POLITECNICO DI MILANO

SCUOLA DI INGEGNERIA DEI SISTEMI

Corso di Laurea Magistrale in
Ingegneria Matematica

ADATTAZIONE ANISOTROPA DI GRIGLIA APPLICATA ALLA SEGMENTAZIONE DI IMMAGINI

Relatore :
Prof. Simona PEROTTO

Correlatore :
Prof. Luca FORMAGGIA

Tesi di Laurea di :
Nicoletta PAPUCCI
Matr. 750733

Anno Accademico 2010/2011

Sommario

In questo lavoro di tesi consideriamo il problema della segmentazione delle immagini nell'ambito di una risoluzione agli elementi finiti. Incorporiamo il metodo *Region-Scalable Fitting Energy* con il noto algoritmo di Split Bregman, in modo da ottenere una tecnica robusta e efficiente per la segmentazione di immagini con intensità disomogenea. Successivamente usiamo un approccio adattativo, basato su uno stimatore anisotropo a posteriori dell'errore di discretizzazione. Attraverso diversi esempi applicativi, confrontiamo i risultati prodotti attraverso la procedura adattativa con quelli ottenuti senza adattazione o con una strategia di adattazione euristica. In particolare mostriamo come le griglie adattate siano più efficienti, sia in termini di costo computazionale (cioè constano di un minor numero di elementi), sia per la maggiore regolarità delle segmentazioni così ottenute.

Abstract

In this work we deal with the image segmentation problem by resorting to the finite element approach. We merge the Region-Scalable Fitting Energy method with the well-known Split Bregman algorithm, in order to obtain a robust and efficient technique suited to segment images with intensity inhomogeneity. Then we use an adaptive approach based on an anisotropic a posteriori error estimator for the discretization error associated to the finite element discretization. By means of several practical examples, we compare the results yielded via the adaptive procedure with those obtained without any grid adaptation or with a heuristic mesh adaptation strategy. In particular we show that the adapted triangulations are more efficient, since they allow to improve the quality of the image contours (essentially in terms of smoothness), while reducing the number of mesh elements (i.e. the computational cost).

Indice

Introduzione

Il trattamento delle immagini è una disciplina molto vasta che comprende l'insieme delle tecniche di analisi e modifica delle immagini digitali: tra queste citiamo, a titolo d'esempio, il *deblurring*, il *denoising*, l'*inpainting*, la segmentazione e la ricostruzione di superfici. Lo scopo comune a queste discipline è l'estrazione di informazioni a partire dall'immagine oppure il miglioramento della qualità della stessa. Le applicazioni possono essere svariate, in campo medico, artistico o tecnologico (si veda, ad esempio, [20, 38, 41]).

Lo spiccato interesse di questi problemi ha suscitato, a partire dagli anni '90, una crescente attenzione della comunità scientifica nei confronti della definizione di modelli e metodi matematici in grado di descrivere e gestire tali processi. Questi modelli e questi metodi fanno essenzialmente riferimento alla teoria delle equazioni differenziali a derivate parziali o al calcolo variazionale. Risulta evidente quindi la necessità di discretizzare le equazioni di interesse, in modo da poterne calcolare almeno una soluzione approssimata. Nell'ambito del trattamento delle immagini, tra i metodi di discretizzazione più ricorrenti citiamo il metodo alle differenze finite ([19, 22, 74]) e il metodo agli elementi finiti ([17, 58, 32]). Il primo è quello di gran lunga più inflazionato a causa delle proprietà intrinseche delle immagini digitali (ogni pixel può infatti essere associato ad un nodo della griglia strutturata tipica di un metodo alle differenze finite). L'uso di una discretizzazione agli elementi finiti risulta invece sicuramente più atipico.

Il presente lavoro di tesi può essere collocato in questa seconda classe di contributi: si tratta di una scelta certamente innovativa, forse un po' azzardata, ma consapevole. Infatti competenze specifiche maturate nell'ambito dell'adattazione anisotropa di griglia, unite ad una buona dose di curiosità intellettuale ci hanno spinto ad applicare ad un nuovo campo, quale quello della segmentazione delle immagini (cioè riconoscimento dei suoi contorni), tecniche di adattazione già ampliamente consolidate in svariati contesti. In particolare, ricorreremo ad una adattazione anisotropa di griglia guidata da stimatori a posteriori per l'errore di discretizzazione.

L'idea alla base dell'utilizzo di tecniche di adattività di griglia in questo frangente è la seguente: i contorni degli oggetti contenuti in un'immagine occupano una regione limitata della stessa, quindi sembra inutilmente

costoso risolvere il problema differenziale in modo accurato su tutto il dominio. Al contrario sembra computazionalmente più proficuo aumentare l'accuratezza della risoluzione proprio nelle aree dell'immagine corrispondenti alla posizione dei suoi contorni. Questo obiettivo può essere facilmente raggiunto a patto di avere una griglia adattata in corrispondenza di tali contorni. In questa tesi si mira proprio a verificare, attraverso ampia validazione numerica, come le griglie adattate permettano di ridurre la complessità del problema in esame. Notiamo che con una discretizzazione alle differenze finite si sarebbe potuto fare adattazione di griglia solamente usando griglie non conformi, tipologia di griglia non di nostro interesse.

Nel primo capitolo della tesi, dopo aver descritto brevemente la struttura delle immagini digitali, introduciamo formalmente il problema della segmentazione di un'immagine. La trattazione segue l'ordine cronologico di sviluppo dei vari modelli, a partire dal pionieristico lavoro di D. Mumford e J. Shah, fino alle sue evoluzioni più recenti. Concludiamo descrivendo il modello, detto *Region Scalable Fitting Energy*, che abbiamo scelto per analizzare gli eventuali vantaggi apportati da un'adattazione di griglia.

Tutti gli approcci per la segmentazione di un'immagine presentati nel primo capitolo sono basati sulla minimizzazione di opportuni funzionali non necessariamente convessi che, come tali, non ammettono necessariamente un unico minimo globale. Per questo motivo nel secondo capitolo introduciamo un algoritmo di segmentazione delle immagini basato sull'uso di funzionali convessi, derivato da uno dei modelli presentati nel primo capitolo. In seguito descriviamo un algoritmo di minimizzazione particolarmente efficace proprio per la risoluzione di problemi convessi.

Il terzo capitolo tratta il problema dell'adattività di griglia. Innanzitutto forniamo una panoramica delle diverse tecniche di adattività ricorrenti in letteratura, sia quelle euristiche, sia quelle guidate da stime dell'errore di discretizzazione. Ci soffermiamo successivamente sulla teoria dei cosiddetti stimatori a posteri di tipo *recovery-based*, che sono poi quelli da noi utilizzati per la segmentazione delle immagini.

Nel quarto capitolo ci concentriamo dapprima sul contesto anisotropo per l'adattazione di griglia. Successivamente forniamo alcuni esempi di stime anisotrope sia a priori sia a posteriori per l'errore di discretizzazione. Infine spieghiamo come tali stime possano essere utilizzate nella pratica per guidare un'adattazione anisotropa di griglia.

L'ultimo capitolo è dedicato alla descrizione dell'algoritmo implementato e all'analisi dei diversi casi test. In un primo momento ci focalizziamo sui vantaggi apportati da un'adattività di griglia di tipo euristico rispetto alla soluzione del problema su una griglia non adattata. Confrontiamo poi tali risultati con quelli ottenuti usando una procedura più rigorosa, guidata da uno stimatore a posteriori anisotropo dell'errore.

Capitolo 1

La segmentazione delle immagini

Le immagini che popolano la vita quotidiana - in fotografie, giornali o film - non sono altro che proiezioni bidimensionali di oggetti tridimensionali. Nella realtà tali oggetti sono facilmente riconoscibili come entità distinte, grazie alle loro caratteristiche geometriche e alla loro posizione nello spazio. Questo processo innato di separazione effettuato dal nostro cervello può essere riprodotto in maniera automatica sulle immagini digitali tramite la cosiddetta *segmentazione*, il cui studio sarà argomento di questo capitolo. Lo scopo della segmentazione è proprio quello di semplificare la rappresentazione delle immagini suddividendole nelle loro regioni più significative, che possono essere localizzate sfruttando caratteristiche salienti come la posizione dei bordi o l'intensità del colore. Più precisamente, la segmentazione è il processo con il quale si classificano i pixel dell'immagine che hanno caratteristiche (colore, intensità, texture) comuni, separandoli dalle regioni adiacenti con caratteristiche significativamente differenti.

L'ambito è ovviamente quello del trattamento delle immagini e le applicazioni sono svariate. Ad esempio, l'impiego delle tecniche di segmentazione è molto diffuso in medicina, dove tali tecniche possono essere sfruttate per effettuare diagnosi, misurare l'estensione di un tessuto o eseguire operazioni chirurgiche assistite dal computer [57]. Inoltre, il riconoscimento di contorni effettuato su una sequenza di immagini, come quella fornita da una risonanza magnetica, permette di ricostruire la forma tridimensionale degli oggetti [76]. Altri campi di applicazione sono l'intelligenza artificiale o la *machine vision*, scienza che si occupa di fornire tecnologie di controllo e applicazioni industriali basate sull'analisi automatica delle immagini [69].

Grazie alla varietà di applicazioni e a causa della difficoltà del problema, nel tempo sono stati sviluppati numerosi algoritmi per la segmentazione delle immagini e la ricerca è a tutt'oggi ancora attiva. Nella Sezio-

ne 1.2 citiamo gli approcci principali, per poi concentrare l'attenzione sui metodi basati su equazioni a derivate parziali nelle sezioni successive. In particolare, il modello di Mumford e Shah, ricordato nella Sezione 1.3, ha rappresentato il primo tentativo di inquadrare il problema della segmentazione come minimizzazione di un funzionale il cui valore dipende dalle informazioni estrapolate dall'immagine. Esso ha posto le basi dei metodi di segmentazione sviluppati negli anni successivi, i più noti dei quali sono presentati in questo capitolo in ordine cronologico.

I metodi ai contorni attivi, dettagliati nella Sezione 1.4, mirano ad ottenere una descrizione precisa della geometria dei contorni sfruttando la nozione di funzione di *level-set*. Come si vedrà nel seguito il vantaggio di tali metodi rispetto a quello di Mumford-Shah risiede nella maggiore semplicità della loro trattazione teorica.

Un problema quasi insormontabile nell'ambito del trattamento delle immagini deriva dalle caratteristiche fortemente differenti che possono assumere le figure! Infatti esistono immagini dai contorni molto ben definiti, altre invece in cui i contorni non esistono e sono difficilmente riconoscibili anche ad occhio nudo (si pensi, ad esempio, alla rappresentazione di una galassia). Per questo motivo si vedrà che non esiste un approccio sempre vincente; nella trattazione si cercherà di evidenziare i limiti di ogni metodo e di introdurre la tecnica che permette di sormontarli, nel caso essa esista.

Purtroppo gli approcci presentati in Sezione 1.4 sono basati sulla minimizzazione di funzionali non necessariamente convessi, di cui quindi non è possibile trovare un minimo globale tramite le usuali tecniche del calcolo delle variazioni. Gli algoritmi che permettono di riformulare questi metodi come problemi di minimizzazione convessa e di trovarne la soluzione unica verranno presentati nel prossimo capitolo.

1.1 Immagini digitali

L'acquisizione di un'immagine è il processo di conversione analogico-digitale della stessa. Tale processo è composto da una prima fase di filtraggio, a cui segue il campionamento del segnale continuo, e infine la quantizzazione dei campioni. Così facendo si ottiene una rappresentazione discreta dell'immagine, che può essere di tipo *vettoriale* o *bitmap*.

Alla prima categoria appartengono le immagini formate da un insieme di primitive geometriche (quali punti, linee, poligoni), assemblate poi per ottenere le forme più complesse. Le immagini vettoriali sono immagazzinate nei computer attraverso le equazioni matematiche di tali forme geometriche, e non hanno una risoluzione fissa. Possono infatti essere riscalate arbitrariamente per produrre una figura con la risoluzione desiderata. Questo tipo di rappresentazione è adatto per descrivere figure semplici, ed è particolarmente utilizzato per il rendering tridimensionale.

Diversamente, le immagini bitmap sono costituite da una matrice di punti, detti *pixel* (dall'inglese picture element), a ognuno dei quali è assegnato un valore ottenuto con il processo di quantizzazione. Tale valore dipende dalle caratteristiche di intensità e di colore della sorgente analogica nel punto corrispondente al pixel. La descrizione dell'immagine è quindi effettuata punto per punto; il numero dei pixel, corrispondente al livello di discretizzazione, influenza la risoluzione e di conseguenza la qualità dell'immagine. Si possono distinguere diversi tipi di immagini in base alle informazioni processate per assegnare un valore ai pixel. Le più comuni sono due:

- immagini in scala di grigi: ogni pixel assume un valore che indica l'intensità del grigio di quel punto.
- immagini a colori: ogni pixel contiene informazioni sul livello di intensità dei colori fondamentali. Nel modello di colore RGB, uno dei più utilizzati, i colori fondamentali sono tre: rosso, verde e blu.

