

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

**Rimozione di artefatti visivi di tipo ringing
da immagini ottenute tramite
risonanza magnetica**

Tesi di laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.
Fabiana Zama

Correlatore:
Chiar.ma Prof.
Germana Landi

Presentata da:
Michele Bandini

**I Sessione
2021/2022**

Introduzione

L'*imaging* a risonanza magnetica è un metodo diagnostico che sfrutta le proprietà fisiche della materia immersa in un campo magnetico, per ottenere immagini dei tessuti studiati. L'acquisizione dei segnali avviene nel dominio di Fourier di dimensioni finite. Limiti fisici e tecnici degli strumenti di misura, quali il numero dei campioni acquisiti e la sensibilità dei sensori utilizzati, contribuiscono a deteriorare la qualità delle immagini ricostruite. Questo porta ad immagini imperfette, e una di queste imperfezioni sono gli artefatti di Gibbs.

Su questi ultimi si concentra il lavoro di questa tesi, il cui obiettivo è analizzare l'algoritmo proposto da E. Kellner et. al. (da *Magnetic resonance in medicine*, volume 76, numero 5 del 2016, pagine 1574-1581), per rimuovere questi artefatti senza introdurre alterazioni di sfocamento. Infatti, uno dei problemi che si incontrano nell'applicare tecniche standard per eliminare l'effetto Gibbs è quello di introdurre sfocamento dei dettagli, abbassando così la qualità globale dell'immagine.

Una parte importante del lavoro di questa tesi è stata la realizzazione di uno strumento software che consentisse di utilizzare l'algoritmo all'interno di script Matlab, mediante trasformazione del codice C++ fornito dagli autori. Quindi, si è provata l'efficacia del software mediante un'ampia sperimentazione su dati sintetici e su acquisizioni reali. I risultati ottenuti sono coerenti con gli obiettivi dell'algoritmo: rimuovere gli artefatti di Gibbs quando sono presenti, lasciando praticamente immutate le immagini dove sono assenti, riducendo al minimo l'effetto di sfocamento.

Questo testo è così strutturato: il primo capitolo spiega come funziona l'elaborazione di segnali (e in particolare di immagini) nel dominio di Fourier, dando un background teorico e mostrando anche a livello pratico con alcuni esempi grafici; il secondo capitolo parla della risonanza magnetica, della fisica che le sta dietro e di come entrino in gioco i concetti introdotti nel capitolo precedente; il terzo entra nel vivo dell'algoritmo descrivendone il funzionamento teorico; il quarto, infine, mostra degli esempi dell'algoritmo in azione, sia in immagini create *ad hoc*, ma anche in vere scansioni di risonanza magnetica.

Indice

1	Elaborazione di immagini nel dominio di Fourier	7
1.1	Trasformata di Fourier 1D	7
1.1.1	Trasformata di un segnale rettangolare	8
1.1.2	Trasformata di Fourier discreta 1D	9
1.2	Trasformata di Fourier 2D	10
1.2.1	Trasformata di Fourier discreta 2D	10
1.2.2	Trasformata di un rettangolo	11
1.3	Trasformata di Fourier in ambito numerico	11
1.3.1	Zero-Padding	11
1.3.2	Convoluzione con finestra	11
1.3.3	Artefatti di Gibbs	13
2	L'imaging a risonanza magnetica	17
2.1	Funzionamento della risonanza magnetica	17
2.1.1	Principi fisici della risonanza magnetica	17
2.1.2	Componenti di un sistema di risonanza magnetica	18
2.2	Acquisizioni di immagini e k-spazio	18
2.2.1	K-spazio	19
2.2.2	Trasformata di Fourier inversa sul k-spazio	19
2.3	Sfide nella ricostruzione dei segnali	19
2.3.1	Effetto Gibbs su immagini	20
3	L' algoritmo	21
3.1	Tecnica 1D	21
3.2	Tecnica 2D	22
4	L' algoritmo in azione	25
4.1	Algoritmo su immagini artificiali	25
4.1.1	Linee parallele	25
4.1.2	<i>Phantom</i> di Shepp-Logan	27
4.2	Algoritmo su immagini vere	28

4.2.1	Assiale	30
4.2.2	Sagittale	30
4.2.3	Commento sui risultati	31
A	Come utilizzare il software su <i>Windows</i>	35
	Bibliografia	37

Capitolo 1

Elaborazione di immagini nel dominio di Fourier

Nell'elaborazione dei segnali, uno dei processi più importanti è quello di estrarre da essi le varie frequenze che li compongono e studiarle separatamente. Questo è possibile grazie alla **Trasformata di Fourier**, che permette di passare dal dominio temporale o spaziale (quello usuale) a quello delle frequenze (anche detto *dominio di Fourier*), effettuare delle operazioni: nel caso continuo il tutto può essere fatto senza perdere informazioni, e si cercherà di minimizzare la perdita nel caso discreto.

Dal punto di vista teorico, avere informazioni su un segnale nel dominio delle frequenze è equivalente ad averle nel dominio spaziale, ma all'atto pratico in molte applicazioni si possiede o l'una o l'altra forma; questo vale anche per le immagini, che altro non sono che segnali 2D. Tuttavia, in genere, difficilmente gli strumenti di acquisizione sono perfetti, e quando si acquisisce un'immagine alcune informazioni sulle frequenze più alte (che si riferiscono ai dettagli che presentano un improvviso cambio d'intensità) vengono parzialmente perse. Per capire cosa provoca queste perdite di informazioni e come compensare ad esse, è necessario prima comprendere bene il funzionamento della Trasformata di Fourier, in particolare quella discreta.

1.1 Trasformata di Fourier 1D

Data una funzione $f(t)$, la sua trasformata di Fourier $\mathcal{F}\{f(t)\}$ è definita dall'equazione

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-i2\pi\mu t} dt \quad (1.1)$$

dove i è la costante immaginaria, e μ è una variabile continua. Poiché la t è la variabile di integrazione, la trasformata di f si può scrivere come F , funzione solo di μ , ossia

$$F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi\mu t} dt \quad (1.2)$$

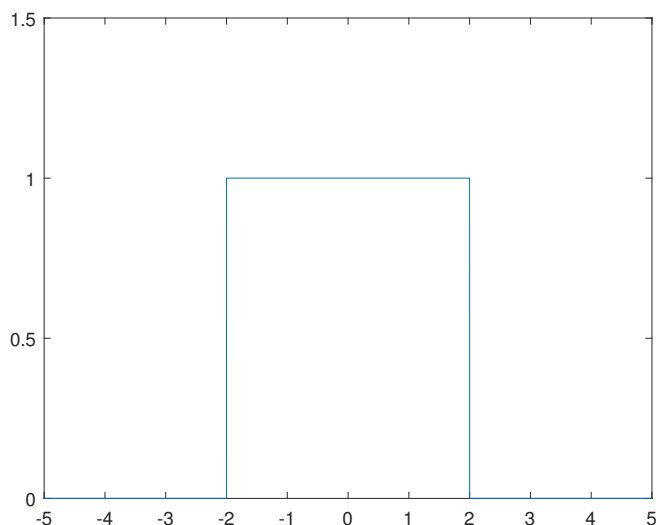


Figura 1.1: Grafico della funzione rettangolare con $T = 4$ e $A = 1$

Vediamo ora un esempio, che sarà utile anche più avanti.

1.1.1 Trasformata di un segnale rettangolare

Prendiamo una funzione rettangolare, ossia che assume valore 0 ovunque tranne che per un intervallo centrato in 0:

$$f(t) = \begin{cases} A & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{altrimenti} \end{cases} \quad (1.3)$$

Il grafico 1.1 mostra il caso con $A = 1$ e $T = 4$. Si mostra facilmente che la sua trasformata di Fourier è la funzione

$$F(\mu) = AT \frac{\sin(\pi\mu T)}{\pi\mu T} \quad (1.4)$$

che è rappresentata, per gli stessi valori di A e T di prima, dal grafico 1.2. Quella mostrata in 1.4 è la funzione *sinc*. Più formalmente:

$$\text{sinc}(m) = \frac{\sin(\pi m)}{\pi m} \quad (1.5)$$

Questa è una funzione molto importante, che tornerà anche in seguito.

In genere comunque la trasformata di Fourier è una funzione complessa. Si considera

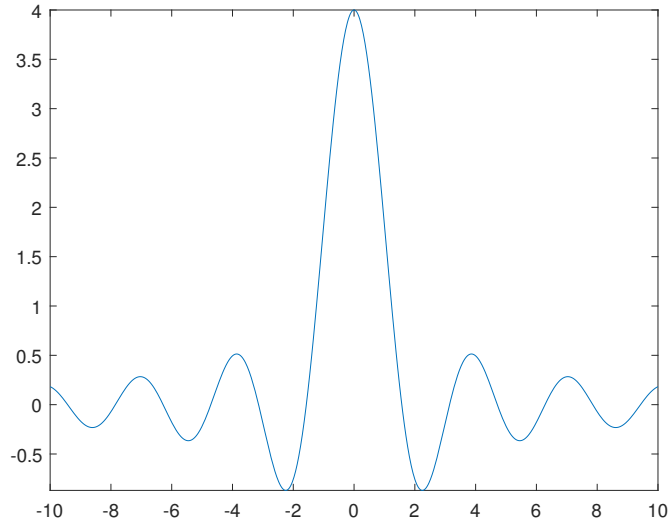


Figura 1.2: Grafico della trasformata di Fourier della funzione rappresentata in 1.1

perciò di solito lo *spettro di Fourier*, che altro non è che il modulo della trasformata. Quindi la 1.4 diventa:

$$|F(\mu)| = AT \frac{|\sin(\pi\mu T)|}{\pi\mu T} \quad (1.6)$$

che si può vedere, per gli stessi parametri di prima, nella figura 1.3.

