ELABORAZIONE DI SEGNALI E IMMAGINI

M. Bertero – P. Boccacci

bertero@disi.unige.it
boccacci@disi.unige.it

Filtraggio di un segnale

Al fine di migliorare la qualità di un segnale digitale una tecnica di primaria importanza è il filtraggio. Con il quale si possono enfatizzare alcune caratteristiche o rimuoverne altre.

Essenzialmente il filtraggio è una funzione di intorno, nel quale il valore che assume un bin nel segnale in uscita è determinato dall'applicazione di un operatore che agisce sui valori dei bins ad esso circostanti nel segnale d'ingresso. Per questo tale operatore viene detto operatore locale. Il filtro agisce su una finestra, solitamente di ampiezza molto più piccola della durata del segnale.

Si ditinguono due classi di filtri: lineari e non lineari.

Nei primi l'operatore definito sulla finestra dà in uscita un valore che è una combinazione lineare dei valori dei bin compresi nella finestra. Si possono definire filtri lineari che puliscono il segnale dal rumore oppure che esaltano le discontinuità.

Nei secondi non è possibile definire un operatore lineare; solitamente sono operatori di rango, cioè operatori che agiscono sui valori dei bin dopo averli ordinati.

Vedremo che la differenza sostanziale tra i due tipi di filtri è che, mentre per i primi si può applicare la trasformata di Fourier con tutte le sue proprietà, nei secondi questa operazione non è possibile.

Filtri lineari 1D

Finora abbiamo imparato a campionare un segnale e abbiamo simulato il rumore che, inevitabilmente lo affligge, in misura più o meno importante a seconda delle caratteristiche dello strumento.

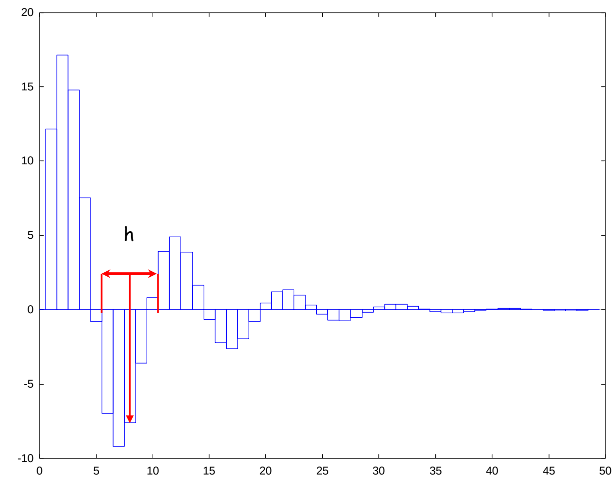
Una misura di quanto è importante il segnale è data dal rapporto segnale rumore. Questa misura, solitamente è valutata in Decibel ed è definita come:

$$SNR = 20 \log \left(\frac{\text{varianza del segnale}}{\text{varianza del rumore}} \right) \text{db}$$

SNR= Signal to Noise Ratio

Studiamo ora come pulire un segnale dal rumore attraverso i filtri lineari e non lineari.

Il più semplice filtro lineare è quello che viene chiamato "media mobile". In pratica si fa scorrere una finestra di ampiezza molto piccola (e solitamente dispari) sul segnale e si sostituisce al valore del bin corrispondente al centro della finestra il valore che deriva dalla media dei bin compresi nella finestra.



Formalmente se *f[m]* è il segnale e *h[m']* i valori dei pesi sulla finestra , w la larghezza della finestra , w_0=(w-1)/2 (w dispari)

$$f_{filtrato}[m] = \sum_{m'=-w_0}^{m'=w_0} f[m-m']h[m'] = \frac{1}{w} \sum_{m'=-w_0}^{m'=w_0} f[m-m']$$

Alcune considerazioni

E' chiaro che quando la finestra si trova agli estremi dell'intervallo di campionamento del segnale, la somma si estende al di fuori del segnale stesso.

Per ovviare a questo inconveniente si possono operare alcune scelte differenti

1) Non calcolare i valori dei bordi. Il segnale in uscita è definito in un numero di punti N-2w_0

2) Aggiungere degli zeri: zero-padding

3) Pensare il segnale prolungato per periodicità. Come vedremo quest'ultima soluzione permette di estendere l'utilizzo della trasformata di Fourier al filtraggio.

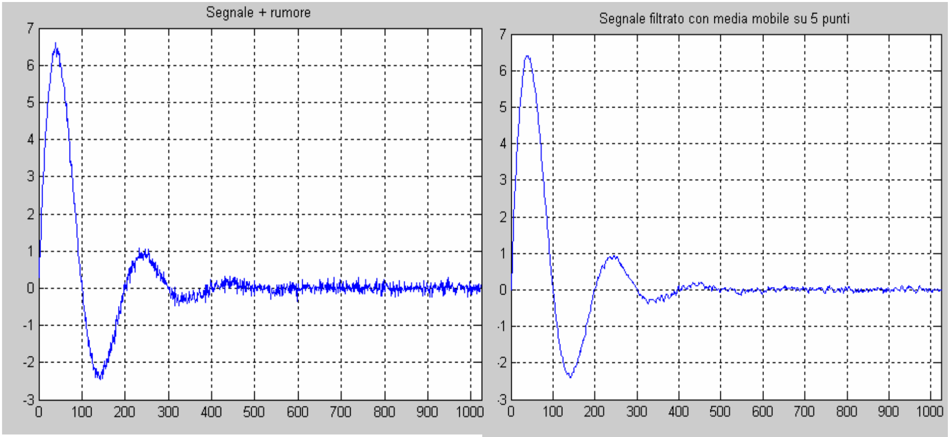
$$f_{filtrato}[m] = \sum_{m'=-w_0}^{m'=w_0} f[m-m']h[m'] \quad = \frac{1}{w}\sum_{m'=-w_0}^{m'=w_0} f[m-m']$$

