



# ELABORAZIONE DELLE IMMAGINI

SEGMENTAZIONE DELLE IMMAGINI

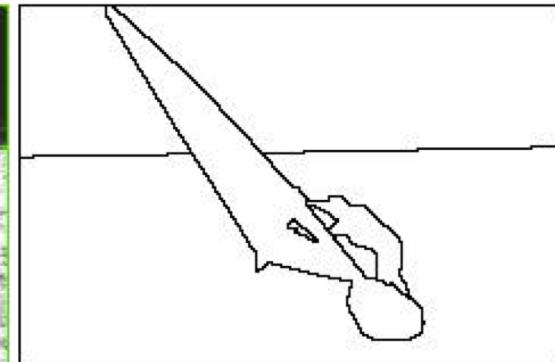
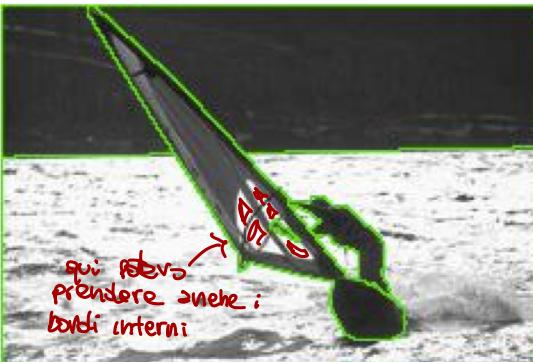
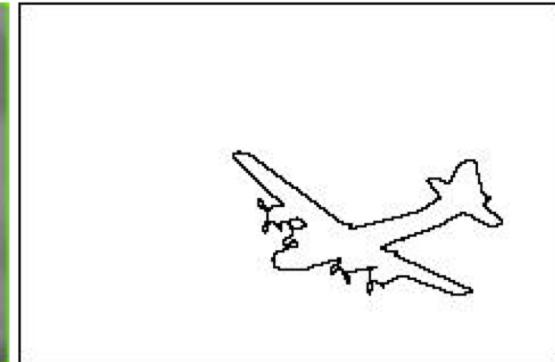
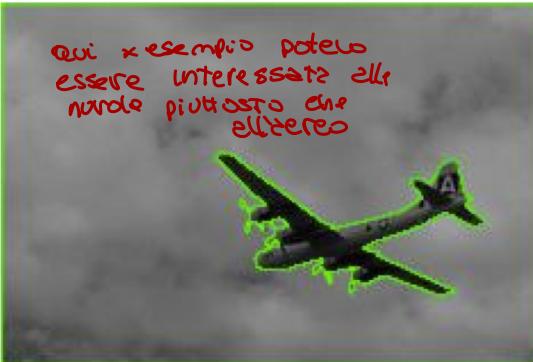
## SEGMENTAZIONE

- **Obiettivo** della **segmentazione** è **suddividere** un'**immagine** nelle **regioni** o negli **oggetti** che la compongono
- Rappresenta uno degli **obiettivi** più **importanti** ed allo stesso tempo più **difficili** da raggiungere
- La **qualità** della **segmentazione** può determinare l'esito delle elaborazioni successive
- La maggior parte degli **algoritmi** si basa su una delle **due proprietà** dei base dei **valori di intensità**
  - **Discontinuità**  
(i bordi)
  - **Similarità**  
(caratteristiche simili)

Sono caratteristiche attraverso le quali posso individuare delle regioni.

# SEGMENTAZIONE

Non c'è una sola soluzione alla segmentazione  
essendo un lavoro manuale è una percezione  
molto soggettiva \*



- <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

## SEGMENTAZIONE

- Se si sfruttano le **discontinuità**, la segmentazione sarà guidata da bruschi **cambiamenti** di intesità, es.: **edge**
  - Canny
  - Harris
  - Hough ("~~Hough~~")
- Se si sfruttano le **similarità**, si **raggruppano pixel simili** in base a dei criteri di similarità
  - Sogliatura
  - Region growing
  - Split and merge
  - Clustering



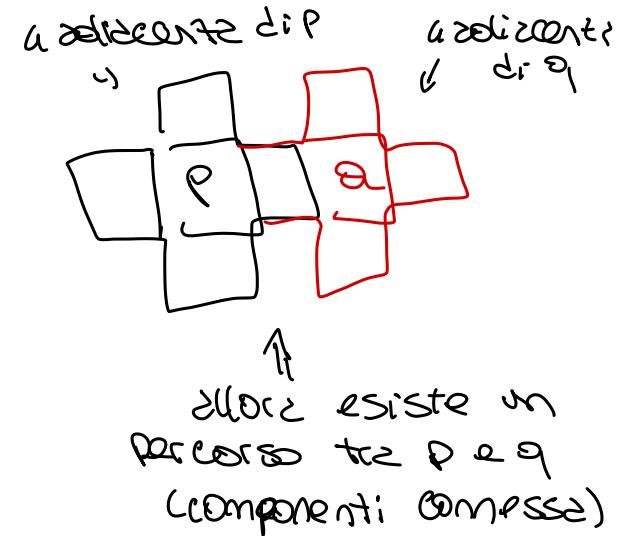
## FONDAMENTI

- Denotiamo con  $R$  la regione occupata dall'**immagine**
- La **segmentazione** dell'immagine consiste nel **partizionare**  $R$  in  $n$  **sottoregioni**  $R_1, R_2, \dots, R_n$  tali che

- $\bigcup_{i=1}^n R_i = R$  (segmentazione completa) (Unione delle regioni)
- $R_i$  è un insieme连通的  $i = 1, 2, \dots, n$  (pixel 4/8连通的) (i pixel di ogni regione rappresentano una componente连通的 esistente un percorso tra i pixel della regione)
- $R_i \cap R_j = \emptyset$  (regioni disgiunte) (non si intersecano)
- $Q(R_i) = \text{Vero}$  per  $i = 1, 2, \dots, n$  (tutti i pixel in  $R_i$  rispettano la proprietà in  $Q$ ) (es. se tutti i pixel sono nello stesso intervallo allora = V)
- $Q(R_i \cup R_j) = \text{Falso}$   $\forall R_i$  e  $R_j$  adiacenti (pixel in regioni adiacenti devono avere proprietà diverse) ( $\Rightarrow$  il predicato dell'unione delle regioni è Falso, se fosse vero, allora significherebbe che erano unica regione, poiché entrambi il predicato è vero (rispettano una regola))
- $Q$  è un predicato logico definito sui punti di una regione  $R_i$

## ADIACENZA, CONNETTIVITÀ E REGIONI

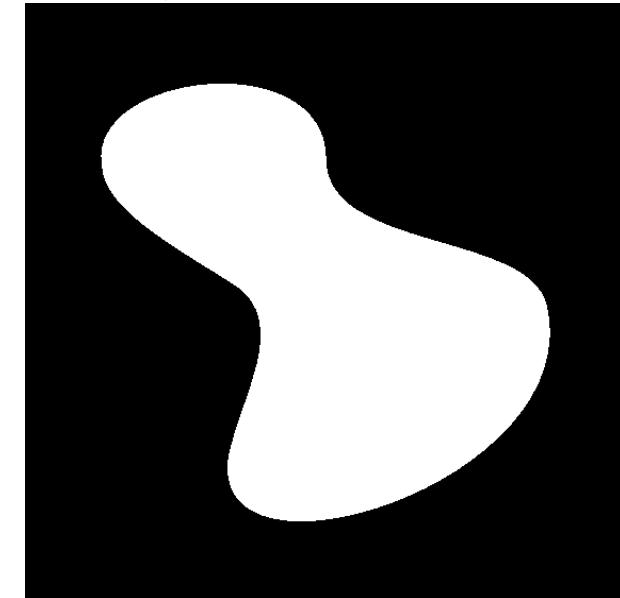
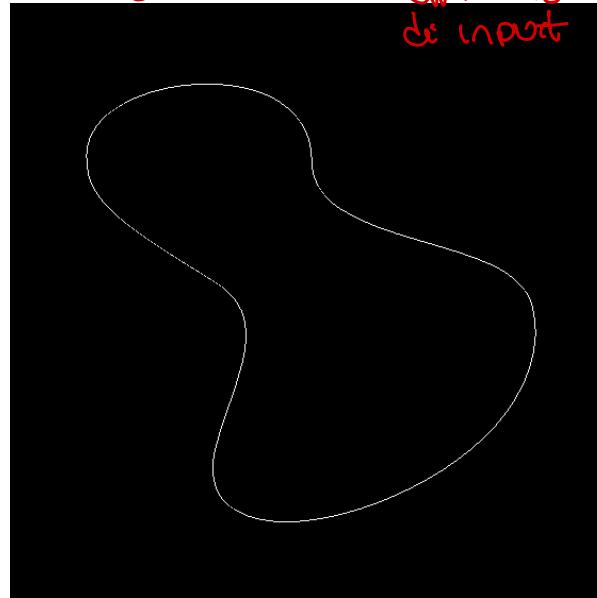
- **Adiacenza** rispetto ai **valori** in un **insieme V**
  - **4-adiacenza**: due pixel  $p$  e  $q$  con valori in  $V$   $q \in N_4(p)$
  - **8-adiacenza**: due pixel  $p$  e  $q$  con valori in  $V$   $q \in N_8(p)$
  - **m-adiacenza**: due pixel  $p$  e  $q$  con valori in  $V$ 
    - $q \in N_4(p)$
    - $q \in N_D(p)$  e  $N_4(p) \cap N_4(q)$  non contiene pixel in  $V$
- Un **path** da  $p$  a  $q$  è una **sequenza di pixel adiacenti** dal pixel  $p$  al pixel  $q$ 
  - $p$  e  $q$  si dicono **connessi**
- I **pixel connessi** di una regione  $S$  formano una **componente connessa**
- **Se in S esiste una sola componente connessa S è una regione**



