

FILTRAGGIO SPAZIALE LINEARE

Il filtraggio spaziale consente di modificare un'immagine (es. per migliorarne la qualità o per estrarre informazione di interesse) variando il livello di grigio di ogni pixel in funzione dei livelli di grigio dei pixel vicini. Si parla di filtraggio lineare quando il valore del pixel in output è il risultato della combinazione lineare dei pixel in un suo intorno. Il filtraggio lineare viene effettuato tramite la convoluzione 2-D dell'immagine $A(x,y)$ con la matrice $k(x,y)$ dei pesi che è quindi un filtro lineare bidimensionale.

$$B(x,y)=k(x,y)*A(x,y)$$

Come si calcola la convoluzione 2D?

Si considerino la matrice $A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix}$

e il filtro $k = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$

Il calcolo della convoluzione in un pixel (r,c) richiede di eseguire quattro passi:

- 1) Ruotare di 180° la matrice k :

$$h = \text{rot}90(k,2);$$

ruota k di $90^\circ \times 2$

$$h = \begin{bmatrix} 2 & 9 & 4 \\ 7 & 5 & 3 \\ 6 & 1 & 8 \end{bmatrix}$$

- 2) Determinare il pixel centrale di h .

$$\text{centro} = \text{floor}((\text{size}(h)+1)/2)$$

$$\text{size}(h) = 3, 3$$

$$+1 = 4, 4$$

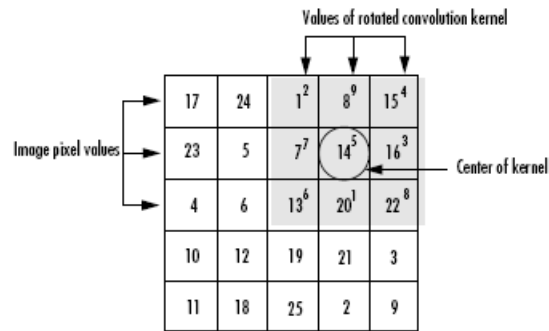
$$/2 = 2, 2$$

$$\text{floor} = 2, 2$$

arrotonda all'intero più vicino verso $-\infty$

- 3) Sovrapporre il pixel centrale di h al pixel (r,c) della matrice A e moltiplicare punto a punto i pixel corrispondenti di h e di A .
- 4) Sommare i risultati delle moltiplicazioni e assegnare il valore ottenuto al pixel (r,c) della matrice risultato B .

Quindi la convoluzione in (r,c)=(2,4) si ottiene come:



$$B(2,4) = 1*2+8*9+15*4+7*7+14*5+16*3+13*6+20*1+22*8=575$$

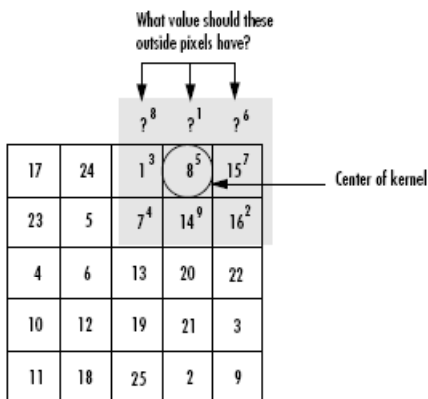
Come si calcola la convoluzione 2D e quindi come si esegue il filtraggio spaziale lineare in Matlab?

Si utilizza la funzione *imfilter.m*

Il comando ***B=imfilter(A,k,'conv')*** effettua il filtraggio spaziale lineare della matrice A mediante il filtro k. Note:

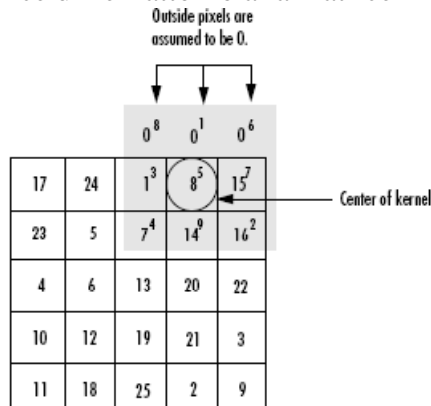
- Padding dei bordi

Sovrapponendo la matrice filtro ai pixel di bordo della matrice immagine, si ha che alcuni elementi del filtro cadono all'esterno dell'immagine.