1.1.2 Trasformata di Fourier discreta 1D

Un segnale viene di solito rappresentato come funzione a valori e argomenti discreti, poiché così viene interpretato digitalmente. Perciò consideriamo una funzione $f(x)$, campionata in M punti equidistanti. Senza perdita di generalità, scegliamo $x = 0, 1, \dots, M-1$. La trasformata di Fourier discreta (*Discrete Fourier Transform*, **DFT**) ha la forma:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-iu \frac{2\pi}{M} x} \quad u = 0, \dots, M-1 \quad (1.7)$$

Esiste anche la forma inversa, la *IDFT* (*Inverse Discrete Fourier Transform*) che assume la forma:

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{iu \frac{2\pi}{M} x} \quad x = 0, \dots, M-1 \quad (1.8)$$

dove u è la variabile di frequenza.

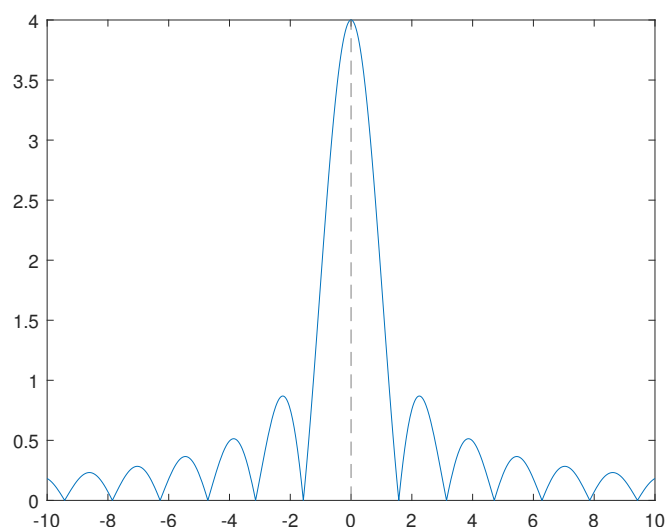


Figura 1.3: Grafico dello spettro di Fourier della funzione rappresentata da 1.1

In questo elaborato verrà usata soprattutto la trasformata nella sua forma discreta, poiché è più adatta alla forma che i segnali hanno quando rappresentati al calcolatore elettronico.

1.2 Trasformata di Fourier 2D

In questo elaborato i segnali da studiare saranno soprattutto immagini, ossia segnali 2D. Per questo motivo è necessario estendere quanto detto per i segnali 1D nel caso a due dimensioni.

1.2.1 Trasformata di Fourier discreta 2D

Data una funzione $f(x, y)$ che rappresenta un'immagine digitale di dimensione $M \times N$, la sua *DFT* è:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (1.9)$$

dove, come similmente al caso 1D, u e v sono variabili discrete nel dominio delle frequenze e vanno da 0 a $M - 1$ e da 0 a $N - 1$ rispettivamente.

Data $F(u, v)$, si può ottenere $f(x, y)$ usando la *IDFT* 2D:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (1.10)$$

1.2.2 Trasformata di un rettangolo

Similmente al caso 1D, vediamo la trasformata di un rettangolo. In questo caso però si tratta di una immagine di un rettangolo, che si può vedere in 1.4a. Certi software, come *Matlab*, mettono verso il bordo i valori più alti dello spettro di Fourier, come nella 1.4b, ma in genere si effettua lo *shift* e si preferisce lavorare con la figura come in 1.4c. Per visualizzare meglio, si prende spesso il logaritmo dei valori, e si ottiene così una figura come in 1.4d.

1.3 Trasformata di Fourier in ambito numerico

Il software di analisi numerica *Matlab* offre delle funzioni molto veloci per effettuare la *DFT*, usando l'algoritmo della *Fast Fourier Transform* (*FFT*). La *FFT* è, come dice il nome, molto più veloce che calcolare la trasformata normalmente, passando da un tempo $O(N^2)$ ad uno $O(N \log N)$, dove N è la lunghezza del segnale da elaborare; è inoltre più stabile numericamente. La sua inversa è detta *IFFT*, *Inverse Fast Fourier Transform*. Vediamo alcune delle tecniche e dei problemi che si incontrano in ambito numerico.

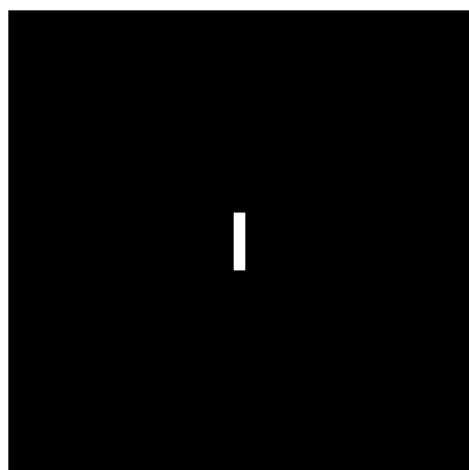
1.3.1 Zero-Padding

Data un'immagine, può essere necessario ottenerne un'altra a risoluzione maggiore, a partire dalla prima. Per farlo si può trasformare la stessa nello spazio di Fourier, ingrandirla, e poi ritrasformarla nello spazio originale. Facendolo, si ottiene un'immagine più grande, ma essendo costruita a partire dalle stesse informazioni che si avevano in quella più piccola, mostrerà delle imperfezioni.

Di solito per farlo si usa lo *zero-padding*: si ingrandisce l'immagine nel dominio di Fourier mettendo degli zeri dove non si hanno informazioni, ottenendo quindi una figura che è nera fuori dal quadrato che rappresenta la figura di partenza. Riprendendo l'esempio del rettangolo nella figura 1.4a, nella 1.5a si vede lo spettro di Fourier della 1.4b con il *padding*, e la 1.5b è l'immagine ricostruita: ha una risoluzione maggiore, ma di fatto la qualità è peggiorata.

1.3.2 Convoluzione con finestra

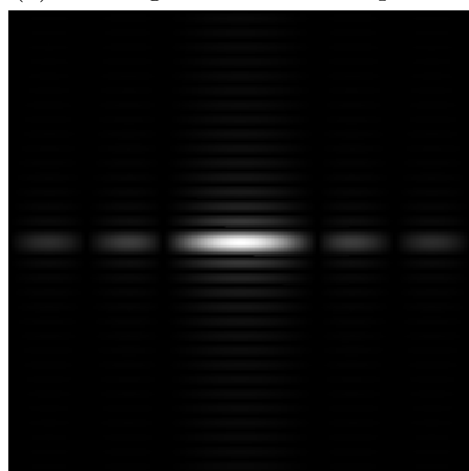
Quando si ottengono segnali nelle applicazioni, soprattutto quando si parte dal dominio di Fourier e si cerca di passare a quello spaziale, non si ricevono tutte le informazioni:



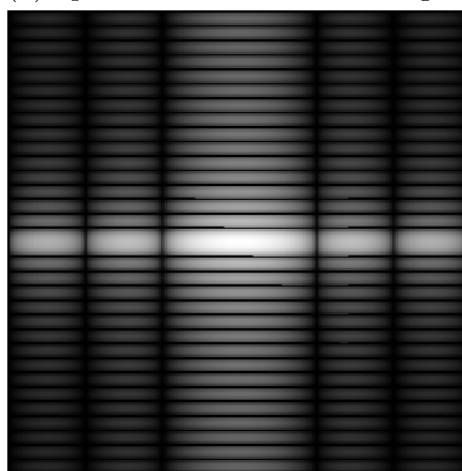
(a) Rettangolo nel dominio spaziale



(b) Spettro di Fourier del rettangolo

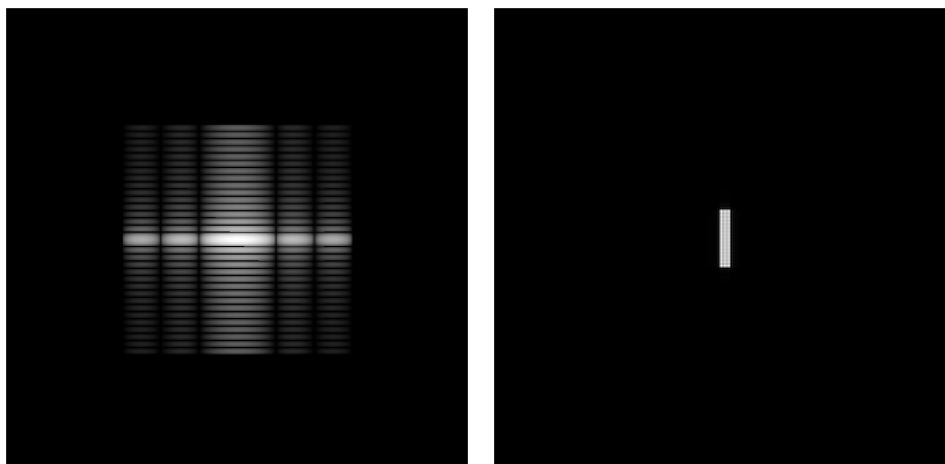


(c) Spettro di Fourier *shiftato* del rettangolo



(d) Logaritmo dello spettro di Fourier *shiftato*

Figura 1.4: Immagine di un rettangolo e la sua trasformata

(a) Spettro di Fourier con *padding*

(b) Rettangolo ricostruito

Figura 1.5: Spettro di Fourier logaritmico del rettangolo come in figura 1.4d ma ingrandito con *zero-padding*, e il rettangolo ottenuto applicando la trasformazione inversa di Fourier

in particolare quelle più lontane dal centro (in una configurazione in cui le frequenze più basse sono al centro) non si riescono ad ottenere. Per simulare l'effetto si può effettuare una convoluzione con una finestra più piccola dell'immagine.