## SEGMENTAZIONE **EDGE BASED**

- Se si sfruttano le **discontinuità**, si assume che i **bordi** siano sufficientemente **diversi tra le regioni** e dallo **sfondo** in modo da poter sfruttare le intensità locali

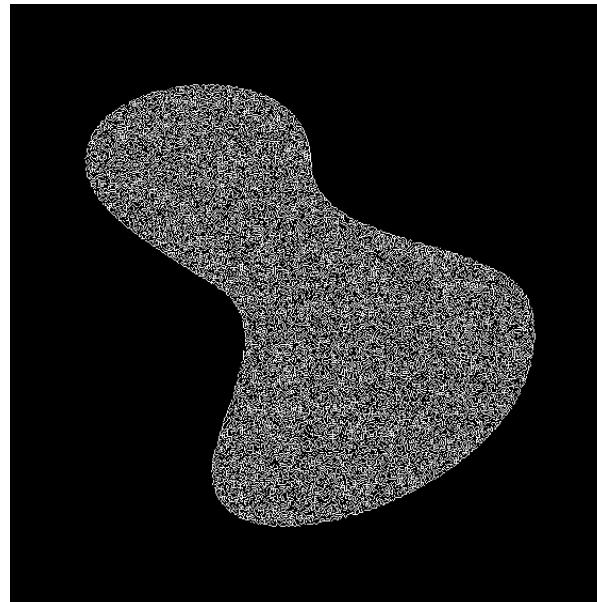
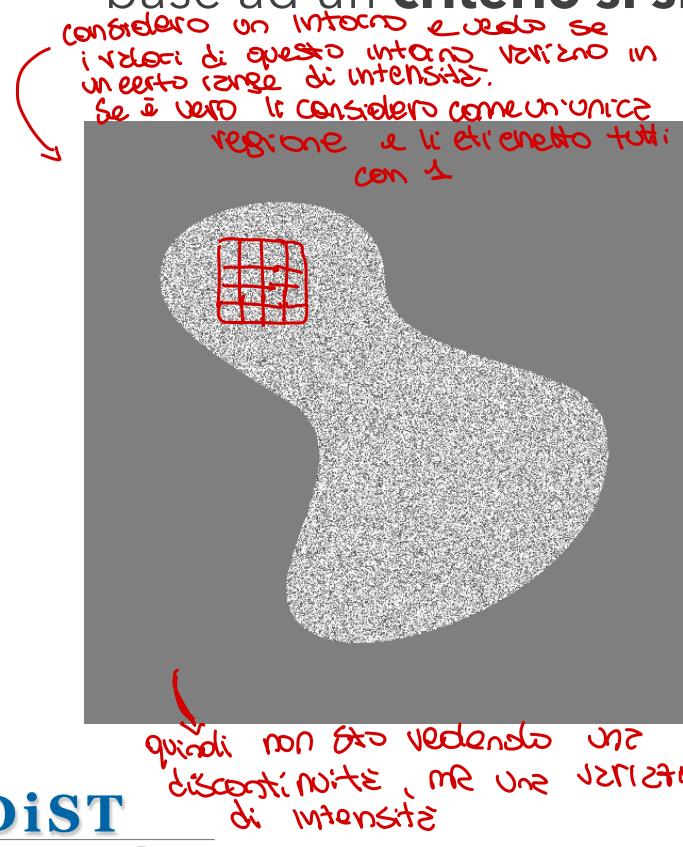
sto applicando una  
estraggo il bordo → moltiplico gli intensità  
di input  
gli intensità → imposto pixel interni al bordo a  
bianco



↑ la risoluzione dell'operatore è ideale  
non c'è rumore  
Elim Parte II – Prof. A. Ferone

## SEGMENTAZIONE REGION BASED

- Se si sfruttano le **similarità**, si **partiziona** l'immagine in **regioni simili** in base ad un **criterio di similarità**



Elim Parte II – Prof. A. Ferone

## SEGMENTAZIONE EDGE BASED

→ vogliamo individuare sequenze di punti  
che presentano una certa discontinuità

Elim Parte II – Prof. A. Ferone

## EDGE, LINEE E PUNTI

- Siamo interessati a **tre caratteristiche** di un'immagine
  - **Edge**: insiemi di pixel di edge, ovvero pixel in cui si presenta una repentina variazione di intensità
  - **Linee**: segmenti di edge in cui l'intensità ai lati della linea è minore o maggiore dell'intensità dei pixel della linea
  - **Punti**: linee di lunghezza e larghezza pari a 1 pixel
- Per **individuare** le **variazioni** di intensità a cui siamo interessati, utilizziamo le **approssimazioni** delle **derivate prime e seconde** definite in termini di differenze

## DERIVATA PRIMA DI UN'IMMAGINE

- Poiché l'immagine è una funzione discreta, la definizione classica di derivata non può essere applicata
- È necessario definire un operatore che soddisfi le principali **proprietà della derivata prima**:
  1. Uguale a zero dove l'intensità è costante (le derivate di una costante sono 0)
  2. Diversa da zero per una transizione di intensità
  3. Costante sulle rampe in cui la transizione di intensità è costante, allora le derivate sono proporzionali alla pendenza delle rampe
- L'operatore di derivazione naturale è la differenza tra l'intensità dei pixel vicini (**differenziazione spaziale**)

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

Elim Parte II – Prof. A. Ferone

## DERIVATA SECONDA DI UN'IMMAGINE

- Analogamente, l'operatore di **derivata seconda** può essere definito come:

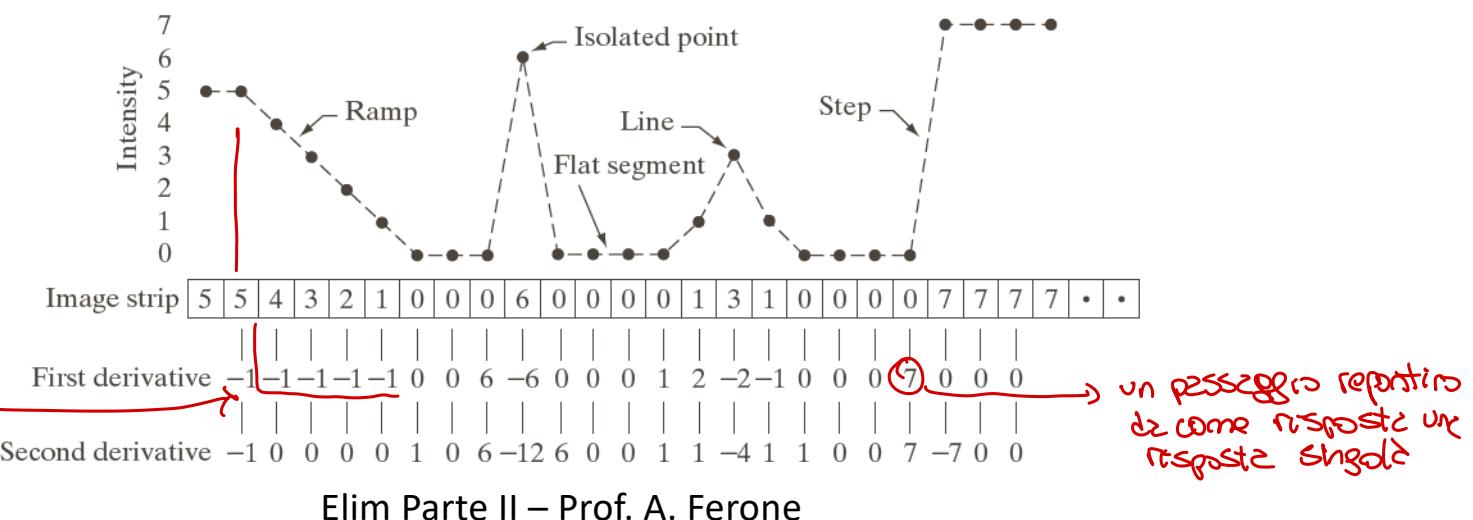
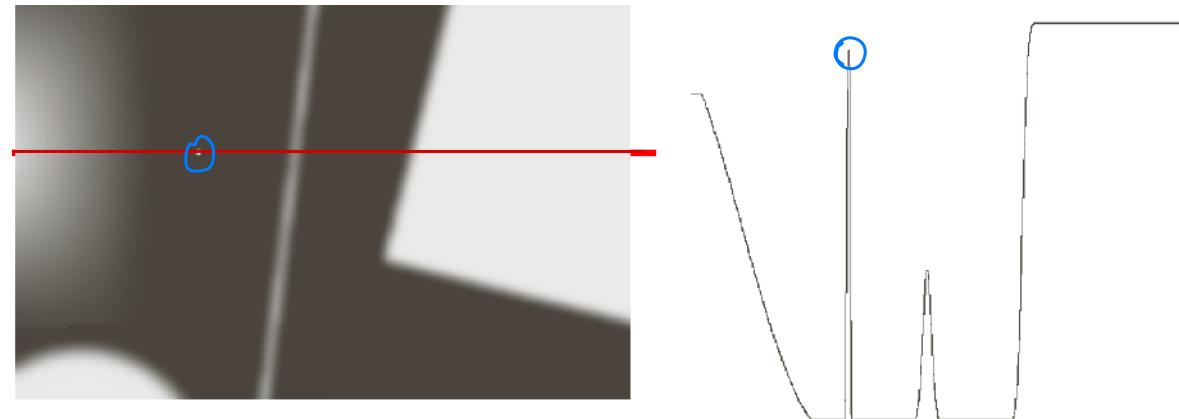
$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= f(x+1) - f(x) - (f(x) - f(x-1)) \\ &= f(x+1) - 2f(x) + f(x-1)\end{aligned}$$

- Soddisfa le seguenti **proprietà**:

1. È uguale a zero dove l'intensità è costante
  2. È diverso da zero all'inizio di un passo (o rampa) di intensità
  3. È uguale a zero sulle pendenze costanti delle rampe
- $\frac{\partial^2 f}{\partial x^2}$  è definita usando i pixel precedente e successivo
- (siccome la derivata prima  
è costante, la derivata  
seconda delle costante è 0)

## CARATTERISTICHE DELLE DISCONTINUITÀ

sto analizzando i  
rampi di intensità  
di queste linee



## PROPRIETÀ DELLE DERIVATE

- **Proprietà** delle derivate prima e seconda
  - In presenza di **edge a rampa**, la **derivata prima** produce **edge spessi** mentre la **derivata seconda** produce **edge sottili**
  - La **risposta** della **derivata seconda** in presenza di punti isolati è **più forte** rispetto a quella della derivata prima
  - Sia sugli **edge a rampa** che su quelli **a gradino**, la **derivata seconda** ha **segni opposti**
  - Il **segno** della **derivata seconda** può essere utilizzato per **determinare** se un **edge** è una transizione **chiaro/scuro** (derivata seconda negativa) o **viceversa** (derivata seconda positiva)

## PUNTI ISOLATI

- In un'immagine digitale, le **derivate seconde** rispetto a  $x$  e  $y$  sono calcolate come:

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

- Quindi, il **Laplaciano** risulta:

$$\begin{aligned}\nabla^2 f(x, y) &= f(x+1, y) + f(x-1, y) + f(x, y+1) \\ &\quad + f(x, y-1) - 4f(x, y)\end{aligned}$$

- Può essere considerata anche la derivata lungo le diagonali:

$$\begin{aligned}\nabla^2 f(x, y) &+ f(x-1, y-1) + f(x+1, y+1) \\ &+ f(x-1, y+1) + f(x+1, y-1) - 4f(x, y)\end{aligned}$$

Elim Parte II – Prof. A. Ferone

## FILTRO LAPLACIANO

0	1	0
1	-4	1
0	1	0

$$\begin{aligned}\nabla^2 f(x, y) = & f(x+1, y) + f(x-1, y) + f(x, y+1) \\ & + f(x, y-1) - 4f(x, y)\end{aligned}$$

Filtro Laplaciano invariante alle rotazioni di 90°

1	1	1
1	-8	1
1	1	1

$$\begin{aligned}\nabla^2 f(x, y) = & f(x-1, y-1) + f(x+1, y+1) \\ & + f(x-1, y+1) + f(x+1, y-1) - 4f(x, y)\end{aligned}$$

Filtro Laplaciano invariante alle rotazioni di 45°

Elim Parte II – Prof. A. Ferone

## PUNTI ISOLATI

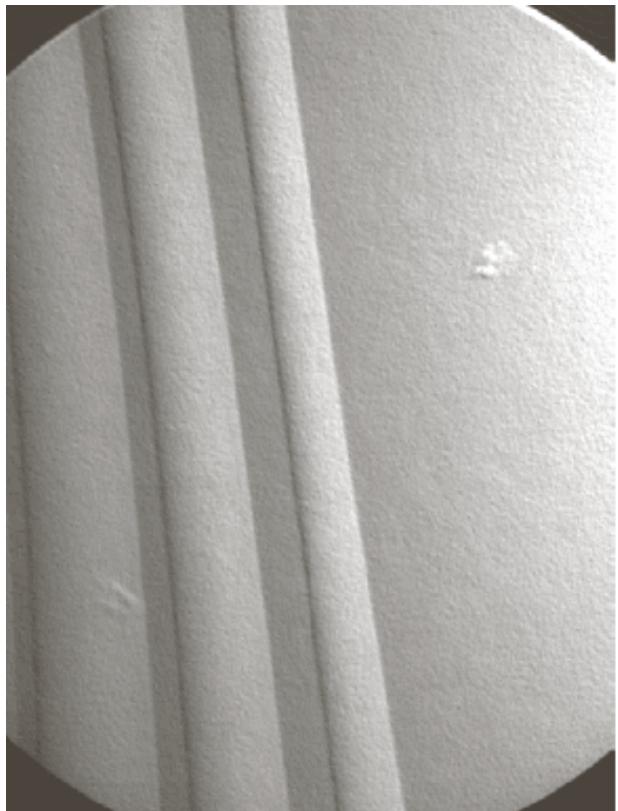
- Dopo aver applicato il **filtro Laplaciano**, si utilizza una **soglia** sulla risposta del filtro per determinare i punti di discontinuità

(Pixel ottenuti con laplaciano superano una certa soglia)

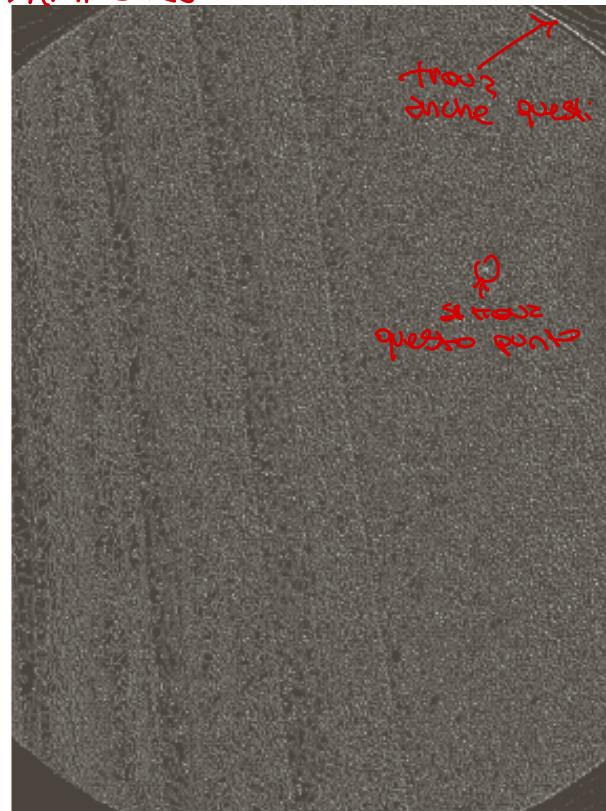
$$g(x, y) = \begin{cases} 1 & \text{se } |R(x, y)| \geq T \\ 0 & \text{altrimenti} \end{cases}$$