Un'altra considerazione va fatta nell'osservare che il filtro agisce, in pratica, ribaltato rispetto al punto centrale della finestra infatti, se w=3,w_0 =1:

$$f_{filtrato}[3] = \frac{1}{w}\sum_{m'=-w_0}^{m'=w_0} f[m-m'] = \frac{1}{3}\left(f[4] + f[3] + f[2]\right)$$

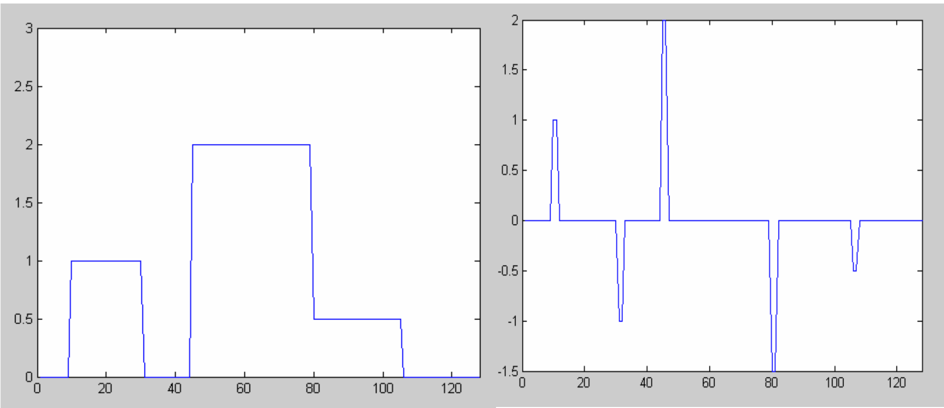
Esempi filtri 1D 1/2

h=1/5[1 1 1 1 1]



Esempi filtri 1D 2/2

h=[1 0 -1]



Prodotto di convoluzione

Date due successioni periodiche *h[m], f[m]* di periodo *N*, il loro prodotto di convoluzione indicato con *h*f*, è la successione periodica di periodo *N* definita da:

$$g[m] = (h * f)[m] = \sum_{m'=0}^{N-1} h[m-m']f[m']$$

Vediamo ora le differenze e le analogie con lil filtraggio definito precedentemente

$$f_{filtrato}[m] = \sum_{m'=-w_0}^{m'=w_0} f[m-m']h[m']$$

Le uniche differenze sono l'estensione della sommatoria a tutto l'intervallo e lo scambio tra *f* e *h* .

Se ora noi pensiamo al filtro, che era limitato alla finestra di ampiezza w, come parte di un vettore periodico di periodo N (dove gli elementi mancanti sono stati riempiti con zeri) possiamo estendere la sommatoria a tutto *[0,N-1]*.

Se ora dimostriamo che il prodotto di convoluzione è commutativo, abbiamo interpretato la formula del prodotto di convoluzione di due successioni periodiche come un filtraggio. (ovvero il filtraggio come il prodotto di convoluzione)

Il prodotto di convoluzione è commutativo cioè:

$$h*f = f*h$$

Dimostrazione: Posto m"=m-m' si ha:

$$(f*h)[m] = \sum_{m'=0}^{N-1} f[m-m']h[m'] = \sum_{m''=m}^{m-N+1} f[m'']h[m-m''] =$$

$$= \sum_{m''=0}^{N-1} h[m-m'']f[m''] = (h*f)[m]$$

Nel caso precedente si avrebbe h[-1]=h[0]=h[1]=1/3 ,

Riportando tali valori in un vettore periodico di lunghezza N, si ha h[0]=h[1]=1/3 h[2]=h[3]=...=h[N-2]=0, h[N-1]=1/3.

Altre proprietà del prodotto di convoluzione

1) Se g=h*f allora g è periodica con periodo N:

$$g[m \pm N] = \sum_{m'=0}^{N-1} h[m-m' \pm N]f[m'] = \sum_{m'=0}^{N-1} h[m-m']f[m'] = g[m]$$

2) Proprietà commutativa

a*b=b*a

3) Proprietà associativa

a*(b*c)=(a*b)*c

4) Proprietà distributiva

a*(b+c)=a*b+b*c

Teorema di convoluzione

Il teorema di convoluzione permette di utilizzare la trasformata di Fourier. Infatti afferma:

Posto g=h*f vale la seguente relazione tra le DFT

G[k]=H[k]F[k]

Dimostrazione:

Dalla definizione di DFT:

$$G[k] = \sum_{m=0}^{N-1}\left(\sum_{m'=0}^{N-1} h[m-m']f[m']\right)e^{-i\frac{2\pi k}{N}m} = \qquad \text{scambio le sommatorie}$$

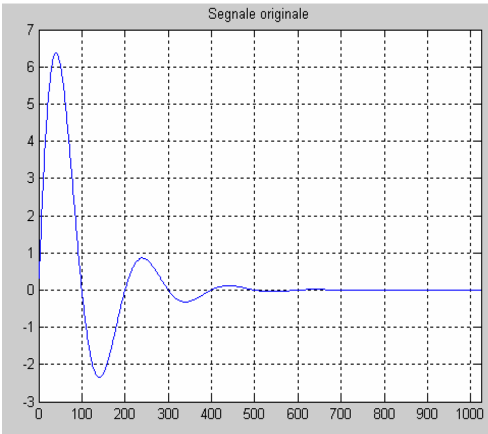
$$= \sum_{m'=0}^{N-1}\left(\sum_{m=0}^{N-1} h[m-m']e^{-i\frac{2\pi k}{N}m}\right)f[m'] = \qquad \text{pongo m"=m-m'}$$

