

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

PRINCIPIO DI PENALIZZAZIONE UNIFORME
NEL PROBLEMA DI SUPER-RISOLUZIONE DI
IMMAGINI

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.ssa
Fabiana Zama

Presentata da:
Rossana Zanetti

III Sessione
Anno Accademico 2019/2020

Indice

Introduzione	iii
1 Super-Risoluzione di Immagini	1
1.1 Digitalizzazione di un'immagine	1
1.2 Sistema di formazione dell'immagine	2
1.3 Tecniche di Super-Risoluzione ad Immagine Singola	3
1.4 Approcci non iterativi: Interpolazione	5
1.5 Tecniche Reconstruction-based	6
2 Regolarizzazione	8
2.1 Variazione Totale	9
2.2 Scelta del parametro di regolarizzazione	10
2.3 Cenni storici	11
2.4 Principio di Penalizzazione Uniforme	12
2.5 Principio di Penalizzazione Uniforme locale	13
3 Algoritmi di Ottimizzazione	16
3.1 Richiami sui Problemi di Ottimizzazione	16
3.2 Algoritmo Primale-Duale Ibrido	18
3.3 Algoritmo FISTA Chambolle	20
4 Algoritmo del Gradiente Prossimale di Bregman	25
4.1 Prerequisiti	26
4.2 Caratteristiche generali e Applicazioni	28
5 Risultati Numerici	30
5.1 Costruzione del problema test	31
5.2 Parametri di valutazione	32
5.3 Regolarizzazione globale	33
5.3.1 Immagine Brain	33
5.3.2 Immagine Geometric	39

5.4	Regolarizzazione locale	44
5.4.1	Immagine Brain	45
5.4.2	Immagine Geometric	50
5.5	Up-sampling 3 per immagine Brain	56
	Conclusioni	59
	Bibliografia	60

Introduzione

In ambiti come lo studio delle immagini mediche, l'elaborazione delle fotografie satellitari o l'analisi delle termoinmagini, i dettagli assumono una grande rilevanza. L'aumento della risoluzione spaziale dell'immagine è il processo utilizzato per evidenziare più chiaramente i dettagli in essa contenuti. Durante il processo con cui una scena continua reale viene trasformata in un'immagine digitale, intervengono diversi fattori che creano fenomeni di sfocamento e *rumore*, rendendo l'immagine meno nitida. A partire da una o più immagini a bassa risoluzione è possibile, però, operare delle rielaborazioni che ne migliorino la qualità. Si tratta delle tecniche di Super-Risoluzione di immagini, che consistono in metodi volti a ricostruire un'immagine ad alta risoluzione (*High Resolution*, HR) utilizzando le informazioni contenute nei dati LR (*Low Resolution*). Il problema di Super-Risoluzione di immagini è oggetto di grande interesse in molte applicazioni, tra cui quelle sopra citate, e può essere trattato seguendo diversi approcci. Negli approcci basati sulla ricostruzione (*reconstruction-based*) si risolve il problema formulandolo come un problema di ottimizzazione, derivante dalla discretizzazione di un modello continuo che descrive il sistema di formazione di un'immagine digitale. Tale modello presenta un sistema lineare, tipicamente mal condizionato, in cui il termine noto è l'immagine a bassa risoluzione e l'incognita rappresenta l'immagine HR che si vuole ricostruire. Si tratta, quindi, di un problema inverso mal posto, ovvero tale da presentare una soluzione molto sensibile alle perturbazioni nei dati. Per questo motivo, nelle tecniche SR *reconstruction-based* viene operata una regolarizzazione, ovvero una tecnica che impone regolarità alla soluzione calcolata. All'interno del problema di ottimizzazione compare, dunque, un termine di regolarizzazione, composto da un operatore discreto e da un parametro di regolarizzazione. Centrale nel presente lavoro di tesi è stato lo studio di una procedura di determinazione automatica del parametro di regolarizzazione per un problema di Super-Resolution, chiamata Principio di Penalizzazione Uniforme. Tale metodo fornisce una regola di scelta del parametro basata sulla volontà di attribuire lo stesso peso, all'interno del problema di ottimizzazione, ai due termini che lo compongono: un termine

che misura la consistenza dei dati e un termine che riguarda la regolarizzazione. Poichè una regolarizzazione effettuata con un singolo parametro globale può non essere sufficiente a rendere la soluzione più stabile, soprattutto se l'immagine presenta molte variazioni tra i pixel, il progetto di questa tesi si è orientato sull'estensione del Principio di Penalizzazione Uniforme al caso di regolarizzazione con parametri multipli, calibrati a livello locale. Questo ha permesso di ottenere parametri di regolarizzazione differenti, a seconda delle variazioni presenti tra i pixel dell'immagine, ed ottenere una ricostruzione più fedele. Di fondamentale importanza è stata la valutazione dei diversi elementi presenti nel modello discreto che descrive il problema di ricostruzione. I test numerici sono stati svolti a partire da due diverse immagini in bianco e nero e con l'utilizzo di due algoritmi noti di ottimizzazione, che sono stati opportunamente modificati per includere le procedure sopra citate di scelta del parametro di regolarizzazione globale e locale.

Il primo capitolo introduce il processo di formazione di un'immagine digitale e la sua modellizzazione, che porta alla formulazione del problema di Super-Risoluzione a Singola Immagine. Il capitolo contiene poi una discussione sui diversi approcci possibili per risolvere tale problema. Nel secondo capitolo, che riguarda la regolarizzazione, vengono presentati l'operatore di Variazione Totale e il criterio di Penalizzazione Uniforme per la scelta del parametro globale e locale. Nel capitolo 3, dopo alcuni richiami sui problemi di ottimizzazione, vengono descritti gli algoritmi utilizzati nei test numerici, con le relative modifiche per l'utilizzo del criterio di scelta del parametro di regolarizzazione precedentemente introdotto. Il quarto capitolo contiene un'analisi solo teorica di un metodo di risoluzione di un problema di ottimizzazione con funzione obiettivo non convessa, utile per problemi di Super-Resolution più generali. Infine, nel quinto capitolo sono riportati tutti i risultati trovati durante le prove numeriche, in cui è emerso che uno dei due algoritmi è più efficiente dell'altro e che le forti variazioni tra i pixel delle due immagini considerate rendono poco rilevante la differenza tra regolarizzazione globale e locale.

Capitolo 1

Super-Risoluzione di Immagini

1.1 Digitalizzazione di un'immagine

Il processo di acquisizione di un'immagine avviene attraverso strumenti hardware dotati di sensori e convertitori. Le apparecchiature fotografiche convertono una scena reale continua in una matrice di valori numerici, in cui ogni elemento è detto pixel e rappresenta una precisa porzione della scena originale. Nel caso di un'immagine in bianco e nero, si parla di una matrice 2-dimensionale, in cui ogni pixel ha un valore compreso nell'intervallo $[0, 255]$, a seconda della sua intensità di grigio: il valore 0 corrisponde ad un pixel nero, il valore 255 ad un pixel bianco.

Il processo di digitalizzazione di un'immagine è sempre imperfetto e ciò è dovuto a diversi fattori. Tra gli elementi che concorrono alla formazione di degradazioni sull'immagine catturata vi sono le limitazioni fisiche degli strumenti utilizzati, come la bassa densità dei sensori o la dimensione finita dell'obiettivo, gli effetti atmosferici e le distorsioni ottiche. In più, durante il processo di digitalizzazione, possono verificarsi effetti di sfocamento, detto *blur* ottico, e di rumore casuale. L'input di un sistema di cattura di un'immagine è una scena naturale continua, approssimata tramite segnali che possono essere contaminati da turbolenze atmosferiche. Questi segnali vengono convertiti per formare l'immagine ad alta risoluzione desiderata. Essa poi incorre in fenomeni di sfocamento e rumore e viene sottocampionata in pixel, per l'azione di sensori come i *CCD detectors (Coupled Charged Device)*, ovvero dispositivi a scorrimento di carica. In molte applicazioni e situazioni reali sono richieste immagini ad alta risoluzione, in cui i dettagli siano ben visibili e i contorni definiti. Basti pensare all'elaborazione di immagini mediche, termiche o satellitari, o ai processi di riconoscimento automatico di oggetti in una fotografia. La risoluzione di un'immagine descrive la quantità

di dettagli in essa visibili e può essere classificata in molti modi diversi: risoluzione per pixel, risoluzione spaziale, spettrale, temporale e radiometrica.

Definizione 1.1. (Risoluzione Spaziale) Un'immagine digitale è costituita da piccoli elementi chiamati pixel. Per risoluzione spaziale si intende la densità di pixel in un'immagine e si misura in pixel per unità di area.

Come detto in precedenza, la risoluzione spaziale di un'immagine risulta limitata in primo luogo dagli strumenti di acquisizione e digitalizzazione fotografica. Migliorare la qualità o la quantità dei sensori utilizzati può, tuttavia, risultare costoso e insufficiente. Inoltre, vanno tenuti in considerazione gli effetti di sfocamento ottico, i limiti fisici dei sensori, l'aberrazione delle lenti. Per questo, si cerca di accettare e modellizzare le degradazioni, per post-processare le immagini catturate. Oltre al miglioramento delle apparecchiature fisiche, vi sono quindi algoritmi che tentano di eliminare le degradazioni di un'immagine, scambiando il costo economico con il costo computazionale.

1.2 Sistema di formazione dell'immagine

Quando si parla di un sistema per la formazione di un'immagine digitale si intende un sistema fisico di trasmissione di segnali. Per le immagini 2-dimensionali il segnale può essere una funzione di due variabili spaziali. In molte circostanze, è possibile fare una modellizzazione continua e lineare del sistema di formazione di immagini, indicando ad esempio con (s, t) il vettore delle coordinate spaziali e con $x(s, t)$ il segnale di ingresso del sistema. Se $g(u, v)$ rappresenta il segnale d'uscita, la relazione tra oggetto reale ed immagine si esprime con

$$g(u, v) = \iint k(u, s, v, t)x(s, t)dsdt \quad (1.1)$$

dove $k(u, s, v, t)$ è detta Point Spread Function, o risposta in impulso del sistema. L'effetto della PSF sull'oggetto x è il *blurring*, ovvero uno sfocamento che rende i dettagli meno nitidi. Molto spesso i sistemi di formazione di immagini hanno l'interessante proprietà di essere spazio invarianti. In termini più precisi, la forma dell'immagine di un punto luminoso non dipende dalla posizione del punto ed il sistema è invariante per traslazioni. Si ha quindi

$$k(u, s, v, t) = H(u - s, v - t)$$

e la Point Spread Function viene identificata con la funzione H . Il modello di formazione di immagini risulta, allora, descritto da

$$g(u, v) = \int \int H(u - s, v - t)x(s, t)dsdt \quad (1.2)$$

Ne segue che l'immagine $g(u, v)$ è il prodotto di convoluzione di $x(s, t)$ e H . Si suppone che la PSF soddisfi:

1. $H(u, v) \geq 0$;
2. $\int H(u, v)dudv = 1$.

Nel modello continuo introdotto, $g(u, v)$ è la distribuzione spaziale della grandezza fisica che rappresenta l'immagine dell'oggetto $x(s, t)$. Tale grandezza fisica è un segnale che viene convertito in segnale digitale, solitamente tramite i *CCD detectors* sopra citati. Essi sono array di sensori, in cui ogni sensore fornisce una misura locale del valore dell'immagine $g(u, v)$.

Per discretizzare il modello di formazione dell'immagine, si deve discretizzare, innanzitutto, il prodotto di convoluzione con la PSF. Inoltre, è necessario introdurre un'approssimazione del *noise*, il rumore casuale che colpisce l'immagine. Il modello più utilizzato per il rumore è quello del *noise* bianco gaussiano, che verrà mostrato più in dettaglio nel capitolo relativo ai risultati numerici.

1.3 Tecniche di Super-Risoluzione ad Immagine Singola

Parlare di Super-Risoluzione (SR) significa parlare di tecniche numeriche finalizzate alla ricostruzione di immagini ad alta risoluzione (HR) a partire da una o più immagini a bassa risoluzione (LR) osservate. Per le sue potenzialità, il problema di Super-Risoluzione è di grande interesse in molte applicazioni. Il processo di ricostruzione SR può prevedere l'utilizzo di una singola immagine a bassa risoluzione come punto di partenza, o di più *frames* a bassa risoluzione raffiguranti la medesima scena. Nel primo caso, che è quello analizzato in questo lavoro di tesi, si parla di problema di Super-Resolution a Singola Immagine, ovvero problema SISR. Come riportato in [2], il lavoro pionieristico di Tsai e Huang sulla Super-Resolution presentava la formulazione di una tecnica SR sul dominio di frequenza delle immagini e

sulle proprietà della Trasformata di Fourier. Gli approcci relativi al dominio di frequenza, tuttavia, risultavano molto limitativi, poichè efficaci solo nel modello di osservazione di un'immagine che essi erano in grado di trattare. Le difficoltà di questi metodi hanno quindi portato alla formulazione di diversi approcci in cui la SR veniva eseguita sul dominio spaziale delle immagini. La ricerca attuale utilizza soprattutto tecniche di questo tipo, per via della loro flessibilità nel modellizzare tutti i tipi di degradazione di un'immagine. Nelle tecniche SR a Singola Immagine, la ricostruzione si basa sulla risoluzione di un problema inverso discretizzato tramite un sistema lineare. Il problema SISR può essere descritto a parole come segue:

Sia data un'immagine a bassa risoluzione (LR), di dimensioni $M \times N$ e sia r un intero maggiore di 1. Ricostruire un'immagine ad alta risoluzione (HR), che rappresenti la stessa scena di quella LR, ma che abbia dimensioni $rM \times rN$.

Da questa formulazione si passa alla discretizzazione del modello continuo (1.2) presentato nel paragrafo relativo al sistema di formazione immagini. Sia $G \in \mathbb{R}^{M \times N}$ l'immagine a bassa risoluzione osservata e sia $X \in \mathbb{R}^{(rM \times rN)}$ l'immagine ad alta risoluzione che si vuole ricostruire. Il modello matematico per la formazione di un'immagine LR può essere scritto nel modo seguente:

$$g = SHx + \eta, \quad (1.3)$$

dove $g \in \mathbb{R}^m$ e $x \in \mathbb{R}^n$ sono, rispettivamente, vettorizzazioni delle matrici G ed X , quindi $m = MN$, $n = r^2MN$. La matrice H rappresenta l'operatore di *blur* discreto che agisce sull'immagine x ed S è l'operatore di down-sampling, che mappa i dati HR nei dati LR. Questo ridimensionamento viene eseguito attraverso il fattore di *up-sampling* r citato sopra. L'operatore di *blur* è, invece, la discretizzazione della Point Spread Function (PSF), che modella lo sfocamento ottico dovuto alla dimensione finita dell'obiettivo della fotocamera e che è un operatore mal-posto. Il vettore $\eta \in \mathbb{R}^m$ rappresenta il rumore additivo. Lo scopo delle tecniche SR è risalire all'immagine x , partendo dai dati a bassa risoluzione: si tratta di un problema inverso. Le matrici coinvolte sono molto sparse e il sistema lineare è tipicamente mal condizionato.

Definizione 1.2. (Mal-posizione secondo Hadamard) Un problema si dice mal-posto secondo Hadamard se non ammette soluzione, oppure ne ammette più di una e se la soluzione non dipende con continuità dai dati.

Inoltre, nei sistemi reali di immagine, queste matrici sono sconosciute e devono essere stimate a partire dalle osservazioni LR disponibili, cosa che rende il problema ancora più mal-condizionato. Per queste ragioni, come si vedrà meglio nella trattazione successiva, risulta necessaria una regolarizzazione.

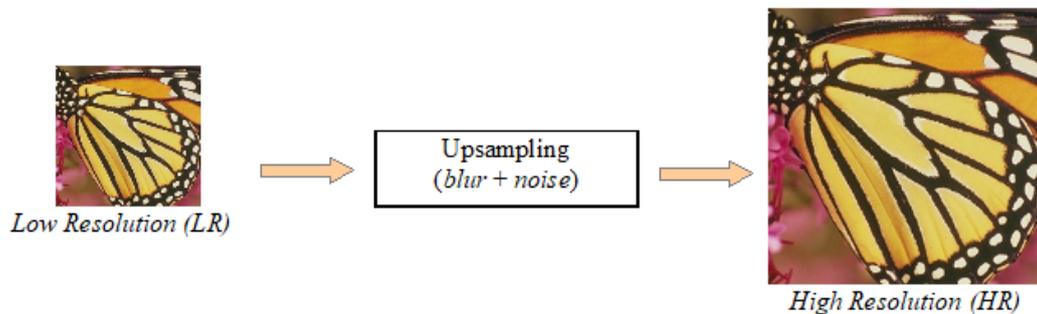


Figura 1.1: Modello SISR: Ricostruire immagine HR a partire dal dato LR

1.4 Approcci non iterativi: Interpolazione

Il metodo più semplice, non iterativo, per la ricostruzione SR sul dominio spaziale è quello dell'interpolazione a singola immagine. Si tratta della tecnica più immediata e prevede tre fasi:

- Step 1** Registrazione dell'immagine a bassa risoluzione;
- Step 2** Interpolazione per aumentarne le dimensioni;
- Step 3** Rimozione dello sfocamento e del rumore.