- dove  $g$  è l'immagine di output,  $T$  è una soglia non negativa ed  $R$  è la risposta del filtro
- L'**intensità** di un **punto isolato** sarà abbastanza **diversa** dalle intensità dei suoi **8 vicini**
  - nel centro  sarà molto diverso (in un fotogramma)
- La differenza di intensità tra i punti vicini viene controllata dal **parametro  $T$**

## PUNTI ISOLATI



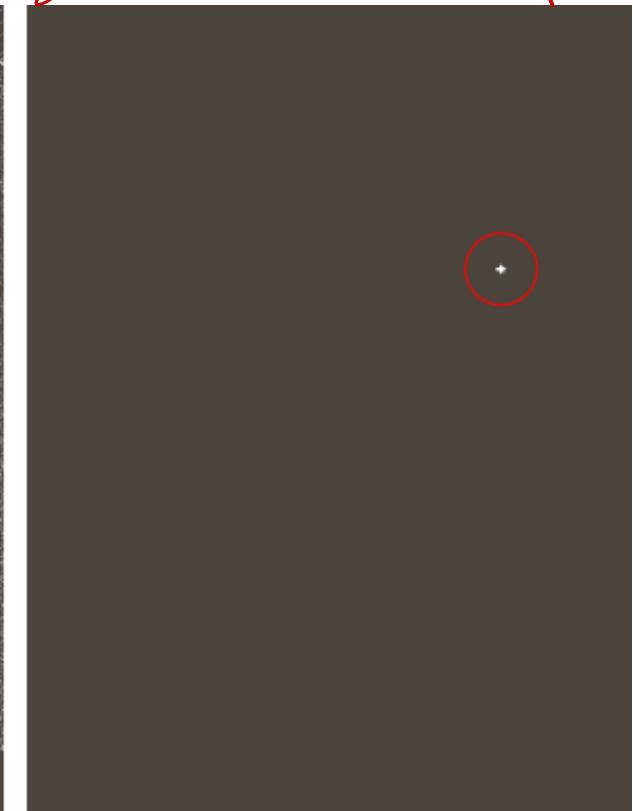
LAPLACEAU



trovò  
anche questi

si trova  
questo punto

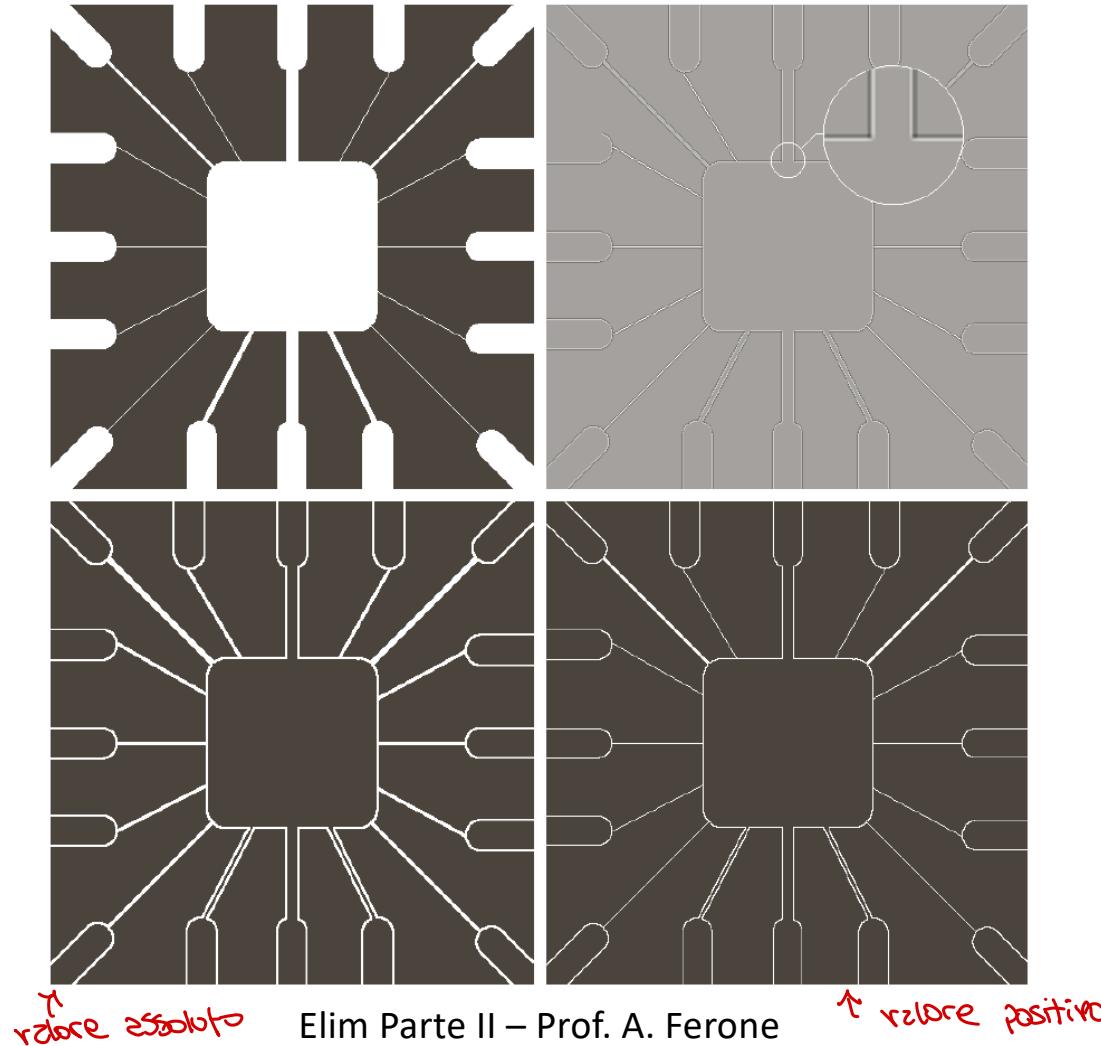
SOGlia (N.B l'immagine è stata tagliata  
perciò non trovi i bordi)



LINEE non ho capito [Sandi audi 45:00 min]

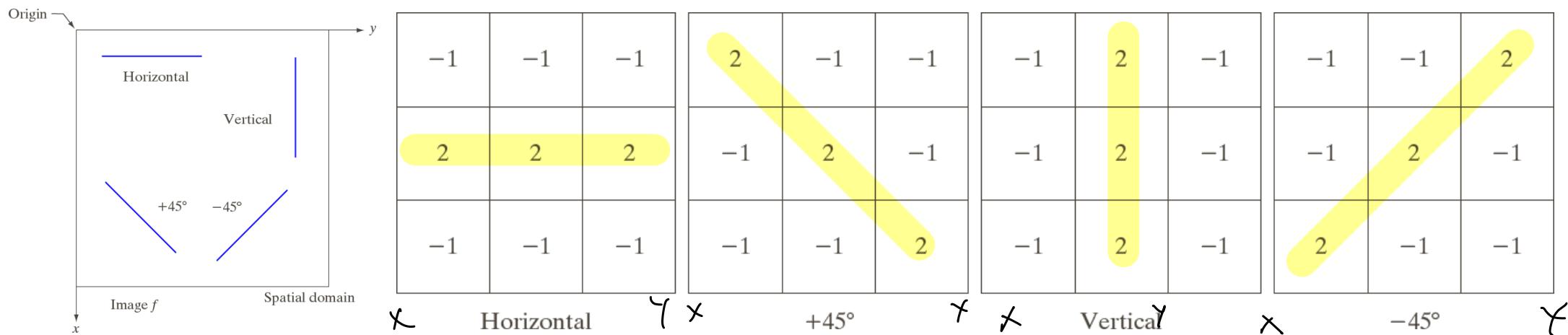
- Anche in questo caso si può utilizzare il **Laplaciano**, ma bisogna **gestire la doppia risposta** della derivata seconda
- A tal fine è possibile utilizzare
  1. il **valore assoluto** della risposta
  2. solo i **valori positivi** (eventualmente sogliati per attenuare l'effetto del rumore)
- Nel secondo caso si ottengono **linee più sottili**
- Se le **linee** sono **più ampie** rispetto alla **dimensione del filtro**, si otterrà l'**effetto** di una "**valle**" di valori nulli che separa le due linee