Nella figura 1.6 si vede l'operazione in azione: partendo dallo spettro di Fourier visto nella 1.4c, restringendosi a un'immagine più piccola attorno al centro, si ottiene un effetto simile a quello del *padding*. L'immagine 1.6b è infatti ricostruita a partire dall'immagine del dominio di Fourier che però è stata troncata, e si sono quindi perse informazioni: esteticamente è molto simile alla 1.5b, anche se la risoluzione dell'immagine teoricamente è la metà.

1.3.3 Artefatti di Gibbs

Quando si lavora con immagini digitali ci si ritrova sempre a dover affrontare problemi che derivano dalla natura limitata degli strumenti di acquisizione. In particolare, quando le immagini sono ottenute in seguito a una *IDFT* fatta a partire da uno spettro di Fourier incompleto, si possono ottenere i cosiddetti artefatti di Gibbs, anche detti di ringing perché sono come anelli attorno alle parti dell'immagine che variano drasticamente in intensità. Per vedere bene come si presenta in 2D basta andare a 2.3.1, ma nella 1.7c si vedono molto bene le “onde” che oscillano a partire dalla discontinuità.

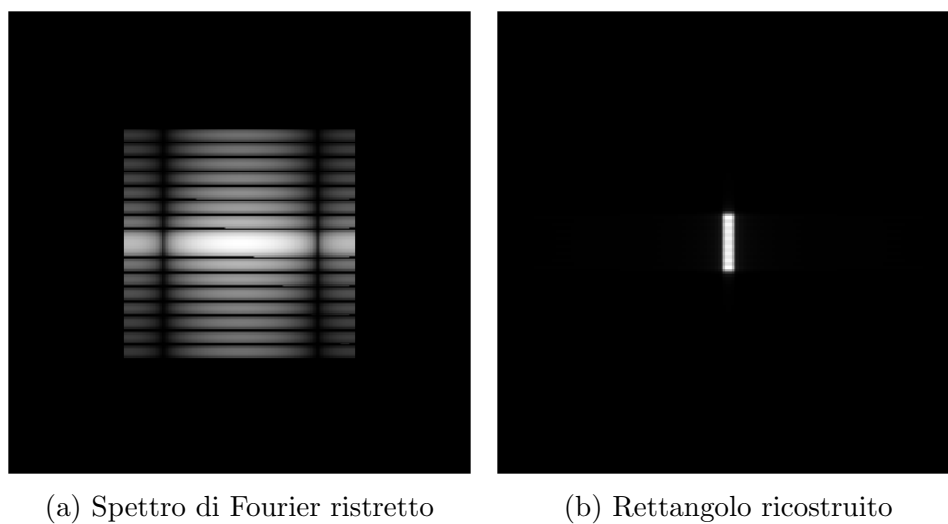
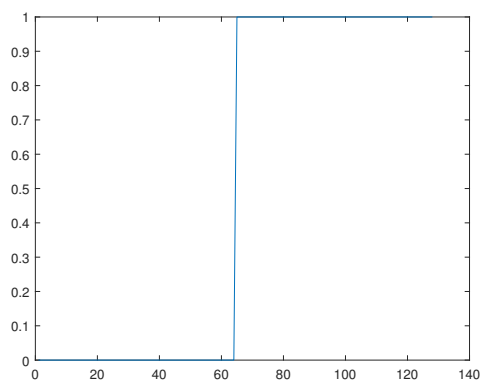
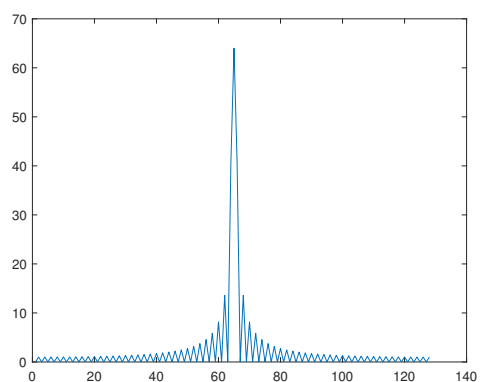


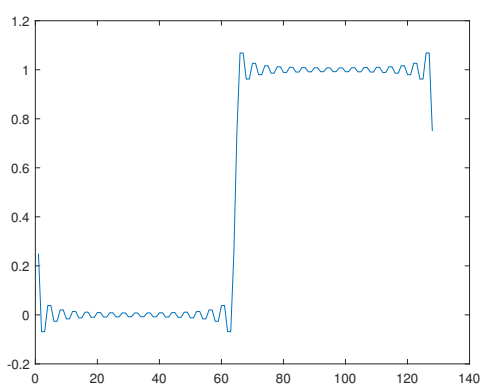
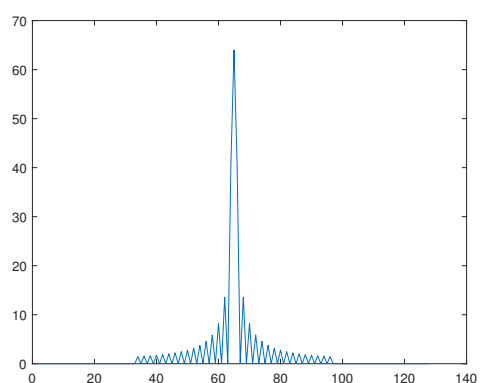
Figura 1.6: Spettro di Fourier logaritmico, convoluto con una finestra attorno al centro, e rettangolo ottenuto tramite trasformata inversa di Fourier



(a) Segnale 1D con una discontinuità



(b) Spettro di Fourier del segnale

(c) Segnale ricostruito che presenta *ringing*

(d) Spettro di Fourier troncato del segnale

Figura 1.7: Un segnale 1D che presenta un brusco salto, ricostruito poi da una *DFT* troncata

Capitolo 2

L'imaging a risonanza magnetica

L'imaging a risonanza magnetica, in breve *MRI*, è una tecnica usata in radiologia per generare immagini a scopo medico. Sfrutta la proprietà di alcuni nuclei atomici di assorbire l'energia di alcune frequenze radio quando immersi in un campo magnetico; è una tecnica diagnostica molto precisa e anche molto sicura, poiché non utilizza radiazioni ionizzanti, dannose per il paziente.

L'equazione dell'*MRI* può essere espressa come un'entità a due dimensioni della forma

$$S(k_x, k_y) = f[I(x, y)] \quad (2.1)$$

dove S è il segnale misurato, I l'immagine reale e f lo schema di codifica delle informazioni spaziali. Se f è invertibile, si può ottenere I dalla trasformata inversa:

$$I(x, y) = f^{-1}S(k_x, k_y) \quad (2.2)$$

Il segnale S dipende da vari parametri, come i tempi di rilassamento $T1$ e $T2$, o la densità protonica ρ .

2.1 Funzionamento della risonanza magnetica

Per capire meglio come vengono catturate le immagini nell'*MRI*, è utile prima capire com'è costruita e come funziona una macchina per la risonanza magnetica.

2.1.1 Principi fisici della risonanza magnetica

Una delle proprietà fisiche della materia sfruttate dalla risonanza magnetica è lo *spin*, che è presente in tutte le particelle. Esso può essere negativo o positivo, e se molteplici particelle con *spin* opposti si accoppiano possono eliminarne ogni manifestazione osservabile. Nella risonanza magnetica, sono gli *spin* nucleari disaccoppiati che sono importanti.

Quando gli *spin* sono posti in un campo magnetico B , mostrano una risonanza che ha una frequenza ben definita, la *frequenza di Larmor* ν , che ha equazione

$$\nu = \gamma B \quad (2.3)$$

dove γ è il rapporto giromagnetico della particella. Quando una particella assorbe un fotone può spostarsi da uno stato di bassa energia a uno ad alta energia, dove l'energia assorbita dev'essere uguale alla differenza tra i due stati ed è data dall'equazione

$$E = h\nu \quad (2.4)$$

dove h è la costante di Planck, che equivale a $6,626\,070\,15 \times 10^{-34}$ J · s.