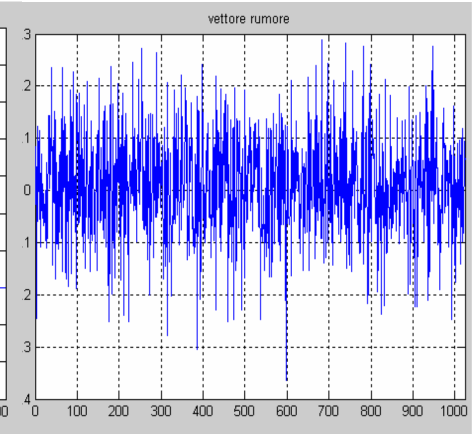
$$= \sum_{m'=0}^{N-1}\left(\sum_{m''=-m'}^{N-1-m'} h[m'']e^{-i\frac{2\pi k}{N}(m'+m'')}\right)f[m'] =$$

$$= \sum_{m'=0}^{N-1}\left(\sum_{m''=0}^{N-1} h[m'']e^{-i\frac{2\pi k}{N}m''}\right)f[m']e^{-i\frac{2\pi k}{N}m'} = H[k]F[k]$$

Esempi filtri 1D-Fourier

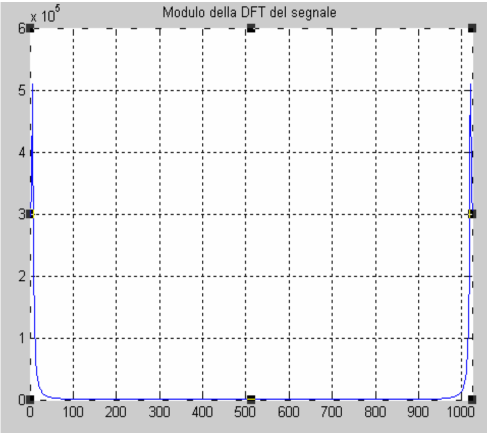
Utilizziamo la trasformata di Fourier per "ripulire" un segnale dal rumore. Pensiamo il rumore come guassiano "bianco". Vediamo nelle diapositive seguenti il significato di questo aggettivo.

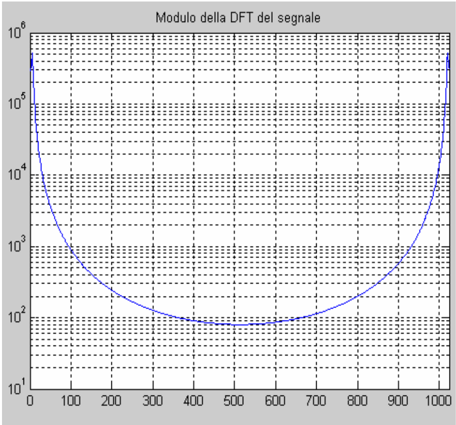




Esempi filtri 1D-Fourier

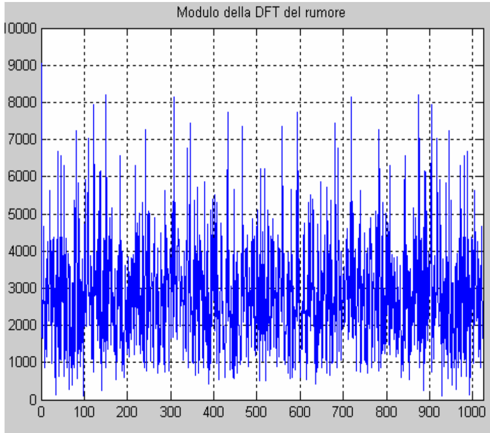
Calcoliamo la trasformata di Fourier del segnale e ne visualizziamo il modulo in scala lineare e logartmica

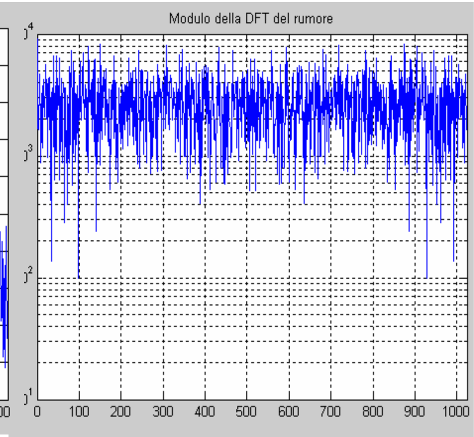




Esempi filtri 1D-Fourier

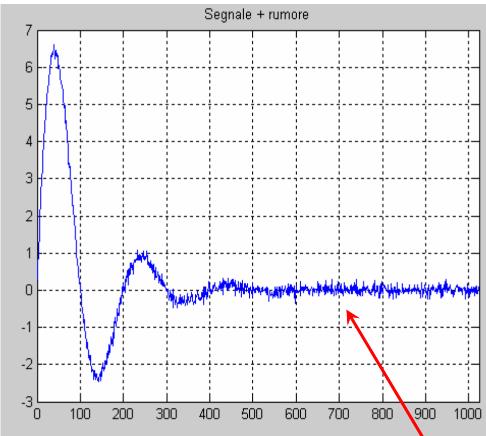
Calcoliamo la trasformata di Fourier del rumore e ne visualizziamo il modulo in scala lineare e logartmica