# LINEE



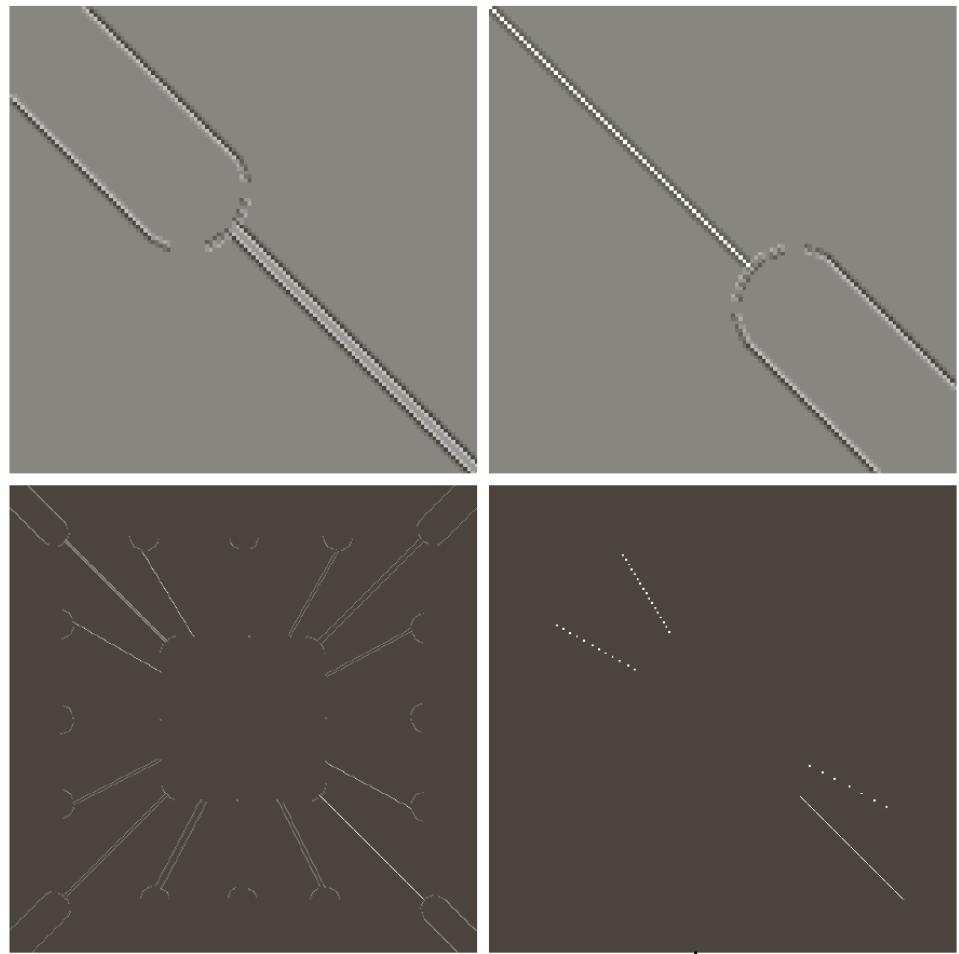
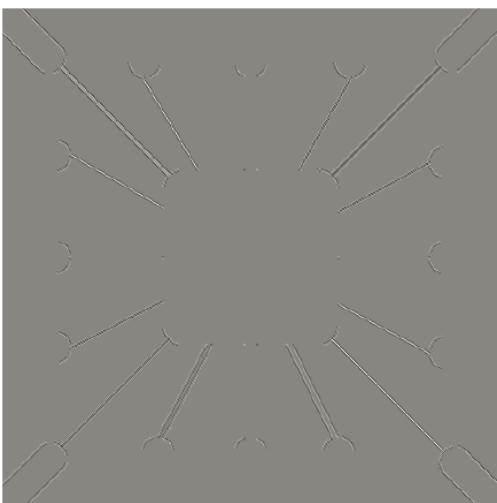
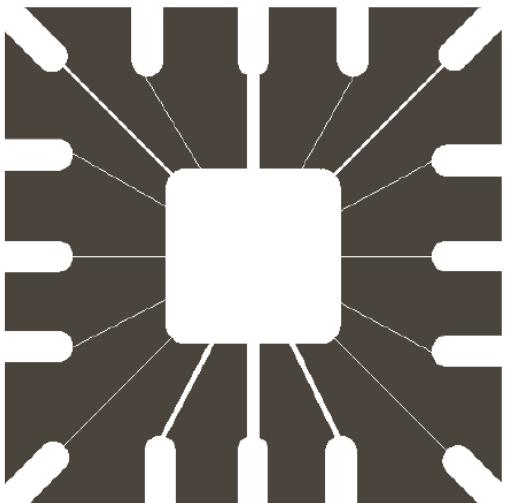
## LINEE DIREZIONI SPECIFICHE

- Il filtro **Laplaciano** è **isotropico**, ovvero la **risposta** è **indipendente** dalla **direzione** (rispetto alle direzioni orizzontale, verticale e diagonali)
- Per individuare rette con **direzioni specifiche** è possibile utilizzare **filtri specifici** all'immagine
- Per "trovare" la direzione dominante, selezionare la risposta più forte



Elim Parte II – Prof. A. Ferone

LINEE +45°



T' n.b. nessuno zeta ha quelli non  
è  $\pm 45^\circ$ , perché simili (sono meno  
forti)

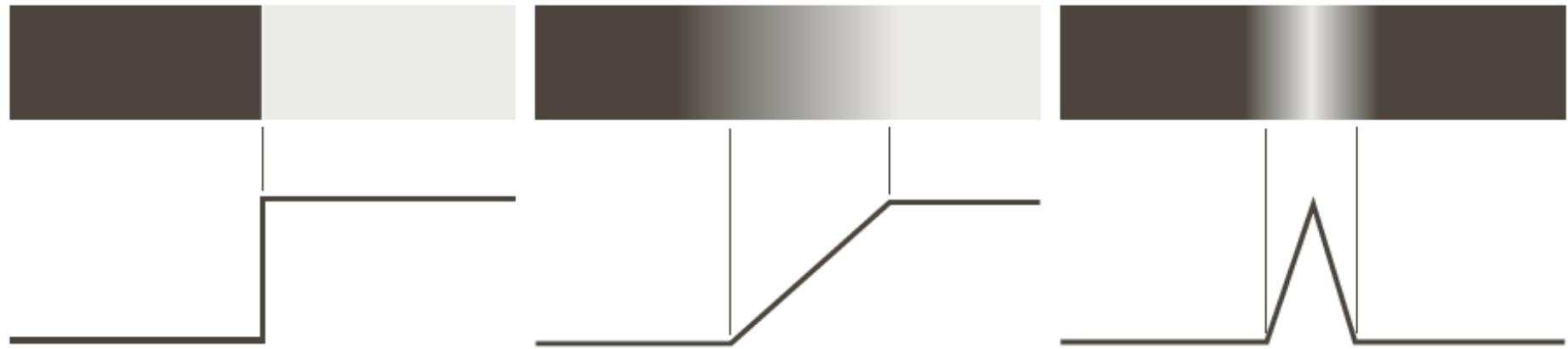
applico quindi una soglia  
x far risultare solo quelli  
è  $\pm 45^\circ$

Elim Parte II – Prof. A. Ferone

## MODELLI DI EDGE

- I **modelli di edge** sono **classificati** in base ai **profili di intensità**
- **Edge a gradino**: transizione tra due livelli di intensità ad una distanza ideale di 1 pixel
- **Edge a rampa**: edge sfocati e rumorosi appaiono come una **transizione graduale** e non netta come nel caso precedente. Diffilmente si tratta di linee sottili
- **Roof edge**: associato al bordo di una regione ha una base determinata dallo spessore e dalla sfocatura della linea
- Spesso nelle immagini si trovano più tipi di edge con profili diversi da quelli ideali

