

**Alma Mater Studiorum - Università di Bologna**

**Campus di Cesena**

**Scuola di Ingegneria e Architettura**

**Corso di Laurea in Ingegneria Biomedica**

**Rassegna, implementazione e confronto di diversi  
metodi di filtraggio per le Bio-Immagini**

Elaborato in

Laboratorio di Bioingegneria

Relatore

Prof.ssa Cristiana Corsi

Presentata da

Gabriele Bucci

Sessione Terza

Anno Accademico 2012-2013

# INDICE

---

*Introduzione.....pag 3*

*Capitolo 1 :*

*Filtraggio nel Dominio dello Spazio.....pag 5*

*Capitolo 2 :*

*Filtraggio nel Dominio della Frequenza.....pag 23*

*Capitolo 3 :*

*Realizzazione Interfaccia Grafica per il Filtraggio delle Immagini...pag 38*

*Conclusioni .....pag 46*

*Bibliografia.....pag 47*

*Ringraziamenti.....pag 48*

---

# INTRODUZIONE

---

Questo lavoro di tesi si occuperà del filtraggio e dell'elaborazione delle bioimmagini per ridurre gli artefatti ed ottenere una immagine in cui il rumore è stato rimosso, rendendo più semplice ed efficace la successiva estrazione di indicatori clinici, diagnostici e prognostici, dall'analisi dei dati filtrati.

La medicina ha svolto enormi progressi negli ultimi decenni, in particolare nello sviluppo dei macchinari diagnostici, quali ecografi, risonanza magnetica, TAC, macchine radiografiche, elettrocardiografo, PET.

Originariamente molte di queste macchine sono state progettate in modo da fornire un output analogico, quindi stampare le immagini su pellicola oppure il tracciato su carta millimetrata; oggi è in corso un processo di digitalizzazione che rende obsolete le vecchie macchine (od utilizzabili previa conversione ad output digitali) e permette di ottenere le immagini direttamente in formato digitale.

Sensori specifici acquisiscono il segnale utile che successivamente viene elaborato per ottenere il risultato finale; ciò comporta un enorme risparmio di tempo rispetto alla modalità analogica; tanto che alcuni esami sono svolti e refertati in tempo reale, con conseguente possibilità di agire tempestivamente.

I suddetti macchinari forniscono ai medici importanti informazioni che consentono la formulazione della diagnosi, però, con le attuali tecnologie, le immagini possono risultare di scarsa qualità e quindi difficilmente utilizzabili; possono risultare distorte, affette da rumore digitale (dovuto all'amplificazione del segnale utile e al rumore di fondo) oppure con livelli di luminosità non ottimali (possono esser troppo scure o eccessivamente chiare).

Da qui si comprende la necessità di elaborare le immagini ottenute per evidenziarne i dettagli e garantire al medico materiale di qualità; poiché, l'errata interpretazione di un dettaglio può avere gravi ripercussioni sul paziente.

Nel mio progetto di tesi, ho sviluppato un interfaccia grafica, con il programma Matlab, che permette all'utilizzatore di caricare l'immagine desiderata e modificarla attraverso l'applicazione di varie forme di filtraggio, riuscendo in questo modo ad ottenere un risultato accettabile.

Inoltre è possibile visualizzare le immagini salvate con lo standard DICOM (Digital Imaging and Communication In Medicine) che definisce il formato ed il tipo di comunicazione tra le varie macchine in ambito medico.

Si può simulare l'immagine affetta da vari tipi di rumore e osservare l'effetto dei vari tipi di filtraggio nell'eliminazione di questi ultimi.

Il funzionamento dettagliato e il codice utilizzato per la scrittura dell'applicazione verrà trattato più dettagliatamente nel capitolo 3 della mia trattazione.

Applicare un filtro ad un immagine significa andarne a modificare le caratteristiche, per esaltare certe porzioni (come i contorni o i dettagli più piccoli) o attenuarne altre (ad esempio per ridurre gli effetti del rumore).

Esistono due grandi famiglie di filtri:

- filtraggio nel dominio dello spazio
- filtraggio nel dominio della frequenza

Queste due tipologie saranno descritte nel dettaglio nei capitoli seguenti.

# CAPITOLO 1

## FILTRAGGIO NEL DOMINIO DELLO SPAZIO

---

I filtri spaziali vengono applicati ad un'immagine, che è un segnale bidimensionale discreto, cioè che può assumere solamente certi valori definiti. Come suggerisce il titolo, la variabile non è il tempo, come nella maggior parte dei segnali digitali, ma lo spazio, poiché si è interessati alle variazioni delle tonalità nello spazio appunto.

L'immagine in digitale assume la forma di una matrice di valori (ogni valore corrisponde ad una diversa intensità assegnata al pixel corrispondente nella matrice) così come il filtro, quindi si applica una convoluzione tra due matrici.

In generale la convoluzione tra due funzioni prevede la rotazione di  $180^\circ$  di una funzione rispetto all'origine e il suo scorrimento sull'altra funzione, durante il quale si esegue un calcolo, in questo caso la somma dei prodotti.

La convoluzione tra due funzioni bidimensionali può essere espressa mediante la seguente formula (1.1):

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

$$x=0,1,2\dots M-1 \quad y=0,1,2\dots N-1$$

### Equazione 1.1

essa rappresenta il progressivo scorrimento di una maschera filtro, ruotato precedentemente di  $180^\circ$ , sull'immagine e il calcolo della somma dei prodotti in ogni posizione.

Rapportando il discorso alle immagini, l'obiettivo è quello di assegnare un nuovo valore ad ogni pixel mediante la sovrapposizione del mio filtro (kernel) all'immagine originale e facendo la sommatoria dei prodotti tra i valori

dell'immagine e quelli del kernel ed andando a sostituire il risultato al pixel centrale.

L'immagine filtrata è creata dalle modifiche indotte dal filtro su ogni pixel dell'immagine di input.

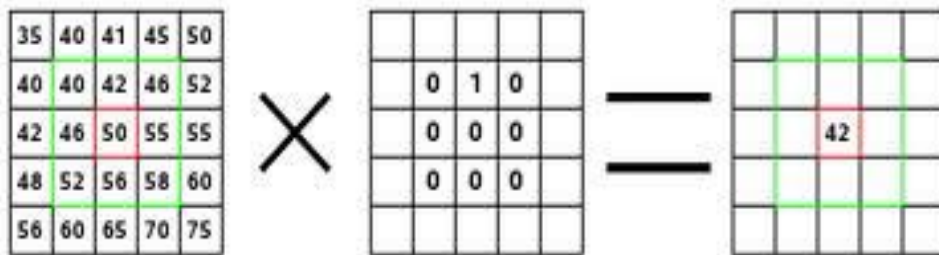


Figura 1.1

L'operazione di convoluzione viene poi effettuata per ogni pixel dell'immagine fino ad ottenere il risultato finale.

Dipendentemente dalle dimensioni del kernel che si utilizza, vi sono delle cornici dell'immagine a cui non si riesce ad applicare la convoluzione, quindi il risultato del filtraggio sarà un'immagine di dimensioni inferiori all'originale di tante cornici pari alla parte intera della metà delle dimensioni del kernel che generalmente è quadrato e di dimensioni dispari.

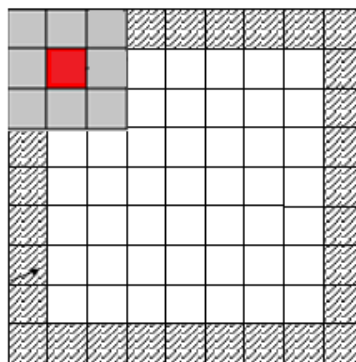


Figura 1.2

A questo problema si possono apporre principalmente due soluzioni:

1. effettuare la procedura chiamata “ Zero Padding ” , cioè aggiungere all’immagine originale M-1 righe e N-1 colonne di zeri, con un kernel di M righe e N colonne.  
Questo processo è adatto per immagini con fondo nero, in quanto, se fosse di un altro colore, gli zero aggiunti andrebbero a modificare i valori ottenuti nei bordi dell’immagine finale.  
L’errore sarà tanto più rilevante quanto più rilevanti sono le dimensioni del filtro.
2. applicare un prolungamento periodico, i valori mancanti sono sostituiti da quelli dell’immagine stessa, applicandola in serie a tutto il piano, in questo modo, quando il kernel andrà a coprire i bordi della mia immagine, prenderà in considerazione i pixel dell’immagine adiacente.

I filtri spaziali si dividono in :

- lineari
- non lineari

Questa suddivisione dipende dall’operazione di convoluzione, se è lineare i filtri vengono nominati come lineari appunto.

### ➤ **Filtri Lineari**

Il filtraggio lineare restituisce come risultato un valore dato dalla combinazione lineare dei pixel nell’intorno del pixel centrale.

I pesi della combinazione lineare sono i valori assegnati dalla maschera filtro ad ogni pixel dell’immagine sottostante.

La risposta R di un filtraggio lineare è rappresentabile vettorialmente mediante una somma di prodotti:

$$R = w_1z_1 + w_2z_2 + \dots + w_{mn}z_{mn} = \sum_{k=1}^{mn} w_kz_k$$

Equazione 1.2

in cui i valori  $w$  sono i coefficienti di un filtro di dimensioni  $m \times n$  e i valori  $z$  sono le intensità corrispondenti nell'immagine da filtrare; se si vuole utilizzare l'operazione di convoluzione bisogna ricordare di ruotare preventivamente il filtro di  $180^\circ$ .

I valori  $w$  dei coefficienti devono essere tali che la somma di tutti deve dare 1.

I filtri lineari si suddividono funzionalmente in due categorie differenti:

- I. filtri di smoothing
- II. filtri di sharpening

### *I. Filtri di Smoothing Lineari*

I filtri di smoothing sono utilizzati principalmente per effettuare operazioni di regolarizzazione e per ridurre il rumore di fondo presente nelle immagini acquisite eliminando i dettagli inutili e correggendo piccole imperfezioni interne alle regioni omogenee che potrebbero compromettere la qualità del risultato.

Un effetto indesiderato di questi filtri è uno sfocamento dei contorni dell'immagine.

Sono chiamati anche filtri low pass, poiché vanno ad eliminare le alte frequenze che sono quelle caratteristiche delle brusche transizioni di tonalità.

### **Filtro di Media**

L'implementazione per il filtraggio attraverso un filtro di media di dimensioni  $m, n$  di un'immagine di dimensioni  $M$  e  $N$  è espressa dalla formula:

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

Equazione 1.3

L'espressione si può tradurre a parole come l'applicazione del kernel del filtro alla matrice dell'immagine, il tutto diviso la somma dei vari coefficienti del kernel stesso.



$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{25} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

Figura 1.3

L'effetto dei vari filtri di media dipende in maniera rilevante dalle dimensioni dei kernel, a dimensioni maggiori corrisponde un filtraggio più aggressivo. In matlab il filtro di media è implementato in due modi differenti, attraverso il filtro average (fspecial('average')) e attraverso il filtro di media (medfilt2). La differenza sostanziale sta nel fatto che, attraverso l'average si può utilizzare solamente un kernel quadrato, mentre col filtro di media il numero di righe e colonne è regolabile singolarmente.

### **Filtri di Media Ponderata – Gaussiano**

I filtri di media ponderata utilizzano lo stesso principio dei filtri di media, l'unica differenza è nei valori degli elementi del kernel, infatti si vuole dare maggior rilevanza al pixel centrale, attribuendogli un valore più elevato rispetto ai pixel vicini.