L'immagine a bassa risoluzione catturata viene posta nella griglia di un'immagine ad alta risoluzione di dimensioni maggiori e si utilizza un metodo di interpolazione per riempire le posizioni dei pixel mancanti. Poi, all'immagine HR ottenuta vengono tolti il *blur* ed il rumore con un algoritmo di deconvoluzione.

Tutti gli approcci basati sull'interpolazione sono poco complessi e molto intuitivi. Tuttavia, poichè non viene aggiunta alcuna informazione all'immagine, la loro qualità è limitata, a causa anche della naturale mal-posizione del problema SR. Inoltre, in questi approcci l'errore di registrazione può arrivare ad espandersi molto e la mancanza di una regolarizzazione a priori rischia di creare problemi. In questo lavoro di tesi, l'interpolazione è stata utilizzata come confronto per gli altri metodi presi in esame.

Altri algoritmi, non approfonditi in questa tesi, per la ricostruzione di immagini prevedono approcci statistici, in cui sia l'immagine HR che i movimenti tra gli input LR sono visti come variabili aleatorie, o approcci basati su esempi.

(a) Immagine LR, 80×80 (b) Immagine ricostruita, 320×320

Figura 1.2: Esempio di interpolazione bicubica con upfactor 3

1.5 Tecniche Reconstruction-based

Tra i diversi approcci presenti in letteratura per la Super-Risoluzione di immagini, vi sono quelli basati sulla ricostruzione. Si tratta di tecniche finalizzate alla risoluzione di un problema di ottimizzazione, in cui la funzione obiettivo è la somma di due termini: un termine di *fit*, o consistenza dei dati, e un termine di regolarizzazione. Il problema si può quindi formulare nel modo seguente:

$$\min_x \{F(x) + G(x)\}, \quad (1.4)$$

dove F è il termine di consistenza dei dati e G è la parte relativa alla regolarizzazione. Di fondamentale importanza è fissare un opportuno modello di degradazione che esprima il rapporto tra il dato HR e il dato LR, per stabilire il termine F , e considerare un opportuno funzionale G che induca stabilità sulla soluzione finale. Regolarizzare significa proprio imporre condizioni di regolarità sulla soluzione calcolata e, in questo modo, sopprimere alcune delle componenti della soluzione corrotte dal rumore. Un problema da tenere in considerazione è il fatto che, nelle applicazioni reali, non è sempre possibile conoscere l'effettiva sorgente del rumore presente in un'immagine, né modellizzare perfettamente i parametri che descrivono le sue degradazioni. Il rumore che viene utilizzato in molti problemi test è un rumore gaussiano, che non sempre corrisponde al rumore effettivo dell'immagine. Per questo motivo, molti sforzi della ricerca sulle tecniche per la ricostruzione SR sono volti ad aumentare la loro "robustezza".

Quando il problema di minimo derivante da (1.3) contiene un rumore di tipo gaussiano, il naturale termine di consistenza dei dati che viene utilizzato è il quadrato della norma 2 del residuo. Il problema è formulato quindi come segue:

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \| SHx - g \|_2^2 + G(x) \right\}. \quad (1.5)$$

Un'altra possibilità per il termine di consistenza dei dati, utilizzata in presenza di rumore poissoniano o di altro tipo, è la funzione di Kullback-Liebler, una misura della prossimità di due vettori Ax e b non negativi molto utilizzata, descritta in [3]. Tale divergenza si scrive come:

$$d(b, Ax) := \frac{1}{2} \sum_{i=1}^m \left\{ b_i \log \frac{b_i}{(Ax)_i} + (Ax)_i - b_i \right\}. \quad (1.6)$$

In questo secondo caso, il problema di minimo da risolvere prende in considerazione una funzione obiettivo data dalla somma della funzione KL e del termine di regolarizzazione ed ha la seguente forma:

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} d(g, SHx) + G(x) \right\} \quad (1.7)$$

Lo svantaggio della funzione $F = d(g, SHx)$ è quello di non avere gradiente globalmente Lipschitz-continuo. Tuttavia, come si vedrà in seguito, tale funzione soddisfa una condizione detta L-smad, che rende possibile la formulazione di generalizzazioni per i metodi di ottimizzazione. In questo lavoro di tesi è stato approfondito il caso di rumore gaussiano e di termine di consistenza dei dati in norma 2, ma il capitolo 4 presenta un algoritmo utilizzabile per un problema con funzione obiettivo data dalla Kullback-Liebler sommata al termine di regolarizzazione.

Capitolo 2

Regolarizzazione

Dalla descrizione della mal-posizione del problema SISR, emerge il rischio di una soluzione troppo sensibile alle perturbazioni sui dati. Per imporre condizioni di stabilità alla soluzione di un problema inverso mal posto e sopprimere le sue componenti corrotte dal rumore, si utilizzano i metodi di regolarizzazione. I metodi di filtraggio spettrale, ad esempio, sono metodi di regolarizzazione che utilizzano la decomposizione in valori singolari delle matrici presenti nel sistema. Tra questi metodi, si possono ricordare la decomposizione a valori singolari troncata (TSVD) ed il metodo di Tikhonov. Quest'ultimo, prevede di determinare la soluzione regolarizzata di un problema inverso mal posto $Ax = b$ come soluzione del problema di minimo

$$\min_x \frac{1}{2} \| Ax - b \|^2 + \lambda \| Lx \|^2 \quad (2.1)$$

dove L viene detto operatore di regolarizzazione e $\lambda > 0$ parametro di regolarizzazione.

L'operatore utilizzato in questo studio per la regolarizzazione del problema di Super-Risoluzione a Singola Immagine è il funzionale di Variazione Totale, che viene descritto in dettaglio nel presente capitolo. Come visto nel metodo di Tikhonov, il termine di regolarizzazione presenta anche un parametro λ , il cui ruolo è fondamentale per evitare che la soluzione sia troppo o troppo poco regolarizzata. In questo capitolo, viene introdotta la definizione di Variazione Totale e viene presentato il criterio utilizzato per la scelta automatica del parametro di regolarizzazione, ovvero il principio di Penalizzazione Uniforme, nel caso di regolarizzazione globale e, soprattutto, locale.

2.1 Variazione Totale

La Variazione Totale è un funzionale di regolarizzazione efficace, che permette di stabilizzare anche funzioni con ampie oscillazioni e discontinuità "a salto". Seguendo la definizione di [6], se $\Omega_{[a,b]}$ è l'insieme delle partizioni σ di $n + 1$ punti dell'intervallo $[a, b]$, ovvero $\sigma = \{a = x_0, \dots, x_n = b\}$, allora

Definizione 2.1. Sia $f : [a, b] \rightarrow \mathbb{R}$, si dice Variazione Totale di f su $[a, b]$ l'espressione

$$TV(f) = \sup \sum_i |f(x_i) - f(x_{i-1})|, \quad (2.2)$$

dove l'estremo superiore è preso su tutte le partizioni σ .

Se f è liscia, moltiplicando e dividendo il termine di destra per $\Delta x_i = x_i - x_{i-1}$ e prendendo il limite per $\Delta x_i \rightarrow 0$, si ottiene la rappresentazione

$$TV(f) = \int_a^b \left| \frac{df}{dx} \right| dx, \quad (2.3)$$

che in due dimensioni diventa

$$TV(f) = \int_a^b \int_a^b |\nabla f| dx dy. \quad (2.4)$$

Il termine $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ indica il gradiente di f .

Ponendo $G(x) = \lambda TV(x)$ nel problema (1.4), si ha che il problema SISR diventa

$$x = \arg \min_x \left\{ \frac{1}{2} \| SHx - g \|_2^2 + \lambda TV(x) \right\}, \quad (2.5)$$

nel primo caso, oppure

$$x = \arg \min_x \left\{ \frac{1}{2} KL(x) + \lambda TV(x) \right\}, \quad (2.6)$$

nel caso della Kullback-Liebler.

Il vantaggio di utilizzare la Variazione Totale come termine di regolarizzazione è il fatto che essa è sostanzialmente una norma L_1 del gradiente e gioca un ruolo importante quando è richiesta una stima accurata delle discontinuità della soluzione, è infatti capace di ricostruire i contorni in maniera ottimale. Negli esperimenti numerici riportati in questo lavoro di tesi, la TV è stata implementata in diversi modi, a seconda che la regolarizzazione venisse fatta a livello globale o a livello locale, e a seconda che si utilizzasse la norma 1

o la norma 2 del gradiente. La Variazione Totale a livello globale è stata approssimata utilizzando la sua forma isotropica, riportata in [6]. Se

$$G_x = \nabla_x X, \quad (2.7)$$

$$G_y = \nabla_y X, \quad (2.8)$$

la variazione totale risulta espressa come

$$TV(X) = \sum_i \sqrt{(G_x)_i^2 + (G_y)_i^2} \quad (2.9)$$

se si utilizza la norma 2 del gradiente, oppure come

$$TV(X) = \sum_i |G_x|_i + |G_y|_i \quad (2.10)$$

qualora si utilizzi la norma 1. Come si vedrà meglio in seguito, però, la regolarizzazione può essere più efficace se viene calibrata a livello locale. Per questo, è stata utilizzata una Variazione Totale Localmente Pesata (LWTV, *Locally Weighted Total Variation*). Estendendo le formule (2.9) e (2.10), si ha:

$$\sum_i \lambda_i (|G_x|_i + |G_y|_i), \quad (2.11)$$

oppure

$$\sum_i \lambda_i \sqrt{(G_x)_i^2 + (G_y)_i^2}. \quad (2.12)$$

2.2 Scelta del parametro di regolarizzazione

Dopo aver discusso del ruolo del funzionale di regolarizzazione, risulta necessaria una riflessione sul parametro di regolarizzazione, che è stato fino ad ora indicato con λ . La scelta di tale parametro è di fondamentale importanza nel problema di ricostruzione di immagine, poichè esso determina il peso del termine di regolarizzazione sulla soluzione. In linea generale, un buon parametro di regolarizzazione deve garantire che ci sia il giusto equilibrio tra il termine di consistenza dei dati e il termine di regolarizzazione, per evitare sia di avere una soluzione troppo regolarizzata, quindi troppo lontana da quella reale, sia di avere una soluzione troppo instabile. Uno degli obiettivi di questo lavoro di tesi è stato quello di creare problemi test per determinare automaticamente i parametri di regolarizzazione, a livello globale e, soprattutto, locale. In letteratura sono presenti diversi criteri per la scelta automatica del parametro di regolarizzazione in un problema di ottimizzazione.

- Il **Principio della Discrepanza** è un metodo che cerca di determinare il parametro di regolarizzazione in modo tale che il residuo della corrispondente soluzione sia quanto più possibile simile alla norma del rumore. È richiesta, quindi, la conoscenza della norma del *noise*, che non sempre si ha.
- Il criterio della **L-curva** analizza il comportamento della norma della soluzione e del residuo al variare del parametro e non utilizza alcuna conoscenza a priori della norma del rumore. Tale metodo prevede la creazione di un grafico con una tipica forma ad "L", che ha in ascissa il logaritmo della norma del residuo e in ordinata il logaritmo della norma della soluzione calcolata. Il valore d'angolo del grafico è il parametro che viene scelto.
- Il criterio della **GCV** (*Generalized Cross Validation*) deriva dall'idea secondo cui un buon parametro di regolarizzazione dovrebbe essere in grado di predire anche un dato omesso e si basa su considerazioni di tipo statistico.

Nei paragrafi successivi viene introdotto il criterio utilizzato in questo lavoro per la scelta del parametro di regolarizzazione, chiamato Principio di Penalizzazione Uniforme.

2.3 Cenni storici

Alle origini del principio di Penalizzazione Uniforme per la scelta del parametro di regolarizzazione vi è il metodo di Miller, introdotto in [10]. Nell'articolo del 1970 citato, Miller propone un metodo per trattare un problema di minimo con funzione obiettivo data dalla somma di un termine di *fit* dei dati e di una funzione di regolarizzazione, pesata con un parametro determinato in modo automatico. Il criterio utilizzato per la scelta del parametro si basa sulla norma 2 del residuo e porta ad un metodo di regolarizzazione, in quanto la soluzione regolarizzata tende alla soluzione reale, quando il rumore tende a 0. Come prosecuzione del metodo di Miller, va ricordato il caso di applicazione per un problema di Nuclear Magnetic Resonance (NMR) 2-dimensionale, riportato in [9]. La procedura utilizzata in tale studio per la scelta del parametro generalizza il principio di *Uniform Penalty* (UPEN) introdotto in [11]. Il metodo risolve un problema di ottimizzazione per una

funzione data dalla somma di un termine di consistenza dei dati ed un termine di penalizzazione in norma L_2 con parametri multipli. Se f indica la soluzione di tale problema, il termine di regolarizzazione viene preso come

$$\lambda \| f \|^2 \quad (2.13)$$

nel caso di un parametro di regolarizzazione globale, oppure come

$$\sum_{i=1}^N \lambda_i (Lf)_i^2, \quad (2.14)$$

nel caso di regolarizzazione calibrata a livello locale. Il termine $L \in \mathbb{R}^N$ indica l'operatore Laplaciano discreto. I parametri λ e λ_i , già in questa trattazione riguardante NMR, sono calcolati imponendo il principio di Penalizzazione Uniforme, che viene di seguito visto in dettaglio.

2.4 Principio di Penalizzazione Uniforme

Come visto in precedenza, il problema di minimo per il problema SISR con tecnica reconstruction-based considerato ha la forma:

$$\min_x \{F(x) + G(x)\}, \quad (2.15)$$

ovvero

$$\min_x \left\{ \frac{1}{2} \| SHx - g \|^2 + \lambda TV(x) \right\} \quad (2.16)$$

dove le matrici S ed H sono, rispettivamente, l'operatore di *up-sampling* e quello di *blur*, presentati in precedenza, e il termine noto g corrisponde all'immagine LR, eventualmente contenente rumore.

L'idea del principio di Penalizzazione Uniforme è quella di imporre una scelta del parametro di regolarizzazione λ in modo tale da garantire che il termine di consistenza dei dati e il termine di regolarizzazione siano uguali, quando al posto della variabile x si pone l'immagine esatta, ovvero

$$\lambda TV(x_{\text{esatta}}) = \frac{1}{2} \| SHx_{\text{esatta}} - g \|^2 \quad (2.17)$$

Il parametro di regolarizzazione viene scelto, quindi, in relazione al residuo ed ha l'espressione

$$\lambda = \frac{\| SHx_{\text{esatta}} - g \|^2}{2TV(x_{\text{esatta}})} \quad (2.18)$$

Poichè non si conosce l'immagine esatta, al posto di x viene utilizzata una sua approssimazione, ricostruita dall'algoritmo scelto.

$$\lambda = \frac{\|SHx - g\|^2}{2TV(x)} \quad (2.19)$$

L'idea seguita per l'aggiornamento del parametro λ è stata, quindi, quella di creare un ciclo della seguente forma:

$x^0 = g$

for $k = 0, 1, 2, 3, \dots$

$$\lambda^{(k)} = \frac{\|SHx^{(k)} - g\|^2}{2TV(x^{(k)})}$$

$$x^{(k+1)} = \arg \min_x \left\{ \frac{1}{2} \|SHx^{(k)} - g\|_2^2 + \lambda^{(k)} TV(x^{(k)}) \right\} \quad (2.20)$$

end

2.5 Principio di Penalizzazione Uniforme locale

Il rischio di una regolarizzazione a singolo parametro globale è la tendenza a regolarizzare troppo la soluzione, rendendola distante da quella reale, oppure a regolarizzarla poco. Una strategia per ottenere qualche miglioramento consiste nell'utilizzare una regolarizzazione a livello locale, calibrata punto per punto. Come detto precedentemente, uno degli obiettivi di questo studio è stato proprio quello di sperimentare una regolarizzazione effettuata a livello locale per le tecniche di ricostruzione di immagine utilizzate. Si parla di scegliere parametri di regolarizzazione multipli e di operare una regolarizzazione pixel per pixel. La regolarizzazione locale ha il doppio vantaggio di rendere più evidenti i contrasti tra zone diverse dell'immagine e mantenere un buon andamento degli errori di *fitting* dei dati. I valori approssimati per i λ_{ij} , calcolati ad ogni iterazione, determinano una matrice di parametri di regolarizzazione, grazie alla quale risulta possibile un miglioramento dell'accuratezza del metodo. Il principio utilizzato per la scelta dei parametri ad ogni iterazione è stato quello della Penalizzazione Uniforme, nella sua seguente generalizzazione.