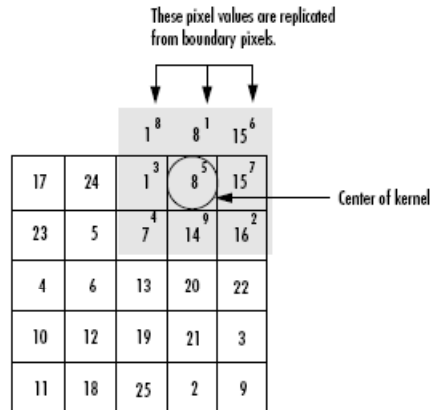


Diverse possibilità:

- Default: viene creata una cornice di zeri attorno alla matrice A (zero-padding)



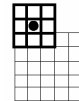
- `imfilter(A,k,'conv','symmetric')` o `imfilter(A,k,'conv','replicate')`: vengono replicati i pixel più esterni dell'immagine



- `imfilter(A,k,'conv','circular')`: si periodicizza l'immagine (convoluzione circolare).
- Dimensione dell'immagine B in uscita.
Il risultato della convoluzione 2D è una matrice di dimensione $\text{size}(A)+\text{size}(k)-1$. L'immagine generata in uscita dalla funzione `imfilter` può essere:
 - delle stesse dimensioni dell'immagine A in ingresso (default). Il pixel centrale del filtro viene quindi sovrapposto ai soli pixel dell'immagine A originaria:



- posta uguale al risultato della convoluzione (solo in questo caso la convoluzione è commutativa), `imfilter(A,k,'conv','full')`:



- La funzione `imfilter` opera anche su immagini 3D/4D e con filtri sia 2D che 3D.

ESERCIZIO

Filtrare l'immagine contenuta nel file 'rattorace.gif' con il filtro

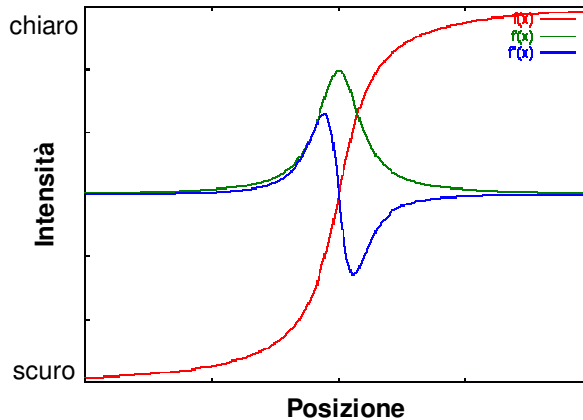
`k=ones(15,15)/(15^2);`

provando le diverse opzioni della funzione `imfilter`.

ESTRAZIONE DI CONTORNI

Lungo i contorni di un'immagine si ha una significativa variazione di intensità. Come possono essere individuati i contorni in un'immagine?

Dominio 1D continuo, funzione f(x)



I contorni possono essere individuati cercando:

- i massimi (minimi) della derivata prima
- i passaggi per lo zero (variazioni di segno) della derivata seconda.

Dominio 2D continuo, funzione f(x,y)

I contorni possono essere individuati cercando:

- i massimi del modulo del vettore gradiente

$$\nabla f(x, y) = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right] \quad |\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2}$$

- i passaggi per lo zero del Laplaciano

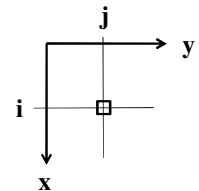
$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Dominio 2D discreto

Per estrarre i contorni occorre utilizzare dei filtri che approssimino il calcolo del gradiente e del Laplaciano: si approssimano le derivate con le differenze finite.

- Le componenti del gradiente nel pixel (i,j) possono essere approssimate come:

$$\frac{\partial f}{\partial x} \approx f(i-1, j) - f(i+1, j); \quad \frac{\partial f}{\partial y} \approx f(i, j-1) - f(i, j+1)$$



- Il Laplaciano nel pixel (i,j) può essere approssimato come:

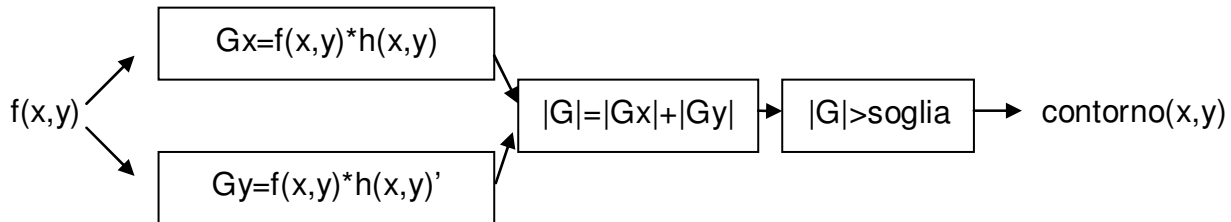
$$\frac{\partial^2 f}{\partial x^2} \approx (f(i-1, j) - f(i, j)) - (f(i, j) - f(i+1, j)) = f(i-1, j) - 2f(i, j) + f(i+1, j)$$

$$\frac{\partial^2 f}{\partial y^2} \approx \dots = f(i, j-1) - 2f(i, j) + f(i, j+1)$$

$$\nabla^2 f(x, y) \approx f(i-1, j) + f(i+1, j) + f(i, j-1) + f(i, j+1) - 4f(i, j)$$

