

**POLITECNICO DI MILANO
SCUOLA DI INGEGNERIA INDUSTRIALE E
DELL'INFORMAZIONE**

Corso di Laurea Magistrale in Ingegneria Matematica



**Restauro di immagini tomografiche
mediche tramite l'uso di metodi di image
inpainting**

Relatore: Prof.ssa Simona Perotto

Correlatore: Dott.ssa Elena Faggiano

Tesi di Laurea Magistrale di:
Michele Haile, Matricola 754547

Anno Accademico 2012/2013

Sommario

Le protesi metalliche come ad esempio le protesi dentarie, le protesi d'anca e i device cardiaci inducono artefatti nelle immagini di tomografia computerizzata. Per questo motivo, negli ultimi decenni sono stati proposti diversi metodi per la riduzione degli artefatti da metallo. Tra i vari metodi proposti, i metodi basati sull'interpolazione, si basano sull'identificazione sul sinogramma della zona corrotta e sulla sua sostituzione tramite l'utilizzo di informazioni provenienti da proiezioni vicine non corrotte, ad esempio tramite interpolazione lineare o tramite tecniche di inpainting. In questa tesi proponiamo di utilizzare l'algoritmo di image inpainting proposto da Bertalmio et al. nel 2000 basato sulle equazioni di Naver-Stokes. Proponiamo quindi un'implementazione del metodo ad elementi finiti con streamline diffusion. Abbiamo quindi testato il metodo sia su dati sintetici che su dati clinici. Si vede che tale implementazione presenta risultati soddisfacenti anche in confronto con altri metodo presenti in letteratura e con contenuti costi computazionali.

Indice

Introduzione	7
1 Tomografia	9
1.1 La trasformata di Radon	9
1.1.1 Definizione e proprietà	9
1.1.2 Radon e Fourier: Teorema della sezione centrale . . .	11
1.1.3 Inversione della trasformata di Radon	12
1.1.4 Esempio	14
1.1.5 Positura del problema inverso	20
1.2 Imaging medico	21
1.3 Tomografia Assiale Computerizzata	22
1.4 Metal Artifact Reduction	24
2 Immagini digitali	29
2.1 Concetti introduttivi	29
2.2 Immagini: segnali 2D	31
2.3 Filtri	34
2.4 Estrazione contorni	38
2.5 Istogrammi e modelli di rumore	41
2.6 Analisi e sintesi	43
3 Restauro di immagini digitali	47
3.1 Introduzione	47
3.2 Metodi basati su EDP	48
3.2.1 Il modello di TV inpainting di Chan e Shen	52
3.2.2 Euler's elastica	53
3.2.3 Confronto tra metodo TV ed metodo dell'Elastica . .	56
3.2.4 Metodo di Bertalmio et al.	57

3.2.5	Una teoria unificante	61
3.3	Sintesi di texture	65
3.3.1	Efros, sintesi non parametrica	66
3.3.2	Criminsi	68
3.3.3	Bugeau, Bertalmò, Caselles e Sapiro	69
3.4	Un approccio combinato	71
4	Restauro delle trasformate di Radon	75
4.1	Instabilità della ricostruzione	75
4.2	Pre-Processing	78
4.3	Struttura metodi di interpolazione	80
4.4	Interpolazioni	82
4.4.1	Interpolazione di Bertalmio	82
4.4.2	Interpolazione polinomiale	88
5	Risultati numerici	89
5.1	Dati sintetici	89
5.2	Dati clinici	98
5.3	Osservazioni	102
Conclusioni e prospettive future		105
Bibliografia		107

Introduzione

Nel 1917 il Professor Johann Radon propose una tecnica di ricostruzione di una funzione definita su un piano a partire da un particolare insieme di integrali della funzione stessa. Tali risultati furono le fondamenta su cui si costruì per arrivare allo sviluppo delle moderne tecniche di imaging (mediche e non) del quale facciamo uso tutt'ora. Infatti nel 1971 si ottenne la prima TAC di un paziente umano grazie ad Godfrey Hounsfield e Allan Mcleod Cormack, i quali implementarono la ricostruzione dimostrata del Professor Radon anni or sono. Nel caso dell'imaging medico possono però presentarsi dei problemi nel caso nel paziente siano presenti protesi metalliche permanenti che impediscono il corretto funzionamento di tali tecniche. Al fine di restaurare i risultati ottenuti nel caso sopracitato, negli anni sono state sviluppate tecniche basate su algoritmi di restauro di immagini. In questo lavoro si presenta dunque si presenta una breve spiegazione di come funziona la Tomografia Assiale Computerizzata ed i problemi che sorgono nel caso di presenza di componenti metalliche permanenti. Nel secondo si introduce cosa siano le immagini digitali, alcuni concetti di base e delle comuni operazioni che si effettuano su di esse. Nel terzo presentiamo una panoramica sui metodi di restauro di immagini digitali. Nel quarto, entriamo nel dettaglio dell'implementazione di tali metodi capitolo mentre nel quinto presentiamo i risultati ottenuti.

Capitolo 1

Tomografia

1.1 La trasformata di Radon

Introduciamo ora una breve panoramica dei risultati ottenuti dal Professor Radon riguardo la ricostruzione 2D ([1],[2]).

1.1.1 Definizione e proprietà

La trasformata di Radon è una trasformata integrale che mappa una funzione $f(x, y)$ definita su un piano, nell'insieme dei suoi integrali di linea:

$$f(x, y) \mapsto Rf$$

dove Rf , data la retta l di pendenza p e intercetta q , sarà definita come:

$$R(p, q) = \int_{-\infty}^{\infty} f(x, px + q) dx$$

Al variare di p e di q si ottiene una superficie anch'essa definita su un piano. Fissato p e variando q si avranno rette parallele tra loro lungo le quali si integra f . Fissato q e variando p invece si ha un fascio di rette tutte di diversa inclinazione passanti per il punto del piano $(0, q)$.

Vogliamo ora esprimere R in coordinate polari. Supponiamo di integrare f lungo rette parallele dirette come l'asse y . Allora avremo che

$$R(\theta = 0, x) = \int_{-\infty}^{\infty} f(x, y) dy$$

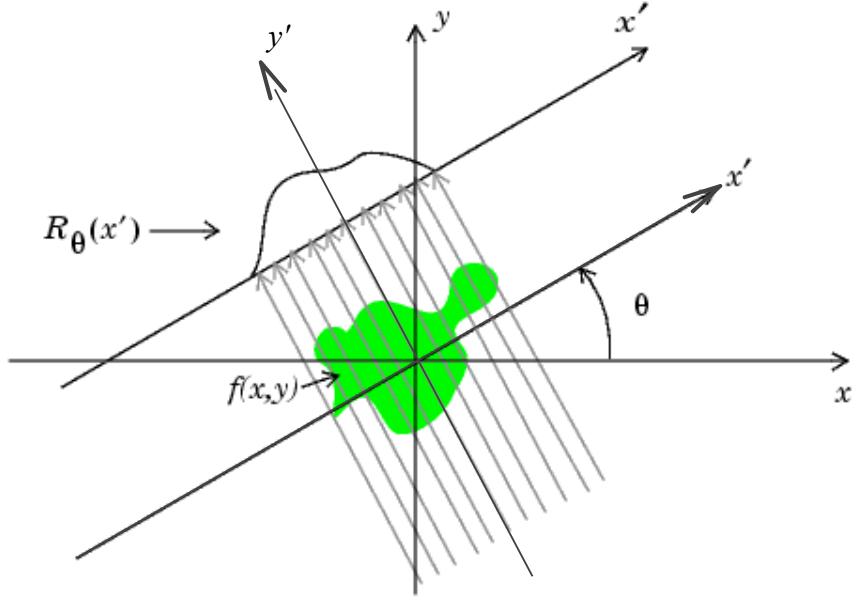


Figura 1.1: Sistema di riferimento ruotato

Se ora ruotiamo il sistema (x, y) di θ gradi, (fig 1.1) otteniamo il sistema di riferimento ruotato:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rispetto a tale sistema di riferimento, R si può scrivere come:

$$R(\theta, x') = \int_{-\infty}^{\infty} f(x', y') dy' = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta) dx dy$$

Facciamo notare che il determinante della matrice di rotazione è 1 quindi il rapporto tra i differenziali $dxdy$ e $dx'dy'$ è unitario.

La trasformata di Radon gode delle seguenti proprietà:

- Linearità

$$R(\theta, x')[\alpha f(x, y) + \beta g(x, y)] = \alpha R(\theta, x')[f(x, y)] + \beta R(\theta, x')[g(x, y)]$$

- Periodicità con periodo $T = 2\pi$

- Simmetria rispetto a $\theta = \pi$, ossia

$$R(\theta, x') = R(\theta + \pi, -x')$$

- Condizione sui momenti: $\forall k \geq 0$ intero avremo che il momento di ordine k :

$$G_k(\theta) := \int_{-\infty}^{\infty} x'^k R(x', \theta) dx'$$

sarà un polinomio omogeneo di ordine k rispetto a $\omega = (\cos\theta, \sin\theta)$

- Invarianza per rotazione: sia A una matrice di rotazione 2×2 e sia $M_A f(x, y) = f(A(x, y)')$ allora

$$(RM_A)f(x, y) = (M_A Rf)(\theta, x')$$

- Invarianza per dilatazione: sia $a > 0$ un numero reale positivo e sia D_a l'operatore di dilatazione radiale $D_a f(x, y) = f(ax, ay)$ allora

$$(RD_a f)(\theta, x') = \frac{1}{a} D_a(Rf)(\theta, x')$$

Osserviamo inoltre che la il *density plot* della trasformata di Radon prende il nome di sinogramma.

1.1.2 Radon e Fourier: Teorema della sezione centrale

Vediamo ora la relazione tra la trasformata di Fourier di f

$$M(k, l) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi i(kx+ly)} dx dy$$

e la trasformata di Fourier di R con θ fissato

$$P(\theta, k') = \int_{-\infty}^{\infty} R(\theta, x') e^{-2\pi i k' x'} dx'$$

Avremo che:

$$\begin{aligned}
 P(\theta, k') &= \int_{-\infty}^{\infty} R(\theta, x') e^{-2\pi i k' x'} dx' \\
 &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(x', y') dy' \right) e^{-2\pi i k' x'} dx' \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') e^{-2\pi i k'(x \cos \theta + y \sin \theta)} dx dy \\
 &= M(k' \cos \theta, k' \sin \theta)
 \end{aligned}$$

Ossia la trasformata di Fourier di una proiezione ottenuta ad un angolo θ rispetto all'asse x , $P(\theta, k')$, coincide con la trasformata di Fourier della $f(x, y)$, $M(k, l)$ calcolata lungo una retta nel piano (k, l) che forma un angolo θ con l'asse k .

1.1.3 Inversione della trasformata di Radon

Veniamo ora ad un punto cardine: come ricostruire f a partire dalla sua trasformata di Radon. La relazione tra le due trasformate di Fourier vista nel paragrafo precedente ci permette di ottenere due importanti risultati a riguardo:

1. Inversione: abbiamo appena visto che $P(\theta, k') = M(k' \cos \theta, k' \sin \theta)$, dunque prendendo la trasformata di R definita in coordinate polari e riportandola in coordinate cartesiane possiamo ricostruire f con la formula di inversione della trasformata di Fourier.
2. Unicità dell'inversione: se $f \in L^2$ ed $Rf = 0$ quasi ovunque allora $f = 0$ quasi ovunque

Dunque si può, almeno in teoria, utilizzare direttamente il teorema della sezione centrale per la ricostruzione della $f(x, y)$. Realmente è però impossibile avere le proiezioni per ogni coppia (θ, x') . Ne consegue che i campioni dello spettro di R saranno noti su un reticolo polare discreto. Per risalire a $f(x, y)$ si possono interpolare questi dati e stimare i campioni su un reticolo cartesiano. Questa procedura di interpolazione è inevitabilmente soggetta ad errori ad alta frequenze spaziale dove i dati si diradano introducendo così artefatti indesiderati nella superficie ricostruita. Per questi motivo i metodi di ricostruzione basati sull'inversione della trasformata di Fourier non vengono utilizzati spesso. Passiamo dunque ad un metodo alternativo sempre

basato sul teorema della sezione centrale. Sia $M(\theta, k')$ la trasformata di Fourier di f espressa in coordinate polari t.c. $k = k' \cos \theta$ e $l = k' \sin \theta$, allora possiamo antitrasformare secondo Fourier:

$$f(x, y) = \int_0^{2\pi} \int_0^\infty P(\theta, k') e^{2\pi i k' (x \cos \theta + y \sin \theta)} k' dk' d\theta$$

ottenendo così che se $R(\theta, x')$ è nota per ogni $\theta \in [0, 2\pi]$, allora f è ricavabile. Dalla proprietà di simmetria della trasformata di Radon ricaviamo, applicando la trasformata di Fourier, che $P(\theta, k') = P(\theta + \pi, -k')$, ergo possiamo riscrivere f come

$$\begin{aligned} f(x, y) &= \int_0^\pi \int_{-\infty}^\infty P(\theta, k') e^{2\pi i k' (x \cos \theta + y \sin \theta)} |k'| dk' d\theta \\ &= \int_0^\pi \int_{-\infty}^\infty P(\theta, k') e^{2\pi i k' x'} |k'| dk' d\theta \\ &= \int_0^\pi C(\theta, x') d\theta \end{aligned}$$

Con

$$C(\theta, x') = \int_{-\infty}^\infty P(\theta, k') |k'| e^{2\pi i k' x'} dk'$$

che prende il nome di *retroproiezione filtrata*.

Nel caso in cui si ricostruisca la funzione f da un numero finito di proiezioni, per il teorema del campionamento di Shannon-Nyquist avremo che la massima frequenza rappresentabile k'_{max} sarà t.c. $k'_{max} \leq \frac{1}{2\Delta x'}$ (con $\Delta x'$ che rappresenta il passo minimo sull'asse x'). Prendiamo dunque, come scelta naturale, il filtro

$$B(w) = \begin{cases} |w| & w \leq 1/(2\Delta x') \\ 0 & \text{otherwise} \end{cases}$$

Ne consegue:

$$\begin{aligned} C(\theta, x') &= \int_{-\infty}^\infty P(\theta, k') B(k') e^{2\pi i k' x'} dk' \\ &= \mathcal{F}^{-1}\{\mathcal{F}\{R\}\mathcal{F}\{b\}\} \\ &= R(\theta, x') * b(x') \end{aligned}$$

con

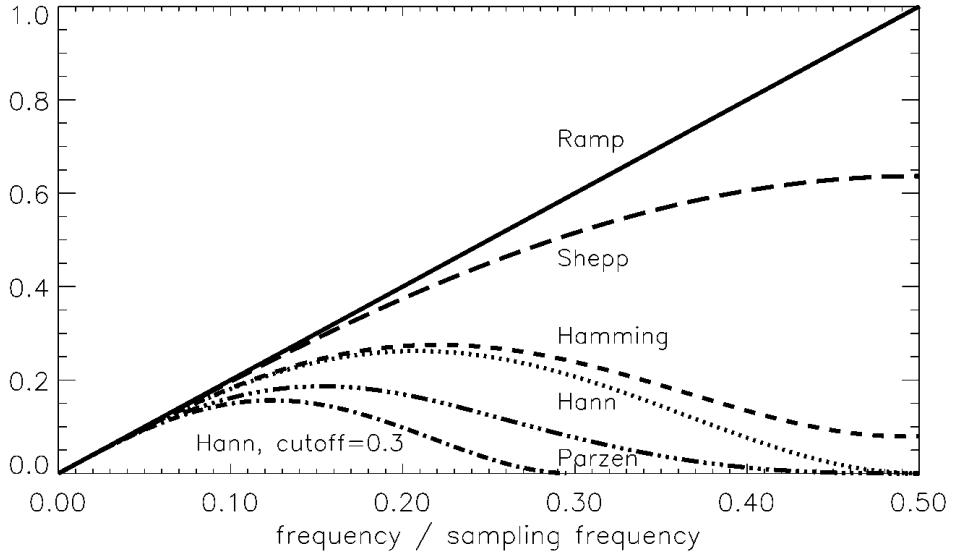


Figura 1.2: Possibili scelte di filtro per le retroproiezioni

$$\begin{aligned}
 b(x') &= \int_{-\infty}^{\infty} B(k') e^{2\pi i k' x'} dk' \\
 &= \frac{1}{2(\Delta x')^2} \frac{\sin(\pi x'/\Delta x')}{\pi x'/\Delta x'} - \frac{1}{4(\Delta x')^2} \left(\frac{\sin(\pi x'/2\Delta x')}{\pi x'/2\Delta x'} \right)^2
 \end{aligned}$$

Ricapitolando la retroproiezione filtrata consiste nei seguenti passi:

- Filtrare $R(\theta, \cdot) \quad \forall \theta$
- Retroproiettare $R(\theta, x')$ nel piano (x, y) lungo la retta $x \cos \theta + y \sin \theta = x', \quad \forall x'$
- Sommare (integrare) le proiezioni ottenute in θ da zero a π .

Osserviamo che altre scelte di B più smooth sono possibili. In figura 1.2 possiamo vedere le possibili scelte più comuni.

1.1.4 Esempio

Consideriamo ora una funzione di esempio $f(x, y)$ come quella in figura 1.3, ottenuta dal classico fantoccio Shepp-Logan di figura 1.5a

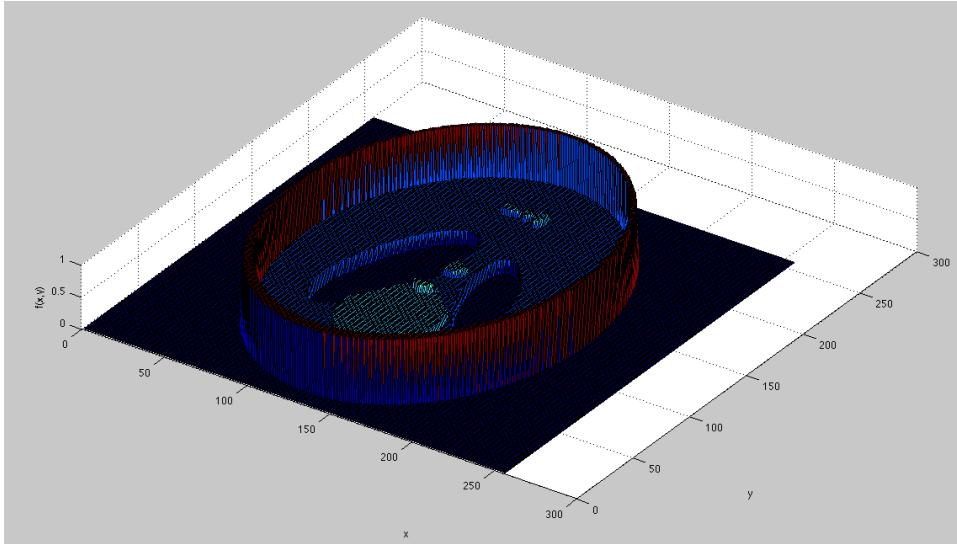


Figura 1.3: Superficie fantoccio Shepp-Logan

Facendo la trasformata di Radon di f , integrando per θ che varia di un grado alla volta da 0° a 359° , otteniamo la superficie R di figura 1.4 e l'immagine (vista dall'alto) di fig 1.5b.

Passiamo ora al problema inverso, ossia ricavare f da R . Come visto in precedenza, i passi da seguire sono:

1. Filtraggio trasformata di Radon con $b(x')$. Otteniamo così la superficie $C(x', \theta)$ di fig. 1.6 e fig. 1.7. Per avere un'idea più chiara della differenza di comportamento tra R e C in fig. 1.8 si vede un confronto tra le due.
2. Retroproiezione. Per ogni angolo θ la retroproiezione è una funzione di (x, y) generata da $C(\theta, \cdot)$, che viene mantenuto costante lungo le linee di integrazione $x \cos \theta + y \sin \theta$. In figura 1.9 vediamo la superficie generata da $C(20^\circ, \cdot)$.
3. Integrazione. Nella figura 1.10 vediamo come le somme parziali delle retroproiezioni filtrate si evolvono ogni 36° gradi. In figura 1.11 vediamo la superficie f ricostruita.

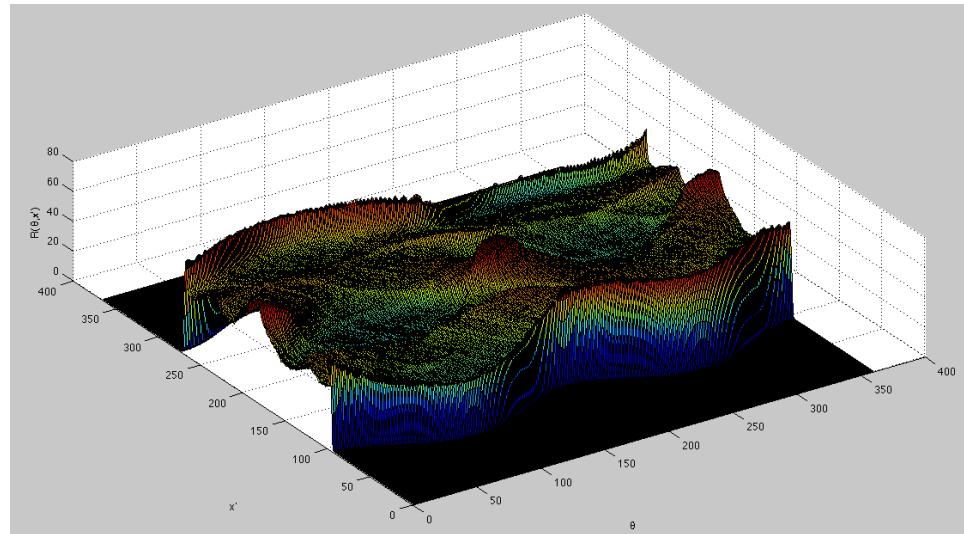
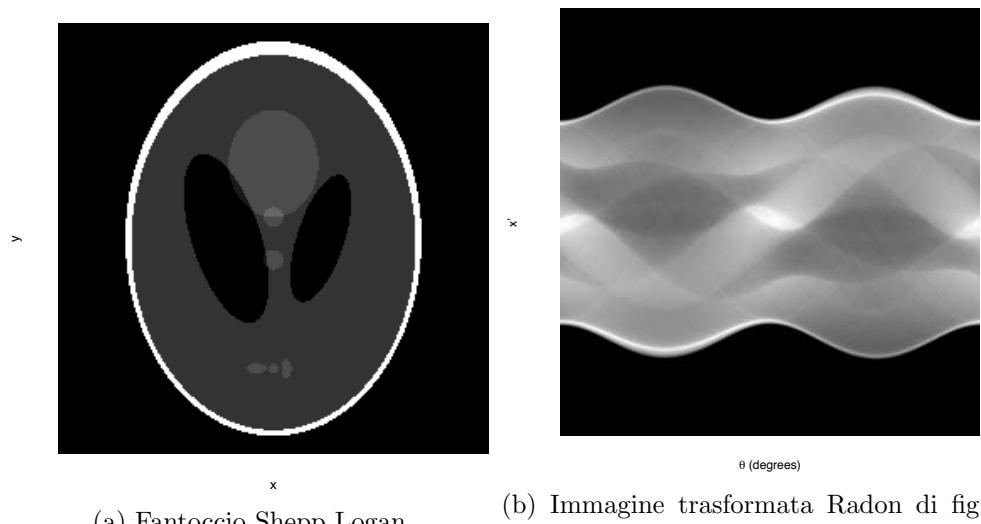
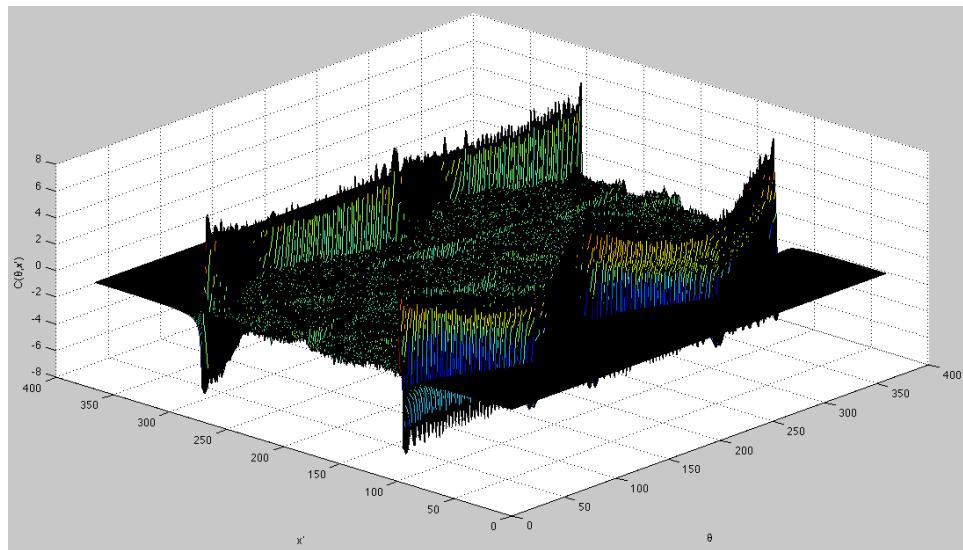
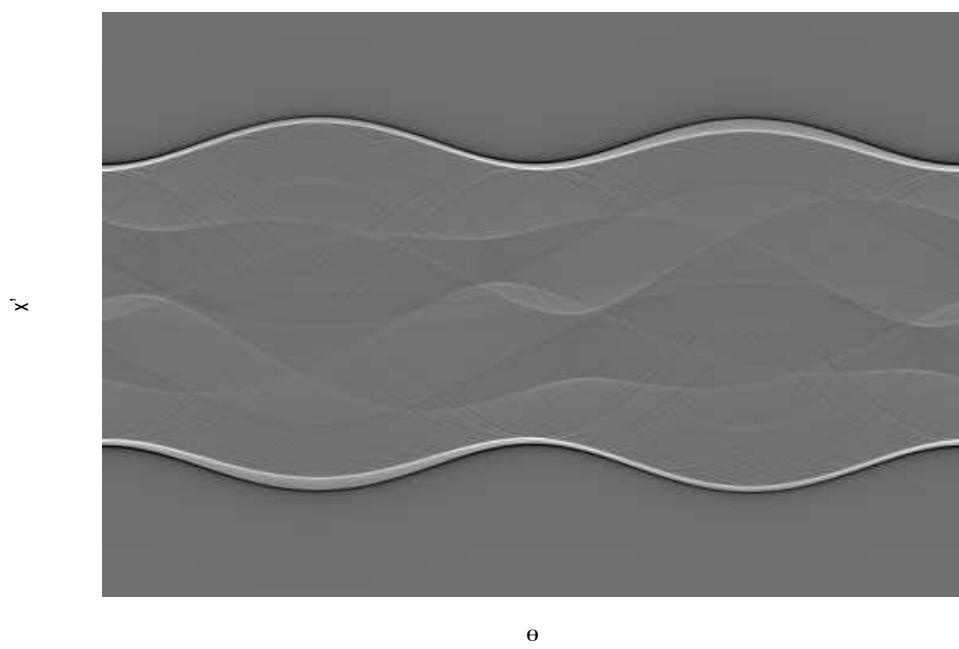
Figura 1.4: Superficie $R(\theta, x')$ 

Figura 1.5: Fantoccio Shepp Logan e relativo sinogramma

Figura 1.6: Trasformata di Radon filtrata con $|k'|$ Figura 1.7: Immagine Trasformata di Radon filtrata con $|k'|$

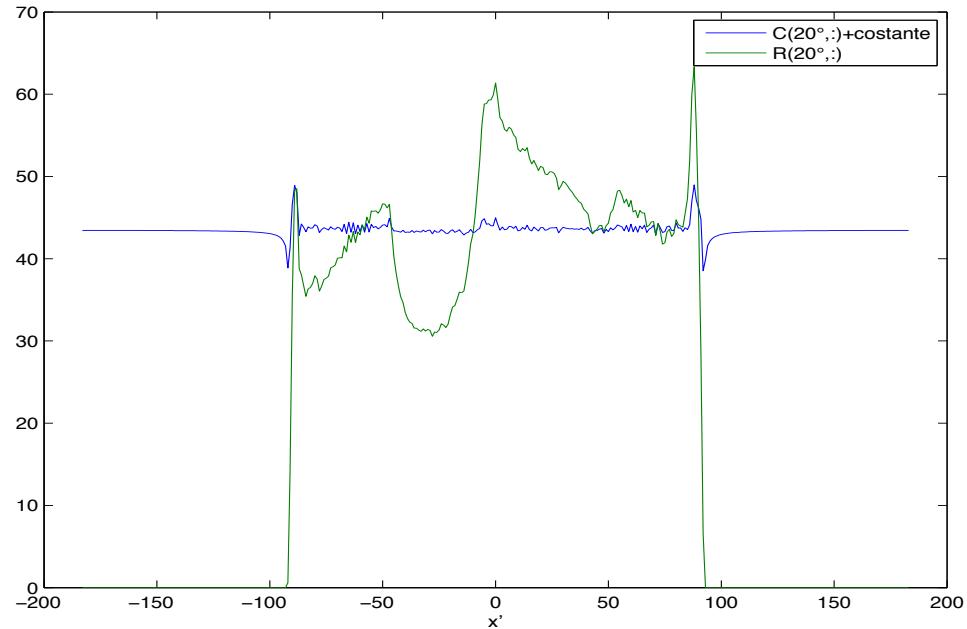


Figura 1.8: Confronto tra $R(\theta = 20^\circ, \cdot)$ e $C(\theta = 20^\circ, \cdot) + \text{costante}$