Il numero di toni di grigio (o di colori) che possono essere rappresentati dipende dalla quantità di bit utilizzati per descrivere un pixel. Questo numero, la cosiddetta *profondità* p, limita il numero dei valori i che può assumere l'intensità luminosa in ogni pixel:

$$i \in \mathbb{N} : i \leq 2^p - 1.$$

Per $p = 1$ si ottiene un'immagine in bianco e nero, per $p = 8$ un'immagine a 256 toni di grigio (o colori), e infine per $p = 24$ un'immagine costituita da 16,7 milioni di toni.
Concentreremo la nostra trattazione sulle immagini bitmap a scala di grigi, in quanto ogni immagine a colori possiede un'analoga rappresentazione in scala di grigi che non compromette l'esito della segmentazione.

Al fine di inquadrare formalmente il problema della segmentazione, introduciamo una rappresentazione analitica delle immagini digitali. Innanzitutto il dominio aperto e limitato $\Omega \subset \mathbb{R}^2$ è il supporto dell'immagine stessa, spesso costituito da un semplice rettangolo. L'immagine può essere pensata come una funzione

$$u(\mathbf{x}) : \Omega \to [0, 255], \quad \forall \mathbf{x} \in \Omega$$

il cui valore in $\mathbf{x}$ corrisponde all'intensità locale della luce.

1.2 Alcuni algoritmi euristici

I numerosi approcci alla segmentazione presenti in letteratura derivano dalla richiesta di segmentare immagini intrinsecamente molto diverse tra loro. Infatti non esiste una soluzione universale al problema della segmentazione, e spesso è necessario combinare tecniche differenti per ottenere i

risultati desiderati. L'obiettivo di questa sezione è innanzitutto quello di fornire una panoramica di alcuni metodi di segmentazione che chiameremo *euristici*, per poi proseguire con l'introduzione di un'ambientazione più formale.

Thresholding

Si tratta della forma più semplice di segmentazione, largamente diffusa nei software di grafica. In pratica consiste nel confrontare l'intensità di ogni pixel con un dato valore di soglia: se inferiore, il pixel apparterrà allo sfondo, altrimenti sarà in primo piano. Come si può immaginare, i risultati ottenuti con questo metodo sono molto approssimativi, e la ricerca del valore di tolleranza ottimale non è immediata [67].

Clustering

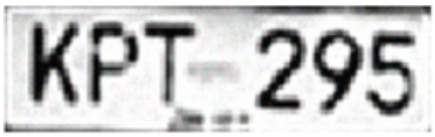
Il clustering consiste nella suddivisione del dominio iniziale in sottodomini attraverso una tecnica iterativa. L'algoritmo più diffuso è il *K-mean*: inizialmente si partiziona l'immagine in K regioni (o cluster) e si trovano i loro centri. Poi ogni pixel viene associato al cluster il cui centro è più vicino in termini di colore, intensità, posizione o una media pesata di questi fattori. In seguito si ricalcola il valore centrale dei cluster e si ripete lo step precedente fino a convergenza. La soluzione ottenuta non è necessariamente quella ottima. Esistono diversi algoritmi di clustering, che si differenziano sia per il criterio con cui si sceglie il numero di cluster, sia per il tipo di logica iterativa (fuzzy, probabilistica, split and merge) [48].

Region-growing

Il primo metodo di questo tipo comparso in letteratura è il cosiddetto *seeded region growing* [1]. Questo metodo richiede in input una serie di regioni (dette semi), ognuna corrispondente ad un oggetto da segmentare. In seguito le regioni vengono allargate aggiungendo il pixel tra quelli confinanti con l'intensità più simile al valore medio della regione; si procede così finché non si allocano tutti i pixel. Una variante di tale metodo sfrutta le informazioni statistiche di ogni regione per valutare, tramite test d'ipotesi, quali pixel inglobare. Altre varianti invece non richiedono l'imposizione delle condizioni iniziali, cioè l'assegnazione dei seeds [61].

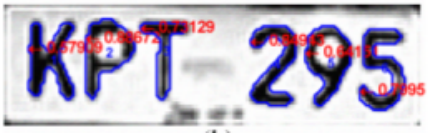
Edge-detection

Il riconoscimento dei contorni è un campo a sé stante e ben sviluppato del trattamento delle immagini. La sola individuazione dei bordi non permette tuttavia di riconoscere le regioni significative di un'immagine: infatti molti bordi possono non essere chiusi o formare complesse intersezioni. Si tratta comunque di un buon punto di partenza, motivo per cui queste tecniche vengono spesso usate come basi per altri metodi di segmentazione [80].



(a) Immagine originale

(b) Thresholding multiplo



(c) Clustering

(d) Edge detection

(e) Split-and-Merge

Figura 1.1: Confronto tra diverse tecniche di segmentazione. I valori in rosso indicano la media pesata della distanza tra gli oggetti originali e quelli trovati con segmentazione.

Grafi

In questo caso l'immagine viene rappresentata come un grafo pesato, in cui i nodi sono associati ad uno o più pixel e i pesi dei lati indicano la somiglianza tra i pixel adiacenti. Per partizionare in modo efficace i nodi si possono sfruttare vari algoritmi (random walker, minimum spanning tree, tagli...) [68, 44].

Split and Merge

Questo algoritmo iterativo analizza l'immagine iniziale e, nel caso essa non sia omogenea, la divide in quattro sottodomini; procede poi in modo analogo sui sottodomini così individuati. Viceversa, se quattro sottodomini sono omogenei, vengono uniti. Questo procedimento ricorsivo si applica finché non si ottiene l'immagine segmentata [46].

In Figura 1.1 si riporta uno studio comparativo tra quattro delle tecniche di segmentazione citate, tratto da [65], in cui si possono apprezzare pregi e difetti di ciascun metodo. In 1.1(b) si utilizza un metodo di thresholding multiplo, cioè si fissano più valori di soglia: quando il numero di classi da separare è troppo alto si perde precisione e, come risultato, anche i contorni dell'immagine vengono segmentati. Molto più preciso è il clustering mostrato in 1.1(c), nonostante soffra di dipendenza dai dati iniziali e sia necessario effettuarlo più volte per ottenere un buon risultato. La seg-

mentazione basata sul riconoscimento dei contorni mostrata in figura 1.1(d) non riesce ad identificare con precisione tutti i contorni: si nota infatti che i bordi evidenziati sono leggermente disconnessi. Il risultato peggiore è mostrato in 1.1(e), in cui l'algoritmo di split-and-merge con un unico valore di soglia non riesce a catturare la complessità dell'immagine. Si noti infine come nessuno di questi metodi riesca a riconoscere, in particolare, i contorni interni della lettera P e del numero 9.

La scarsa qualità della segmentazione ottenuta attraverso tali metodi euristici giustifica l'esigenza di un'analisi più teorica del problema. Segmentare un'immagine significa trovare una curva Γ che partiziona il dominio Ω nell'insieme delle sue componenti connesse Ω_j:

$$\Omega = \bigcup_j \Omega_j$$

in modo tale che l'immagine iniziale u_0 sia discontinua attraverso i bordi Γ e viceversa le sue variazioni all'interno di ogni regione Ω_j siano di piccola entità. Ogni Ω_j corrisponde ad un oggetto fisico, o ad una forma geometrica facilmente distinguibile di cui si vuole determinare il contorno $\partial\Omega_j$.

L'obiettivo quindi è duplice: si vuole ottenere una partizione dell'immagine nelle sue regioni significative $\{\Omega_j, j = 1, \ldots\}$ e, allo stesso tempo, descrivere il contorno Γ di tali regioni come entità geometrica. La difficoltà risiede proprio nel formulare un unico problema che risponda ad entrambe queste necessità. Una soluzione formalmente rigorosa si basa sulla minimizzazione di un opportuno funzionale energia ed è descritta nella prossima sezione.

1.3 Modello di Mumford-Shah

Mumford e Shah proposero nel 1989 un modello di segmentazione innovativo: per la prima volta un'analisi teorica rigorosa unifica la ricerca delle regioni di un'immagine a quella del loro contorno [53]. Nonostante questo metodo soffra di una trattazione teorica abbastanza complicata e la buona positura del problema sia garantita unicamente in uno spazio funzionale piuttosto ristretto, esso ha posto le basi per le più recenti evoluzioni nel campo della segmentazione.

Il problema è formulato come minimizzazione di un funzionale energia rispetto alle incognite u e K, rispettivamente approssimazione continua a tratti dell'immagine e insieme dei contorni ossia delle discontinuità di u. Richiedere a priori la regolarità di K sarebbe troppo restrittivo; si cerca quindi la soluzione in una classe più ampia di funzioni non continue, dove la lunghezza è data dalla misura di Hausdorff (N-1)-dimensionale $\mathcal{H}^{N-1}(K)$, di cui si riporta la definizione [63].

Definizione: *Per ogni sottoinsieme F di* $\mathbb{R}^n$ *ed ogni numero reale non negativo s la* ***misura di Hausdorff*** *s dimensionale di F è data da*

$$\mathcal{H}^s(F) = \lim_{\delta \to 0^+} H^s_\delta(F) = \sup_{\delta > 0} \mathcal{H}^s_\delta(F)$$

dove, per $0 < \delta < +\infty$,

$$\mathcal{H}^s_\delta(F) = \frac{\omega_s}{2^s} \inf \left\{ \sum_{j=1}^{+\infty} (diam\ U_j)^s : \quad F \subset \bigcup_{j=1}^{+\infty} U_j,\ diam\ U_j \leq \delta \right\}$$

con $diam\ U_j = \sup\{|x - y| : x, y \in U_j\}$. *La costante*

$$\omega_s = \pi^{s/2} \left(\int_0^\infty e^{-x} x^{s/2} dx \right)^{-1}$$

è positiva e finita per ogni s.

Si cerca quindi la coppia (u, K) che minimizza il funzionale

$$F(u, K) = \int_{\Omega \setminus K} (u - u_0)^2 dx + \alpha \int_{\Omega \setminus K} |\nabla u|^2 dx + \beta \mathcal{H}^{N-1}(K), \tag{1.1}$$

dove K è un insieme chiuso, u_0 è l'immagine iniziale, α e β sono delle costanti di scala positive che giocano il ruolo di fattori di penalizzazione e $u \in H^1(\Omega \setminus K)$. Si noti che, nel seguito, si usa la notazione standard per indicare gli spazi di funzioni Lebesgue misurabili [64]:

- $C^{n,\alpha}(\Omega)$, con n intero non negativo e $0 \leq \alpha \leq 1$, è lo spazio delle funzioni f che soddisfano, insieme con le loro derivate fino all'ordine n-esimo, la condizione di Hölder:

  $$|f(x) - f(y)| \leq C|x - y|^\alpha$$

  per qualche costante positiva C.

- $L^p(\Omega)$, con $1 \leq p < \infty$, è lo spazio di Banach (per $p = 2$ di Hilbert) delle funzioni misurabili:

  $$f : \Omega \to \mathbb{R} \ \text{ t.c. } \ \|f\|_p = \left(\int_\Omega |f(x)|^p dx \right)^{1/p} < \infty.$$

- $L^\infty(\Omega)$ è lo spazio di Banach delle funzioni limitate quasi ovunque:

  $$f : \Omega \to \mathbb{R} \ \text{ t.c. } \ \|f\|_\infty = \inf\{C \geq 0 : |f(x)| \leq C \text{ q.o. } \} < \infty.$$

- $W^{k,p}(\Omega)$ è lo spazio di Sobolev delle funzioni

$$f : \Omega \to \mathbb{R} \in L^p(\Omega) \text{ t.c. } D^\alpha f \in L^p(\Omega)$$

per ogni multi-indice $\forall \alpha \in \mathbb{N}^2$ con $|\alpha| \leq k$. In particolare vale $H^k(\Omega) = W^{k,2}(\Omega)$.

Il significato dei tre termini costituenti il funzionale F in (1.1) rispecchia gli obiettivi principali di un problema di segmentazione. Il primo termine è il cosiddetto termine di fedeltà, in quanto assicura che u sia una buona approssimazione in norma $L^2(\Omega \setminus K)$ dell'immagine iniziale. Il secondo permette di identificare le regioni, i cui contorni sono individuati da un valore elevato di ∇u; il terzo previene un'eccessiva segmentazione.