Se si vuole applicare un filtro gaussiano, dando quindi un peso differente ai vari pixel, si utilizzano kernel diversi, caratterizzati appunto da una distribuzione gaussiana :

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

Equazione 1.4

implementabile con kernel di questo tipo:

$$\frac{1}{10} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figura 1.4

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

Figura 1.5

La figura (1.4) è esemplificativa di kernel che implementano filtri di media ponderata, dando peso maggiore al pixel centrale, mentre la figura (1.5) ritrae il kernel che implementa filtri gaussiani.

I valori del kernel per il filtraggio gaussiano devono soddisfare l'equazione che rappresenta la distribuzione gaussiana (1.4).

La modalità con cui vanno a modificare l'immagine è la stessa dei filtri di media, l'unica differenza sono i valori degli elementi del kernel.

In seguito si possono osservare gli effetti di questi due tipologie di filtraggio sulla stessa immagine :

- Filtro di Media [5x5]

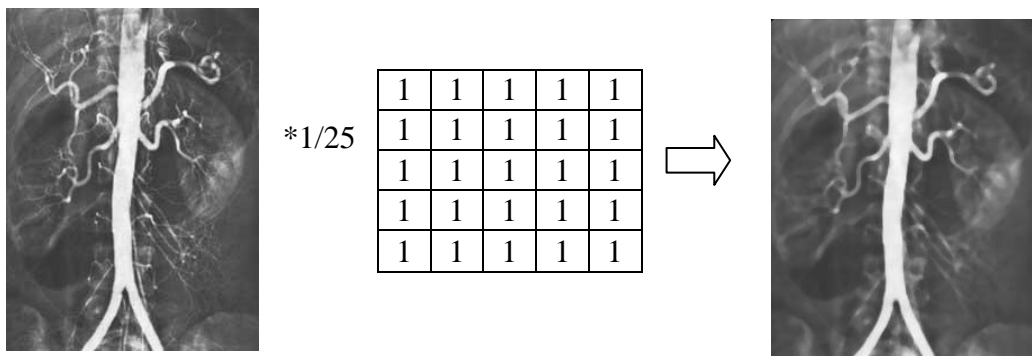


Figura 1.6

- Filtro di Media Ponderata - Gaussiano [5x5]

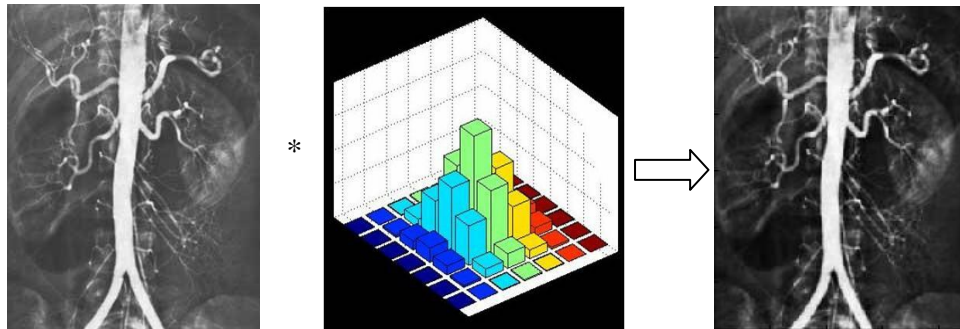


Figura 1.7

## II. Filtri di Sharpening Lineari

L'obiettivo di questa famiglia di filtri è quello di mettere in evidenza le transizioni di intensità; la loro struttura è basata sugli operatori di derivata prima e derivata seconda.

Perciò devono presentare le seguenti caratteristiche:

1. assumono un valore nullo nelle aree d'intensità costante
2. Sono diversi da zero su un gradino (o rampa) d'intensità
3. Sono zero lungo una rampe di inclinazione costante

La derivata seconda, come si vedrà più chiaramente dalla definizione, evidenzia in maniera più netta i dettagli rispetto alla derivata prima, proprietà principe per questo tipo di filtraggio.

### Filtri di Prewitt

Per lo sharpening lineare attraverso i suddetti filtri, si utilizzano le derivate prime, le quali vengono implementate sfruttando un'approssimazione del gradiente.

Per una funzione  $f(x,y)$  il gradiente di  $f$  alle coordinate  $(x,y)$  è definito come un vettore colonna:

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$$

Equazione 1.5

Il vettore indica la direzione dove si ha la variazione più importante di intensità rispetto alla posizione (x,y), in parole povere indica il punto in cui l'intensità dell'immagine aumenta in maniera più considerevole rispetto all'attuale posizione.

A questo punto è utile definire il modulo del vettore gradiente col nome di magnitudine:

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

Equazione 1.6

Per poter applicare i filtri alle immagini è necessario implementarle attraverso delle matrici filtro, prendendo una matrice campione (figura 1.8) sarà possibile definire i filtri di Prewitt e di Sobel.

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

Figura 1.8

Questa tipologia di filtri va ad esaltare i contorni ed i dettagli dell'immagine (edge detector), attraverso il calcolo di un'approssimazione del gradiente:

$$g_x = \frac{df}{dx} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{df}{dy} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

Equazione 1.7

Questa formulazione è facilmente traducibile nelle seguenti matrici filtro:

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

Figura 1.9

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

Figura 1.10

La prima matrice (1.9) rappresenta la derivata parziale lungo l'asse x mentre la seconda (1.10) lungo l'asse y, quindi una crescita d'intensità lungo queste due direzioni.

In altre parole, la matrice (1.9) permette di evidenziare le linee orizzontali dell'immagine, confrontando le intensità tra due pixel lungo la stessa colonna, la (1.10) consente di evidenziare le linee verticali.

Una caratteristica interessante di queste matrici è che la somma di tutti i coefficienti da 0, ciò indica che daranno come risultato 0 in aree con intensità costante, comportamento conforme all'operatore derivata.

In seguito si possono apprezzare gli effetti di questo operatore, l'immagine risulta però particolarmente scura quindi la si somma a quella originale per ottenere il risultato voluto, con contorni particolarmente accentuati; (Figura 1.11).

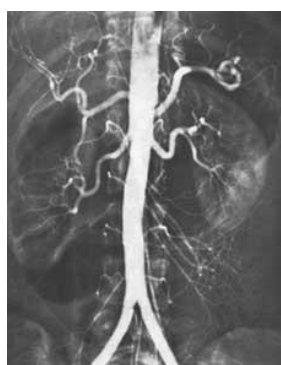
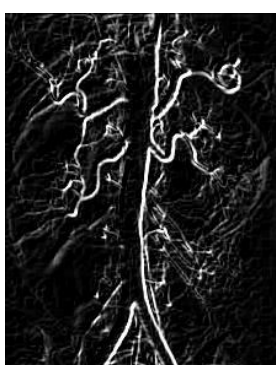


Immagine Originale



Filtro Prewitt

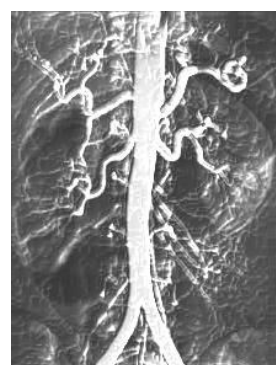


Immagine Originale + Prewitt

Figura 1.11

## Filtro Sobel

Questa tipologia di filtro ha le stesse caratteristiche del filtro di Prewitt, lo scopo è quello di calcare i contorni della figura, però vengono utilizzate formule, e quindi kernel, leggermente differenti rispetto ai precedenti.

$$g_x = \frac{df}{dx} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{df}{dy} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Equazione 1.8

La differenza sostanziale con Prewitt è che viene data maggior rilevanza al pixel centrale, assegnando un coefficiente 2.

Le equazioni precedenti si traducono nelle due maschere filtro seguenti:

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

Figura 1.12

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

Figura 1.13

La prima matrice (1.12) serve ad evidenziare gli edge presenti lungo l'asse x, mentre la seconda (1.13) risalta quelli lungo l'asse y, come già spiegato per i filtri di prewitt.

Nelle immagini seguenti si possono apprezzare gli effetti del filtro sobel sull'immagine, cogliendo anche le analogie con i filtri di Prewitt:

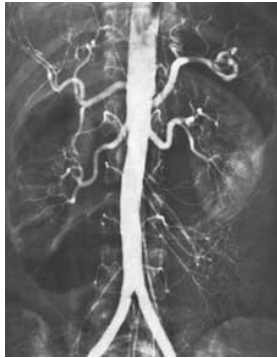
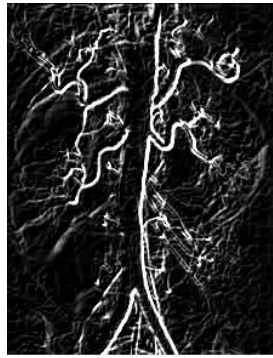
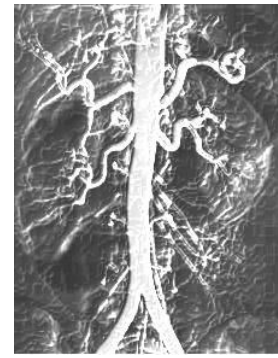


Immagine Originale



Filtro Sobel



Filtro Sobel + Originale

**Figura 1.14**

### **Metodo Laplaciano**

Questo metodo di sharpening si basa sull'utilizzo delle derivate seconde, che, come accennato precedentemente, oltre ad aver una maggiore capacità di evidenziare i dettagli rispetto alla derivata prima sono anche di più semplice implementazione.

Innanzitutto si effettua una formulazione discreta e successivamente si costruisce una maschera filtro, che abbia la proprietà dell'isotropia, cioè che la sua risposta sia indipendente dalla direzione delle discontinuità dell'immagine a cui esso è applicato.

Il Laplaciano è uno degli operatori più semplici che soddisfa questa caratteristica, il quale, applicato ad una funzione bidimensionale  $f(x,y)$  ha la seguente forma:

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

**Equazione 1.9**

con:

$$\frac{\delta^2 f}{\delta x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

**Equazione 1.10**

$$\frac{\delta^2 f}{\delta y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$

Equazione 1.11

Andando a sostituire le equazioni 1.10 e 1.11 nella 1.9 si ottiene la formulazione estesa:

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

Equazione 1.12

L'equazione precedente (1.12) può essere implementata mediante la seguente maschera filtro (1.15):

0	1	0
1	-4	1
0	1	0

Figura 1.15

1	1	1
1	-8	1
1	1	1

Figura 1.16

Aggiungendo due termini all'equazione (1.12) di forma simile all'equazione (1.9) si possono aggiungere le direzioni diagonali, ottenendo la maschera raffigurata nella figura (1.16).

Il valore -8 centrale è dovuto all'aggiunta di due termini  $-2f(x,y)$  introdotti dalle diagonali.

Il Laplaciano, come già esposto precedentemente, mette in risalto i contorni delle immagini ma pone in secondo piano le zone in cui l'intensità varia più lentamente, restituendo un'immagine con linee che spiccano su un fondo scuro (Figura 1.17).

Per recuperare le tonalità che variano più gradualmente si può sommare l'immagine ottenuta con il Laplaciano all'immagine originale, in questo modo si otterrà un risultato visivamente più simile all'originale, quindi recuperando lo sfondo, ma con i contorni che spiccano più chiaramente, obiettivo del Laplaciano.

Ciò che ho descritto a parole può essere tradotto in formule:



$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$$

Equazione 1.13

In cui  $f(x,y)$  è la mia immagine in input mentre  $g(x,y)$  è il risultato finale (1.18).