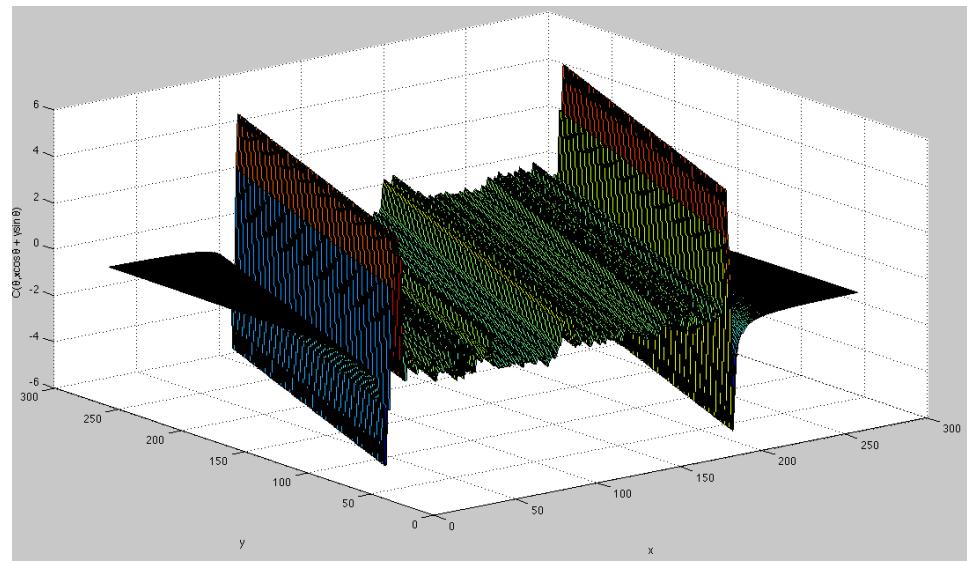


Figura 1.9: Retroproiezione filtrata per $\theta = 20^\circ$

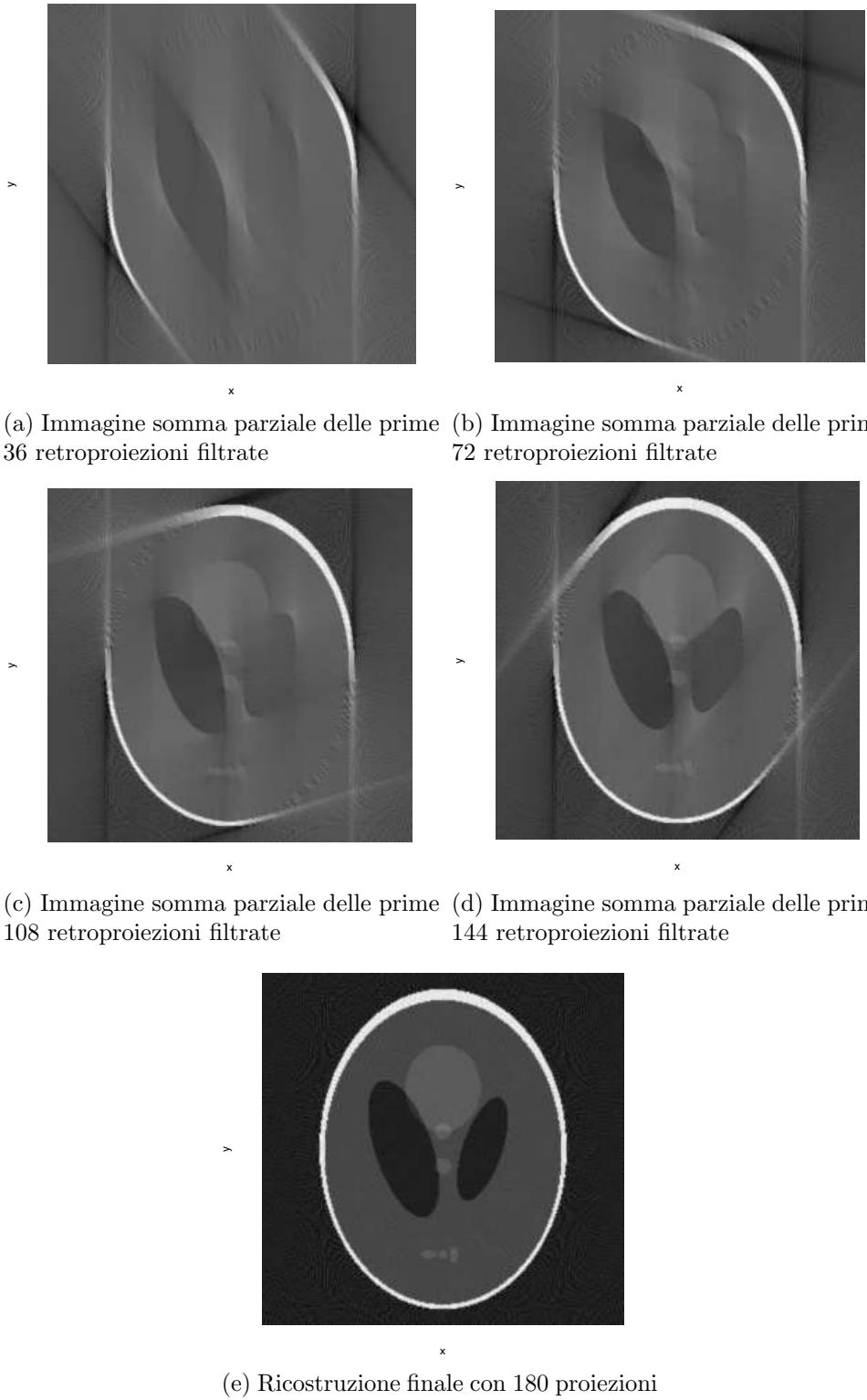


Figura 1.10: Ricostruzioni parziali con FBP

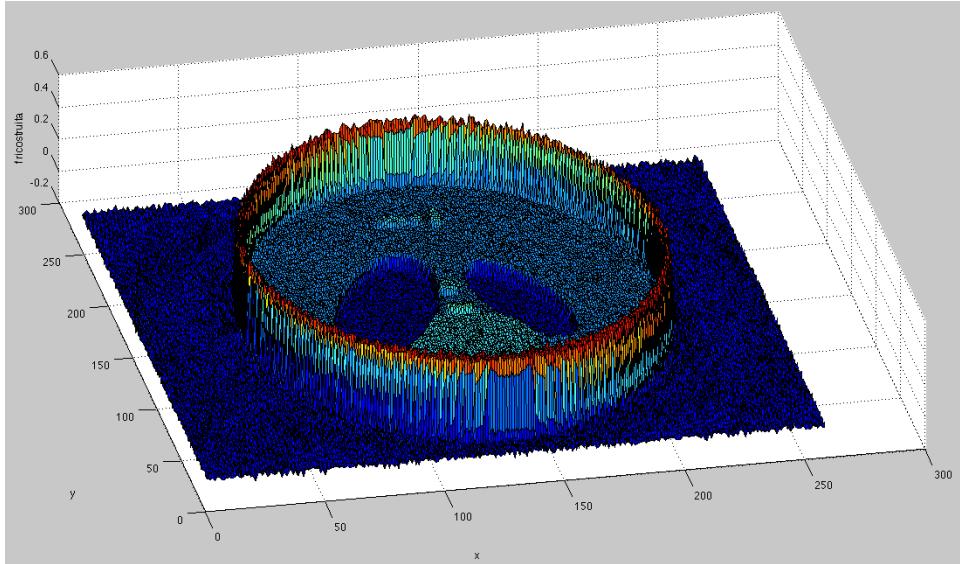


Figura 1.11: Ricostruzione finale con 180 proiezioni

1.1.5 Positura del problema inverso

Abbiamo visto in precedenza che nel caso di funzioni sufficientemente regolari (e.g. funzioni a supporto compatto e continue a tratti) la trasformata di Radon ha kernel di dimensione nulla ($f = 0$ q.o. $\iff Rf = 0$ q.o.). Questo non garantisce però l'unicità dell'inversione della trasformata di radon in quanto come condizione aggiuntiva necessaria v'è che il rango dell'operatore di inversione abbia dimensione pari all'intero spazio. Nel discreto sicuramente non avremo unicità nell'inversione, in quanto ovviamente non possiamo avere le proiezioni per ogni $\theta \in [0, 2\pi]$. La ragione sta nel seguente **Teorema**:

Siano $\theta_1, \dots, \theta_k$ un insieme finito di angoli e sia $K \subset \mathbb{R}^2$ un compatto con $f \in C_0^\infty(K)$. Sia $K_0 \subset U \subset K$ con U aperto. Allora esiste $f_0 \in C_0^\infty(K)$, tale che $f_0 = f$ su K_0 ed $Rf_0(\cdot, \theta_j) = 0$, $j = 1, \dots, k$.

dunque supponendo anche che si possano avere tutti i valori della variabile x' avendo un numero discreto di proiezioni si perde l'unicità.

Per quanto riguarda la stabilità si può dimostrare il seguente **teorema**: *sia f con supporto incluso nella palla B , allora:*

$$\frac{1}{C_{B,s}} \|Rf\|_{H^{s+1/2}} \leq \|f\|_{H^s} \leq C_{B,s} \|Rf\|_{H^{s+1/2}}$$

con $C_{B,s} > 1$.

Ne consegue che Rf è mezza derivata più regolare di f . il motivo di tale instabilità contenuta sta nel filtro $|k'|$ che viene applicato alle proiezioni nella FBP. Nel caso nei dati vi siano errori piccoli ma molto oscillanti, questi verranno amplificati dal filtraggio. Ergo la perdita di regolarità di f rispetto a Rf .

Osserviamo infine che l'inversione della trasformata di Radon si può fare in più modi (per esempio tramite ricostruzione iterativa basata su metodi di massima verosimiglianza), ma l'algoritmo di retroproiezione filtrata, come vedremo in seguito, è l'algoritmo maggiormente utilizzato nei macchinari per la Tomografia Assiale Computerizzata e dunque l'unico che considereremo.

1.2 Imaging medico

Gli ultimi decenni del XX secolo hanno testimoniato un significativo avanzamento nell'Imaging medico. Dalla scoperta dei raggi X a opera di Wilhelm Conrad Rontgen nel 1895 e per i 75 anni successivi, l'Imaging clinico si è basato sui convenzionali tubi a raggi X e sul binomio schermo di rinforzo - pellicola. Da quando nei primi anni '70 Allen Cormack e Godfrey Hounsfield hanno introdotto la Computed Tomography (TAC) basata sul modello matematico di Radon visto nel capitolo precedente l'Imaging clinico è entrato in una fase rivoluzionaria. Questo approccio ha permesso infatti di visualizzare per la prima volta le strutture anatomiche degli organi interni e di evidenziarne le relative patologie, in modo non invasivo e con una precisione senza precedenti. In modo analogo, l'applicazione di queste metodiche alla Medicina Nucleare, associata al continuo progresso della tecnica nel campo elettronico e informatico, hanno accelerato e ampliato lo sviluppo dell'Imaging, tanto da consentire modalità di scansione multidimensionali non solo morfologiche ma anche dinamiche, funzionali e metaboliche. Anche se la correlazione e la comprensione, dal punto di vista semeiotico, delle relazioni tra disturbi fisiologici e lesioni anatomiche è ancora oggetto di studio, si delinea la possibilità di una nuova e più completa comprensione delle varie patologie.

1.3 Tomografia Assiale Computerizzata

Tra le svariate applicazioni della trasformata di Radon, come appena accennato, una delle più importanti si trova nell'imaging medico. In particolare nella tomografia computerizzata. Tomografia ([3]): dal greco *τομοσ* (sezione) e *γραφετε* (scrivere), è un insieme di tecniche di imaging atte a determinare la struttura interna di un corpo non trasparente inviando dei segnali (onde, radiazioni) attraverso quest'ultimo. Comunemente vengono usate onde elettromagnetiche a svariate frequenze (raggi X, raggi γ , etc.) oppure onde acustiche. Nella tomografia computerizzata l'immagine ottenuta non è il risultato di misurazioni dirette come nelle lastre, ma come vedremo in seguito è frutto di un elaborazione delle rilevazioni.

La TAC si basa sul principio che la capacità di assorbimento o di dispersione dei raggi X dipenda dalla composizione del corpo in esame, quantificata dal coefficiente di attenuazione μ . Tale coefficiente è una funzione che dipende da diversi fattori (densità, energia dei raggi X, distanza dalla sorgente), ma soprattutto dalla densità di elettroni della sostanza, con diretta proporzionalità. Dunque, sostanze molto dense presenti nel corpo, o comunque con un elevato numero di elettroni avranno un alto coefficiente di attenuazione. Questo spiega perché le ossa, che contengono un'alta percentuale di calcio (20 elettroni), potassio (19 elettroni), fosforo (15 elettroni) e magnesio (12 elettroni), hanno un valore di μ più alto rispetto ai tessuti morbidi, composti principalmente da carbonio (6 elettroni), azoto (7 elettroni) e ossigeno (8 elettroni). Si considera come nullo il coefficiente di attenuazione dell'aria per comodità di calcolo, così da non avere attenuazione al di fuori del corpo.

Consideriamo ora un piano fissato che taglia un corpo umano (fig. 1.12a). Supponiamo di avere dei sottili fasci di raggi X paralleli tra loro che attraversano il corpo, emessi da una fila di sorgenti esterni al corpo, lungo una retta x' che forma un angolo θ con l'asse x . Supponiamo inoltre di avere una fila di detectors, disposti lungo una linea parallela a x' , sempre all'esterno del corpo, che misurano le intensità dei raggi X una volta attraversato il corpo (fig 1.12b). Cerchiamo di determinare la variazione di densità nella sezione del corpo a partire dalle rilevazione di intensità dei detectors ottenute ad ogni rotazione θ .

Prendendo come sistema di riferimento il piano (x', y') ottenuto della rotazione piana di θ gradi a partire dal piano (x, y) avremo che, per la legge

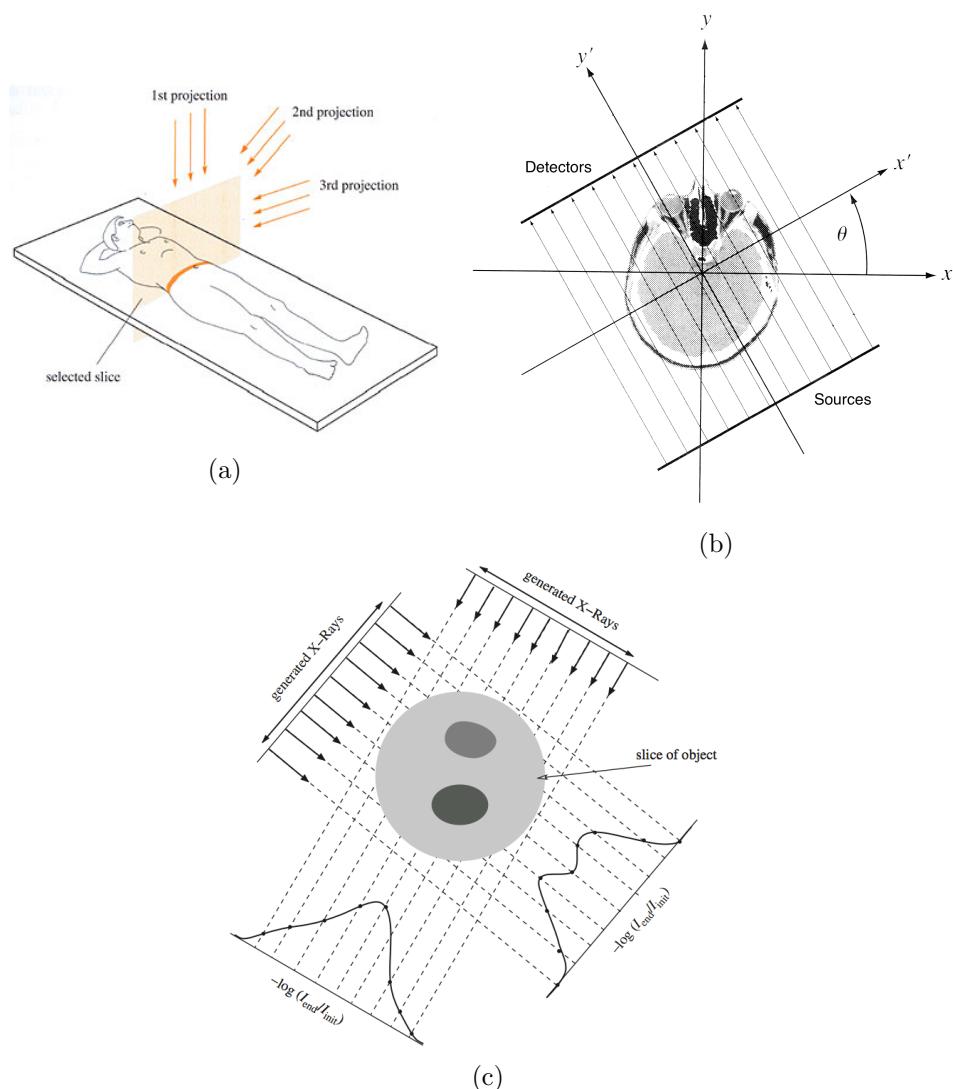


Figura 1.12: Immagini esemplificative funzionamento TAC

di Beer, lungo le rette dei fasci di raggi X la variazione di $I(x', y')$ sarà:

$$\frac{dI}{dy'} = -\mu(x', y')I$$

Integrando lungo le rette parallele a y' otteniamo:

$$\int_{I_0}^I \frac{ds}{s} = - \int_{-\infty}^{\infty} \mu(x', y') dy'$$

$$\log\left(\frac{I}{I_0}\right) = - \int_{-\infty}^{\infty} \mu(x', y') dy'$$

dunque al variare di θ avremo che il logaritmo del rapporto tra le due intensità corrisponde alla trasformata di Radon del coefficiente di attenuazione:

$$p(\theta, x') = \int_{-\infty}^{\infty} \mu(x', y') dy'$$

con p che prende il nome di *sinogram matrix* dove si salvano i valori in funzione dei valori discreti di θ e x' .

La tac dunque ricostruisce l'andamento del coefficiente di attenuazione tramite l'algoritmo di retroproiezione filtrata (FBP: filtered back projection) visto in sezione 1.1.3, applicato a $\log\left(\frac{I_0}{I}\right)$.

Nella pratica, anziché il coefficiente di attenuazione μ , si preferisce usare il coefficiente di Hounsfield (misurato di HU), dove, per una particolare sostanza X , il relativo coefficiente sarà attenuto tramite normalizzazione rispetto al coefficiente di attenuazione dell'acqua:

$$H_X = \frac{\mu_X - \mu_{acqua}}{\mu_{acqua}} \times 1000$$

1.4 Metal Artifact Reduction

Nel caso in cui nella regione in osservazione della TAC vi siano presenti oggetti di elevata densità elettronica (per esempio impianti metallici come peacemakers, viti o protesi) solo un ridotto numero di fotoni ad alta energia riesce a raggiungere i detectors. Ne consegue che $\log\left(\frac{I_0}{I}\right)$ in tali posizioni, assume valori elevatissimi del tutto inconsistenti con il resto dei valori della

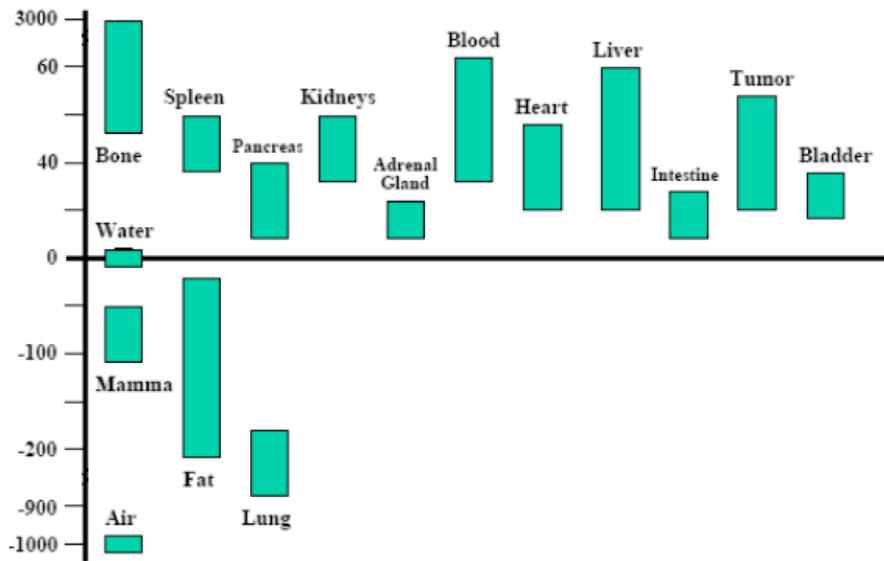


Figura 1.13: HU per diverse componenti corporee

trasformata di Radon. Applicando poi l'algoritmo di FBP alla trasformata ottenuta si formano nell'immagine ricostruita degli artefatti con forti contrasti scuri e chiari del tutto incoerenti con il resto dell'immagine. Facciamo notare che dato che il metallo blocca i raggi X indipendentemente dall'angolo da cui arrivano ne consegue che sul sinogramma sarà presente una spessa curva sinusoidale molto chiara per ogni componente metallica presente. La TAC è uno degli strumenti di imaging più utilizzati ogni giorno nelle pratiche cliniche. Tuttavia la presenza di tali artefatti diminuisce la qualità dell'immagine e di conseguenza influenza negativamente la capacità del medico di effettuare una corretta diagnosi. Nella radioterapia, per esempio, le strisce bianche dovute al metallo possono mettere in ombra il tumore, rendendone difficile la delineazione. Inoltre questo influisce anche sul calcolo delle dosi di radiazioni della terapia.

Negli ultimi 30 anni, diversi algoritmi sono stati sviluppati per ridurre la presenza di tali artefatti ([4], [9]). Tali algoritmi MAR (metal artifact reduction) possono essere divisi in due categorie principali:

- Metodi di interpolazione
- Metodi iterativi

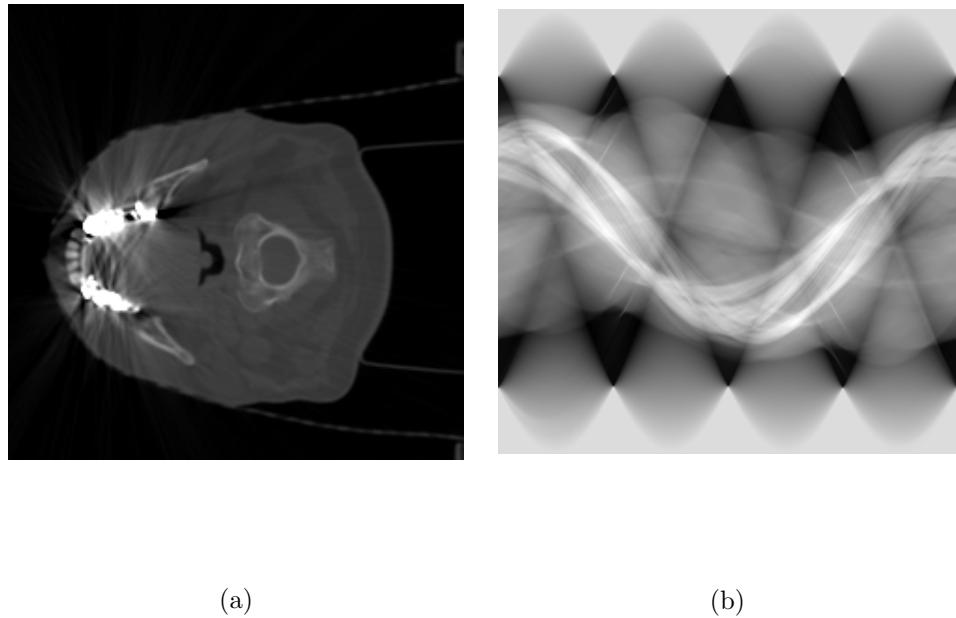


Figura 1.14: TAC con presenza di artefatti e relativo sinogramma

Nei primi le proiezioni attraverso il metallo vengono considerati dati mancanti. Il sinogramma incompleto viene poi completato tramite interpolazione a partire dai dati non corrotti. Come vedremo in seguito, varie tecniche vengono utilizzate in questo ambito: interpolazione lineare, polinomiale, direzionale, interpolazione tramite l'uso di FFT su nodi non equispaziati ed interpolazione tramite metodi di inpainting basati sulla minimizzazione di un funzionale variazionale o sulla risoluzione di una equazione differenziale alle derivate parziali.

Nei metodi di ricostruzione iterativa, viene modellizzata la fisica dietro il problema degli artefatti. Oltre al fenomeno di attenuazione eccessiva descritto in precedenza (noto come beam hardening), alcune tra le altre cause di deterioramento della qualità dell'immagine sono: la presenza di rumore, dispersione ottica (scattering) e aliasing spaziale. L'uso di tali metodi è utile quando le proiezioni attraverso il metallo possono ancora fornire delle informazioni utili. Dunque se il metallo è troppo denso e impedisce a tutti i fotoni di arrivare ai detectors, tali metodi sono inutilizzabili. Si capisce dunque che i metodi iterativi richiedono il possesso del sinogramma originale, che spesso

non è ottenibile dai tomografi clinici. I metodi iterativi si dividono in due categorie principali: algebrici (e.g. algebraic reconstruction ART) e statistici (maximum likelihood expectation maximization (MLEM), Bayesian based EM optimization, TV-based compressed sensing (CS) reconstructions).

Osserviamo che, in un confronto tra i due, i metodi di interpolazione sono computazionalmente più economici e possono essere applicati partendo dall'immagine ricostruita a discapito di dover introdurre inevitabilmente ulteriore errore dovuto alla FBP.

Capitolo 2

Immagini digitali

2.1 Concetti introduttivi

Un immagine può essere definita come una funzione $f(x, y)$ dove (x, y) sono coordinate spaziali piane, e l'ampiezza di f per ogni (x, y) è chiamata intensità o livello di grigio dell'immagine in tal punto. Quando x, y ed i valori di f sono tutti finiti definiamo l'immagine *digitale*. Un immagine digitale è composta da un numero finito di elementi, ognuno dei quali ha una posizione ed un valore. Questi elementi sono noti come di *picture elements*, *image elements*, *pel*, *pixels*. I valori di ampiezza di f , sono quantità scalari positive il cui significato fisico è determinato dalla natura dell'origine dell'immagine. Quando un immagine viene generata da un processo fisico, i suoi valori sono proporzionali all'energia radiata da una sorgente fisica (e.g onde elettromagnetiche), di conseguenza $f(x, y)$ deve essere non nulla e finita, ergo $0 < f(x, y) < \infty$. Nonostante l'estrema varietà di metodi di acquisizione di immagini, l'obbiettivo nell'elaborazione è sempre lo stesso in tutti i metodi: generare immagini digitali a partire dai dati rilevati. L'output della maggiore parte dei sensori è una tensione con forma d'onda continua, la cui ampiezza ed il comportamento spaziale dono indice del comportamento del fenomeno fisico osservato. Per creare immagini digitali, abbiamo bisogno di convertire il segnale continuo prodotto dai sensori in forma digitale. Questo implica il coinvolgimento di due processi: il campionamento e la quantizzazione. L'idea di base dietro il campionamento e la quantizzazione è illustrata nella figura 2.1, dove vediamo un'immagine continua $f(x, y)$ che vogliamo convertire in immagine digitale. Il campionamento si riferisce al

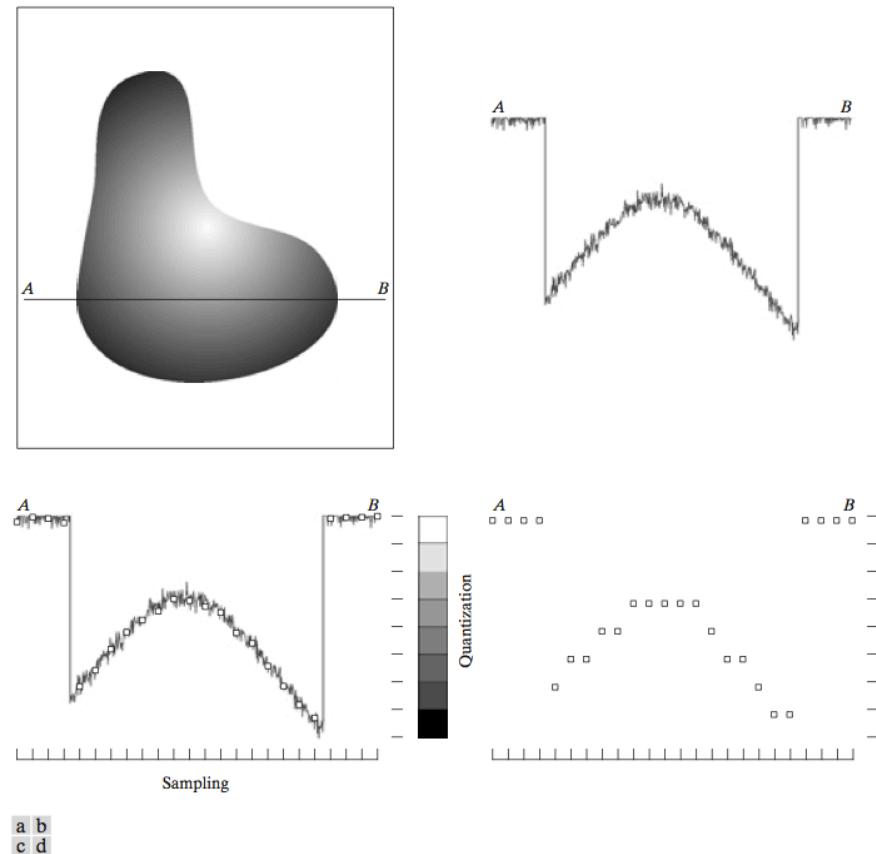


Figura 2.1: Campionamento e quantizzazione: (a) Immagine continua. (b) valori assunti da f lungo il segmento AB. (c) campionamento e quantizzazione. (d) digitalizzazione di f lungo il segmento AB. [6]

discretizzare le coordinate spaziali mentre la quantizzazione corrisponde nel discretizzare i valori di ampiezza di f . Partendo dall’alto e ripetendo il procedimento descritto in figura per ogni linea orizzontale si produce così una versione digitale dell’immagine continua. Il risultato del processo di campionamento e quantizzazione è una matrice di numeri reali. Assumendo che un’immagine continua $f(x, y)$ viene campionata con M righe e N colonne avremo che

$$f(x, y) = \begin{pmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{pmatrix}$$

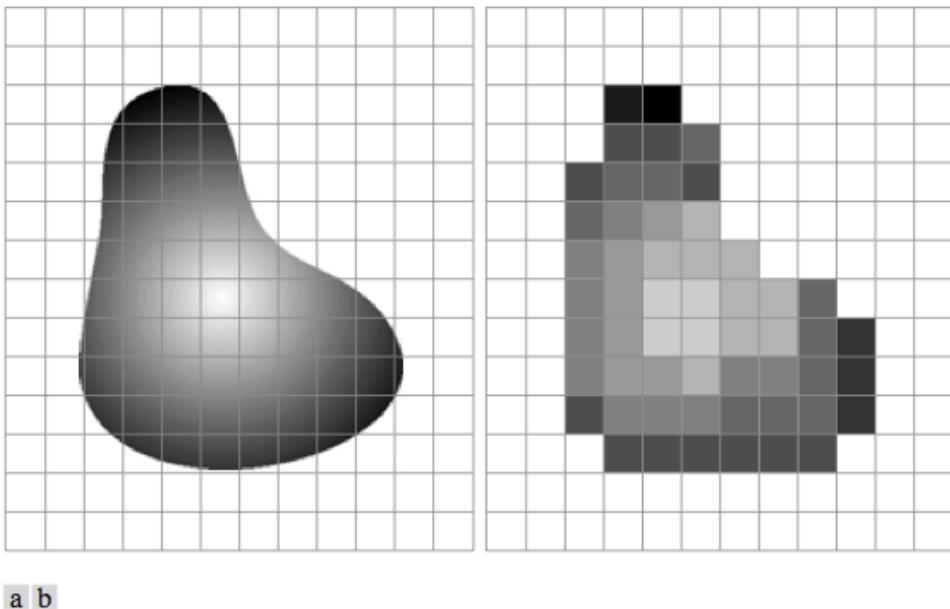
**a b**

Figura 2.2: (a) Immagine continua. (b) Immagine digitale. [6]

dove ogni elemento dell'immagine corrisponde ad un pixel. IL processo di digitalizzazione richiede decisioni riguardo la scelta dei valori di M e N e per il numero L di livello di grigio. La scelta di M ed N determinerà la risoluzione spaziale dell'immagine, ossia la misura del più piccolo dettaglio distinguibile all'interno dell'immagine. L invece determina il contrasto ossia la risoluzione dei livello di grigio

2.2 Immagini: segnali 2D

Una immagine digitale può essere considerata come un segnale discreto in due dimensioni. In particolare, ogni immagine può essere considerata come una sequenza di impulsi:

$$f(n_1, n_2) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} f(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)$$

Vediamo ora un importante classe di operatori per il trattamento delle immagini. Un applicazione T si dice sistema LTI se valgono le proprietà di

linearità

$$T[ax_1(n_1, n_2) + bx_2(n_1, n_2)] = ay_1(n_1, n_2) + by_2(n_1, n_2)$$

e di invarianza spaziale

$$T[x_1(n_1 - m_1, n_2 - m_2)] = y_1(n_1 - m_1, n_2 - m_2)$$

L'azione di T su una immagine si può scrivere come

$$\begin{aligned} y(n_1 - m_1, n_2 - m_2) &= T[x(n_1 - m_1, n_2 - m_2)] \\ &= T \left[\sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} f(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2) \right] \\ &= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} x(k_1, k_2) T[\delta(n_1 - k_1, n_2 - k_2)] \end{aligned}$$

dunque la risposta impulsiva $h(n_1, n_2)$ invece si definisce come $h(n_1, n_2) = T[\delta(n_1, n_2)]$ e l'azione dell'operatore T si può scrivere come

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$$

Passiamo ora all'analisi spettrale delle immagini. L'estensione al caso discreto e periodico (prendiamo la ripetizione periodica della nostra immagine) della trasformata di Fourier si scrive come

$$F(k, l) = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(m, n) e^{-j2\pi(k \frac{m}{M} + l \frac{n}{N})}$$

che corrisponde anche'essa ad un segnale periodico e discreto. La sintesi invece si effettua con la formula di inversione

$$f(m, n) = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} F(k, l) e^{j2\pi(k \frac{m}{M} + l \frac{n}{N})}$$

La trasformata discreta di Fourier gode di svariate proprietà:

- linearità

- separabilità:

$$x(n_1, n_2) = x_1(n_1)x_2(n_2) \iff X(k, l) = X_1(k)X_2(l)$$

- convoluzione:

$$f(n_1, n_2) = x(n_1, n_2) * y(n_1, n_2) \iff F(k, l) = X(k, l)Y(k, l)$$

- moltiplicazione:

$$f(n_1, n_2) = x(n_1, n_2)y(n_1, n_2) \iff F(k, l) = X(k, l)*Y(k, l)$$

- valore iniziale

$$F(0, 0) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} f(k_1, k_2) = NME[f]$$

Dunque $F(0, 0)$ è proporzionale al valore medio dei livelli di grigio dell'immagine, mentre le componenti $F(k, l)$ forniscono informazioni su come le componenti (k, l) sono contenute nel segnale.

