



Dipartimento di Matematica e Informatica

Corso di Laurea Specialistica in Informatica

MULTIMEDIA

Prof. S.Battiato

Relazione sul progetto :

Estensione del software ImageJ con tecniche di
enhancement nel dominio della frequenza



Annalisa Cappello

Indice

1. Introduzione	3
2. Caratteristiche comuni a tutti i plugins	6
3. Il plugin LowpassFilters	9
4. Il plugin HighpassFilters	14
5. Il plugin BandControl	18
6. Il plugin Laplacian	20
7. Il plugin SharpeningFilters	21
8. Conclusioni	23

1. Introduzione

Le tecniche di elaborazione nel dominio della frequenza trovano fondamento teorico nel teorema di convoluzione, che fa corrispondere all'operazione di filtraggio spaziale definita come :

$$g(x, y) = f(x, y) * h(x, y) = \frac{1}{M * N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$$

l'operazione di filtraggio nel dominio della frequenza definita come :

$$G(u, v) = F(u, v) * H(u, v)$$

$F(u, v)$ è detta *trasformata di Fourier* dell'immagine $f(x, y)$ e può essere calcolata attraverso la formula

$$\mathfrak{F}\{f(x, y)\} = F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy$$

che riorganizza i dati dell'immagine nello spazio delle frequenze. Nello spazio di Fourier l'immagine viene rappresentata sotto un altro punto di vista: ogni pixel contiene contemporaneamente un'informazione relativa all'ampiezza (cioè al fatto che l'immagine contenga o meno una struttura periodica) e una relativa alla fase (cioè dove le eventuali strutture periodiche sono localizzate).

$H(u, v)$ è il filtro, così chiamato perché agisce solo su alcune frequenze della trasformata lasciando le altre immutate.

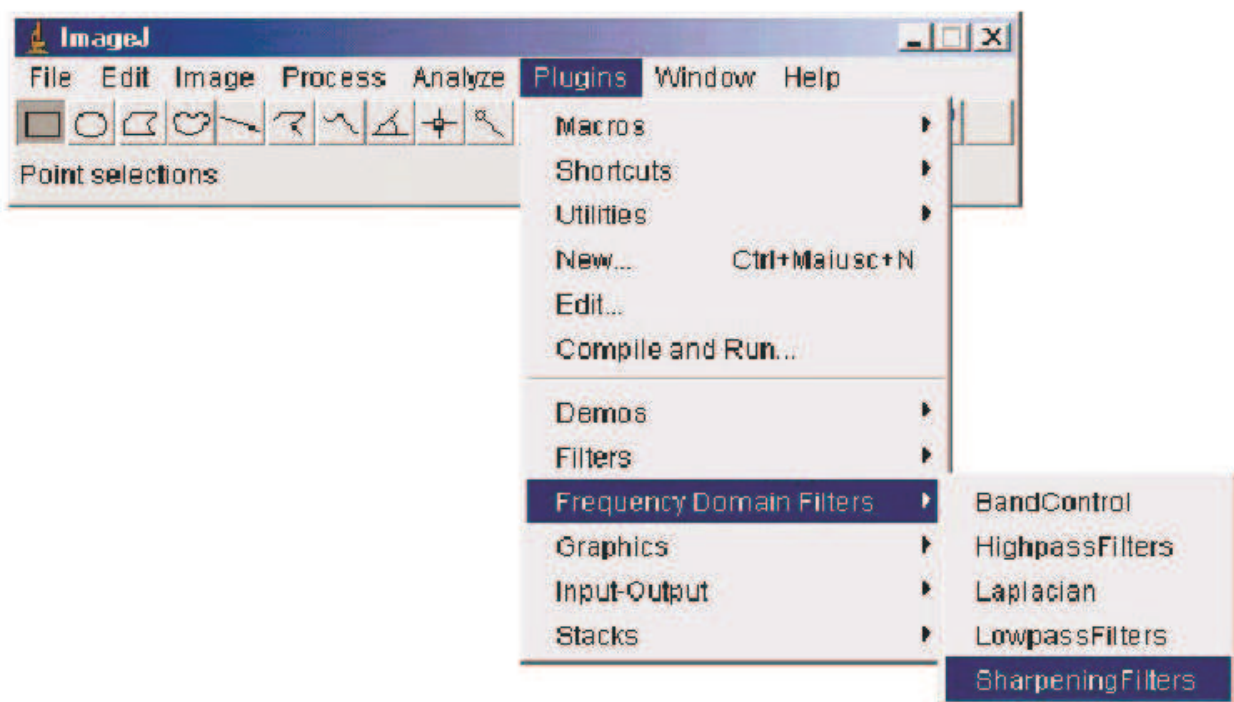
L'obiettivo dell'elaborazione di un'immagine nel dominio della frequenza è, quindi, quello di trovare un filtro $H(u, v)$ tale che l'immagine di output $g(x, y)$ esibisca le proprietà desiderate. $g(x, y)$ viene ottenuta attraverso l'antitrasformata:

$$g(x, y) = \mathcal{F}^{-1}\{F(u, v) * H(u, v)\} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (F(u, v) * H(u, v)) e^{i2\pi(ux+vy)} dudv$$

Implementare trasformata e antitrasformata nelle loro forme classiche richiederebbe un numero di operazioni proporzionale ad N^2 (nel caso di un'immagine di dimensioni $N \times N$). Per questo motivo si preferisce usare la cosiddetta *Fast Fourier Transform* (FFT), che attraverso opportune tecniche di decomposizione riesce ad abbassarne la complessità ad $O(N \log_2 N)$.

In particolare ImageJ implementa la *Fast Hartley Transform*, basata sul codice Pascal originale scritto da Arlo Reeves e che si può trovare all'indirizzo <http://rsb.info.nih.gov/ij/docs/ImageFFT/>.

Lo scopo del mio progetto è stato l'estensione del software ImageJ con plugins che implementassero tecniche di "enhancement" nel dominio della frequenza. I plugins che ho creato sono stati sistemati nel menu Plugins all'interno di una cartella che ho chiamato *Frequency Domain Filters*



Il plugin LowpassFilters permette di effettuare un filtraggio passa-basso con il filtro ideale, di Butterworth o Gaussiano.

Il plugin HighpassFilters permette di effettuare un filtraggio passa-alto con il filtro ideale, di Butterworth o Gaussiano.

Il plugin BandControl contiene le classi necessarie per effettuare un filtraggio band-pass o band-reject.

Il plugin Laplacian filtra l'immagine corrente attraverso il filtro Laplaciano.

Infine il plugin SharpeningFilters implementa l'High Boost filter e l'High Frequency Enphasys utilizzando uno qualsiasi dei filtri passa-alto studiati.

2. Caratteristiche comuni a tutti i plugins

All'inizio di ogni file è fondamentale importare tutti i packages necessari: possono essere le librerie standard java oppure i packages di ImageJ in modo, per esempio, da poter usare altri plugin o creare un dialogo con l'utente:

```
import ij.*;  
import ij.process.ImageProcessor.*;  
import ij.process.*;  
import ij.process.FHT.*;  
import ij.plugin.filter.*;  
import ij.plugin.*;  
import ij.plugin.FFT.* ;  
import java.lang.Math.* ;  
import java.lang.String.*;  
import javax.swing.*;  
import ij.measure.Calibration;  
import ij.gui.* ;
```

Tutti i plugin implementano l'interfaccia `PlugInFilter` che permette di caricare ed elaborare immagini. I metodi dell'interfaccia `PlugInFilter` sono:

➤ *void run(Image Processor ip)*

Questo metodo manda in esecuzione il plugin. L'input passato come argomento è il processore di immagine. Il processore può essere modificato direttamente o può fornire dati per creare un nuovo processore o una nuova immagine, in modo che l'immagine originale rimanga invariata. L'immagine rimane bloccata fino a quando l'esecuzione del plugin non termina. Il metodo *run* contiene la chiamata al metodo che effettua il filtraggio.