Il problema risulta formulato come

$$\min_x \left\{ \frac{1}{2} \| SHx - g \|^2 + \sum_{i=1}^{N_0} \lambda_i \phi_i(x) \right\}, \quad (2.21)$$

dove N_0 è il numero degli elementi di x , λ è la vettorizzazione della matrice dei parametri e

$$\phi_i(x) = \begin{cases} |G_x|_i + |G_y|_i \\ \sqrt{(G_x)_i^2 + (G_y)_i^2} \end{cases} \quad (2.22)$$

a seconda che si utilizzi la norma 1 o la norma 2 del gradiente, dove G_x e G_y sono definite come in (2.7). Nel caso locale, si utilizza quindi la Variazione Totale Localmente Pesata.

Definizione 2.2. (Principio UPEN Generalizzato) Scegliere i parametri di regolarizzazione λ_i in modo che, in una soluzione x , i termini $\lambda_i \phi_i(x)$ siano costanti, per ogni i con $\phi_i(x) \neq 0$, ovvero:

$$\lambda_i \phi_i(x) = c, \quad \forall i = 1, \dots, N_0, \quad \text{t.c.} \quad \phi_i(x) \neq 0, \quad (2.23)$$

dove c è una costante positiva.

Assumendo di avere un limite superiore ϵ per il termine di consistenza dei dati nella soluzione x^* del problema senza rumore,

$$\frac{1}{2} \| SHx^* - g \|^2 \leq \epsilon^2, \quad (2.24)$$

allora, seguendo il criterio di Miller, la costante c viene scelta per bilanciare il termine di consistenza dei dati e quello di regolarizzazione, nel modo seguente:

$$c = \frac{\epsilon^2}{N_0} \quad (2.25)$$

In questo modo si ha che

$$\frac{1}{2} \| SHx - g \|^2 + \sum_{i=1}^{N_0} \lambda_i \phi_i(x) \leq 2\epsilon^2. \quad (2.26)$$

Da (2.23) e (2.24) si ottiene l'espressione dei λ_i :

$$\lambda_i = \frac{\epsilon^2}{N_0 \phi_i(x)} \quad (2.27)$$

ovvero

$$\lambda_i = \frac{\|SHx - g\|^2}{2N_0\phi_i(x)} \quad (2.28)$$

Per il calcolo automatico dei parametri locali è stata usata una procedura iterativa, simile a quella presentata nel caso globale. Sia τ una tolleranza fissata e ρ un piccolo parametro di soglia, che permette di evitare problemi creati dai pixel a gradiente nullo. Il seguente algoritmo, la cui convergenza è dimostrata in [15], schematizza la metodologia seguita.

Algorithm 1

- 1: Scelta di un iterato iniziale $x^{(0)}$;
- 2: Scelta dei parametri $\rho, \tau \in (0, 1)$;
- 3: **Iterazione:**
- 4: Porre $\epsilon^{(k)} = \|SHx^{(k)} - g\|^2$
- 5: Porre

$$\lambda_i^{(k)} = \frac{\epsilon^{(k)}}{2N_0(\phi_i(x^{(k)}) + \rho)}$$

- 6: Calcolare

$$x^{(k+1)} = \arg \min_x \left\{ \|SHx^{(k)} - g\|^2 + \sum_{i=1}^{N_0} \lambda_i^{(k)} |\nabla x|_i \right\}$$

- 7: Porre $k = k + 1$;
 - 8: **Condizione d'arresto** $\frac{\|x^{(k+1)} - x^{(k)}\|}{\max(1, \|x^{(k+1)}\|)} < \tau$
-

Per attenuare la sensibilità al rumore presente nelle soluzioni $x^{(k)}$, il calcolo di ϕ_i è stato modificato come segue:

$$\phi_i = \max_{\mu \in I_i} (|\nabla \tilde{x}|_\mu),$$

dove I_i rappresenta un intorno 3×3 del pixel i -esimo e \tilde{x} è ottenuta applicando un filtro gaussiano alla soluzione $x^{(k)}$. Per maggiori dettagli, si veda anche [11].

Capitolo 3

Algoritmi di Ottimizzazione

Nell'approccio di tipo *reconstruction-based* considerato, il problema di Super-Risoluzione a Singola Immagine si presenta, dunque, come riportato in (2.20). Il problema di minimo in esso contenuto, si può riformulare nel modo seguente:

$$\min_x \{F(x) + G(x, \lambda^{(k)})\} \quad (3.1)$$

dove F è il termine di *fit* dei dati e G è il termine di regolarizzazione che dipende dal parametro λ , aggiornato ad ogni iterazione. In questo capitolo vengono richiamate alcune definizioni relative ai problemi di ottimizzazione non vincolata e vengono presentati i due metodi utilizzati nelle prove numeriche di questa tesi: l'algoritmo SPDHG (*Scaled Primal Dual Hybrid Gradient*) e l'algoritmo SRFC (*Super-Resolution FISTA Chambolle*, FISTA = *Fast Iterative Thresholding Algorithm*). Il lavoro svolto su entrambi è stato quello di formulare una generalizzazione per trattare il caso di regolarizzazione multipla con parametro locale.

3.1 Richiami sui Problemi di Ottimizzazione

Un generico problema di ottimizzazione non vincolata si presenta nella forma

$$\min_x f(x),$$

dove $x \in \mathbb{R}^n$ ed $f : \mathbb{R}^n \rightarrow \mathbb{R}$ funzione regolare. Lo scopo di un problema di questo tipo è la determinazione, se esiste, di

$$x^* = \arg \min_{x \in \mathbb{R}^n} f(x).$$

Definizione 3.1. Si definisce punto di **minimo locale** di f un punto x^* tale che esista un $\epsilon > 0$ per cui

$$f(x^*) \leq f(x) \quad (3.2)$$

per ogni x tale che $\|x - x^*\| < \epsilon$.

Se la disuguaglianza in 3.2 è stretta, x^* si dice minimo locale in senso stretto.

Definizione 3.2. Si definisce punto di **minimo globale** di f un punto x^* tale che

$$f(x^*) \leq f(x) \quad (3.3)$$

per ogni $x \in \mathbb{R}^n$.

Il teorema di Weierstrass assicura che una funzione continua con dominio compatto e non vuoto ammette un punto di minimo globale su di esso. Vi sono poi le cosiddette condizioni di ottimalità, che garantiscono che un punto x^* sia una soluzione ottima (locale o globale) del problema.

Teorema 3.1. (*Condizione necessaria del primo ordine*) Se x^* è un punto di minimo locale di f e f è differenziabile con continuità in un intorno aperto di x^* , allora $\nabla f(x^*) = 0$.

Un punto x^* tale che $\nabla f(x^*) = 0$ è detto **punto stazionario** di f .

Teorema 3.2. (*Condizioni necessarie del secondo ordine*) Se x^* è un punto di minimo locale di f e f è due volte differenziabile con continuità in un intorno aperto di x^* , allora $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*)$ è semidefinita positiva.

Teorema 3.3. (*Condizioni sufficienti del secondo ordine*) Se f è due volte differenziabile con continuità in un intorno aperto di x^* , se $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*)$ è semidefinita positiva, allora x^* è un punto di minimo locale in senso stretto di f .

Il caso interessante per il presente studio è quello delle funzioni obiettivo convesse, per le quali vengono riportati alcuni risultati.

Definizione 3.3. (Funzione convessa) Una funzione f è convessa in \mathbb{R}^n se

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

per ogni $0 \leq \alpha \leq 1$ e per ogni $x, y \in \mathbb{R}^n$.

Il seguente teorema riporta le condizioni di ottimalità nel caso convesso:

Teorema 3.4. Se f è convessa, allora ogni punto di minimo locale x^* è un punto di minimo globale di f .

Se f è strettamente convessa, allora esiste un unico punto di minimo globale.

Se f è convessa e differenziabile, allora ogni punto stazionario x^* è un punto di minimo globale di f .

3.2 Algoritmo Primale-Duale Ibrido

Una funzione convessa è continua in ogni punto del suo dominio e per le funzioni continue è possibile dare la definizione di subgradiente, che risulterà utile per l'algoritmo presentato di seguito.

Definizione 3.4. (Subgradiente) Sia f una funzione convessa con dominio M e sia x_0 un punto interno di M . Il vettore $g(x_0)$ che soddisfa la relazione

$$f(x) - f(x_0) \geq \langle g(x_0), x - x_0 \rangle, \quad \forall x \in M, \quad (3.4)$$

è detto subgradiente di f in x_0 .

Definizione 3.5. (ϵ -subgradiente) Sia f una funzione convessa con dominio M e sia x_0 un punto interno di M e $\epsilon \geq 0$ un numero reale fissato. Un vettore $g \in \mathbb{R}^n$ che soddisfa la relazione

$$f(x) - f(x_0) \geq \langle g, x - x_0 \rangle - \epsilon, \quad \forall x \in M, \quad (3.5)$$

è detto ϵ -subgradiente di f in x_0 .

Date queste definizioni, è possibile introdurre il metodo SPDHG (*Scaled Primal Dual Hybrid Gradient*). Si tratta di un algoritmo in grado di risolvere un problema di minimo con funzione obiettivo convessa, richiede solo operazioni matrice-vettore e semplici proiezioni ed utilizza un'iterazione di tipo ϵ -subgradiente, che genera una successione

$$x^{(\ell+1)} = x^{(\ell)} - \alpha_\ell u^{(\ell)}, \quad (3.6)$$

dove $\alpha_\ell > 0$ scalare positivo è detto lunghezza del passo ed $u^{(\ell)}$ è un ϵ -subgradiente di f in $x^{(\ell)}$. Nell'algoritmo SPDHG vengono generati in modo automatico i parametri $(\alpha_\ell, \epsilon_\ell, u^{(\ell)})$. Per delineare il metodo, viene ripresa la descrizione fatta dagli autori di [5]. Il problema SISR con parametro globale espresso come

$$\min_x \left\{ \frac{1}{2} \| SHx - g \|_2^2 + \lambda^{(k)} TV(x) \right\}, \quad (3.7)$$

con $x \geq 0$, è il caso particolare di un problema di minimo formulato nel modo seguente

$$\min_{x_0 \in \mathbb{R}^n} f_0(x) + f_1(Ax) + \Phi(x), \quad (3.8)$$

dove $f_0(x) = \frac{1}{2} \| SHx - g \|_2^2$ è il termine di *fit*, $f_1(Ax) = \lambda^{(k)} TV(x)$ è il termine di regolarizzazione e $\Phi(x)$ è la funzione indicatrice di X ottante

positivo, che rappresenta i vincoli di positività. In questo contesto, la TV viene espressa come

$$TV(x) = \sum_i \| A_i x \|, \quad (3.9)$$

con

$$A_i x = \begin{pmatrix} x_{p+1,q} - x_{p,q} \\ x_{p,q+1} - x_{p,q} \end{pmatrix},$$

con $p = 1, \dots, M$, $q = 1, \dots, N$.

L'iterazione di base del metodo SPDHG prevede i seguenti passaggi:

$$y^{(\ell+1)} = P_{I,Y}(y^{(\ell)} + \tau_\ell \lambda^{(k)} A x^{(\ell)}) \quad (3.10)$$

$$u^{(\ell)} = d^{(\ell)} + \lambda^{(k)} A^T y^{(\ell+1)} \quad (3.11)$$

$$x^{(\ell+1)} = P_{D_\ell^{-1},X} \left(x^{(\ell)} - \frac{\alpha_\ell}{\min(1, \| u^{(\ell)} \|_{D_\ell})} D_\ell u^{(\ell)} \right) \quad (3.12)$$

dove $d^{(\ell)} = H^T S^T (S H x^{(\ell)} - g)$, $\{D_\ell\}$ è una sequenza di matrici simmetriche definite positive con autovalori limitati e $\{\tau_\ell\}$ e $\{\alpha_\ell\}$ sono le successioni delle lunghezze dei passi rispettivamente dual e primal. I è l'insieme dei punti di x a gradiente non nullo e $P_{I,Y}$ denota la proiezione standard euclidea sull'insieme Y . Il metodo è un caso particolare di un metodo di tipo ϵ -subgradiente *forward-backward scaled*, dove il passo *forward* è dato dalla (3.10) e dalla (3.11) e fornisce un ϵ -subgradiente di $f_0 + f_1 \circ A$ e il passo *backward* (3.12) di $i_X(x)$ è la proiezione dell'ottante positivo con rispetto della metrica indotta da D_ℓ^{-1} . Per la descrizione dettagliata dell'algoritmo, si rimanda all'articolo [5].

Oggetto di questo studio, più che l'implementazione dell'algoritmo, è stato il suo utilizzo all'interno della procedura per la determinazione automatica dei parametri di regolarizzazione per il problema SISR. Nel caso di aggiornamento del parametro di regolarizzazione a livello globale, l'algoritmo SPDHG è stato utilizzato direttamente nella forma sopra presentata. Per la regolarizzazione locale, è risultata, invece, necessaria un'estensione di tale algoritmo. Come già detto, quando si opera una regolarizzazione locale non si ha più un parametro scalare, ma una matrice di parametri di regolarizzazione. Per questo motivo, l'algoritmo SPDHG è stato modificato con l'inserimento della Variazione Totale Localmente Pesata. In questo caso,

$$f_1(Ax) = \sum_i \lambda_i^{(k)} \| A_i x \| = \sum_i \| \tilde{A}_i x \|, \quad (3.13)$$

con $\tilde{A}_i = \lambda_i^{(k)} A_i$.

Poichè, a meno di sostituire A con \tilde{A} nelle (3.10) e (3.11), l'algoritmo risulta

identico sia nel caso globale che in quello locale, viene di seguito riportato solo il codice del primo.

Algorithm 2 Algoritmo SPDHG

Scegliere $\ell = 0$, $x^{(0)} \in X$.

FOR $\ell = 0, 1, 2, \dots$

STEP 0. Calcolare l' ϵ -subgradiente $u^{(\ell)}$ di $f_0(x^{(\ell)}) + f_1(Ax^{(\ell)})$ con i passi (3.10) e (3.11)

STEP 1. Calcolare $f_0(x^{(\ell)}) + f_1(Ax^{(\ell)})$

STEP 2. Calcolare la matrice di *scaling*, come in [16]

STEP 3. Calcolare una stima del valore ottimale della funzione obiettivo nell'iterata ℓ -esima, f_ℓ^{lev}

STEP 4. Aggiornare il passo e calcolare la nuova iterata

$$\alpha_\ell = \frac{f_0(x^{(\ell)}) + f_1(Ax^{(\ell)}) - f_\ell^{\text{lev}}}{\max(1, \|u^{(\ell)}\|_{D_\ell})}$$

$$x^{(\ell+1)} = P_{D_\ell^{-1}, X} \left(x^{(\ell)} - \alpha_\ell D_\ell \frac{u^{(\ell)}}{\max(1, \|u^{(\ell)}\|_{D_\ell})} \right).$$

STEP 7. $\sigma_{\ell+1} = \sigma_\ell + \alpha_\ell$ e tornare allo STEP 1.

END

3.3 Algoritmo FISTA Chambolle

Il secondo metodo utilizzato nelle prove numeriche di questo lavoro è stato l'algoritmo SRFC (*Super-Resolution FISTA Chambolle*). Anche in questo caso, l'algoritmo è stato aggiornato ed utilizzato per risolvere il problema di ottimizzazione all'interno della procedura iterativa per la determinazione dei parametri di regolarizzazione locali. Come per SPDHG, viene innanzitutto mostrata la struttura generale di SRFC per un parametro di regolarizzazione globale, quindi a partire dal problema di minimo SISR formulato come

$$\min_x \left\{ \frac{1}{2} \|SHx - g\|_2^2 + \lambda^{(k)} TV(x) \right\}. \quad (3.14)$$

Sia $J(x) = \lambda^{(k)} TV(x)$ ed $f(x) = \frac{1}{2} \|SHx - g\|_2^2$. La funzione f scritta come discrepanza in norma 2 è differenziabile con continuità ed ha un gradiente L -Lipschitz continuo, $L > 0$. All'interno dell'algoritmo SRFC compaiono l'algoritmo FISTA (*Fast Iterative Shrinkage Thresholding Algorithm*) e l'iterazione di Chambolle. Nel caso del problema (3.14) con parametro di

regolarizzazione globale, la TV è implementata nella sua forma isotropica, vista in (2.9) e (2.10). Il primo passo, costituito dal metodo FISTA, consiste nei seguenti passaggi:

Algorithm 3 Algoritmo FISTA

Scegliere $0 \leq \beta \leq \frac{2}{L}$, $t_1 = 1$, $x_0 \in \mathbb{R}^N$.