Il motivo molto importante per cui viene introdotta la trasformata discreta di Fourier sta nella proprietà di convoluzione e nell'algoritmo di calcolo della trasformata FFT (fast Fourier transform).

L'azione di un filtro h con risposta impulsiva $h(n_1, n_2)$ può essere scritta come

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$$

il che richiede un costo computazionale all'incirca di $(N+M)^2$ flops (floating point operations). Per la proprietà di convoluzione avremo che

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2) \iff Y(k, l) = X(k, l)H(k, l)$$

ergo

$$y = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} X(k, l)H(k, l)e^{j2\pi(k\frac{m}{M} + l\frac{n}{N})}$$

la quale operazione grazie all'algoritmo di FFT ha un costo computazione dell'ordine di $(N + M) \log_2(N + M)$

2.3 Filtri

Vediamo ora degli esempi di comuni elaborazioni spaziali delle immagini.

- Filtro passa basso: attenuano le alte frequenze e mantengono le più basse. in figura 2.3a possiamo osservare l'andamento del modulo della funzione di trasferimento di un tipico filtro passa basso con

$$h(m, n) = \frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

I filtri passa basso vengono usati per il blurring dell'immagine e per la riduzione del rumore, supponendo che esso sia additivo (entreremo di più nel merito di questa questione in seguito). L'operazione di blurring è normalmente utilizzata in fase di pre-elaborazione, allo scopo di eliminare piccoli dettagli inutili o addirittura dannosi per le successive elaborazioni (per esempio finalizzate all'estrazione di regioni piuttosto ampie), ovvero di compensare piccole imperfezioni quali le interruzioni che spesso si verificano nelle linee di contorno. Un filtro lineare di smoothing, come quello dell'esempio, è semplicemente un filtro di media mobile: esso sostituisce ad ogni pixel la media dei valori dei pixel nell'intorno definito dalla maschera, così riducendo l'entità delle differenze di grigio tra punti vicini. Il più semplice può essere realizzato mediante una maschera con pesi tutti uguali, di valore tale da dare un risultato compatibile con la scala dei grigi adottata. E' proprio l'introduzione di tali valori che dà luogo al blurring, ed anche alla riduzione del rumore. L'effetto di smoothing è tanto più accentuato quanto più grande è la dimensione dell'intorno. La riduzione dell'evidenza dei dettagli presenti nell'immagine, e quindi del contrasto, è una operazione che riduce le componenti spaziali di alta frequenza presenti nell'immagine, e questo spiega dunque la loro validità come filtri passa basso.

- Filtri passa alto: attenuano le basse frequenze e mantengono le più alte.

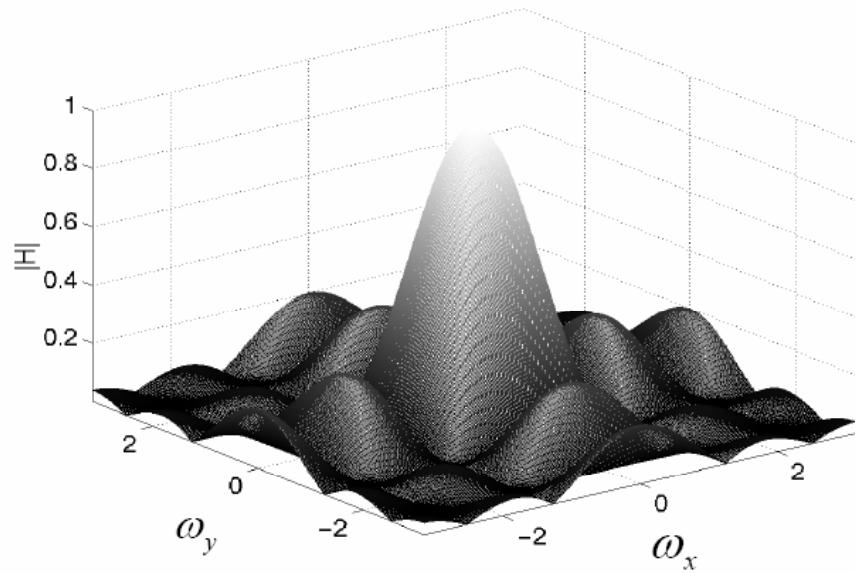
I filtri spaziali di sharpening (passa alto) sono utilizzati per aumentare il contrasto locale dell'immagine, in modo da arricchire i dettagli fini, o per evidenziare i contorni degli oggetti. A tal fine questi filtri lavorano aumentando le differenze tra pixel vicini. Sono pertanto complementari ai filtri di smoothing. Possono quindi provocare l'aumento del rumore presente nell'immagine, per cui la loro azione va spesso compensata con quella di un filtro per la riduzione del rumore. Evidenziare le differenze tra pixel significa esaltare il contenuto delle componenti ad alta frequenza spaziale presenti nell'immagine, per cui queste operazioni sono dei filtraggi passa alto. Per accrescere le differenze tra pixel vicini, una maschera deve avere i pesi centrali di segno opposto a quelli periferici. Esempio:

$$h(m, n) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

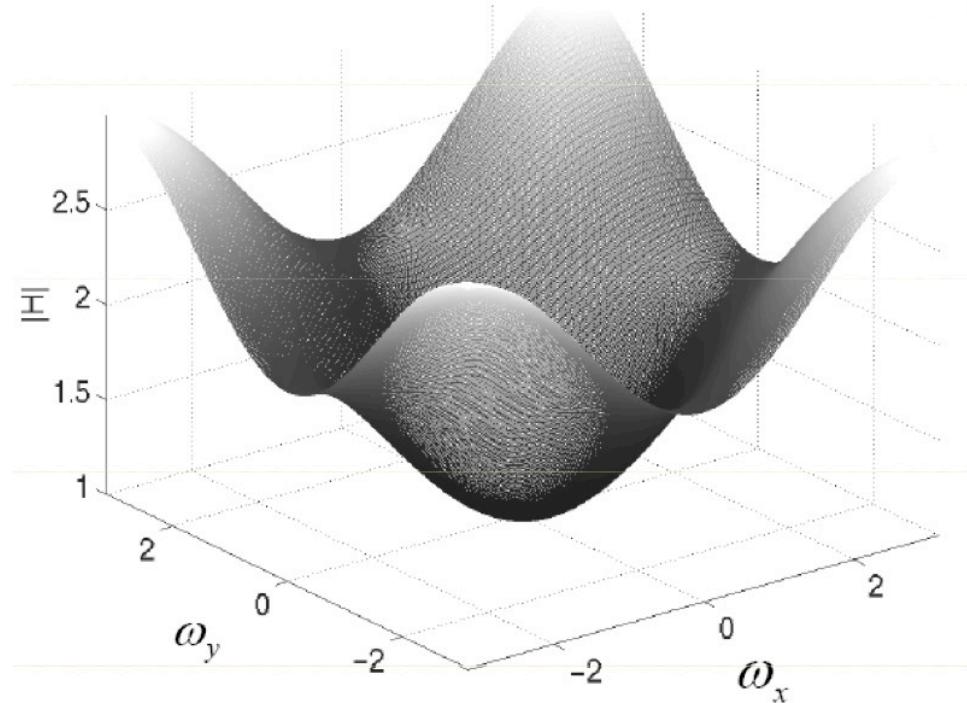
il cui modulo della funzione di trasferimento presenta l'andamento di figura 2.3b. Una caratteristica tipica di questi filtri è la totale eliminazione della informazione di sfondo.

- Filtri passa banda: combinando opportunamente dei filtri passa basso e passa alto si possono ottenere dei filtri che mantengono solo determinate frequenze all'interno di un range limitato, ed attenuano le frequenze al di fuori di questo range.
- Filtro mediano: con questo filtro ogni pixel viene sostituito dal valore mediano dei pixel nell'intorno. Si tratta di un filtro non lineare. Nell'intorno di ogni pixel si devono ordinare i valori dei pixel in esso contenuti, compreso quello centrale, ed il valore mediano di essi è assegnato al pixel centrale. Quindi l'effetto del filtro mediano è di forzare i pixel ad assumere un valore uguale a quello di uno dei pixel circostanti, eliminando così eventuali picchi isolati di intensità, che sono quelli con cui di solito si manifesta il cosiddetto rumore impulsivo.

Osserviamo che Il filtro passa alto con somma dei pesi pari a zero ha in pratica un effetto derivativo sull'immagine cui è applicato: l'applicazione di operazioni di differenza per aumentare la nitidezza di un'immagine digitale



(a) Funzione di trasferimento filtro passa basso



(b) Funzione di trasferimento filtro passa alto

Figura 2.3: Funzioni di trasferimento filtri passa basso e passa alto

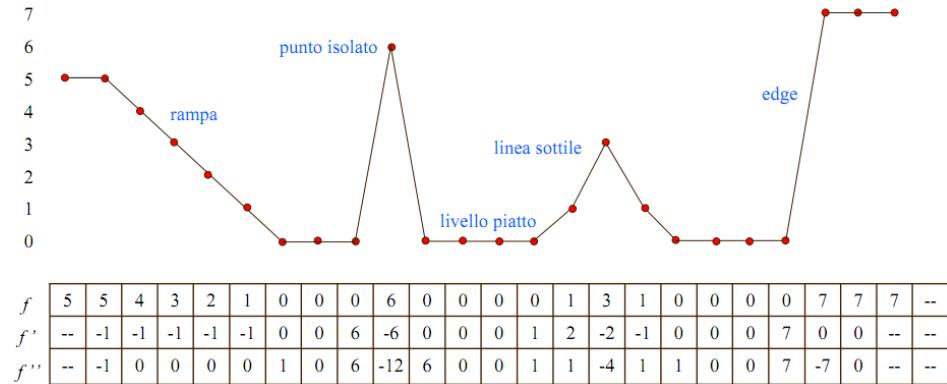


Figura 2.4

è analoga alla differenziazione di una funzione continua, così come l'operazione di media è analoga all'integrazione. E' quindi abbastanza naturale che i metodi di sharpening possano direttamente basarsi su operazioni esplicite di derivazione, in particolare mediante l'applicazione di operatori come il gradiente o il laplaciano. E' opportuno, in via preliminare, prendere in considerazione alcune proprietà fondamentali dell'operazione di derivazione nel discreto, limitandosi a considerare, per semplicità di esposizione, il caso $1D$. In particolare, in figura 2.4 osserviamo il comportamento delle derivate in corrispondenza a particolari andamenti del livello di grigio $f(x)$ (qui vengono usate differenze finite in avanti, ma il comportamento deve essere qualitativamente simile indipendentemente dall'approssimazione usata per le derivate).

Dai valori riportati, si può osservare che la derivata prima è diversa da zero lungo tutta la rampa, mentre la derivata seconda è diversa da zero solo all'inizio e alla fine della rampa. Poiché i contorni nelle immagini digitali sono di questo tipo (più o meno lenti), si può concludere che la derivata prima produce contorni piuttosto spessi, mentre la derivata seconda dà luogo a edge più sottili ma "doppi". In corrispondenza al punto isolato, la risposta della derivata seconda è più intensa dato il carattere più aggressivo della derivata seconda rispetto alla derivata prima nell'evidenziare brusche variazioni del livello di grigio. Quindi la derivata seconda è in grado di accentuare i dettagli fini (compreso il rumore) meglio della derivata prima. La risposta delle due derivate è di intensità simile nel caso dell'edge, ma si può ancora notare il "doppio" edge rilevato dalla derivata seconda. Questa caratteristica viene

sfruttata negli algoritmi di estrazione dei contorni (edge detection).

Riassumendo, in generale:

- le derivate prime danno luogo a contorni più spessi
- le derivate seconde hanno una risposta più forte ai dettagli fini
- le derivate prime hanno una risposta più forte agli edges ripidi
- le derivate seconde producono una risposta doppia agli edges

Da queste considerazioni, emerge come l'uso della derivata seconda appaia in realtà più adeguato per operazioni di miglioramento di qualità, soprattutto se l'immagine contiene dettagli fini. Entrambe le operazioni trovano peraltro la loro applicazione principale, da sole o in combinazione con altre, nella estrazione dei contorni.

2.4 Estrazione contorni

Lo scopo del riconoscimento dei contorni (edge detection) è marcare i punti di un'immagine digitale in cui l'intensità luminosa cambia bruscamente. Bruschi cambiamenti delle proprietà di un'immagine sono di solito il sintomo di eventi o cambiamenti importanti del mondo fisico di cui le immagini sono la rappresentazione. Questi cambiamenti possono essere ad esempio: discontinuità della profondità, discontinuità dell'orientamento delle superfici, modifica delle proprietà dei materiali, e variazioni dell'illuminazione proveniente dall'ambiente circostante. L'operazione di riconoscimento dei contorni genera immagini contenenti molte meno informazioni rispetto a quelle originali, poiché elimina la maggior parte dei dettagli non rilevanti al fine dell'individuazione dei contorni, conservando invece le informazioni essenziali per descrivere la forma e le caratteristiche strutturali e geometriche degli oggetti rappresentati. Esistono molti metodi per riconoscere i contorni, ma la maggior parte può essere raggruppata in due categorie: metodi basati sulla ricerca (search-based) e metodi basati sull'attraversamento dello zero (zero-crossing). I metodi basati sulla ricerca riconoscono i contorni cercando i massimi ed i minimi della derivata del primo ordine dell'immagine, di solito cercando la direzione in cui si ha il massimo gradiente locale. I metodi zero-crossing cercano i punti in cui la derivata del secondo ordine passa per lo zero, solitamente il laplaciano.

- Metodo del gradiente: dopo aver calcolato la derivata (caso 1D) o il gradiente (caso 2D), il passo successivo è applicare una soglia, per valutare i punti dell’immagine in cui potrebbe essere presente un contorno. Più bassa è la soglia, più linee verranno individuate, ed i risultati diverranno sempre più affetti da disturbi causati dal rumore di fondo, con conseguenti maggiori possibilità di falsi riconoscimenti. Più alta è la soglia e più ci sarà il rischio di non rilevare contorni formati da linee sottili o segmentate. Come menzionato in precedenza la derivata prima genera degli edges abbastanza spessi, quindi spesso in seguito al thresholding viene applicata una procedura morfologica di edge thinning, al fine di rimuovere i punti indesiderati su un contorno di una immagine. I metodi di questa categoria si differenziano tra loro per come viene approssimato il gradiente. Osserviamo inoltre che il modulo del gradiente è un rilevatore non direzionale, ossia ha un comportamento isotropo. Per ottenere un rilevatore direzionale, per esempio anzichè fare il thresholding del modulo del gradiente si può fare il thresholding della derivata direzionale nella direzione interessata.
- Metodi basati sul Laplaciano: i metodi zero-crossing cercano i punti in cui la derivata seconda (1D) o il laplaciano (2D) passa per lo zero. Il Laplaciano è un operatore isotropo, la cui risposta è pertanto indipendente dal direzione delle discontinuità presenti nell’immagine. In altri termini, è dotato di invarianza alla rotazione. Esso è più sensibile al rumore rispetto al gradiente e a causa del rumore l’applicazione del laplaciano genera falsi edge. Inoltre il Laplaciano genera contorni più spessi. Quindi l’applicazione del Laplaciano da un lato porta il vantaggio di avere informazioni puntuali sulla rilevazione dei contorni, mentre lo svantaggio è quello di una maggiore sensibilità al rumore. Utilizzare un algoritmo di zero-crossing senza l’applicazione di una soglia, porterebbe ad interpretare ogni variazione di intensità come fosse un contorno, dunque il metodo non solo trova gli zeri del laplaciano ma valuta se la varianza nell’intorno dello zero sia maggiore di una certa soglia. In figura 2.5c possiamo vedere come non tutti gli zeri del laplaciano corrispondono ad un edge dell’immagine. Analogamente ai metodi della categoria precedente, metodi di questa categoria si differenziano tra loro per come viene approssimato il laplaciano.

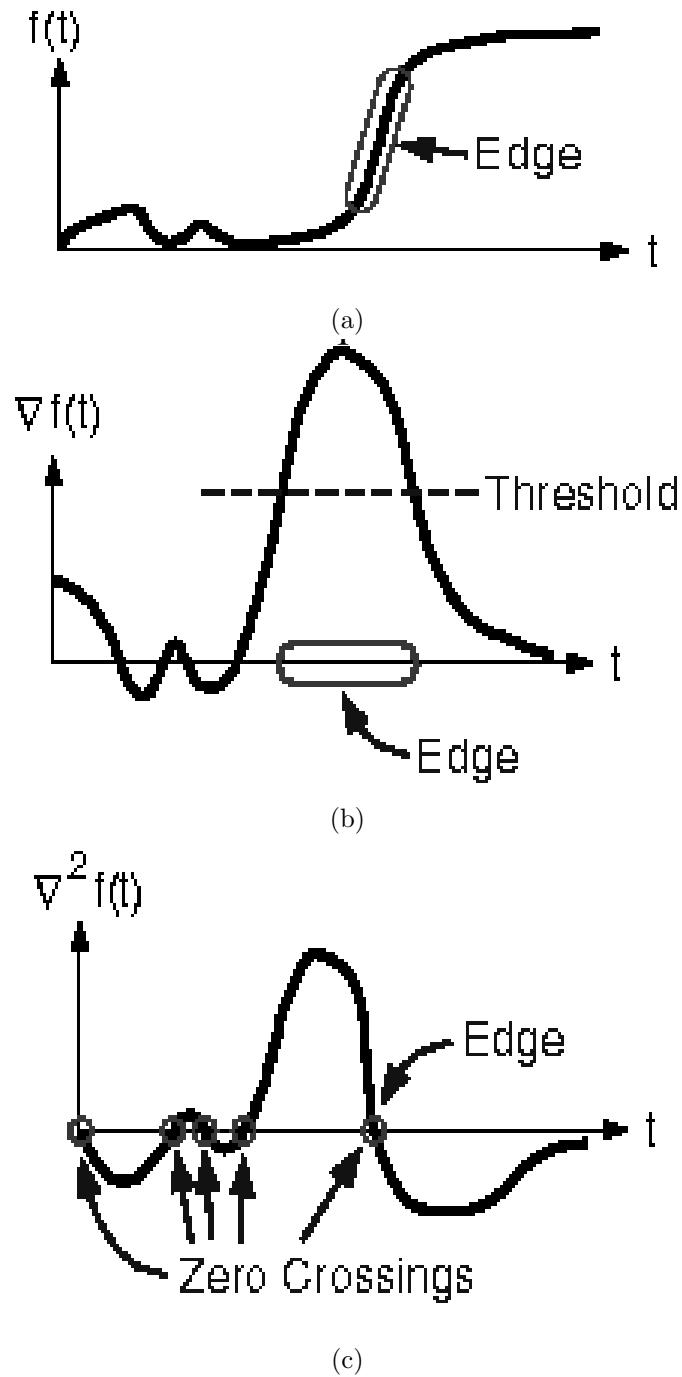


Figura 2.5: Estrazione contorni con thresholding gradiente e zero-crossing del laplaciano

2.5 Istogrammi e modelli di rumore

La distribuzione dei livelli di grigio può essere valutata misurandola con un istogramma. L'istogramma di un'immagine f si ottiene partizionando \mathbb{R} in intervalli congruenti semiaperti di lunghezza $\Delta I > 0$

$$I_k = [k\Delta I, (k+1)\Delta I) \quad k \in \mathbb{Z}$$

e contando le occorrenze di f nei sottointervalli I_k

$$c_k = |\{(i, j) \in [0, N-1] \times [0, M-1] : f(i, j) \in I_k\}|$$

Un istogramma può essere interpretato come una stima della funzione densità di probabilità (pdf) di un processo casuale con densità p costante in ogni intervallo I_k con valore

$$p_k = \frac{c_k}{I_k MN} \quad k \in \mathbb{Z}$$

L'istogramma è uno strumento importante per la descrizione di una immagine ed è invariante per rotazioni e non dipende dalle dimensioni. Tali proprietà ci forniscono delle qualità che possono essere utilizzate ad esempio per riconoscere una immagine anche se ha subito delle trasformazioni. In pratica l'istogramma sembra essere un possibile descrittore di una immagine.

Vediamo ora come l'istogramma possa aiutare per capire la natura del rumore di un'immagine. Le principali sorgenti di rumore nelle immagini digitali emergono durante l'acquisizione delle immagini (digitalizzazione) e/o durante la trasmissione. Le performance dei sensori di acquisizione delle immagini sono affette da una varietà di fattori, come le condizioni ambientali durante l'acquisizione, e dalla qualità dei dispositivi di sensing. Per esempio, nell'acquisizione delle immagini con una CCD camera, i livelli di luce e la temperatura del sensore sono i fattori di maggiore importanza nel rumore della immagine risultante. Le immagini sono corrotte durante la trasmissione principalmente a causa della interferenza nel canale trasmittivo. Per esempio, una immagine trasmessa usando una rete wireless potrebbe essere corrotta da un fulmine o da un altro disturbo atmosferico.

Il modello di rumore più semplice è quello additivo, dove supponiamo

che l'immagine corrotta f^δ sia:

$$f^\delta = f + \delta$$

dove δ è una variabile casuale. Altri possibili modelli sono

$$f^\delta = f\delta$$

e

$$f^\delta = f + \delta|\nabla u|$$

o più generalmente

$$f^\delta = f(\delta)$$

Le seguenti sono le più comuni pdf che si possono trovare in applicazioni di image processing:

- Gaussiano (fig.2.6a):

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

- Rayleigh (fig.2.6b):

$$p(z) = \begin{cases} \frac{2}{b} e^{-\frac{(z-\mu)^2}{2\sigma^2}} & z \geq a \\ 0 & z < a \end{cases}$$

- Gamma (fig.2.6c):

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

- Esponenziale (fig.2.6d):

$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

- Uniforme (fig.2.6e)

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

- Impulsi (fig.2.6f)

Data un'immagine con presenza di rumore uno dei possibili approcci è quello di regolarizzarla con un filtro di media, un filtro mediano o con qualche altro filtro regolarizzante e poi, in base al modello di rumore scelto, si osservano gli istogrammi dell'immagine rumorosa e dell'immagine regolarizzata e si cerca di capire quale distribuzione casuale meglio approssima l'errore.

2.6 Analisi e sintesi

Il processo di convoluzione di un'immagine con un banco di filtri è detto di analisi. Un esempio di analisi di immagine mediante filtri di smoothing è la piramide Gaussiana.

$$P_G(u)_n = \begin{cases} u & n = 1 \\ S^\downarrow(G_\sigma * P_G(u)_{n-1}) & 1 < n \leq N \end{cases}$$

Una piramide di un'immagine è una collezione di rappresentazioni di quell'immagine a scale diverse. Tipicamente ogni strato della piramide ha una lunghezza che è la metà dello strato precedente e il doppio del successivo e il parametro della Gaussiana è scelto tra 1 e 2. Nel caso di una piramide Gaussiana, ad ogni strato viene applicato un filtro Gaussiano e il risultato viene poi sottocampionato per ottenere lo strato successivo. Il sottocampionamento nel caso di fattori di scala 1/2 può essere fatto prendendo i pixels con entrambe coordinate pari. L'analisi fatta in questa maniera ci permette così di analizzare meglio le informazioni contenute in un'immagine a diverse frequenze e a diverse scale di grandezza.

Una piramide Gaussiana è molto ridondante perché intuitivamente mantiene sempre le informazioni alle basse frequenze e rimuove via via sempre più informazione alle alte frequenze. Ogni strato della piramide Gaussiana, se supercampionato in modo opportuno, diventa una predizione approssimata dello strato precedente e la differenza tra lo strato effettivo e quello

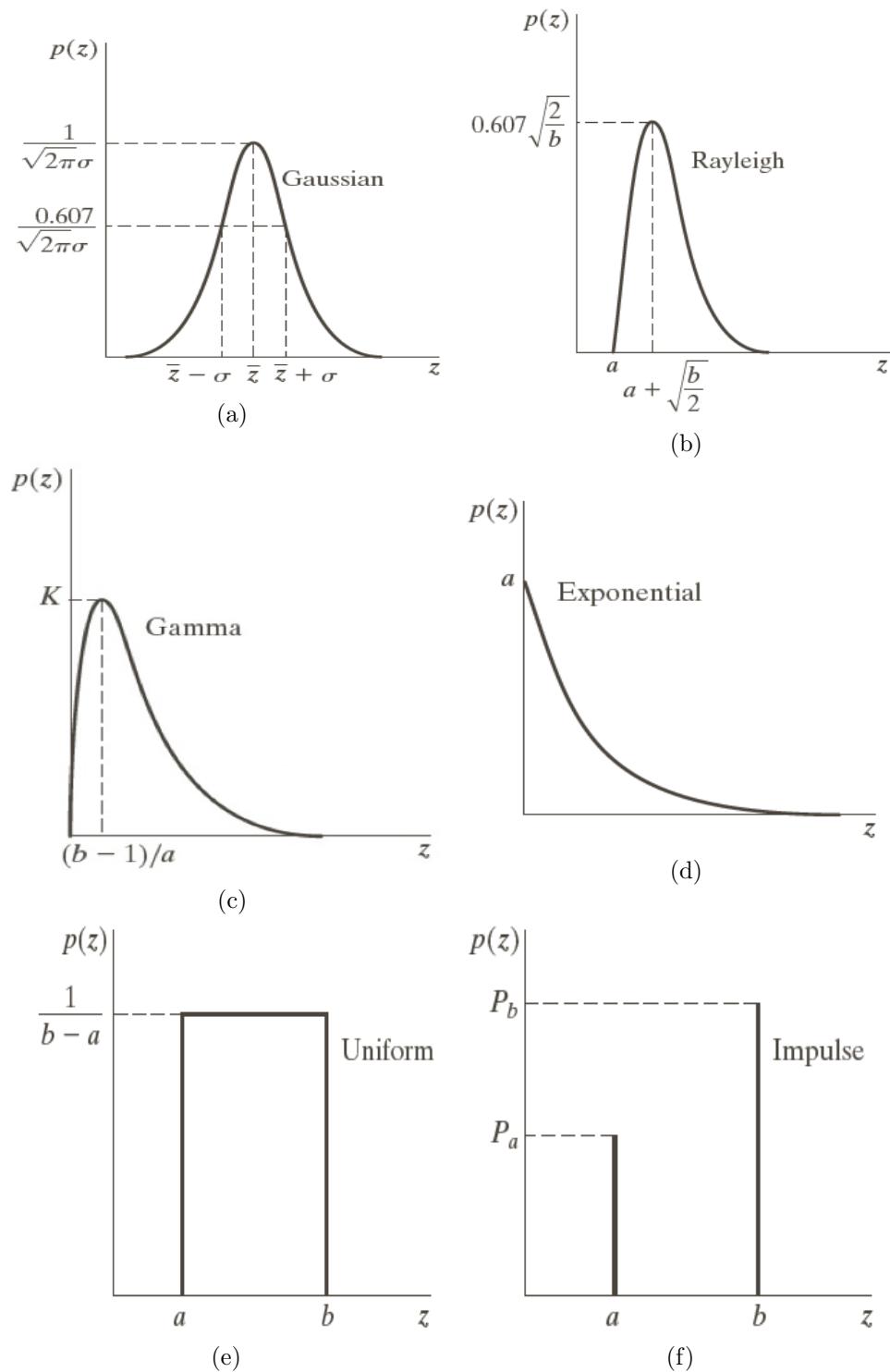


Figura 2.6: Grafici pdf distribuzioni casuali di errore

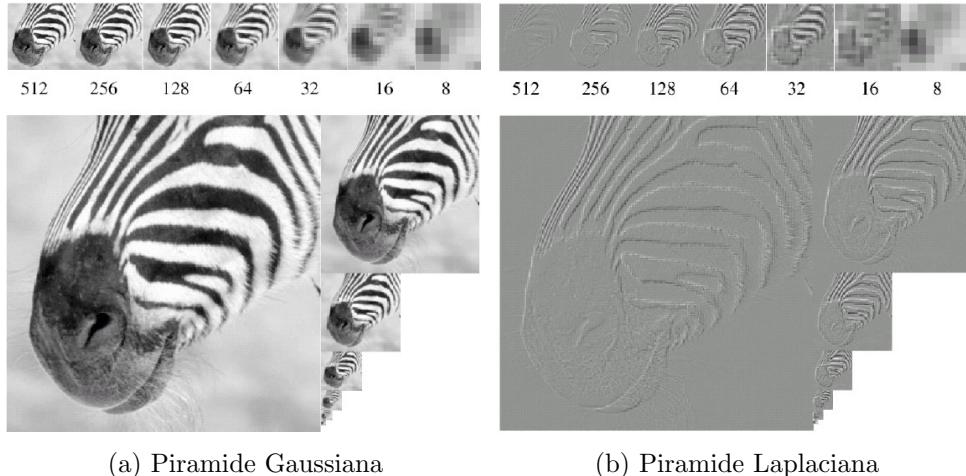


Figura 2.7: Esempio di analisi con piramide Gaussiana e Laplaciana

predetto produce un’immagine di errore. Prendendo le immagini di errore per ogni strato otteniamo una piramide detta Laplaciana. Il sovraccampionamento, nel caso di fattori di scala 1/2, trasforma ogni pixel in un quadrato 2x2.

$$P_L(u)_n = \begin{cases} P_G(u)_N & n = N \\ P_G(u)_n - S^\dagger(P_G(u)_{n+1}) & 1 < n \leq N \end{cases}$$

Il processo di sintesi ricostruisce un’immagine a partire dalla piramide Laplaciana. In particolare, dalla piramide Laplaciana possiamo ricostruire in senso inverso l’intera piramide Gaussiana, il cui strato ultimo è l’immagine originale.

$$P_G(u)_n = \begin{cases} P_L(u)_N & n = N \\ S^\dagger(P_L(u)_{n+1}) + P_L(u)_n & 1 < n \leq N \end{cases}$$

$$u = P_G(u)_1$$

Capitolo 3

Restauro di immagini digitali

3.1 Introduzione

Tra i restauratori museali ed in generale gli artisti del restauro, con il termine *inpainting* si indica la pratica del ritocco o del recupero di antichi dipinti danneggiati. Lo scopo è quello di rimuovere crepe o recuperare parti mancanti in maniera che il ritocco passi inosservato all'occhio umano. Il processo di inpainting consiste nel riempire le parti mancanti di un'immagine rovinata usando informazioni ottenute dai dintorni dell'area mancante o alterata. Si ha dunque la possibilità di restaurare immagini con parti danneggiate a causa di invecchiamento, umidità o altre condizioni ambientali, oppure graffi dovuti al deterioramento. Inoltre si può pensare di recuperare parti dell'immagine la cui visualizzazione viene ostruita da oggetti sovrapposti. Le applicazioni sono svariate: dal restauro di antichi affreschi, vecchi dipinti, fotografie, film alla riduzione degli artefatti nelle immagini mediche.