Per attivare il magnetismo macroscopico in un oggetto, lo si espone a un forte campo magnetico esterno B_0 , per creare un momento magnetico M_0 . Inizialmente, la magnetizzazione totale e il campo esterno puntano nella direzione z , che è quella della magnetizzazione longitudinale M_z . La componente trasversale M_{xy} è zero. Se si fornisce energia al sistema a sufficienza, si può far giacere M_0 sul piano x-y con l'applicazione di un campo B_1 . Questo crea una componente trasversale M_{xy} ma, dopo aver spento il campo B_1 , il sistema cede energia all'ambiente, anche detto *reticolo*, e la costante che governa questo processo si chiama infatti $T1$ o tempo di rilassamento *spin-reticolo*. La costante $T2$, similmente, descrive il ritorno all'equilibrio della magnetizzazione trasversale M_{xy} , ed è detto tempo di rilassamento *spin-spin*.

2.1.2 Componenti di un sistema di risonanza magnetica

Dal punto di vista della strumentazione che compone uno scanner per *MRI*, i componenti *hardware* principali sono pochi. Uno è il magnete permanente, che genera il già citato campo B_0 . Sono poi presenti molte bobine, di cui le principali sono quelle dette *bobine di gradiente*, che modificano l'intensità del campo magnetico lungo i tre assi spaziali.

Altri componenti fondamentali sono il sistema a radio-frequenza (anche detto RF), che genera il campo rotante B_1 , e il computer che si occupa di convertire e ricostruire l'immagine sulle basi delle informazioni raccolte dagli altri componenti. È sull'elaborazione di queste immagini che ci concentreremo principalmente.

2.2 Acquisizioni di immagini e k-spazio

L'immagine viene acquisita in tre passi principali: la scelta della fetta assiale tramite l'uso del gradiente G_z ; la codifica di diverse fasi in righe con il gradiente G_y ; infine, la creazione delle colonne con diverse frequenze usando il gradiente G_x .

2.2.1 K-spazio

Il segnale viene campionato diverse volte, riempiendo una fetta del volume alla volta. Lo spazio dove viene campionato il segnale discreto è detto *k-spazio*, e altro non è che il dominio di Fourier descritto nel capitolo 1.

Il *k-spazio* raccoglie i segnali a bassa frequenza al centro dei dati acquisiti e le frequenze alte lontane dal centro. I segnali a bassa frequenza forniscono informazioni sul contrasto, sulle caratteristiche dell'immagine più ampie. Le alte frequenze mostrano i cambiamenti più rapidi e le caratteristiche più sottili.

2.2.2 Trasformata di Fourier inversa sul *k-spazio*

Nell'*MRI*, la ricostruzione usando la *IFFT* si fa in due passaggi. Prima si effettua la *IFFT* 1D sulle righe, poi sulle colonne. Ovviamente si potrebbe invertire quest'ordine, ma comunque la possibilità di dividere il processo in due passaggi è dato dalla linearità e dalla separabilità della *DFT*.

Nella pratica, ci sono alcune attività che vanno fatte prima e dopo l'applicazione della *IFFT*, e che in totale si possono riassumere in questi sei passaggi:

1. Leggere l'*header* dei dati: contiene informazioni sui dati dell'*MRI*, come la grandezza dell'immagine, l'*offset*, ecc.
2. Leggere le informazioni nel *k-spazio*.
3. Effettuare la *IFFT* nella direzione K_x .
4. Effettuare la *IFFT* nella direzione K_y .
5. *Shiftare* l'immagine ottenuta per avere le caratteristiche nella giusta posizione.
6. Mostrare l'immagine.

2.3 Sfide nella ricostruzione dei segnali

I problemi nel ricostruire i segnali ottenuti con l'*MRI* sono già in parte stati descritti al capitolo 1. Essi derivano dal troncamento prematuro dell'informazione nel dominio di Fourier, di cui il nostro *k-spazio* è solo una finestra. Questo provoca un calo nella risoluzione spaziale, come già visto in 1.3.2, che a volte può rendere certi dettagli incomprensibili; può provocare artefatti visivi; può infine far comparire l'effetto di Gibbs attorno ai bordi, di cui si era già parlato in 1.3.3.

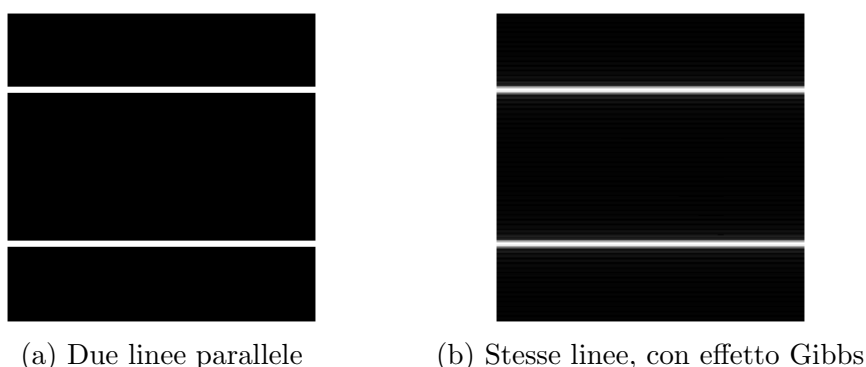


Figura 2.1: Un'immagine con due linee, e una ottenuta sottocampionando la prima nel dominio di Fourier

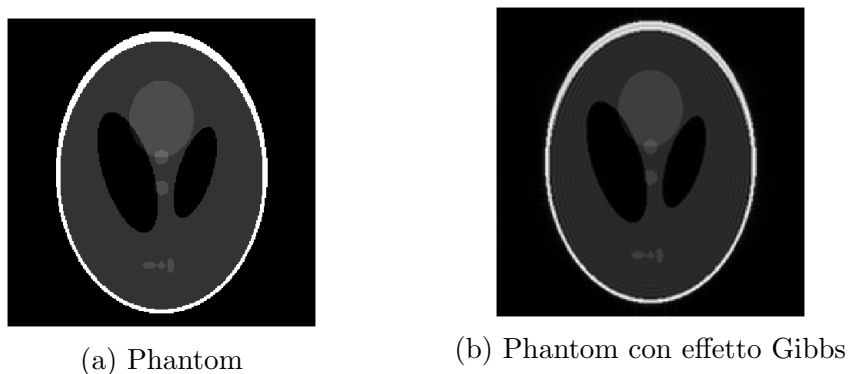


Figura 2.2: Il Phantom di Shepp-Logan, visto normalmente e ottenuto sottocampionando nel dominio di Fourier

2.3.1 Effetto Gibbs su immagini

L'effetto di Gibbs, che nella figura 1.7 era stato mostrato per i segnali 1D, compare anche nelle figure 2D ed è un ostacolo per la visione dei risultati dell'*MRI*. L'immagine 2.1 mostra abbastanza bene l'effetto: l'immagine a sinistra sono due linee parallele, quella a destra è ottenuta eliminando nel dominio di Fourier le frequenze più alte, e si vedono così delle piccole linee parallele vicine alle due principali, che formano l'effetto *ringing*.

Ma, per avvicinarsi di più alle immagini di *MRI*, vediamo la figura 2.2, che è il *Phantom* di Shepp-Logan: immagine molto usata per la risonanza magnetica poiché, con regole matematiche, permette di creare quell'immagine a qualsiasi dimensione e, nella forma, ricorda vagamente un cervello visto dall'alto. Si vede subito che, vicino alle differenze più grandi di colore, si creano delle "increspature", che formano l'effetto di Gibbs. Questo è il tipo di *ringing* che possiamo aspettarci dalle immagini ottenute con la risonanza magnetica.

Capitolo 3

L'algoritmo

L'algoritmo oggetto di questo capitolo è stato descritto in [4], per cui anche la spiegazione in questo capitolo sarà simile a quella nel *paper* in questione.

Si tratta di una tecnica che rimuove l'effetto Gibbs senza però andare ad intaccare altre informazioni dell'immagine, ad esempio sfocandola o diminuendone la risoluzione. Per questo verrà chiamato semplicemente *algoritmo di unringing*.

3.1 Tecnica 1D

Per comprendere il funzionamento dell'algoritmo su immagini, è necessario prima capire come esso agisce sui segnali uno-dimensionali. L'algoritmo 2D, infatti, si basa sul funzionamento di quello 1D, con alcuni dovuti accorgimenti.

Proviamo perciò a prendere un segnale 1D, dove $I_0(x)$ è quello che misuriamo, e $c_0(k)$ i suoi coefficienti nella *DFT*. In pratica abbiamo:

$$I_0(x) = \frac{1}{N} \sum_{k=0}^{N-1} c_0(k) \cdot e^{-\frac{2\pi i}{N} kx} \quad (3.1)$$

Da questo, creiamo $2M$ immagini $I_s(x)$, con $s = -M \dots M - 1$:

$$I_s(x) = \frac{1}{N} \sum_{k=0}^{N-1} c_0(k) \cdot e^{-\frac{2\pi i}{N} k(x + \frac{s}{2M})} \quad (3.2)$$

Il segnale I_s è una versione spostata di quello originale, dove l'intero s indica uno spostamento di $\frac{s}{2M}$ pixel. Per ogni pixel x , va cercato lo spostamento ottimale che minimizzi le oscillazioni: si può utilizzare qualsiasi metodo che misuri le variazioni; in questo algoritmo, in particolare, si usa la variazione totale (*TV*) nell'intorno del punto, ossia si calcola semplicemente la differenza in valore assoluto tra i valori dei pixel vicini.

La definizione di intorno è particolarmente importante: si supponga di avere un pixel x su un bordo. In quel caso, conviene escludere x per fare in modo che, misurando le oscillazioni, non si includa l'effettivo salto che è normale che sia nel bordo, ma solo le oscillazioni date dal *ringing*.