- METODI BASATI SUL CALCOLO DEL GRADIENTE**

Sia $h(x,y)$ un filtro per il calcolo delle differenze finite lungo x , es. $h(x,y)=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$



ESERCIZIO

Provare ad estrarre i contorni dell'immagine tac contenuta nel file `pet_tac.mat`.

```

gx=imfilter(tac,h);
gy=imfilter(tac,h');
g=abs(gx)+abs(gy);
contorno=g;
contorno(contorno<=soglia)=0;
contorno(contorno>soglia)=1;
  
```

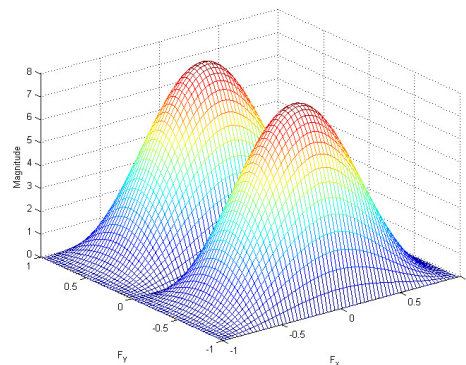
I metodi per l'estrazione di contorni basati sul calcolo del gradiente differiscono per il filtro utilizzato.

- metodo di sobel.

Calcola il gradiente dell'immagine applicando la maschera di sobel; considera come punti del contorno quelli in cui il modulo del gradiente supera una certa soglia.

```

h=fspecial('sobel');
h= 1  2  1
    0  0  0
   -1 -2 -1
  
```

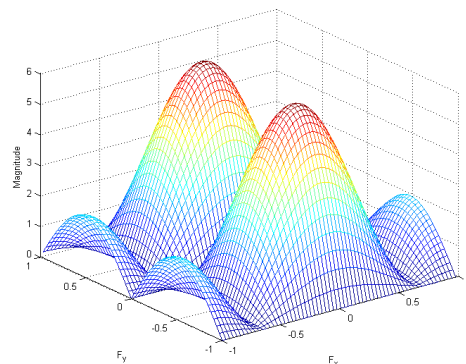


- metodo di prewitt

Come sobel, ma per calcolare il gradiente utilizza la maschera di prewitt.

```

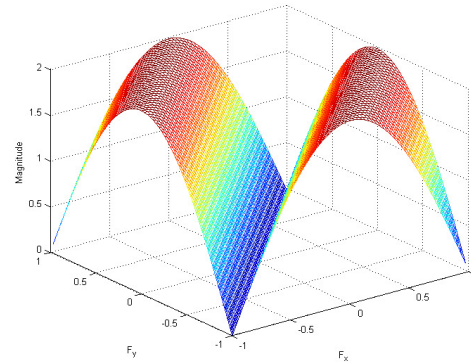
h=fspecial('prewitt');
h= 1  1  1
    0  0  0
   -1 -1 -1
  
```



3. metodo di roberts

Come sobel, ma per calcolare il gradiente utilizza le maschere di roberts.

$$h = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



4. metodo di canny

Calcola il gradiente applicando all'immagine la derivata di un filtro gaussiano. Per estrarre i contorni applica al gradiente due diverse soglie, una per i contorni deboli e una per i contorni forti. I contorni deboli vengono inclusi nel contorno estratto solo se connessi ai contorni forti.

FUNZIONE DI MATLAB edge.m

Riceve in ingresso un'immagine di tipo intensità o binario e crea in uscita l'immagine binaria dei contorni. Consente di scegliere tra diversi metodi per la detezione dei contorni.

*contorno=*edge(*im*, 'sobel', *soglia*, *direzione*);

Tramite il parametro *direzione* ('horizontal', 'vertical', 'both') è possibile selezionare la direzione lungo la quale cercare i contorni.