I metodi di inpainting si dividono principalmente in due categorie: metodi basati su EDP e metodi su sintesi di texture. Prima di entrare nel dettaglio di tali metodi facciamo una distinzione introduttiva ([13]):

- I primi solitamente sono progettati per connettere i contorni e/o prolungare le curve di livello in maniera adeguata all'interno del dominio di inpainting. Tali metodi sono mirati ad ottenere una soluzione all'interno del dominio estrapolando caratteristiche geometriche dell'immagine, in particolare i contorni. La maggior parte di tali metodi producono risultati insoddisfacenti se il dominio di inpainting è circon-



Figura 3.1: Confronto tra un immagine restaurata con sintesi di texture (destra) e PDE based method (centro). [13]

dato da regioni con texture pronunciate, anzichè zone con distribuzione del colore uniforme dove invece ottengono buoni risultati.

- Gli algoritmi di sintesi di texture invece usano samples dei dati dell’immagine disponibili e puntano a riempire il dominio di inpainting in maniera tale che, opportune statistiche tra pixels vicini, corrispondano con quelle del sample. Lo scopo di questi metodi è dunque quello di ricreare dettagli all’interno della regione di inpainting.

In fig.3.1 possiamo vedere come, nel caso di larghi domini di inpainting circondati da parti con texture pronunciate, il metodo basate sulle EDP si limiti semplicemente a sfocare mentre la sintesi di texture raggiunga risultati soddisfacenti.

3.2 Metodi basati su EDP

L’inpainting è essenzialmente un problema di interpolazione. Ciò che rende questo problema un problema assolutamente non banale è la complessità delle funzioni che rappresentano i valori di intensità di un’immagine. Un valido modello per immagini prive di texture viene fornito dallo spazio BV, ossia:

Definizione Data una funzione u appartenente a $L^1(\Omega)$ la variazione totale di u in Ω è definita come:

$$V(u, \Omega) := \sup \left\{ \int_{\Omega} u(x) \operatorname{div} \phi(x) dx : \phi \in C_c^1(\Omega, \mathbb{R}^n), \|\phi\|_{L^\infty(\Omega)} \leq 1 \right\}$$

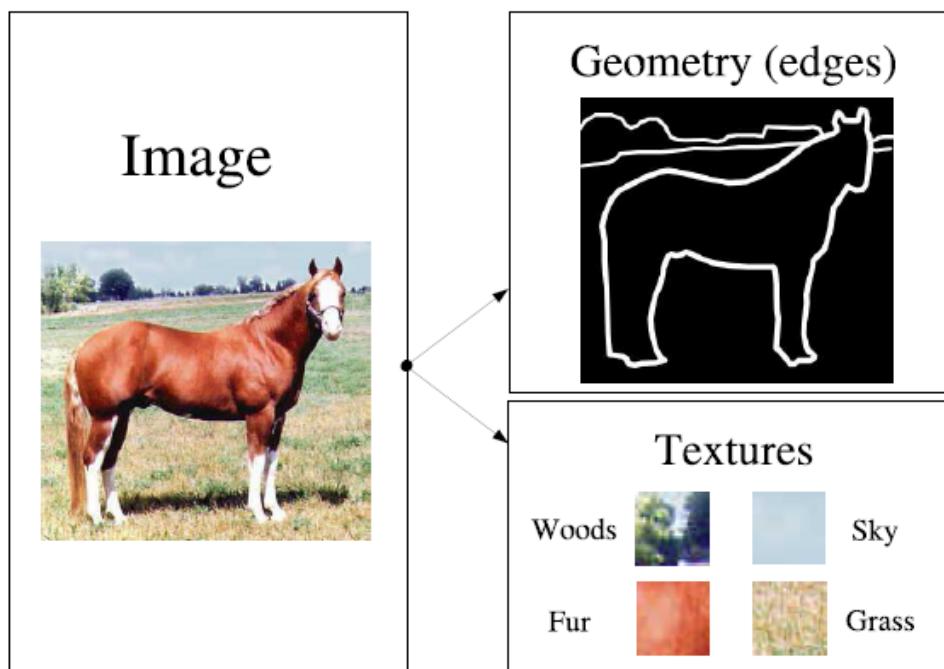


Figura 3.2: Ogni immagine si può scomporre in parte strutturale (geometrica) e texture, i metodi basati sulle EDP funzionano bene per la parte strutturale, mentre i metodi di sintesi per la parte di texture

Lo spazio delle funzioni a variazione limitata (BV) può dunque essere definito come:

$$BV(\Omega) = \{u \in L^1(\Omega) : V(u, \Omega) < +\infty\}$$

All'interno di tale spazio funzionale viene dunque concesso alle immagini di possedere una delle loro caratteristiche visuali più importanti: gli *edges* (contorni), ossia delle discontinuità nella distribuzione di intensità dell'immagine. In letteratura non sono presenti molti lavori riguardo l'approssimazione in BV , per due ragioni principalmente: il primo è che raramente questo tipo di problemi sorgono al di fuori del trattamento delle immagini, mentre il secondo è che tali problemi sono fortemente mal posti. La seconda ragione si comprende meglio con un semplice problema 1-D. Immaginiamo di avere i valori (o anche le derivate) di una funzione f nei punti $a - h$ e $a + h$. Se f è regolare, allora per $h \rightarrow 0$, possiamo applicare interpolanti regolari come quello di Lagrange o di Hermite per dedurre i valori f su $(a-h, a+h)$ con un certo livello di precisione garantito. Ma per una funzione $f \in BV$, tutti gli interpolanti regolari non riescono a funzionare correttamente, indipendentemente da quanto h sia piccolo, in quanto anche punti singoli possono avere massa non nulla e provocare salti all'interno dell'intervallo. Fortunatamente le immagini non sono così tanto intrattabili. Ogni immagine è una proiezione 2-D di una finestra del nostro mondo 3-D, dove ogni oggetto, spesso, presenta regolarità geometriche o regolarità nel riflettere la luce. Queste regolarità diminuiscono la malposizione del problema. Data un'immagine, se la copriamo con un pezzo di carta di dimensioni moderate, e chiediamo ad una persona di indovinare cosa c'è dietro nell'immagine originale, quasi certamente, chiunque fornirà una risposta razionale. Per esempio, se una mela verde viene coperta parzialmente da un pezzo di carta, allora uno tende prima a stimare i contorni incompleti, e poi a riempire i contorni di verde all'interno. Tutte queste decisioni vengono effettuate congetture che, scientificamente parlando, prendono il nome di *Inferenza Bayesiana*. I due ingredienti principali dell'inferenza Bayesiana sono il modello *a priori* e il modello *dei dati*. Il secondo nel nostro caso è semplice: i dati disponibili non sono nient'altro che parte di un'immagine completa che voglia restaurare. Dunque il modello a priori gioca un ruolo cruciale nel processo decisionale. Per l'esperimento sopracitato, la conoscenza a priori della forma ed il colore

aiuta una persona ad effettuare un ragionevole inpainting. Al fine di sviluppare un modello di inpainting che si applichi a problemi generali, bisogna trovare il giusto compromesso tra la necessità di rendere l'algoritmo il più generale possibile, ed il fatto che maggiore è la conoscenza a priori e minore sarà la cattivo condizionamento del problema [8].

In termini matematici, l'inpaiting di un immagine può essere descritto nella seguente maniera [10]: sia u_0 una data immagine definita sul dominio Ω . Il problema consiste nel ricostruire l'immagine originale u nel dominio danneggiato $D \subset \Omega$ che prende il nome d dominio di inpaiting. Più precisamente, sia $\Omega \subset \mathbb{R}^2$ un dominio aperto e limitato con frontiera Lipschitziana, B_1, B_2 due spazi di Banach ed $u_0 \in B_1$. Un generico approccio variazionale al problema di inpaiting può essere scritto come

$$\min_{u \in B_2} E(u) = \min_{u \in B_2} \left(R(u) + \frac{\lambda}{2} \|u_0 - u\|_{B_1}^2 \right) \quad (3.1)$$

dove $R : B_2 \rightarrow \mathbb{R}$ e

$$\lambda(x) = \begin{cases} \lambda_0 & \Omega \setminus D \\ 0 & D \end{cases} \quad (3.2)$$

è la funzione caratteristica di $\Omega \setminus D$ moltiplicato per la costante $\lambda_0 \gg 1$. $R(u)$ denota il termine di regolarizzazione, mentre il termine $\|\lambda(u_0 - u)\|_{B_1}$ è il termine di *fedeltà* del nostro algoritmo di inpaiting. Scrivendo le equazioni di Eulero-Lagrange per il problema (3.1) otteniamo

$$\partial_t u(t, x) = \lambda(x)(u_0(x) - u(t, x)) + \partial_u R(u) \quad (3.3)$$

dove $\partial_u R(u)$ indica la derivata di Frechét di $R(u)$ e con $u(t = 0, \cdot) = u_0(\cdot)$ in Ω più opprtune condizioni al bordo. In generale $B_2 \subseteq B_1$, indicativo dell'effetto di *smoothing* del termine di regolarizzazione sull'estremante $u \in B_2$.

In funzione della scelta del termine R e della scelta degli spazi B_1 e B_2 , sono stati sviluppati svariati approcci di inpaiting. Ovviamente non ci aspettiamo che il problema di minimo 3.1 ammetta soluzione unica, in quanto il problema di inpaiting stesso non può avere soluzione unica. Vediamo un esempio: sia u_0 l'immagine di figura 3.3 l'immagine da restaurare sul dominio dato dall'intersezione del rettangolo bianco e del rettangolo nero. Dal punto di vista della percezione umana è una situazione incerta, che

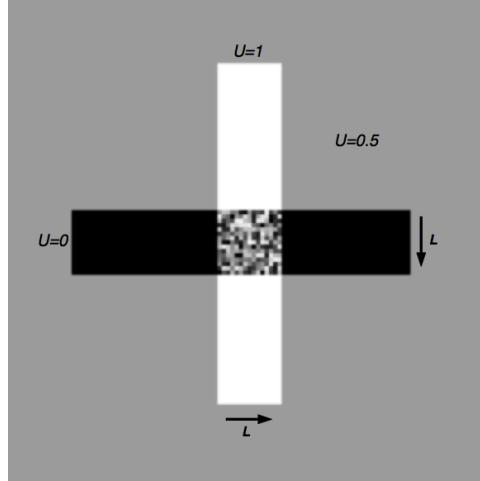


Figura 3.3: Esempio di non unicità del restauro

deve venir dunque riflessa dalla non unicità delle soluzioni dei metodi di inpainting.

Vediamo ora alcuni tra gli svariati metodi presenti in letteratura.

3.2.1 Il modello di TV inpainting di Chan e Shen

Uno dei primi modelli sviluppati è quella della variazione totale (TV) [8], dove

$$R(u) = \int_{\Omega} |\nabla u| \, dx$$

denota la variazione totale di u , con $B_1 = L^2(\Omega)$ e $B_2 = BV(\Omega)$ con condizioni al bordo di Dirichlet

$$u = u_0 \quad \text{su} \quad \partial D$$

Data $u_0 \in BV(\Omega)$, che assume valori compresi tra 0 e 1, è possibile dimostrare l'esistenza di un inpainting ottimale secondo il funzionale della variazione totale. Nel caso della presenza di rumore nell'immagine su $\Omega \setminus D$, Chan e Shen hanno proposto una versione modificata del metodo dove la condizione al contorno diventa:

$$\frac{1}{\text{Area}(\Omega \setminus D)} \int_{\Omega \setminus D} (u - u_0)^2 = \sigma^2$$

dove σ^2 è la varianza del rumore, stimabile da u_0 su $\Omega \setminus D$. Con tale vincolo stiamo assumendo che

$$u_0(x) = u_c(x) + e(x)$$

con $e(x)$ rumore indipendente dalla soluzione “pulita” u_c . Anche per questa versione si può dimostrare l’esistenza di $u_c \in BV(\Omega)$. L’equazione di Eulero-Lagrange diventa

$$\partial_t u = \lambda(u_o - u) - \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$$

3.2.2 Euler’s elastica

Il metodo che vedremo ora si basa su un modello a priori per curve piane. Introduciamo dunque tale modello ([8]).

Definizione: una curva Γ si dice curva elastica di Eulero se risulta essere una curva di equilibrio per il seguente funzionale di energia elastica:

$$E_2[y] = \int_{\gamma} (a + bk^2) ds$$

dove ds denota la lunghezza dell’elemento d’arco, $k(s)$ la curvatura scalare, e a e b due costanti positive. Ulteriori possibili vincoli potrebbero includere le posizioni e le direzioni normali ai due estremi. Tramite calcolo delle variazioni, si dimostra dalla formula del funzionale dell’energia che una curva elastica deve soddisfare l’equazione

$$2k''(s) + k^3(s) = \frac{a}{b}k(s)$$

Vediamo ora il fondamento logico che lega le capacità interpolative delle curve elastica ed il loro possibile uso nell’inpainting.

L’idea di base: Consideriamo la camminata casuale di un ubriaco partendo dall’origine del piano ed assumiamo che ogni passo sia dritto. Fissato un intero N , proviamo a capire la possibile distribuzione di tutte i possibili percorsi di N passi. Sappiamo che:

- (a) Siano h_k le dimensioni del passo k -esimo, allora $\{h_k : k = 1, 2, \dots, N\}$ sono variabili i.i.d. di tipo esponenziale $\lambda \exp(-h\lambda)$ con media positiva $\frac{1}{\lambda}$

- (b) Sia θ_k l'orientamento del passo k -esimo, dato dall'angolo formato dalla direzione k -esima e l'asse x , con ovviamente $\theta_0 = 0$. Sia $\Delta\theta_k = \theta_k - \theta_{k-1}$ ($k = 1, 2, \dots, N$) la svolta fatta al passo k -esimo. L'ipotesi di base è quella che, ad ogni passo k , maggiore è il passo h_k maggiore sarà l'incertezza sul valore di $\Delta\theta_k$. Precisamente, $\Delta\theta_k$ è una Gaussiana $N(0, h_k\sigma)$. Inoltre $\{n_k = \frac{\Delta\theta_k}{h_k} : k = 1, 2, \dots, N\}$ è un insieme di variabili indipendenti e indipendenti da h_k .

Dunque, una camminata poligonale di N -passi γ viene completamente determinata una volta noti i dati

$$\{h_k : k = 1, 2, \dots, N\} \cup \{\Delta\theta_k : k = 1, 2, \dots, N\}$$

con la verosimiglianza data da

$$\begin{aligned} & \lambda^N \exp(-\lambda(h_1 + \dots + h_N)) dh_1 \dots dh_N \\ & \times (\sqrt{2\pi}\sigma)^{-N} \exp\left(-\frac{1}{2\sigma^2}[(\Delta\theta_1/h_1)^2 + \dots + (\Delta\theta_N/h_N)^2]\right) dn_1 \dots dn_N \end{aligned}$$

Integrando, a meno di costante moltiplicativa, otteniamo la densità di probabilità

$$\exp\left(-\lambda L(\gamma) - \frac{1}{2\sigma^2} \|\kappa^2\|_\gamma^2\right)$$

con L che indica la lunghezza e $\|\kappa^2\|_\gamma$ che indica l'analogo discreto di $\int_\gamma \kappa^2 ds$. Di conseguenza la minimizzazione dell'energia elastica con $a = \lambda$ e $b = \frac{1}{2\sigma^2}$ equivale alla stima della massima verosimiglianza di tali curve casuali. Questo è dunque l'aspetto di inferenza Bayesiana presente dietro le curve elastiche, che razionalizza l'uso dell'elastica nell'inpainting delle immagini.

La costruzione del funzionale: Consideriamo ora di interpolare con una curva elastica tutti le curve isofote $\Gamma_\lambda = \{x \in \Omega : u_0(x) = \lambda\}$, una per una, partendo dalle informazioni sul bordo. La traccia di ogni Γ_λ sulla frontiera di D , ci dice come accoppiare i punti sul bordo. Siano $p_1, p_2 \in \partial\Omega$ punti con lo stesso livello di grigio λ , con le normali computate dai dati dell'immagine disponibili all'esterno di D , \vec{n}_1 e \vec{n}_2 . Allora interpoliamo la λ -isofota con l'elastica Γ'_λ :

$$\Gamma'_\lambda = \operatorname{argmin}_{\gamma_\lambda} \int_{\gamma_\lambda} (a + bk^2) ds = \operatorname{argmin}_{\gamma_\lambda} E_2[\gamma_\lambda]$$

dove γ_λ passa per p_1 e p_2 , e $\dot{\gamma}_\lambda \perp \vec{n}_i$ con $i = 1, 2$. Al variare di λ lungo il bordo (supponiamo per comodità $0 \leq \lambda \leq 1$), otteniamo una famiglia di elastiche

$$\mathcal{F}' = \{\Gamma'_\lambda : 0 \leq \lambda \leq 1\}$$

si vede allora che \mathcal{F}' minimizza il seguente funzionale dell'energia :

$$E[\mathcal{F}] = \int_0^1 E_2[\gamma_\lambda] d\lambda$$

per tutti gli insiemi di curve $\mathcal{F} = \{\gamma_\lambda : 0 \leq \lambda \leq 1\}$ che siano compatibili con le condizioni al bordo, o più generalmente

$$E_w[\mathcal{F}] = \int_0^1 w(\lambda) E_2[\gamma_\lambda] d\lambda$$

con w funzione di pesi positivi. Sia $u = u_D$ un inpainting ammissibile, allora lungo le isofote la curvatura sarà data da

$$\kappa = \nabla \cdot \vec{n} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$$

sia dt l'incremento infinitesimo di u lungo la direzione normale n allora avremo

$$\frac{d\lambda}{dt} = |\nabla u| \quad \text{or} \quad d\lambda = |\nabla u| dt$$

di conseguenza

$$\begin{aligned} R(u) &= E_w[\mathcal{F}] \\ &= \int_0^1 w(\lambda) \int_{\gamma_\lambda:u=\lambda} (a + b\kappa^2) ds d\lambda \\ &= \int_1^0 \int_{\gamma_\lambda:u=\lambda} w(u) \left(a + b \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \right) |\nabla u| dt ds \\ &= \int_D w(u) \left(a + b \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \right) |\nabla u| dx \end{aligned}$$

Ora l'energia è dunque espressa in funzione della sola u . La funzione $w(u)$ può essere fissata a 1, anche se spesso viene definita in funzione dell'isto-

gramma $h(\lambda)$ dell'immagine. Come condizioni al contorno imponiamo

$$\begin{aligned} u &= u_0 \quad \text{su } \partial D \\ |\kappa(p)| &< \infty \quad \text{su } \partial D \\ \int_{\partial D} |\nabla u| &= 0 \end{aligned}$$

L'ultima condizione deriva naturalmente dal fatto che abbiamo ipotizzato che $u_0 \in BV(\Omega)$ e che

$$\int_{\partial D} |\nabla u_0| = 0$$

Purtroppo, a differenza del metodo TV, per il metodo dell'elastica è difficile studiare il problema dell'esistenza di una soluzione per via della non convessità del funzionale dell'energia. L'equazione di Eulero-Lagrange per questo metodo si dimostra essere:

$$\begin{aligned} \partial_t u &= \lambda(u_o - u) - \nabla \cdot V \\ V &= (a + b\kappa^2)\vec{n} - \frac{2b}{|\nabla u|} \frac{\partial k|\nabla u|}{\partial \vec{t}} \vec{t} \end{aligned}$$

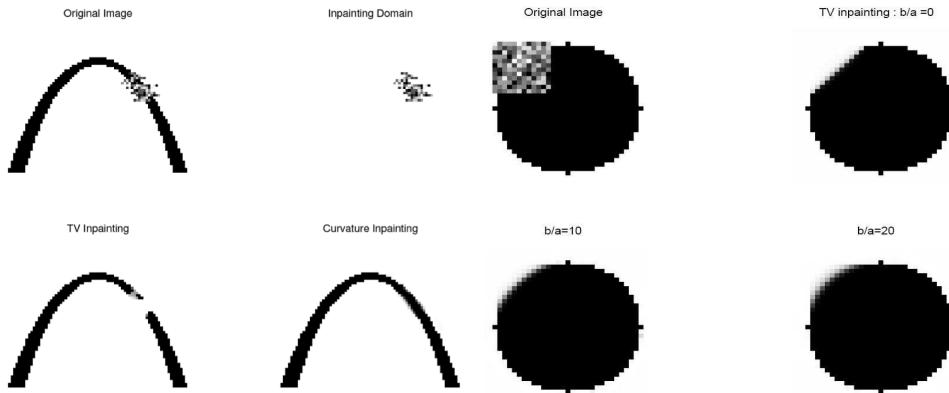
Osserviamo infine che se $b/a = 0$ il metodo dell'elastica equivale al metodo TV.

3.2.3 Confronto tra metodo TV ed metodo dell'Elastica

Metodi del secondo ordine come il metodo TV presentano degli inconvenienti come l'incapacità di estendere gli edges all'interno del dominio e l'incapacità di propagare *smoothly* le curve di livello dell'immagine sempre all'interno del dominio. Questi due problemi fanno riferimento a due principi importanti dell'inpainting, rispettivamente il *principio di connettività* e il *principio di conservazione della curvatura*.

La ragione di questo problemi va cercata nel fatto che la regolarizzazione di ordine due, penalizzando la lunghezza delle curve di livello, connette quest'ultime partendo dalla frontiera del dominio di inpainting lungo la traiettoria più breve, interpolando linearmente. Se dunque $R(u) = \int_{\Omega} |\nabla u| dx$ (TV), allora utilizzando la formula della coarea avremo

$$\min_u \int_{\Omega} |\nabla u| dx \iff \min_{\Gamma_\lambda} \int_{-\infty}^{\infty} \text{length}(\Gamma_\lambda) d\lambda \quad (3.4)$$



(a) *Principio di connettività*: nel caso di domini di inpainting larghi, TV non riesce a prolungare i contorni all'interno del dominio
(b) *Principio di conservazione della curvatura*: TV tronca il cerchio mentre nel caso dell'Elastica abbiamo un inpainting regolare che tiene conto della curvatura

Figura 3.4: TV inpainting e Metodo della curva elastica a confronto

dove $\Gamma_\lambda = \{x \in \Omega : u(x) = \lambda\}$ è la curva di livello per l'intensità λ . Se consideriamo invece un metodo di ordine maggiore di due come il metodo dell'equazione elastica di Eulero abbiamo

$$\min_u \int_{\Omega} (a + bk^2)|\nabla u| dx \iff \min_{\Gamma_\lambda} \int_{-\infty}^{\infty} a \text{length}(\Gamma_\lambda) + b \text{curvature}^2(\Gamma_\lambda) d\lambda$$

dunque non solo la lunghezza delle curve di livello viene penalizzata ma anche la loro curvatura (dove la penalizzazione di ciascuno dei due dipende dalla rapporto b/a). Il risultato è una continuazione regolare delle curve di livello nel dominio di inpainting. Un'altra interpretazione possibile per le performance migliori nel caso generale dei metodi di ordine maggiore di due, è quella che per avere un problema ben posto, nel caso TV si impongono solo condizioni al bordo di Dirichlet, mentre nel caso dell'Elastica si impongono condizioni anche sul gradiente della funzione sul bordo [10].

3.2.4 Metodo di Bertalmio et al.

L'approccio più naturale per risolvere un problema di inpainting è quello di mimare come i restauratori professionisti effettuano l'inpaint manualmente.

Come fa presente Bertalmio in [11], quando degli esperti in materia del Minneapolis Institute of Arts sono stati consultati riguardo il loro *modus operandi* essi hanno sottolineato i seguenti punti fondamentali:

- Non vi è assolutamente una maniera unica per risolvere il problema.
- L'immagine, nel suo intero, determina come riempire il vuoto nel dominio di inpainting, avendo come scopo quello di restaurare l'armonia dell'immagine.
- La struttura dell'area circondante D viene continuata all'interno del gap, e le curve di livello vengono disegnate lungo i prolungamenti delle curve di livello che toccano il bordo del dominio.
- Le diverse regioni all'interno di D , come definite dalle isolinee, vengono riempite con il colore in accordo con le isolinee su ∂D .
- I piccoli dettagli vengono disegnati (e.g piccoli punti bianchi su un cielo altrimenti uniformemente blu), in altre parole le texture vengono aggiunte.

Vediamo ora come tradurre questo approccio in termini matematici. Le curve di livello dell'immagine $u(x, y) = c$ con u intensità di grigio, per il teorema del Dini, avranno come vettore tangente $\nabla^\perp u$, che sappiamo essere la direzione di minima crescita di u . Assumiamo allora che la direzione di estensione dei contorni sia data da $\nabla^\perp u$ sul bordo del dominio di inpainting. Successivamente prendiamo come indice di *smoothness*

$$\Delta u$$

ossia il Laplaciano. Ricordiamo che, $|\Delta u|$ assume valori significativi per zone ad alta varianza, mentre assume valori trascurabili per regioni regolari. Δu permette dunque di estrapolare i contorni ed il rumore di un'immagine come abbiamo visto nella sezione 2.4.

Ergo, tendendo conto delle osservazioni elencate in precedenza, decidiamo di propagare Δu lungo $\nabla^\perp u$ dal bordo del dominio di inpainting. Lo schema risultante è il seguente

$$u_t = \nabla^\perp u \cdot \nabla \Delta u$$

oppure, nel caso si voglia aggiungere della diffusione,

$$u_t = \nabla^\perp u \cdot \nabla \Delta u + \nu \nabla \cdot (g(|\nabla u|) \nabla u)$$

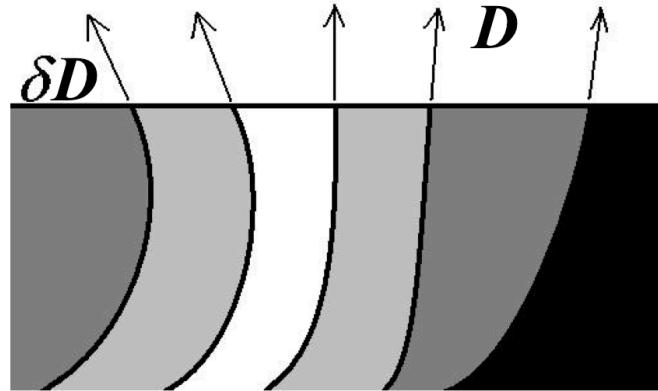


Figura 3.5: *Il problema di inpainting secondo l'approccio di Bertalmio, et al.*. Il laplaciano viene propagato lungo le direzioni delle isolinee. (Le curve verticali rappresentano $\nabla^\perp u$)

in quanto lo scopo è quello di evolvere verso la soluzione stazionaria dove le curve di livello di u e quelle di Δu sono parallele. Quando tutta l'informazione viene propagata la soluzione del nostro problema di inpaiting u^* sarà t.c.

$$\nabla^\perp u^* \cdot \nabla \Delta u^* = 0$$

Le condizioni al bordo da accoppiare alla soluzioni saranno

$$\begin{cases} u^* = u_{orig} & \partial D \\ \nabla^\perp u^* = \nabla^\perp u_{orig} & \partial D \end{cases}$$

Analogia tra fluidi e immagini Analizziamo ora l'analogia tra la dinamica dei fluidi e l'inpainting delle immagini secondo Bertalmio [12].

Consideriamo un fluido viscoso incompressibile. Le equazioni che ne governano la dinamica sono le equazioni di Navier-Stokes:

$$\begin{cases} v_t + v \cdot \nabla v - \nu \Delta v + \nabla p = 0 & D \\ \nabla \cdot v = 0 & D \end{cases}$$

con opportune condizioni al bordo, dove $v = [v_1, v_2]$ è la velocità, p è la pressione e ν è la viscosità cinematica. Nel caso bidimensionale dal vincolo di incompressibilità si dimostra l'esistenza di una funzione di flusso Ψ che soddisfa $v = \nabla^\perp \Psi$. L'equazione di conservazione della quantità di moto può essere riscritta in termini di vorticità, ossia applicando il rotore ad ambo i

mebri. Otteniamo così:

$$\omega_t + v \cdot \nabla \omega - \nu \Delta \omega = 0$$

Riscrivendo la vorticità in termini di Ψ abbiamo

$$\omega = \nabla \times v = \nabla \times \nabla^\perp \Psi = \Delta \Psi$$

che sostituita all'equazione del momento della quantità di moto stazionaria ($\omega_t = 0$) diventa:

$$v \cdot \nabla \omega - \nu \Delta \omega = \nabla^\perp \Psi \cdot \nabla \Delta \Psi - \nu \Delta \Delta \Psi = 0$$

che nel caso non viscoso diventa

$$\nabla^\perp \Psi \cdot \nabla \Delta \Psi = 0$$

ergo, si ha equilibrio quando le curve di livello della vorticità coincidono con quelle della funzione di flusso.

L'analogia è ora chiara: la soluzione di inpainting ottimale e la soluzione di flusso 2D stazionario inviscido soddisfano la stessa equazione.