L'argomento può essere passato attraverso il metodo

➤ *int setup(java.lang.String arg, ImagePlus imp)*

Questo metodo setta il *plugin filter* per l'uso. La stringa *arg* che può essere anche non indicata, rappresenta l'argomento per lo specifico plugin nel file *IJ_Props.txt*. *imp* è l'immagine correntemente attiva, passata come argomento (in automatico) da ImageJ. Il metodo *setup* restituisce un flag che rappresenta le capacità del filtro. Tutti i plugins che ho implementato restituiscono il flag *DOES ALL* cosicché possano processare tutti i tipi di immagine.

Il filtraggio avviene nel dominio della frequenza, quindi bisogna calcolare la trasformata dell'immagine. Quando si visualizza lo spettro di Fourier come immagine di intensità, in genere, esso manifesta una dinamica molto più grande di quella riproducibile da un tipico display, per cui risultano visibili solo le parti più luminose dello spettro. Per questo motivo, alla trasformata dell'immagine che dovrà essere visualizzata viene applicata una compressione di tipo logaritmico:

$$D(u, v) = c * \log(1 + F(u, v))$$

c è una costante scelta opportunamente in modo che i valori trasformati appartengano al range desiderato.

L'operazione di compressione logaritmica non viene, invece, effettuata sulla trasformata che dovrà essere moltiplicata per il filtro.

La FFT sviluppata da ImageJ può essere applicata solo su immagini quadrate le cui dimensioni sono potenze di due. Inoltre, nel dominio della frequenza, sono necessari particolari accorgimenti ai bordi in modo da evitare i cosiddetti “*wrap around errors*”. Per entrambi questi motivi, una volta trovate le dimensioni dell'immagine, viene calcolata la più piccola immagine ricavata dall'originale che la contiene interamente le cui dimensioni siano potenze di due. Al centro di questa nuova

immagine completamente nera viene inserita l'immagine originale. La FFT può essere così calcolata sulla nuova immagine senza causare nessun tipo di errore.

Per effettuare il filtraggio è necessario applicare al filtro uno shift dell'origine nel punto $M/2$, $N/2$ (con M e N rispettivamente la larghezza e l'altezza dell'immagine originale).

Inoltre, dato che possono essere filtrati tutti i tipi di immagine, se l'ImageProcessor dell'immagine originale è un ColorProcessor (cioè l'immagine è RGB), prima di calcolare la trasformata viene calcolata (attraverso il metodo *getBrightness()*) la luminosità, in modo da poterla aggiungere a filtraggio concluso.

L'immagine originale che serve da input al plugin non viene modificata.

L'immagine di output viene creata applicando l'antitrasformata al prodotto del filtro per la trasformata e ritagliando la nuova immagine nel dominio spaziale tenendo conto delle dimensioni originali.

3. Il plugin LowpassFilters

I filtri passa-basso (lowpass o di smoothing) enfatizzano le lente transizioni e attenuano quelle brusche. Ciò corrisponde, nel dominio della frequenza, ad un'enfaticazione delle basse frequenze ed una conseguente attenuazione delle alte frequenze.

Il filtro ideale passa-basso elimina completamente tutte le componenti che distano dall'origine centrata del rettangolo delle frequenze più una certa soglia D_0 , chiamata *frequenza di taglio*. Il filtro $H(u,v)$ si ottiene con la seguente formula:

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

dove $D(u,v)$ è la distanza dell'origine data da

$$D(u, v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

Al variare della frequenza di taglio D_0 , il filtro permetterà il passaggio di diverse porzioni dello spettro di potenza dell'immagine. Ecco un esempio di filtro ideale passa-basso con frequenza di taglio pari a 20.