Gli autori in [53] fanno la seguente congettura riguardo all'esistenza e alla regolarità della soluzione del problema di minimo (1.1):

Congettura 1 (di Mumford-Shah)
Esiste una coppia (u, K) che minimizza F tale che l'insieme di discontinuità di K è l'unione di un numero finito di curve $C^{1,1}$. Inoltre, ogni curva può terminare solo in due modi: o con un'estremità libera, o in una giunzione tripla in cui tre curve si incontrano con un angolo di $2\pi/3$.

La difficoltà nello studio di (1.1) deriva dalla diversa natura delle incognite: u è una funzione definita in uno spazio N-dimensionale, mentre K è un insieme (N-1)-dimensionale. Per avere un risultato di esistenza e unicità del minimo servendosi del metodo diretto del calcolo delle variazioni, il problema deve essere ambientato in una topologia in cui F sia semicontinuo inferiormente e le sequenze minimizzanti siano compatte [70]. Sia E un insieme di Borel di $\mathbb{R}^N$ con frontiera topologica ∂E; allora si può facilmente provare che la mappa $E \to \mathcal{H}^{N-1}(\partial E)$ non gode di semicontinuità inferiore rispetto a nessuna topologia compatta. Per ovviare a questo problema è necessario sostituire in (1.1) l'incognita K con l'insieme dei salti di u definito come $S_u = \{t \in \mathbb{R} : u^-(t) \neq u^+(t)\}$. Si ottiene il funzionale

$$G(u) = \int_\Omega (u - u_0)^2 dx + \alpha \int_\Omega |\nabla u|^2 dx + \beta \mathcal{H}^{N-1}(S_u).$$

L'incognita u dovrebbe appartenere allo spazio $BV(\Omega)$

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

delle funzioni a variazione limitata. La variazione totale di u si definisce come:

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

dove $C_c^1(\Omega, \mathbb{R}^n)$ è lo spazio delle funzioni di $\mathbb{R}^n$ continue e differenziabili, con supporto compatto contenuto in Ω. BV è uno spazio di Banach dove, per ogni $u \in BV(\Omega)$, sono definite la norma e la seminorma:

$$\|u\|_{BV} = \|u\| + V(u, \Omega), \qquad |u|_{BV} = V(u, \Omega). \tag{1.2}$$

Allo spazio BV appartengono alcune funzioni (come ad esempio quella di Vitali-Cantor) cosiddette patologiche a causa della loro particolarità di essere continue e monotone ma con differenziale nullo quasi ovunque. Per queste funzioni si ha che

$$\inf_{u \in BV(\Omega)} G(u) = 0,$$

quindi la ricerca di un minimo globale risulta inutile. Il minimo di $G(u)$ deve essere cercato in uno spazio più ristretto, che escluda le funzioni patologiche citate precedentemente. L'esistenza di una soluzione viene provata da De Giorgi et al. in [28], introducendo il nuovo spazio funzionale $SBV(\Omega)$ delle funzioni speciali a variazione limitata. Per definire questo spazio bisogna partire dall'espressione della derivata nel senso delle distribuzioni di $u \in BV(\Omega)$, che può essere scomposta in tre termini nel modo seguente:

$$Du = \nabla u dx + D_j u + D_c u.$$

Il primo termine è la parte assolutamente continua del differenziale di u, il secondo è la parte di salto e $D_c u$ è la parte di Cantor. Lo spazio di funzioni $SBV(\Omega)$ è il sottospazio proprio delle funzioni $BV(\Omega)$ tali per cui $D_c u = 0$.

L'equivalenza tra i due problemi:

$$\inf_{u,K} \left\{ \begin{array}{ll} F(u,K), & u \in H^1(\Omega \setminus K) \cap L^\infty(\Omega), \\ & K \subset \Omega, K \text{ chiuso}, \mathcal{H}^{N-1}(K) < \infty \end{array} \right\}$$

$$\inf_u \{G(u), u \in SBV(\Omega) \cap L^\infty(\Omega)\}$$

viene provata da Ambrosio in [3], dimostrando così l'esistenza di un minimo per il funzionale F. Esso può essere calcolato appoggiandosi al seguente

Teorema 1.1

Sia (u, K) una soluzione di (1.1) che soddisfi la Congettura 1, per cui K è composto da un numero finito di curve γ_i. Allora:

$$\begin{cases} \alpha \triangle u = -u_0 \ \ su\ \Omega \setminus K, \\ \dfrac{\partial u}{\partial N} = 0 \ \ su\ \partial\Omega \text{ e sui due lati di ogni } \gamma_i, \\ e(u^+) - e(u^-) + \beta curv(\gamma_i) = 0 \ \ su\ \gamma_i, \end{cases}$$

dove $e(u) = (u - u_0)^2 + \alpha|\nabla u|^2$, u^+ *e* u^- *sono le tracce di* u *su ogni lato di* K, $curv(\gamma_i)$ *è la curvatura di* γ_i, α *e* β *sono le costanti di scala del funzionale* F *e* u_0 *l'immagine iniziale.*

La dimostrazione si può trovare in [6], nella sezione 4.2.

Lo studio della regolarità di K assume un'importanza particolare; infatti, come mostrato in Figura 1.2, le condizioni di regolarità imposte dalla Congettura 1 non sono in grado di garantire l'unicità della soluzione. La congettura di Mumford-Shah non è dunque ancora stata dimostrata; tuttavia A. Bonnet ha fatto degli importanti progressi in questo senso [13].

L'approccio di Mumford-Shah ha una evidente limitazione pratica, infatti l'impossibilità di differenziare il funzionale in uno spazio adatto non permette di applicare le equazioni di Eulero-Lagrange; inoltre la discretizzazione dell'insieme di discontinuità risulta molto complicata. Una soluzione numerica del funzionale di Mumford-Shah non può così essere trovata direttamente; bisogna considerarne una sua approssimazione, quindi applicare una discretizzazione agli elementi finiti o alle differenze finite. Per trovare un funzionale che approssimi $F(u, K)$ (o $G(u)$) ci si può appoggiare alla nozione di Γ-convergenza e creare una serie convergente di funzionali regolari definiti su spazi di Sobolev. Le soluzioni presenti in letteratura che si basano su questo metodo sono diverse, la prima nonché più conosciuta è dovuta ad Ambrosio e Tortorelli [4].

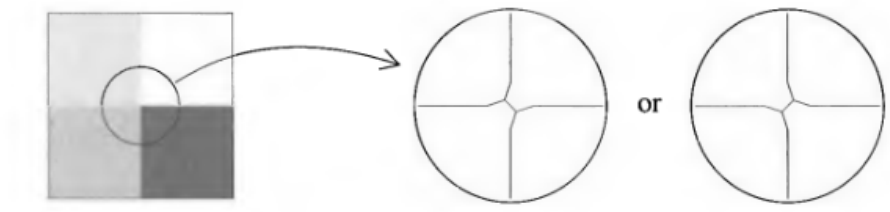


Figura 1.2: Due segmentazioni equivalenti per una data immagine (tratta da [6]).

1.4 Metodi ai contorni attivi

L'obiettivo dei modelli presentati in questa sezione non è la ricerca di una partizione ottima dell'immagine come nel modello di Mumford-Shah, bensì l'individuazione delle sue frontiere. La geometria del contorno viene descritta attraverso una curva che evolve nel tempo e nello spazio, e la segmentazione si ottiene come risultato naturale di questo processo.

I metodi ai contorni attivi possono essere suddivisi in due classi principali, in base al tipo di informazioni che processano per guidare i mo-

vimenti della curva. Quelli basati sui contorni (*edge-based models*) usano informazioni locali sul gradiente di intensità per attrarre la curva verso i confini dell'oggetto da segmentare. Al contrario i metodi basati sulle regioni (*region-based models*) associano ad ogni area dell'immagine una data funzione, basandosi sulle informazioni di colore e intensità.

I primi studi riguardanti i metodi edge-based sono antecedenti alla formulazione del modello di Mumford-Shah; essi vengono introdotti nel paragrafo 1.4.1 insieme alle evoluzioni successive. L'approccio al problema della segmentazione fornito da questi metodi è del tutto innovativo; infatti l'evoluzione della curva K, l'incognita principale, viene descritta con un'equazione alle derivate parziali risolta con un metodo di *level-set*. Ciononostante l'applicazione di tali metodi è limitata a figure semplici con ridotte variazioni di intensità.

I modelli region-based godono di maggiore efficienza in quanto sono meno sensibili al rumore e permettono di processare immagini con contorni poco definiti. Inoltre sono molto robusti rispetto all'inizializzazione della curva. Purtroppo tali modelli si basano sull'assunzione che l'intensità delle immagini sia statisticamente omogenea in ogni regione (*PC : piecewise constant*), ipotesi che limita il loro campo di applicazione. Nonostante la letteratura dell'ultimo decennio sia ricca di metodi di segmentazione region-based, nel paragrafo 1.4.2 si è scelto di dettagliare unicamente l'approccio proposto in [22] da Chan e Vese (in quanto utile per la trattazione successiva). Questi due autori propongono sia un modello PC, sia un secondo modello che permette di superare i limiti del primo grazie ad un approccio multifase. Qui si assume che le regioni siano descritte da funzioni lisce a tratti (*PS: piecewise smooth*); in questo modo diventa possibile segmentare immagini generiche con forti gradienti di intensità, seppure si paghi il prezzo di una bassa efficienza computazionale.

1.4.1 Metodi *edge based*

Snakes

Il primo modello in letteratura che si propone di descrivere il movimento di una curva verso i contorni delle immagini è dovuto a Kass, Witkin e Terzopoulos [47]. Nel 1987 questi autori hanno utilizzato un funzionale energia per controllare l'evoluzione di tale curva, a cui hanno dato il nome di *snake*.
Più nel dettaglio, si consideri lo spazio delle curve di $\mathbb{R}^2$:

$$C = \left\{ c : [a,b] \to \Omega, C^1 \text{ a tratti, t.c. } c(a) = c(b) \right\},$$

dove si vuole minimizzare l'energia data da:

$$J(c) = \int_a^b |c'(q)|^2 dq + \beta \int_a^b |c''(q)|^2 dq + \lambda \int_a^b g^2(|\nabla u_0(c(q))|) dq, \quad (1.3)$$

con β e λ moltiplicatori di Lagrange, e u_0 immagine iniziale. La funzione $g(|\nabla u_0|)$, detta *edge detector function*, è di fondamentale importanza in quanto assume valori vicini allo zero nell'intorno dei contorni. Essa gode delle seguenti proprietà:

- $g : [0, +\infty) \to (0, +\infty)$;
- g è regolare e monotona decrescente;
- $g(0) = 1$, $\lim_{s \to +\infty} g(s) = 0$.

Solitamente si sceglie $g(s) = 1/(1 + s^2)$.

I primi due termini in (1.3) rappresentano l'energia interna e rendono il comportamento della curva assimilabile a quello di una membrana, aumentando la sua regolarità. L'ultimo termine è l'energia esterna, che dipende dall'immagine iniziale u_0, e indica quanto la curva tenda ad avvicinarsi ai bordi (*feature driven energy*).

L'esistenza di un minimo locale nello spazio di Sobolev $[W^{2,2}(a, b)]^2$ si prova facilmente, visto che il dominio è chiuso e limitato. Tuttavia, non essendo il funzionale convesso, non si possono ottenere risultati di unicità. Un altro svantaggio di questo approccio è la dipendenza di J dalla parametrizzazione c della curva: è possibile ottenere soluzioni diverse modificando la parametrizzazione della stessa curva iniziale.
Nella pratica questo problema può essere risolto numericamente incorporandolo in uno schema dinamico tempo-dipendente, assimilando cioè il movimento della curva ad un'evoluzione temporale. Si tratta però di uno schema instabile e con grande sensibilità alle condizioni iniziali.

Una variante dovuta a Cohen ([26]) introduce una forza esterna che induce lo *snake* a comportarsi come un palloncino che viene gonfiato fino ad adattarsi ai contorni dell'immagine. Il cosiddetto *balloon model* riduce l'instabilità del metodo originario. Il modello si ottiene sostituendo a

(a) Contorno iniziale (b) Metodo di Kass et al. (c) Balloon method

Figura 1.3: Confronto tra il modello originale degli *snake* e la modifica di Cohen (immagine tratta da [26]).

$g^2(|\nabla u_0|)$ in (1.3) il termine:

$$F = k_1 N - k\frac{\nabla u_0}{||\nabla u_0||}$$

dove N è il versore normale alla curva, k_1 è un coefficiente che regola l'intensità della forza e k una costante positiva dello stesso ordine di k_1. L'effetto di tale forza F è di esercitare una pressione verso l'esterno, proprio come se si stesse introducendo dell'aria in un palloncino, in modo da evitare i minimi locali. Ad esempio in Figura 1.3, si vede che nel metodo originale di Kass et al. la curva si blocca appena incontra un punto isolato, cosa che non succede invece con l'aggiunta della forza esterna.