Fluidi	Immagini
funzione di flusso Ψ	intensità immagine u
velocità del fluido $v = \nabla^\perp \Psi$	direzione isolinee di u , $\nabla^\perp u$
vorticità $\omega = \Delta \Psi$	smoothness $\omega = \Delta u$
viscosità ν	diffusività ν

Tabella 3.1: Tabella riassuntiva analogia tra le grandezze caratteristiche dei fluidi e delle immagini

Pertanto, al fine di risolvere il problema di Bertalmio, dobbiamo trovare la soluzione stazionaria per il problema inviscido, per il quale problema abbiamo un'ottima conoscenza del quadro teorico e numerico. Ovviamente è possibile riscrivere il problema in termini della velocità ($v = \nabla^\perp u$, campo di vettori tangenti alle isolinee):

$$\begin{cases} v \cdot \nabla v + \nabla p = \nu \nabla \cdot (g(|\nabla v|) \nabla v) & D \\ \nabla \cdot v = 0 & D \\ v = \nabla^\perp u_{orig} & \partial D \end{cases}$$

per poi ottenere u risolvendo un problema di Poisson:

$$\begin{cases} \Delta u = \nabla \times v & D \\ u = u_{orig} & \partial D \end{cases}$$

con la velocità calcolata al punto precedente.

Per quanto riguarda la diffusività ν la scegliamo di modo che in presenza di forti gradienti di u sia trascurabile in modo tale da risaltare gli edges. Una scelta possibile potrebbe essere

$$\nu(\|\nabla^\perp u\|) = \frac{1}{1 + \frac{\|\nabla^\perp u\|}{k}}$$

con k parametro di tuning.

3.2.5 Una teoria unificante

Vediamo ora come Chan, Kang e Shen nell'articolo [8] mostrano come i tre metodi di inpainting visti fin'ora possano essere visti come istanze di un singolo metodo.

Assumiamo l'immagine da restaurare abbastanza regolare (almeno appartenente a $W^{1,2}$) con la curvatura ben definita. Ipotizziamo inoltre che la variazione δu abbia supporto compatto, in modo tale che nelle integrazioni per parti spariscano i termini di bordo. Sia $\phi \in C^1(R, [0, \infty))$ funzione fissata, e sia

$$R[u] = \int_D \phi(\kappa) |\nabla u| dx.$$

il termine regolarizzante del metodo dell'Elastica di Eulero generalizzato. Allora la prima variazione è data da

$$\frac{\partial R}{\partial u} = -\nabla \cdot \vec{V}$$

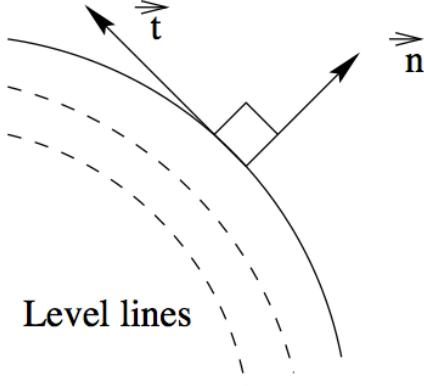


Figura 3.6: Vettore tangente e vettore normale lungo le curve di livello

con

$$\vec{V} = \phi(\kappa) \vec{n} - \frac{\vec{t}}{|\nabla u|} \frac{\partial(\phi'(\kappa) |\nabla u|)}{\partial \vec{t}}$$

Dove $\vec{n} = \frac{\nabla u}{|\nabla u|}$ è il campo normale ascendente, mentre \vec{t} è il campo tangente (la cui esatta orientazione non importa vista la parità di \vec{t} nell'espressione.) (fig. 3.6)

Dimostrazione: Per comodità di scrittura poniamo $\langle f \rangle = \int_D f dx$.

Allora:

$$\begin{aligned} \delta R &= \langle \delta |\nabla u| \phi(\kappa) \rangle \\ &= \langle \phi(\kappa) \delta |\nabla u| \rangle + \langle |\nabla u| \delta \phi(\kappa) \rangle \\ &= \langle \phi(\kappa) \frac{\nabla u}{|\nabla u|} \rangle + \langle \phi'(\kappa) \rangle \\ &= -\langle \nabla \cdot [\phi(\kappa) \vec{n}] \delta u \rangle + \langle \phi'(\kappa) |\nabla u| \delta \kappa \rangle. \end{aligned}$$

Sviluppiamo dunque il termine di variazione della curvatura:

$$\begin{aligned} \delta \kappa &= \delta \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right) = \nabla \cdot \left[\frac{1}{|\nabla u|} \nabla (\delta u) + \nabla u \delta \left(\frac{1}{|\nabla u|} \right) \right] \\ &= \nabla \cdot \left[\frac{1}{|\nabla u|} \nabla (\delta u) - \left(\nabla u \frac{\nabla u}{|\nabla u|^3} \right) \nabla (\delta u) \right] \\ &= \nabla \cdot \left[\frac{1}{|\nabla u|} \{I - \vec{n} \otimes \vec{n}\} \nabla (\delta u) \right]. \end{aligned}$$

Con I che indica la trasformazione identità e $P = \vec{n} \otimes \vec{n}$ che indica la proie-

zione ortogonale nella direzione normale. Ne consegue che:

$$\begin{aligned}
 \langle \phi'(\kappa) |\nabla u| \delta \kappa \rangle &= \langle \phi'(\kappa) |\nabla u| \nabla \cdot \left[\frac{1}{|\nabla u|} \{I - \vec{n} \otimes \vec{n}\} \nabla (\delta u) \right] \rangle \\
 &= \langle -\nabla \left(\phi'(\kappa) |\nabla u| \right) \left[\frac{1}{|\nabla u|} \{I - \vec{n} \otimes \vec{n}\} \nabla (\delta u) \right] \rangle \\
 &= -\{I - \vec{n} \otimes \vec{n}\} \left\{ \frac{1}{|\nabla u|} \nabla \left(\phi'(\kappa) |\nabla u| \right) \right\} \nabla (\delta u) \rangle \\
 &= \langle \nabla \cdot \left[(I - \vec{n} \otimes \vec{n}) \left\{ \frac{1}{|\nabla u|} \nabla \left(\phi'(\kappa) |\nabla u| \right) \right\} \right] \delta u \rangle.
 \end{aligned}$$

sfruttando così la simmetria di $I - P$. Ne consegue che:

$$\begin{aligned}
 \frac{\partial R}{\partial u} &= -\nabla \cdot \left[\phi(\kappa) \vec{n} - \frac{1}{|\nabla u|} (I - \vec{n} \otimes \vec{n}) \nabla \left(\phi'(\kappa) |\nabla u| \right) \right] \\
 &= -\nabla \cdot \left[\phi(\kappa) \vec{n} - \frac{1}{|\nabla u|} \frac{\partial (\phi'(\kappa) |\nabla u|)}{\partial \vec{t}} \vec{t} \right] \\
 &= -\nabla \cdot \vec{V}.
 \end{aligned}$$

Dove abbiamo usato il fatto che $I = \vec{n} \otimes \vec{n} + \vec{t} \otimes \vec{t}$ e $\vec{t} \otimes \vec{t} \nabla f = \vec{t} \frac{\partial f}{\partial \vec{t}}$

Un altro risultato importante prima di vedere le analogie tra i tre metodi è la seguente proposizione: Data una trasformazione morfologica regolare di u :

$$u \rightarrow g(u), \quad g'(u) > 0$$

allora $\vec{V}_{g(u)} = \vec{V}_u$, ossia il flusso \vec{V} è morfologicamente invariante.

Dimostrazione:

Ricordando che κ, \vec{n} e \vec{t} sono morfologicamente invarianti, avremo

$$\begin{aligned}
 \frac{1}{|\nabla g(u)|} \frac{(\partial \phi'(\kappa) |\nabla g(u)|)}{\partial \vec{t}} \vec{t} &= \frac{1}{g'(u) |\nabla u|} \frac{\partial (g'(u) \phi'(\kappa) |\nabla u|)}{\partial \vec{t}} \vec{t} \\
 &= \frac{g'(u)}{g'(u) |\nabla u|} \frac{\partial (\phi'(\kappa) |\nabla u|)}{\partial \vec{t}} \vec{t} \\
 &= \frac{1}{|\nabla u|} \frac{\partial (\phi'(\kappa) |\nabla u|)}{\partial \vec{t}} \vec{t}
 \end{aligned}$$

dove è stato applicato il risultato secondo cui sia u che $g'(u)$ sono nella stessa direzione tangente. Si può quindi concludere che $\vec{V}_{g(u)} = \vec{V}_u$

Consideriamo ora il metodo CDD (Curvature Driven Diffusion) svilup-

pato da Chan e Shen:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{g(\kappa)}{|\nabla u|} \nabla u \right) = \nabla \cdot (g(\kappa) \vec{n})$$

dove $g : R \rightarrow [0, +\infty)$ è una funzione continua con $g(0) = 0$ e $g(\pm\infty) = +\infty$.

Se prendiamo $g = 1$ torniamo ad avere il classico metodo TV con diffusione anisotropa (a meno del termine di *fedeltà* $\lambda(u - u_0)$). L'introduzione del termine $g(\kappa)$ ha lo scopo di penalizzare grandi curvature, ed incoraggiare quelle piccole, visto che il coefficiente d diffusione in questo caso è $D = \frac{g(\kappa)}{|\nabla u|}$. Il metodo di Bertalmio, invece, si può riscrivere in maniera più generale come

$$\frac{\partial u}{\partial t} = \nabla^\perp u \cdot \nabla L(u)$$

ricordando che $\nabla^\perp u = |\nabla u| \vec{t}$ e che L è un indice di smoothness. Abbiamo visto in precedenza che:

$$\begin{aligned} \vec{V}_n &= \phi(\kappa) \vec{n} \\ \vec{V}_t &= - \left(\frac{1}{|\nabla u|^2} \frac{\partial(\phi'(\kappa)|\nabla u|)}{\partial \vec{t}} \right) \nabla^\perp u \end{aligned}$$

dunque se prendiamo

$$g(\kappa) = \phi(\kappa)$$

Il flusso normale \vec{V}_n corrisponde esattamente a quello del metodo CDD. D'altra parte invece se prendiamo la divergenza della componente tangenziale abbiamo

$$\nabla \cdot \vec{V}_t = \nabla^\perp u \cdot \nabla \left(- \frac{1}{|\nabla u|^2} \frac{\partial(\phi'(\kappa)|\nabla u|)}{\partial \vec{t}} \right)$$

che coincide duqne con il metodo di Bertamio nel caso si decida di prendere come misura di smoothness

$$L_\phi = - \frac{1}{|\nabla u|^2} \frac{\partial(\phi'(\kappa)|\nabla u|)}{\partial \vec{t}}$$

sviluppando tale termine otteniamo:

$$L_\phi = - \frac{1}{|\nabla u|^2} \left(\phi''(\kappa) |\nabla u| \frac{\partial \kappa}{\partial \vec{t}} + \phi'(\kappa) [\nabla \otimes \nabla u](\vec{n}, \vec{t}) \right)$$

con $[\nabla \otimes \nabla u](\bullet, \bullet)$ forma bilineare Hessiana di u . Dunque in un caso

semplificato come $\phi(s) = |s|$ con $\kappa \neq 0$, abbiamo

$$L_\phi = \frac{\pm 1}{|\nabla u|^2} [\nabla \otimes \nabla u](\vec{n}, \vec{t})$$

che a meno di una costante moltiplicativa, assomiglia molto alla scelta del laplaciano, come misura di smoothness infatti:

$$\Delta u = \text{traccia}(\nabla \otimes \nabla u) = [\nabla \otimes \nabla u](\vec{n}, \vec{n}) + [\nabla \otimes \nabla u](\vec{t}, \vec{t})$$

In conclusione, il metodo generalizzato dell'Elastica, combina il sia il meccanismo del trasporto della smoothness *lungo* le isofote, sia il meccanismo di diffusione CDD *attraverso* le isofote.

3.3 Sintesi di texture

Come osservato in precedenza nel caso di regioni mancanti di grosse dimensioni, o per regioni con presenza di texture, i metodi basati sulle PDE non riescono a ricreare tali texture all'interno del dominio, limitandosi solamente a ricreare solo una soluzione offuscata in contrasto con il resto dell'immagine. In uno dei primi lavori apparsi in letteratura, Efros e Leung [14] definiscono texture come un pattern visuale in un infinito piano 2D che, ad una certa scala, presenta una distribuzione stazionaria. Dato dunque un campione finito di una certa texture, lo scopo è quello di sintetizzare (ricreare) altri campioni della stessa texture. Questo approccio è basato su un principio ai autosimilarità, dove si postula che un'immagine presenta diverse ripetizioni di informazioni. Senza ulteriori ipotesi il problema è chiaramente mal posto, in quanto il campione potrebbe anche essere preso da uno svariato numero di textures differenti. Assumiamo dunque che il campione sia abbastanza grande da catturare in qualche maniera la stazionarietà della texture, e che la scala (approssimata) dell'elemento di texture (*texel*) sia nota. Le textures, tradizionalmente, vengono classificate come regolari (texels che si ripetono) o stocastiche (senza texels espliciti). In realtà la maggior parte delle texture di immagini reali stanno nel mezzo di questi due estremi.

I metodi di sintesi di texture sono numerosi, ma principalmente si dividono in due categorie ([15]):

- Per le texture molto regolari, metodi che dato un piccolo campione di texture pura, cercano di replicarlo *ad infinitum*
- Per le texture con presenza di moderata stocasticità, metodi che dato un campione, cercano un altro campione il più simile possibile a tale campione, all'esterno del dominio di inpainting, e ne copiano i valori di intensità.

Presentiamo ora la notazione usata ed alcuni tra i metodi della seconda categoria.

Notazione: L'insieme dei pixels viene indicato con Ω , $u : \Omega \rightarrow [0, 255]$ rappresenta l'intensità di grigio dell'immagine. D è il dominio di inpainting con relativa maschera M

$$M(p) = \begin{cases} 0 & p \in \Omega \setminus D \\ 1 & p \in D \end{cases}$$

Indichiamo con Ψ_s la finestra (insieme quadrato di pixels) di dimensione $|\Psi_s| = (2s+1) \times (2s+1)$ centrato in 0: $\Psi_s = \{k = (k_1, k_2) \mid k_1, k_2 \in (-s, s)\}$, mentre $P_s(p)$ sarà la finestra centrata sul pixel p anziché l'origine. Infine definiamo una *mappa di corrispondenza* $\phi : \Omega \rightarrow \Omega \setminus D$ come:

$$\phi(p) = \begin{cases} p = (x, y) & \text{se } p = (x, y) \in D^c \\ q \in D^c & \text{se } p \in D \end{cases}$$

3.3.1 Efros, sintesi non parametrica

Nel metodo di Efros la texture viene modellata come un campo casuale di Markov, ossia assumendo che la distribuzione di probabilità dei valori di grigio dell'immagine per ogni pixel, noti i valori in un intorno di tale pixel, non dipende dal resto dell'immagine. L'intorno è una finestra quadrata, la cui dimensione è un parametro globale critico. L'input dell'algoritmo è un insieme di *patches* appartenenti a D^c , e lo scopo è quello di selezionare una patch appropriata per ciascuno dei pixels del dominio D . Questo viene fatto computando una distanza tra la parte nota dell'intorno del pixel appartenente al dominio (chiaramente partendo dalla frontiera) e ciascuna delle patches di input. Più precisamente, considerando la seguente definizione di distanza d :

$$d(p, q) = \sum_{k \in \Psi_s} \|u(p+k) - u(q+k)\|^2$$

Algoritmo di Efros
while $D \neq \emptyset$ do
1 - Selezione pixel p dal bordo ∂D
2 - Trovare $\hat{p} : \hat{p} = \arg \min_{q \in D^c} d^*(p, q)$
3 - Porre $u(p) = u(\hat{p})$, $M(p) = 1$, $D = D \setminus p$
end while

Tabella 3.2: Algoritmo di Efros

ne consegue che la ricerca del pixel \hat{p} candidato a sostituire il pixel $p \in D$ corrisponde nel risolvere il problema:

$$\hat{p} = \arg \min_{q \in D^c} d(p, q)$$

In pratica, solo la parte nota dell'intorno di p viene usata nel calcolo della distanza, di conseguenza avremo

$$\hat{p} = \arg \min_{q \in D^c} \sum_{k \in \Psi_s} M(p+k) \|u(p+k) - u(q+k)\|^2 \quad (3.5)$$

corrispondente alla nuova definizione di distanza

$$d^*(p, q) = \sum_{k \in \Psi_s} M(p+k) \|u(p+k) - u(q+k)\|^2$$

Al fine di avere sempre un intorno parzialmente noto, i pixels verrano sostituiti partendo dal bordo del dominio verso l'interno.

Vediamo dunque la schematizzazione di tale procedimento nella tabella 3.2. Notiamo che il problema di sintesi appena descritto consiste nel trovare per ogni pixel nel dominio D il valore di intensità che minimizza l'equazione 3.5. Tale problema è analogo a trovare la mappa di corrispondenza ϕ tale che

$$\phi(p) = \begin{cases} p = (x, y) & \text{se } p = (x, y) \in D^c \\ \arg \min_{q \in D^c} d^*(p, q) & \text{se } p \in D \end{cases}$$

Dalla descrizione appena fatta si vede che l'algoritmo di Efros è un algoritmo *greedy* ad un passo: una volta che un pixel $p \in D$ assume un valore, questo rimane invariato. Questa tecnica è soggetta a problemi comuni agli algoritmi greedy, essendo l'ordine di riempimento e la dimensione del patch Ψ_s parametri critici. Questo si riflette, per esempio, nella propagazione del-

l'errore dovuto all'errata imposizione dei valori di intensità in alcuni pixels. Ricordiamo infatti che ad ogni iterazione si la funzione M viene aggiornata con i pixels già restaurati.

3.3.2 Criminsi

Criminsi e Pérez nel loro articolo [15] propongono un evoluzione dell'idea di Efros ottenuta cambiando l'ordine di scelta dei pixels da restaurare. Anzichè usare il metodo *onion peel* usato comunemente propongono l'uso di un indice di priorità per ogni pixel della frontiera del dominio di inpainting ad ogni iterazione del dominio. Vediamo dunque come viene definita la priorità. Dato $p \in \partial D$, la priorità $P(p)$ viene definita come

$$P(p) = C(p)D(p)$$

dove $C(p)$, $D(p)$ prendono il nome di termine di confidenza e di dati, definiti come segue:

$$C(p) = \frac{\sum_{k \in \Psi_s \cap \bar{D}} C(p+k)}{|\Psi_s|}$$

e

$$D(p) = \frac{|\nabla^\perp u(p) \cdot n(p)|}{\alpha}$$

con α fattore di normalizzazione, $n(p)$ vettore normale (unitario) a ∂D in p . L'inizializzazione viene fatta con $C(p) = 0 \forall p \in D$ e $C(p) = 1 \forall p \in D^c$. Il coefficiente di confidenza può essere visto come un indice della quantità di informazione affidabile nell'intorno del pixel p . L'intenzione è quella di riempire prima i patch con il maggior numero di pixels non corrotti o già restaurati. Questo automaticamente comporta una certa preferenza verso alcune forme di tratti di frontiera (p.e. angoli). Il coefficiente dei dati D è una funzione della forza delle isofote che incontrano la frontiera di D . Questo termina aumenta la priorità dei patch con le isofote che fluiscono all'interno del dominio. Una volta computata $P(p) \forall p \in \partial D$ la sintesi viene fatta analogamente al metodo di Efros con la differenza che anzichè assegnare un pixel alla volta viene assegnato un valore a tutti i pixels del patch.

 Algoritmo di Criminsi

```

while  $D \neq \emptyset$  do
    1 - Identificare  $\partial D$ 
    2 - Calcolare  $P(p) \forall p \in \partial D$ 
    3 - Trovare  $\hat{p} : \hat{p} = \arg \max_{p \in \partial D} P(p)$ 
    4 - Trovare  $\hat{q} : \hat{q} = \arg \min_{q \in D^c} d^*(\hat{p}, q)$ 
    5 - Copiare  $\Psi_{\hat{q}}$  al posto di  $\Psi_{\hat{p}}$ 
    6 - Aggiornare valori confidenza  $C(p) = C(\hat{p}) \forall p \in D \cap \Psi_{\hat{p}}$ 
    7 - Porre  $M(\Psi_{\hat{p}}) = 1$ ,  $D = D \setminus \Psi_{\hat{p}}$ 
end while
    
```

Tabella 3.3: Algoritmo di Criminsi

3.3.3 Bugeau, Bertalmìo, Caselles e Sapiro

Bugeau, Bertalmìo, Caselles e Sapiro nel loro articolo [17] propongono un algoritmo che parte dalla sintesi di Efros e vi aggiunge vincoli di coerenza. Per coerenza si intende una sorta di rigidità nella scelta della mappa di corrispondenza ϕ , ossia fare in modo che se *al pixel x corrisponde il pixel $\phi(x)$* , *allora il pixel scelto per $x+k$ sia vicino a $\phi(x)+k$* . Nell'algoritmo gli autori propongono di usare l'output dell'algoritmo di Efros come prima iterazione. Il motivo di tale approccio è quella di raffinare la texture ottenuta, in quanto spesso presenta discontinuità negli edges. Al fine dunque di implementare la coerenza definiamo la distanza d_1 come:

$$d_1(p, q, \phi) = \sum_{k \in \Psi_s} \|u(\phi(p+k)) - u(q+k)\|^2$$

L'algoritmo proposto viene schematizzato nella tabella 3.4.

Oltre alla scelta della dimensione di Ψ_s e all'ordine di scelta dei pixels sulla frontiera, un altro parametro critico è la scelta della distanza. La distanza che viene usata più spesso negli algoritmi di sintesi di texture è l'SSD (Sum of Squared Difference) la quale risulta in realtà problematica per una vasta classe di immagini. Infatti spesso negli algoritmi iterativi come quello appena descritto la scelta della distanza SSD porta ad ottenere un inpainting troppo omogeneo. Gli autori propongono dunque una l'uso di una diversa distanza. Introduciamo la definizione di distanza di Bhattacharyya

Algoritmo di Bugeau, Bertalmò, Caselles e Sapiro 1

A - INITIALIZATION STEP

- 1 - $D^0 = D$
- 2 - Processare tutti i pixels:

```
while  $D^0 \neq \emptyset$  do
    a - Selezione pixel  $p$  dal bordo  $\partial D^0$ 
    b - Trovare  $\phi^0(p) : \phi^0(p) = \arg \min_{q \in D^c} d^*(p, q)$ 
    c - Porre  $u(p) = u(\phi^0(p))$ ,  $M(p) = 1$ ,  $D^0 = D^0 \setminus p$ 
end while
```

B - ITERATIVE PROCESS

- $n = 1$
- repeat**

- 1 - $D^n = D$
- 2 - Processare tutti i pixels:

```
while  $D^n \neq \emptyset$  do
    a - Selezione pixel  $p$  dal bordo  $\partial D^n$ 
    b - Trovare  $\phi^n(p) : \phi^n(p) = \arg \min_{q \in D^n} d_1(p, q, \phi^{n-1})$ 
    c - Porre  $u(p) = u(\phi^n(p))$ ,  $M(p) = 1$ ,  $D^n = D^n \setminus p$ 
end while
```

- 3 - $n = n + 1$
- until** Convergence

Tabella 3.4: Algoritmo di Bugeau, Bertalmò, Caselles e Sapiro 1

d_B :

$$d_B(P_p, P_q) = \sqrt{1 - \sqrt{\sum_{i=1}^B \rho_p(i)\rho_q(i)}}$$

dove P_p, P_q sono due patch centrali nei pixels p e q , con relativi istogrammi ρ_p, ρ_q . Il motivo di tale scelta è il fatto che un patch regolare si può distinguere da un patch con texture dalla sua distribuzione, la quale viene approssimata dall'istogramma. Inoltre la distanza d_B è invariante per rototraslazioni dei patches. Nonostante tali proprietà la distanza di Bhattacharyya non è comunque sufficiente per la sintesi e dunque gli autori propongono la nuova distanza d_T :

$$d_T(P_p, P_q) = \frac{1}{|\Psi_s|} d(P_p, P_q) d_B(P_p, P_q)$$

Uno dei principali lati negativi di questo algoritmo è il costo computazionale. Per restaurare ogni pixel si confronta il patch di tal pixel con tutti i pixels dell'immagine usando una distanza basato sul calcolo degli istogrammi, che richiede un ulteriore elevato costo computazionale. Al fine di velocizzare l'esecuzione gli autori propongono due accorgimenti. Il primo è quello di usare un metodo di pruning, ossia nel performare lo step di inizializzazione, per ogni pixel, si memorizzano gli N pixels di D^c con i più piccoli valori di distanza di texture. Chiamiamo insieme di pruning $C(p)$ l'insieme appena descritto. Nell'esecuzione della seconda parte dell'algoritmo, anziché cercare un match per il pixel p in tutto D^c , lo si cerca solo in $C(p)$.

in conclusione l'algoritmo diventa quello della tabella 3.5

3.4 Un approccio combinato

La maggior parte delle immagini, in realtà non sono fatte puramente di texture o puramente di struttura geometriche semplici, dunque un restauro puramente basato su EDP o puramente basato su sintesi di texture spesso non è sufficiente al restauro. Una terza via per restaurare le immagini è dunque un approccio combinato dei due metodi. In questi metodi, anzichè fare l'inpainting direttamente sull'immagine originale, questa viene decomposta nella somma di due immagini, una che rappresenti la struttura di base

Algoritmo di Bugeau, Bertalmò, Caselles e Sapiro 2

A - INITIALIZATION STEP

- 1 - $D^0 = D$
- 2 - Processare tutti i pixels:

while $D^0 \neq \emptyset$ **do**

- a - Selezione pixel p dal bordo ∂D^0
- b - Trovare $\phi^0(p) : \phi^0(p) = \arg \min_{q \in D^c} d_t(p, q)$
- c - Riempire l'insieme $C(p)$ con i migliori N candidati
- d - Porre $u(p) = u(\phi^0(p))$, $M(p) = 1$, $D^0 = D^0 \setminus p$

end while

B - ITERATIVE PROCESS

- $n = 1$
- repeat**

- 1 - $D^n = D$
- 2 - Processare tutti i pixels:

while $D^n \neq \emptyset$ **do**

- a - Selezione pixel p dal bordo ∂D^n
- b - Trovare $\phi^n(p) : \phi^n(p) = \arg \min_{q \in C(p)} d_1(p, q, \phi^{n-1})$
- c - Porre $u(p) = u(\phi^n(p))$, $M(p) = 1$, $D^n = D^n \setminus p$

end while

- 3 - $n = n + 1$

until Convergence

Tabella 3.5: Algoritmo di Bugeau, Bertalmò, Caselles e Sapiro 2

mentre l'altra rappresentante la texture (ed il rumore). Lo scopo è quello di ottenere due immagini adatte rispettivamente ad un inpainting basato su PDE ed uno su sintesi di texture. Una volta processate, le due immagine vengono poi sovrapposte nuovamente per ottenere l'immagine finale. I blocchi fondamentali di tale processo sono dunque:

1. Filtraggio/decomposizione dell'immagine
2. Inpainting strutturale
3. Sintesi di texture
4. Sovrapposizione

Per quanto riguarda il primo punto sono possibili diversi approcci: alcuni autori scompongono l'immagine u come $u = L + H$ ottenendo H (L) con un filtro passa alto (passa basso) ed L (H) come $u - H$ ($u - L$). In seguito è possibile scomporre H con una piramide laplaciana così come propongono [18] e [17]. Altri invece (per esempio [13]) applicano all'immagine un'equazione di diffusione non lineare, per esempio:

$$\frac{\partial u}{\partial \tau} = \nabla \cdot (d(\|\nabla u(x, y)\|) \cdot \nabla u)$$

$$d(s) = \frac{1}{1 + \frac{s^2}{\lambda^2}}$$

dove la soluzione presa ad un opportuno tempo τ_0 (chiamiamola G) sarà la decomposizione geometrica della nostra immagine mentre $u - G$ sarà la componente di texture. Il processing delle due immagini ottenute dalla decomposizione viene fatto con una delle tecniche menzionate nei paragrafi precedenti (in letteratura sono presenti un vasto numero di altri algoritmi possibili). Nel caso poi dell'uso della piramide di laplaciana [17] propongo di partire dall'immagine più piccola della piramide e di applicare, per intero, l'algoritmo 3.5. Dopodichè usare il risultato ottenuto come inizializzazione per il passo successivo dove si eseguirà solo la parte B-ITERATIVE PROCESS sull'immagine successiva della piramide. Iterando tale processo si ottiene la sintesi della piramide.

Infine lo sovrapposizione consiste nella semplice somma algebrica delle due immagini.

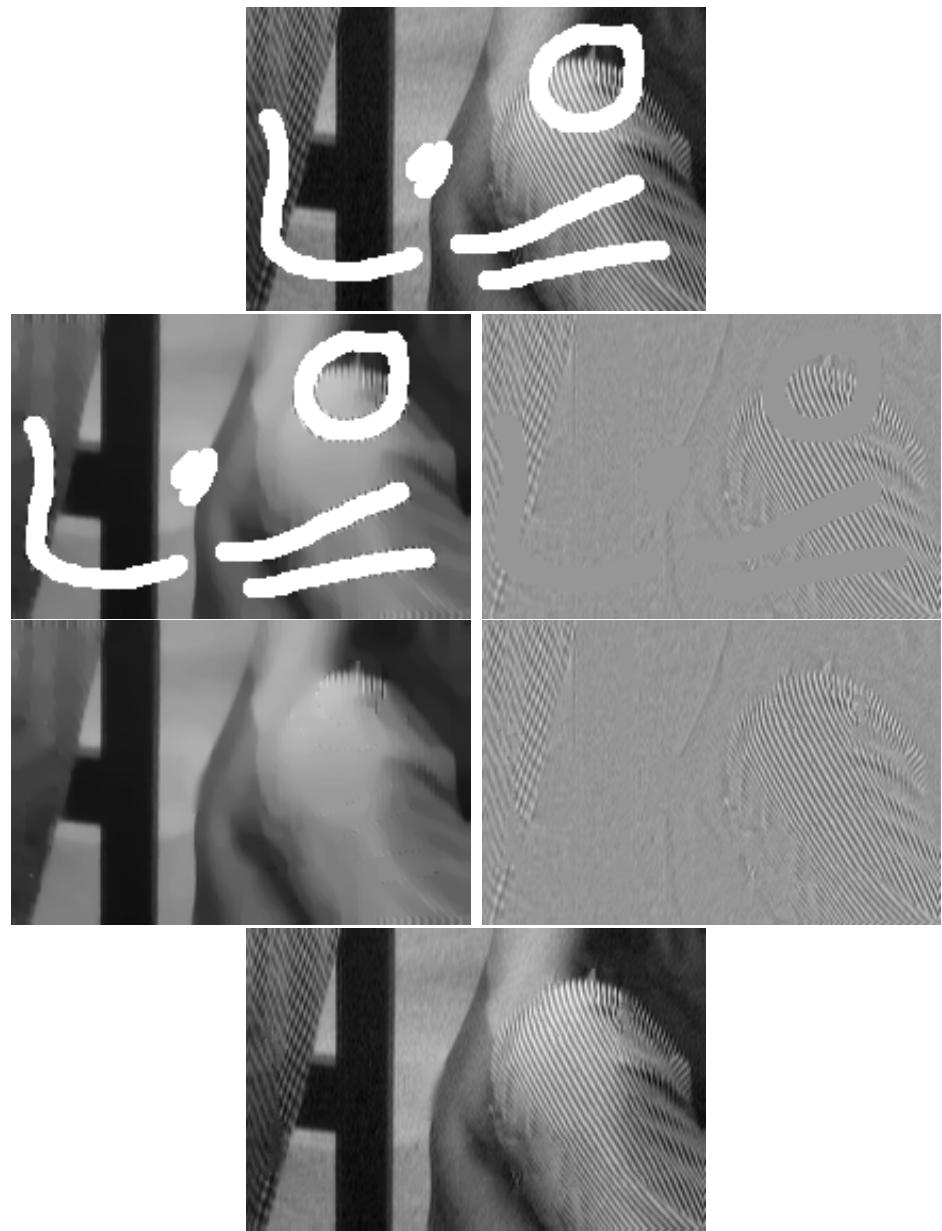


Figura 3.7: PDE inpainting e sintesi di texture. prima riga: immagine da restaurare, seconda riga: scomposizione dell’immagine nelle due parti, terza riga: inpainting, quarta riga: sovrapposizione

Capitolo 4

Restauro delle trasformate di Radon

Lo scopo di questo lavoro di tesi è quello di restaurare le trasformate di Radon che presentano artefatti dovuti alla presenza di protesi metalliche usando le tecniche di restauro di immagini digitali. Tale approccio, come già citato nella sezione 1.4 viene giustificato da svariati articoli presenti in letteratura da diversi anni. La ricostruzione tramite algoritmo FBP è la più usata negli scanneri tomografici di uso commerciale, in quanto produce immagini di buona qualità in brevi tempi di esecuzione. Ne consegue che il nostro studio, come molti altri in letteratura, si focalizza nello sviluppo di tecniche di MAR che siano compatibili ed il più efficienti possibile con l'uso di tale tecnica di ricostruzione.