Figura 1.17

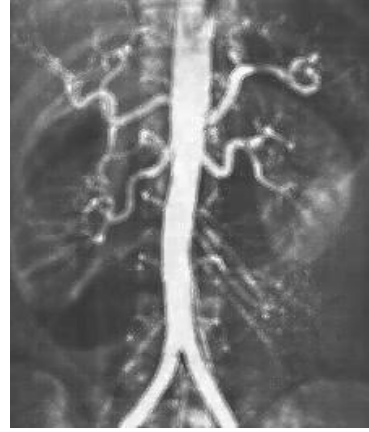


Figura 1.18

### **Unsharp masking e filtraggio Highboost**

Il nocciolo della tecnica dell'unsharp masking consiste nel sottrarre una versione sfocata dell'immagine originale all'immagine stessa col fine di accentuarne i contorni.

Il processo può esser riassunto per punti:

1. sfocare l'immagine originale
2. sottrarre immagine sfocata dall'originale
3. aggiungere la maschera (quella ottenuta al punto 2) all'originale

Si può assegnare  $\overline{f(x, y)}$  all'immagine sfocata, per ottenere la maschera si procede con la sottrazione di  $\overline{f(x, y)}$  all'immagine di input:

$$g(x, y)_{mask} = f(x, y) - \overline{f(x, y)}$$

Equazione 1.14

Infine si aggiunge la maschera all'immagine originale, moltiplicata per un opportuno coefficiente di proporzionalità:

$$g(x,y) = f(x,y) + K * g(x,y)_{mask}$$

Equazione 1.15

Il coefficiente K, quindi la porzione pesata di maschera che verrà aggiunta all'originale, determina la tipologia di filtraggio effettuata; infatti se  $K=1$  è un'operazione di unsharp mask, se  $K>1$  si ha il filtraggio highboost.

Sottraendo il segnale sfocato a quello in input si evidenziano nettamente i punti in cui si verifica il cambio di pendenza dell'intensità del segnale, sommando il tutto all'immagine originale si ottiene un'immagine che ha come base quella che avevo in ingresso ma con la differenza di avere i contorni in risalto.

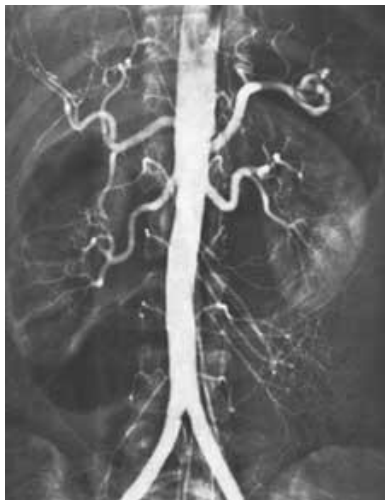


Figura 1.19

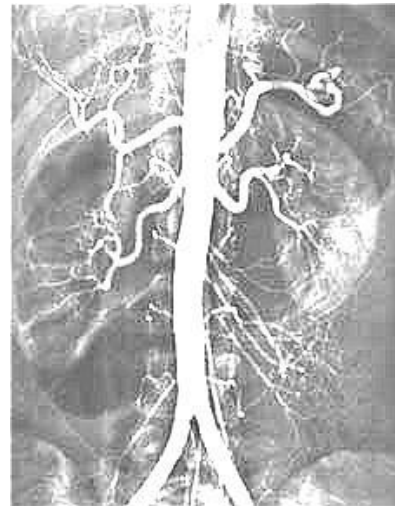
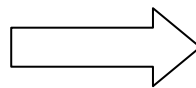


Figura 1.20

### ➤ Filtri Non Lineari

Non utilizzano, come suggerisce il titolo, l'operazione di convoluzione lineare, ma usano altri metodi di filtraggio.

Il principale filtro non lineare è il filtro Mediano, esso si basa sulle statistiche d'ordine, la cui risposta si traduce nell'ordinare i pixel contenuti all'interno della zona dell'immagine inclusa nella matrice di filtro ed andare a sostituire al pixel centrale della matrice il valore centrale della sequenza ordinata; questo processo comporta una perdita d'informazioni di tipo spaziale sui pixel dell'intorno.

## **Filtro Mediano**

Il funzionamento generale del filtro è stato descritto brevemente nelle righe antecedenti.

Questa tipologia di filtri, insieme al filtro di media, è molto diffusa per la riduzione del rumore eventualmente presente nelle immagini digitali, in particolare per alcuni tipi di rumore casuale come quello a impulso (sale e pepe) è particolarmente efficace e determina una sfocatura minore rispetto al corrispondente filtro di media lineare della stessa dimensione.

Per applicare il filtro per prima cosa si deve determinare la dimensione della matrice filtro che andrà a scorrere l'immagine, una volta scelta la dimensione voluta la matrice scorre l'immagine e ordina i pixel dell'immagine che sono all'interno della matrice e ne individua la mediana (la mediana è l'elemento al centro della serie di valori, cioè che ha un numero uguale di termini sia alla sua destra che alla sua sinistra); infine sostituisce la mediana al pixel dell'immagine corrispondente al pixel centrale della matrice filtro (Figura 1.21).

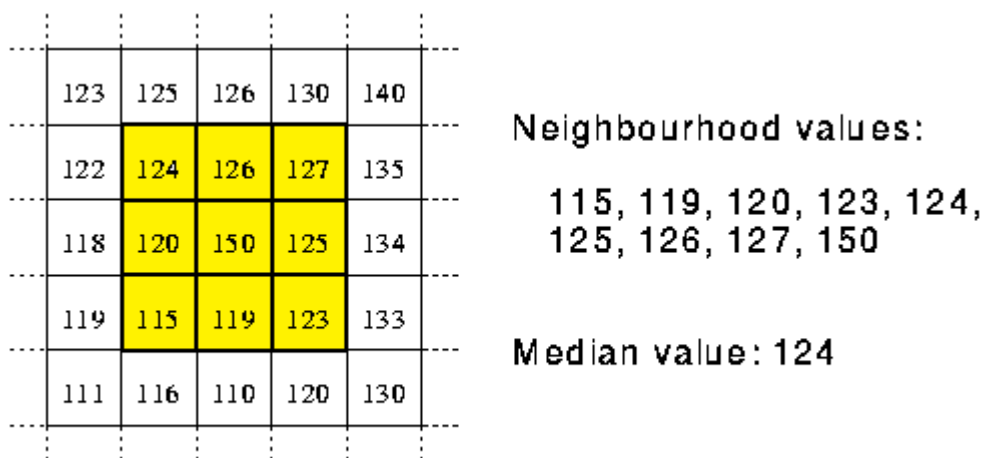


Figura 1.21

L'effetto del filtro mediano è quello di forzare i pixel dell'immagine a non variare in maniera eccessiva il proprio livello di intensità rispetto ai pixel dell'intorno.

Questo particolare effetto fa in modo che pixel con livelli d'intensità eccessivamente differenti rispetto ai vicini vengano eliminati, questo risulta estremamente vantaggioso per il rumore sale e pepe.

Al contrario del filtro di media che crea nuove gradazioni di colore attenuando in maniera discriminata tutte le frequenze spaziali e causando di conseguenza sfocature e perdite di dettagli, il filtro mediano non deteriora l'immagine, ma elimina semplicemente i picchi.

Si possono vedere gli effetti di questo filtro nelle seguenti immagini di un aorta, nella figura (1.16) si ha l'immagine distorta dal rumore sale e pepe, in quella successiva (1.17) si ha il risultato dell'applicazione del filtro mediano.

I risultati parlano da soli, nonostante una lieve sfocatura l'immagine è nettamente migliorata.

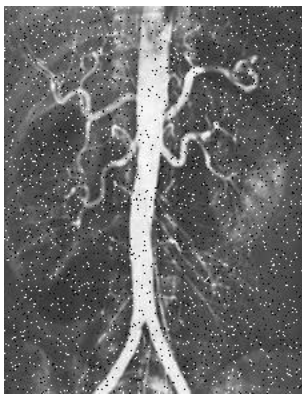


Figura 1.22

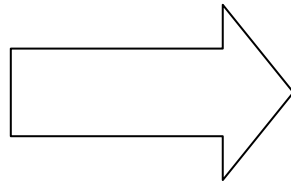


Figura 1.23

### **Filtro Anisotropo**

Questo tipo di filtraggio sfrutta il filtraggio gaussiano descritto precedentemente, ma, a differenza di quest'ultimo, l'immagine non viene sfocata indiscriminatamente.

Perona e Malik hanno proposto un metodo iterativo basato sulla funzione di diffusione, che prevede l'applicazione selettiva del filtro gaussiano all'immagine originale.

Iterando l'applicazione del filtro gaussiano sulla stessa immagine si ottiene l'eliminazione progressiva del rumore lasciando inalterate le strutture dell'immagine.

Il filtro viene applicato a seconda delle caratteristiche locali dell'immagine, mediante il coefficiente di diffusione  $c(x, y, t)$ :

$$I_t = \nabla[c(x, y, t)\nabla I]$$

Equazione 1.16

in cui  $t$  rappresenta un fattore di scala.

Per far sì che i contorni restino inalterati ed il filtro venga applicato soltanto alle zone interne,  $c$  dev'esser prossima a 0 nei contorni e circa 1 nelle zone interne.

Un ottima funzione per individuare i contorni è il gradiente, per un modulo del gradiente elevato probabilmente si è di fronte ad un bordo.

Utilizzando il gradiente  $c$  può esser definita come segue:

$$c = g(\|\nabla I(x, y, t)\|)$$

**Equazione 1.17**

con  $g$  equivalente ad una funzione che lega  $c$  al modulo del gradiente, definita attraverso due forme distinte:

$$g(\|\nabla I\|) = e^{-\left(\frac{\|\nabla I\|}{K}\right)^2}$$

**Equazione 1.18**

$$g(\|\nabla I\|) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}$$

**Equazione 1.19**

Come si può notare, queste funzioni hanno valore 1 in corrispondenza di valori nulli del gradiente e, all'aumentare del gradiente, quindi nelle zone dei contorni, queste funzioni tendono a 0 e quindi tende a 0 anche il coefficiente di diffusione  $c$ .

$K$  è un parametro che indica il modulo del gradiente in corrispondenza del quale si avrà il massimo sfocamento, esso è implementabile dall'utente a seconda delle immagini; più è alto questo valore, meno selettiva sarà l'azione del filtro.

Le immagini seguenti permettono di visualizzare gli effetti del filtro al variare di  $K$ :



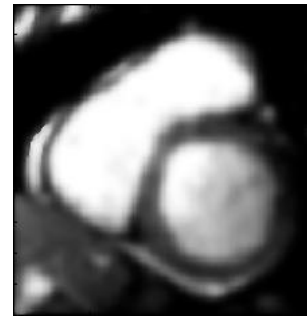
Originale



Anisotropo K=6



Anisotropo K=20



Anisotropo K=80

**Figura 1.24**

Come si può evincere dalle immagini precedenti (1.24), aumentando il valore di K si applica smoothing a valori sempre più elevati di gradiente, tanto da andare ad interessare anche i contorni.

# CAPITOLO 2

## FILTRAGGIO NEL DOMINIO DELLA FREQUENZA

---

Gli strumenti di acquisizione e trasmissione di segnali oggi giorno utilizzano segnali digitali quindi è necessario effettuare un'operazione di campionamento per trasformare i segnali tempo continui in tempo discreti. Teoricamente il procedimento è semplice, basta moltiplicare il segnale tempo continuo originale per un treno di impulsi con periodo  $\Delta T$ :

$$\tilde{f}(t) = \sum_{n=-\infty}^{+\infty} f(t)\delta(t - n\Delta T)$$

Equazione 2.1

Avendo così ottenuto il segnale in ingresso come un insieme di campioni discreti, per passare al dominio della frequenza si deve utilizzare la Trasformata di Fourier Discreta (DFT).