Un altro accorgimento è quello di separare i due lati $x' < x$ e $x' > x$, utile nei casi in cui ci sono due bordi vicini: nel caso che essi abbiano due spostamenti ottimali, si vuole trovare il migliore tra i due, non un compromesso.

Per formalizzare queste idee, l'oscillazione è misurata nel lato sinistro (+) e in quello destro (-):

$$D_s^\pm(x) = \sum_{n=k_1}^{k_2} |I_s(x \pm n) - I_s(x \pm (n-1))| \quad (3.3)$$

dove $K = [k_1, k_2]$ definisce l'intorno dove le oscillazioni vengono misurate. La finestra K è l'unico parametro di questo metodo. I risultati indicano che l'algoritmo è relativamente robusto rispetto a questa scelta. Ora, per entrambi i lati, + e -, vengono trovati indipendentemente gli spostamenti ottimali che minimizzano la misura di oscillazione D :

$$t^\pm(x) = \arg \min_{-M \leq s < M} D_s^\pm(x) \quad (3.4)$$

e da qui, si può determinare il minimo tra $D_{t^+(x)}^+(x)$ e $D_{t^-(x)}^-(x)$: si ha quindi lo spostamento ottimale $t(x)$.

Ora, si conosce lo spostamento ottimale che per ogni pixel minimizza le oscillazioni e, tendenzialmente, anche gli artefatti di *ringing*. Tuttavia, poiché gli spostamenti potrebbero introdurre distorsioni nelle immagini, è necessario tornare indietro alla griglia originale, cioè valutando $I_{t(x)}(x')$ nella posizione non intera $x' = x - t(x)/(2M)$. Formalmente:

$$I_{\text{unring}}(x) := I_{t(x)}(x - t(x)/(2M)) \quad (3.5)$$

Per farlo, si può usare qualsiasi schema di interpolazione. In questo caso, si è usata una semplice interpolazione lineare. In effetti, si potrebbero usare schemi di livello più alto, ad esempio polinomiali, ma potrebbero introdurre nuove oscillazioni. Il caso estremo di usare la funzione *sinc* per interpolare porterebbe all'immagine originale.

3.2 Tecnica 2D

Per estendere l'algoritmo al caso 2D, parrebbe sensato cercare lo spostamento 2D ottimale (s_x, s_y) . Tuttavia, un bordo diagonale con orientamento arbitrario produce un motivo di *ringing* a scacchiera, che in generale non verrebbe rimosso da uno spostamento simultaneo in 2D. Inoltre l'interpolazione in 2D introdurrebbe una sfocatura che quella in 1D evita.

Si usa quindi un approccio diverso: si introducono due funzioni a sella da applicare nello spazio di Fourier del tipo

$$\begin{aligned} G_x &= \frac{1 + \cos k_y}{(1 + \cos k_y) + (1 + \cos k_x)} \\ G_y &= \frac{1 + \cos k_x}{(1 + \cos k_y) + (1 + \cos k_x)} \end{aligned} \quad (3.6)$$

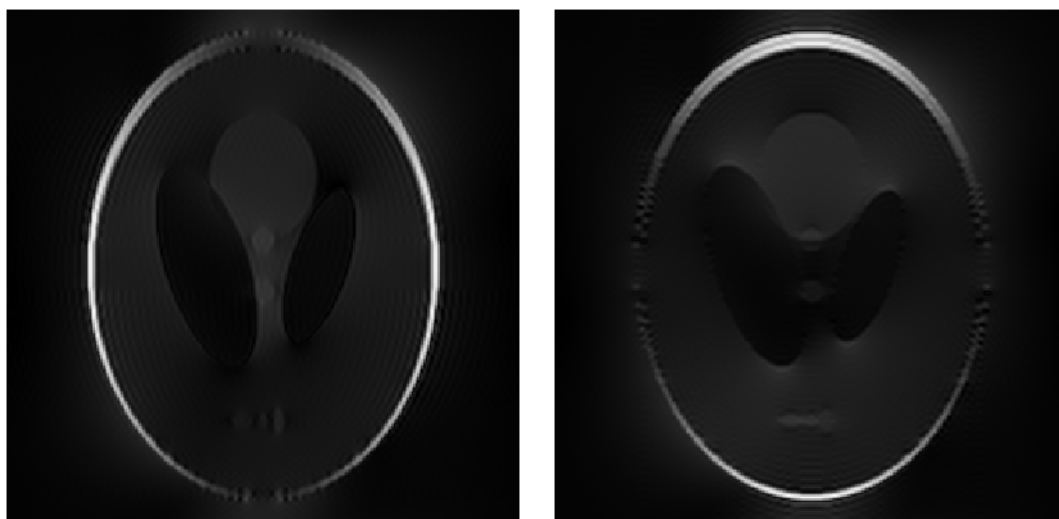
che filtrano l'immagine, aumentando l'effetto Gibbs in una direzione ma diminuendolo nell'altra perpendicolare.

Una caratteristica fondamentale di questi filtri è che soddisfano la normalizzazione $G_x + G_y = 1$. Si applicano questi filtri all'immagine originale per creare le due immagini modificate:

$$\begin{aligned} I_x &= \mathcal{F}^{-1}\{\mathcal{F}\{I\} \cdot G_x\} \\ I_y &= \mathcal{F}^{-1}\{\mathcal{F}\{I\} \cdot G_y\} \end{aligned} \quad (3.7)$$

dove $\mathcal{F}\{\cdot\}$ indica la trasformata di Fourier. Nella figura 3.1 si nota l'effetto sull'immagine di partenza 2.2b: le immagini, oltre ad avere un forte effetto *ringing* ognuna in una direzione, sono anche più scure perché, sommate, devono restituire l'immagine originale. Si applica quindi la tecnica di rimozione del ringing 1D spiegata in 3.1: nella direzione x per I_x , e in quella y per I_y , per produrre le immagini J_x e J_y .

Infine, queste due sono sommate per ottenere l'immagine finale $J = (J_x + J_y)$. Si noti che l'azione dei filtri (G_x, G_y) non va considerata come una normale tecnica di filtraggio, ma va pensata come una componente nella procedura intera. Grazie alla normalizzazione $G_x + G_y = 1$, immagini senza artefatti di *ringing*, con $J_x = J_y$, rimarrebbero immutate. Questo assicura che lo *smoothing* introdotto sia minimo.



(a) Phantom con *ringing* nella direzione x (b) Phantom con *ringing* nella direzione y

Figura 3.1: Il Phantom di Shepp-Logan, con *ringing* su una sola direzione

Capitolo 4

L'algoritmo in azione

Per vedere l'effetto dell'algoritmo sono stati effettuati diversi esperimenti, sia su immagini artificiali sia su immagini derivanti da vere risonanze magnetiche. In questo capitolo sono raccolti alcuni risultati.

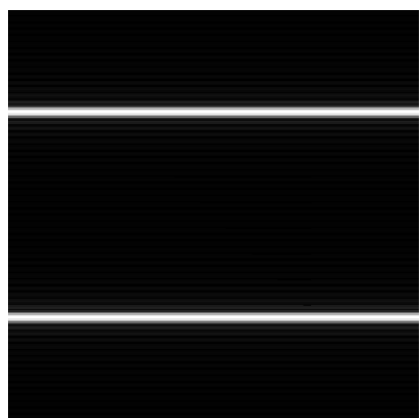
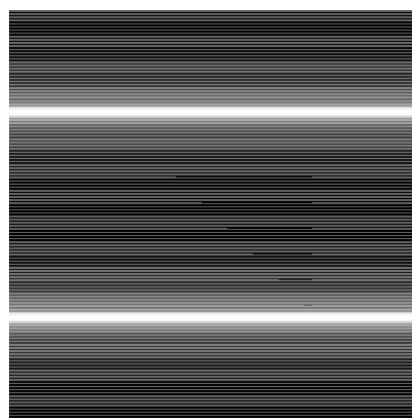
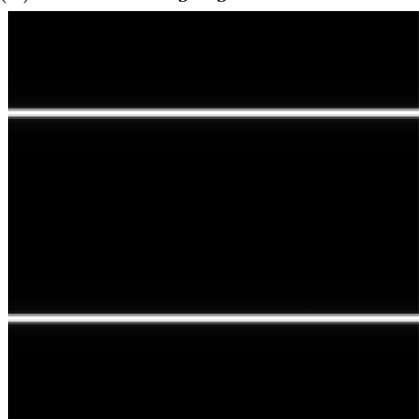
4.1 Algoritmo su immagini artificiali

Prima di tutto ha senso testare l'algoritmo su immagini create al computer, su cui abbiamo controllo e delle quali conosciamo sia la *ground truth*, ovvero quale aspetto dovrebbero avere senza artefatti, e l'entità dell'effetto di *ringing*, così da poter valutare meglio l'effetto dell'algoritmo.