4.1 Instabilità della ricostruzione

Come abbiamo visto nel paragrafo 1.4 il restauro dell'immagine viene fatto a livello del sinogramma in quanto in quel caso l'insieme dei dati corrotti consiste in un numero di tracce metalliche facilmente localizzabile, mentre gli artefatti presenti sull'immagine ricostruita sono delle spesso difficili da localizzare con precisione. Inoltre, come spiegato nella sezione 1.3, i dati raccolti dai detectors degli scanneri sono proprio i valori della trasformata di Radon e dunque sono dati non manipolati, ossia non soggetti ad errori o approssimazioni dovuti ad ulteriori elaborazioni. Nel nostro caso invece, non abbiamo accesso ai dati raccolti dagli scanneri, ma solo all'immagine

ricostruita. Dunque al fine di poter applicare le tecniche di inpainting dobbiamo ricalcolare la trasformata di Radon dell'immagine. Questo implica che una volta ricostruita l'immagine per una seconda volta si sarà introdotto un ulteriore errore nell'immagine dovuto alla ricostruzione (come spiegato nel capitolo 1) ed implica anche un ulteriore difficoltà per noi vista la minore regolarità del sinogramma ottenuto.

Vediamo ora di quantificare l'errore che viene introdotto con la ricostruzione di Radon al variare della scelta del filtro di retroproiezione (fig.1.2) e della scelta degli angoli di proiezione, supponendo che siano tutti equispaziati con passo $\Delta\theta$. Sia $f(x, y)$ il fantoccio di Shepp-Logan dell'esempio 1.1.4 e sia $R(\theta, x')$ la sua trasformata di Radon. Sia inoltre $\hat{f}(x, y)$ il fantoccio ricostruito con relativa trasformata $\hat{R}(x, y)$ allora gli errori che valuteremo saranno:

$$e_f = \frac{\|f - \hat{f}\|_{H^1}}{\|f\|_{H^1}}$$

e

$$e_R = \frac{\|R - \hat{R}\|_{H^1}}{\|R\|_{H^1}}$$

Dalle tabelle 4.1 possiamo subito osservare che:

- al diminuire di $\Delta\theta$ l'errore diminuisce finché non rimane pressoché costante. Dalla teoria sappiamo che l'errore dovrebbe tendere a zero al diminuire di $\Delta\theta$. Ovviamente avendo però risoluzione spaziale finita, al passare sotto una certa soglia di $\Delta\theta$ non stiamo in realtà aggiungendo nessuna informazione. Inoltre, come menzionato in precedenza, al diminuire di $\Delta\theta$ il contributo del rumore aumenta in quanto aumentano le proiezioni.
- Nel caso preso in esame (fantoccio di Shepp-Logan), indipendentemente dal filtro scelto, $\Delta\theta = 0.5$ è sufficiente per la ricostruzione.
- il filtro Ramp nonostante sia il più sensibile al rumore è quello che in questo semplice caso di esempio funziona meglio
- si ha una conferma empirica dell'instabilità dell'inversione della trasformata infatti un errore relativo sulla trasformata dell'ordine del 3%,4% corrisponde ad un significativo errore relativo sull'immagine.

Ramp filter			Shepp filter		
$\Delta\theta$	$e_f \%$	$e_R \%$	$\Delta\theta$	$e_f \%$	$e_R \%$
1	21.2082	3.7674	1	22.2638	3.8768
1/2	18.7296	3.5902	1/2	20.6208	3.7276
1/3	18.2894	3.5536	1/3	20.3278	3.6969
1/4	18.1662	3.5320	1/4	20.2535	3.6766
1/5	18.1010	3.5209	1/5	20.2056	3.6572
1/6	18.0694	3.4830	1/6	20.1860	3.6271
1/7	18.0445	3.5237	1/7	20.1659	3.6480
1/8	18.0450	3.4852	1/8	20.1717	3.6286

Hamming filter			Hann filter		
$\Delta\theta$	$e_f \%$	$e_R \%$	$\Delta\theta$	$e_f \%$	$e_R \%$
1	27.6450	4.4536	1	28.4510	4.5281
1/2	27.2154	4.3110	1/2	28.0795	4.3801
1/3	27.1276	4.2758	1/3	28.0013	4.3431
1/4	27.1094	4.2566	1/4	27.9845	4.3232
1/5	27.0899	4.2216	1/5	27.9668	4.2858
1/6	27.0882	4.2052	1/6	27.9657	4.2711
1/7	27.0746	4.1956	1/7	27.9520	4.2570
1/8	27.0852	4.2051	1/8	27.9626	4.2705

Tabella 4.1: Errori di ricostruzione

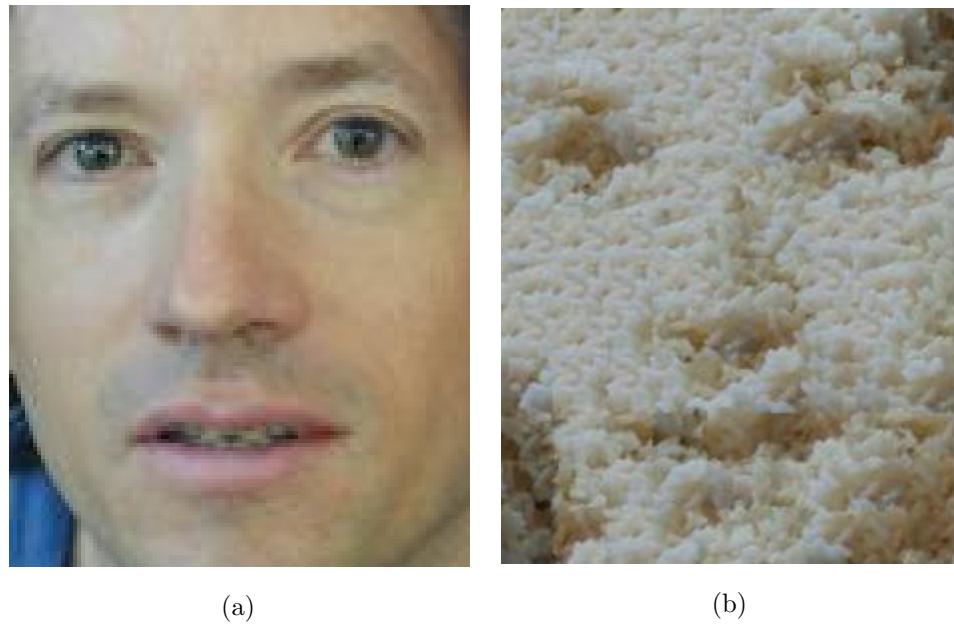


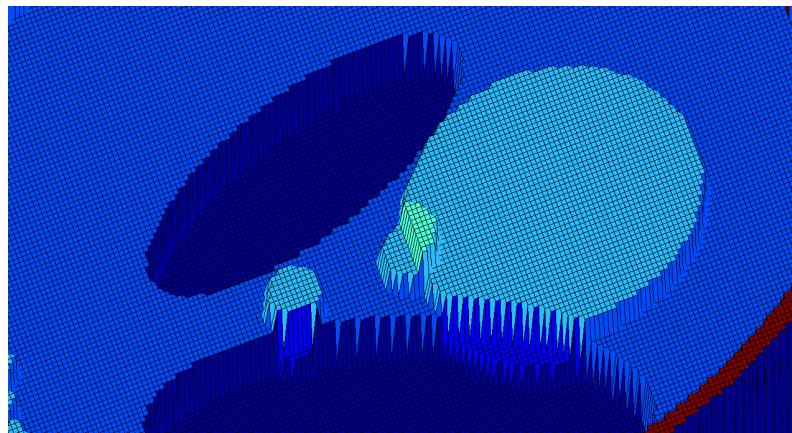
Figura 4.1: Esempio di trasferimento di texture [16]

Sottolineiamo nuovamente quanto sia mal posto il nostro problema di restauro, in quanto costretti ad operare sulla trasformata di Radon dell'immagine precedentemente ricostruita.

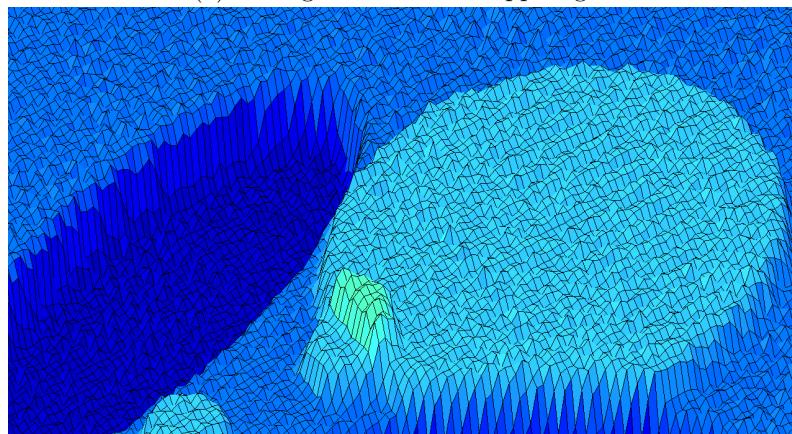
In altre situazioni un errore dell'ordine 20% sarebbe inaccettabile, ma nel caso delle immagini entra in gioco il fattore dell'inferenza visiva. Confrontando le figure 1.5a e 1.11 (rispettivamente fantoccio originale e ricostruito) possiamo osservare che nonostante un errore del 21.2% l'immagine è discretamente ricostruita, ossia è possibile distinguere senza troppi problemi tutti i dettagli dell'immagine. Un esempio estremo è quello di figura 4.1 in cui nonostante sicuramente lo scarto tra le due immagine sia elevatissimo, si ha una forte risposta da parte del sistema visivo ad alcuni elementi di maggiore importanza percettiva, dunque la capacità di elaborazione del cervello umano permette comunque di trovare un forte nesso tra le due immagini.

4.2 Pre-Processing

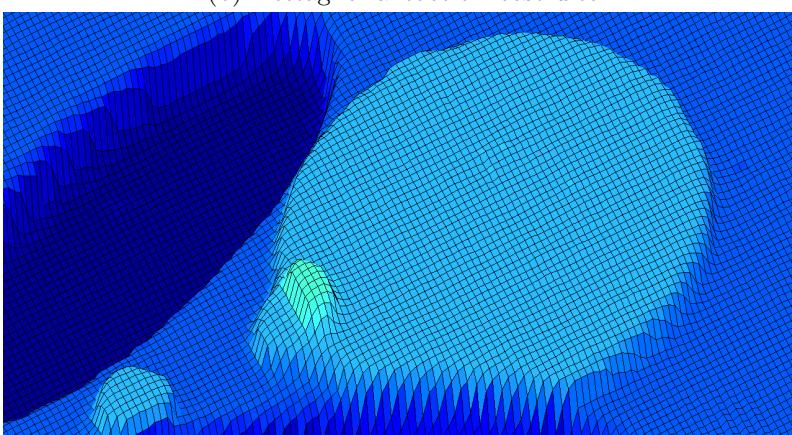
Al fine di ridurre l'errore di ricostruzione nell'immagine in cui dovremo fare l'inpainting abbiamo visto che un semplice ed efficace pre-processing possibile è l'applicazione di un filtro mediano all'immagine. A confermarlo oltre un



(a) Dettaglio fantoccio Shepp-Logan



(b) Dettaglio fantoccio ricostruito



(c) Dettaglio fantoccio ricostruito dopo median filtering

Figura 4.2: Esempio uso filtro mediano

abbassamento di circa due punti percentuali su e_f , è un chiaro miglioramento visivo.

4.3 Struttura metodi di interpolazione

Come menzionato nella sezione 1.4 i metodi di restauro si dividono in due categorie principali: metodi di interpolazione e metodi iterativi. Vediamo ora la struttura di un generico metodo di interpolazione (fig. 4.3):

1. Estrazione parte metallica da immagine di partenza
2. Trasformata di Radon dell'immagine di partenza e dell'immagine della parte metallica
3. Identificazione dominio
4. Interpolazione nel dominio di inpainting
5. Inversione sinogramma restaurato
6. Somma dell'immagine ottenuta al punto precedente con la parte metallica

Passiamo ora ad una descrizione dettagliata di come noi abbiamo affrontato questi punti:

1. Dalla sezione 1.3 sappiamo che data una TAC con i valori di grigio equivalenti alle Hounsfield Units, nessun valore maggiore di 3000 HU può essere fisiologico. Dunque facciamo un thresholding dell'immagine di partenza con soglia pari a 3000.
2. Le due trasformate di Radon vengono fatte scegliendo $\Delta\theta$ in base a e_f come spiegate in precedenza. Per avere una buona ricostruzione bisogna variare l'angolo su uno span di almeno 180° . Non è necessario doverlo fare su 360° per via della simmetria citata nel primo capitolo, ma per questioni di implementazione in alcune occasioni si è preferito fare la trasformata su uno span di 360° .
3. Nel nostro caso l'identificazione del dominio viene fatta sottraendo il sinogramma della parte metallica al sinogramma dell'immagine originale. Questa è la tecnica più usata essendo la più semplice ma anche la più naturale. Altre tecniche sono possibile per segmentare la

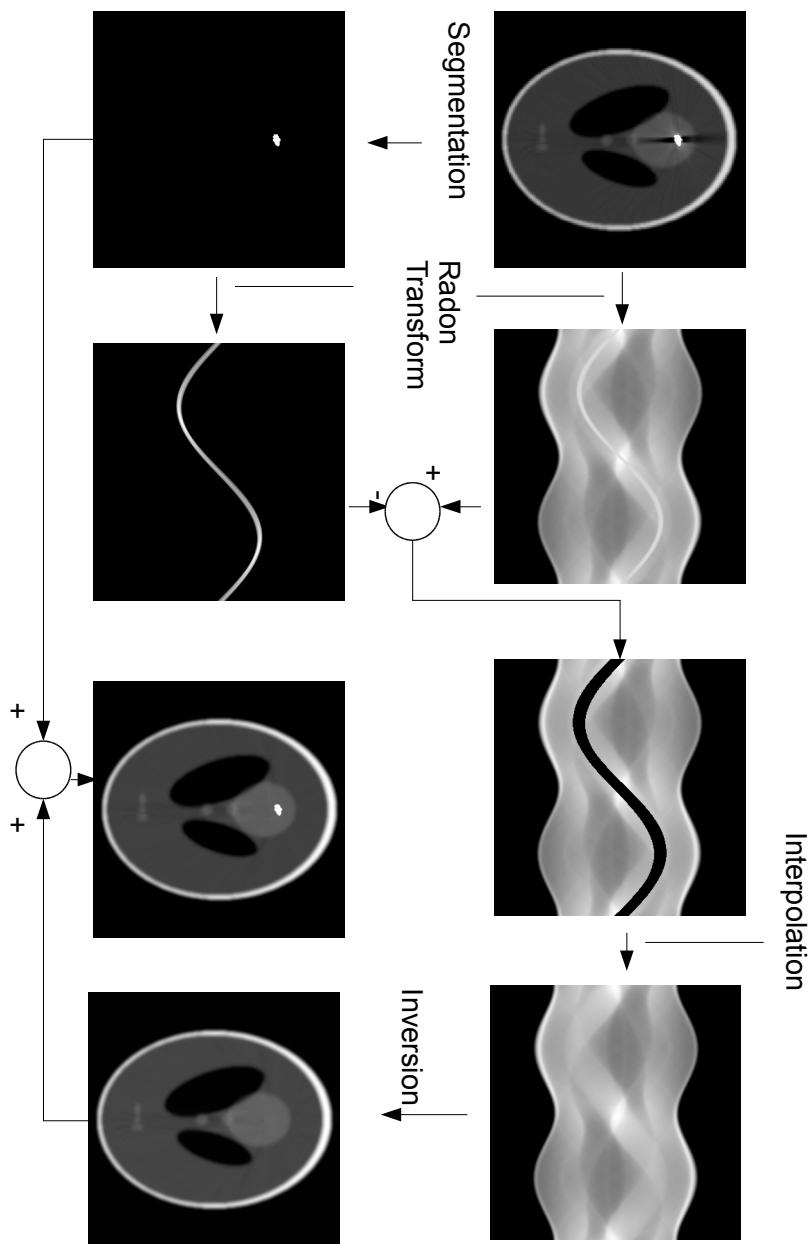


Figura 4.3: Riepilogo grafico struttura metodi di interpolazione

traccia relativa alla parte metallica sul sinogramma dell'immagine di partenza. Osserviamo che l'uso della differenza dei due sinogrammi è indicativo del fatto che si assume l'assenza di informazioni all'interno del supporto della trasformata (dell'inviluppo convesso) dell'immagine metallica.

4. Nella sezione 4.4 dettagliamo i metodi utilizzati nella nostra tesi
5. Ricostruzione tramite FBP con filtro a scelta dell'immagine ottenuta dal sinogramma restaurato

4.4 Interpolazioni

4.4.1 Interpolazione di Bertalmio

Le equazioni (1): Nel presente lavoro si è implementato il metodo di Bertalmio ad elementi finiti tramite l'uso dei software Matlab e Freefem++. Supponendo di avere preso le proiezioni da 0 a 2π le equazione da risolvere secondo Bertalmio et al. sono dunque:

$$\left\{ \begin{array}{ll} v \cdot \nabla v - \nu \Delta v + \nabla p = 0 & D \\ \nabla \cdot v = 0 & D \\ v(\cdot, 0) = v(\cdot, 2\pi) & \\ v = \nabla^\perp u_{orig} & \partial D \setminus \{(x', \theta) \mid \theta = 0 \vee \theta = 2\pi\} \end{array} \right. \quad (4.1)$$

$$\left\{ \begin{array}{ll} \nabla^\perp u = v & D \\ u(\cdot, 0) = u(\cdot, 2\pi) & \\ u = u_{orig} & \partial D \setminus \{(x', \theta) \mid \theta = 0 \vee \theta = 2\pi\} \end{array} \right. \quad (4.2)$$

dove D è il dominio identificato in precedenza, ν è il coefficiente di diffusività, $u_{orig}(x', \theta)$ è la trasformata di Radon dell'immagine da restaurare, $u(x', \theta)$ è la soluzione nel dominio e le condizioni al bordo periodiche sono dovute alla periodicità della trasformata.

Il gradiente: Come possiamo vedere come condizioni al bordo nel problema in velocità abbiamo bisogno del gradiente del sinogramma di partenza. L'approssimazione di tale gradiente può essere fatta tramite differenze finite,

anche se queste sono molto sensibili al rumore. Una stima più accurata del gradiente può essere fatta prendendo il gradiente calcolato con le differenze finite ed applicando un filtro di media a tal gradiente. Molto importante è la scelta della dimensione del patch del filtro di media. Al variare di tale parametro infatti si catturerà l'andamento della dinamica dell'immagine su scale diverse. Ne consegue che non possibile stimare il gradiente sulla frontiera del dominio in quanto ci si troverebbe a fare una media su dei dati corrotti. Per ovviare a tale inconveniente si è deciso di dilatare il dominio di inpainting così da poter calcolare il gradiente sul bordo senza problemi. Per quanto riguarda il gradiente sui contorni verticali dell'immagine, vista la proprietà di periodicità, questo è stato calcolato prendendo l'estensione periodica dell'immagine e poi usando differenze finite e media aritmetica.

Triangolazione: La triangolazione del dominio viene fatta su Matlab ed in seguito esportata su FreeFem++. Senza entrare troppo nel dettaglio specifico dell'implementazione vediamo i passi che sono stati effettuati per la triangolazione:

1. Come spiegato in precedenza, a questo punto abbiamo già identificato il dominio, la cui funzione caratteristica è quella di figura 4.4b.
2. Dilatiamo il dominio al fine di poter calcolare meglio il gradiente.
3. Sapendo che dovremmo in seguito imporre condizioni periodiche agli estremi verticali, controlliamo che anche la funzione caratteristica χ_D sia periodica. In caso non lo sia forziamo noi la periodicità.
4. Identifichiamo i punti che formano il contorno dell'immagine (figura 4.4c)
5. Tramite l'uso di una classe implementata in Matlab (DelaunayTri) otteniamo una triangolazione strutturata con i vertici dati dai pixels della funzione indicatrice χ_D .
6. Così facendo otteniamo in realtà la triangolazione dell'inviluppo concavo del nostro dominio (figura 4.4d).
7. Al fine di avere la giusta triangolazione dobbiamo fornire dei vincoli alla classe DelaunayTri, ossia dobbiamo fornire le coppie di vertici dei

lati che compongono ogni curva chiusa che delimita il dominio (figure 4.4e e 4.4f).

8. Una volta forniti vincoli alla classe, otteniamo la triangolazione desiderata (figura 4.5).
9. Costruiamo le matrice necessarie ([20]) ad esportare la matrice al fine di renderla utilizzabile da FreeFem++.

Le equazioni (2): Le equazioni che in realtà risolviamo sono leggermente diverse da quelle di Bertalmio. Le principali differenze sono che il termine di diffusione anisotropa viene sostituito da uno di diffusione isotropa, viene aggiunto un termine di penalizzazione nella parte del dominio ottenuta dalla dilatazione e si applica il rotore alla prima equazione del sistema 4.2. Otteniamo dunque

$$\left\{ \begin{array}{ll} v \cdot \nabla v - h B \nabla \cdot ((v \cdot \nabla v) \cdot v) + \nabla p + \lambda (v - \nabla^\perp u_{orig}) = 0 & D \\ \nabla \cdot v = 0 & D \\ v(\cdot, 0) = v(\cdot, 2\pi) & \\ v = \nabla^\perp u_{orig} & \partial D \setminus \{(x', \theta) \mid \theta = 0 \vee \theta = 2\pi\} \end{array} \right. \quad (4.3)$$

$$\left\{ \begin{array}{ll} \Delta u + \lambda (u - u_{orig}) = \nabla \times v & D \\ u(\cdot, 0) = u(\cdot, 2\pi) & \\ u = u_{orig} & \partial D \setminus \{(x', \theta) \mid \theta = 0 \vee \theta = 2\pi\} \end{array} \right. \quad (4.4)$$

con D dominio dilatato, h diametro massimo della triangolazione (ci garantisce consistenza della stabilizzazione), $B = |v|^{-1}$ fattore di normalizzazione, λ funzione caratteristica della differenza tra il dominio dilatato e quello ottenuto all'inizio. Facciamo notare che il motivo della scelta di usare la stabilizzazione streamline diffusion sta nel fatto che il metodo di Bertalmio comporta solo il trasporto e non la diffusione, dunque data la consistenza del metodo streamline la nostra versione fa sì che il metodo assomigli di più a quello di Bertalmio piuttosto che quello con la diffusione isotropa.

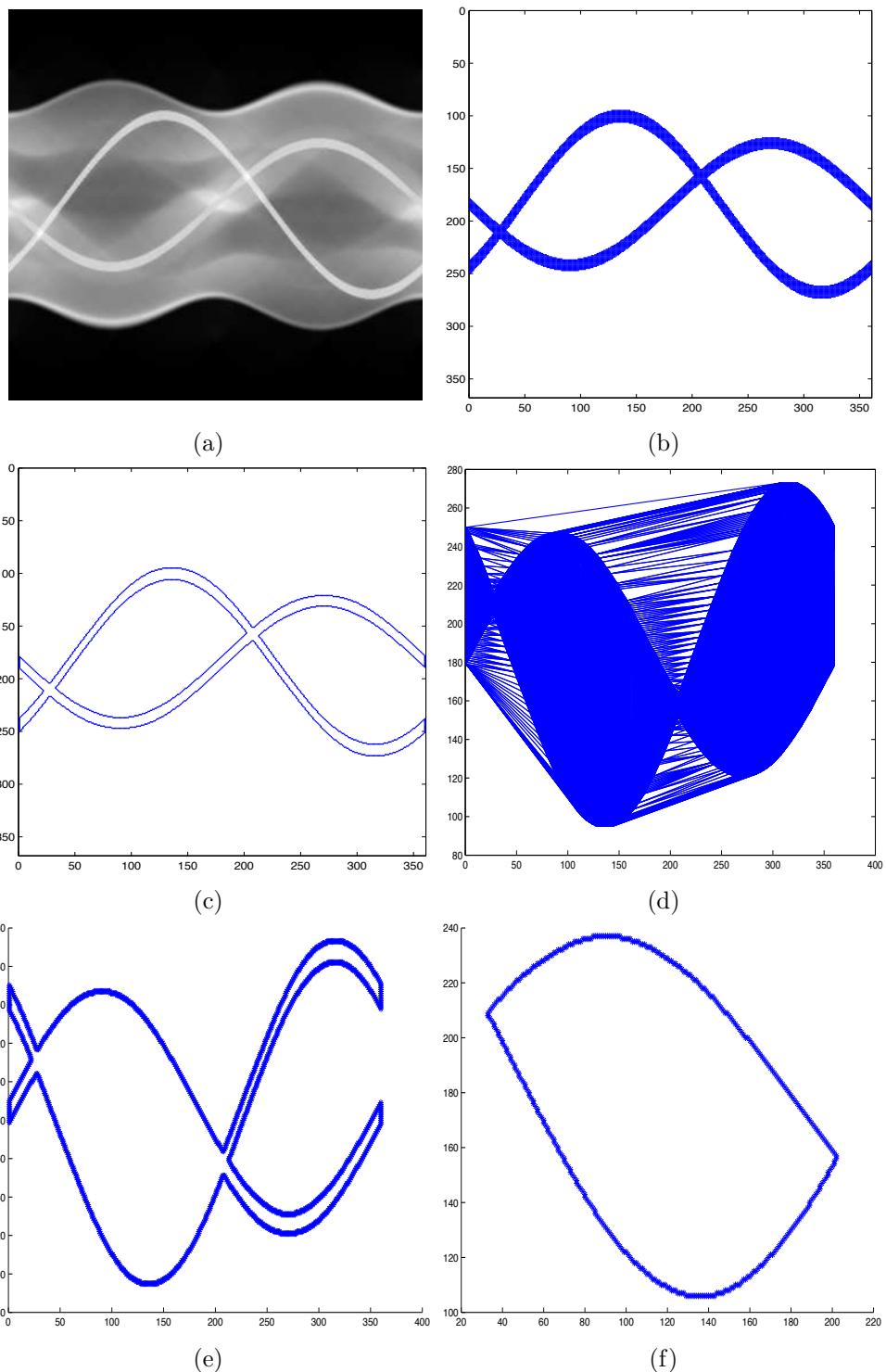


Figura 4.4: Fasi algoritmo meshing

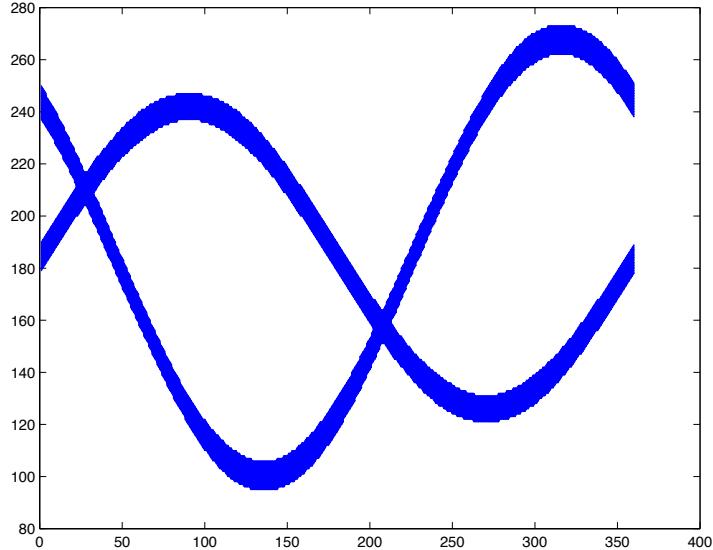


Figura 4.5: Triangolazione finale

La formulazione debole Dati i seguenti spazi ad elementi finiti continui

$$\begin{aligned} V &= V(D) = \{w(x', \theta) \mid w = \sum_k w_k \eta_k(x', \theta), \quad w_k \in \mathbb{R}, \quad (x', \theta) \in D, \quad w(\cdot, 0) = w(\cdot, 2\pi)\} \\ Q &= Q(D) = \{q(x', \theta) \mid q = \sum_k q_k \psi_k(x', \theta), \quad q_k \in \mathbb{R}, \quad (x', \theta) \in D\} \\ U &= U(D) = \{\varphi(x', \theta) \mid \varphi = \sum_k \varphi_k \zeta_k(x', \theta), \quad \varphi_k \in \mathbb{R}, \quad (x', \theta) \in D, \quad \varphi(\cdot, 0) = \varphi(\cdot, 2\pi)\} \end{aligned}$$

con η_k, ψ_k, ζ_k appartenenti ad opportuni spazi polinomiali, integrando per parti le equazioni 4.3 otteniamo la seguente formulazione debole:

$$?(v, p) \in V \times Q : v = \nabla^\perp u_{orig} \text{ in } \partial D \setminus \{(x', \theta) \mid \theta = 0 \vee \theta = 2\pi\}$$

$$\left\{ \begin{array}{l} h B \int_D (v \nabla v)(v \nabla w) + \int_D (v \nabla v) w - \int_D p (\nabla \cdot w) + \lambda \int_D (v - \nabla^\perp u_{orig}) w = 0 \\ \int_D q (\nabla \cdot u) = 0 \end{array} \right. \quad (4.5)$$

$$\forall (w, q) \in V \times Q$$

Integrando per parti le equazioni 4.4 otteniamo invece:

$$?u \in U : u = u_{orig} \text{ in } \partial D \setminus \{(x', \theta) \mid \theta = 0 \vee \theta = 2\pi\}$$

$$\int_D \nabla u \nabla \varphi + \lambda \int_D (u - u_{orig}) \varphi = \int_D (\nabla \times v) \varphi \quad \forall \varphi \in U \quad (4.6)$$

La risoluzione su FeeFem++ Data la non linearità del problema 4.5 la risoluzione di tale problema viene fatta con un metodo di punto fisso inizializzato con la soluzione del problema di Stokes alla prima iterazione. Ad ogni iterazione del metodo si risolve il sistema lineare ottenuto con un metodo GMRES precondizionato con la matrice di massa in pressione, per evitare che con l'aumentare dei punti di griglia cresca troppo il numero di iterazioni necessarie per arrivare a convergenza. Infine per evitare di dover introdurre troppa diffusione ad ogni passo del metodo di punto fisso si fa un adattazione di griglia in funzione della matrice Hessiana della velocità.