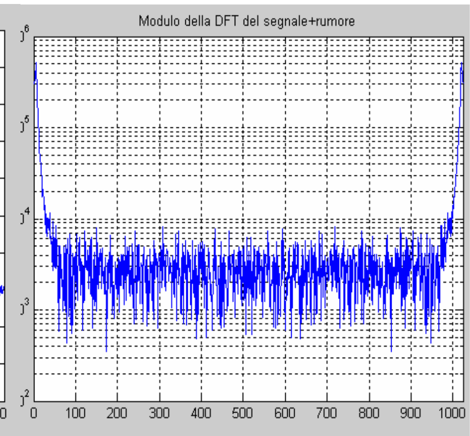




Esempi filtri 1D-Fourier

Ora visualizziamo il segnale con il rumore e la sua trasformata in modulo (scala logaritmica)

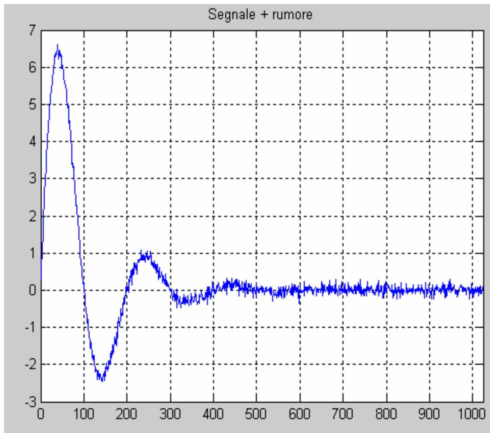


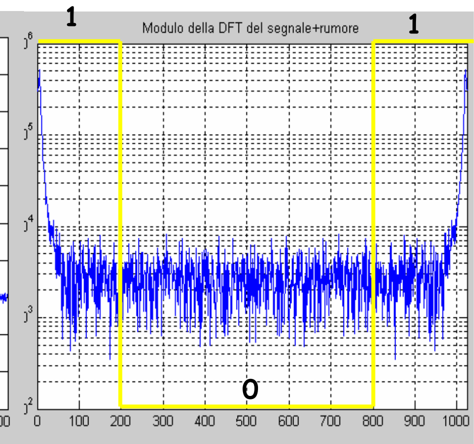


Dove il segnale è più basso il rumore è predominante

Esempi filtri 1D-Fourier

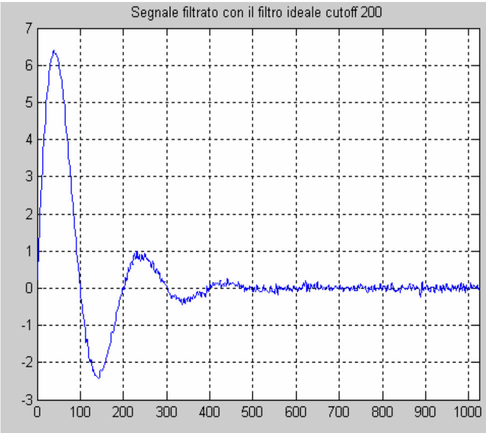
Proviamo a filtrare il rumore moltiplicando la trasformata di Fourier del segnale rumoroso per una finestra.

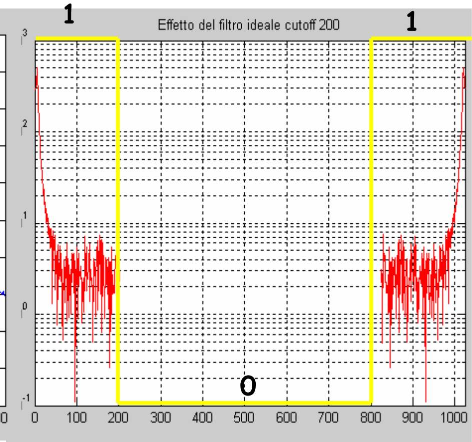




Esempi filtri 1D-Fourier

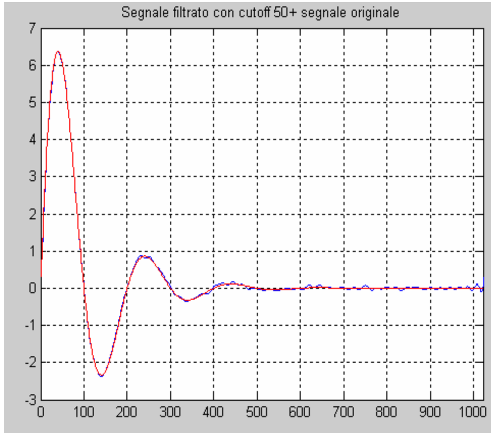
…. E facciamo l'antitrasformata.