A differenza del filtro ideale, il filtro passa-basso di Butterworth non determina un taglio netto in frequenza, in quanto la sua funzione di trasferimento non presenta una brusca transizione. Il luogo delle frequenze di taglio è definito dai punti del piano per i quali la $H(u,v)$ ricade all'interno di una frazione del suo massimo valore. Definiamo la funzione di trasferimento per il filtro passa-basso di Butterworth di frequenza di taglio D_0 e ordine n :

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}}$$

con la distanza $D(u,v)$ definita come nel filtro ideale.

Al crescere dell'ordine del filtro, il profilo diventa sempre più netto e quindi la funzione di trasferimento si avvicina a quella del filtro ideale (assumendone, quindi, caratteristiche e difetti). Ecco un esempio di filtro di Butterworth passa-basso con frequenza di taglio pari a 20 e ordine uguale a 2.



Infine, i filtri gaussiani sono particolarmente importanti perché sia la trasformata diretta che quella inversa sono delle funzioni gaussiane reali. Essi presentano delle

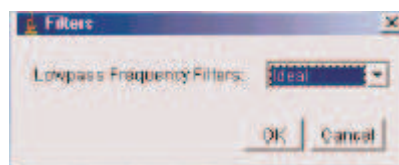
caratteristiche simili ai filtri di Butterworth di piccolo ordine, ma sono più semplici da realizzare. Il filtro $H(u,v)$ è definito nel seguente modo:

$$H(u,v) = e^{\frac{-D^2(u,v)}{2D_0^2}}$$

con la distanza $D(u,v)$ definita come nel filtro ideale. Ecco un esempio di filtro gaussiano passa-basso con frequenza di taglio pari a 20.



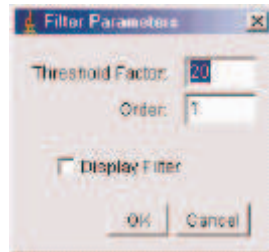
Il tipo di filtro viene selezionato attraverso il seguente *genericDialog* :



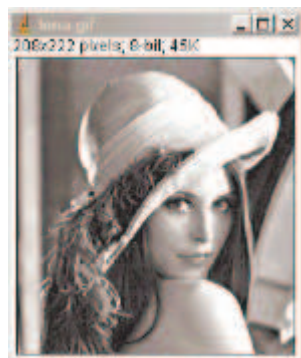
mentre la frequenza di soglia e se visualizzare il filtro vengono selezionati dall'utente attraverso una seconda finestra di dialogo così strutturata:



Se l'utente ha scelto di effettuare un filtraggio mediante il filtro di Butterworth, oltre alla frequenza di taglio bisogna specificare l'ordine, quindi, invece della precedente finestra di dialogo, si aprirà una finestra come la seguente



Vediamo l'applicazione di questi filtri (con frequenza di soglia uguale a 20) sulla seguente immagine a 8 bit:



La trasformata di questa immagine è :



Applicando il filtro ideale si ottiene:



Come si può notare questo filtro causa un fenomeno di sfocatura ad anello chiamato effetto *ringing*. Ciò è dovuto alla funzione antitrasformata del filtro ideale passa-basso. Gli anelli generati hanno un raggio inversamente proporzionale al valore della frequenza di taglio del filtro e tendono ad interferire anche in relazione alla distanza tra due impulsi.

Applicando il filtro di Butterworth alla stessa immagine si ottiene:



Applicando il filtro Gaussiano:



4. Il plugin HighpassFilters

I filtri passa-alto (o highpass) operano in modo completamente speculare rispetto ai filtri passa-basso. Essi, cioè, enfatizzano le brusche transizioni e attenuano quelle lente. Ciò corrisponde, nel dominio della frequenza, ad un'enfatizzazione delle alte frequenze ed una conseguente attenuazione delle basse frequenze.