STEP 0. $y_1 = x_0$

Iterazione $\ell \geq 1$

$$\begin{aligned} x_\ell &= \text{prox}_{\beta, J}(y_\ell - \beta \nabla f(y_\ell)) \\ t_{\ell+1} &= 1 + \frac{\sqrt{1 + 4t_\ell^2}}{2} \\ y_{\ell+1} &= x_\ell + \left(\frac{t_\ell - 1}{t_{\ell+1}} \right) (x_\ell - x_{\ell-1}) \end{aligned}$$

END

Nell'algoritmo compare un operatore chiamato operatore di prossimità, che viene definito come

$$\text{prox}_{\beta, J}(y_\ell - \beta \nabla f(y_\ell)) = \arg \min_x J(x) + \frac{1}{2\beta} \| (y_\ell - \beta \nabla f(y_\ell)) - x \|^2. \quad (3.15)$$

Per calcolarlo, entra in gioco l'iterazione di Chambolle, descritta in [12] ed in [14]. Si tratta di un metodo utilizzato per risolvere il problema di minimo

$$\min_{x \in X} \frac{\|x - g\|^2}{2\beta} + J(x) \quad (3.16)$$

dove $X = \mathbb{R}^{N \times N}$ spazio euclideo, $g \in X$, $\beta > 0$, $J(x) = TV(x)$. Ponendo $g = y_k - \beta \nabla f(y_k)$, il problema (3.15) è equivalente al problema risolto dall'iterazione di Chambolle, che risulta essere un problema di *denoising*. In [14], viene mostrato che la soluzione di tale problema ha la forma

$$x = g - \pi_{\beta K_J}(g), \quad (3.17)$$

dove $\pi_{\beta K_J}$ denota la proiezione sull'insieme K_J definito come

$$K_J = \{v \in L^2(\Omega, \mathbb{R}) : \langle v, x \rangle_{L^2(\Omega, \mathbb{R})} \leq J(x) \quad \forall x \in L^2(\Omega, \mathbb{R})\}$$

Sia $Y = X \times X$. Introducendo una discretizzazione del gradiente di $x \in X = \mathbb{R}^{N \times N}$ tramite differenze finite in avanti ed indicando con u l'immagine x per maggiore chiarezza, si ha

$$Du(i, j) = (\partial_x u(i, j), \partial_y u(i, j)), \quad \forall 1 \leq i, j \leq N, \quad (3.18)$$

con

$$\partial_x u(i, j) = \begin{cases} u(i+1, j) - u(i, j) & \text{se } i < N, \\ 0 & \text{se } i = N, \end{cases} \quad (3.19)$$

e

$$\partial_y u(i, j) = \begin{cases} u(i, j+1) - u(i, j) & \text{se } j < N, \\ 0 & \text{se } j = N. \end{cases} \quad (3.20)$$

Ne segue che la discretizzazione della Variazione Totale $J(u)$ è

$$J(u) = \sum_{1 \leq i, j \leq N} |Du(i, j)|_Y \quad (3.21)$$

con prodotto scalare euclideo standard su Y dato da

$$\langle p, q \rangle_Y = \sum_{1 \leq i, j \leq N} p_1(i, j)q_1(i, j) + p_2(i, j)q_2(i, j),$$

per ogni $p, q \in Y$, $p = (p_1, p_2)$ e $q = (q_1, q_2)$. Si può quindi riscrivere la (3.21) come

$$J(u) = \sup\{\langle p, Du \rangle_Y : p \in Y, |p(i, j)| \leq 1 \quad \forall i, j\} \quad (3.22)$$

per ogni $x \in X$. Per proseguire, si utilizza l'operatore discreto di divergenza, $\text{div}: Y \rightarrow X$, tale che

$$\langle -\text{div}p, u \rangle_X = \langle p, Du \rangle_Y, \quad \forall p \in Y, \forall u \in X,$$

dove $\langle u, v \rangle_X = \sum_{i, j} u(i, j)v(i, j)$. si può facilmente verificare che, $\forall p = (p_1, p_2) \in Y$, div è dato da

$$(\text{div}p)(i, j) = \begin{cases} p_1(i, j) & \text{se } i = 1, \\ -p_1(i-1, j) & \text{se } i = N, \\ p_1(i, j) - p_1(i-1, j) & \text{altrimenti,} \end{cases} + \begin{cases} p_2(i, j) & \text{se } j = 1, \\ -p_2(i, j-1) & \text{se } j = N, \\ p_2(i, j) - p_2(i, j-1) & \text{altrimenti.} \end{cases}$$

Dalla (3.22) e dalla definizione dell'operatore divergenza, si deduce che

$$J(u) = \sup_{v \in K_J} \langle u, v \rangle_X,$$

dove K_J è dato da

$$\{\text{div}p : p \in Y, |p(i, j)| \leq 1, \quad \forall i, j\}.$$

Il calcolo della proiezione $\pi_{\beta K_J}(g)$ si traduce nella risoluzione del seguente problema di minimo:

$$\min \left\{ \|\beta \text{div}p - g\|_X^2 : p \in Y, |p(i, j)|^2 - 1 \leq 0, \forall i, j \right\}.$$

Il metodo di Chambolle, descritto nell'algoritmo seguente, consiste in un metodo di discesa a gradiente semi-implicito e risulta molto efficiente e preciso. Nel caso scalare, che è stato descritto fino ad ora, il parametro di regolarizzazione risulta uguale a $\lambda^{(k)}\beta$, dove $\lambda^{(k)}$ è trovato con il Principio UPEN e β è il parametro presente nel passo FISTA. Dall'articolo [14], risulta anche che una scelta di $\delta_t = \frac{1}{8}$ è sufficiente per la convergenza del metodo.

Algorithm 4 Algoritmo di Chambolle scalare

Input: $\delta_t > 0$ parametro, $tol > 0$.

while $\max_{1 \leq i, j \leq N} \{|p^{n+1}(i, j) - p^n(i, j)|\} > tol$, **do**

for all pixel (i, j)

$$p(i, j) \leftarrow \frac{p(i, j) + \delta_t D(\operatorname{div} p - \lambda^{(k)} \beta g)(i, j)}{1 + \delta_t |D(\operatorname{div} p - \lambda^{(k)} \beta g)(i, j)|}$$

end for

end while

return $u = g - \lambda^{(k)} \beta \operatorname{div} p$.

Per utilizzare l'algoritmo appena descritto all'interno della procedura di determinazione automatica dei parametri di regolarizzazione locali, è stato necessario formulare una sua estensione, inserendo la Variazione Totale Localmente Pesata. La discretizzazione riportata in (3.21) è stata sostituita con la seguente:

$$J(u) = \sum_{i,j} \lambda_{i,j} |Du(i, j)|. \quad (3.23)$$

Conseguentemente, si è ottenuto un operatore di divergenza pesata, $\tilde{\operatorname{div}}$, moltiplicando ogni termine della (3.3) per $\lambda_{i,j}$. Nell'algoritmo 5 si ripercorrono i passi della generalizzazione dell'algoritmo di Chambolle per la regolarizzazione locale.

Algorithm 5 Algoritmo di Chambolle locale

Input: $\delta_t > 0$ parametro, $tol > 0$, β , λ .

While $\max_{1 \leq i, j \leq N} \{|p^{n+1}(i, j) - p^n(i, j)|\} > tol$ **do**

for all pixel (i, j)

$$p(i, j) \leftarrow \frac{p(i, j) + \delta_t D(\tilde{\text{div}}p - \beta g)(i, j)}{1 + \delta_t |D(\tilde{\text{div}}p - \beta g)(i, j)|}$$

end for

end while

return $u = g - \beta \tilde{\text{div}}p$.

Capitolo 4

Algoritmo del Gradiente Prossimale di Bregman

Nella sezione 1.5, è stata introdotta la funzione di divergenza di Kullback-Liebler, presentata come alternativa possibile alla discrepanza in norma 2 utilizzata fino ad ora nel termine di consistenza dei dati del problema SISR. Questo capitolo contiene la descrizione di un algoritmo in grado di generalizzare i metodi del tipo Gradiente Coniugato nel caso di una funzione obiettivo che, come la KL, non abbia gradiente Lipschitz continuo. L'algoritmo presentato permette di risolvere problemi di minimizzazione anche non convessi e prende il nome di *Bregman Proximal Gradient Algorithm with Extrapolation*, BPG_e. Il riferimento bibliografico della seguente trattazione è l'articolo [3], in cui gli autori descrivono l'algoritmo, ne provano la convergenza e ne mostrano alcune applicazioni.

L'assunzione della continuità secondo Lipschitz del gradiente della funzione obiettivo è piuttosto restrittiva. In molte applicazioni reali, infatti, le funzioni obiettivo da minimizzare non soddisfano tale condizione. Proprio per questo, nonostante il metodo del gradiente di Cauchy sia ancora alla base di molti studi, sono stati di recente introdotti diversi nuovi metodi, più generali. Un altro punto su cui la ricerca si concentra è quello di accelerare gli algoritmi già presenti in letteratura. Come in precedenza, si consideri il problema di minimo

$$\min\{\Phi(x) := F(x) + G(x); x \in \mathbb{R}^d\} \quad (4.1)$$

dove F è una funzione non convessa, differenziabile in modo continuo, e G è una funzione di dominio non nullo, semi-continua dal basso e convessa. L'algoritmo classico del Gradiente Prossimale prevede di risolvere il problema

riformulandolo nel seguente modo:

$$x^{k+1} = \arg \min_x \left\{ G(x) + \langle \nabla F(x^k), x - x^k \rangle + \frac{1}{2\lambda_k} \|x - x^k\|^2 \right\} \quad (4.2)$$

dove $k \in \mathbb{N}$ e λ_k è la dimensione del passo ad ogni iterazione. Un primo passo in avanti rispetto a questo algoritmo è l'introduzione di uno step di interpolazione, che permette una maggiore velocità di convergenza.

$$\begin{cases} y^k = x^k + \beta_k(x^k - x^{k-1}) \\ x^{k+1} = \arg \min_x \left\{ G(x) + \langle \nabla F(y^k), x - y^k \rangle + \frac{1}{2\lambda_k} \|x - y^k\|^2 \right\} \end{cases} \quad (4.3)$$

dove il termine $\beta_k(x^k - x^{k-1})$ è detto termine di estrapolazione. L'iterazione riportata in (4.3) viene detta Algoritmo del Gradiente Prossimale con Estrapolazione. Questo metodo, che funziona piuttosto bene, una volta calibrati i parametri β_k , presenta tuttavia un limite. Esso richiede, infatti, che la funzione F abbia gradiente globalmente Lipschitz continuo. Il nuovo algoritmo introdotto per sopperire a tale problema è il BPGe, che non richiede più la condizione sulla continuità del gradiente di F ed ha la seguente forma:

$$\begin{cases} y^k = x^k + \beta_k(x^k - x^{k-1}) \\ x^{k+1} = \arg \min_x \left\{ G(x) + \langle \nabla F(y^k), x - y^k \rangle + \frac{1}{\lambda_k} D_h(x, y^k) \right\} \end{cases} \quad (4.4)$$

dove D_h è una distanza di Bregman, secondo la definizione data nel paragrafo seguente. La condizione richiesta per la convergenza del metodo deriva dalla teoria di Bregman ed ha la seguente forma: si richiede che l'algoritmo soddisfi

$$D_h(x^k, y^k) \leq \rho C_k D_h(x^{k-1}, x^k) \quad (4.5)$$

dove $C_k \in (0, 1]$ e $\rho \in (0, 1)$ sono due parametri.

4.1 Prerequisiti

Vengono introdotte di seguito alcune definizioni preliminari allo studio dell'algoritmo, come quella di distanza di Bregman e di funzioni L-smad.

Definizione 4.1. Sia S un sottoinsieme aperto, convesso e non vuoto di \mathbb{R}^d . Associata ad S , una funzione $h : \mathbb{R}^d \rightarrow (-\infty, \infty]$ si dice distanza *kernel generating* se soddisfa le seguenti condizioni:

- h ha dominio non nullo, è semi-continua dal basso e convessa, con $\text{dom } h \subset \bar{S}$ e $\text{dom } \partial h = S$;
- h è C^1 su $\text{int dom } h = S$.

La funzione h viene anche chiamata funzione di Bregman. La classe delle distanze che generano kernel viene indicata con $\mathcal{G}(S)$. Data $h \in \mathcal{G}(S)$, si definisce distanza di Bregman $D_h : \text{dom } h \times \text{int dom } h \rightarrow [0, +\infty)$

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle. \quad (4.6)$$

Definizione 4.2. Una coppia di funzioni (f, h) tali che $h \in \mathcal{G}(S)$, $f : \mathbb{R}^d \rightarrow (-\infty, \infty]$ ha dominio non nullo, è semi-continua dal basso e $\text{dom } h \subset \text{dom } f$, differenziabile in modo continuo su S , è detta L-smad (*Smooth Adaptable*) su S se esiste $L > 0$ tale che $Lh - f$ ed $Lh + f$ siano convesse su S .

Viene ora presentata la definizione di funzione μ -relativamente convessa in senso debole, che estende la nozione di convessità debole.

Definizione 4.3. Una funzione f viene detta μ -relativamente convessa in senso debole rispetto ad h su S se esiste $\mu > 0$ tale che $f + \mu h$ è convessa su S .

Quando f è convessa, $\mu = 0$. Quando (f, h) è L-smad su S , ovviamente f è L -relativamente convessa in senso debole rispetto ad h . Di conseguenza, $\mu \leq L$.

Definizione 4.4. (*Proprietà di Kurdyka-Łojasiewicz*)

Sia $f : \mathbb{R}^d \rightarrow (-\infty, +\infty]$ una funzione di dominio non nullo, semi-continua dal basso.

1. La funzione f si dice dotata della proprietà di Kurdyka-Łojasiewicz in $\bar{x} \in \text{dom } \partial f := \{x \in \mathbb{R}^d : \partial f(x) \neq \emptyset\}$ se esistono $\eta \in (0, +\infty]$, un intorno U di \bar{x} ed una funzione $\psi : (0, \eta) \rightarrow \mathbb{R}$ che soddisfano: $\psi(0) = 0$, $\psi \in C^1(0, \eta)$, è continua in 0, e per ogni $s \in (0, \eta) : \psi'(s) > 0$, tali che per ogni $x \in U \cap [f(\bar{x}) < f(x) < f(\bar{x} + \eta)]$, valga la seguente disuguaglianza

$$\psi'(f(x) - f(\bar{x})) \cdot \text{dist}(0, \partial f(x)) \geq 1.$$

2. Se f soddisfa la proprietà KL in ogni punto di $\text{dom } \partial f$, allora f viene chiamata funzione di Kurdyka-Łojasiewicz.

Con tale proprietà, gli autori di [3] dimostrano la convergenza globale delle successioni generate dall' algoritmo BPG.

4.2 Caratteristiche generali e Applicazioni

Utilizzando le definizioni del paragrafo precedente, si torna ora a prendere in esame il problema non convesso (4.1), assumendo che la funzione h appartenga a $\mathcal{G}(\mathbb{R}^d)$, (F, h) sia L-smad ed F sia μ -relativamente convessa in senso debole rispetto ad h . Lo pseudo-codice riportato sotto descrive la struttura dell'algoritmo BPGe.

Algorithm 6 Algoritmo BPGe

Dati: h, F, μ, TOL .

Inizializzazione: $x^0 = x^{-1} \in \text{int dom} h$ e $0 < \lambda_k \leq 1/L$.

Iterazione: FOR $k = 0, 1, 2, \dots, k_{max}$

$$y^k = x^k + \beta_k(x^k - x^{k-1}),$$

dove β_k viene trovato con **Line Search**, descritta nell'algoritmo seguente. Poi calcolare

$$x^{k+1} \in \arg \min_x \left\{ G(x) + \langle \nabla F(y^k), x - y^k \rangle + \frac{1}{\lambda_k} D_h(x, y^k) \right\}$$

fino a che EXIT(TOL) non viene soddisfatta.

Per trovare un parametro β_k appropriato ad ogni iterazione, viene usata una ricerca in linea.

Algorithm 7 Ricerca in linea per β_k

Dati: $0 < \eta < 1, \beta_0 \in [0, 1), 0 < \rho < 1$.

Input: $x^{k-1}, x^k \in \text{int dom } h, C_k = \frac{\lambda_k^{-1}}{\lambda_k^{-1} + \mu}$.

Iterazione:

$$\bar{\beta} = \beta_0$$

WHILE $D_h(x^k, x^k + \bar{\beta}(x^k - x^{k-1})) > \rho C_k D_h(x^{k-1}, x^k)$,

$$\bar{\beta} = \eta \bar{\beta}$$

Return: $\beta_k = \bar{\beta}$.