Questo discorso è analogo per le immagini, essendo segnali bidimensionali, si possono trasportare nel dominio della frequenza attraverso la DFT 2-D appunto bidimensionale:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

Equazione 2.2

in cui  $f(x,y)$  è un'immagine digitale di dimensioni  $M \times N$  e  $u=0,1,2,\dots,M-1$  e  $v=0,1,2,\dots,N-1$ .

Il problema sta nel fatto che la funzione campionata nel dominio spaziale ha una trasformata che è una ripetizione periodica di quella del segnale originale; ciò significa che  $F(u,v)$  è infinitamente periodica lungo  $u$  e  $v$ .

Questo dà origine a possibili interferenze tra i vari periodi durante il processo convolutivo (Figura 2.1) causando il cosiddetto errore di wraparound.

A questo si può ovviare effettuando sull'immagine un processo di padding nel dominio spaziale, prima di effettuare la trasformata di Fourier.

Nella figura (2.1) è mostrato l'errore di wraparound per le funzioni monodimensionali, il discorso vale ugualmente estendendolo alle due dimensioni.

Avendo due funzioni  $f(x,y)$  e  $h(x,y)$  composte da  $A \times B$  e  $C \times D$  campioni aggiungiamo alcuni zero in modo che abbiano le stesse dimensioni, cioè  $P$  e  $Q$ .

Quindi l'errore di wraparound si può evitare impostando:

$$P \geq A + C - 1$$

$$Q \geq B + D - 1$$

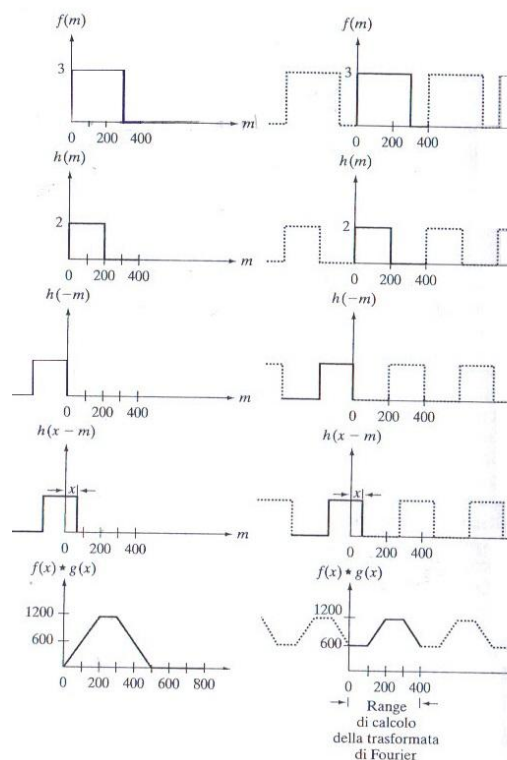


Figura 2.1



Filtraggio nel dominio dello spazio e della frequenza sono correlati strettamente dal teorema della convoluzione, ricordando la formula della convoluzione:

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

Equazione 1.1

Il teorema della convoluzione si può esporre nel modo seguente:

$$F\{f(t) \star h(t)\} = H(\mu)F(\mu)$$

Equazione 2.3

Questa formula sta ad indicare che la trasformata di Fourier della convoluzione tra due funzioni nel dominio spaziale è uguale al prodotto della trasformata delle due funzioni nel dominio della frequenza, il che è valido anche viceversa:

$$F\{f(t) \star h(t)\} \Leftrightarrow H(\mu)F(\mu)$$

Equazione 2.4

Il discorso è ugualmente corretto se si ribaltano le parti in gioco, cioè si può affermare che la convoluzione nel dominio della frequenza è analoga alla moltiplicazione nel dominio spaziale.

$$f(t)h(t) \Leftrightarrow H(\mu) \star F(\mu)$$

Equazione 2.5

Il filtraggio in frequenza consiste quindi, nel modificare la trasformata di Fourier di un'immagine e calcolarne poi l'antitrasformata, per tornare nel dominio spaziale.

Di seguito è proposta la formulazione base del filtraggio in frequenza:

$$g(x, y) = F^{-1}[H(u, v)F(u, v)]$$

Equazione 2.6

$F^{-1}$  è la IDFT, cioè l'antitrasformata discreta di Fourier che permette di tornare nel dominio spaziale,  $F(u, v)$  è la trasformata dell'immagine di input,  $H(u, v)$  è

la funzione filtro che modifica la trasformata originale e  $g(x,y)$  è l'immagine filtrata nel dominio spaziale, ovvero il risultato del filtraggio in frequenza.

$F$  e  $H$  sono matrici di dimensioni  $M \times N$ , cioè della stessa dimensione dell'immagine in ingresso, il prodotto  $F(u,v)H(u,v)$  non è un prodotto matriciale ma puntuale.

I principali termini dell'equazione (2.6)  $H(u,v)$  e  $F(u,v)$  possono essere esaminati più in dettaglio:

### $F(u, v)$

Questa è la trasformata dell'immagine in ingresso, ottenuta, come già detto, con la DFT.

Una volta ottenuta, per elaborarla, è utile centrarla all'origine del nostro sistema di riferimento, in modo da avere le basse frequenze nell'origine e quindi riuscire a sovrapporre in maniera più efficiente il filtro.

$$f(x, y)e^{j2\pi(\frac{u_0x}{M} + \frac{v_0y}{N})} \Leftrightarrow F(u - u_0, v - v_0)$$

Equazione 2.7

Inserendo  $u_0$  e  $v_0$  uguali rispettivamente a  $M/2$  e  $N/2$  si pone la trasformata al centro del sistema di riferimento, in modo tale da avere un periodo completo all'interno del mio intervallo  $[0, M-1]$  e  $[0, N-1]$ .

Andando a sostituire nell'equazione (2.7) ad  $u_0$   $M/2$  ed a  $v_0$   $N/2$  si ottiene una forma esponenziale di questo tipo:

$$e^{j\pi x} = (-1)^x$$

Equazione 2.8

Andando a sostituire la (2.8) nella (2.7) si ottiene questa formulazione finale:

$$f(x, y)(-1)^{x+y} = F(u - \frac{M}{2}, v - \frac{N}{2})$$

Equazione 2.9

La figura seguente mostra, visivamente, quanto descritto in precedenza mediante formule.

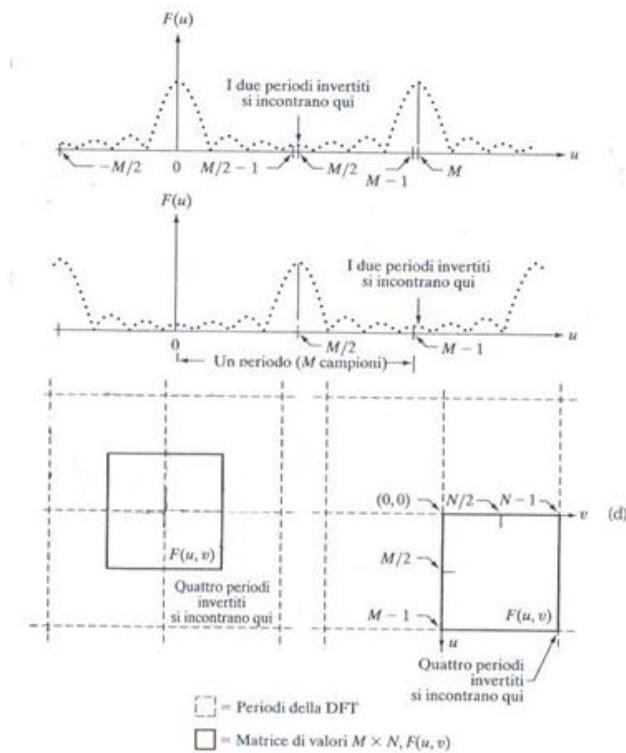


Figura 2.2

La trasformata di Fourier può essere espressa anche nelle singole componenti, infatti rappresenta un vettore di una certa lunghezza (Modulo) che forma un dato angolo (Fase) con l'origine del sistema di riferimento.

$$F(u, v) = |F(u, v)|e^{j\varphi(u, v)}$$

Equazione 2.10

$$|F(u, v)| = [R(u, v)^2 + I(u, v)^2]$$

Equazione 2.11

$$\varphi(u, v) = \tan^{-1} \left( \frac{I(u, v)}{R(u, v)} \right)$$

Equazione 2.12

Lo spettro (modulo) determina le ampiezze delle onde sinusoidali che si combinano per formare il segnale di input, maggiore è lo spettro ad una data frequenza, maggiore sarà l'ampiezza delle onde di quella particolare frequenza.

Le basse frequenze di un immagine sono relate alle componenti di intensità che variano gradualmente mentre le alte frequenze sono legate alle brusche variazioni d'intensità.

La fase invece, misura lo spostamento delle sinusoidi rispetto all'origine, si ottiene una matrice di fase i cui valori sono gli angoli, essa fornisce informazioni sulla posizione degli oggetti nell'immagine.

Diversamente a ciò che si potrebbe intuitivamente pensare, la fase ha un peso più rilevante nella ricostruzione delle immagini dallo spettro, come si può notare nelle immagini seguenti.

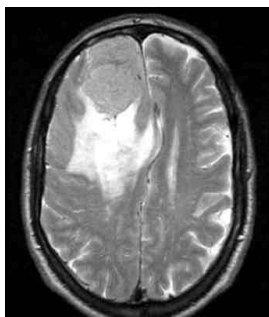


Figura 2.3

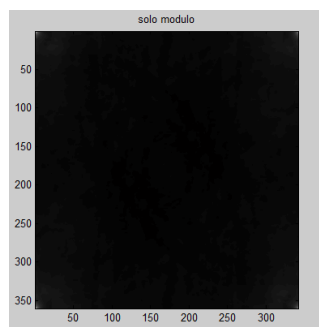


Figura 2.4

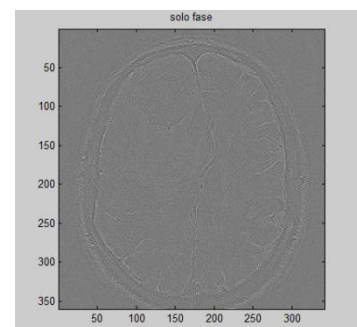


Figura 2.5

L'immagine ricostruita unicamente con la fase non è chiarissima ma è comunque possibile intravedere i contorni dell'immagine originale, a differenza di quella ricostruita col modulo, che da una schermata uniformemente nera.

## $H(u,v)$

Questo termine rappresenta il filtro nel dominio della frequenza, che andrà a modificare la trasformata dell'immagine per ottenere gli effetti desiderati.

I filtri possono derivare da filtri spaziali oppure essere implementati direttamente in frequenza.

- Filtri derivati da filtri spaziali

- filtro laplaciano
- unsharp Masking e filtraggio highboost

### Laplaciano

Il Laplaciano può essere implementato nel dominio della frequenza utilizzando:

$$H(u, v) = -4\pi^2 \left[ \left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2 \right]$$

Equazione 2.13

avendolo impostato al centro del rettangolo della frequenza di dimensioni P e Q.