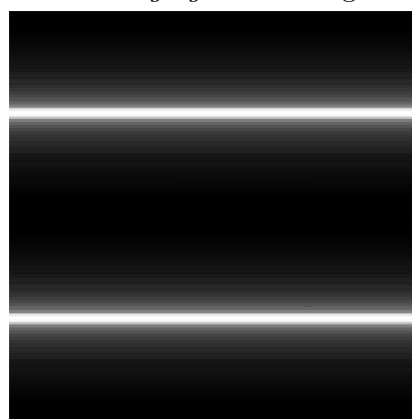
4.1.1 Linee parallele

Proviamo innanzitutto a vedere come agisce l'algoritmo su un'immagine molto semplice, nello specifico le linee già viste nella figura 2.1. Per aumentare la visibilità dell'effetto *ringing*, si può guardare l'immagine 4.1b, che è la stessa ma con l'intensità del bianco in scala logaritmica. Nella 4.1c si vede il risultato dopo l'algoritmo: se non è chiaro, c'è anche la 4.1d in scala logaritmica, dove la differenza è palese. Un altro modo per notare la differenza è guardarla direttamente in 4.1e.

Tutto questo va interpretato in un'ottica dove solo la 4.1a è data, magari ottenuta da una misurazione. L'obiettivo è quello di avvicinarsi il più possibile alla *ground truth*, cioè l'immagine che sappiamo di dover ottenere, che è la 4.1f. Ovviamente nei casi reali non si sa qual è il risultato vero da ottenere, e in generale è più difficile da capire perché le immagini comunemente ottenute (ad esempio nell'*MRI*) hanno molte più caratteristiche.

(a) Effetto *ringing* in scala lineare(b) Effetto *ringing* in scala logaritmica

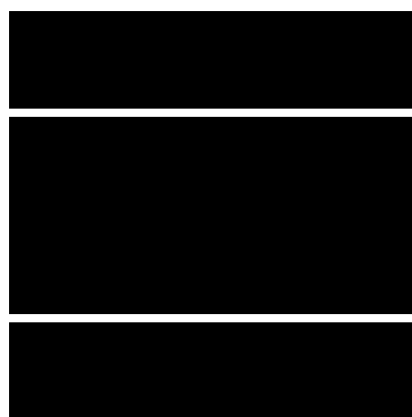
(c) Risultato dell'algoritmo, in scala lineare



(d) Risultato dell'algoritmo in scala logaritmica



(e) Differenza tra prima e dopo l'algoritmo in scala lineare

(f) *Ground truth* in scala lineare, cioè l'immagine originaleFigura 4.1: Linee parallele, prima e dopo l'algoritmo di *unringing*

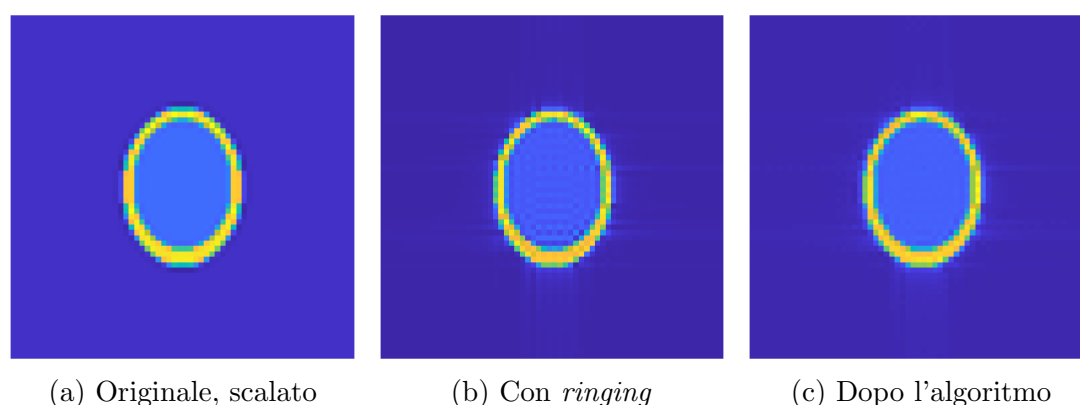


Figura 4.2: Il Phantom di Shepp-Logan, sezione 20 di 128

4.1.2 *Phantom* di Shepp-Logan

Per avere un esempio più adeguato al problema in questione, cioè a quello delle immagini da risonanza magnetica, è opportuno studiare qualcosa che gli assomigli di più, come il già visto (nella figura 2.2) *Phantom* di Shepp-Logan. Ne esiste anche una versione 3D, che simula (nelle varie sezioni) le differenti immagini che si ottengono facendo l'*MRI* di qualcosa di complesso come un cervello: si può trovare la funzione che lo genera in [5].

Analizziamo quindi alcune delle 128 sezioni del *Phantom* 3D, in ordine come sarebbero se rappresentassero una scansione trasversale (cioè dall'alto verso il basso) di un cervello umano. Per farlo le abbiamo rappresentate a colori, per mostrare meglio il contrasto tra le aree di diverse intensità, che rischia di perdersi con semplici immagini in bianco e nero. Per ognuna delle sezioni in esame mostreremo: l'immagine originale; l'immagine scalata nel dominio spaziale; l'immagine scalata convolvendo con una finestra nel dominio di Fourier (si veda 1.3.2); e quest'ultima, dopo che è stata elaborata con l'algoritmo.

Sezione 20

Partiamo dalla ventesima sezione (figura 4.2), poiché quelle precedenti sono o completamente vuote o mancano dell'area all'interno dell'ellisse. Nella figura 4.2b si nota già un grande effetto ringing all'interno dell'ellisse; la 4.2c lo rimuove in buona parte, tanto che l'immagine risultante è molto simile alla 4.2a, ossia è molto simile all'originale scalato, e la differenza principale sono gli aloni che si vedono al di fuori dell'ellisse, che non pregiudicano la comprensione delle caratteristiche dell'immagine.

Sezione 48

La quarantottesima sezione ha due ellissi all'interno di quella grande: una scura e una chiara. L'effetto è quello che ci si aspetterebbe, con una particolarità: in 4.3b si nota

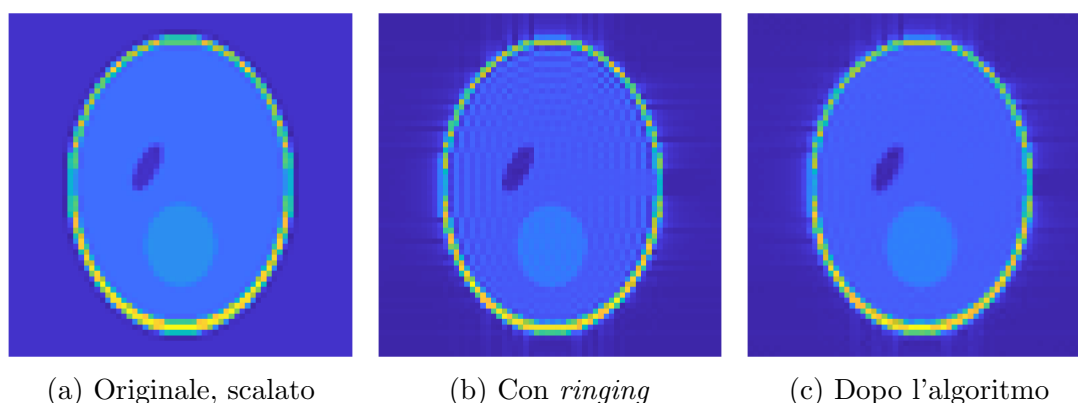


Figura 4.3: Il Phantom di Shepp-Logan, sezione 48 di 128

come non ci sia quasi ringing “a destra” dell’ellisse scura, probabilmente a causa dei vari effetti di Gibbs dati dalle varie differenze di colore che si sommano, annullandosi.

Sezione 64

La sessantaquattresima sezione, essendo in mezzo, è estremamente completa, mostrando tutte le caratteristiche a dimensioni massime. Ancora una volta, si nota come la 4.4d sia più simile alla 4.4b che alla 4.4c, se non per gli aloni attorno. Si noti come, scalando, il fatto che i piccoli dettagli in alto della 4.4a siano tre viene perso: l’algoritmo non può aiutare a recuperare questo tipo di informazioni perse.

Sezione 80

Procedendo oltre la metà le immagini sono più o meno simili, eccetto per due piccoli cerchietti che si formano in mezzo al *Phantom*, che sono visibili nell’ottantesima sezione. Tuttavia, i risultati sono comunque come ce li si aspetterebbe: in 4.5b si nota che il *ringing* non influenza e non è influenzato particolarmente da questi piccoli dettagli.

4.2 Algoritmo su immagini vere

Dopo aver analizzato gli effetti dell’algoritmo su immagini costruite interamente al computer, dove avevamo il controllo sulla quantità del *ringing* e sapevamo quali erano i risultati da ottenere, ha senso provarlo anche da immagini prese dalla realtà.

I volumi presi in esame provengono da una vera *MRI*, e sono risultati di vere scansioni di cervelli umani. Faremo come in 4.1.2, e mostreremo alcune fette significative. Va tuttavia notato che le immagini in questione non presentano apprezzabili effetti di *ringing*, per cui i risultati mostrati potranno sembrare deludenti.