Geodesic Active Contours

Come evidenziato dallo studio di Caselles et al. del 1997 ([18]), il secondo termine di (1.3), la cui funzione sarebbe quella di ridurre la curvatura, in realtà è ridondante. Infatti come si vedrà in seguito (1.7) il modello riduce implicitamente la curvatura anche per $\beta = 0$. È quindi naturale sostituire $J(c)$ con il seguente funzionale definito sull'insieme C:

$$J_1(c) = \int_a^b |c'(q)|^2 dq + \lambda \int_a^b g^2(|\nabla u_0(c(q))|)dq. \tag{1.4}$$

Si può inoltre eliminare la scomoda dipendenza dalla parametrizzazione della curva e definire il funzionale intrinseco:

$$J_2(c) = 2\sqrt{\lambda} \int_a^b g(|\nabla u_0(c(q))|)|c'(q)|dq. \tag{1.5}$$

È semplice mostrare che (1.5) è intrinseco. A tale scopo si consideri una nuova parametrizzazione della curva $q = \phi(r), \phi : [a', b'] \to [a, b], \phi' > 0$ e la si sostituisca in (1.5). Indicando $c(\phi(r)) = \bar{c}(r)$ si ottiene

$$\begin{aligned} J_2(c) &= 2\sqrt{\lambda} \int_{a'}^{b'} g(|\nabla u_0(c \circ \phi(r))|)\phi'(r) \cdot |(c \circ \phi)'(r)|(\phi'(r))^{-1} dr \\ &= \sqrt{\lambda} \int_{a'}^{b'} g(|\nabla u_0(\bar{c}(r))|)|\bar{c}'(r)|dr, \end{aligned}$$

ritrovando un'espressione del tutto analoga alla (1.5).

Il funzionale (1.5) non è altro che la definizione di lunghezza pesata di una curva, ottenuta moltiplicando la lunghezza Euclidea per il termine $g(|\nabla u_0(c(q))|)$.

L'equivalenza tra i due funzionali definiti in (1.4) e (1.5) non è però scontata. Si ripropongono nel seguito i passi principali della dimostrazione data da Aubert e Blanc-Féraud in [5]. Si basa sulla seguente nozione di equivalenza tra problemi di minimizzazione, introdotta dagli stessi autori:

Definizione: *Due problemi di minimizzazione $J_1(c)$ e $J_2(c)$ sono equivalenti se esiste un intorno $I(c)$ di c in cui il flusso che riduce maggiormente $J_1(c)$ riduce anche $J_2(c)$ e viceversa.*

Il primo passo per provare l'equivalenza dei due funzionali consiste nel calcolo delle loro derivate rispetto al tempo. Il parametro temporale t viene introdotto per descrivere la famiglia di curve in movimento $c(t,q)$ tali che $c(0,q) = c(q) \in C$; per comodità si pone $J_i(t) = J_i(c(t,q))$, con $i = 1,2$. Integrando per parti le espressioni dei funzionali e effettuando qualche passaggio algebrico si trova:

$$\begin{aligned}\frac{1}{2}J_1'(t) = \int_a^b \left\langle \frac{\partial c}{\partial t}, \left[-\kappa \left|\frac{\partial c}{\partial q}\right|^2 + \langle g\nabla g, N\rangle\right] N \right. \\ \left. + \left[\langle g\nabla g, T\rangle - \left\langle T, \frac{\partial^2 c}{\partial q^2}\right\rangle\right] T \right\rangle dq. \\ \frac{1}{2}J_2'(t) = \int_a^b \left|\frac{\partial c}{\partial q}\right| \left\langle \frac{\partial c}{\partial t}, \langle \nabla g, N\rangle N - \kappa g N \right\rangle dq\end{aligned}$$

dove N è il versore normale alla curva, T quello tangenziale e κ la curvatura. Si ricava facilmente che la direzione lungo la quale J_1 si riduce più rapidamente è data da:

$$\frac{\partial c}{\partial t} = \left(\kappa \left|\frac{\partial c}{\partial q}\right|^2 - \langle g\nabla g, N\rangle\right) N + \left(\left\langle T, \frac{\partial^2 c}{\partial q^2}\right\rangle - \langle g\nabla g, T\rangle\right) T \qquad (1.6)$$

Analogamente, si trova che la direzione lungo la quale una curva con lunghezza pesata data da $J_2(c)$ si contrae più velocemente è:

$$\frac{\partial c}{\partial t} = (\kappa g - \langle \nabla g, N\rangle)N. \qquad (1.7)$$

Sostituendo il flusso per $J_1(t)$ dato da (1.6) nell'espressione di $J_2'(t)$ e studiando il segno dell'integranda così ottenuta, si trova che tale flusso è una direzione di discesa anche per $J_2(t)$. Analogamente, si prova il viceversa e si ha così l'equivalenza dei due funzionali secondo la definizione appena data.

Il problema iniziale è quindi sostituito dalla ricerca di un minimo globale in C per il funzionale $J_2(c)$, più pratico in quanto intrinseco (cioè non dipendente dalla particolare parametrizzazione della curva) e facilmente discretizzabile.

Il modello di Caselles et al. può essere ulteriormente migliorato aggiungendo in (1.7) un termine con coefficiente α col doppio scopo di controllare espansione e restringimento del contorno e migliorare l'individuazione di oggetti concavi:

$$\frac{\partial c}{\partial t} = (\kappa g - \langle \nabla g, N\rangle + \alpha g)N. \qquad (1.8)$$

Così si può scegliere una curvatura κ con segno non costante, che permette di individuare gli oggetti concavi, a patto che $\alpha \geq 0$ sia abbastanza grande perché il segno di $\alpha + \kappa$ non cambi.

La formulazione (1.5)-(1.8) del problema di segmentazione, nota in letteratura come *Geodesic Active Contours* (GAC), è conveniente rispetto a quella originariamente introdotta da Kass et al.. Il suo vantaggio risiede nella possibilità di scrivere le equazioni di Eulero Lagrange associate al funzionale $J_2(c)$ in formulazione Euleriana con un approccio di tipo *level set*, che non potrebbe invece essere usato per trovare un minimo di $J_1(c)$.

La nozione di level set, introdotta da Osher e Sethian in [56], gode di grande successo anche nel campo della segmentazione grazie alla sua immediatezza, che la rende adatta ad identificare l'insieme dei contorni di un'immagine. Questa formulazione è basata sull'idea che una curva può essere vista come il livello zero di una funzione in uno spazio di dimensione maggiore. Cioè l'interfaccia $\partial\Sigma \subset \mathbb{R}^n$ può essere vista come la linea di livello zero di una funzione $\phi : \mathbb{R}^n \to \mathbb{R}$ (si veda la Figura 1.4). Per convenzione si prende $\Sigma = \{x : \phi(x) > 0\}$.

In genere tale approccio viene utilizzato per trovare una soluzione ad equazioni del tipo:

$$\begin{cases} \dfrac{\partial c}{\partial t} = JN, \\ c(0,q) = c_0(q), \end{cases} \tag{1.9}$$

che descrivono l'evoluzione di una curva $c(t,q)$ che si muove lungo la sua componente normale con una velocità J. Si supponga innanzitutto l'esistenza di una funzione $u : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ tale che

$$u(t, c(t,q)) = 0, \quad \forall q, \forall t \geq 0.$$

Ipotizzando che u sia sufficientemente regolare, si può calcolare il suo differenziale rispetto a t e sostituirvi l'espressione della velocità data in (1.9),

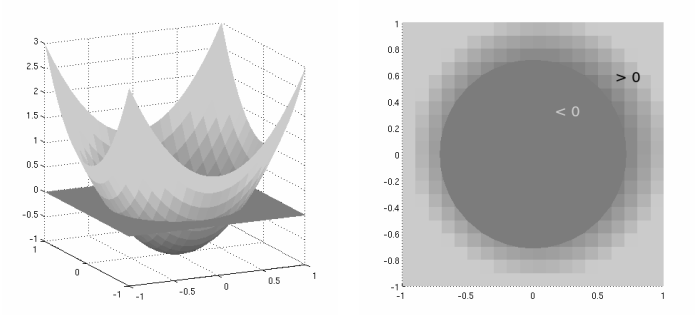


Figura 1.4: Basi del metodo di level set: una circonferenza può essere vista come il livello zero di una funzione di dimensione maggiore.

ottenendo:

$$\frac{\partial u}{\partial t}(t, c(t,q)) + \langle \nabla u(t, c(t,q)), JN \rangle = 0.$$

Si può considerare u come una funzione definita su tutto lo spazio $\mathbb{R}^+ \times \Omega$; quindi tenendo conto dell'espressione del vettore normale N, si riscrive la formulazione di level set (1.9) come:

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = J|\nabla u(t,x)| \ \text{ per } (t,x) \in (0,\infty) \times \Omega \\ u(0,x) = \bar{d}(x, c_0), \\ \dfrac{\partial u}{\partial N} = 0 \ \text{ per } (t,x) \in (0,\infty) \times \partial\Omega, \end{cases} \tag{1.10}$$

dove $c_0(q)$ è il contorno iniziale e $\bar{d}(x, c_0)$ è la funzione distanza dotata di segno:

$$\bar{d}(x, c_0) = \begin{cases} +\, d(x, c_0) & \text{se } x \text{ è esterno a } c_0, \\ -\, d(x, c_0) & \text{se } x \text{ è interno a } c_0. \end{cases}$$

A condizione che J sia ben definita su tutto lo spazio, la curva $c(t,q)$ che si cerca è data dall'insieme di livello zero di u.

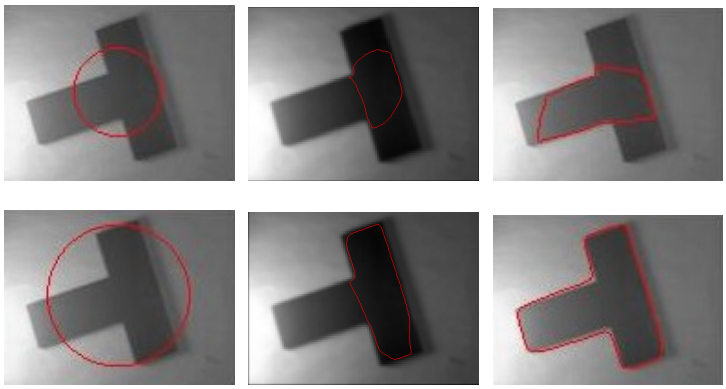
L'equazione (1.10) è chiamata equazione di Hamilton-Jacobi e gode di alcune proprietà molto interessanti. Uno dei vantaggi principali di questa formulazione è la sua versatilità rispetto ai cambi di topologia di $u = 0$. Infatti l'insieme di livello zero può modificarsi, spezzarsi o fondersi, ma tutti questi cambiamenti sono intrinseci nel modello e non devono essere presi in considerazione nell'approssimazione numerica. Inoltre tale approccio si estende facilmente a dimensioni maggiori, e gli elementi geometrici intrinseci come la curvatura possono essere espressi in funzione di u. Queste proprietà spiegano l'ampio utilizzo che si fa dei metodi di level set in svariati campi applicativi.

Ritornando alla soluzione del problema di segmentazione, il flusso in (1.8) è del tipo (1.9). Svolgendo passaggi analoghi a quelli appena visti, si può ottenere la sua formulazione di level set:

$$\frac{\partial u}{\partial t} = g(|\nabla u_0|) \left(\operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) + \alpha \right) |\nabla u| + \langle \nabla g, \nabla u \rangle, \tag{1.11}$$

con le stesse condizioni al bordo e iniziali di (1.10). L'azione del primo termine ferma l'evoluzione della curva quando essa raggiunge i contorni degli oggetti. Il secondo termine invece aumenta l'attrazione della curva verso i suddetti contorni. Esistenza ed unicità della soluzione di questa PDE si dimostrano attraverso la teoria delle sopra- e sotto-soluzioni (si veda [6], sezione 4.3).

Analizzando l'espressione di (1.11) si nota ciò che è già stato accennato all'inizio della sezione: la parte dell'equazione derivante dall'energia interna dipende in maniera importante dal gradiente dell'immagine iniziale.



(d) Contorno iniziale (e) Metodo di KWT (f) Metodo GAC

Figura 1.5: Confronto tra il metodo di Kass-Witkin e Terzopoulos e il Geodesic Active Contour, modificando la curva iniziale (immagine ottenuta con `MATLAB`).

Quindi si tratta di modelli non adatti ad individuare i contorni di oggetti non ben definiti. Inoltre questo modello riesce a riconoscere un oggetto unicamente quando la curva iniziale c_0 circonda i suoi contorni, e non può individuare allo stesso tempo le frontiere interne ed esterne. Si può concludere che questi metodi godono di proprietà di segmentazione unicamente locali, con un'alta sensibilità alle condizioni iniziali. Questo limite è evidenziato nelle immagini di Figura 1.5: in alto il contorno iniziale non circonda l'immagine, e né il metodo originale di Kass, né quello basato sui contorni attivi riescono ad individuare le frontiere dell'oggetto. Invece in basso la scelta della curva iniziale permette al modello GAC di riconoscere i contorni dell'immagine geometrica, anche se non in maniera precisa.