Impostato il filtro si ottiene l'immagine laplaciana ed infine quella filtrata, aggiungendo all'originale quella laplaciana.

Il processo è simile a quello visto spazialmente, espresso in formule si ha che:

$$\nabla^2 f(x, y) = F^{-1}\{H(u, v)F(u, v)\}$$

Equazione 2.14

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$$

Equazione 2.15

con  $c = 1$  poiché  $H(u, v)$  è negativa, altrimenti  $c = -1$ .

### Unsharp Masking e Filtraggio Highboost

Per effettuare questo tipo di filtraggio è necessario introdurre un filtro passa basso, che mantiene inalterate le basse frequenze ed azzerà le frequenze più elevate a partire da una particolare frequenza denominata frequenza di taglio.

$$g_{mask}(x, y) = f(x, y) - f_{LP}(x, y)$$

Equazione 2.16

$$f_{LP} = F^{-1}[H_{LP}(u, v)F(u, v)]$$

Equazione 2.17

$f_{LP}$  è un'immagine sfocata analoga al caso spaziale, che successivamente andrà sottratta a quella originale:

$$g_{mask}(x, y) = f(x, y) + k g_{mask}(x, y)$$

Equazione 2.18

con  $k = 1$  se si desidera l'unsharp masking,  $k > 1$  per il filtraggio Highboost.

## • Filtri implementati in Frequenza

I filtri in frequenza, come quelli spaziali, possono essere del tipo Passa Basso (Low Pass) o Passa Alto (High Pass).

I filtri passa basso sono utilizzati per lo smoothing delle immagini, infatti eliminano le alte frequenze, che sono indicatrici delle repentine variazioni di intensità nell'immagini.

Quelli passa alto invece, funzionano in maniera opposta, eliminano le basse frequenze e mantenendo immutate quelle alte, quindi effettuando lo sharpening dell'immagine.

Esistono tre tipi differenti di filtro:

- filtri ideali
- filtri di butterworth
- filtri gaussiani

Queste tipologie sono presenti sia come passa alto che come passa basso.

○ *Filtri Passa Basso*

▪ Ideali

Un filtro di questo tipo è quello che lascia passare tutte le frequenze all'interno di un cerchio di raggio  $D_0$  (denominata frequenza di taglio), centrato nell'origine, ed elimina (o attenua) tutte le frequenze al di fuori di questo cerchio.

$$H_{LP}(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{se } D(u, v) \geq D_0 \end{cases}$$

Equazione 2.19

in cui  $D(u, v)$  è la distanza di un punto dal centro del rettangolo della frequenza:

$$D(u, v) = \left[ \left( u - \frac{P}{2} \right)^2 + \left( v - \frac{Q}{2} \right)^2 \right]$$

Equazione 2.20

Il termine “ ideale “ deriva dal fatto che le frequenze all'interno del cerchio rimangono inalterate mentre quelle all'esterno vengono completamente annullate, una transizione così netta è irrealizzabile in hardware, è ottenibile solamente attraverso simulazioni al computer.

La figura (2.6) è un filtro passa basso ideale, la parte bianca è la zona in cui le frequenze passano, quella nera quella in cui vengono eliminate mentre la figura (2.7) è la visione 3-D del filtro.

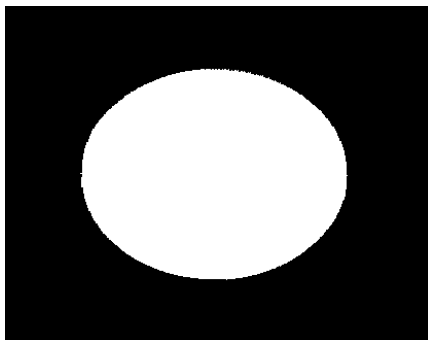


Figura 2.6

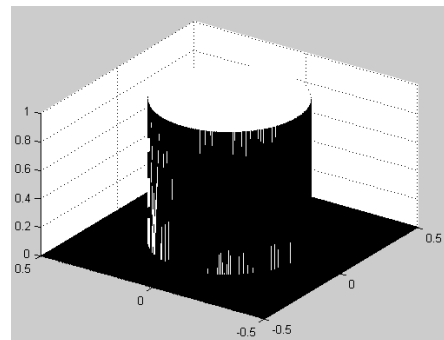


Figura 2.7

- Butterworth

I filtri di Butterworth sono caratterizzati da un valore,  $n$  che indica il loro ordine e da una frequenza di taglio  $D_0$ .

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

Equazione 2.21

Questa tipologia di filtri, a differenza di quelli ideali, non hanno una discontinuità così netta tra le frequenze filtrate e quelle che vengono lasciate passare.

Per questo la frequenza di taglio è definita in un modo più “elastico”, nei casi in cui si ha una transizione graduale, come questo, la frequenza di taglio è indicata come il luogo dei punti in cui  $H(u,v)$  è minore di una determinata frazione del suo valore massimo.

La figura (2.8) rappresenta un filtro di butterworth di ordine 1, la (2.9) un filtro di ordine 5, la (2.10) uno di ordine 2, come si può osservare, varia in maniera evidente la pendenza della curva.

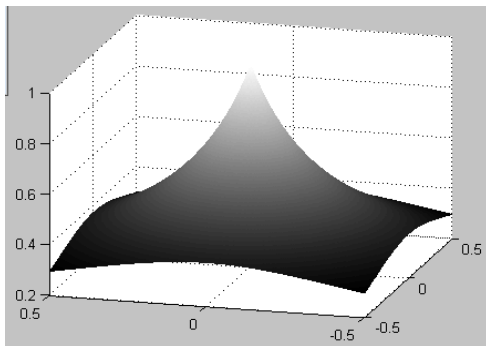


Figura 2.8

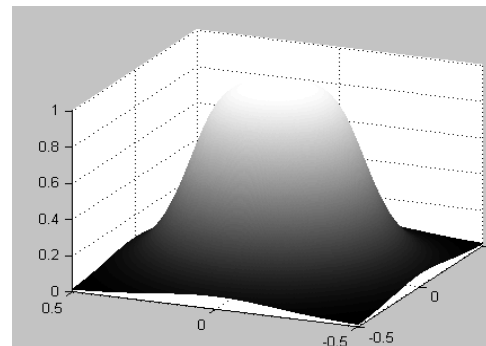


Figura 2.9

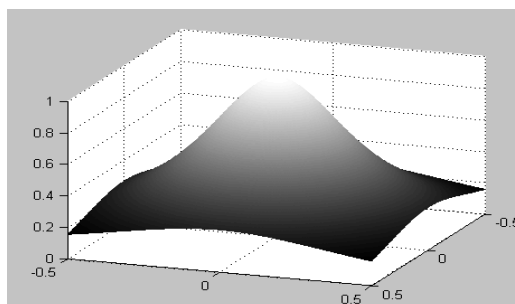


Figura 2.10



- Gaussiani

Anche questa tipologia di filtri, come quelli di Butterworth, non hanno una separazione brusca al valore della frequenza di taglio, ma hanno una forma “ a campana “.

L’implementazione avviene mediante equazioni di questa tipologia:

$$H_{GaussLP} = e^{-D^2(u,v)/2D_0^2}$$

Equazione 2.22

$D_0$  è la frequenza di taglio, aumentandola o riducendola si può allargare o restringere la campana della gaussiana.

La figura (2.11) rappresenta un filtro Gaussiano implementato con l’Equazione (2.22) con  $D_0 = 0.3$ .

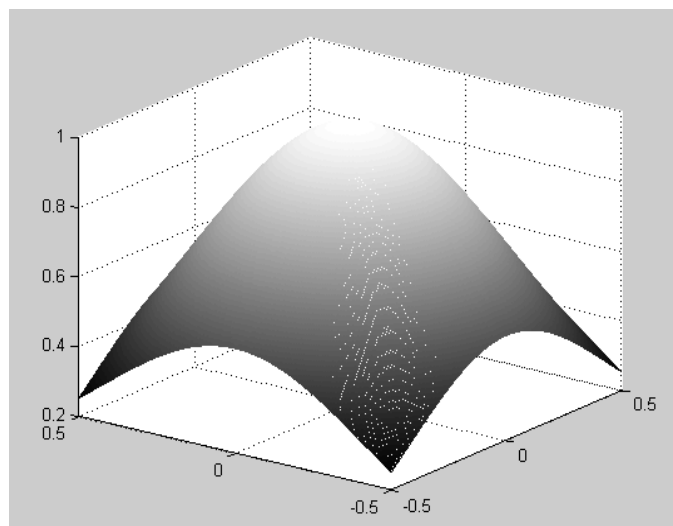


Figura 2.11

- *Filtri Passa Alto*

I filtri passa alto operano in modo opposto a quelli passa basso, infatti eliminano le basse frequenze e lasciano inalterate quelle alte.

In questo modo si eliminano le componenti con intensità variabile gradualmente, come lo sfondo, e si accentuano quelle che variano rapidamente, come i contorni e i dettagli fini.

Proprio per questo i passa alto sono utilizzati per lo sharpening delle immagini. Anche tra i passa alto vi sono le tre tipologie di filtri elencati precedentemente per i passa basso.

Tutti i filtri sono matrici di dimensioni  $P \times Q$ , cioè di grandezza uguale alle immagini dopo aver effettuato il padding.

- Ideali

Il filtro passa alto ideale si ottiene a partire dal filtro passa basso, sempre ideale, attraverso la formula:

$$H_{HP} = 1 - H_{LP}$$

Equazione 2.23

$$H_{HP}(u, v) = \begin{cases} 1 & \text{se } D(u, v) \geq D_0 \\ 0 & \text{se } D(u, v) \leq D_0 \end{cases}$$

Equazione 2.24

In frequenza, ha la forma di un cilindro con un netto passaggio tra le frequenze tagliate e quelle lasciate immutate, sempre con  $D_0$  come frequenza di taglio e  $D(u, v)$  data dall'equazione (2.20).

Il funzionamento è l'esatto contrario di quello passa basso, le frequenze all'interno del cilindro vengono tagliate, mentre quelle all'esterno rimangono immutate.

La figura (2.12) rappresenta la visione 3-D del filtro high pass, invece la figura (2.13) è la visione 2-D filtro high pass.

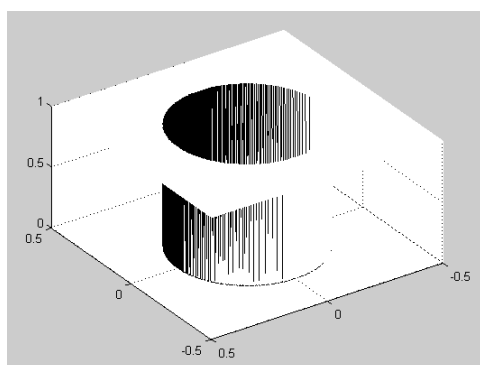


Figura 2.12

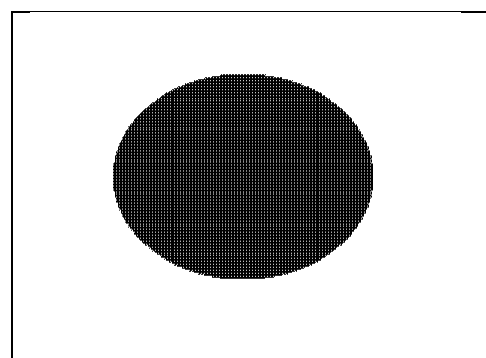


Figura 2.13

- Filtri di Butterworth

Questo filtro può essere considerato una forma intermedia tra il filtro ideale e quello gaussiano, ha un andamento più graduale rispetto all'ideale ma più netto rispetto al gaussiano.