*contorno=*edge(*im*, 'canny', *soglia*, *sigma*);

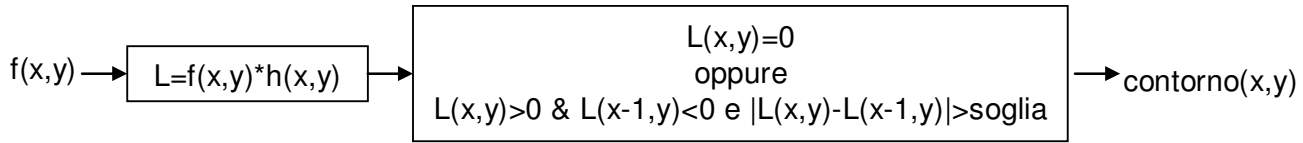
Soglia è un vettore di due elementi [*soglia* bassa, *soglia* alta]. Se viene specificato uno scalare *S* si avrà automaticamente [*0.4*S*, *S*]. *Sigma* è la deviazione standard del filtro gaussiano (default=1).

ESERCIZIO

Provare ad estrarre i contorni delle immagini tac e pet contenute nel file *pet_tac.mat* utilizzando la funzione *edge* ed utilizzando i diversi metodi basati sul gradiente.

- METODI BASATI SUL CALCOLO DEL LAPLACIANO**

Sia $h(x,y)$ un filtro per il calcolo del Laplaciano di un'immagine, es. $h(x,y)=\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$



ESERCIZIO

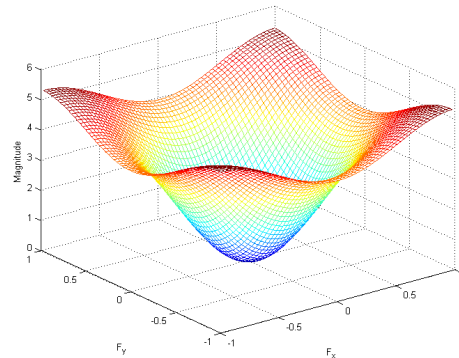
Provare ad estrarre i contorni dell'immagine tac contenuta nel file `pet_tac.mat`. Utilizzare la funzione `edge.m` con l'opzione `'zerocross'`.

```
contorno=edge(tac,'zerocross',soglia,h);
```

I metodi per l'estrazione di contorni basati sul calcolo del Laplaciano differiscono per il filtro utilizzato.

1. $h=fspecial('laplacian')*6$

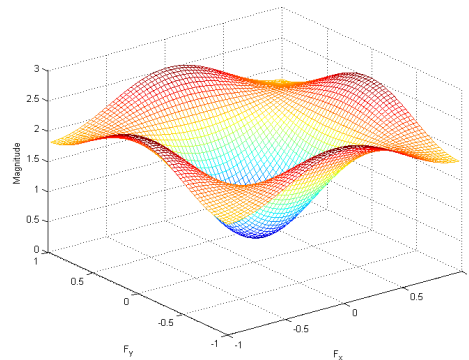
$$h = \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$



2. laplaciano di gaussiana

Applica il filtro laplaciano di gaussiana all'immagine ed estrae i contorni come punti di passaggio per lo zero sull'immagine risultante.

```
lapl=fspecial('laplacian');
gauss=fspecial('gaussian');
prova_log=conv2(lapl,gauss,'same');
```

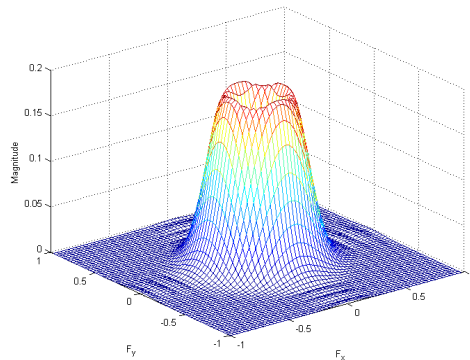


FUNZIONE DI MATLAB edge.m

`contorno=edge(im,'zerocross',soglia,h);`

`contorno=edge(im,'log',soglia,sigma);`

Sigma è la deviazione standard del filtro log (default=2). La dimensione del filtro è NxN con $N=\text{ceil}(3*\text{sigma})*2+1$.



ESERCIZIO

Provare ad estrarre i contorni delle immagini tac e pet contenute nel file pet_tac.mat utilizzando la funzione edge ed utilizzando i diversi metodi basati sul Laplaciano.

Provare ad applicare i diversi metodi per l'estrazione di contorni alle seguenti immagini:

- *MANO.jpg*
- *disco1.jpg*
- *fant_pet.mat*
- *mri.mat* (estrarre una fetta)
- *SPECT.jpg*