Come condizione d'arresto per il ciclo, si può fornire un massimo numero di iterazioni k_{max} ed imporre un controllo sulle iterate successive:

$$\frac{\|x^k - x^{k-1}\|}{\max(1, \|x^k\|)} \leq TOL.$$

L'algoritmo BPG_e permette la risoluzione di un problema di ottimizzazione non convessa, con funzione obiettivo che non abbia gradiente Lipschitz continuo. Un caso di applicazione è proprio quello della ricostruzione SR a Singola Immagine, laddove al posto della discrepanza in norma 2 venga usata la funzione di Kullback-Liebler come termine di *fit* dei dati. La funzione KL non ha gradiente globalmente Lipschitz continuo, ma soddisfa la condizione di L-smad insieme ad una distanza *kernel generating* chiamata Entropia di Burg e definita come

$$h(x) = - \sum_{j=1}^d \log x_j. \quad (4.7)$$

Ne segue che la distanza di Bregman corrispondente è

$$D_h(x, y) = \sum_{j=1}^d \left\{ \frac{x_j}{y_j} - \log \left(\frac{x_j}{y_j} \right) - 1 \right\}. \quad (4.8)$$

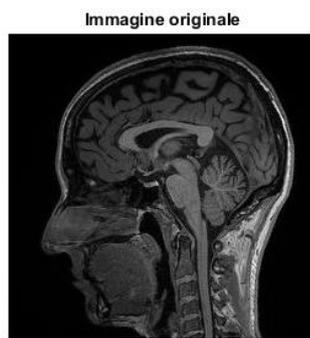
Il problema SISR con funzione KL può essere risolto con l'algoritmo BPG_e, utilizzando anche in questo caso un'opportuna tecnica di regolarizzazione e scelta dei parametri.

Durante la stesura di questo lavoro di tesi, sono stati fatti alcuni test numerici relativi al caso di ricostruzione SR con funzione KL, che hanno previsto l'implementazione di una funzione per la distanza di Bregman, di una funzione per il calcolo della funzione obiettivo e del suo gradiente e l'utilizzo di alcuni algoritmi noti come quello di Gauss-Newton. Tuttavia, lo studio sulla regolarizzazione globale e locale della soluzione per questo problema richiede ulteriore sviluppo, non trattato da questa tesi, che si è concentrata maggiormente sull'analisi del caso di termine di *fit* in norma 2.

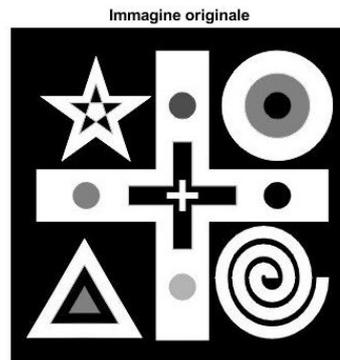
Capitolo 5

Risultati Numerici

In questo capitolo vengono riportati tutti i risultati numerici trovati nei diversi problemi test studiati. Le immagini su cui i metodi sono stati confrontati sono due:



(a) Immagine Brain 249×249



(b) Immagine Geometric 320×320

La prima è un'immagine medica con molti livelli di grigio, mentre la seconda è un'immagine artificiale, con diverse zone piatte. Per entrambe è stato creato un problema test per la ricostruzione SR a Singola Immagine con regolarizzazione globale ed uno per quella con regolarizzazione locale, applicando sia l'algoritmo SPDHG che l'algoritmo SRFC. I calcoli sono stati eseguiti sul GM Cluster del Dipartimento di Matematica dell'Università di Bologna, composto da 4 Intel(R) Xeon(R) CPU @ 2,3 GHz con 512 Gb di Ram, utilizzando il software Matlab R2019a.

5.1 Costruzione del problema test

In questa sezione si considera il caso di regolarizzazione globale. Per risolvere il problema SISR, sono stati definiti gli operatori necessari alla discretizzazione del problema di minimo

$$x^{k+1} = \min_x \left\{ \frac{1}{2} \| SHx - g \|^2 + \lambda TV(x) \right\} \quad (5.1)$$

L'immagine x è quella ad alta risoluzione, che viene presa in input come matrice 2D, normalizzata e resa divisibile per il fattore di sottocampionamento, tramite la funzione *modcrop* di Matlab. S è la matrice di *upsampling* ed H quella di *blur*. Per questi due operatori sono state implementate due funzioni, A e AT , rispettivamente corrispondenti ad SH e ad $H^T S^T$. All'interno di questi operatori viene eseguito il sottocampionamento dell'immagine HR, con un fattore di *upsampling* richiesto in input e posto nei test uguale a 3 o a 4. Il sottocampionamento corrisponde all'eliminazione di alcune righe e colonne, poste ad una distanza data proprio da tale fattore. Inoltre, all'immagine viene applicato un *blur*, costituito da un filtro gaussiano 7×7 , con varianza 1.6. Il vettore g risultante è la vettorizzazione dell'immagine LR, ottenuta sottocampionando e sfocando quella esatta x presa in input. Alle immagini è stato aggiunto artificialmente anche un rumore gaussiano, con *noiselevel*= 0.01. Lo pseudo-codice seguente è relativo all'aggiunta di rumore gaussiano.

```

η =randn(size(imgLR));
η = η / ||η||fro;
w =noiselevel*|| imgLR||fro . * η;
g =imgLR+w;

```

L'immagine g è il punto di partenza per la ricostruzione SR. In tutti i casi considerati, il metodo di interpolazione bicubica, eseguito con la funzione *imresize* di Matlab, è stato preso come confronto per gli altri algoritmi utilizzati.

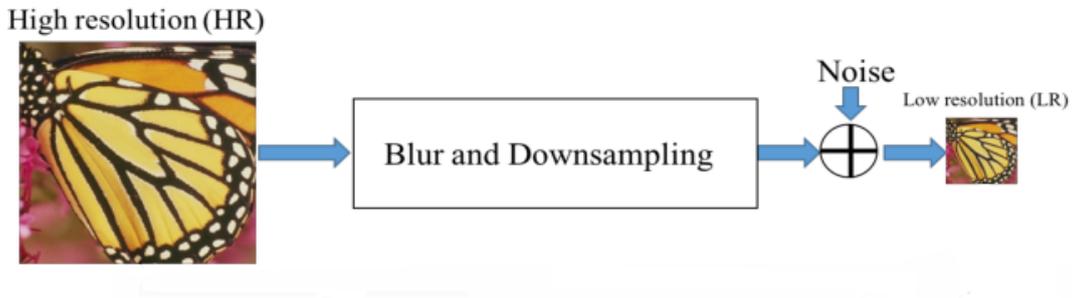


Figura 5.2: Costruzione del problema test

5.2 Parametri di valutazione

Per valutare l'accuratezza dei metodi, ovvero la distanza dell'immagine ricostruita da quella originale, si è deciso di utilizzare diversi errori e parametri di valutazione, descritti di seguito. Sia x^* una soluzione trovata.

Errore relativo. È il primo errore che è stato calcolato ed esprime la distanza dell'immagine ricostruita da quella originale:

$$\text{err} = \frac{\|x - x^*\|}{\|x\|}$$

RMSE. Si tratta della radice quadrata dell'errore quadratico medio, ed è un parametro che decresce all'aumentare della qualità della ricostruzione. La sua espressione è la seguente:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N (x_{i,j} - x_{i,j}^*)^2}{NM}}$$

PSNR. La PSNR, ovvero *Peak Signal to Noise Ratio*, è un parametro che cresce quanto più l'immagine ricostruita è vicina a quella di partenza ed ha la seguente formulazione:

$$\text{PSNR} = 20 \log \left(\frac{1}{\text{RMSE}} \right)$$

SSIM. Si tratta dello *Structural Similarity Index*, un valore che sta tra 0 e 1 e valuta la buona ricostruzione dell'immagine anche localmente. Per il calcolo di questo parametro è stata utilizzata la funzione Matlab *ssim*.

Tempo d'esecuzione. Per ogni problema test è stato salvato il tempo d'esecuzione, misurato in secondi ed ottenuto sommando il tempo di ogni iterazione.

Confronto visivo. Anche il semplice confronto visivo è importante nei problemi di Super-Risoluzione, per verificare la qualità dei diversi metodi di regolarizzazione e di ricostruzione utilizzati.

5.3 Regolarizzazione globale

Dopo aver costruito l'immagine LR sfocata, sottocampionata e con rumore, ha inizio l'iterazione per la ricostruzione SR, inserita all'interno di quella relativa alla determinazione automatica del parametro scalare di regolarizzazione. In questo primo caso, il principio di Penalizzazione Uniforme prevede la scelta di un parametro λ iniziale pari a

$$\lambda = \frac{\| SHx - g \|^2}{2TV(x)} \quad (5.2)$$

e poi un ciclo in cui viene utilizzato l'algoritmo SPDHG oppure l'algoritmo SRFC per la ricostruzione dell'immagine con il parametro λ aggiornato. I criteri d'arresto utilizzati in questa fase sono stati

$$\frac{\| x - x_{\text{old}} \|}{\max(1, \| x \|)} < \text{TOL} \quad (5.3)$$

che valuta la distanza tra due iterate successive,

$$|\lambda - \lambda_{\text{old}}| < \text{TOL} \quad (5.4)$$

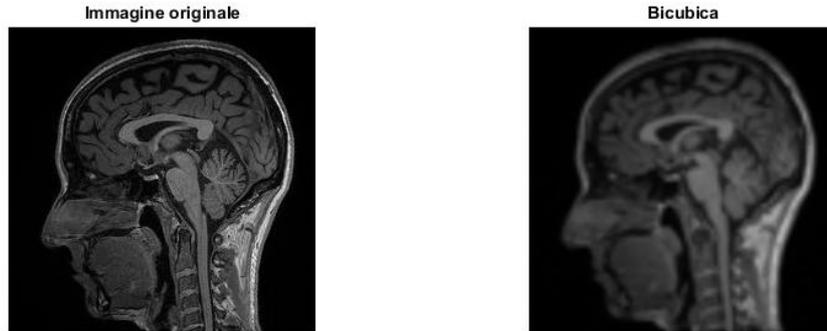
che controlla, invece, la distanza tra due parametri di regolarizzazione successivi. Per entrambi è stata usata $\text{TOL}=10^{-6}$. È stato posto anche un limite massimo sul numero di iterazioni pari a 100.

Su entrambe le immagini è stato provato sia un upfactor pari a 3 che un upfactor pari a 4. Come punto di partenza per gli algoritmi di ottimizzazione è stata scelta l'immagine ottenuta per interpolazione bicubica da quella LR. I risultati ed i grafici riportati di seguito riguardano il caso di regolarizzazione con parametro globale.

5.3.1 Immagine Brain

Vengono riportati i risultati trovati per la prima immagine nel caso di regolarizzazione globale con parametro scalare. I grafici presentano l'anda-

mento di PSNR, RMSE e λ . Nelle tabelle, il valore della PSNR iniziale corrisponde al valore della PSNR per l'immagine ricostruita con interpolazione bicubica.



(a) Immagine originale

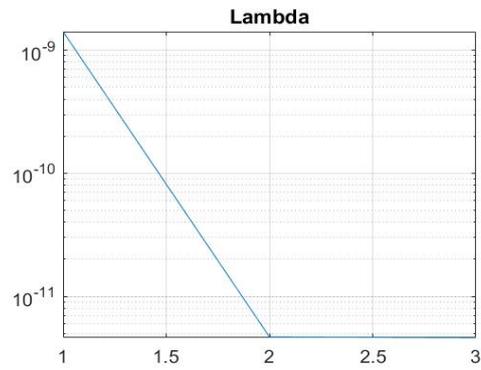
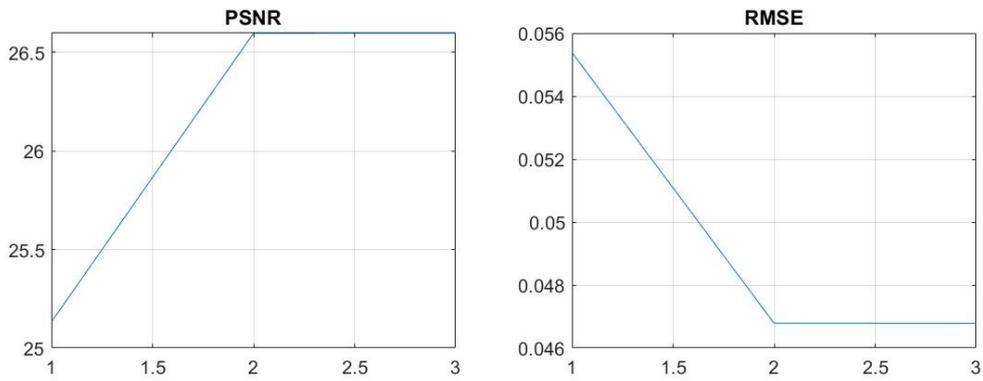
(b) Ricostruzione con interpolazione bicubica, upfactor 3

Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	2	25	27	0.7976	19.8002
SRFC	4	25	27	0.7987	131.0679

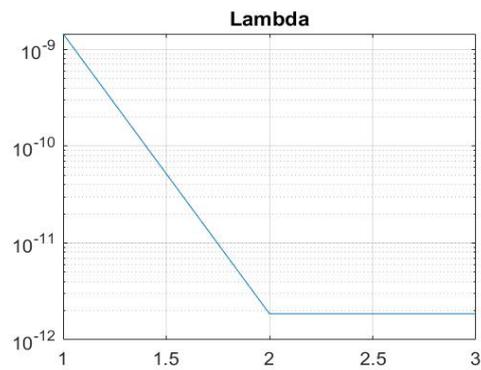
Tabella 5.1: Immagine Brain, regolarizzazione globale, upfactor 3

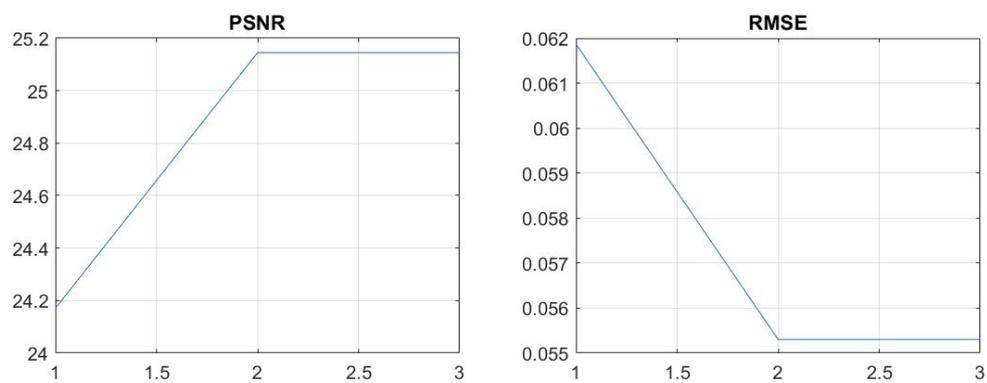
Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	2	24	25	0.7238	19.7067
SRFC	4	24	25	0.7364	118.1158

Tabella 5.2: Immagine Brain, regolarizzazione globale, upfactor 4

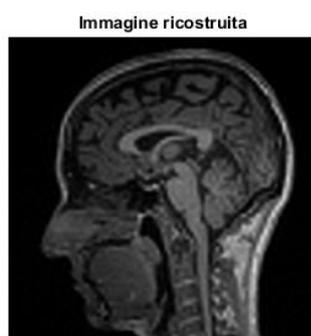
Figura 5.4: λ immagine Brain, SPDHG, upfactor 3

(a) PSNR immagine Brain, SPDHG, upfactor 3 (b) RMSE immagine Brain, SPDHG, upfactor 3

Figura 5.6: λ immagine Brain, SPDHG, upfactor 4



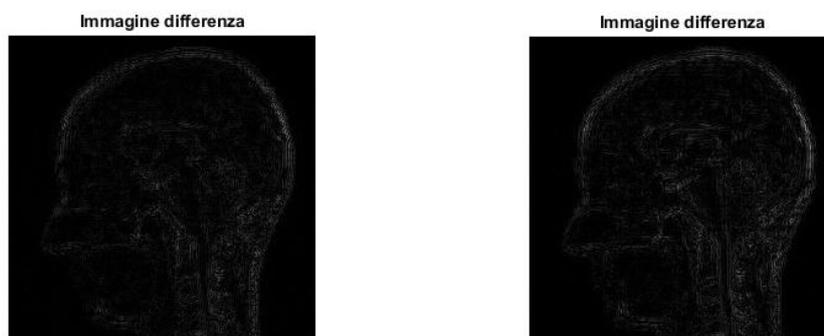
(a) PSNR immagine Brain, SPDHG, upfactor 4, (b) RMSE immagine Brain, SPDHG, upfactor 4