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

Equazione 2.25

$$H_{ButterHP} = 1 - H_{ButterLP}$$

Equazione 2.26

con  $D_0$  frequenza di taglio ed  $n$  che indica l'ordine del filtro.

Avendo un passaggio più graduale rispetto a quelli ideali si otterranno risultati più tenui.

La figura (2.14) rappresenta un filtro di butterworth di ordine 1, la (2.15) un filtro di ordine 2, la (2.16) uno di ordine 5, come si può osservare, varia in maniera evidente la pendenza della curva.

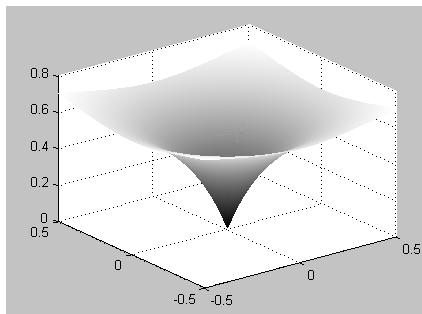


Figura 2.14

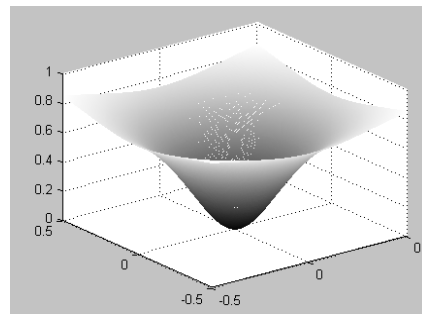


Figura 2.15

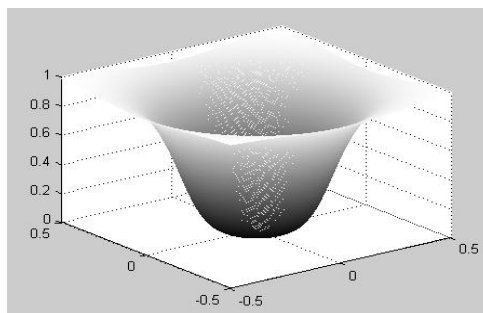


Figura 2.16

- Filtri Gaussiani

Il loro nome deriva dal fatto che hanno una distribuzione gaussiana in frequenza, la cui frequenza di taglio è posta a distanza  $D_0$  dal centro del rettangolo della frequenza.

$$H_{GaussLP} = 1 - H_{GaussHP} = 1 - e^{-D^2(u,v)/2D_0^2}$$

Equazione 2.27

I risultati sono ancora più tenui dei due casi precedenti.

La figura (2.17) rappresenta un filtro gaussiano implementato con l'Equazione (2.27) con  $D_0 = 0.3$ .

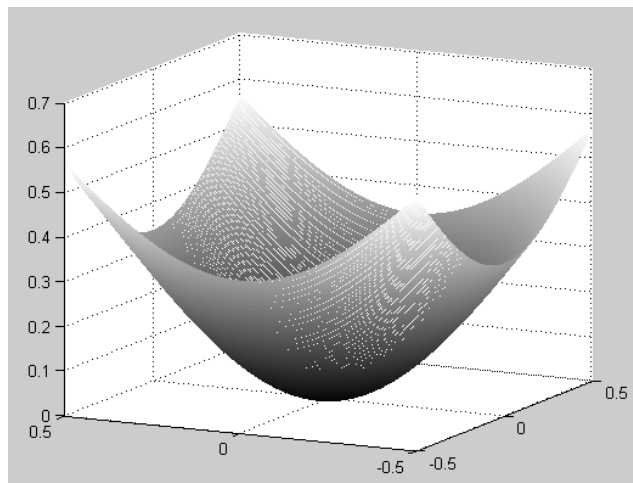


Figura 2.17

Per applicare un filtraggio nel dominio della frequenza bisogna compiere una serie di operazioni, riassumibili in questa semplice scaletta:

1. avendo l'immagine in input  $f(x,y)$  di dimensioni  $M \times N$ , si ricavano  $P$  e  $Q$ , cioè le dimensioni necessarie per effettuare il padding dell'immagine, se immagine e filtro hanno dimensioni uguali di solito si utilizza  $P = 2M$   $Q = 2N$ .

2. si effettua il padding dell'immagine originale formando un'immagine padded di dimensioni P e Q,  $f_p(x, y)$  aggiungendo il numero necessario di zeri.
3. si moltiplica  $f_p(x, y)$  per  $(-1)^{x+y}$  in modo da aver la trasformata centrata nel rettangolo della frequenza.
4. si calcola la DFT di  $f_p(x, y)$ ,  $F(u, v)$ .
5. si imposta una funzione filtro  $H(u, v)$  di dimensioni P x Q con centro nelle coordinate P/2 e Q/2.
6. si trasla la trasformata dell'immagine filtrata nella posizione originale moltiplicandola nuovamente per  $(-1)^{x+y}$
7. si ottiene l'immagine filtrata con:

$$g_p(x, y) = \{real [F^{-1}[G(u, v)]]\}$$

Equazione 2.28

Viene estratta solamente la parte reale per evitare eventuali componenti parassite complesse che potrebbero inficiare i risultati con errori di calcolo.

8. il risultato finale si ottiene estraendo la regione di dimensioni M x N dalla  $g_p(x, y)$ .

# CAPITOLO 3

## REALIZZAZIONE INTERFACCIA GRAFICA PER IL FILTRAGGIO DELLE IMMAGINI

---

Il mio progetto di tesi contempla anche la realizzazione di un interfaccia grafica per il filtraggio delle immagini.

Questa risulta particolarmente comoda per visualizzare gli effetti sulle immagini dei vari tipi di filtro, potendo impostare i parametri di ognuno mediante apposite caselle poste a fianco di ogni pulsante.

L'interfaccia si presenta all'utente come mostrato in figura (3.1):

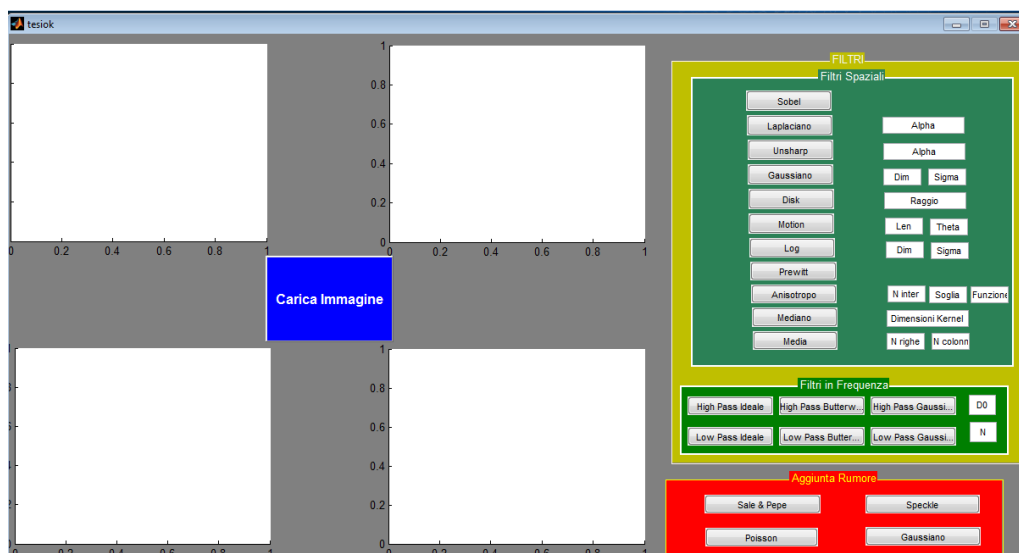


Figura 3.1

Nei riquadri in bianco verranno visualizzate le immagini, disposte in modo tale da favorire il confronto tra di esse, e quindi la valutazione dell'efficacia dei filtri.

Il pulsante centrale, come indicato, serve per caricare le immagini da elaborare; il codice che viene eseguito una volta premuto tale pulsante permette di importare le immagini in qualsiasi formato, compreso Dicom.

Una volta premuto appare una finestra di ricerca, in cui è possibile scegliere le immagini che poi verranno caricate. (Figura 3.2).

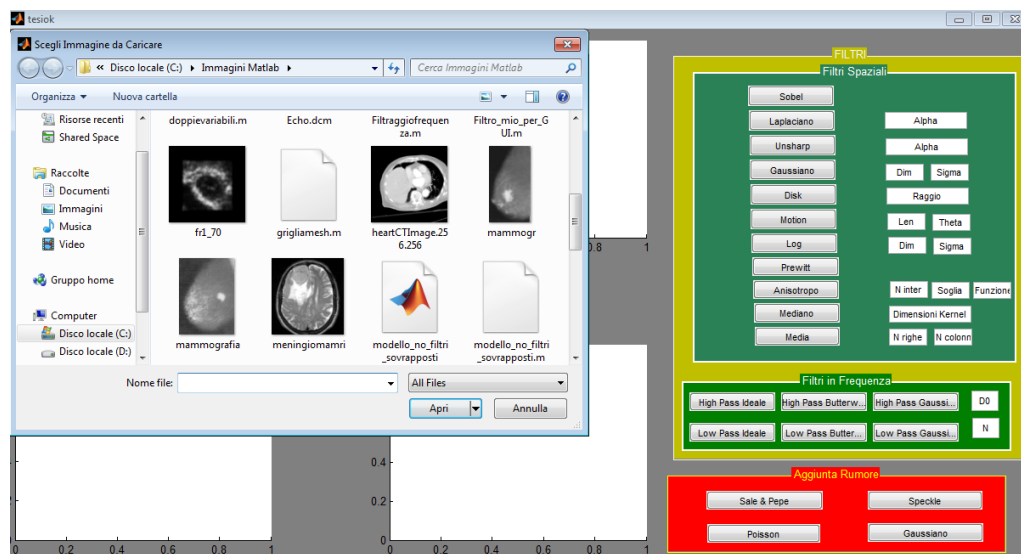


Figura 3.2

Nella sezione destra dell'interfaccia vi sono i pulsanti che permettono di elaborare le immagini col filtro che si ritiene più opportuno, modificandone i parametri a piacere.

Se quest'ultimi non vengono specificati, vengono utilizzati i parametri di default caratteristici di ogni filtro; per i filtri in frequenza invece, non avendo utilizzato le funzioni già disponibili in Matlab, la frequenza di taglio di default è stata impostata a 0.3, e l'ordine, per i filtri di butterworth, a 2.

Sono stati implementati anche alcuni filtri già presenti in Matlab, come il filtro di sharpening log e quelli di smoothing motion e disk.

Sempre sulla destra, nella zona inferiore, è possibile aggiungere il rumore, per simulare i disturbi che possono affliggere le immagini e poter testare l'efficacia dei vari filtri nell'eliminarlo.

Nella figura (3.3) è rappresentata l'interfaccia in cui sono applicati vari tipi di rumore all'immagine originale, partendo da in alto a sinistra ed andando in

sensu orario si hanno rispettivamente il rumore sale e pepe, speckle, gaussiano, e di poisson.

