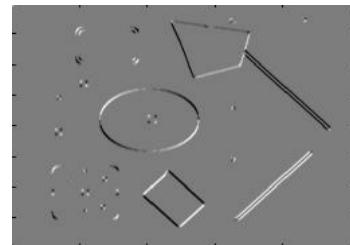
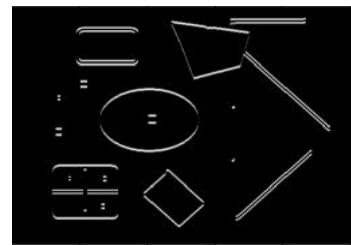
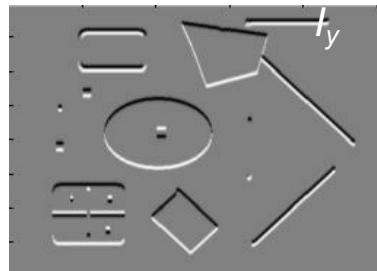
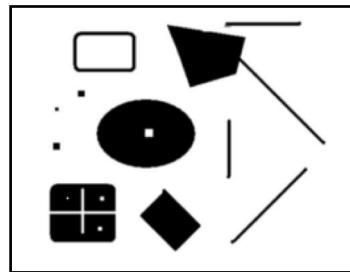
$$E(u, v) = \begin{bmatrix} \sum_{x,y} w(x, y) I_u^2 & \sum_{x,y} w(x, y) I_u * I_v \\ \sum_{x,y} w(x, y) I_u * I_v & \sum_{x,y} w(x, y) I_v^2 \end{bmatrix}$$
$$E(u, v) = \begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix}$$

- determinante = $C_{00} * C_{11} - C_{01} * C_{10}$
- traccia = $C_{00} + C_{11}$
- $R = \text{determinante} - k * \text{traccia}^2$

STEP ALGORITMO DI HARRIS

- Calcolare le derivate parziali rispetto ad x e y (Dx, Dy)
- Calcolare Dx^2, Dy^2 e Dx^*Dy
- Applicare un filtro Gaussiano a Dx^2, Dy^2 e Dx^*Dy
- Si ottengono C_{00}, C_{11}, C_{01} (e quindi anche C_{10})
- Calcolare l'indice R
- Normalizzare l'indice R in $[0, 255]$
- Sogliare R

STEP ALGORITMO DI HARRIS



Elim Parte II – Prof. A. Ferone

ESERCIZI

- Implementare l'algoritmo di Canny
- Implementare l'algoritmo di Harris