(a) SPDHG, upfactor 3

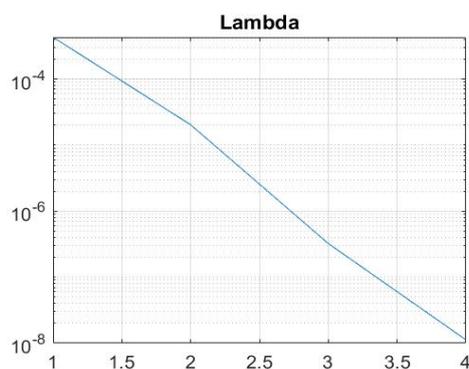
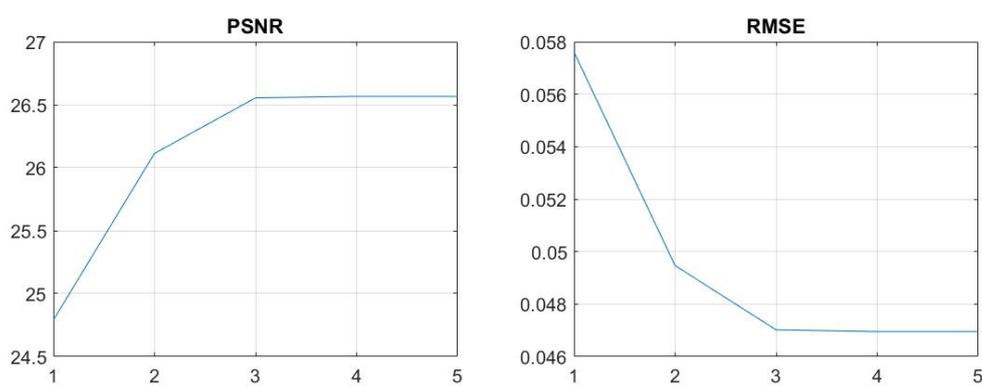


(b) SPDHG, upfactor 4



(a) SPDHG, upfactor 3

(b) SPDHG, upfactor 4

Figura 5.10: λ immagine Brain, SRFC, upfactor 3

(a) PSNR immagine Brain, SRFC, upfactor 3 (b) RMSE immagine Brain, SRFC, upfactor 3

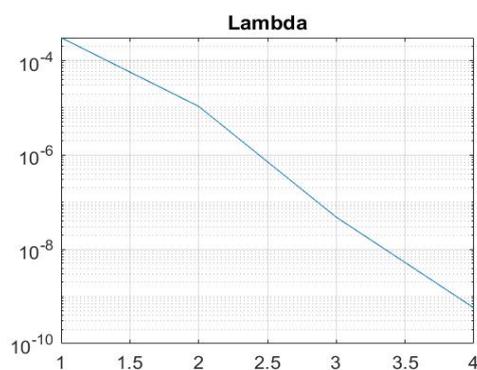
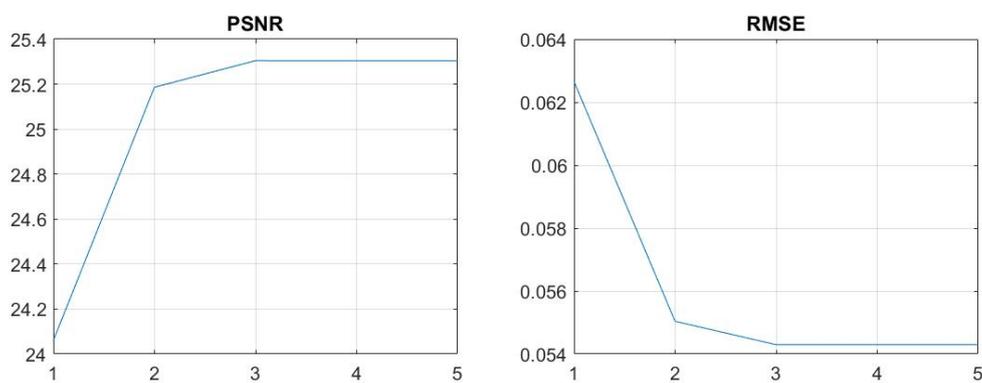
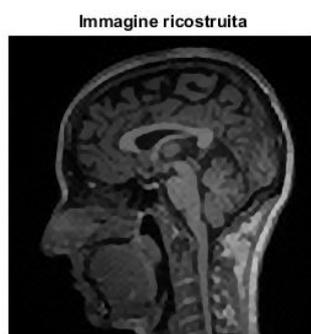


Figura 5.12: λ immagine Brain, SRFC, upfactor 4



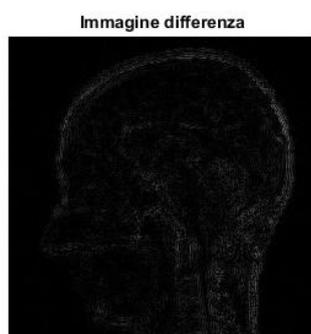
(a) PSNR immagine Brain, SRFC, upfactor 4 (b) RMSE immagine Brain, SRFC, upfactor 4



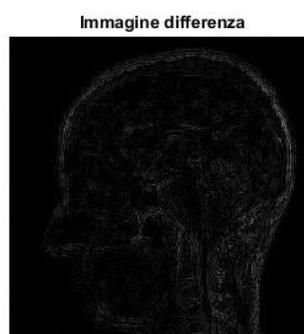
(a) SRFC, upfactor 3



(b) SRFC, upfactor 4



(a) SRFC, upfactor 3



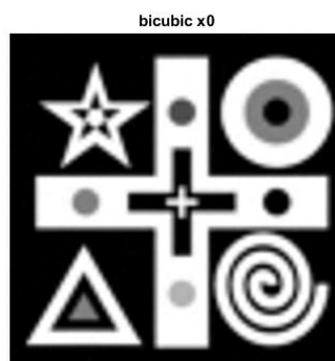
(b) SRFC, upfactor 4

5.3.2 Immagine Geometric

Vengono ora presentati, in modo analogo, i risultati per l'immagine geometrica. Anche in questo caso, viene presa come punto di partenza l'immagine ricostruita per interpolazione bicubica.



(a) Immagine originale



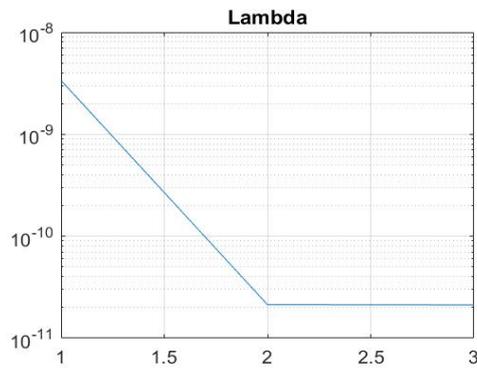
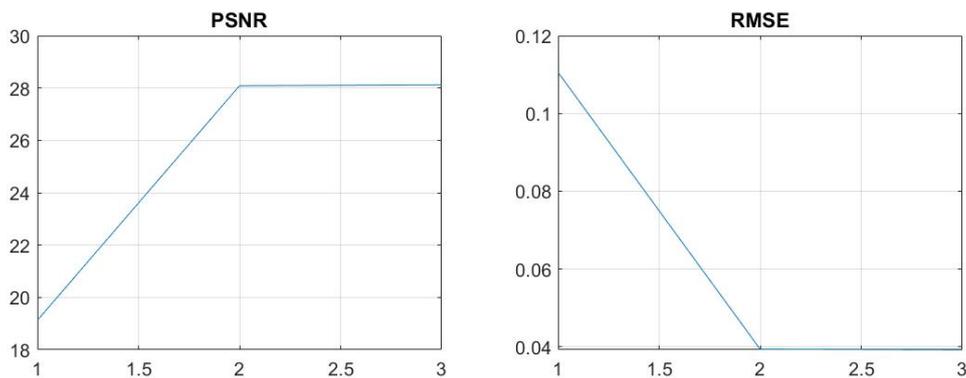
(b) Ricostruzione con interpolazione bicubica, upfactor 3

Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	2	19	28	0.8785	28.0724
SRFC	3	19	30	0.8800	92.2881

Tabella 5.3: Immagine Geometric, regolarizzazione globale, upfactor 3

Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	2	18	24	0.8720	28.5829
SRFC	3	18	28	0.8822	100.9666

Tabella 5.4: Immagine Geometric, regolarizzazione globale, upfactor 4

Figura 5.17: λ immagine Geometric, SPDHG, upfactor 3

(a) PSNR immagine Geometric, SPDHG, upfactor 3 (b) RMSE immagine Geometric, SPDHG, upfactor 3

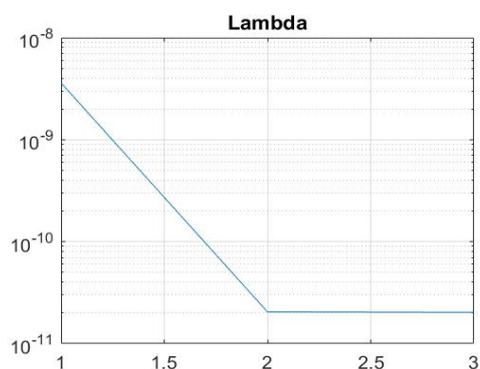
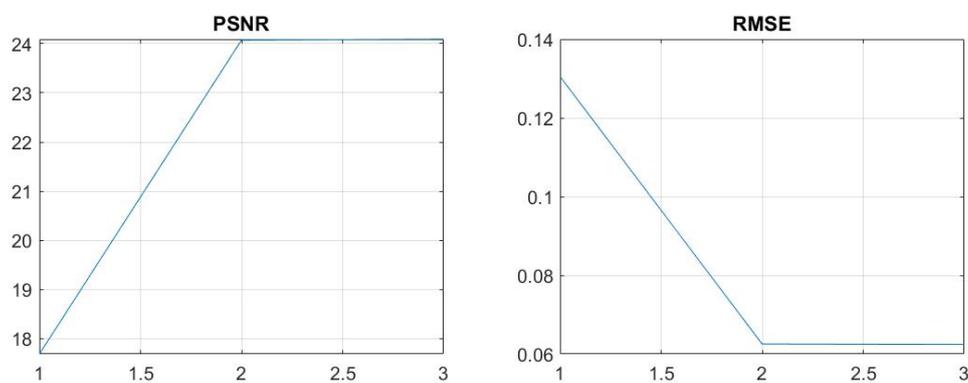


Figura 5.19: λ immagine Geometric, SPDHG, upfactor 4



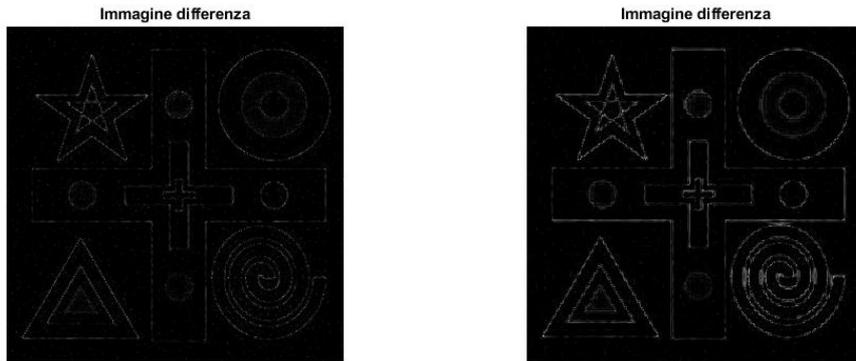
(a) PSNR immagine Geometric, SPDHG, (b) RMSE immagine Geometric, SPDHG, upfactor 4



(a) SPDHG, upfactor 3



(b) SPDHG, upfactor 4



(a) SPDHG, upfactor 3

(b) SPDHG, upfactor 4

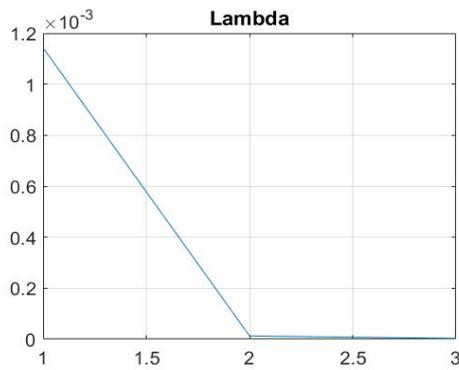
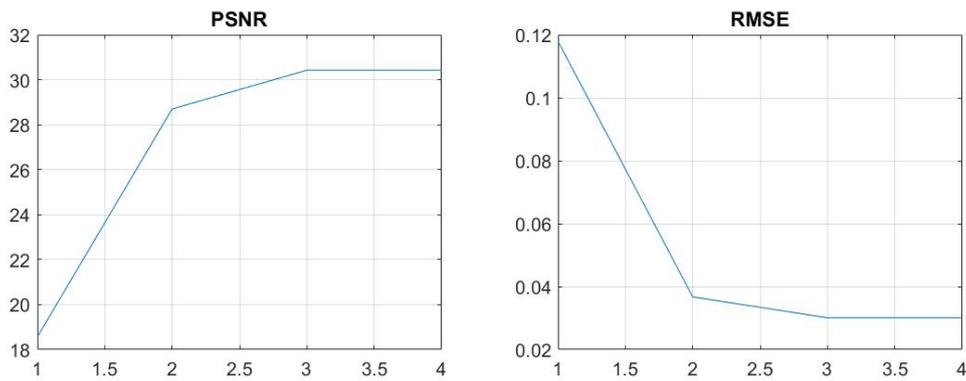


Figura 5.23: λ immagine Geometric, SRFC, upfactor 3



(a) PSNR immagine Geometric, SRFC, (b) RMSE immagine Geometric, SRFC, upfactor 3

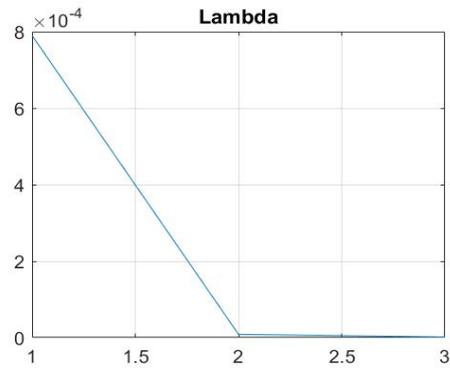
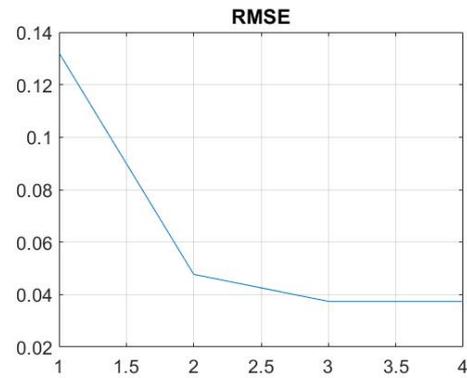
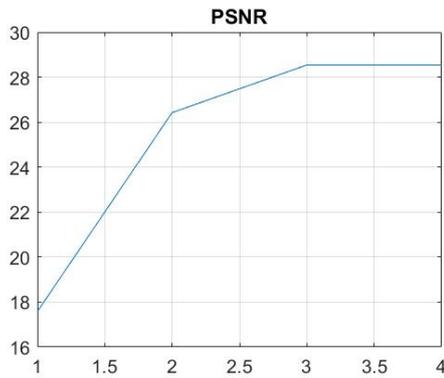


Figura 5.25: λ immagine Geometric, SRFC, upfactor 4



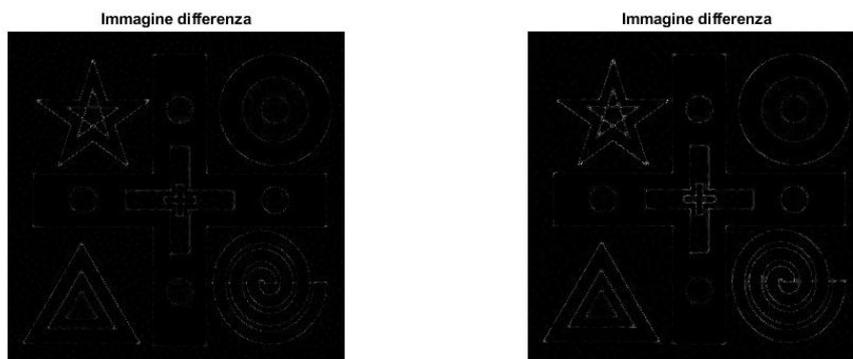
(a) PSNR immagine Geometric, SRFC, upfactor 4 (b) RMSE immagine Geometric, SRFC, upfactor 4