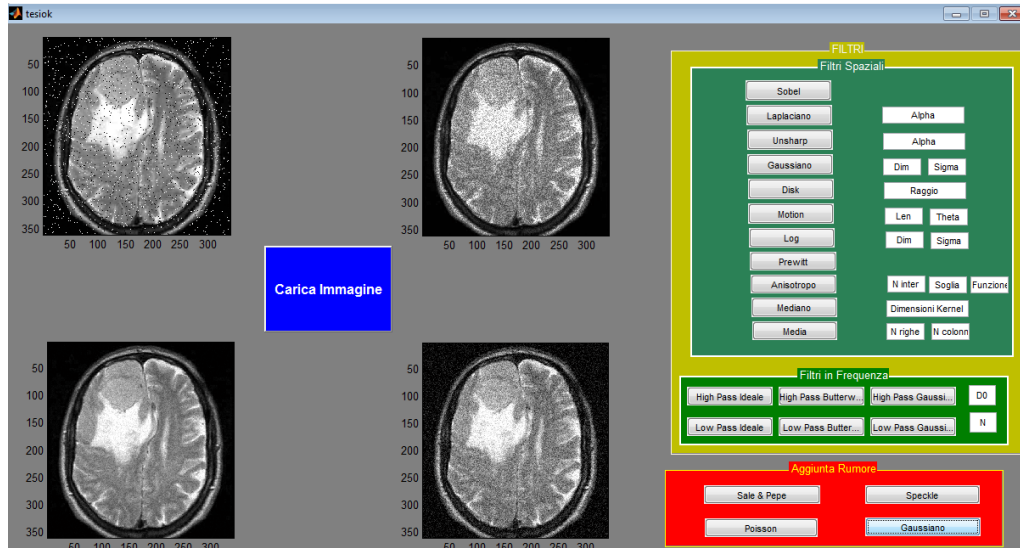


Figura 3.3

Nella figura seguente sono mostrati gli effetti di alcuni filtri di smoothing su un immagine affetta da un rumore impulsivo (sale e pepe), in particolare ho utilizzato il filtro mediano, filtro di media e il filtro gaussiano, a parità di dimensioni del kernel (10x10).

La figura (3.4) in alto a sinistra raffigura l'immagine originale affetta dal rumore, avanzando in sensu orario si hanno in successione il filtro mediano, filtro gaussiano e quello di media.

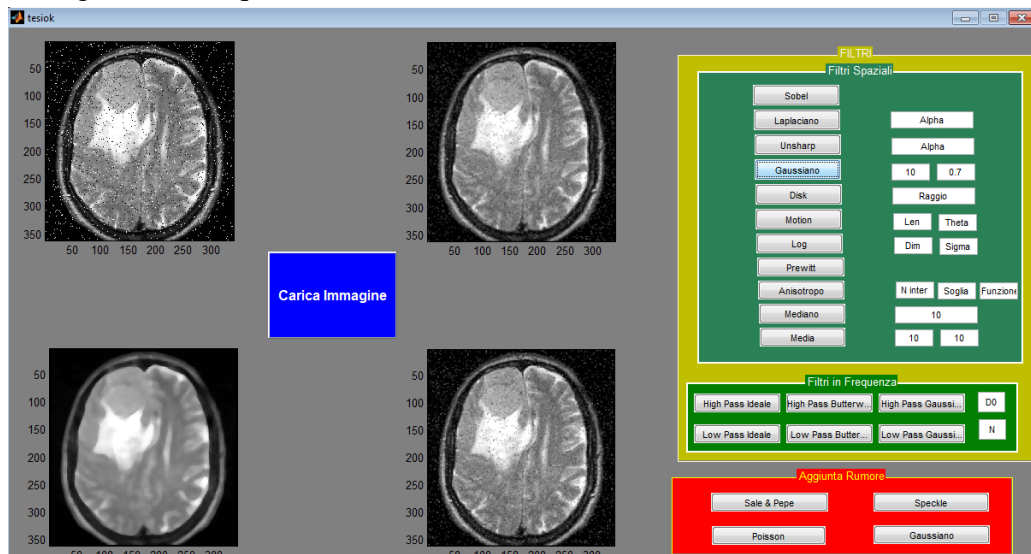


Figura 3.4



Dai risultati si può notare come l'utilizzo del filtro mediano per l'eliminazione di questa particolare tipologia di rumore sia ampiamente giustificato, l'immagine è nettamente migliore delle altre due, nelle quali, nonostante vi sia la scomparsa pressoché totale del rumore, si ha una sfocatura importante, che determina una perdita di dettagli inaccettabile.

Oltre al filtraggio per l'eliminazione del rumore è possibile effettuare anche un processo di sharpening, quindi evidenziare i dettagli dell'immagine per farli risaltare e facilitarne il riconoscimento.

Alcune tipologie di filtri di sharpening sono visibili nella figura seguente, sono stati utilizzati i filtri di sobel, unsharp e log.

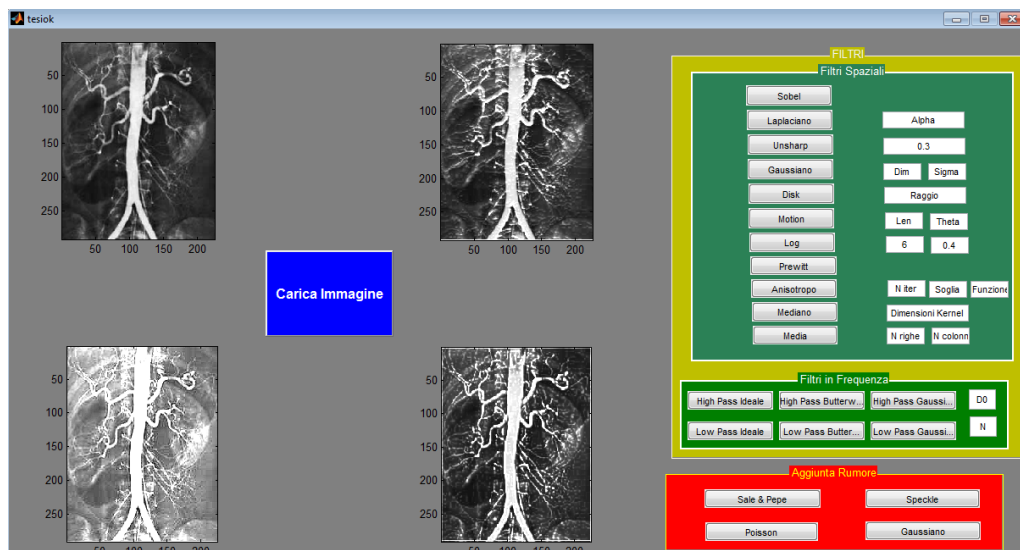


Figura 3.5

Come si può notare, tutti i filtri accentuano particolarmente i dettagli, (da notare la differenza tra l'immagine originale e quelle filtrate nella zona dei capillari più piccoli), in particolare il log restituisce un'immagine ricca di dettagli.

Oltre ai filtri spaziali è possibile eseguire anche quelli in frequenza, implementati nel riquadro più piccolo.

Si può impostare la frequenza di taglio e, per i filtri di butterworth, l'ordine desiderato.

La figura seguente mostra gli effetti del filtraggio passa alto, eliminando le basse frequenze restano solamente i contorni delle immagini, più o meno nitidi a seconda della tipologia di filtro utilizzato.

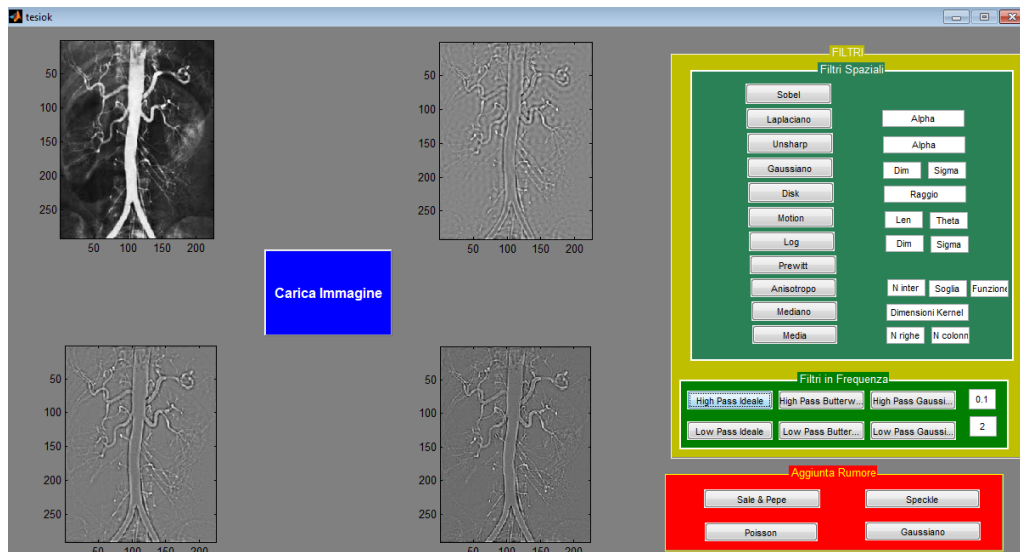


Figura 3.6

Si può vedere, applicando il filtro ideale passa alto (riquadro in alto a destra), un effetto che distorce eccessivamente l'immagine, che la rende inutilizzabile. In basso a sinistra e a destra si possono osservare gli effetti del filtro di butterworth e gaussiano rispettivamente, si può notare l'effetto più tenue del gaussiano rispetto agli altri. Invece i filtri passa basso applicano uno smoothing all'immagine, causando però una sfocatura accentuata (Figura 3.7):

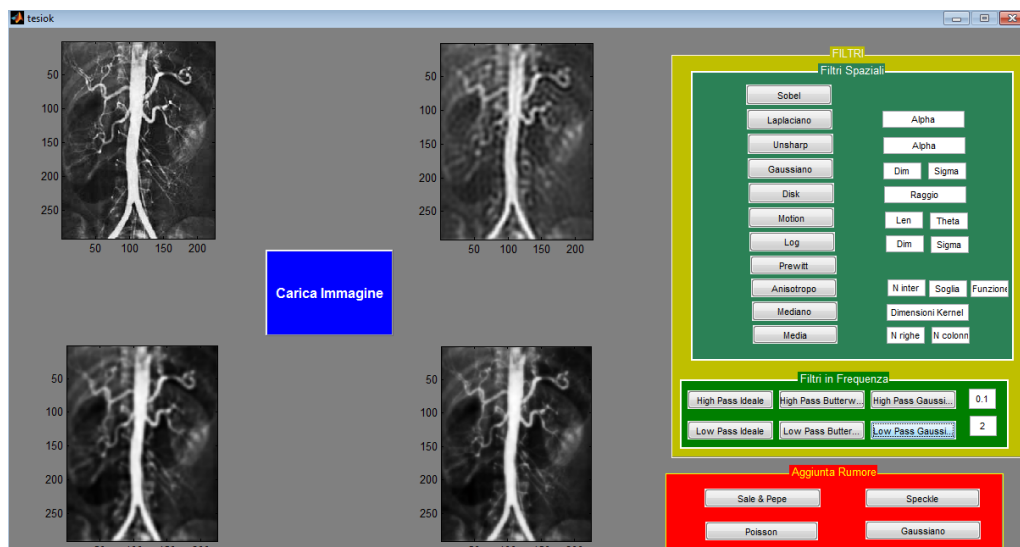


Figura 3.7

Anche in questo caso il filtro passa basso ideale applica una sfocatura molto marcata, che distorce l'immagine.

Un accento particolare va dato al filtro anisotropo, per questo filtro sono settabili diversi parametri: il numero di cicli d'interazione, maggiore è il numero più accurato sarà il risultato, il valore K, che indica il modulo del gradiente in corrispondenza del quale si avrà il massimo sfocamento, ed infine la funzione g utilizzata, 1 per la (1.17) e 2 per la (1.18).