Il filtro ideale passa-alto elimina completamente tutte le componenti che distano dall'origine centrata del rettangolo delle frequenze meno di una certa soglia D_0 , chiamata *frequenza di taglio*. Il filtro $H(u,v)$ si ottiene con la seguente formula:

$$H(u,v) = \begin{cases} 0 & D(u,v) \leq D_0 \\ 1 & D(u,v) > D_0 \end{cases}$$

dove $D(u,v)$ è la distanza dell'origine data da

$$D(u,v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

Al variare della frequenza di taglio D_0 , il filtro permetterà il passaggio di diverse porzioni dello spettro di potenza dell'immagine. Ecco un esempio di filtro ideale passa-alto con frequenza di taglio pari a 20.

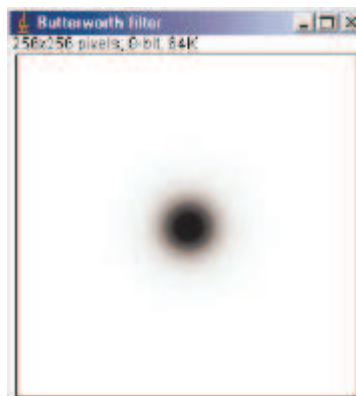


A differenza del filtro ideale, il filtro passa-alto di Butterworth non determina un taglio netto in frequenza, in quanto la sua funzione di trasferimento non presenta una brusca transizione. Definiamo la funzione di trasferimento per il filtro passa-alto di Butterworth di frequenza di taglio D_0 e ordine n :

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}}$$

con la distanza $D(u,v)$ definita come nel filtro ideale.

Al crescere dell'ordine del filtro, il profilo diventa sempre più netto e quindi la funzione di trasferimento si avvicina a quella del filtro ideale (assumendone, quindi, caratteristiche e difetti). Ecco un esempio di filtro di Butterworth passa-alto con frequenza di taglio pari a 20 e ordine uguale a 2.



Infine, il filtro $H(u,v)$ passa-alto gaussiano è definito nel seguente modo:

$$H(u, v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$$

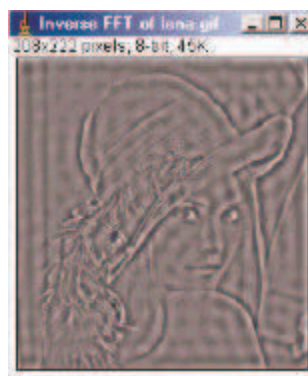
con la distanza $D(u,v)$ definita come nel filtro ideale. Ecco un esempio di filtro gaussiano passa-basso con frequenza di taglio pari a 20.



Le modalità di selezione per il tipo di filtro, la frequenza di soglia, l'ordine e se visualizzare il filtro sono molto simili alle modalità già viste nel plugin LowpassFilters.

Vediamo l'applicazione di questi filtri (con frequenza di soglia uguale a 20) sempre sulla stessa immagine.

Applicando il filtro ideale si ottiene:



Anche nel caso del filtro ideale passa-alto, l'effetto ringing rimane ancora molto visibile.

Applicando il filtro di Butterworth:



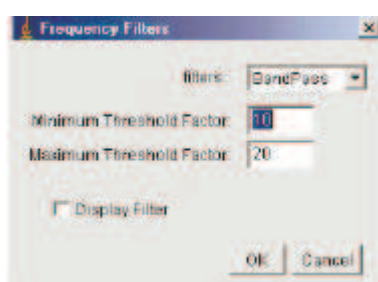
Applicando il filtro Gaussiano si ottiene:



5. Il plugin BandControl

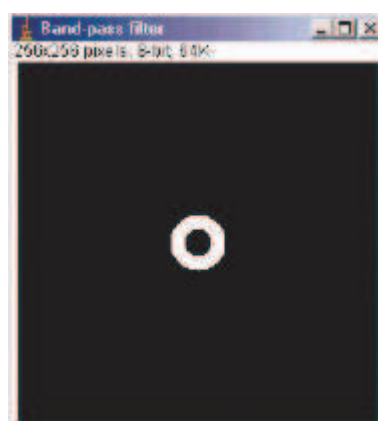
I filtri band-pass lasciano passare solo le componenti presenti all'interno di una specifica banda di frequenza. Invece i filtri band-reject eliminano completamente tutte le frequenze all'interno della banda.