(a) SRFC, upfactor 3



(b) SRFC, upfactor 4



(a) SRFC, upfactor 3

(b) SRFC, upfactor 4

5.4 Regolarizzazione locale

Si passa ora al caso di regolarizzazione con parametro locale, scelto mediante il principio di Penalizzazione Uniforme, visto in (2.5). Il problema test ha una struttura del tutto simile a quella del caso precedente, con l'eccezione delle modifiche relative al parametro di regolarizzazione, che ora è una matrice. Nel caso dell'algoritmo SPDHG, inoltre, sono stati aggiunti alcuni "pesi" nell'espressione del parametro locale λ . In particolare, il gradiente di x , presente nella LWTV, è stato calcolato dopo aver applicato un filtro gaussiano ad x . In seguito, i valori del gradiente sono stati ottenuti prendendo il valore massimo in un intorno di 3×3 pixel. È stata utilizzata la funzione Matlab *ordfilt2* per tale scopo. Il residuo è stato pesato mediante la moltiplicazione per un valore soglia pari a 10^{-6} ; lo stesso valore soglia è stato sommato al denominatore di λ per evitare problematiche relative a pixel con gradiente nullo. Il codice per la scelta del parametro è quindi il seguente:

Algorithm 8 Modifiche per parametro locale

```
p=ordfilt2(grad,9,ones(3,3));
RSQRD=(norm(A(x)-gn))^2;
soglia=soglia*RSQRD;
lambda=RSQRD./(numel(x)*2*(p+soglia));
```

Il criterio d'arresto imposto in questo caso è stato quello sulle x , con $TOL = 10^{-6}$.

$$\frac{\|x - x_{old}\|}{\max(1, \|x\|)} < TOL \quad (5.5)$$

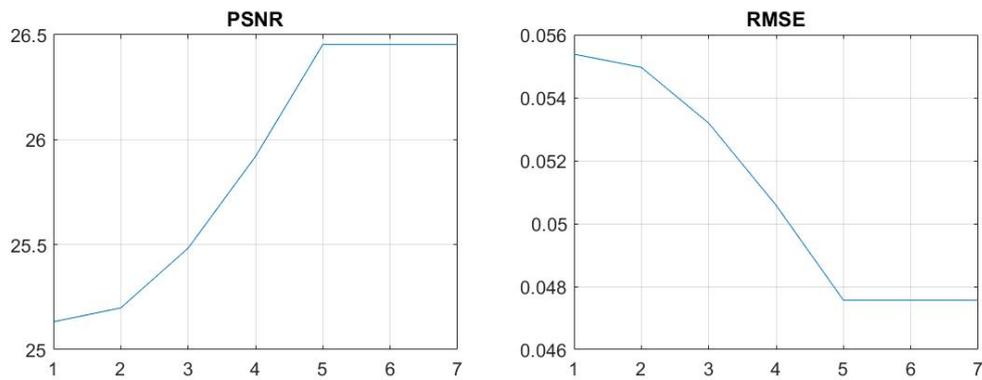
5.4.1 Immagine Brain

Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	6	25	26	0.7959	52.0753
SRFC	2	25	27	0.8031	165.4346

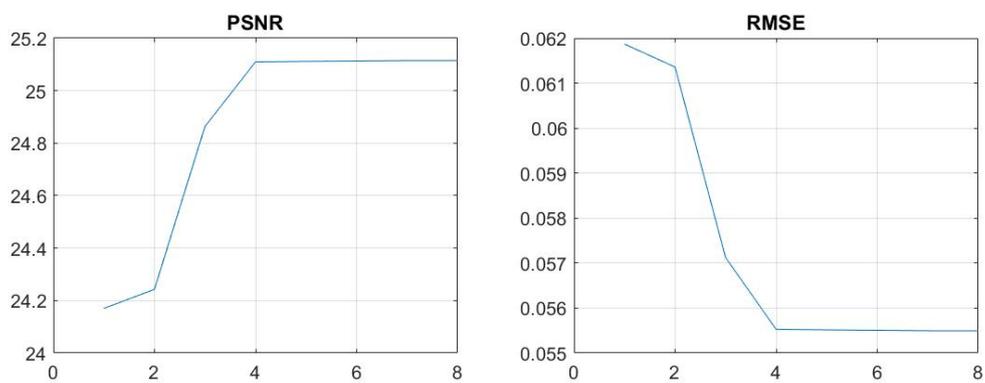
Tabella 5.5: Immagine Brain, regolarizzazione locale, upfactor 3

Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	7	24	25	0.7252	65.5306
SRFC	2	24	25	0.7375	135.0767

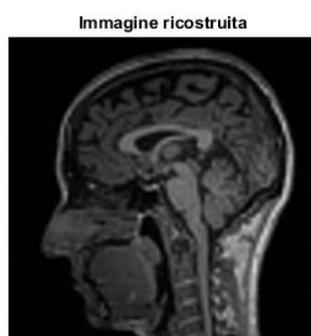
Tabella 5.6: Immagine Brain, regolarizzazione locale, upfactor 4



(a) PSNR immagine Brain, SPDHG, upfactor 3 (b) RMSE immagine Brain, SPDHG, upfactor 3



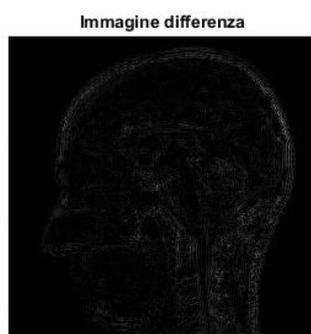
(a) PSNR immagine Brain, SPDHG, upfactor 4, (b) RMSE immagine Brain, SPDHG, upfactor 4



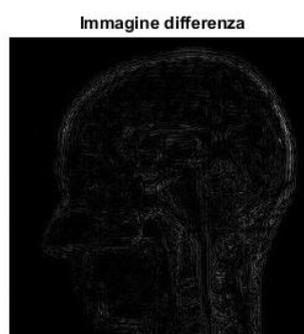
(a) SPDHG, upfactor 3



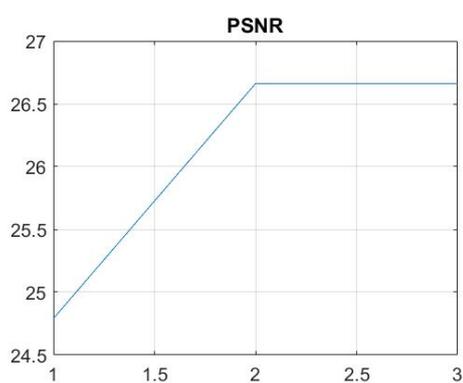
(b) SPDHG, upfactor 4



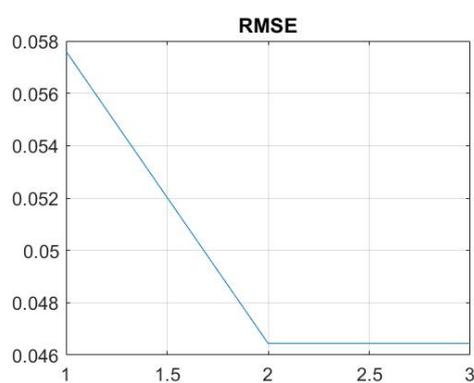
(a) SPDHG locale, upfactor 3



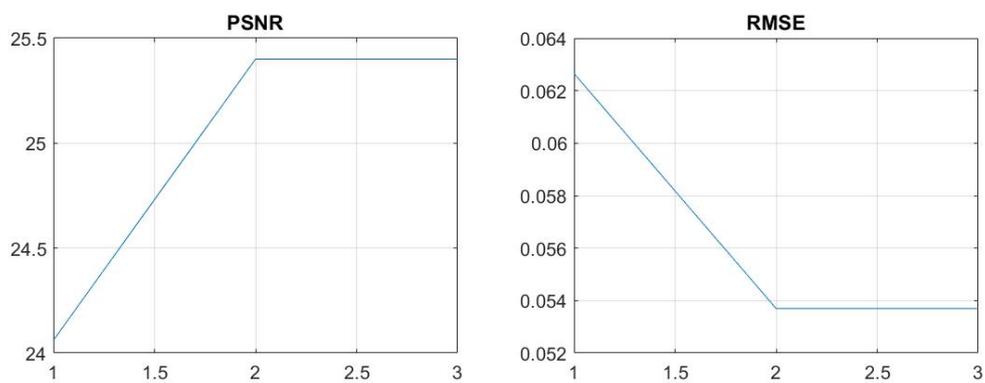
(b) SPDHG locale, upfactor 4



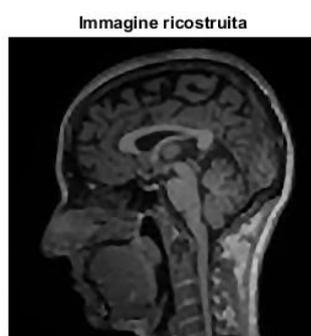
(a) PSNR immagine Brain, SRFC, upfactor 3



(b) RMSE immagine Brain, SRFC, upfactor 3



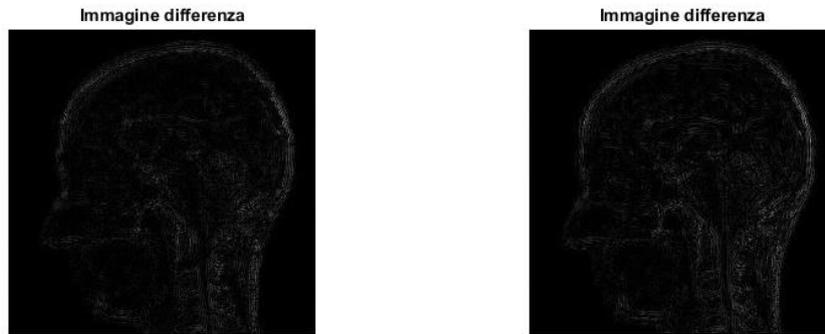
(a) PSNR immagine Brain, SRFC, upfactor 4 (b) RMSE immagine Brain, SRFC, upfactor 4



(a) SRFC, upfactor 3



(b) SPDHG, upfactor 4



(a) SRFC locale, upfactor 3

(b) SRFC locale, upfactor 4

Per l'algoritmo SPDHG con regolarizzazione locale, si possono ottenere miglioramenti modificando il valore di soglia o prendendo la formulazione in norma L_2 della Variazione Totale. Modificando inizialmente il solo valore di soglia e ponendolo pari a 10^{-1} , sono stati ottenuti i seguenti risultati:

PSNR finale	27
SSIM	0.8006
Iterazioni	2
Tempo	19.7763

Tabella 5.7: Brain, SPDHG locale, upfactor 3, soglia= 10^{-1}

Con l'ulteriore modifica della LWTV in norma 2,

PSNR finale	27
SSIM	0.8010
Iterazioni	2
Tempo	20.1367

Tabella 5.8: Brain, SPDHG locale, upfactor 3, soglia= 10^{-1} e TV in norma 2

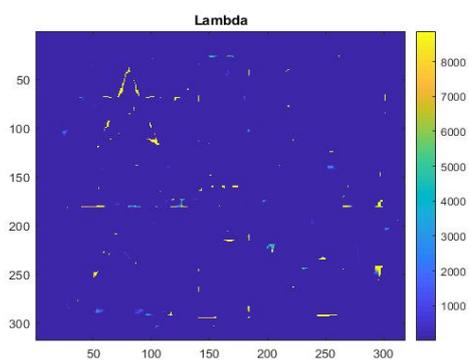
5.4.2 Immagine Geometric

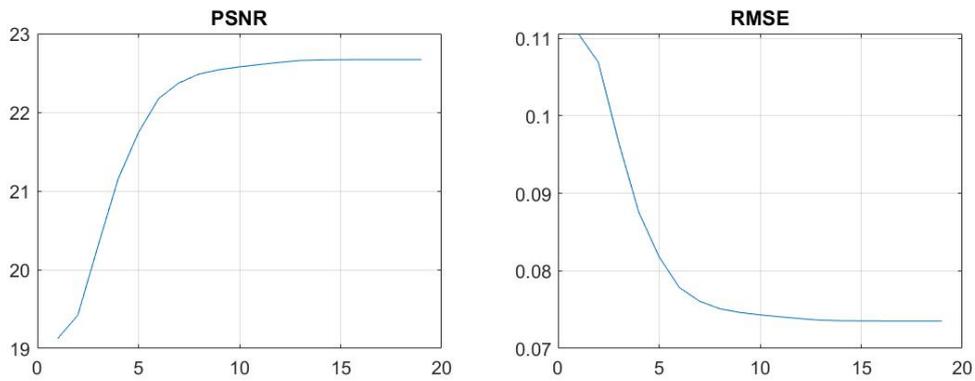
Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	18	19	23	0.8910	265.1208
SRFC	2	19	31	0.8986	119.8985

Tabella 5.9: Immagine Geometric, regolarizzazione locale, upfactor 3

Algoritmo	Iterazioni	PSNR iniziale	PSNR finale	SSIM	tempo
SPDHG	14	18	20	0.8366	211.4037
SRFC	3	18	29	0.8924	323.4106

Tabella 5.10: Immagine Geometric, regolarizzazione locale, upfactor 4

Figura 5.37: λ immagine Geometric, SPDHG, upfactor 3



(a) PSNR immagine Geometric, SPDHG, (b) RMSE immagine Geometric, SPDHG, upfactor 3

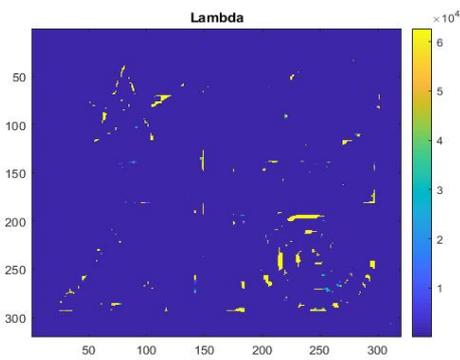
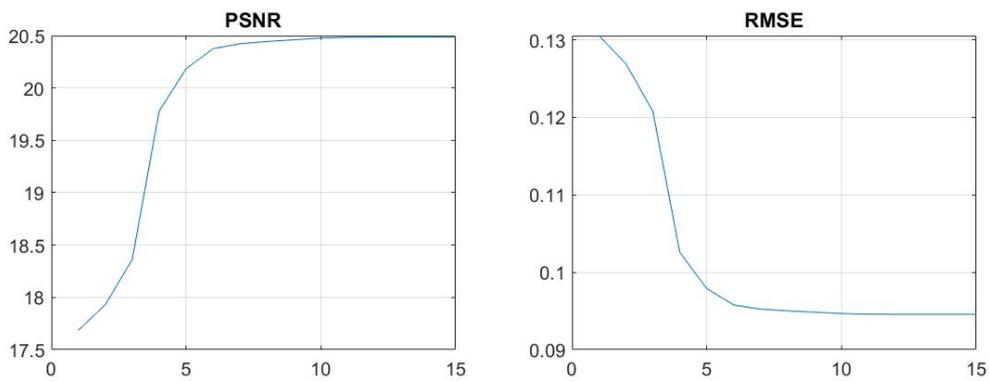


Figura 5.39: λ immagine Geometric, SPDHG, upfactor 4



(a) PSNR immagine Geometric, SPDHG, (b) RMSE immagine Geometric, SPDHG, upfactor 4



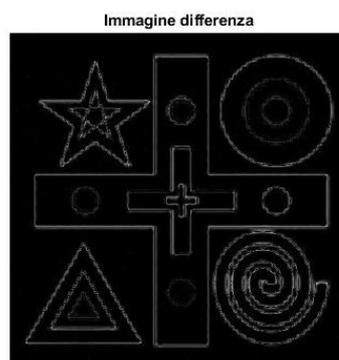
(a) SPDHG, upfactor 3



(b) SPDHG, upfactor 4



(a) SPDHG locale, upfactor 3



(b) SPDHG locale, upfactor 4

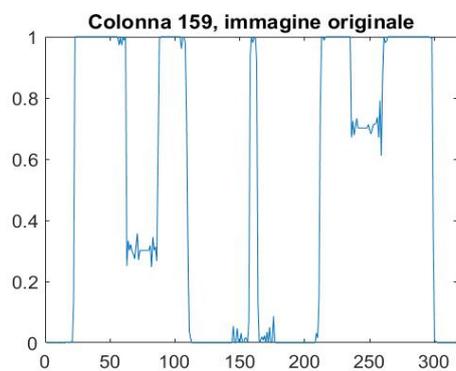
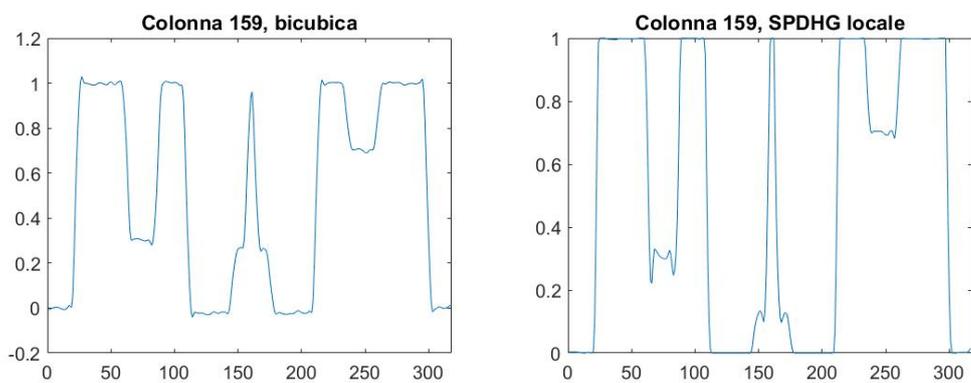