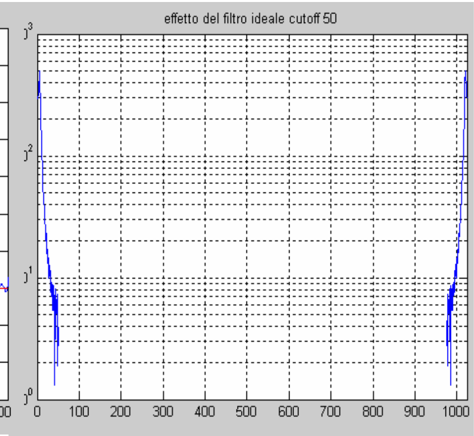




Esempi filtri 1D-Fourier

… non basta restringiamo la finestra portando il cutoff a 50

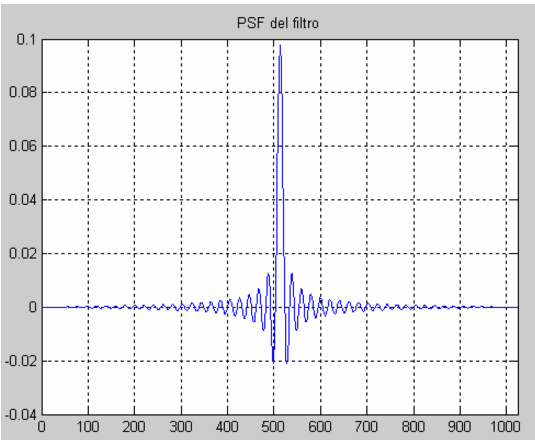




PSF del filtro

Negli esempi ora mostrati abbiamo progettato il filtro analizzando il segnale "rumoroso" in Fourier. Se si guarda l'antitrasformata del filtro si evidenziano le sue caratteristiche nello spazio del segnale.

Questa funzione è detta anche PSF (Point Spread Function) o kernel del filtro.

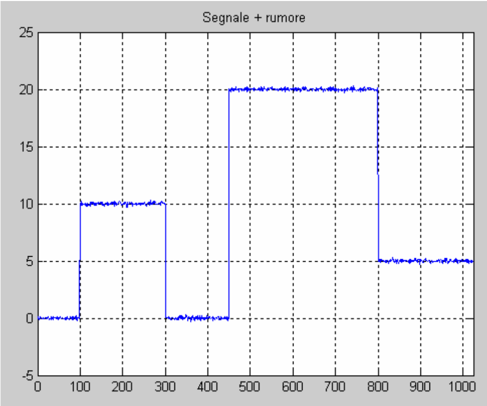


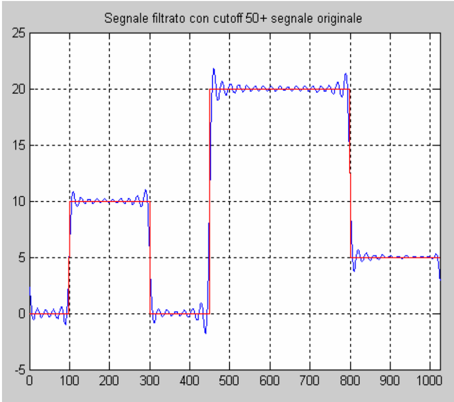
Altri filtri

Quello che abbiamo appena visto è detto filtro passa basso ideale, con lo stesso criterio si possono definire altri tipi di filtri:

- Passa alto
- Passa banda
- Elimina banda (elimina componente)

Il filtro passa basso "ideale" non è utilizzato in pratica perchè la PSF mostra oscillazioni che possono creare artefatti. Proviamo ad applicare il filtro passa basso ideale ad un segnale con discontinuità.





Altri filtri

Esistono altri filtri meno "ripidi": Hamming, Hann, Parzen, Butterworth, tutti "disegnati" sulla banda.

Hamming

$$\begin{cases} 0.5 + 0.5\cos\left(\dfrac{\pi\omega}{\Omega}\right) & \text{per} \quad |\omega| \le \Omega \\ 0 & \text{per} \quad |\omega| > \Omega \end{cases}$$

Hann

$$\begin{cases} 0.5 + 0.46\cos\left(\dfrac{\pi\omega}{\Omega}\right) & \text{per} \quad |\omega| \le \Omega \\ 0 & \text{per} \quad |\omega| > \Omega \end{cases}$$

Altri filtri

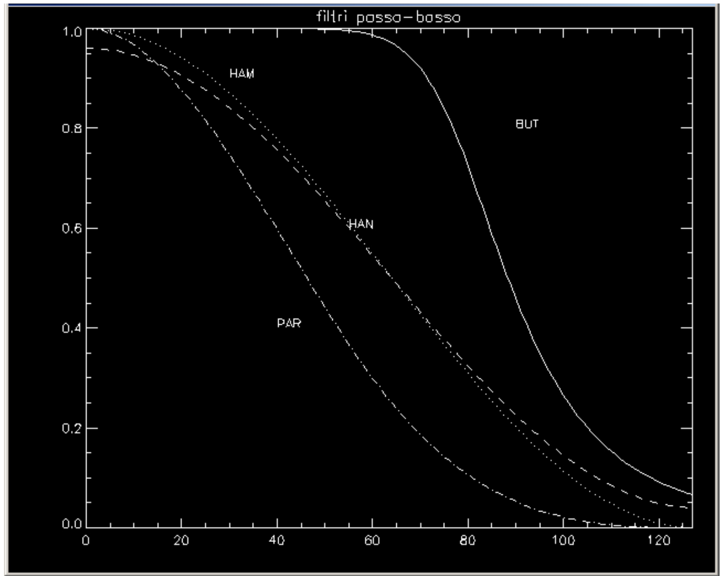
Parzen

$$\begin{cases} 1 - 6\left(\dfrac{|\omega|}{\Omega}\right)^2\left(1 - \dfrac{|\omega|}{\Omega}\right) & \text{per} \quad |\omega| \le \Omega/2 \\ 2\left(1 - \dfrac{|\omega|}{\Omega}\right)^3 & \text{per } \Omega/2 < |\omega| \le \Omega \\ 0 & \text{per} \quad |\omega| > \Omega \end{cases}$$

Butterworth

$$\frac{|\omega|}{\sqrt{1 + \left(\dfrac{\omega}{\omega_s}\right)^{2n}}}$$

Altri filtri

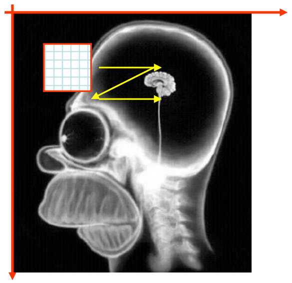


Filtri 2D

Al fine di migliorare l'intelligibilità di una immagine digitale una tecnica di primaria importanza è il filtraggio. Con il quale si possono enfatizzare alcune caratteristiche o rimuoverne altre.