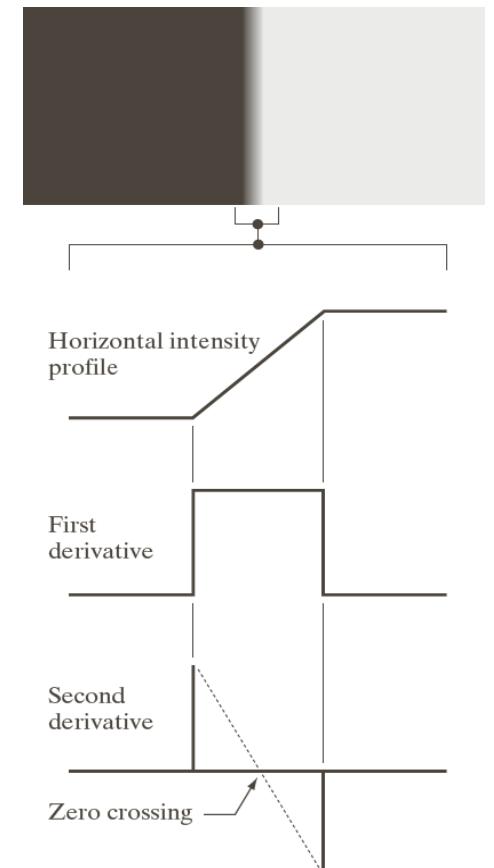
# MODELLI DI EDGE



Elim Parte II – Prof. A. Ferone

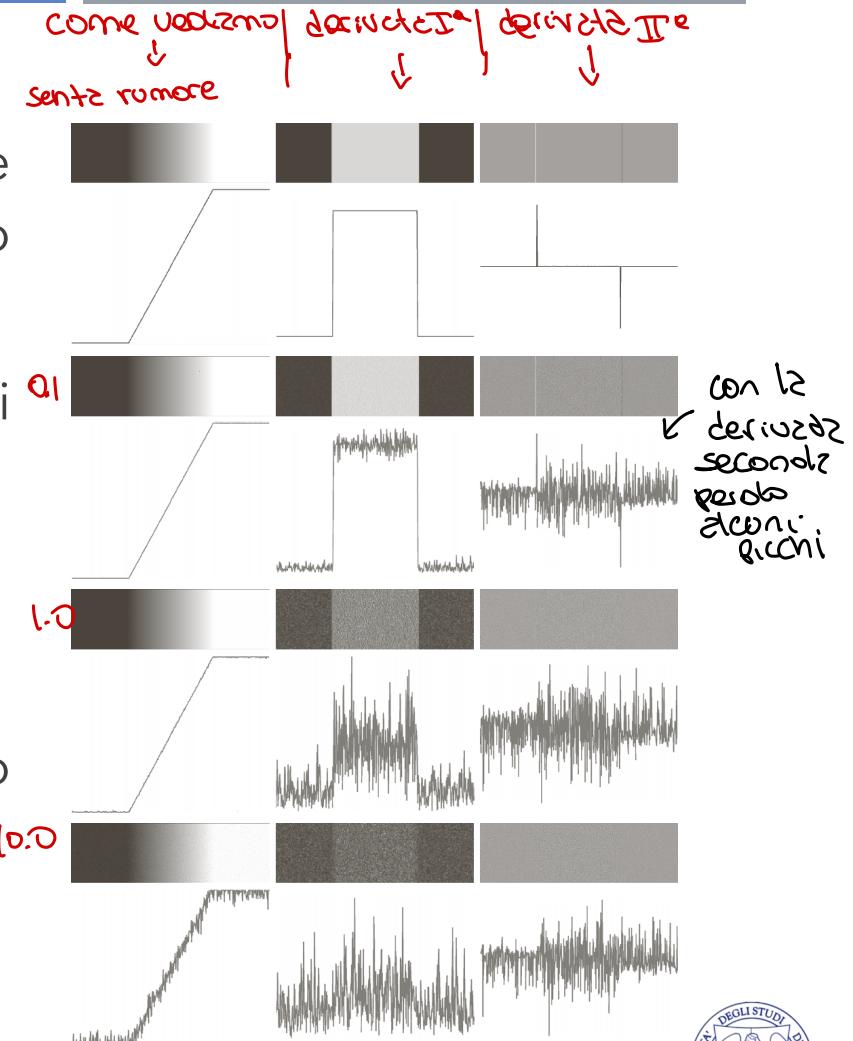
## EDGE A RAMPA

- La **derivata prima** può essere utilizzata per individuare un edge a rampa
- Il **segno della derivata seconda** può essere utilizzato per determinare se un **pixel** si trova sul **lato scuro** o **chiaro** di un edge
- La **derivata seconda** produce **due valori** per ogni edge
- Lo **zero-crossing** (attraversamento dello zero) può essere utilizzato per **trovare il centro di un edge**



## EDGE A RAMPA CON RUMORE

- **Rumore gaussiano** con media nulla e deviazione standard 0.1, 1.0 e 10.0 (poco visibile nelle immagini)
- Passaggi fondamentali per l'individuazione di edge
  1. Applicare lo **smoothing** per ridurre il rumore
  2. Individuare i **punti di edge**
  3. **Selezionare** solo i **punti di edge** che fanno **parte di un edge**



## INDIVDUAZIONE BASATA SU GRADIENTE

- Il gradiente è un **vettore** formato dalle **derivate parziali**

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Il vettore **gradiente punta** nella **direzione di massima variazione**
- La **magnitudo del gradiente**,  $M(x, y)$  è un'immagine delle stesse dimensioni di  $f()$

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

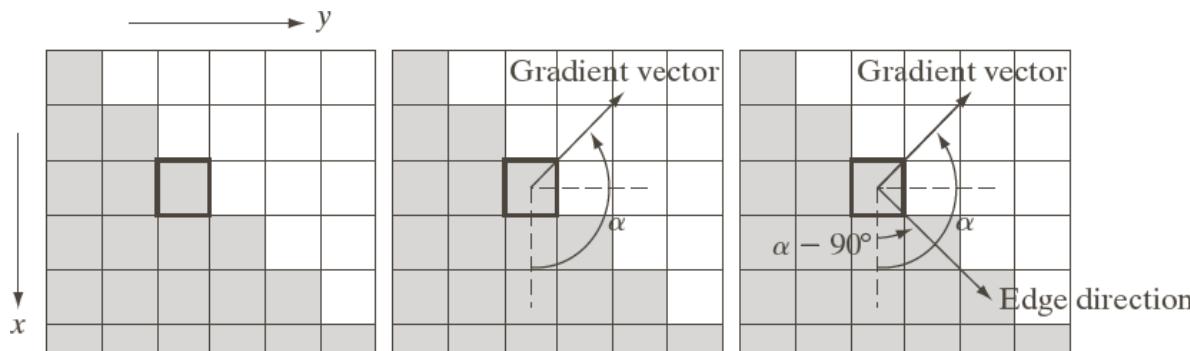
$$M(x, y) \approx |g_x| + |g_y|$$

## 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**



quindi l'edge è a  
90° rispetto al  
vettore

Elim Parte II – Prof. A. Ferone

## OPERATORI DI GRADIENTE

- Definizioni base:

$$g_x(x, y) = f(x + 1, y) - f(x, y)$$

$$g_y(x, y) = f(x, y + 1) - f(x, y)$$

$$g_x: \begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array}$$

$$g_y: \begin{array}{|c|} \hline -1 \\ \hline 1 \\ \hline \end{array}$$

- Operatori di Roberts:

$$g_x(x, y) = f(x + 1, y + 1) - f(x, y)$$

$$g_y(x, y) = f(x, y + 1) - f(x - 1, y)$$

$$g_x: \begin{array}{|c|c|} \hline -1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$$

$$g_y: \begin{array}{|c|c|} \hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array}$$

Elim Parte II – Prof. A. Ferone

# OPERATORI DI GRADIENTE

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Prewitt

0	1	1
-1	0	1
-1	-1	0

-1	-1	0
-1	0	1
0	1	1

Prewitt

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

orientati

ottengo Sx

Sobel

verticali

Elim Parte II – Prof. A. Ferone

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2

+45°

Sobel

-45°

Cioé applicando il filtro all'immagine, in output avrò le stesse immagine delle stesse dimensioni, in cui ogni pixel contiene la componente  $S_x \circ S_y$  applicato l'operatore di Sobel orizzontale e verticale

J unz volte ottenute quindi le 2 immagini (siccome ho applicato 2 filtri sobel - c1 )  
Applico la magnitudo delle due posizioni:  $M(i,j) = \sqrt{g_x(i,j)^2 + g_y(i,j)^2}$   
e poi a ogni pixel mi calcolo la direzione  $\theta(i,j) = \tan^{-1} \left| \frac{g_y(i,j)}{g_x(i,j)} \right|$