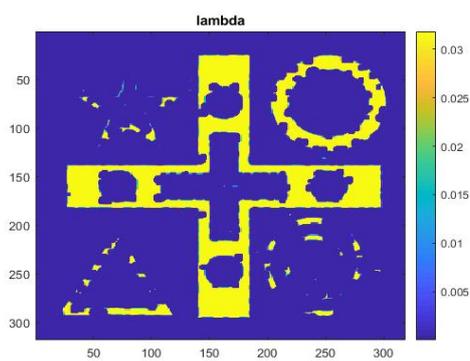
Figura 5.43: Colonna 159 dell'immagine originale Geometrica

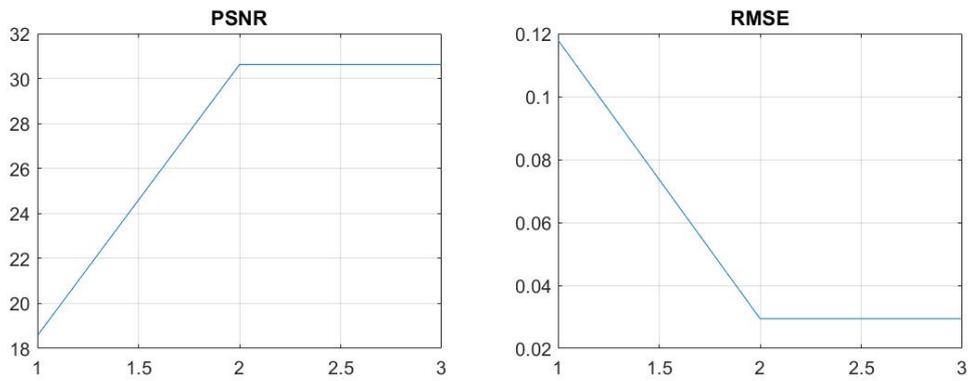


(a) Interpolazione bicubica, upfactor 3

(b) SPDHG locale, upfactor 3

Figura 5.44: Colonna 159 dell'immagine ricostruita

Figura 5.45: λ immagine Geometric, SRFC, upfactor 3



(a) PSNR immagine Geometric, SRFC, (b) RMSE immagine Geometric, SRFC, upfactor 3

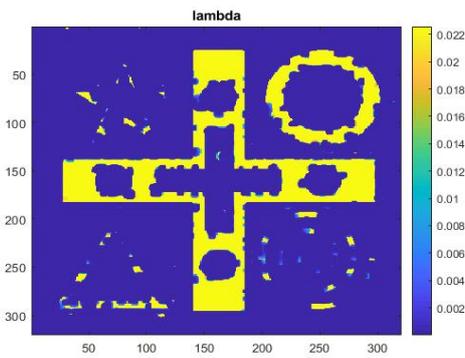
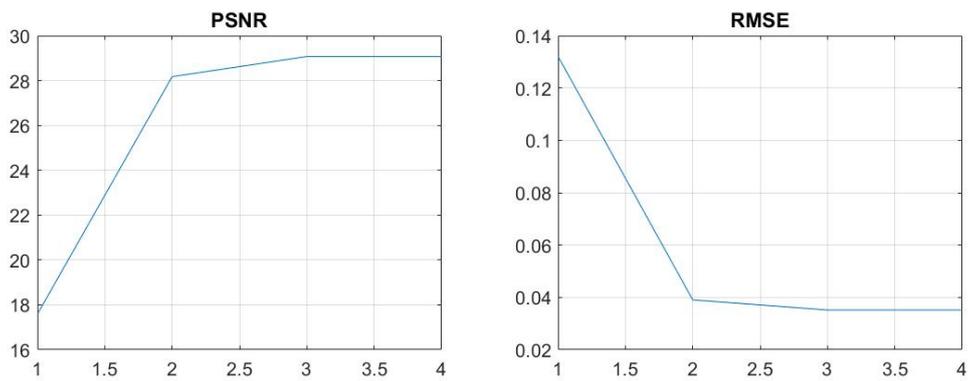


Figura 5.47: λ immagine Geometric, SRFC, upfactor 4



(a) PSNR immagine Geometric, SRFC, (b) RMSE immagine Geometric, SRFC, upfactor 4



(a) SRFC, upfactor 3



(b) SRFC, upfactor 4



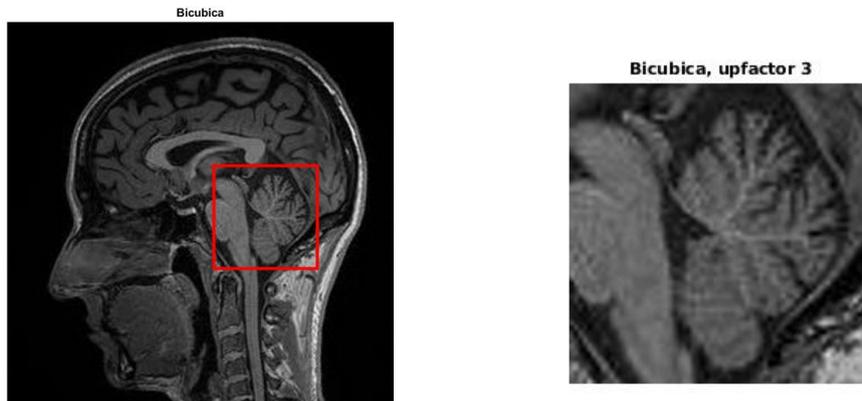
(a) SRFC locale, upfactor 3



(b) SRFC locale, upfactor 4

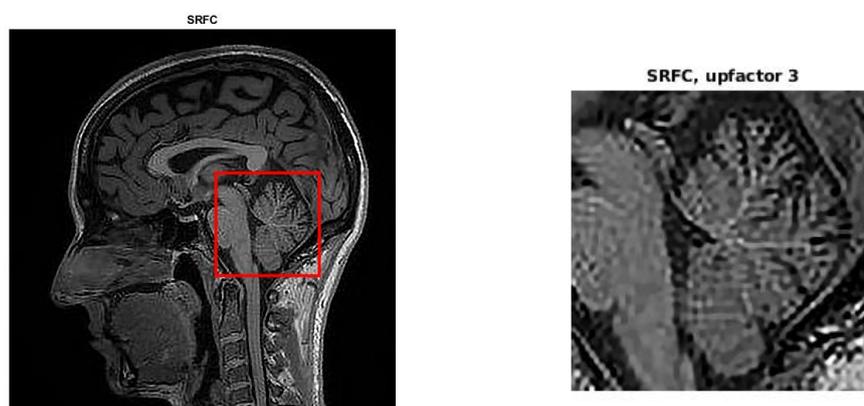
5.5 Up-sampling 3 per immagine Brain

Come ultima prova, è stato applicato l'algoritmo SRFC, con regolarizzazione globale e locale, all'immagine Brain 249×249 , utilizzata fino ad ora come immagine ad alta risoluzione di partenza. Su di essa, che viene in questo caso considerata come immagine LR, è stato applicato un up-sampling 3, fino ad ottenere un'immagine con risoluzione 747×747 . Non avendo un'immagine di riferimento di queste dimensioni, non è stato possibile calcolare i valori di PSNR o RMSE ed è quindi stato utilizzato il solo confronto visivo. I primi risultati che vengono presentati sono relativi alla prova con regolarizzazione globale. Viene mostrato un dettaglio dell'immagine ricostruita con interpolazione bicubica e di quella ricostruita con SRFC. Poi, viene riportato il grafico di una riga di entrambe le immagini ricostruite.



(a) Immagine ricostruita 747×747 (b) Dettaglio 200×200 a partire dal pixel (400, 280)

Figura 5.51: Interpolazione Bicubica, upfactor 3



(a) Immagine ricostruita 747×747 (b) Dettaglio 200×200 a partire dal pixel (400, 280)

Figura 5.52: Algoritmo SRFC

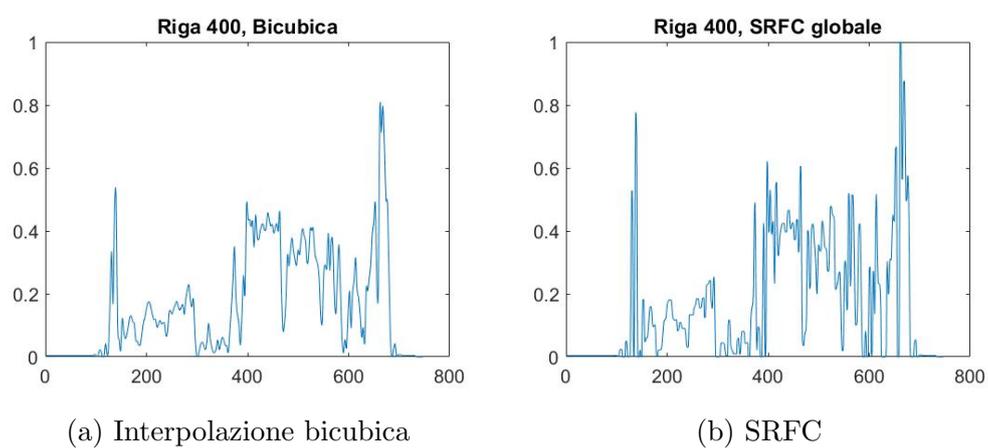
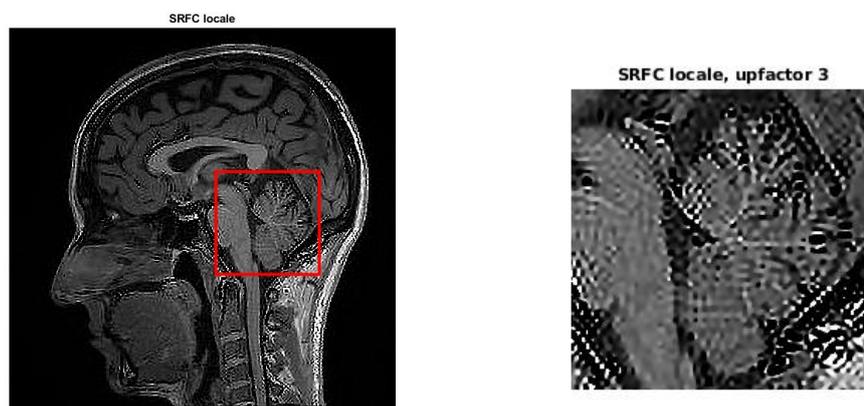


Figura 5.53: Riga 400 dell'immagine 747×747

Il test è stato eseguito anche con scelta del parametro di regolarizzazione a livello locale, mantenendo un upfactor pari a 3. Anche in questo caso, vengono riportati un dettaglio dell'immagine ricostruita ed il grafico di una sua riga.



(a) Immagine ricostruita 747×747 (b) Dettaglio 200×200 a partire dal pixel (400, 280)

Figura 5.54: Algoritmo SRFC locale

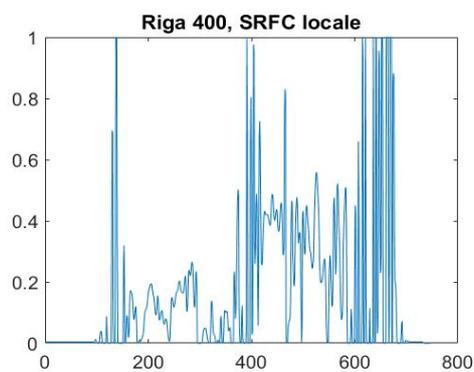


Figura 5.55: Riga 400 dell'immagine 747×747

Conclusioni

Il presente lavoro di tesi ha previsto lo studio e la sperimentazione del principio di Penalizzazione Uniforme per la scelta del parametro di regolarizzazione, a livello globale e locale, in un problema di Super-Risoluzione a Singola Immagine. Gli algoritmi utilizzati per la risoluzione del problema di ottimizzazione in esso contenuto sono stati l'algoritmo SPDHG e l'algoritmo SRFC, entrambi modificati per permettere la scelta locale del parametro di regolarizzazione. I metodi utilizzati si sono rivelati piuttosto versatili, tanto da consentire la generalizzazione al caso locale e il loro utilizzo sia su immagini LR ottenute sottocampionando e sfocando le immagini prese in input, sia direttamente su di esse, ottenendo immagini di dimensioni tre volte maggiori di quelle di partenza. I risultati trovati per le due immagini considerate hanno dimostrato come, in generale, il metodo SRFC sia più efficiente: presenta, infatti, valori migliori dei parametri di valutazione e degli errori considerati. Tuttavia, specialmente nel caso locale, tale metodo ha riportato un tempo d'esecuzione molto maggiore rispetto ad SPDHG, a causa dei vari pesi aggiuntivi previsti. Da osservare è anche la poca differenza tra i due metodi nel caso dell'immagine Brain. Ciò può essere spiegato considerando che tale immagine è caratterizzata da diversi livelli di grigio, a differenza dell'immagine Geometrica, che ha molte zone piatte. A discapito delle aspettative, non è stato rilevato un contributo importante nel passaggio dalla regolarizzazione globale a quella locale, fatto probabilmente dovuto alla scelta delle immagini. Sarebbe di grande interesse testare i metodi su un'immagine maggiormente regolare, che presenti variazioni più graduali tra i livelli di grigio, per verificare i miglioramenti che possono essere ottenuti con una regolarizzazione a livello locale. Il lavoro sperimentale svolto in questa tesi merita quindi ulteriori sviluppi, anche per quanto concerne la scelta dei parametri, il livello di rumore applicato, i valori di soglia e di tolleranza utilizzati. Inoltre, per estendere lo studio al caso di rumore diverso da quello gaussiano ed utilizzare la funzione di Kullback-Liebler come termine di consistenza dei dati, sarebbe utile implementare una procedura che utilizzi l'algoritmo BPG_e, presentato nel capitolo 4.

Bibliografia

- [1] Han Yu, Ting-Zhu Huang, Liang-Jian Deng, Xi-Le Zhao, *Super-resolution via a fast deconvolution with kernel estimation*, EURASIP Journal on Image and Video, 2017.
- [2] J. Yang, T. Huang, L.J. Deng, *Image super-resolution: Historical overview and future challenges*, CRC Press, 2017.
- [3] Xiaoya Zhang, Roberto Barrio, M. Angeles Martinez, Hao Jiang, Lizhi Cheng, *Bregman Proximal Gradient Algorithm with Extrapolation for a Class of Nonconvex Nonsmooth Minimization Problems*, IEEE Access, 2019.
- [4] X. Yang, J. Zhang, Y. Liu, X. Zheng, K. Liu, *Super-resolution image reconstruction using fractional-order total variation and adaptive regularization parameters*, Springer-Verlag GmbH, Germany, 2018.
- [5] D. Lazzaro et al, *A fast subgradient algorithm in image super-resolution*, Journal of Physics: Conf, 2017.
- [6] Curtis R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
- [7] P. Cascarano, *Algoritmi di Super Resolution per immagini termiche*, Tesi magistrale, Alma Mater Studiorum Università di Bologna, a.a. 2017-2018.
- [8] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 2006.
- [9] V. Bortolotti, G. Landi, F. Zama, *2DNMR data inversion using locally adapted multi-penalty regularization*, 2020.
- [10] K. Miller, *Least Squares Methods for Ill-Posed Problems with a Prescribed Bound*, SIAM J. Math Anal, 1970.

-
- [11] V. Bortolotti, R.J.S. Brown, P. Fantazzini, G. Landi, F. Zama, *Uniform Penalty inversion of two-dimensional NMR relaxation data*, IOP science, 2016.
 - [12] A. Chambolle, *An Algorithm for Total Variation Minimization and Application*, J. Math. Imag. Vis., 2004.
 - [13] A. Beck, M. Teboulle, *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM Journal On Imaging Science, 2009.
 - [14] J. Duran, B. Coll, C. Sbert, *Chambolle's Projection Algorithm for Total Variation Denoising*, IPOL, 2013.
 - [15] K. Ito, B. Jin, T. Takeuchi, *A Regularization Parameter for NonSmooth Tikhonov Regularization*, SIAM, 2011.
 - [16] S. Bonettini, A. Benfenati, V. Ruggiero, *Scaling Techniques for ϵ -Subgradient Methods*, SIAM, 2016.