Un ulteriore svantaggio dei metodi GAC si può dedurre dalla particolare condizione iniziale imposta: $u(0, x)$ è una funzione distanza dotata di segno. Però l'equazione (1.11) non assicura che $u(t, x)$ rimanga tale al passare del tempo. Si impone quindi la necessità di reinizializzare regolarmente u risolvendo un'ulteriore PDE, riducendo così drasticamente l'efficienza computazionale.

1.4.2 Metodi *region based*

Questi modelli presentano alcuni vantaggi rispetto a quelli descritti nella sezione precedente. Innanzitutto essi non si limitano a considerare le variazioni del gradiente, ma sfruttano le informazioni di tutta la regione circostante il contorno per controllare l'evoluzione della curva. La quantità di maggior interesse in tale contesto è il valore medio dell'intensità dell'immagine nelle regioni considerate: esso è meno sensibile al rumore locale

e, se sfruttato adeguatamente, permette di segmentare anche immagini con contorni poco definiti. Inoltre i metodi region-based sono poco sensibili alla posizione iniziale della curva e permettono l'individuazione delle frontiere interne.

I modelli di Chan e Vese

Uno dei metodi più popolari di questa classe è stato sviluppato da Tony Chan e Luminita Vese nel 1999 ([22]). Si tratta di una combinazione tra i metodi classici ai contorni attivi e il modello di Mumford-Shah. Al contrario dei primi però, il termine di arresto non dipende dal gradiente di intensità dell'immagine, bensì da una sua particolare segmentazione.

Si consideri, per semplicità, un'immagine come quella in Figura 1.6 composta unicamente da due regioni con intensità costanti a tratti, di valori u_0^i e u_0^o rispettivamente, e si immagini che la frontiera tra queste due aree sia il contorno $\mathcal{C}$ che si vuole ricostruire. Siano c_1 e c_2 due costanti, che rappresentano la media dell'immagine iniziale u_0 all'interno e all'esterno di C, la curva in evoluzione. Nei metodi di level set quest'ultima è il contorno di livello zero di una funzione continua $\phi : \Omega \to \mathbb{R}$, cioè $C = \{(x,y) \in \Omega : \phi(x,y) = 0\}$ ed è positiva all'interno di C e negativa all'esterno.

Seguendo [22] si introduce un'energia, la *fitting energy*, il cui minimo si trova quando la curva coincide con il contorno dell'immagine ($C = \mathcal{C}$), cioè quando i valori calcolati c_i sono vicini a quelli reali:

$$\begin{aligned} F(\phi, c_1, c_2) &= \mu \int_\Omega \delta(\phi)|\nabla\phi| + \nu \int_\Omega H(\phi)dxdy \\ &+ \lambda_1 \int_\Omega |u_0 - c_1|^2 H(\phi)dxdy \\ &+ \lambda_2 \int_\Omega |u_0 - c_2|^2 (1 - H(\phi))dxdy \end{aligned} \tag{1.12}$$

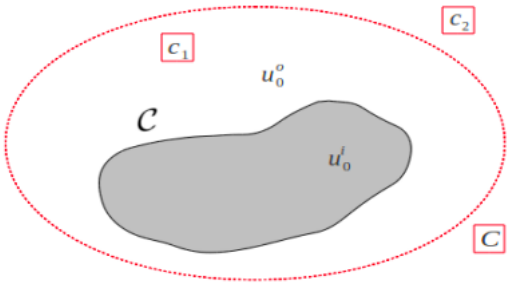


Figura 1.6: Immagine da segmentare: in nero il contorno da ricostruire e in rosso la curva in evoluzione.

dove $H(\cdot)$ è la funzione di Heaviside e $\delta(x) = \frac{d}{dx}H(x)$ la sua derivata nel senso delle distribuzioni; $\mu > 0, \nu \geq 0, \lambda_1 > 0$ e $\lambda_2 > 0$ sono costanti positive. I primi due termini sono di regolarizzazione e rappresentano rispettivamente la lunghezza di C e l'area al suo interno, mentre i restanti due termini indicano in che misura le regioni individuate dalla curva C corrispondono con quelle da segmentare.

L'espressione delle costanti c_1 e c_2 si trova minimizzando la fitting energy per ϕ fissato:

$$\begin{aligned} c_1(\phi) &= \frac{\int_\Omega u_0 H(\phi) dxdy}{\int_\Omega H(\phi(x,y)) dxdy}, \\ c_2(\phi) &= \frac{\int_\Omega u_0 (1 - H(\phi)) dxdy}{\int_\Omega (1 - H(\phi(x,y))) dxdy}. \end{aligned}$$

Si può riconoscere nella prima espressione la media di u_0 all'interno della regione individuata da $\{\phi \geq 0\}$ e nella seconda la sua media in $\{\phi < 0\}$.

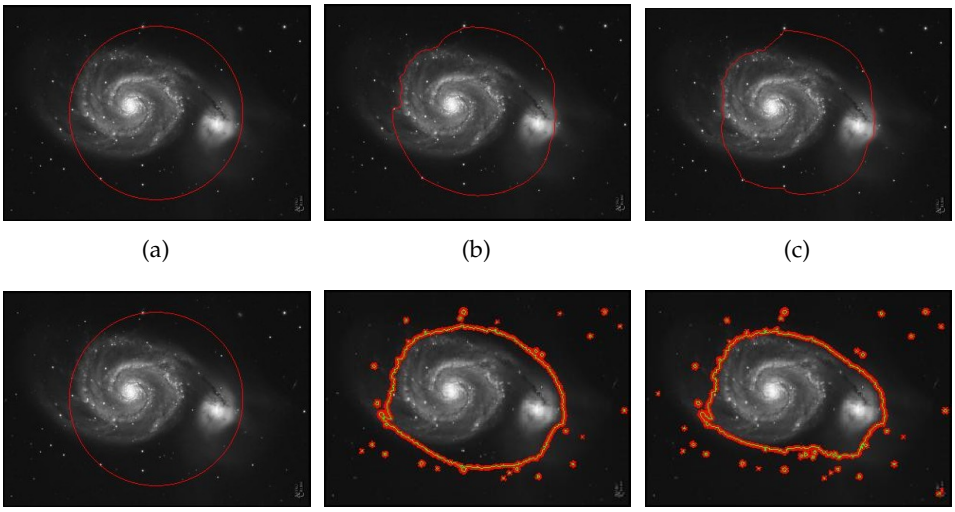
Sostituendo in (1.12) queste espressioni, è ora possibile minimizzare l'energia rispetto a ϕ inserendo un parametro temporale che identifichi la direzione di massima discesa. Si ottiene la seguente equazione di Eulero-Lagrange:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right] & \text{in } \Omega, \\ \frac{\delta(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial n} = 0 \text{ su } \partial\Omega. \end{cases} \tag{1.13}$$

Ai fini di aumentare la regolarità, è necessario utilizzare nella pratica delle approssimazioni C^∞ di H e δ, che indichiamo rispettivamente con H_ϵ e δ_ϵ, tali che $\delta_\epsilon(x) = H'_\epsilon(x)$ nel senso delle distribuzioni. Una scelta standard risulta la seguente:

$$H_\epsilon(x) = \frac{1}{2} \left[1 + \frac{2}{\pi} \arctan \left(\frac{x}{\epsilon} \right) \right], \quad \delta_\epsilon(x) = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + x^2}. \tag{1.14}$$

Introducendo questa approssimazione non solo si semplifica il calcolo della soluzione di (1.13), ma si indirizza anche la ricerca del minimo verso il minimo globale. Infatti, in questo modo, l'equazione (1.13) ha effetto su tutte le curve di livello e non unicamente su quelle di livello zero; si trova così il minimo globale indipendentemente dalla scelta del contorno iniziale. Inoltre i contorni interni vengono individuati automaticamente. In Figura 1.7 si cerca di segmentare un'immagine priva di contorni definiti e si mette a



(a) (b) (c)

(d) Contorno iniziale (e) 500 iterazioni (f) 1000 iterazioni

Figura 1.7: Confronto tra il metodo GAC (in alto) e quello di Chan e Vese (in basso). Immagine ottenuta con MATLAB.

confronto la performance di un metodo *edge-based* con quella di uno schema *region-based*. Si apprezzi la precisione del modello di Chan e Vese che individua addirittura le stelle sparse nella galassia!

Esistono tuttavia alcuni aspetti critici anche di questo metodo. Sono essenzialmente due: innanzitutto, è necessario reinizializzare ϕ perché sia una distanza rispetto alla sua curva di livello zero. Si tratta di una procedura standard per un approccio di tipo level set alla quale si è già accennato alla fine della sezione precedente. Generalmente viene effettuata risolvendo la seguente equazione di evoluzione (si veda [75]):

$$\begin{cases} \dfrac{\partial \psi}{\partial t} = sign(\phi(t))(1 - |\nabla \psi|) \\ \psi(0, \cdot) = \phi(t, \cdot), \end{cases} \tag{1.15}$$

essendo $\phi(t, \cdot)$ la soluzione di (1.13) al tempo t. La nuova $\phi(t)$ è data dalla soluzione stazionaria di (1.15). Nella pratica non è necessario effettuare questa procedura ad ogni passo temporale, contenendo in questo modo l'aumento del costo computazionale associato a questa fase di restarting.
Il secondo aspetto critico di questo metodo è intrinseco nella sua formulazione: il dominio può essere suddiviso unicamente in due regioni. Non è quindi possibile segmentare immagini presentanti topologie complesse o giunzioni triple.

Nonostante questi limiti, i risultati di segmentazione ottenuti con questo schema sono in generale più che soddisfacenti. Nel complesso infatti

tale modello è tra i più diffusi, visto che rappresenta un ottimo compromesso tra qualità dei risultati, semplicità di implementazione e ridotto tempo computazionale.

Come accennato nell'introduzione a questa sezione, Chan e Vese propongono in [23] anche un'estensione multifase del metodo originale. Questo nuovo modello permette di rappresentare sia immagini PC sia PS con un numero ridotto di funzioni di level set. Nel seguito forniamo un'idea generale di questo approccio, rimandando all'articolo originale per maggiori dettagli.

Si considerino $m = \log n$ funzioni di level set ϕ_i (con $i = 1, \cdots, m$), l'unione dei cui insiemi di livello zero rappresenterà i contorni dell'immagine segmentata. Sia inoltre $\Phi = (\phi_1, \ldots, \phi_m)$ e $H(\Phi) = (H(\phi_1), \ldots, H(\phi_m))$ il vettore delle funzioni di Heaviside in cui ogni componente può assumere unicamente i valori 0 o 1. Tale vettore può assumere n valori differenti, quindi si può suddividere il dominio Ω in 2^m *fasi* (o classi) definite nel seguente modo: due pixels (x_1, y_1) e (x_2, y_2) appartengono alla stessa fase se e solo se $H(\Phi(x_1, y_1)) = H(\Phi(x_2, y_2))$. Così ognuna delle n classi si definisce come l'insieme dei punti

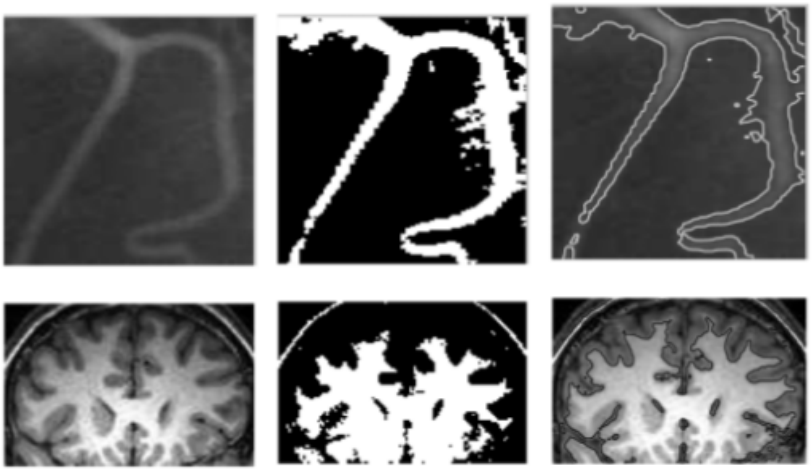
$$\{(x, y) \text{ t.c. } H(\Phi(x, y)) = \mathbf{v} \in H(\Phi(\Omega)), \text{ con } \mathbf{v} \text{ vettore costante}\}.$$

L'insieme delle fasi altro non è che una copertura di Ω, ed ogni pixel appartiene ad una ed una sola fase.
Analogamente a quanto si ha nel modello bifase, si indica con c_I la media dell'immagine iniziale u_0 per i punti appartenenti alla classe I. L'energia da minimizzare è quindi data da:

$$F_n(\Phi, \mathbf{c}) = \sum_{1 \le I \le n = 2^m} \int_\Omega |u_0 - c_I|^2 \chi_I dx dy + \sum_{1 \le i \le m} \mu \int_\Omega |\nabla H(\phi_i)|. \quad (1.16)$$

essendo χ_I la funzione caratteristica della classe I e $\mathbf{c}$ il vettore costante delle intensità medie: $\mathbf{c} = (c_1, \ldots, c_n)$. Si noti che con $n = 2$ e $m = 1$ si ritrova l'energia (1.12) con $\nu = 0$. Le equazioni di Eulero-Lagrange dipendono ovviamente dal numero di fasi considerate, e sono quindi da calcolare caso per caso.