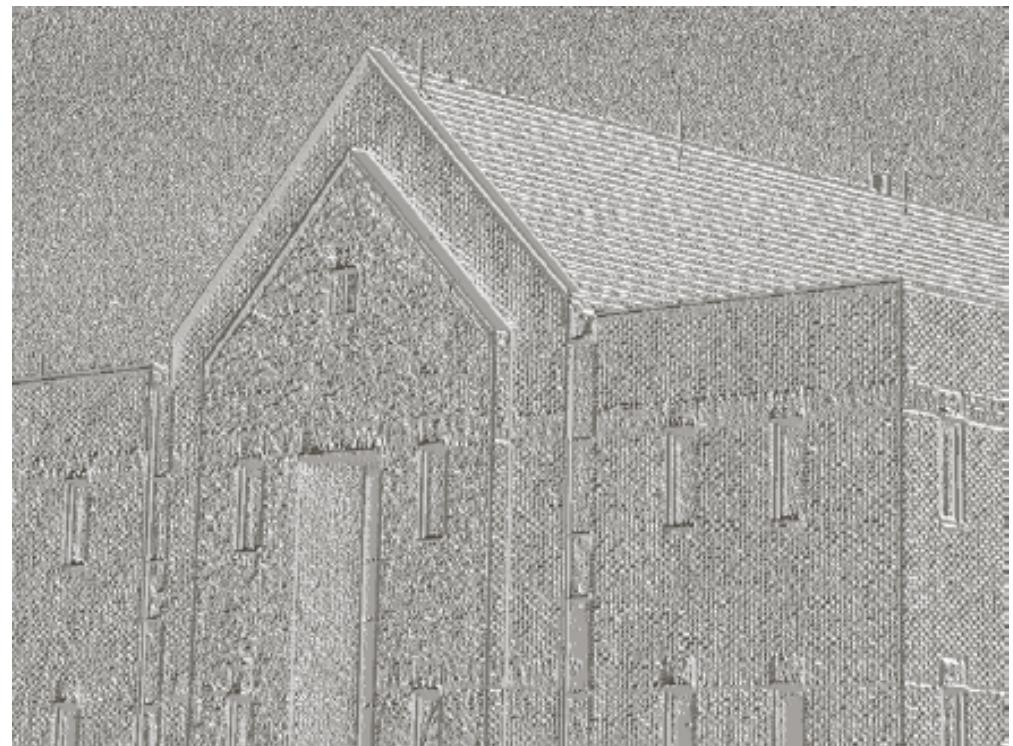
## OPERATORI DI GRADIENTE



# OPERATORI DI GRADIENTE



↑ immagine  
magnitudo



↑ immagine  
orientazione

Elim Parte II – Prof. A. Ferone

## INDIVDUAZIONE BASATA SU GRADIENTE

- La risoluzione dell'immagine può determinare l'esito dell'operazione di estrazione degli edge
- I **dettagli fini** sono assimilati al **rumore**, producendo molti **punti edge**
- Prima di applicare un estrattore di edge è buona norma effettuare uno **smoothing** dell'immagine



## INDIVDUAZIONE BASATA SU GRADIENTE

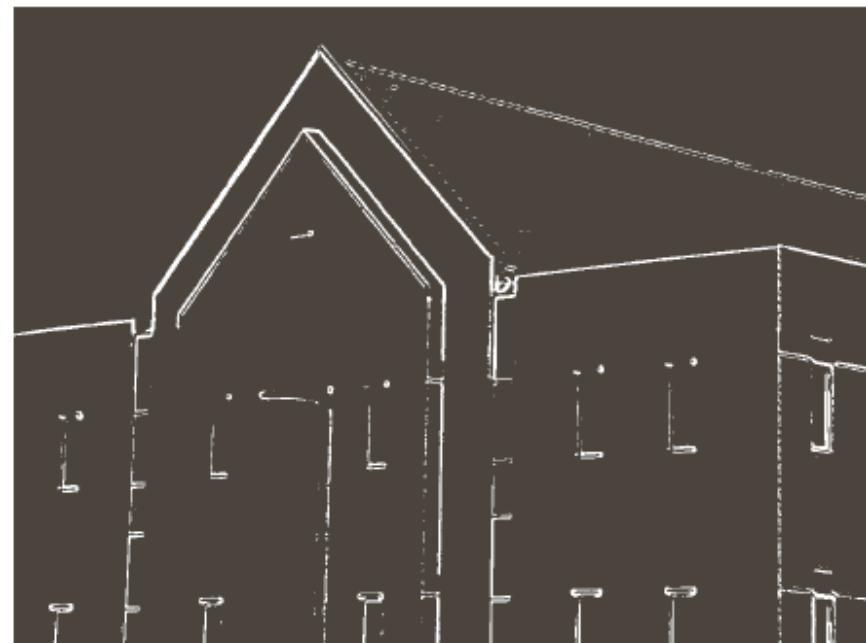
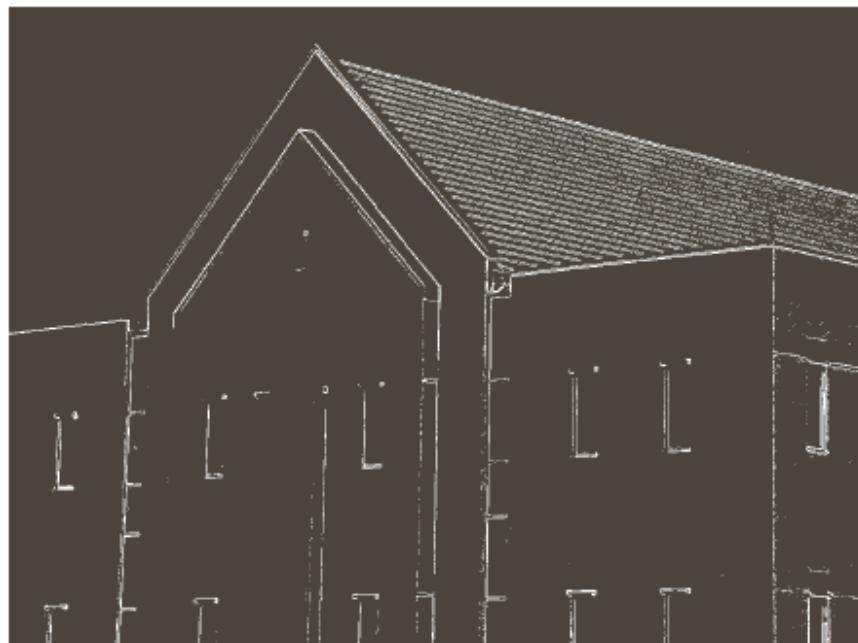
- Per enfatizzare gli **edge diagonali**, è necessario utilizzare le maschere specifiche



Elim Parte II – Prof. A. Ferone

## GRADIENTE E THRESHOLDING

- Per ottenere **immagini gradiente** più "**pulite**", è possibile utilizzare anche la tecnica del **thresholding**, ovvero utilizzare solo i pixel la cui risposta supera una determinata soglia



Elim Parte II – Prof. A. Ferone

## MARR-HILDRETH

Una delle tecniche di smoothing è il filtro GAUSSIANO quindi Laplaciano + GAUSS fà smoothing + individuazione di edge (dunque saranno attenuati i rumori)

- È possibile utilizzare dei **filtri** più **avanzati** che sfruttano le proprietà delle derivate prima e seconda
- Marr e Hildreth proposero il filtro  **$\nabla^2 G \circ \text{LoG}$  (Laplacian of Gaussian)**

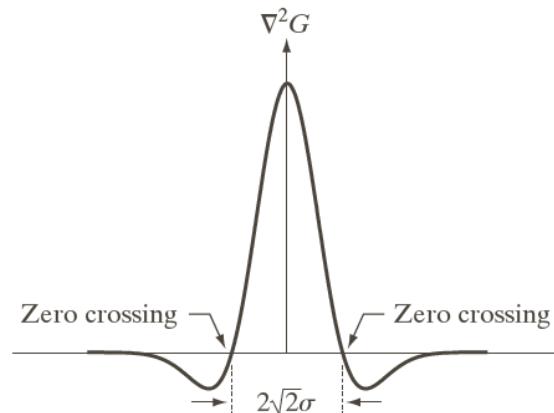
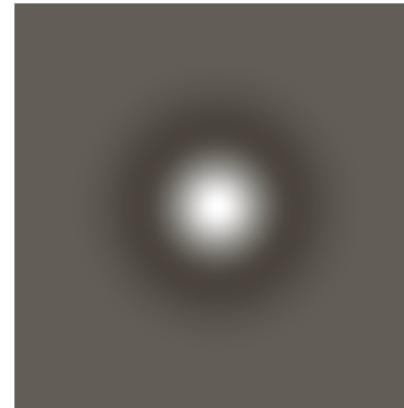
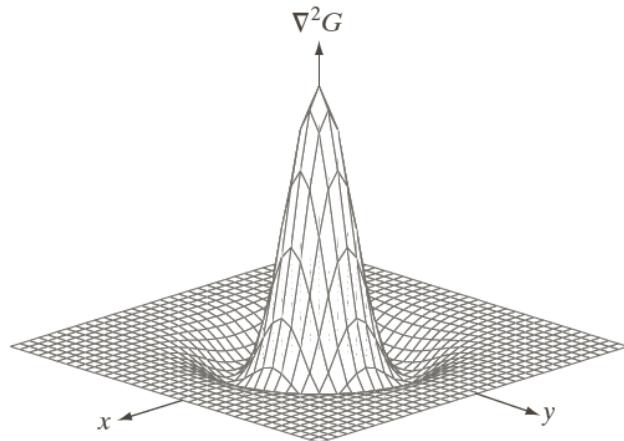
$$\nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial^2 x^2} + \frac{\partial^2 G(x, y)}{\partial^2 y^2}$$