I passi sono:

1. Viene importata la mesh T_h del dominio costruita con Matlab e se ne costruisce un'altra per tutta l'immagine u_{orig} che ci servirà per importare i dati necessari ed in seguito per esportarli. Chiamiamo tale mesh strutturata T_{hsq} .
2. Vengono definiti gli spazi ad elementi finiti a dimensione finta $V_h \subset V, Q_h \subset Q, U_h \subset U$ e V_{hsq} in maniera analoga a V_h ma sulla mesh T_{hsq} . Si dichiarano poi le variabili appartenenti a tali spazi necessarie per l'algoritmo.
3. Vengono definite le seguenti forme bilineari

$$\begin{aligned} a(v_k, w) &= \frac{h}{|v_{k-1}| + \epsilon} \int_D (v_{k-1} \nabla v_k)(v_{k-1} \nabla w) + \lambda \int_D v_k w \\ c(v_k, w) &= \int_D (v_{k-1} \nabla v_k) w \\ b(p_k, w) &= - \int_D p_k (\nabla \cdot w) \\ m(p, q) &= \int_D p q \\ a_{st}(v, w) &= \int_D \nabla v \nabla w \\ F(w) &= \lambda \int_D \nabla^\perp u_{orig} w \end{aligned}$$

4. Ad ogni passo del metodo di punto fisso risolviamo il sistema in pressione di Navier-Stokes con precondizionatore $P = \frac{\|v_{k-1}\|}{h+\epsilon} M$ dove M è la matrice di massa ottenuta dalla dorma bilineare $m(\cdot, \cdot)$. Come

criterio di arresto si è usato

$$\frac{\|v_k - v_{k-1}\|_{H^1}}{\|v_0\|_{H^1}} \leq \varepsilon_{toll}$$

5. Una volta calcolata la soluzione v si risolve il problema 4.6 con un metodo diretto
6. v ed u sono ora definite sulla mesh T_h diversa da quella iniziale. Per definire dunque il valore di ogni pixel dell'immagine si interpola la soluzione sulla mesh T_{hsq} ed in seguito si esportano le soluzioni su Matlab

Inversione soluzione e sovrapposizione Una volta importata la soluzione u su Matlab, si applica l'algoritmo di inversione FBP alla funzione

$$u \chi_D + u_{orig} \chi_{D^c}$$

e si ottiene così l'immagine restaurata ma priva della parte metallica. Dunque come ultima operazione si sovrappone la parte metallica segmentata sull'immagine restaurata.

4.4.2 Interpolazione polinomiale

Un altro metodo spesso usato in letteratura è quello dell'interpolazione lineare. Questo metodo consiste nello scorrere tutte le proiezioni della trasformata $u_{orig}(x', \theta)$ al variare di θ ed interpolare all'interno della striscia di dominio. Dunque supponiamo di fissare θ avremo che lungo il segmento $\chi_D(\cdot, \theta)$ i dati della trasformata sono corrotti. Il metodo consiste dunque nell'approssimare u lungo tale segmento a partire dai dati non corrotti che si trovano negli intorni unilaterali degli estremi di tale segmento.

Capitolo 5

Risultati numerici

5.1 Dati sintetici

Per costruire i dati sintetici siamo partiti dal classico fantoccio di Shepp-Logan (fig. 5.1a) ed abbiamo aggiunto le componenti metalliche oltre che a del rumore gaussiano al fine di rendere il problema più realistico (fig. 5.1b). Ne abbiamo poi calcolato la trasformata di Radon (fig. 5.1c) ed identificato la traccia dovuta alle parti metalliche. Infine per ottenere gli artefatti abbiamo intensificato il valore della trasformata nella parte metallica (fig. 5.1d) al fine di avere una forte discontinuità che una volta filtrata e retroproiettata produrrà, per il fenomeno di Gibbs, i desiderati artefatti (fig. 5.1e). Come spiegato nel capitolo precedente, il restauro viene fatto partendo dalle immagini con gli artefatti (fig. 5.1e) nonostante avessimo già a disposizione la relativa trasformata (fig. 5.1d), questo perchè nel caso dei dati clinici le case produttrici dei macchinari per le TAC si rifiutano di fornire i sinogrammi ottenuti dalle scansioni. Nelle figure 5.2, 5.3, 5.4, e 5.5 possiamo vedere i risultati dell'inpainting dell'algoritmo di Bertalmio.

Nella figura 5.6 possiamo invece osservare due esempi che diano un'idea di come l'algoritmo funzioni molto meglio quando viene usato con il sinogramma originale.

La stragrande maggioranza degli articoli presenti in letteratura presenta simulazioni dei metodi proposti eseguiti con approssimazioni alle differenze finite. Osserviamo in figura 5.7 come il nostro metodo si comporta in confronto con altri metodi presenti in letteratura implementati a differenze finite.

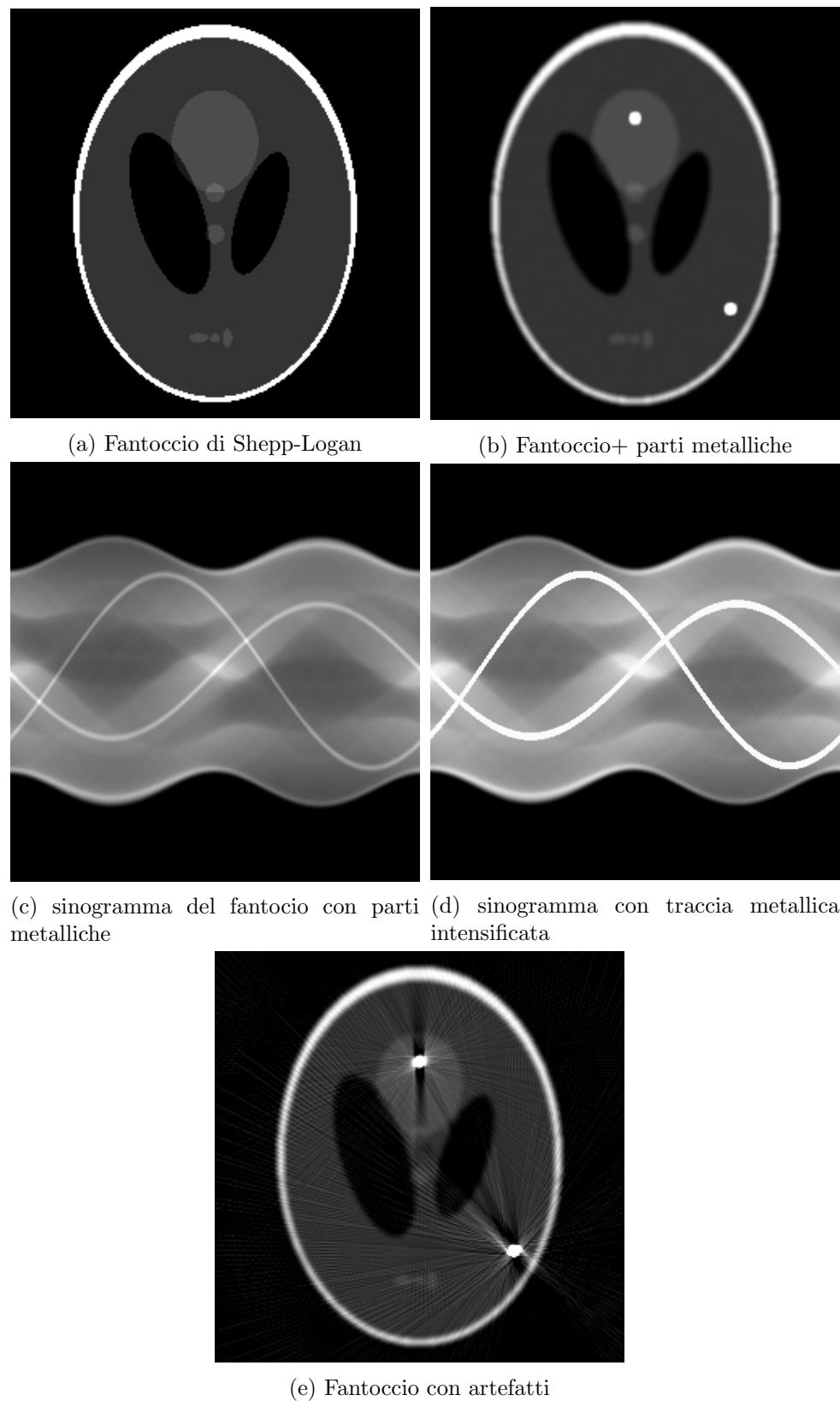


Figura 5.1: Creazione dati sintetici

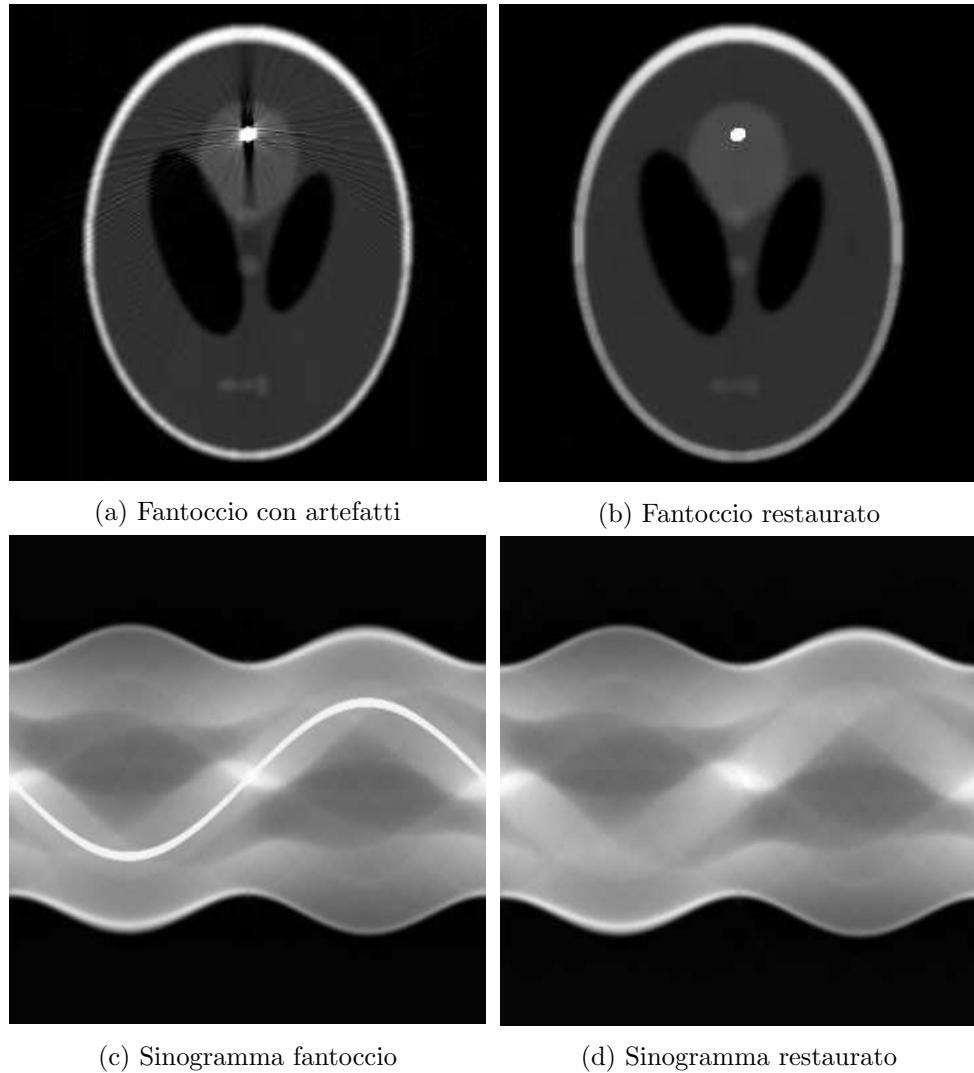


Figura 5.2: Restauro fantoccio con singola componente metallica

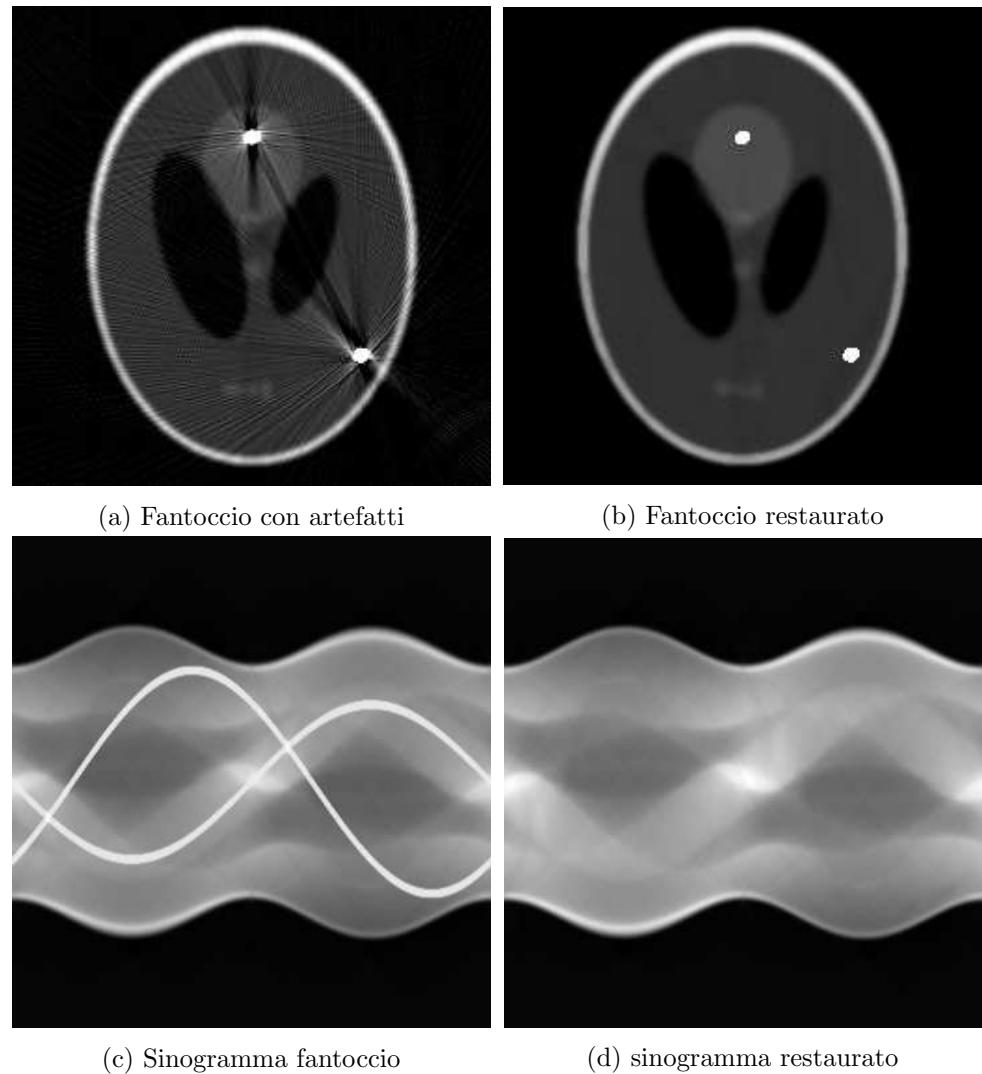


Figura 5.3: Restauro fantoccio con due componenti metalliche

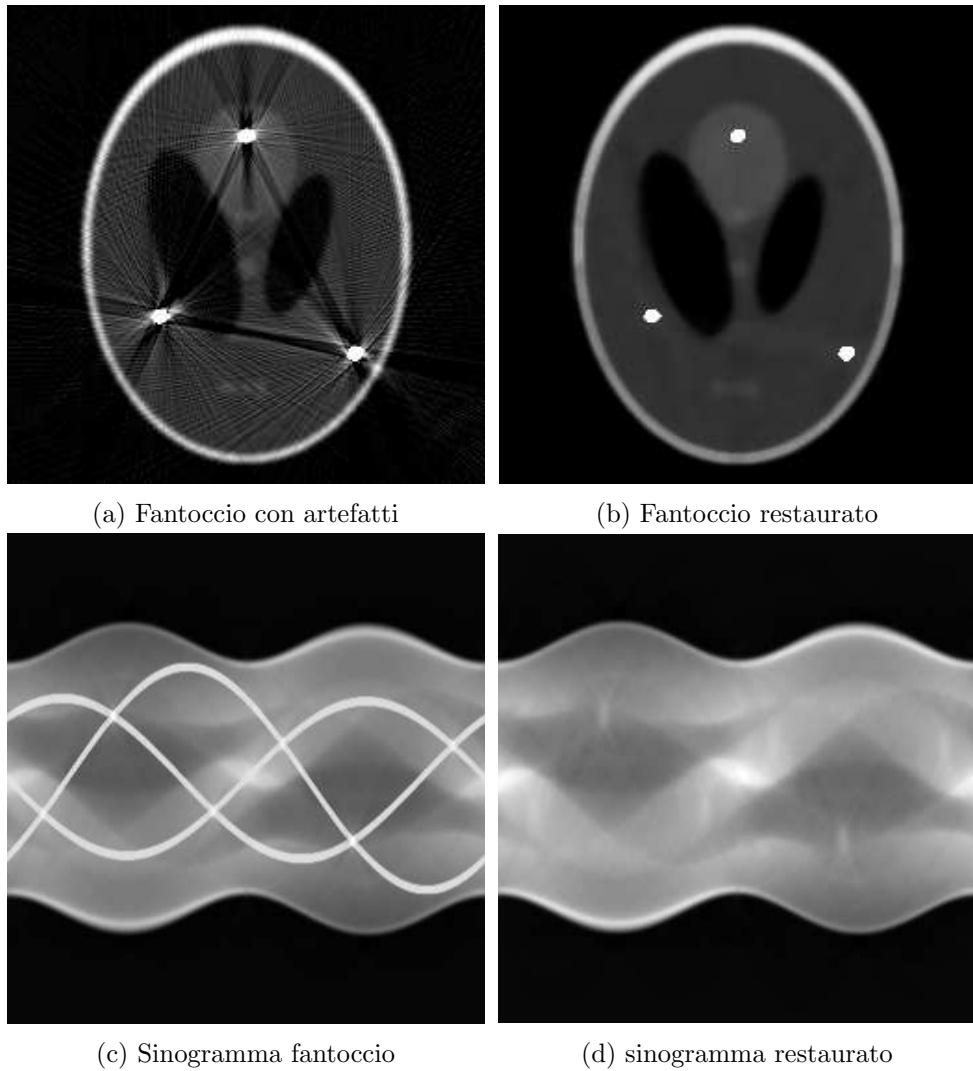


Figura 5.4: Restauro fantoccio con tre componenti metalliche

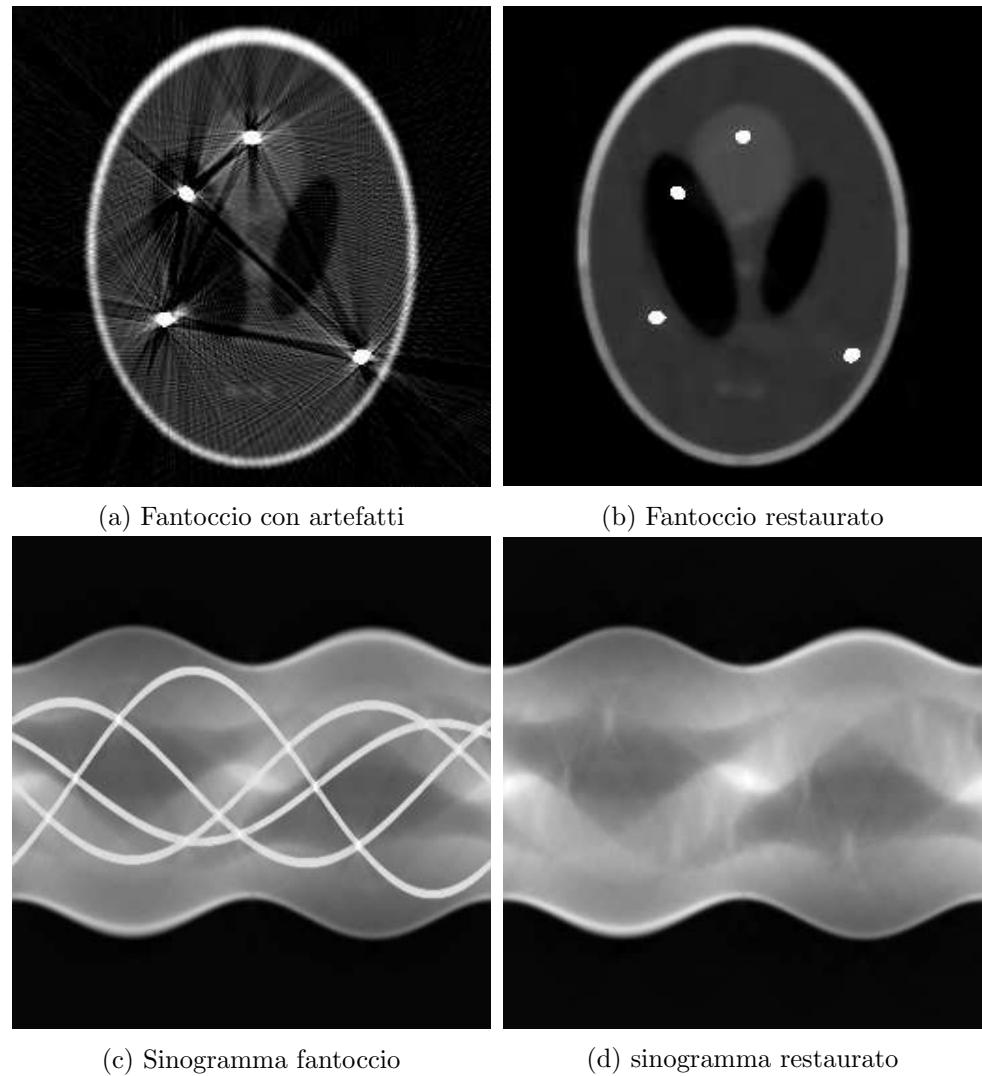


Figura 5.5: Restauro fantoccio con quattro componenti metalliche

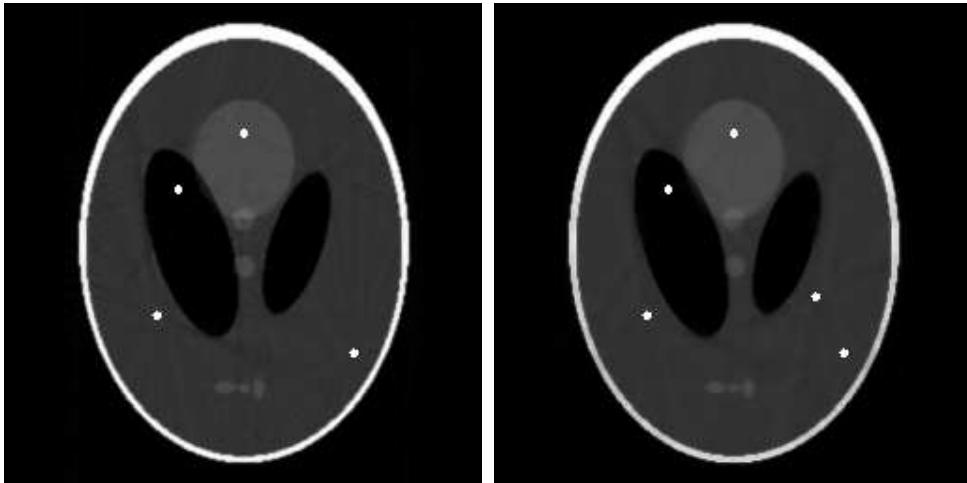


Figura 5.6: Fantocci restaurati da sinogrammi originali

I metodi presi in considerazione sono:

- TV: metodo variazione totale visto nella sezione 3.2.1
- Interpolazione lineare (sezione 4.4.2)
- Equazione del calore: l'equazione 3.3 diventa la nota equazione del calore

$$\partial_t u = -d\Delta u + \lambda(u - u_0)$$

con d coefficiente di diffusione

- Metodo TV-H⁻¹ : metodo sviluppato da Bertozzi, Burger, He and Schonlieb ([21]) consiste in un metodo del quarto ordine dove il termine $\partial_u R(u)$ consiste nel laplaciano della curvatura di u . Dunque l'equazione 3.3 diventa

$$\partial_t u = \Delta \left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right) + \lambda(u - u_0)$$

Per cercare di valutare quantitativamente le performance delle simulazioni usiamo tre indici:

- PSNR: Peak-Signal-to-Noise Ratio, viene usato spesso nell'elaborazione di immagini per validare i risultati di processing in cui si ha

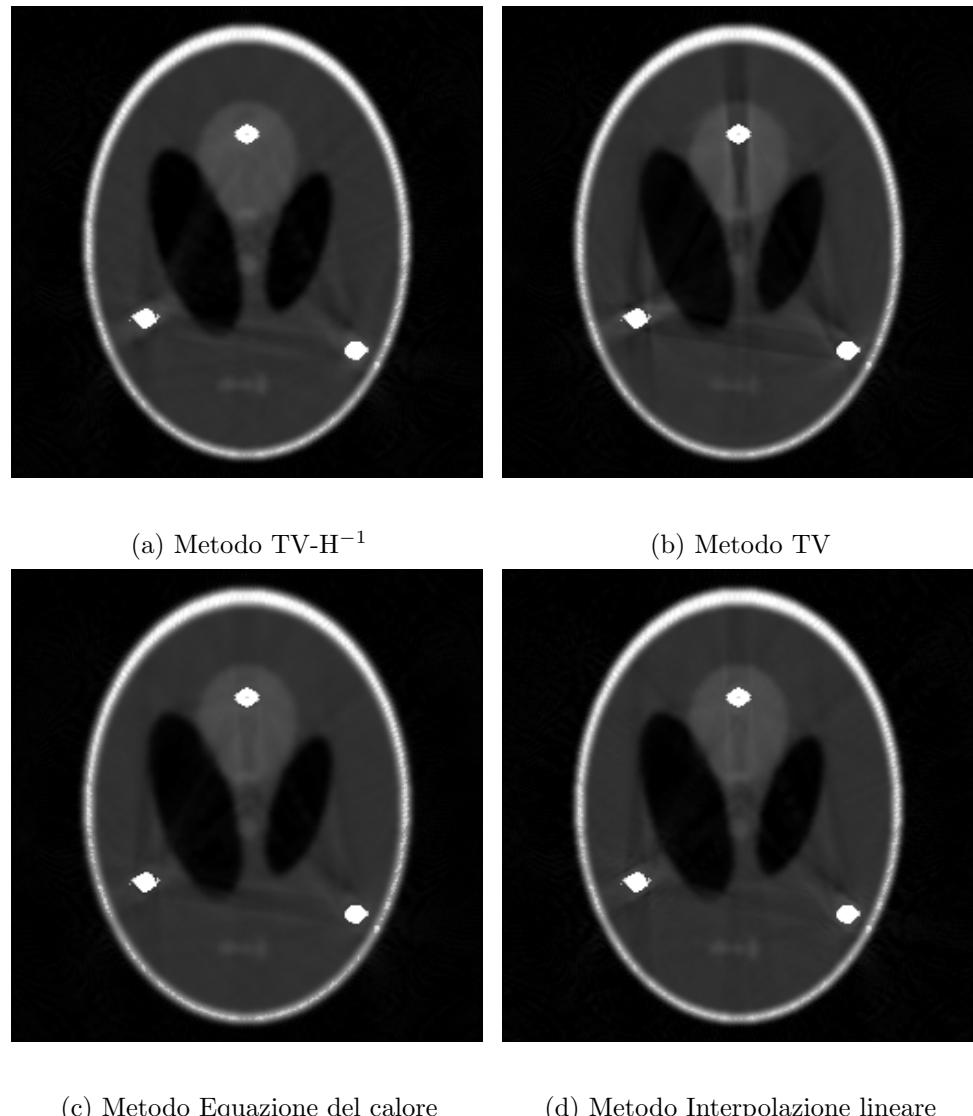


Figura 5.7: Risultati ottenuti con metodi alle differenze finite

disponibilità dell'immagine di riferimento.

$$PSNR = 10 \log_{10} \left(\frac{\max_D^2(u_{ref})}{\|u_{cor} - u_{ref}\|_{L^2}^2} \right)$$

Maggiore è il valore di tale indice, maggiore è la qualità globale del restauro.

- NCC: Normalized Correlation Coefficient

$$\frac{1}{|D|} \int_D \frac{(u_{cor} - \mu_{cor})(u_{ref} - \mu_{ref})}{\sigma_{cor}\sigma_{ref}}$$

dove u_{cor} è il fantoccio restaurato mentre u_{ref} è il fantoccio privo di artefatti. μ e σ sono rispettivamente la media e la deviazione standard. Nel migliore dei casi si ha che NCC assume valore uno. In tal caso u_{ref} e u_{cor} coincidono puntualmente.

- MSSIM: Mean Structure Similarity Index. Indici sviluppato nel 2003 da Zhou Wang (vedere [22]) al fine di valutare meglio la distanza tra due immagini da un punto di vista della percezione visiva umana. Gli autori definiscono l'indice puntuale SSIM dove le medie e le varianze vengono prese in finestre di dimensioni a scelta (nel nostro caso 8×8)

$$SSIM = \frac{(2\mu_{ref}\mu_{cor} + c_1)(2\sigma_{ref,cor} + c_2)}{(\mu_{ref}^2 + \mu_{cor}^2 + c_1)(\sigma_{ref}^2 + \sigma_{cor}^2 + c_2)}$$

e definiscono MSSIM come la media campionaria di SSIM su tutta l'immagine. Tanto più l'indice si avvicina ad avere valore unitario, migliore sarà il restauro.