Gli autori propongono anche un'ulteriore estensione del modello al caso di funzioni lisce a tratti, mostrando come in questo caso il numero necessario di funzioni di level set sia ridotto, così come il costo computazionale. La scelta dell'inizializzazione delle classi e la necessità di ricavare le equazioni di Eulero-Lagrange rendono tuttavia questo metodo di scarso utilizzo pratico, contrariamente alla segmentazione binaria proposta in [22]. Quest'ultima gode invece di grande successo ed è alla base di molti degli approcci per la segmentazione comparsi negli ultimi anni, come quello descritto nel prossimo paragrafo.



(a) Originale (b) Thresholding (c) Chan-Vese

Figura 1.8: Errore di segmentazione per due immagini mediche (in alto una vena ai raggi X e in basso la RMN di un cervello). Tratta da [49].

Region Scalable Fitting (RSF)

In [49] viene presentato un nuovo modello region based a contorni attivi che processa informazioni a livello locale attraverso un parametro di scala controllabile tramite un nucleo gaussiano. Per enfatizzare l'innovativa caratteristica di essere scalabile in spazio, il funzionale da minimizzare viene chiamato *region-scalable fitting energy*. Esso viene incorporato in una formulazione variazionale di level set con termine regolarizzante, che ha la funzione di preservare la regolarità della funzione di level set ed evitare costose procedure di reinizializzazione.
Questo metodo è stato scelto come fulcro del presente lavoro, grazie alle sue buone proprietà e alla qualità della segmentazione risultante. Nel prossimo capitolo si studierà nel dettaglio un algoritmo per la sua soluzione numerica e, nel seguito, si analizzeranno i risultati ottenuti applicando tale algoritmo.

Come visto nel paragrafo precedente, il modello originale di Chan e Vese è limitato dall'uso dei valori mediati c_1 e c_2 che non contengono alcuna informazione locale; quindi nel caso di immagini disomogenee il loro valore può essere molto lontano da quello reale. La difficoltà di segmentare immagini di questo tipo è evidenziata dalla Figura 1.8 che mette a confronto un metodo di thresholding e il modello PC di Chan e Vese: in nessuno dei due casi si riescono a catturare correttamente i contorni delle immagini. Il modello multifase di Chan e Vese, nonostante non presenti più queste limitazioni, è troppo complesso e costoso.

L'approccio RSF supera questi problemi, permettendo la segmentazio-

ne di immagini disomogenee e il riconoscimento dei contorni poco definiti. La novità consiste in una serie di convoluzioni locali con una funzione nucleo $K : \mathbb{R}^n \to [0, +\infty)$ che gode delle seguenti proprietà:

1. $K(-\mathbf{u}) = K(\mathbf{u}), \quad \forall \mathbf{u} \in \mathbb{R}^n$;
2. $\int K(\mathbf{x}) d\mathbf{x} = 1$;
3. $K(\mathbf{u}) \geq K(\mathbf{v}) \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ t.c. $|\mathbf{u}| < |\mathbf{v}|$;
4. $\lim_{|\mathbf{u}| \to \infty} K(\mathbf{u}) = 0$.

Si noti che tali proprietà sono soddisfatte da un nucleo gaussiano con un parametro di scala $\sigma > 0$:

$$K_\sigma(\mathbf{u}) = \frac{1}{(2\pi)^{n/2}\sigma^n} e^{-|\mathbf{u}|^2/2\sigma^2}.$$

Data un'immagine $u_0 : \Omega \to \mathbb{R}^d$, sia C un contorno chiuso che separa Ω in due regioni $\Omega_1 = \textit{fuori}(C)$ e $\Omega_2 = \textit{dentro}(C)$. Per un dato punto $\mathbf{x} \in \Omega$ si definisce l'energia *region-scalable fitting*:

$$E_x^{Fit}(C, f_1(\mathbf{x}), f_2(\mathbf{x})) = \sum_{i=1}^{2} \lambda_i \int_{\Omega_i} K_\sigma(\mathbf{x} - \mathbf{y}) |u_0(\mathbf{y}) - f_i(\mathbf{x})|^2 d\mathbf{y}, \tag{1.17}$$

dove $f_i(\mathbf{x})$ sono dei valori che approssimano l'intensità dell'immagine in Ω_i, mentre λ_1 e λ_2 sono costanti positive che agiscono da pesi per gli integrali, rispettivamente all'esterno e all'interno delle regioni definite da C. Nella pratica, se si vuole prevenire la nascita di nuovi contorni all'esterno di quello già esistente, conviene fissare $\lambda_2 > \lambda_1$, penalizzando così l'aumento dell'area della regione interna. Data la natura del nucleo K_σ, i valori di $u_0(\mathbf{y})$ che influenzano maggiormente l'integrale E_x^{Fit} sono quelli interni ad una regione R centrata nel punto $\mathbf{x}$: $R = \{\mathbf{y} : |\mathbf{x} - \mathbf{y}| \leq 3\sigma\}$. La grandezza di tale regione può essere controllata modificando il parametro di scala σ del nucleo gaussiano.
L'energia in (1.17) può anche essere vista come l'errore quadratico medio dell'approssimazione dei valori di intensità all'interno e all'esterno di C, in cui ad ogni $u_0(\mathbf{y})$ è assegnato il peso $K_\sigma(\mathbf{x} - \mathbf{y})$. Questa scalabilità rispetto alla regione considerata è una caratteristica unica e importante di questo metodo.

Per ottenere il contorno intero dell'oggetto bisogna trovare la curva che minimizza l'energia di fitting per tutti i punti nel dominio. Inoltre è necessario aggiungere un termine di penalizzazione proporzionale alla lunghezza del contorno, come accadeva già nel modello di Chan e Vese. Si ottiene quindi il seguente funzionale dell'energia:

$$E(C, f_1(\mathbf{x}), f_2(\mathbf{x})) = \int_\Omega E_{\mathbf{x}}^{Fit}(C, f_1(\mathbf{x}), f_2(\mathbf{x}))) d\mathbf{x} + \nu |C|. \tag{1.18}$$

Per trovare una soluzione al problema di minimizzazione (1.18) e per gestire facilmente i cambi di topologia si introduce, in analogia a quanto già fatto prima, una formulazione di tipo level set. Sia ϕ la funzione di level set, positiva all'esterno del contorno e negativa all'interno. Il funzionale in (1.17) può essere riscritto come:

$$E_x^{Fit}(\phi, f_1(\mathbf{x}), f_2(\mathbf{x})) = \sum_{i=1}^{2} \lambda_i \int_{\Omega_i} K_\sigma(\mathbf{x}-\mathbf{y})|u_0(\mathbf{y}) - f_i(\mathbf{x})|^2 M_i(\phi(\mathbf{y})))d\mathbf{y} \tag{1.19}$$

dove $M_1(\phi) = H(\phi)$ è la funzione di Heaviside e $M_2(\phi) = 1 - H(\phi)$. Si può allora riformulare l'espressione dell'energia in (1.18):

$$\begin{aligned} E(\phi, f_1(\mathbf{x}), f_2(\mathbf{x})) = &\int_{\Omega_i} E_{\mathbf{x}}^{Fit}(\phi, f_1(\mathbf{x}), f_2(\mathbf{x})))d\mathbf{x} \\ &+\nu \int_{\Omega} |\nabla H(\phi(\mathbf{x}))|d\mathbf{x}, \end{aligned} \tag{1.20}$$

dove l'ultimo termine valuta la lunghezza del contorno di livello zero di ϕ.

Anche in questo contesto la funzione di Heaviside si approssima con la funzione regolare definita in (1.14), e le corrispondenti $M_i^\epsilon(\phi)$ vengono sostituite alle funzioni $M_i(\phi)$ in (1.19), ottenendo così una forma regolarizzata dell'energia (1.20) che denotiamo con $E_\epsilon(\phi, f_1(\mathbf{x}), f_2(\mathbf{x}))$.

Per preservare la regolarità della funzione di level set è necessario introdurre un termine di regolarizzazione identificato con:

$$\mathcal{P}(\phi) = \int_{\Omega} \frac{1}{2}(|\nabla\phi(\mathbf{x})| - 1)^2 d\mathbf{x}. \tag{1.21}$$

In questo modo si assicura la stabilità nell'evoluzione del contorno e si evita la reinizializzazione di ϕ, riducendo così il costo computazionale.

Riassumendo, il funzionale che si vuole minimizzare è:

$$\mathcal{F}(\phi, f_1, f_2) = E_\epsilon(\phi, f_1(\mathbf{x}), f_2(\mathbf{x})) + \mu\mathcal{P}(\phi).$$

Innanzitutto bisogna trovare le funzioni $f_i(\mathbf{x})$ che minimizzano $\mathcal{F}(\phi, f_1, f_2)$, fissata ϕ. Si trova così:

$$f_i(\mathbf{x}) = \frac{K_\sigma(\mathbf{x}) * [M_i^\epsilon(\phi(\mathbf{x}))u_0(\mathbf{x})]}{K_\sigma(\mathbf{x}) * M_i^\epsilon(\phi(\mathbf{x}))}, \quad i = 1, 2. \tag{1.22}$$

Tali funzioni sono assimilabili a delle medie pesate dell'intensità in un intorno del punto considerato; la loro regolarità è assicurata dalle buone proprietà della convoluzione e dei nuclei gaussiani.
Ora, fissate f_1 e f_2, si minimizza il funzionale $F(\phi, f_1, f_2)$ rispetto a ϕ, risolvendo l'equazione di flusso:

$$\frac{\partial\phi}{\partial t} = -\delta_\epsilon(\lambda_1 e_1 - \lambda_2 e_2) + \nu\delta_\epsilon \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \mu\left(\triangle\phi - \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right) \tag{1.23}$$

dove δ_ϵ è la delta di Dirac regolarizzata definita in (1.14), e le e_i sono le funzioni:

$$e_i(\mathbf{x}) = \int_{\Omega_i} K_\sigma(\mathbf{y} - \mathbf{x})|u_0(\mathbf{x}) - f_i(\mathbf{y})|^2 d\mathbf{y}, \quad i = 1, 2. \tag{1.24}$$

Il primo termine in (1.23) assume un'importanza fondamentale in quanto attira il contorno verso i confini degli oggetti. Inoltre, visto che δ_ϵ non è mai nulla, esso influenza l'evoluzione della curva in tutto il dominio, rendendo possibile la formazione di nuovi contorni e velocizzando quindi il raggiungimento del risultato finale. Più ϵ è grande, più questo effetto viene amplificato. Il secondo termine invece serve a mantenere limitata la lunghezza del contorno di livello zero, ed allo stesso tempo a lisciarlo. L'ultimo termine è quello di regolarizzazione, senza il quale la funzione ϕ potrebbe crescere in maniera incontrollata attorno al contorno di livello zero. Come conseguenza $\delta_\epsilon(\phi)$ assumerebbe valori molto piccoli e l'evoluzione del contorno verrebbe rallentata in maniera inesorabile .

La scelta del parametro di scala σ del nucleo Gaussiano è fondamentale in quanto influenza la scalabilità in spazio, aspetto caratteristico di questo modello. Inoltre incide sulla robustezza del metodo rispetto all'inizializzazione del contorno. Per σ abbastanza grande l'algoritmo diventa insensibile alla scelta di ϕ_0, come lo sono i modelli PC; infatti il metodo di Chan e Vese può essere considerato come un caso limite di questo approccio per $\sigma \to \infty$. Per molte immagini reali, relativamente omogenee, $\sigma \simeq 10$ è una buona scelta: lascia una discreta libertà nell'inizializzazione del contorno e riduce il numero di iterazioni necessarie alla convergenza (seppur aumenti il costo di ogni convoluzione).