{  
e le derivate  
seconda del  
filtro GAUSSIANO}

$$\nabla^2 G(x, y) = \left[ \frac{x^2 + y^2 + 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Anche detto **operatore a sombrero**
- È possibile modificare l'**ampiezza** modificando il valore  $\sigma$

LOG



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

Elim Parte II – Prof. A. Ferone

## FILTRO GAUSSIANO

- La dimensione del filtro **n** deve essere scelta pari al più piccolo intero dispari maggiore o uguale  $6\sigma$
- 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 (ampiezza cerniere Gauss)  
    int          ktype = CV_32F   // Type for filter coefficients  
)  
mi ritorna la matrice  
  
filter2D(src,dst,CV_32F,getGaussianKernel(n,sigma));
```

## PROPRIETÀ DEL LOG

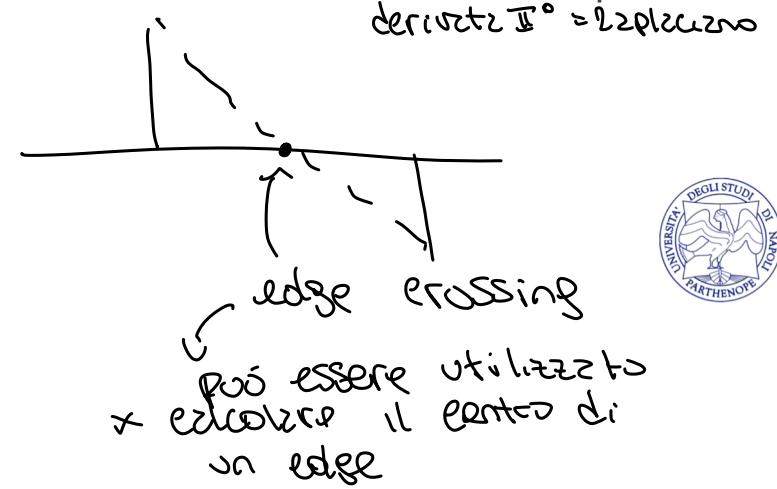
- La funzione Gaussiana ha l'effetto di **sfocare l'immagine**
- Il **Laplaciano** è un operatore **isotropico**
- L'algoritmo di Marr-Hildreth consiste nella convoluzione del filtro LoG con l'immagine di input

$$g(x, y) = [\nabla^2 G(x, y)] * f(x, y)$$

- equivale a

$$g(x, y) = \nabla^2 [G(x, y) * f(x, y)]$$

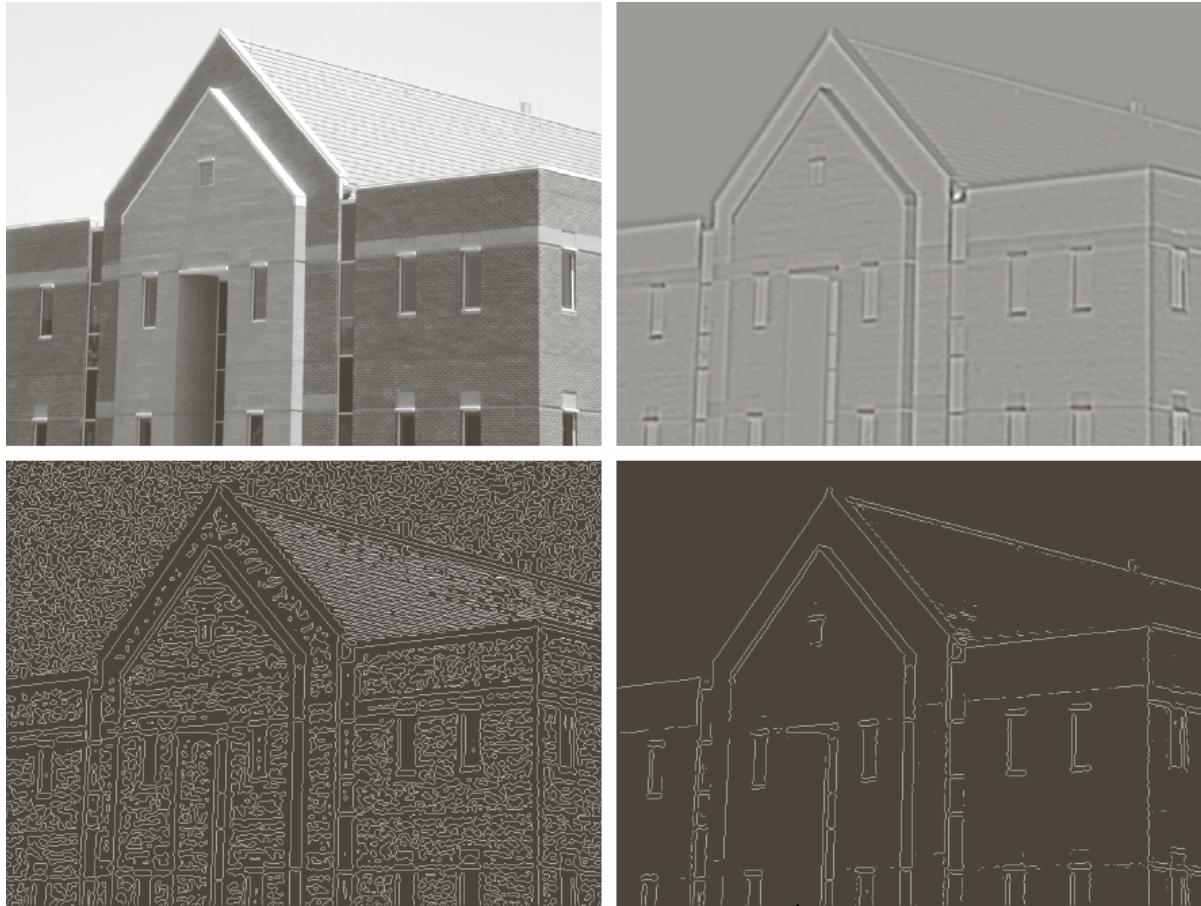
- Ovvero filtrare l'immagine con un **filtro Gaussiano** di dimensione nxn e poi applicare il **filtro Laplaciano**
- Infine bisogna **individuare** gli **zero-crossing**



## ZERO-CROSSING

- A causa del **rumore** o delle **approssimazioni computazionali**, la ricerca di  $g(x,y)=0$  potrebbe non dare il risultato atteso (individuazione degli edge)
- Si considera un **intorno 3x3** centrato in un **pixel p** dell'immagine filtrata
- Uno **zero-crossing** in p implica che i **segni** di almeno **due pixel vicini** opposti **siano diversi** (*vedo a prendere un pixel dell'intorno e vedo se i suoi adiacenti hanno un segno opposto al suo = ho trovato un pixel di edge e lo marco come TRUE = 1*)
  - Sopra-sotto, destra-sinistra e le due diagonali
- Inoltre il **valore assoluto** della **loro differenza** deve superare una **soglia**
- **LoG** può essere approssimato con **DoG (Difference of Gaussians)**

LOG



↑ LOG

Elim Parte II – Prof. A. Ferone

↑ LOG con zero crossing

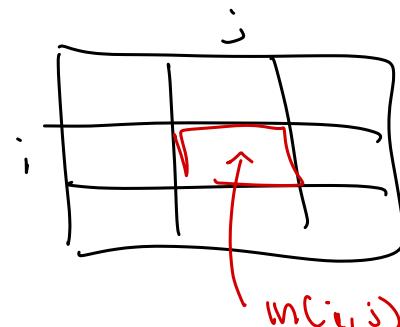
## ESERCIZI

- Utilizzando le funzioni di OpenCV implementare il LoG
- Implementare una funzione che trovi i punti di zero-crossing

threshold in >= ec

1. GLI PASSO LoG (Fai la funzione che sta sopre)

```
th= max(in)*.4
for i=1:in.rows
    for j=1:in.cols
        N = intorno 3x3
        m = min(N)
        M = max(N)
        if(in(i,j)>0)
            flag=m<0?true:false
        else
            flag=M>0?true:false
        if(max-min>th && flag=true)
            out=1
```



vedo se le variazioni  
di intensità sono estreme

$$1-massima \times risultato = 255$$

Elim Parte II – Prof. A. Ferone

edge.CPP

N.b nel suo codice phase è l'arcotangente