Essenzialmente il filtraggio è una funzione di intorno, nel quale il valore che assume un pixel nell'immagine d'uscita è determinato dall'applicazione di un operatore che agisce sui valori dei pixels a lui circostanti nell'immagine d'ingresso. Per questo tale operatore viene detto operatore locale.

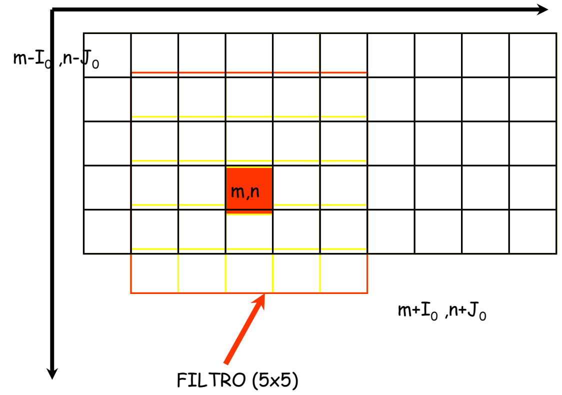
Interpretando l'immagine come una matrice, l'operatore agisce su una finestra di punti nell'intorno del pixel a cui è applicato. Questa operazione viene ripetuta per tutti i pixel esplorando l'immagine dall'alto verso il basso e da sinistra verso destra.



Filtri 2D - II

Formalmente possiamo definire un filtro generico 2D come un operatore che permette di associare il valore numerico di un pixel dell'immagine filtrata *g[m,n]* a quelli di *IxJ* pixels (solitamemte si prende *I* e *J* dispari) in un intorno del corrispondente pixel nell'immagine in ingresso *f[m,n]*.

$$g[m,n] = h(f[m-(I-1)/2, n-(J-1)/2],....\ f[m+(J-1)/2, n+(I-1)/2])$$



$I_0 = (I-1)/2$

$J_0 = (J-1)/2$

Filtri lineari

La finestra con i suoi pesi è chiamata 'nucleo di convoluzione' (convolution kernel) ed la sua forma determina il tipo di filtraggio sull'immagine.

Descivendo i filtri basati sulla convoluzione si usano le seguenti convenzioni. Dato un filtro $h[i,j]$ di dimensioni $I \times J$, consideriamo il centro della matrice **h** nelle coordinate [i=0,j=0] (questo è senz'altro vero se I e J sono dispari) come mostrato in figura:

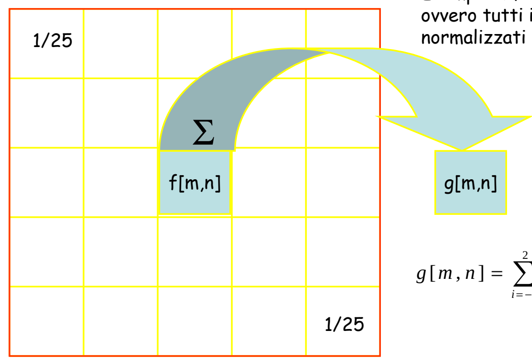
$$\begin{pmatrix}
h[-I_0,-J_0] & \cdots & \cdots & h[0,-J_0] & \cdots & \cdots & h[I_0,-J_0] \\
\vdots & \ddots & \vdots & \vdots & \vdots & \cdot^{\cdot^{\cdot}} & \vdots \\
\vdots & \cdots & h[-1,-1] & h[0,-1] & h[1,-1] & \cdots & \vdots \\
h[-,0] & \cdots & h[-1,0] & h[0,0] & h[1,0] & \cdots & h[,0] \\
\vdots & \cdots & h[-1,1] & h[0,1] & h[1,1] & \cdots & \vdots \\
\vdots & \cdot^{\cdot^{\cdot}} & \vdots & \vdots & \vdots & \ddots & \vdots \\
h[-I_0,J_0] & \cdots & \cdots & h[0,J_0] & \cdots & \cdots & h[I_0,J_0]
\end{pmatrix}$$

Filtri lineari II

Il filtraggio lineare riporta come risultato un valore dato dalla combinazione lineare dei valori dei pixel dell'intorno del pixel di ingresso. I pesi della combinazione lineare sono i valori assegnati dal filtro ad ogni pixel della finestra. Considerando l'operatore applicato con una finestra di dimensioni $I \times J$ su una immagine $M \times N$, il filtraggio avviene tramite la funzione:

$$g[m,n] = \sum_{i=-I_0}^{I_0} \sum_{j=-J_0}^{J_0} h[i,j]\, f[m-i, n-j]$$

Esempio *h* finestra 5x5, media mobile, ovvero tutti i pesi sono uguali e sono normalizzati in modo che la somma sia 1.



$$g[m,n] = \sum_{i=-2}^{2} \sum_{j=-2}^{2} h[i,j]\, f[m-i, n-j]$$

Scriviamo la formula della convoluzione tenendo fisso il centro del filtro cioè il punto *[0,0]*, e consideriamo il primo pixel dell'immagine filtrata.

$$g[0,0] = \sum_{i=-I_0}^{I_0} \sum_{j=-J_0}^{J_0} h[i,j]\, f[-i,-j] \qquad I_0 = \frac{I-1}{2} \qquad J_0 = \frac{J-1}{2}$$

La formula così scritta mette in evidenza che al valore dei pixel "vicino al bordo" nell'immagine filtrata contribuiscono pixel dell'immagine di partenza che sono "fuori" dal dominio dell'immagine stessa.