Per calcolare la banda utile l'utente deve selezionare una frequenza di taglio minima e una frequenza di taglio massima attraverso la seguente finestra:



Come nei precedenti plugins, è possibile scegliere se visualizzare o meno il filtro che si sta applicando.

Utilizzando una frequenza di soglia minima di 10 pixels e una frequenza di soglia massima di 20 pixels, si ottiene il filtro passa-banda



che applicato all'immagine *lena.gif*, dà in output



Utilizzando una frequenza di soglia minima di 45 pixels e una frequenza di soglia massima di 50 pixels al filtro band-reject, si ottiene



che applicato all'immagine *lena.gif*, dà in output



6. Il plugin Laplacian

Il filtro Laplaciano effettua un *enhancement* controllato delle alte frequenze. Si tratta di un filtro la cui risposta è indipendente dalla direzione delle discontinuità (filtri di questo tipo sono detti *isotropici*) e che si basa sulle derivate seconde.

Il filtraggio si realizza nel seguente modo:

$$g(x, y) = \mathfrak{F}^{-1} \left\{ \left[1 + \left(\left(u - \frac{M}{2} \right)^2 + \left(v - \frac{N}{2} \right)^2 \right) \right] F(u, v) \right\}$$

I filtri passa-alto ideale, di Butterworth e Gaussiano hanno in comune lo sfondo nero dovuto all'eliminazione della componente continua. Il filtro Laplaciano ha il grande vantaggio di risolvere questo problema aggiungendo l'immagine originale come sfondo dell'immagine filtrata.

Il risultato sull'immagine *lena.gif* è



7. Il plugin SharpeningFilters

Il filtro High Boost incrementa il contributo dell'immagine originale attraverso una costante A .

La formula generale è :

$$H_{hb}(u, v) = (A - 1) + H_{hp}(u, v)$$

con H_{hb} il filtro High Boost e H_{hp} un qualsiasi filtro passa-alto.

Il filtro High Frequency Emphasis, oltre ad aggiungere un offset per evitare che la componente continua si annulli, moltiplica il filtro per una costante:

$$H_{hfe}(u, v) = A + B * H_{hp}(u, v)$$

con H_{hfe} il filtro High Frequency Emphasis e H_{hp} un qualsiasi filtro passa-alto.

Il filtro principale e il filtro passa-alto da usare nelle formule vengono selezionati dall'utente tramite la finestra



In base al filtro principale e al filtro passa-alto scelti, si aprirà un'altra finestra che premetterà la selezione di tutti i parametri necessari.

Il risultato ottenuto filtrando l'immagine *lena.gif* con l'High Boost filter, usando una costante $A = 3$ e il filtro Gaussiano con frequenza di taglio pari a 40 come filtro passa-alto, è



Mentre l'output del plugin usando l'High Frequency Emphasis con le costanti $A = 0.5$ e $B = 2$ e usando il filtro Laplaciano come filtro passa-alto è



8. Conclusioni

Nonostante esistano altre trasformate utilizzate frequentemente in operazioni come il restauro e la codifica, la trasformata di Fourier mantiene un ruolo di assoluto primato nell'*image processing*.

Questo perchè i vantaggi che la trasformata di Fourier fornisce sono molteplici.

In particolare permette di

- Sopprimere frequenze indesiderate
- Ridurre lo spazio occupato dei dati limitando la degenerazione del segnale
- Rigenerare segnali degradati