Non sono necessari molti cicli iterativi per avere un risultato apprezzabile, sperimentalmente oltre un numero di interazioni pari a 15, si aumenta il tempo di esecuzione e senza ottenere miglioramenti apprezzabili.

Il valore di K dipende dalle varie immagini, a seconda del modulo del gradiente della zona in cui si desidera effettuare lo smoothing, di default è stato impostato  $K=5$ .

La scelta della funzione dipende dai valori di gradiente su cui si vuol fare lo smoothing, la (1.17) copre un range più ristretto, mentre la (1.18) ha un range più ampio, quindi interesserà leggermente anche i contorni dell'immagine; di default è stata impostata la (1.17), quindi il valore 1.

Di seguito propongo un immagine in cui si può vedere l'applicazione del filtro anisotropo variando i diversi parametri:

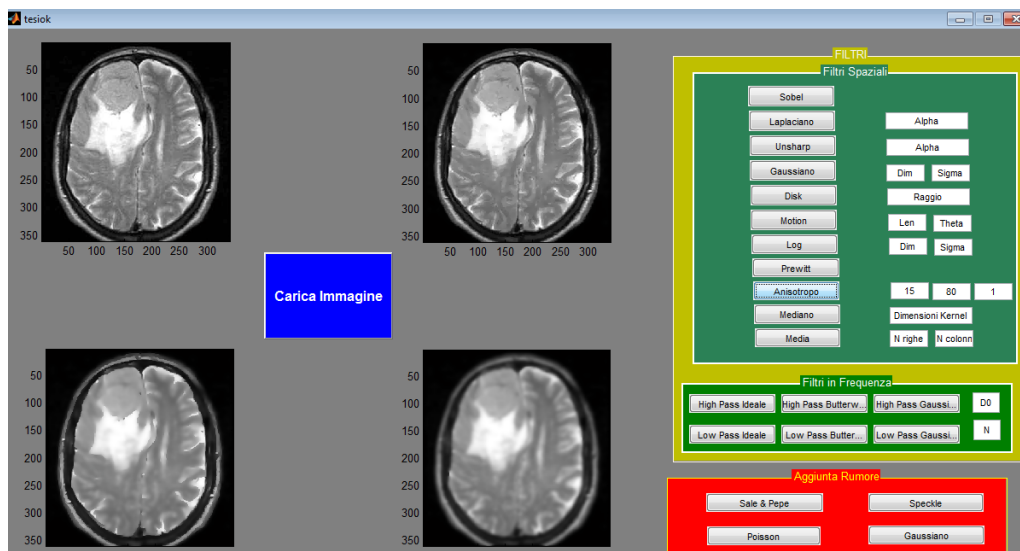


Figura 3.8

In questo caso si ha la variazione di K, in alto a sinistra vi è l'immagine originale, poi, procedendo in senso orario si ha quella con  $K=5$ ,  $K=80$ ,  $K=20$ .

Variando invece il numero d'iterazioni ed impostando  $K=20$  e funzione 1 si ottiene l'effetto mostrato in figura (3.9):

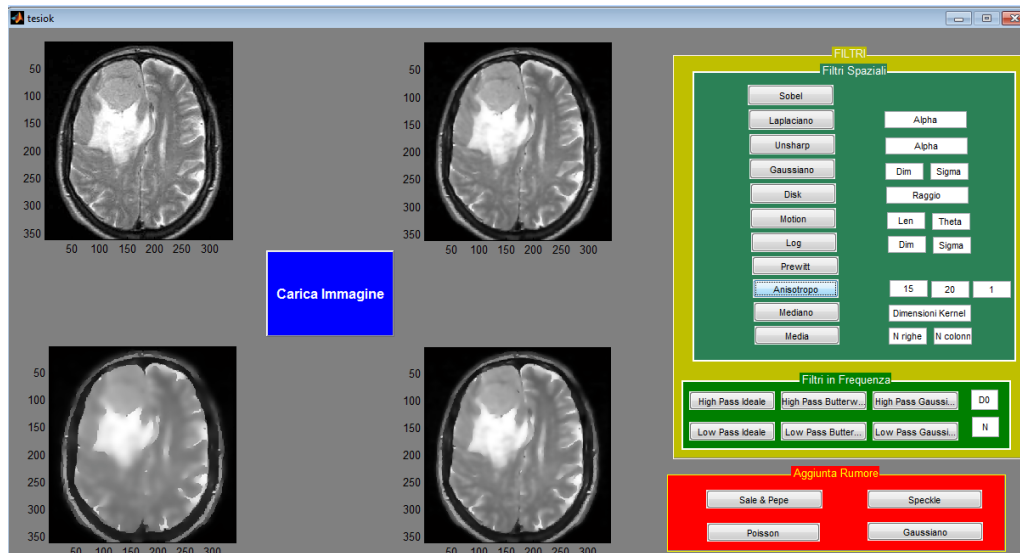


Figura 3.9

Il numero d'interazioni è, sempre procedendo in senso orario, escludendo la prima immagine,  $N=5$ ,  $N=15$ ,  $N=50$ .

L'immagine seguente invece, raffigura le variazioni utilizzando le diverse tipologie della funzione  $g$ :

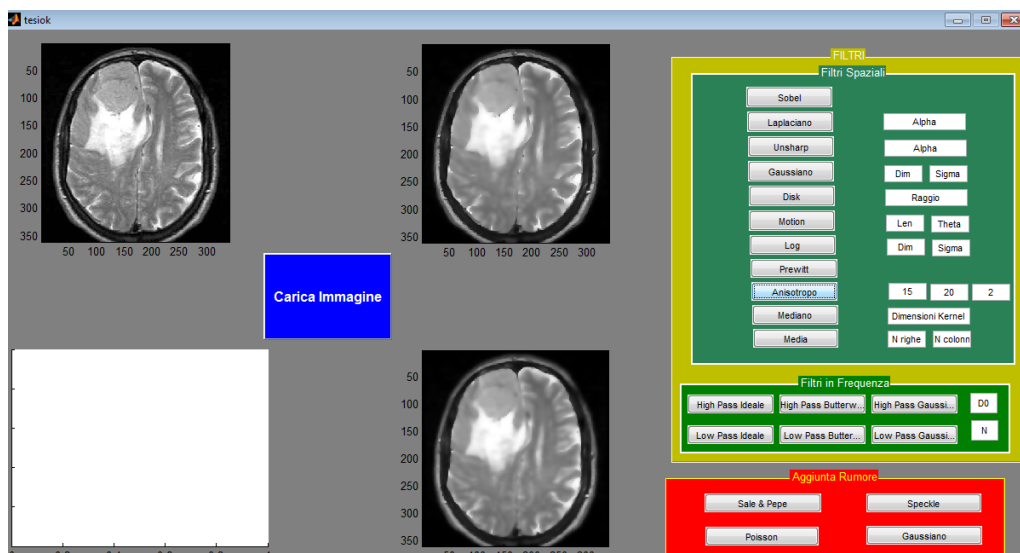


Figura 3.10

Andando in senso orario si ha il risultato del filtraggio ottenuto applicando l'equazione (1.17) poi quello con (1.18), con  $K=20$  e  $N=15$ , le differenze non sono particolarmente accentuate.

# CONCLUSIONI

---

Il mio progetto di tesi abbraccia le varie metodologie di filtraggio applicate alle immagini, a differenza di quelle esaminate nel corso di studi, applicate principalmente a segnali lineari.

Sono stati approfonditi il filtraggio nel dominio spaziale e in quello della frequenza e i loro effetti su diverse immagini.

Sono stati esaminati gli effetti dei filtri di smoothing (sia spaziali che in frequenza) nella riduzione del rumore (Lineari: Media, Gaussiano, Disk, Motion. Non Lineari: Mediano) e dei filtri di sharpening nell' evidenziare i contorni delle figure ( Sobel, Prewitt, Log, Laplaciano).

Inoltre, è stato applicato rumore digitale alle immagini per studiare la capacità dei filtri (smoothing) di eliminarlo, valutando i più efficaci.

Attraverso l'interfaccia sviluppata si è voluto rendere più facile ed intuitiva l'applicazione del filtraggio alle immagini per chi si avvicina a questo campo senza particolari conoscenze pregresse.

Realizzandola ho imparato ad elaborare le immagini e ho acquisito una conoscenza del programma decisamente superiore alle mie nozioni pregresse.

# BIBLIOGRAFIA

---

- “ Elaborazione delle Immagini Digitali “  
Rafael C. Gonzales – Richard E. Woods
- P.Perona and J. Malik, Scale-space and edge detection  
using anisotropic diffusion.
- Digital Image Processing using Matlab  
Gonzalez, Woods, Eddins  
Prentice Hall ed. 2004

# RINGRAZIAMENTI

---

“ Homo faber fortunae suae ”

- Appio Claudio Cieco

Ogni uomo è artefice del proprio destino, con tanto impegno e tenacia sono riuscito a laurearmi ed a coronare il percorso iniziato tre anni fa.

Ho iniziato questa facoltà ignaro delle difficoltà che potevano presentarsi, pensando ingenuamente che non si discostasse esageratamente dalle lauree sanitarie a cui aspiravo.

Ora, che sono al termine di questo percorso di studi, sono convinto che il fato mi abbia fatto scegliere la strada giusta per me, la migliore che potessi desiderare.

Ringrazio la professoressa Cristiana Corsi per l'aiuto che mi ha dato, la disponibilità e la pazienza per rispondere a tutti i miei interrogativi.

Voglio ringraziare la mia FAMIGLIA, che mi ha sempre sostenuto nei momenti difficili, quando le cose non andavano nel verso giusto, ha gioito insieme a me per i miei traguardi, credendo in me sempre.

Un grazie enorme va in particolare ai miei genitori, che fanno di tutto per darmi la possibilità di realizzare i miei obiettivi, come hanno sempre fatto da quando sono nato, anche se non sempre riesco a dimostrarvi il mio affetto.

Volevo ringraziare i miei amici di lungo corso, è dal liceo che ci divertiamo insieme e, anche se sono diventato uno “spezzabolgie” che non esce così spesso come vorreste, mi sopportate ancora!

Grazie a tutti, Luca Mila Bale Palla Luci Babi Sara e Sir, era indispensabile citarvi.

Facendo ingegneria ho conosciuto altre persone che hanno condiviso con me questi anni e spero anche i prossimi.

Come non citare Gigi, il mio celeberrimo compagno di studio, mi hai dato tanto in questi anni, ci siamo divertiti insieme, anche in ambito extrauniversitario, tanto che ti ho forviato a proseguire gli studi insieme, avresti mai pensato di diventare l'amico di uno con la bici dalle ruote gialle??

Sofi Romi Crucio, ho trascorso con voi gli ultimi tre anni, porterò con me i più piccoli particolari, dalle serate da Sofi, ai pranzi intorno al tavolo o sulla cattedra, alle risate che ci siamo fatti insieme; vi ringrazio, avete reso le giornate passate a lezione decisamente uniche, grazie a voi iniziavo la settimana col sorriso.

Last but not least la persona che da ormai 2 anni mi sopporta e che ho conosciuto proprio a lezione, Sara.



Mi sei sempre stata vicina in questi anni, abbiamo condiviso tutto e tu, più di ogni altra, riesci a capire i miei stati d'animo e i miei problemi.  
Ti ringrazio per avermi sempre dato coraggio nelle situazioni più difficili, so che posso contar sempre su di te.