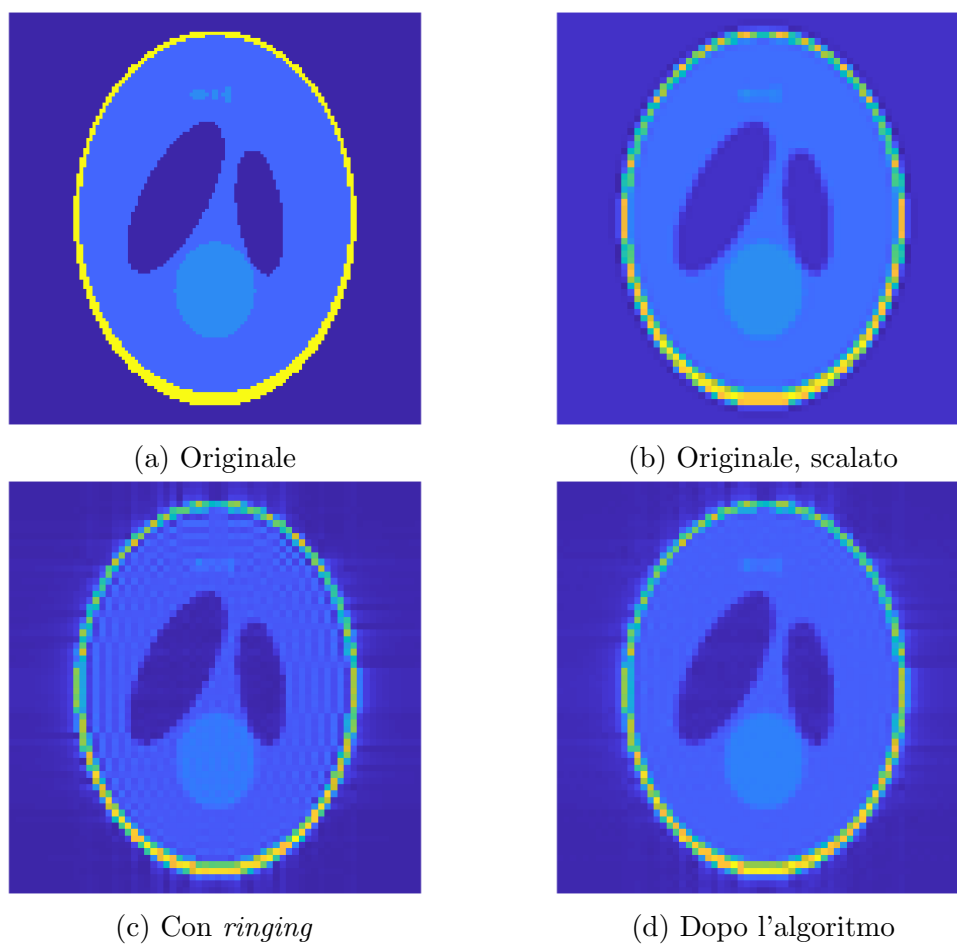


Figura 4.4: Il Phantom di Shepp-Logan, sezione 64 di 128

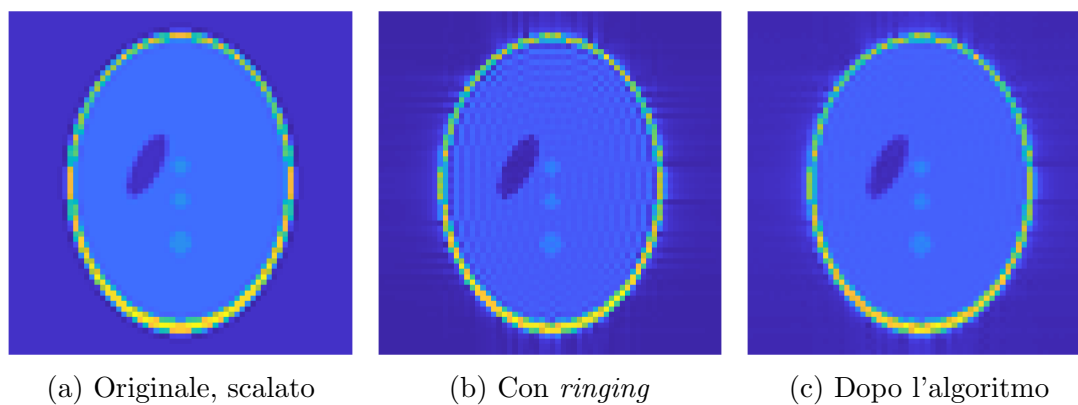
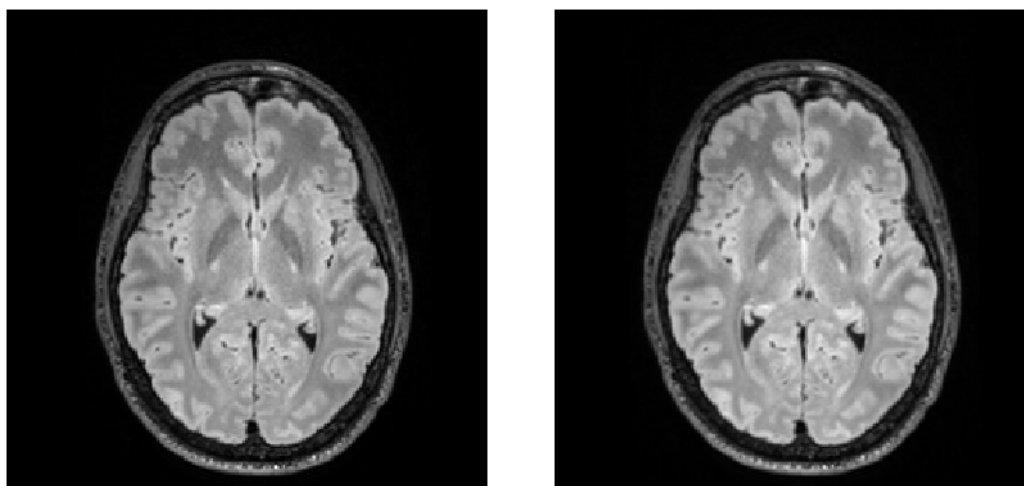


Figura 4.5: Il Phantom di Shepp-Logan, sezione 80 di 128



(a) Immagine originale

(b) Immagine dopo l'algoritmo

Figura 4.6: Fetta 99 della scansione assiale, prima e dopo l'algoritmo

Da un certo punto di vista, però, questo dimostra nuovamente che l'algoritmo non va a intaccare immagini o dettagli all'interno di essi che non siano artefatti di Gibbs. Prenderemo quindi i due volumi in questione, applicheremo l'algoritmo, e per entrambi guarderemo le fette dove l'algoritmo ha impattato di più.

Le scansioni vengono fatte intersecando il corpo umano con diversi piani paralleli tra loro, e poi unendo le intersezioni in un volume tridimensionale. Quella *assiale* utilizza piani perpendicolari all'altezza della persona, mentre quella *sagittale* usa piani ortogonali all'apertura delle braccia.

4.2.1 Assiale

Come detto in precedenza, i risultati sono poco apprezzabili: prendiamo ad esempio solo la fetta 99, quella che numericamente mostra più differenze, che nella figura 4.6 è messa a confronto prima e dopo l'algoritmo: non si nota nessuna differenza visivamente.

Numericamente il pixel dove si riscontra maggiore differenza è all'elemento (229, 135) di questa fetta. Vediamo il confronto nella figura 4.7: nelle due figure, il pixel in questione è stato circondato da un cerchio rosso per individuarlo meglio, ma comunque cambia solo di una sfumatura del grigio.

4.2.2 Sagittale

Discorso simile può essere fatto con la scansione sagittale: la fetta dove la differenza è più marcata è la 88, nella figura 4.8, e nella figura 4.9 si nota lo zoom attorno al pixel

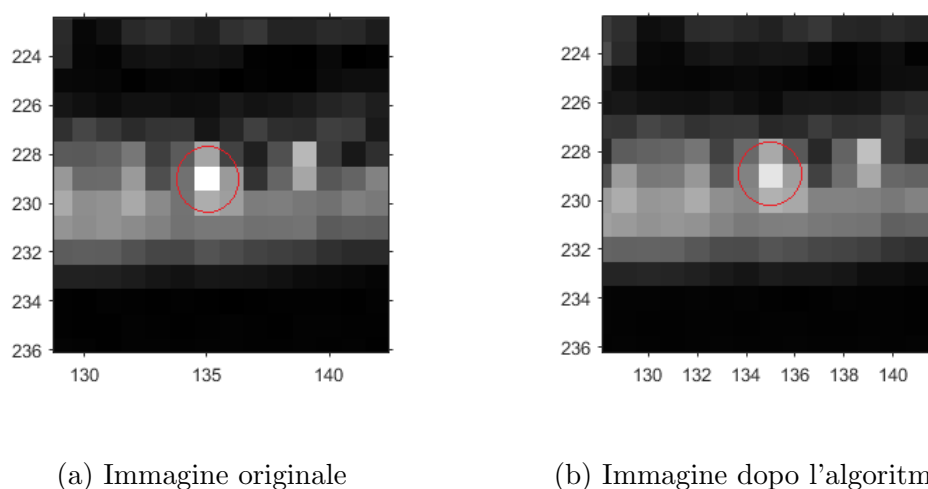


Figura 4.7: Dettaglio dell'immagine assiale dove è presente il massimo cambiamento prima e dopo l'algoritmo, nel pixel al centro del cerchio rosso