Quindi i risultati che si ottengono dalla scansione della finestra del filtro sui pixel di bordo immagine, sono affetti da un errore, prodotto dal fatto che parte della finestra del filtro non copre pixel dell'immagine. Esistono vari metodi per ovviare questo inconveniente.

Un metodo consiste nell'evitare di calcolare questi valori, partendo dal primo pixel che permette alla finestra di coprire i pixel dell'immagine, ciò, tuttavia produce un'immagine in uscita un pò più piccola di quella originale. Filtraggi ulteriori diminuiscono ulteriormente la dimensione...

Ci sono essenzialmente due tecniche che permettono di avere l'immagine in uscita delle stesse dimensioni di quella in ingresso:

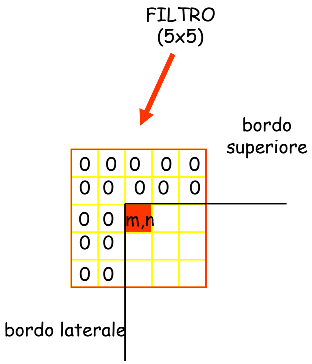
-Zero-padding

-Prolungamento periodico

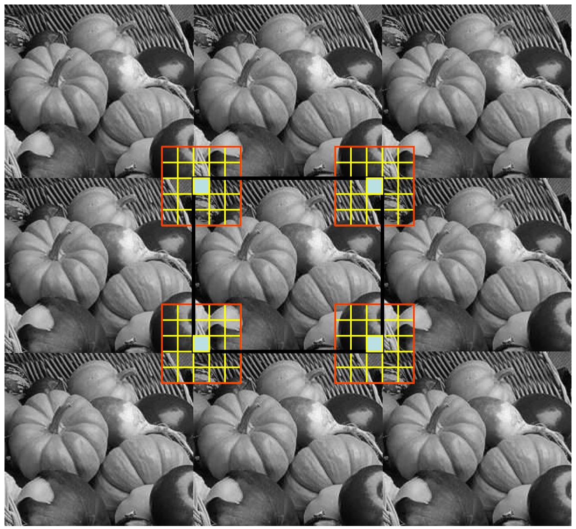
Nel primo caso i valori mancanti sono sostituiti con valori zero ("Zero padding"), e ciò può produrre errore sui pixel di bordo soprattutto se questi non hanno valori numerici vicini allo zero. La regione di bordo affetta da questo errore sarà tanto più larga quanto più è grande la finestra del filtro. Questa tecnica è consigliata solo nel caso di immagini con fondo 'nero'.

Nel secondo caso, i valori mancanti sono sostituiti da quelli dell'immagine stessa per prolungamento periodico. In pratica si piastrella il piano con l'immagine e quando il filtro passa sui bordi si considerano i pixel dell'immagine adiacente.

Effetti di bordo



Zero padding



Prolungamento periodico

Convoluzione 2D

In completa analogia con quello fatto in una dimensione si può affrontare la convoluzione con i filtri lineari attraverso la trasformata di Fourier.

Si estende il filtro alle stesse dimensioni dell'immagine e poi si scrive il prodotto di convoluzione nel seguente modo:

$$g[m,n] = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f[i,j]\, h[m-i, n-j]$$

dove la sommatoria si estende a tutta l'immagine e il filtro è traslato sopra ad essa

Considerazioni sulla convoluzione

- Quando la convoluzione è scritta nella forma standard per una immagine *f[m,n]* di dimensioni *MxN*

$$g[m,n] = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f[i,j]h[m-i,n-j]$$

  vediamo che il nucleo di convoluzione *h[i,j]* è reso speculare rispetto a *i=j=0* per produrre *h[-i,-j]* prima di essere traslato di *[m,n]*.

  Nonostante molti nuclei di convoluzione siano simmetrici e quindi *h[i,j]=h[-i,-j]*, altri non lo sono e quindi occorre porre attenzione all'implementazione degli algoritmi.

- Il costo computazionale di una convoluzione per un nucleo di dimensioni *JxJ* e una immagine di dimensioni *NxN* è $O(J^2)$ per pixel, ovvero $O(N^2 x J^2)$.
- Se l'immagine di partenza ha valori interi, il risultato di un filtraggio in generale avrà valori reali, quindi occorre porre attenzione al tipo delle variabili in gioco.

- La complessità del prodotto di convoluzione si può ridurre nel caso di nuclei *separabili*. Infatti se si può scrivere:

$$h[i,j] = h_r[j]h_c[i]$$

allora il prodotto di convoluzione si scrive

$$g[m,n] = \sum_{i=0}^{I-1} \left\{ \sum_{j=0}^{J-1} h_r[j]f[m-i,n-j] \right\} h_c[i]$$

questo significa, che invece di applicare un filtro bi-dimensionale si applicano due filtri monodimensionali, il primo nella direzione *i* e l'altro nella direzione *j*.

*La complessità passa da O(I*J) a O(I+J) per pixel.*

- Per certi filtri è possibile trovare una *implementazione incrementale* della convoluzione: mentre la finestra di convoluzione si muove sopra l'immagine la colonna di sinistra dell'immagine da elaborare è spostata "fuori" dalla finestra, mentre una nuova colonna entra a destra. Ciò permette di scrivere algoritmi con complessità per pixel *O(costante)*.