	MSSIM	NCC	PSNR
NS	0.7919	0.9032	32.3090
TV	0.5571	0.8974	31.7668
HE	0.5720	0.8969	31.7530
TV-H ⁻¹	0.5984	0.8995	31.8278
LI	0.5898	0.8993	31.8298

Tabella 5.1: Tabella riassuntiva indici di valutazione restauro al variare dei metodi

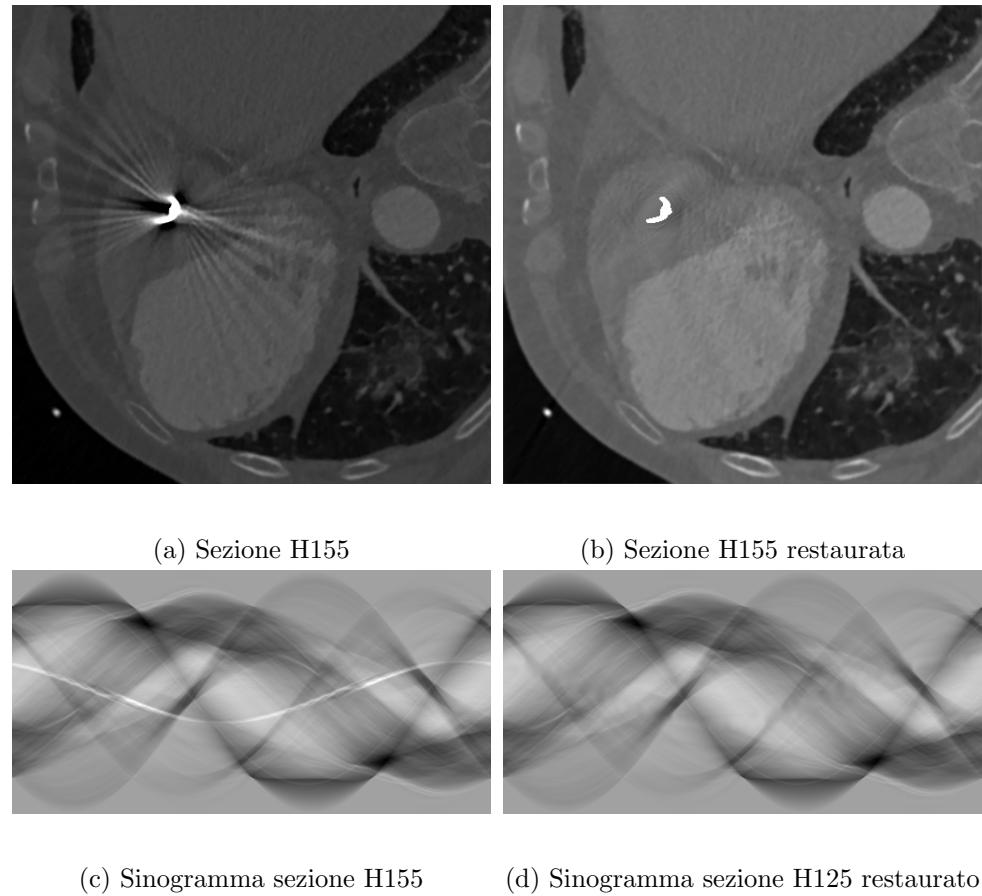


Figura 5.8: Restauro sezione H155

5.2 Dati clinici

Vediamo ora i risultati ottenuti su due sezioni del cuore di un paziente con presenza di peacemaker (fig. 5.8 e fig. 5.10). Nelle figure 5.9 e 5.11 invece vediamo un confronto tra i restauri ottenuti con i diversi metodi alle differenze finite elencati nella sezione precedente. Ovviamente in questo caso non possiamo avere una valutazione quantitativa della bontà dei nostri restauri in quanto non abbiamo disponibilità della TAC del paziente prima dell'impianto del peacemaker.

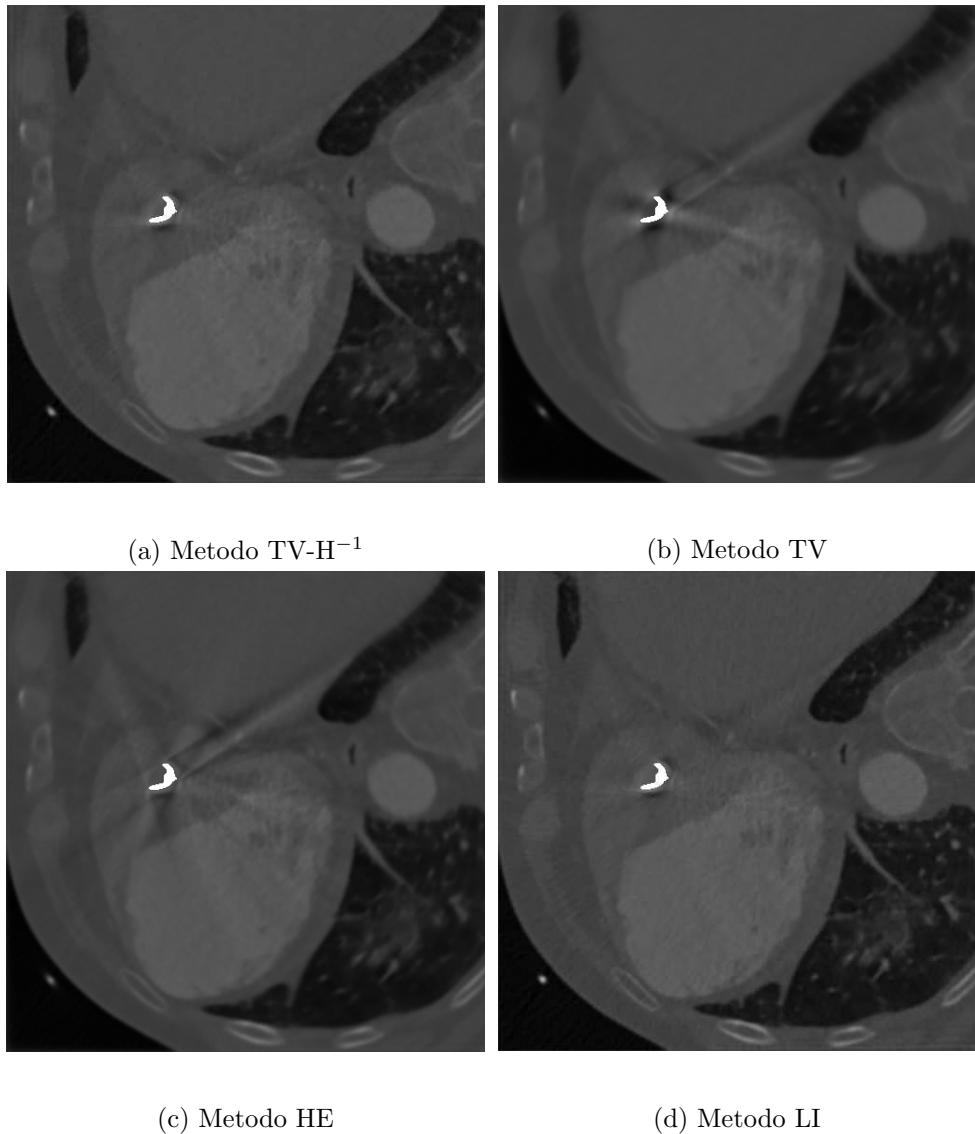


Figura 5.9: Sezione H155 restaurata con i metodi $TV-H^{-1}$, TV, HE e LI

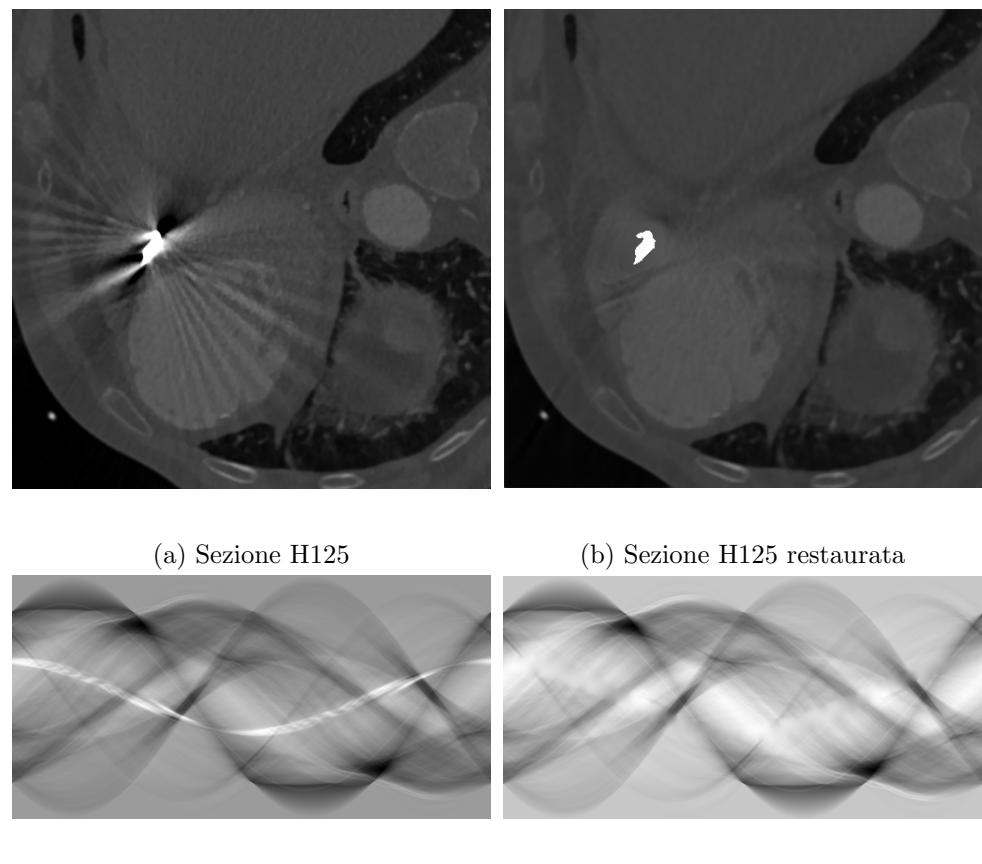


Figura 5.10: Restauro sezione H125

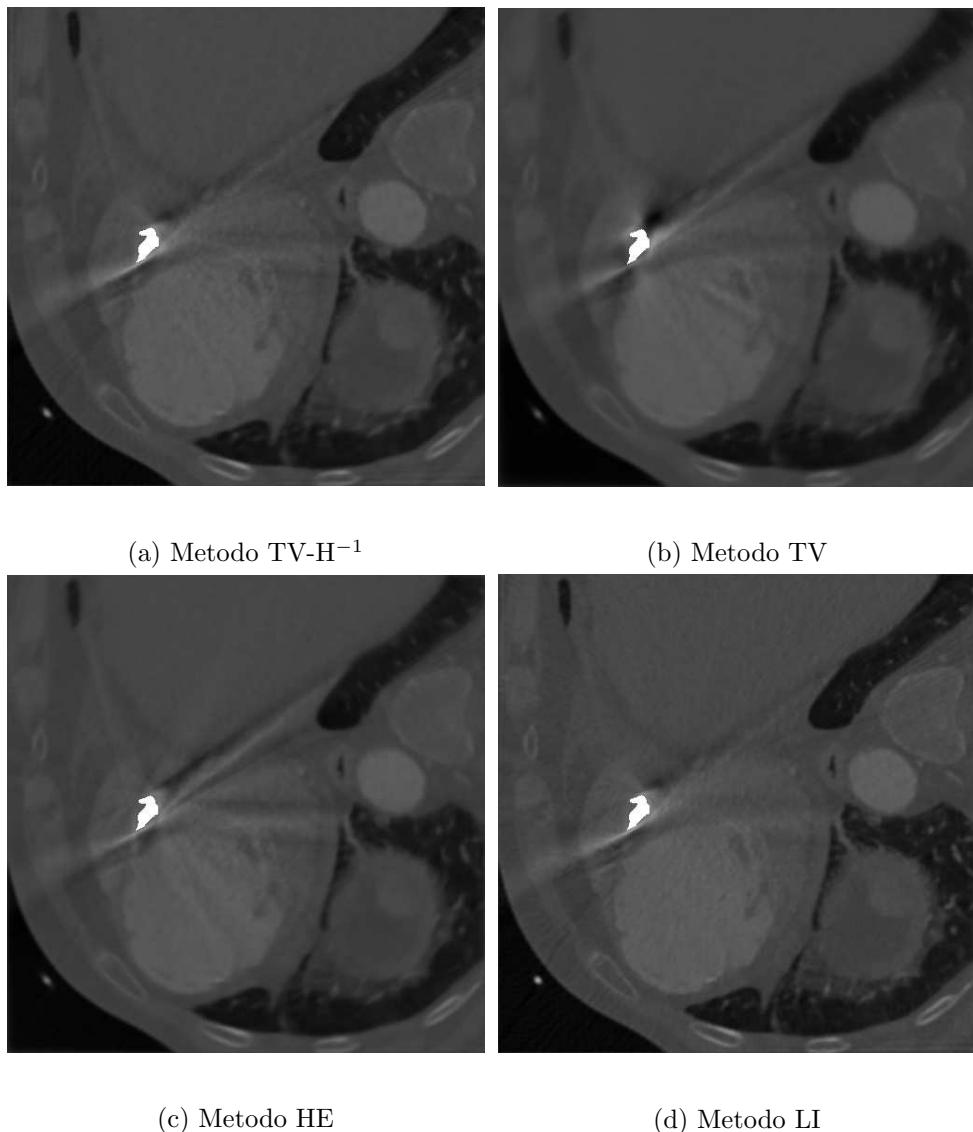


Figura 5.11: Sezione H125 restaurata con i metodi $\text{TV}-\text{H}^{-1}$, TV, HE e LI

5.3 Osservazioni

Riguardo i dati sintetici possiamo vedere come la qualità del restauro del nostro metodo sia migliore rispetto ai metodi alle differenze finite, e questo viene anche confermato dall'esempio di figura 5.7 e dai risultati della tabella 5.1 dove vediamo un netto distacco nei valori dell'indice MSSIM tra il nostro algoritmo e quelli implementati a differenze finite. A conferma della bontà del nostro metodo, si vede che nel caso si faccia il restauro su sinogramma originale (fig. 5.6), questo viene molto bene.

Nel caso clinico il nostro metodo si comporta meglio nella sezione H155 rispetto alla sezione H125 dove la traccia metallica è molto più larga. A differenza dei metodi a differenze finite i nostri risultati permettono di distinguere perfettamente i dettagli lontano della parte metallica, anche se presentano un lieve zona offuscata nell'intorno della parte metallica.

Come anticipato nel capitolo 3 i metodi di basati su PDE non sono atti a ricreare propriamente la texture. Osservando le superfici dei sinogrammi restaurati nel caso dei dati clinici ci accorgiamo infatti che l'interpolazione non ricrea propriamente la dinamica ad alta frequenza nel dominio di inpainting. Al fine di ricreare tale dinamica nel dominio si è provato, purtroppo con insuccesso, nell'applicazione dei seguenti metodi:

- Applicazione diretta dei metodi di sintesi di texture così come descritti nel capitolo 3 al sinogramma. Senza considerare l'elevato costo computazionale che richiedono, tali metodi producono scarsi risultati vista la continuità richiesta dal metodo di inversione FBP. Tali metodi non garantiscono la continuità e dunque invertendo si creano spiacevoli artefatti dovuti al fenomeno di Gibbs che interviene nel filtraggio che viene fatto tramite uso della trasformata di Fourier.
- Approccio combinato 1:
 1. Restauro di R con metodo di Bertalmio
 2. Partizioniamo l'immagine di R in intervalli equispaziati
 3. Per ognuno di questi insiemi valutiamo media e varianza dei dati all'infuori del dominio di inpainting
 4. All'interno del dominio di inpainting aggiungiamo un rumore gaussiano di media nulla e con varianza calcolata al punto precedente nei punti appartenente alla relativa classe.

Il motivo dell’insuccesso di tale metodo è il fatto che in realtà la texture nel nostro caso è indipendente dalla cosiddetta struttura geometrica del sinogramma.

- Approccio combinato 2:

1. Restauro di R con metodo di Bertalmio
2. Scomposizione sinogramma restaurato in una piramide laplaciana: il motivo per cui si è preferito prima fare il restauro e poi scomporre sta nel fatto che altrimenti per il fenomeno di Gibbs avremmo avuto un allargamento del dominio di inpaintng in ogni immagine della piramide. Facciamo notare a questo punto che l’ultimo strato della piramide è comunque restaurato, essendo quello delle basse frequenze dove il nostro algoritmo funziona sufficientemente bene come si può osservare dai dati sintetici.
3. Per ogni strato della piramide si calcola il gradiente, da cui si ricava l’angolo di inclinazione di $\nabla^\perp R$.
4. Per ogni punto dell’intorno del dominio si prende l’insieme di curve che consistono nell’intorno RNA di tale punto ([23]), ossia le proiezioni che a livello dell’immagine di partenza sono vicine alla proiezione presa in considerazione a livello del sinogramma.
5. Si prende la curva con tangente data dal gradiente perpendicolare
6. Si fa un interpolazione lineare lungo la direzione appena descritta

Purtroppo le direzioni considerate producono risultati del tutto sfasati.

- Approccio combinato 3:

1. Restauro di R con metodo di Bertalmio
2. Filtraggio del sinogramma restaurato con filtro usato per l’inversione
3. Scomposizione sinogramma filtrato restaurato in una piramide laplaciana
4. Applicazione sintesi di texture con metodo Bugeau, Bertalmio, Caselles e Sapiro ad ogni livello della piramide tranne l’ultimo

Conclusioni e prospettive future

In questo lavoro è stato presentata un implementazione di un metodo di restaturo di immagini digitali mai implementato finora per il restauro di TAC. Inoltre si è visto come l'implementazione ad elementi finiti, in aggiunta ad ottenere dei risultati soddisfacenti, presenta tempi computazionali molto ridotti rispetto alle implementazioni ad elementi finiti. Il metodo è stato provato dapprima su dati sintetici per provare il nostro metodo su un sinogramma semplificato rispetto a quelli clinici, ma soprattutto per poter avere anche una soluzione di riferimento come paragone. In seguito è stato poi usato con su immagini cliniche.

Il problema di restauro delle TAC nel caso di presenza di artefatti rimane ancora un problema aperto, infatti in letteratura non sono ancora presenti metodi che diano una risoluzione definitiva al problema. Si osserva però che la direzione presa dai metodi di interpolazione è in realtà molto promettente. In questa tesi abbiamo presentato una breve rassegna di metodi di restauro delle immagini che vengono applicati per il restauro delle TAC. In letteraura quello che spesso si osserva è l'uso di tali tecniche nel restauro di sinogrammi senza prendere in considerazione le proprietà della trasformata di Radon. Come prospettiva per il futuro uno dei punti cruciali su cui bisognerebbe focalizzarsi è quello di aggiungere un modello a priori per il restauro delle trasformate di Radon, anzichè trattare tali trasformate come generiche immagini. Le tecniche di inpainting sono tecniche che sono state sviluppate per ottenere risultati che siano accettabili alla vista umana mentre un restauro di una trasformata di Radon dovrà essere prima invertito tramite FBP ed in seguito dovrà essere accettabile alla vista umana, questo vuol dire che il focus di un metodo di restauro delle trasformate di Radon dovrebbe spostarsi

dalla vista umana al metodo di inversione, cosa che purtroppo in letteratura non viene quasi assolutamente considerata. Una possibile strada potrebbe essere quella di basarsi sui risultati di Exterior Tomography ([24],[25],[26]) dove gli autori presentano una trattazione teorica di un problema del tutto simile al nostro ossia considerano la trasformata di Radon di sezioni di oggetti reali dove vi sono parti del tutto impenetrabili dai raggi X. Gli autori definiscono la trasformata esteriore di Radon, ne caratterizzano nucleo ed immagine e trovano una base di polinomi ortogonali di Legendre per dominio e codominio di tale trasformata. Degli svariati risultati che vengono presentati da questi studi ancora nessuno ha provato ha provato ad integrarli nei metodi di interpolazione MAR. Un'altra essenziale integrazione che andrebbe fatta dovrebbe essere l'imposizione della condizione dei momenti. Queste condizioni forzano la trasformata ad avere la dinamica ad alta frequenza di cui abbiamo bisogno per la FBP. A conferma di questo vi è un importante teorema ([2]):

Teorema *Sia $g(x', \theta)$ una funzione regolare a supporto compatto. Tale funzione può essere vista come R_f per una funzione regolare e a supporto compatto f se g soddisfa le condizioni di simmetria e dei momenti.*

Inoltre nel caso si sviluppasse un metodo basato sull'uso di una trasformata con tali polinomi di Legendre vi sono già presenti metodi in letteratura di ricostruzione di un'immagine dai propri momenti tramite l'uso della trasformata di Fourier su cui ci si potrebbe basare.

Chiaramente queste considerazioni da sole non forniranno mai un numero di vincoli sufficiente a poter ricostruire i dati mancanti di un sinogramma, e dunque vanno sicuramente integrate con qualche metodo di interpolazione ad hoc al fine di poter sintetizzare una reale trasformata di Radon secondo i canoni teorici.

Un altro fronte su cui si potrebbe lavorare è quello di un pre-processing ottimale dell'immagine al fine di regolarizzare il più possibile la TAC con presenza di artefatti prima di effettuare il restauro così da ridurre rumore e alterazione prevedibili per via della ricostruzione con FBP. Un ulteriore punto, è quello di una segmentazione ottimale al fine da poter mantenere il maggior numero di dati non corrotti s cui basarsi per interpolare nel dominio di inpainting,

Un ultimo punto potrebbe quello di generalizzare tali metodi di inpainting in 3D potendo così prendere dati anche da TAC di sezione vicini nel

caso siano disponibili.

Bibliografia

- [1] Zdzislaw Meglicki, *Advanced scientific computing: course notes*. Indiana University, 2001.
- [2] Peter Kuchment and Gunther Uhlmann, *The Radon and X-Ray Transforms , lecture notes*. Research Training Group Inverse Problems and Partial differential Equations Department of Mathematics, University of Washington, NSF supported Summer School, 2012
- [3] Kailey Bolles, *Mathematics of medical imaging, inverting the Radon transform*. Whitman College, 2011.
- [4] Catherine Lemmens, David Faul, and Johan Nuyts, *Suppression of Metal Artifacts in CT Using a Reconstruction Procedure That Combines MAP and Projection Completion*. Katholieke Universiteit Leuven, 2009.
- [5] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, Frank Lenzen *Variational Methods in Imaging* edito Springer.
- [6] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*
- [7] Marco Marcon *Advanced topics on video processing, lecture notes* Politecnico di Milano
- [8] Tony F. Chan, Sung Ha Kang, Jianhong Shen *Euler's elastica and curvature based inpaintings* . CAM Report 01-12, Department of Mathematics, UCLA, Los Angeles, CA, 2001.
- [9] Faggiano E., Lorenzi T., Quarteroni A., *Metal Artifact Reduction in Computed Tomography Images by Variational Inpainting Methods*. MOX, Dipartimento di Matematica “F. Brioschi” Politecnico di Milano, Italy, 2013.

- [10] Carola-Bibiane Schonlieb, Andrea Bertozzi, *Unconditionally stable schemes for higher order inpainting*. Institute for Numerical and Applied Mathematics, Georg-August University of Gottingen, Germany and Department of Mathematics, UCLA, Los Angeles, USA 2000.
- [11] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image Inpainting*, SIGGRAPH 2000, pages 417-424
- [12] M. Bertalmio, A. L. Bertozzi, G. Sapiro, *Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting*. Proceedings of the International Conference on Computer Vision and Pattern Recognition , IEEE, Dec. 2001, Kauai, HI, volume I, pp. I-355-I362.
- [13] Harald Grossauer, *A Combined PDE and Texture Synthesis Approach to Inpainting*. Department of Computer Science University of Innsbruck, Austria,
- [14] Alexei A. Efros and Thomas K. Leung, *Texture Synthesis by Non-parametric Sampling*. University of California, Berkeley, 1999.
- [15] A. Crimini and P. Perez, *Object Removal by Exemplar-Based Sampling*. Microsoft Research Ltd., Cambridge, UK
- [16] Alexei A. Efros and William T. Freeman, *Image Quilting for Texture Synthesis and Transfer*. University of California, Berkeley, 2001.
- [17] Aurèle Bugeau, Marcelo Bertalmio, Vicent Caselles and Guillermo Sapiro *A Unifying Framework For Image Inpainting*. Institute For Mathematics And Its Applications, University of Minnesota, 2009.
- [18] Hitoshi Yamauchi, Jorg Haber, and Hans-Peter Seidel, *Image Restoration using Multiresolution Texture Synthesis and Image Inpainting*. Max-Planck-Institut fur Informatik, Saarbrucken, Germany.
- [19] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, *Simultaneous Structure and Texture Image Inpainting*. Institute For Mathematics And Its Applications, University of Minnesota, 2003.
- [20] Frédéric Hecht, Sylvian Auliac, Olivier Pironneau, Jacques Morice, Antoine Le Hyaric and Kohji Ohtsuka *FreeFem++ Guide*. Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, Paris

- [21] C.-B. Schonlieb, A. Bertozzi, M. Burger, L. He et al., *Image inpainting using a fourth-order total variation flow*, 2009.
- [22] Zhou Wang, Member, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity* 2003
- [23] Youngshin Kim, Seongbeak Yoon, and Juneho Yi, *Effective sinogram-inpainting for MAR in x-ray ct images* Computer Vision Laboratory, Sungkyunkwan University, Korea, 2010
- [24] Adel Faridani, *Introduction to the Mathematics of Computed Tomography*, Inside Out: Inverse Problems MSRI Publications Volume 47, 2003
- [25] Eric Todd Quinto, *Exterior and limited-angle tomography in non-destructive evaluation*, Department of Mathematics, Tufts University, Medford, MA, USA, 1998
- [26] Eric Todd Quinto, *Local algorithms in exterior tomography*, Department of Mathematics, Tufts University, Medford, MA, USA, 2004
- [27] Wilson Au, Ryo Takei, *Image Inpainting with the Navier-Stokes Equations*.
- [28] Esther Meyer, Rainer Raupach, Bernhard Schmidt, Andreas H. Mahnken, Marc Kachelrieß, *Adaptive Normalized Metal Artifact Reduction (ANMAR) in Computed Tomography*.
- [29] Maisano G., Micheletti S., Perotto S., Bottasso C.L., *On some new recovery-based a posteriori error estimators*.
- [30] Koji Kobayashi, Atsushi Katsumata, Koichi Ito, Takafumi Aoki, *A Practical Method to Reducing Metal Artifact for Dental CT Scanners*.

Elenco delle figure

1.1	Sistema di riferimento ruotato	10
1.2	Possibili scelte di filtro per le retroproiezioni	14
1.3	Superficie fantoccio Shepp-Logan	15
1.4	Superficie $R(\theta, x')$	16
1.5	Fantoccio Shepp Logan e relativo sinogramma	16
1.6	Trasformata di Radon filtrata con $ k' $	17
1.7	Immagine Trasformata di Radon filtrata con $ k' $	17
1.8	Confronto tra $R(\theta = 20^\circ, \cdot)$ e $C(\theta = 20^\circ, \cdot) + costante$	18
1.9	Retroproiezione filtrata per $\theta = 20^\circ$	18
1.10	Ricostruzioni parziali con FBP	19
1.11	Ricostruzione finale con 180 proiezioni	20
1.12	Immagini esemplificative funzionamento TAC	23
1.13	HU per diverse componenti corporee	25
1.14	TAC con presenza di artefatti e relativo sinogramma	26
2.1	Campionamento e quantizzazione: (a) Immagine continua. (b) valori assunti da f lungo il segmento AB. (c) campionamento e quantizzazione. (d) digitalizzazione di f lungo il segmento AB. [6]	30
2.2	(a) Immagine continua. (b) Immagine digitale. [6]	31
2.3	Funzioni di trasferimento filtri passa basso e passa alto	36
2.4	37
2.5	Estrazione contorni con tresholding gradiente e zero-crossing del laplaciano	40
2.6	Grafici pdf distribuzioni casuali di errore	44
2.7	Esempio di analisi con piramide Gaussiana e Laplaciana	45

3.1	Confronto tra un immagine restaurata con sintesi di texture (destra) e PDE based method (centro). [13]	48
3.2	Ogni immagine si può scomporre in parte strutturale (geometrica) e texture, i metodi basati sulle EDP funzionano bene per la parte strutturale, mentre i metodi di sintesi per la parte di texture	49
3.3	Esempio di non unicità del restauro	52
3.4	TV inpainting e Metodo della curva elastica a confronto	57
3.5	<i>Il problema di inpainting secondo l'approccio di Bertalmio, et. al.. Il laplaciano viene propagato lungo le direzioni delle isolinee. (Le curve verticali rappresentano $\nabla^\perp u$)</i>	59
3.6	Vettore tangente e vettore normale lungo le curve di livello	62
3.7	PDE inpainting e sintesi di texture. prima riga: immagine da restaurare, seconda riga: scomposizione dell'immagine nelle due parti, terza riga: inpainting, quarta riga: sovrapposizione	74
4.1	Esempio di trasferimento di texture [16]	78
4.2	Esempio uso filtro mediano	79
4.3	Riepilogo grafico struttura metodi di interpolazione	81
4.4	Fasi algoritmo meshing	85
4.5	Triangolazione finale	86
5.1	Creazione dati sintetici	90
5.2	Restauro fantoccio con singola componente metallica	91
5.3	Restauro fantoccio con due componenti metalliche	92
5.4	Restauro fantoccio con tre componenti metalliche	93
5.5	Restauro fantoccio con quattro componenti metalliche	94
5.6	Fantocci restaurati da sinogrammi originali	95
5.7	Risultati ottenuti con metodi alle differenze finite	96
5.8	Restauro sezione H155	98
5.9	Sezione H155 restaurata con i metodi $TV-H^{-1}$, TV, HE e LI	99
5.10	Restauro sezione H125	100
5.11	Sezione H125 restaurata con i metodi $TV-H^{-1}$, TV, HE e LI	101

Elenco delle tabelle

3.1	Tabella riassuntiva analogia tra le grandezze caratteristiche dei fluidi e delle immagini	60
3.2	Algoritmo di Efros	67
3.3	Algoritmo di Criminsi	69
3.4	Algoritmo di Bugeau, Bertalmò, Caselles e Sapiro 1	70
3.5	Algoritmo di Bugeau, Bertalmò, Caselles e Sapiro 2	72
4.1	Errori di ricostruzione	77
5.1	Tabella riassuntiva indici di valutazione restauro al variare dei metodi	97

Ringraziamenti

Desidero innanzitutto ringraziare la Professoressa Simona Perotto e la Dott.ssa Elena Faggiano per avermi seguito costantemente durante lo svolgimento di questo lavoro, per i consigli e l'aiuto nella stesura della tesi e per l'opportunità di lavorare su questi affascinanti argomenti.

Un grazie di cuore alla mia famiglia, per essere avermi supportato instancabilmente tutti questi anni.

Ringrazio nuovamente il professor Zanchetta per avermi indicato la strada ed aver acceso in me l'interesse. Le sarò sempre in debito.

Un ringraziamento particolare va al mio caro Zio Mokkonen, non ti dimenticheremo mai.

Ringrazio Massi, per avermi tirato fuori dalla triennale.

Un super ringraziamento a Flavio ed Igor per la pazienza ed il supporto.

Grazie a tutti i compagni di studio di tutti questi anni, troppi per essere menzionati uno per uno.

Grazie ai miei amici fraterni, per la pazienza e per esserci.

Un ringraziamento speciale va alla famiglia allargata: zie, zii, fratelli, sorelle, cugini per aver gioito con noi nei momenti felici ed aver condiviso tristi occasioni. Grazie per il supporto.