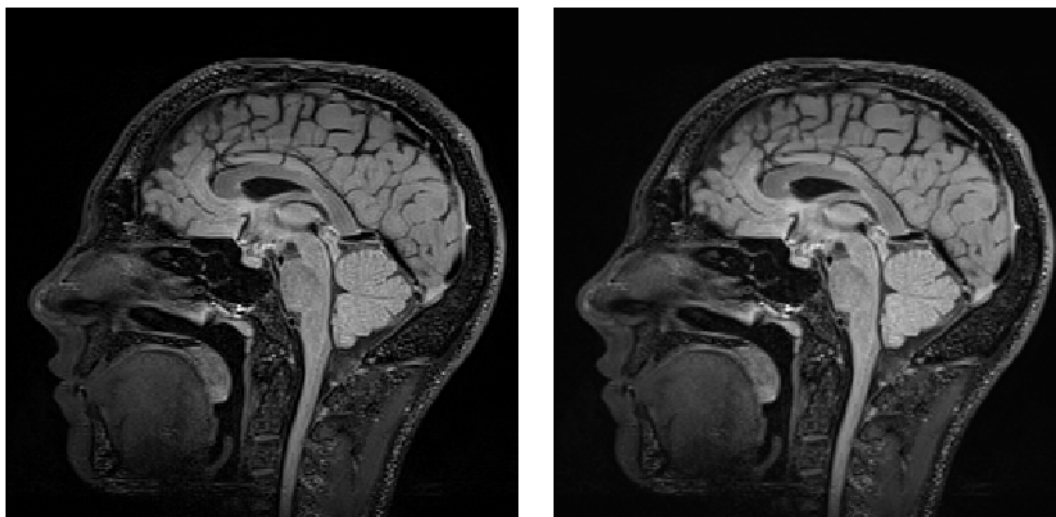
(153,117). In questo caso, però, anche i due pixel accanto sono leggermente schiariti, anche se globalmente questo lascia comunque l'immagine praticamente immutata.

4.2.3 Commento sui risultati

Ovviamente, come già anticipato, le immagini utilizzate hanno dato un risultato abbastanza deludente. Non essendo presenti particolari effetti di *ringing*, hanno soddisfatto solo allo scopo di mostrare che l'algoritmo **non affligge immagini che sono già corrette**. Questo è molto importante, e va unito ad altre considerazioni.

Gli esperimenti con immagini artificiali, in particolare con il *phantom* (4.1.2) mostrano che, in caso di forte presenza di *ringing*, l'algoritmo funziona molto bene. Unito alla consapevolezza che, in sua assenza, l'algoritmo non fa quasi nulla, si può concludere che sia un ottimo compromesso tra efficacia e stabilità.

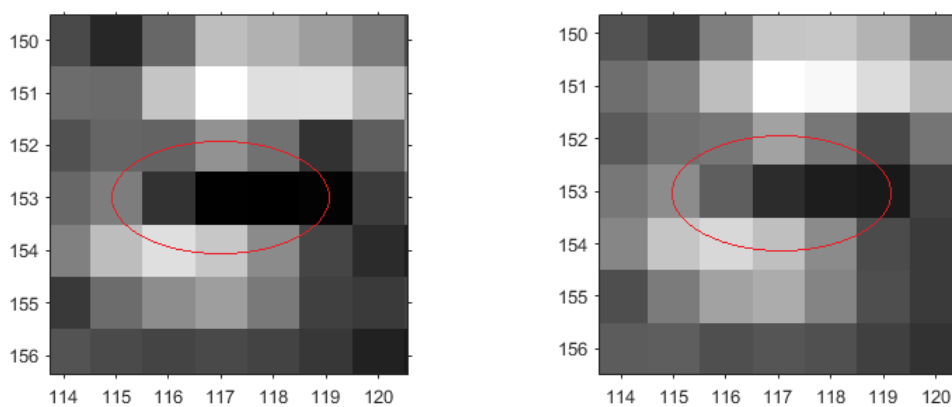
Successivi studi potrebbero testare l'algoritmo su immagini di *MRI* ottenute con l'aiuto di mezzi di contrasto, ossia liquidi che rendono più visibili certi tessuti in fase di scansione. Avendo un maggiore contrasto, per l'appunto, nelle immagini, aumenterebbe probabilmente l'effetto di *ringing*, e potrebbe dare risultati più apprezzabili.



(a) Immagine originale

(b) Immagine dopo l'algoritmo

Figura 4.8: Fetta 88 della scansione sagittale, prima e dopo l'algoritmo



(a) Immagine originale

(b) Immagine dopo l'algoritmo

Figura 4.9: Dettaglio dell'immagine sagittale dove è presente il massimo cambiamento prima e dopo l'algoritmo, nei tre pixel dentro l'ellisse rossa e specialmente nel pixel al centro

Conclusioni

Scopo di questa tesi era di studiare l'algoritmo descritto in [4], sia dal punto di vista teorico sia dal punto di vista pratico, cioè mostrando come rimuovesse gli artefatti di ringing sia da immagini sintetiche che ottenute da misure reali.

I risultati sono positivi: nelle immagini sintetiche con molti artefatti di Gibbs al loro interno, esso si è mostrato estremamente efficace, ripulendo l'immagine dal ringing e lasciandola per il resto pressoché immutata. Per le immagini reali a disposizione, in cui l'effetto Gibbs era debolmente presente, l'algoritmo non ha introdotto alterazioni agendo solo in presenza del debole ringing presente.

Da qui si capisce quindi la caratteristica dell'algoritmo: esso è estremamente specializzato nel rimuovere solo un tipo di artefatto, lasciando invariati tutti gli altri limiti che possono diminuire la qualità di un'immagine. In casi in cui si lavori con immagini di qualità molto bassa bisogna quindi usare anche altre tecniche in combinazione a questa.

In ogni caso questo algoritmo rimane per lo meno un buon punto di partenza, e potrebbe diventare imprescindibile in tutti quei campi in cui ci si aspetta un buon quantitativo di ringing, magari anche estendendosi ad applicazioni differenti dalla risonanza magnetica, per la quale è stato concepito.

Appendice A

Come utilizzare il software su *Windows*

L'algoritmo è fornito in [4] sotto forma di software c++ da lanciare con Matlab, ma prima di poter utilizzarlo deve essere compilato nella macchina specifica. Per farlo, è necessario seguire alcuni passaggi.

Prima di tutto bisogna modificare le righe 93 e 94 del file *ringRm.cpp*, che sfruttano un comportamento che spesso non viene accettato dai compilatori c++, sostituendole con:

```
double* TV1arr = new double[2*nsh+1];  
double* TV2arr = new double[2*nsh+1];
```

È poi necessario avere la libreria *FFTW*, che si trova in <https://www.fftw.org/>. Dopodiché, la spiegazione procede supponendo di usare come compilatore *Microsoft visual c++ 2022*; con altre versioni e/o altri compilatori il procedimento può cambiare.

Installare *Visual studio 2022* che esiste anche in versione gratuita, e aprire *Powershell per sviluppatori*. Andare nella cartella dove si è estratta *FFTW* e trasformare le librerie così:

```
lib /machine:x64 /def:libfftw3f-3.def  
lib /machine:x64 /def:libfftw3-3.def  
lib /machine:x64 /def:libfftw3l-3.def
```

ottenendo dei file *.lib*. Dopodiché, prendere da questa cartella tre file: *libfftw3-3.lib*, *fftw3.h* e *libfftw3-3.dll*, mettendoli nella stessa cartella di *ringRm.cpp*. Aprire poi Matlab in questa cartella ed eseguire il comando `mex ringRm_fix.cpp -lfftw3-3 -compatibleArrayDims`, che creerà il file *ringRm.mexw64*, che sarà il nostro file compilato.

Una volta fatto, si può utilizzare la funzione Matlab *unring* contenuta in *unring.m*, a patto di tenere *unring.m*, *ringRm.mexw64* e *libfftw3-3.dll* nella stessa cartella.

Bibliografia

- [1] Rafael C. Gonzalez, Richard E. Woods *Digital Image Processing*, 2018, *Pearson*
- [2] <https://www.thefouriertransform.com/>
- [3] Abiodun M. Aibinu, Momoh J. E. Salami, Amir A. Shafie, Athaur Rahman Najeeb *MRI Reconstruction Using Discrete Fourier Transform: A tutorial* 2008, *World Academy of Science, Engineering and Technology*.
- [4] Elias Kellner, Bibek Dhital, Valerij G. Kiselev, Marco Reiser *Gibbs-Ringing Artifact Removal Based on Local Subvoxel-Shifts* 2016, *Magnetic resonance in medicine*, 76.5: 1574-1581.
- [5] Matthias Schabel (2022). 3D Shepp-Logan phantom (<https://www.mathworks.com/matlabcentral/fileexchange/9416-3d-shepp-logan-phantom>), MATLAB Central File Exchange. Retrieved June 10, 2022.

Ringraziamenti

Desidero ringraziare innanzitutto la Prof.ssa Zama, per avermi appassionato alla materia e per avermi aiutato sia a trovare l'argomento della tesi, sia a scriverla correttamente. Ringrazio i miei familiari e i miei amici, che mi hanno sempre dato la forza per andare avanti anche nei momenti bui.

Ringrazio i miei compagni di corso, soprattutto Flavia, perché senza di loro non avrei superato alcuni esami.

Ringrazio infine Greta perché mi è stata accanto sempre, mi ha aiutato anche a costo del suo tempo prezioso (più del mio), e perché ha saputo credere in me quando io non ne ero in grado.