Il teorema di convoluzione e le sue applicazioni

Ovviamente il teorema di convoluzione vale anche in due dimensioni:

Siano ***f*** e ***g*** due immagini e ***h*** il nucleo di convoluzione ovvero la PSF del filtro

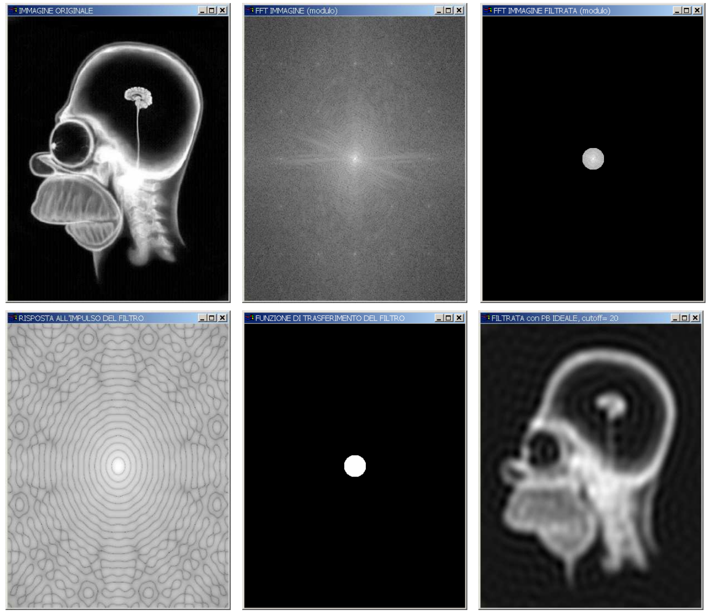
se ***g=h*f*** vale la seguente relazione tra le DFT

G[k,l]=H[k,l]F[k,l]

L'applicazione di un filtro può essere quindi affrontata in due modi:

1) Il filtro è "progettato in Fourier", si scrive quindi direttamente ***H***, si calcola la DFT dell'immagine da fiiltrare, si moltiplicano le due trasformate e si calcola l'antitrasformata del risultato.
2) Il filtro è scritto attraverso il suo nucleo di convoluzione ***h***, poi si calcola ***H***, ***F*** e quindi si applica il teorema di convoluzione come nel caso precedente.

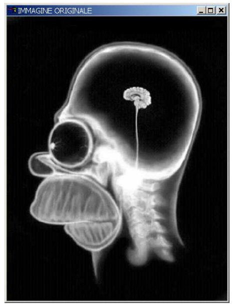
Filtri 2D in Fourier

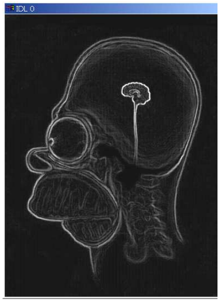


Filtri per estrazione di contorni: filtro di Sobel

Il filtro di Sobel è un filtro separabile e lineare che viene usato per l'estrazione dei contorni, si esegue una convoluzione con le seguenti maschere:

$$X_{mask} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Y_{mask} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

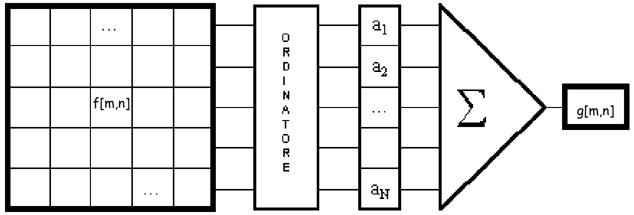




Operatori di rango

Gli operatori di rango sono operatori non lineari.

I valori di grigio contenuti nella finestra W del filtro sono dapprima disposti in ordine crescente secondo il loro valore (rango), e quindi sommati con coefficienti a_i che individuano la funzione del filtro (vedi figura). Si deve notare che con tale operazione si perdono informazioni di tipo spaziale sui pixel dell'intorno, pertanto non si deve confonderla con la somma pesata dei filtri lineari che invece realizza una convoluzione spaziale.



Esempi di operatori di rango sono gli operatori di minimo e massimo e il cosiddetto filtro mediano.

a) Operatore di minimo:

Ogni valore di grigio è sostituito dal minimo valore nella finestra dell'operatore. Dettagli fini e chiari vengono cancellati, mentre le zone più scure si espandono.

I coefficienti del filtro sono tali che l'unico coefficiente diverso da zero, e pari ad uno è quello relativo al minimo.

b) Operatore di massimo:

Ogni valore di grigio, in questo caso, è invece sostituito dal massimo valore nella finestra dell'operatore. Le zone più chiare si espandono a danno di quelle scure, dettagli fini e scuri vengono cancellati.

I coefficienti del filtro sono tali che l'unico coefficiente diverso da zero, e pari ad uno è quello relativo al massimo.

c) Filtro mediano:

Ogni valore di grigio è sostituito dalla mediana (non la media) dei valori nella finestra dell'operatore. Se la finestra contiene I^2 elementi il valore di grigio in uscita sarà quello che è minore o uguale a $(I^2-1)/2$ e maggiore o uguale a $(I^2-1)/2$.

Questo tipo di filtro si utilizza per sopprimere un particolare tipo di rumore, il rumore impulsivo o "sale e pepe" dove una certa percentuale di pixel è saturata (valore massimo) oppure non risponde (valore minimo). Tale filtro è molto efficace per togliere effetti come i "bad pixel" o raggi cosmici nelle immagini astronomiche.

Un esempio del filtro mediano