I risultati sperimentali confermano le buone proprietà di segmentazione di questo metodo. In Figura 1.9 è mostrato il processo di segmentazione per

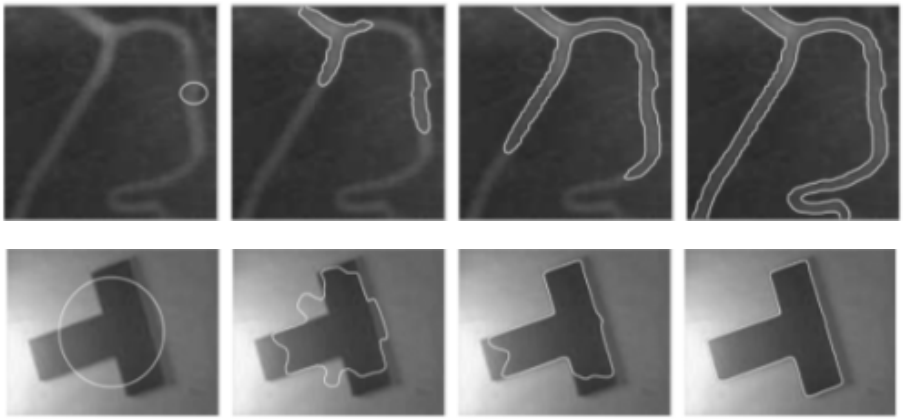


Figura 1.9: Evoluzione della curva nel processo di segmentazione di alcune immagini. Tratta da [49].

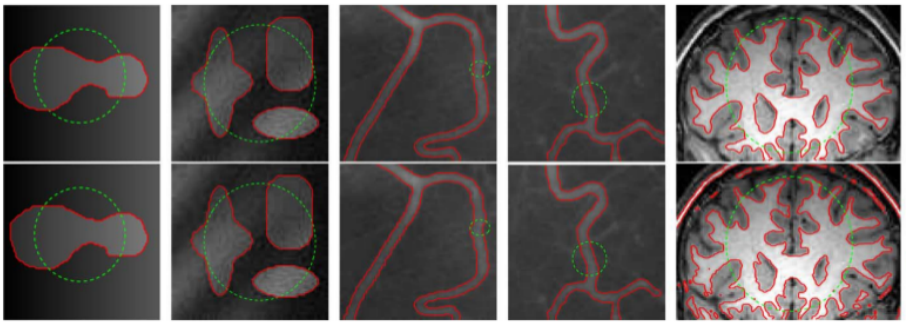


Figura 1.10: Confronto tra il metodo multifase di Chan e Vese (in basso) e il modello RSF (in alto).

due immagine reali con forti disomogeneità. In particolare la prima immagine di una vena ai raggi X è stata usata anche in Figura 1.8 per evidenziare i limiti del modello di Chan-Vese. Limiti superati brillantemente dal metodo RSF, che permette di estrarre dei contorni molto precisi. La segmentazione della seconda immagine è resa complicata dall'illuminazione non uniforme caratterizzante l'immagine iniziale; nonostante ciò il contorno finale è preciso e meglio definito di quello mostrato in Figura 1.5 ottenuto invece con uno schema di tipo active contours.

Un altro confronto significativo effettuato dagli autori in [49] è riportato in Figura 1.10: si paragonano i risultati ottenuti con lo schema RSF e con il modello multifase di Chan-Vese. Il contorno iniziale è disegnato in verde e quello finale in rosso. La segmentazione effettuata dai due metodi è molto simile: infatti anche il modello multifase è capace di gestire le disomogeneità, pur mancando di accuratezza come si può notare nell'ultima immagine. Il vantaggio però più significativo del metodo RSF è il suo ridotto costo computazionale, inferiore di più di un ordine di grandezza rispetto a quello caratterizzante il metodo di Chan-Vese, come si può vedere nella Tabella 1.1.

	Fig.1	Fig.2	Fig.3	Fig.4	Fig.5
Modello RSF	0.84	1.23	8.58	5.78	5.92
Modello CV	30.76	75.51	266.25	142.24	86.07

Tabella 1.1: Confronto dei costi computazionali del modello RSF e del modello di Chan e Vese (CV). Tratta da [49].

Capitolo 2

Ricerca del minimo

Questo capitolo è strutturato in tre blocchi: la Sezione 2.1 contiene una breve introduzione teorica al calcolo delle variazioni, al fine di inquadrare formalmente i problemi di minimizzazione che si stanno analizzando. In seguito, nella Sezione 2.2, si presenta un algoritmo di minimizzazione molto efficace per la soluzione di problemi convessi. Nella Sezione 2.3 tale algoritmo viene applicato ad alcuni dei modelli di segmentazione analizzati nel Capitolo 1, dopo avere sfruttato un'efficiente procedura per trasformare i funzionali concavi in funzionali convessi.

2.1 Introduzione al calcolo delle variazioni

Nel capitolo precedente abbiamo citato risultati del calcolo delle variazioni, senza darne alcuna definizione formale. Per completezza, in questa sezione si colmerà tale mancanza, introducendo i concetti principali necessari a comprendere le scelte successive. Per una trattazione più ampia e completa si può consultare [50].

Sia X uno spazio di Banach e $F : X \to \mathbb{R}$ un funzionale. Per un'introduzione sugli spazi di Banach, sul concetto di funzionale e sulle convergenze ivi definite si può consultare [64]. Si vuole risolvere un generico problema di minimizzazione nello spazio X:

$$\text{trovare } x^* \in X \text{ t.c. } F(x^*) = \inf_{x \in X} F(x). \tag{2.1}$$

Prima di descrivere il risultato principale del calcolo delle variazioni inerente a tale problema, è necessario introdurre alcune definizioni.

Definizione: *Si dice che F è **inferiormente semicontinuo** in x_0 se*

$$F(x_0) \leq \liminf_{n \to \infty} F(x_n)$$

per ogni successione $\{x_n\}$ convergente a x in X. In particolare, F è detto inferiormente semicontinuo nello spazio X se tale relazione vale $\forall x \in X$.

La nozione di semicontinuità inferiore varia in relazione al tipo di convergenza a cui si fa riferimento: se si usa la convergenza forte (rispettivamente debole) in X, si parla di semicontinuità inferiore forte (debole). La semicontinuità inferiore debole implica quella forte, ma non vale il viceversa.

Definizione: *Un funzionale $F : X \to \mathbb{R}$ è **coercivo** in X se, $\forall x \in \mathbb{R}$, la chiusura dell'insieme $\{F \leq x\}$ è sequenzialmente compatta in X, ossia se ogni successione contenuta in tale insieme ammette punto di accumulazione.*

Definizione: *Una successione $\{x_n\} \in X$ si dice **minimizzante** per F in X se*

$$\inf_{y \in X} F(y) = \lim_{n \to \infty} F(x_n).$$

Ogni funzionale F ammette almeno una successione minimizzante.

Siamo ora in grado di enunciare il **metodo diretto** nel calcolo delle variazioni, introdotto da Tonelli all'inizio del secolo scorso ([70]).

Teorema 2.1

Sia $F : X \to \mathbb{R}$ un funzionale coercivo e semincontinuo inferiormente. Allora valgono le seguenti affermazioni:

(i) F ha un punto di minimo in X:

$$\exists x \in X \text{ t.c. } F(x) = \inf_{y \in X} F(y).$$

(ii) Se $\{x_n\}$ è una successione minimizzante per F in X per cui vale $\lim_{n \to +\infty} x_n = x$, allora x è un punto di minimo per F in X.

(iii) Se F non è identicamente $+\infty$, allora ogni sua successione minimizzante ammette una sottosuccessione convergente.

Questo teorema assicura l'esistenza di una soluzione per il problema (2.1), ma non la sua unicità. Per avere unicità del punto di minimo è necessario introdurre il concetto di convessità di un funzionale.

Definizione: *Il funzionale F si dice **convesso** se, $\forall x, y \in X$, vale*

$$F(\alpha x + (1-\alpha)y) \leq \alpha F(x) + (1-\alpha)F(y), \quad \forall \alpha \in [0,1].$$

*Inoltre F si dice **strettamente convesso** se, $\forall x, y \in X$ t.c. $x \neq y$, vale*

$$F(\alpha x + (1-\alpha)y) < \alpha F(x) + (1-\alpha)F(y), \quad \forall \alpha \in (0,1).$$

Si può ora enunciare il risultato di unicità per il problema di minimizzazione (2.1).

Teorema 2.2

Sia $F : X \to \mathbb{R}$ un funzionale strettamente convesso. Allora F ammette al più un punto di minimo in X.

Una volta definito il contesto in cui la ricerca di un minimo globale di (2.1) è giustificata, rimane da affrontare proprio il problema della minimizzazione. La letteratura matematica è ricca di algoritmi sviluppati allo scopo di individuare il minimo globale di un problema di minimizzazione, solitamente sfruttando una procedura iterativa. Nella prossima sezione si introduce una tecnica di recente applicazione all'ambito del trattamento delle immaini, particolarmente adatta ad essere applicata a problemi di segmentazione come quelli a cui si è fatto riferimento nel primo capitolo.

2.2 Algoritmo iterativo di minimizzazione

La procedura iterativa descritta nel seguito fu introdotta da Bregman nel 1967 al fine di identificare i minimi di funzionali convessi ([15]), ed è stata utilizzata in problemi di elaborazione dei segnali.

Tale algoritmo è stato recentemente applicato all'ambito del *denoising* delle immagini da Osher e altri in [55] per risolvere il modello di Rudin, Osher e Fatemi (ROF nel seguito), che consiste nella minimizzazione di

$$J(u) = \min_{u \in BV} \{ |u|_{BV} + \lambda \|u - u_0\|^2_{L^2} \}, \tag{2.2}$$

dove λ è un moltiplicatore di Lagrange e u_0 è l'immagine iniziale. La seminorma BV è stata definita in (1.2).

Il funzionale (2.2) si è rivelato particolarmente difficile da minimizzare con i metodi convenzionali a causa dell'accoppiamento tra termini in norma BV e L^2. Perciò la letteratura è ricca di metodi alternativi per la sua minimizzazione, tra cui l'algoritmo che ci apprestiamo a descrivere.

In particolare siamo interessati ad una variante del metodo di Bregman introdotta nel 2009 da Goldstein e Osher, adatta a risolvere in modo efficiente i problemi di ottimizzazione presentanti un termine di regolarizzazione L^1 ([43]).

2.2.1 Metodo di Bregman

L'algoritmo di Bregman è una procedura di regolarizzazione iterativa basata sulla definizione di *distanza di Bregman* tra due punti $u, v \in X$ associata ad un funzionale convesso $E(\cdot) : X \to \mathbb{R}$:

$$D^p_E(u, v) = E(u) - E(v) - < p, u - v >$$

dove p è il sottogradiente di E in $v \in X$:

$$p \in \partial E(v) \quad \text{se} \quad E(v^*) \geq E(v) + \langle p, v^* - v \rangle, \ \forall v^* \in X.$$

Non si tratta di una vera e propria distanza (non è simmetrica e non soddisfa la disuguaglianza triangolare!), ma vale la relazione $D_E^p(u,v) \geq 0$ e, in particolare, $D_E^p(u,v) = 0$ se $u = v$.

Ora, si considerino due funzionali convessi E e K, con $\min_{u \in \mathbb{R}^n} K(u) = 0$, e il seguente problema libero di minimizzazione

$$J(u) = \min_u \{E(u) + \lambda K(u)\} \tag{2.3}$$

dove $\lambda > 0$ è un fattore di penalizzazione. Questo problema può essere risolto in maniera iterativa costruendo le due successioni seguenti:

$$\begin{aligned} &\text{Dato } u^0 = 0 \text{ e } b^0 = 0, \text{ per } k = 0, 1, \ldots \\ &\quad u^{k+1} = \min_u \{D_E^p(u, u^k) + \lambda K(u)\} \\ &\qquad\quad = \min_u \{E(u) - < p^k, u - u^k > + \lambda K(u)\} \\ &\quad p^{k+1} = p^k - \nabla K(u^{k+1}). \end{aligned}$$

La convergenza di questa procedura è studiata in [55]; si può dimostrare il seguente

Teorema 2.3

Siano E e K dei funzionali convessi, e sia K differenziabile. Se esiste una soluzione u^* del problema (2.3), allora la successione $K(u^k)$:

- *è monotona decrescente: $K(u^{k+1}) \leq K(u^k)$;*
- *converge se $E(u^*) < \infty$:*
  $K(u^k) \leq K(u^*) + E(u^*)/k$.

La rapida convergenza delle iterazioni di Bregman rappresenta uno dei maggiori vantaggi di questo approccio alla minimizzazione; in particolare gode di una convergenza molto rapida quando è applicato a problemi contenenti un termine di regolarizzazione L^1. La spiegazione di questo fatto si può trovare in [43].

È interessante notare, seguendo ([73]), che se si considera il problema di minimizzazione vincolata

$$\min_u \{E(u) : Au = b\} \quad \text{con } A \text{ operatore lineare,} \tag{2.4}$$

riscritto sotto forma di minimizzazione non vincolata con termine di penalizzazione

$$\min_u \left\{ E(u) + \frac{\lambda}{2} ||Au - b||_2^2 \right\}.$$

l'algoritmo di Bregman si può reinterpretare nella forma

$$\begin{aligned}
&\text{Dato } u^0 = 0 \text{ e } b^0 = 0, \text{ per } k = 0, 1, \dots \\
&u^{k+1} = \min_u \left\{ E(u) + \frac{\lambda}{2} ||Au - b^k||_2^2 \right\} \qquad (2.5) \\
&b^{k+1} = b^k + b - Au^{k+1}. \qquad (2.6)
\end{aligned}$$

Grazie ai risultati di convergenza del Teorema 2.3 si ha che $\lim_{k\to\infty} Au^k = b$ e la soluzione u^* è anche soluzione del problema vincolato originale (2.4).

Un ulteriore vantaggio di questo approccio deriva dal fatto che il valore di λ rimane costante, contrariamente a quanto avviene nei metodi di continuazione. Essi infatti consistono nel risolvere il problema (2.4) con una serie di fattori di penalizzazione dal valore crescente $\lambda_1 < \lambda_2 < \cdots < \lambda_n$. Usando un approccio di Bregman è possibile scegliere il valore del parametro di scala che sia ottimale ai fini di una rapida convergenza; inoltre per $\lambda \to \infty$ non si provoca instabilità come accade nei metodi di continuazione.

2.2.2 Metodo di Split Bregman

Come anticipato, questa variante del metodo di Bregman è adatta a risolvere i problemi di ottimizzazione aventi un termine L^1, la cui forma generale è dunque:

$$\min_u \{ \|E(u)\|_1 + K(u) \} \qquad (2.7)$$

dove con $\|\cdot\|_1$ si indica la norma L^1. Si assume inoltre che $K(\cdot)$ e $E(\cdot)$ siano funzionali convessi, e che inoltre $E(\cdot)$ sia differenziabile.

Se $K(\cdot)$ è un termine L^2, la difficoltà risiede nella presenza di *coupling* tra la porzione L^1 e L^2. Il punto di partenza di questo metodo è quindi il disaccoppiamento dei due termini mediante l'introduzione di una quantità ausiliaria. Si risolve:

$$\min_{u,d} \{ \|d\|_1 + K(u) \} \quad \text{t.c.} \quad d = E(u), \qquad (2.8)$$

il che equivale alla seguente minimizzazione

$$\min_{u,d} \left\{ \|d\|_1 + K(u) + \frac{\lambda}{2} \|d - E(u)\|_2^2 \right\}.$$

Per imporre il vincolo si può usare un'iterazione di Bregman

$$\begin{aligned}
(u^{k+1}, d^{k+1}) &= \min_{u,d} \left\{ D_E^p(u, u^k, d, d^k) + \frac{\lambda}{2} \|d - E(u)\|_2^2 \right\} \\
&= \min_{u,d} \left\{ \Phi(u, d) - \langle p_u^k, u - u^k \rangle - \langle p_d^k, d - d^k \rangle + \frac{\lambda}{2} \|d - E(u)\|_2^2 \right\} \\
p_u^{k+1} &= p_u^k - \lambda (\nabla E)^T (E(u^{k+1}) - d^{k+1}) \\
p_d^{k+1} &= p_d^k - \lambda (d^{k+1} - E(u^{k+1})),
\end{aligned}$$

dove si è posto $\Phi(u,d) = \|d\|_1 + K(d)$.

Seguendo la versione semplificata dell'algoritmo, analogamente a (2.5) -(2.6), si ottiene un'iterazione dello schema di *Split Bregman*:

$$(u^{k+1}, d^{k+1}) = \min_{u,d} \left\{ \|d\|_1 + K(u) + \frac{\lambda}{2}\|d - E(u) - b^k\|_2^2 \right\} \quad (2.9)$$

$$b^{k+1} = b^k + (E(u^{k+1}) - d^{k+1}). \quad (2.10)$$

In questo modo si è ridotto il problema (2.7) ad una sequenza di problemi di ottimizzazione non vincolata con l'aggiunta di b^k, che chiameremo *aggiornamento di Bregman*. Si tratta di un algoritmo molto efficiente: infatti le componenti L^1 e L^2 del funzionale sono ora separate, e si può risolvere il problema (2.9) minimizzando separatamente rispetto a u e d nel seguente modo:

$$\text{Step 1:} \qquad u^{k+1} = \min_u \left\{ K(u) + \frac{\lambda}{2}\|d^k - E(u) - b^k\|_2^2 \right\} \quad (2.11)$$

$$\text{Step 2:} \qquad d^{k+1} = \min_d \left\{ \|d\|_1 + \frac{\lambda}{2}\|d - E(u^{k+1}) - b^k\|_2^2 \right\}. \quad (2.12)$$

Il primo step può essere risolto con un'ampia gamma di tecniche, tra cui le più usate sono Gauss-Seidel o alcuni passi del gradiente coniugato. La soluzione del secondo step invece può essere trovata esplicitamente usando i cosiddetti operatori di *shrinkage*:

$$d_j^{k+1} = shrink(E(u)_j + b_j^k, 1/\lambda)$$

dove

$$shrink(x,\gamma) = \frac{x}{|x|} \cdot \max(|x| - \gamma, 0).$$

Lo shrinkage (detto anche *thresholding*) è una semplice tecnica spesso usata per il denoising di segnali e immagini. È stata introdotta da Donoho nel 1994 ([30]) in una duplice forma: operatori di *soft* e *hard* thresholding. I primi hanno attirato maggiormente l'attenzione della comunità scientifica, grazie alla loro efficienza.

L'idea alla base degli algoritmi di shrinkage è piuttosto semplice. Si consideri per semplicità il problema di minimizzazione della funzione scalare $g(x)$ al variare di x_0:

$$g(x, x_0) = \frac{1}{2}(x - x_0)^2 + \lambda|x|^p.$$

Il sottogradiente della funzione g calcolato nel punto di minimo x^* deve essere tale che $0 \in \partial g(x^*, x_0)$. Quindi bisogna risolvere l'equazione:

$$0 \in x - x_0 + \lambda p \cdot \begin{cases} (x)^{p-1} & x > 0 \\ 0 & x = 0 \\ -(-x)^{p-1} & x < 0 \end{cases}$$

La soluzione di tale problema porta all'espressione della funzione $x^* = shrink(x_0, \lambda)$ che trova il minimo globale di $g(x)$. Si noti che tale funzione definisce un unico x^* per ogni valore di x_0, ma non viceversa; diversi valori di x_0 possono essere associati alla stessa soluzione. L'operatore di shrinkage mappa i valori vicini all'origine allo 0, e quelli esterni vengono avvicinati (compressi, shrinked!) all'origine di un fattore λ. Si possono consultare [21, 34] per un ulteriore approfondimento.
Inserendo lo schema iterativo (2.11)-(2.12) all'interno dell'iterazione di Split Bregman definita da (2.9)-(2.10) si ottiene il seguente:

1 ALGORITMO DI SPLIT BREGMAN GENERALIZZATO

while $\|u^k - u^{k-1}\|_2 > tol$ **do**
  for $n = 1 \to N$ **do**
    $$u^{k+1} = \min_u \left\{ K(u) + \frac{\lambda}{2} \|d^k - E(u) - b^k\|_2^2 \right\}$$
    $$d_j^{k+1} = shrink\left(E(u)_j + b_j^k, \tfrac{1}{\lambda} \right)$$
  end for
  $b^{k+1} = b^k + (E(u^{k+1}) - d^{k+1})$
end while

In realtà non è conveniente risolvere il sottoproblema inserito nel ciclo **for** fino a convergenza. Infatti l'eccesso di precisione rischia di andare sprecato quando si calcola il valore del parametro di Bregman b^k al passo successivo.

Empiricamente si trova che nella maggior parte dei casi conviene effettuare un unico aggiornamento delle incognite (u, d). Infatti l'algoritmo converge anche se viene risolto in maniera approssimata rispetto alla variabile u^{k+1}. La robustezza del metodo di Split Bregman è dovuta al fatto che tutti i suoi punti fissi sono minimi del problema originale. Sia (u^*, b^*) punto fisso di (2.9)-(2.10), allora vale $b^* = b^* + E(u^*) - d^*$ e quindi $d^* = E(u^*)$. Per i risultati di convergenza già citati, la coppia (u^*, b^*) soddisfa il problema vincolato (2.8).

2.3 Applicazione ai problemi di segmentazione

2.3.1 *Globally Convex Segmentation*

I risultati introdotti nella Sezione 2.1 permettono di analizzare con maggiore consapevolezza i modelli di segmentazione presentati nel Capitolo 1. Si è intenzionalmente lasciato molto spazio alla descrizione dei limiti di ognuno di essi, in modo da sottolineare la complessità del problema della segmentazione. Tali limiti derivano proprio dalla mancata convessità dei funzionali considerati. In particolare, nel modello di Kass et al. l'energia

(1.3) raggiunge il suo minimo globale quando la curva c si contrae fino a diventare un singolo punto. Quindi tutte le soluzioni non degeneri di questo problema sono solamente minimi locali. Anche il modello di Chan e Vese (1.12) effettua una minimizzazione su un insieme non convesso. Infatti tale problema di ottimizzazione può essere interpretato come la ricerca della migliore approssimazione dell'immagine data tra tutte le funzioni che possono assumere solamente due valori; questo insieme di funzioni non è convesso.

I risultati di entrambi questi modelli sono molto sensibili all'inizializzazione della curva, e spesso il processo di minimizzazione si ferma quando trova un minimo locale. Per risolvere i problemi derivanti dalla non convessità dei funzionali energia, sono stati proposti diversi modelli convessi di segmentazione. Ne introduciamo uno nel seguito.

Chan et al. propongono in [54] una nuova formulazione del modello di Chan e Vese basato sull'uso di funzionali convessi. Tale approccio viene chiamato *globally convex segmentation* (nel seguito GCS) ed è molto affidabile oltre a poter essere integrato con metodi iterativi per trovare rapidamente la soluzione dei problemi di minimizzazione.

Nell'algoritmo originale di Chan e Vese si approssima la funzione di Heaviside con una funzione regolare $H_\epsilon(x)$ il cui supporto non è compatto. Di conseguenza, la soluzione stazionaria di (1.13) coincide con quella di

$$\frac{\partial \phi}{\partial t} = \mu \text{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2 \tag{2.13}$$

dove si è semplicemente omessa la funzione di Heaviside. Tale flusso corrisponde all'equazione di Eulero-Lagrange associata alla seguente energia (per semplicità di scrittura si considera $\nu = 0$):

$$E(\phi) = \int_\Omega |\nabla \phi| + \int_\Omega \phi \left(\lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\right) dx. \tag{2.14}$$

Indicando con $r = \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2$, l'energia in (2.14) si riscrive come segue:

$$E(\phi) = \|\nabla \phi\|_1 + \langle \phi, r \rangle.$$

Il passaggio dalla formulazione originale dell'energia (1.12) a quella in (2.14) è proprio ciò che permette di ottenere un problema convesso, come auspicato. Infatti la funzione $H(\phi)$ è una parametrizzazione delle funzioni binarie, il cui insieme non è convesso. Eliminare la funzione di Heaviside corrisponde a rimuovere il vincolo di non convessità: la minimizzazione viene effettuata su un insieme di funzioni che possono assumere qualsiasi valore intermedio!

Tuttavia il funzionale $E(\phi)$ in (2.14) è omogeneo di primo grado in ϕ, quindi non ammette un minimo unico. Analogamente il flusso (2.13) non

ha uno stato stazionario: con l'avanzare del tempo la funzione di level set ϕ tende a $+\infty$ laddove è positiva, e a $-\infty$ dove è negativa. Tale effetto è intrinseco nella rappresentazione di level set, alla cui non unicità si è già accennato in precedenza. Per risolvere questo problema è sufficiente limitare la funzione ϕ in un intervallo ristretto, chiedendo, ad esempio $0 \leq \phi(x) \leq 1,\ \forall x \in \Omega$. In [54] si dimostra il seguente risultato:

Teorema 2.4

Siano $c_1, c_2 \in \mathbb{R}$ fissati, e sia ϕ^* una qualsiasi soluzione del problema convesso

$$\min_{0 \leq \phi \leq 1} \{ \|\nabla \phi\|_1 + \langle \phi, r \rangle \}. \tag{2.15}$$

Allora per q.o. $\mu \in [0,1]$ la funzione caratteristica

$$\chi_{\Omega_C}(\mu) = \{x : \phi^*(x) \geq \mu\}$$

è un minimo globale di (1.12).

La regione Ω_C è l'area della curva in evoluzione C, indipendente dal valore di μ. Questo teorema afferma che i minimizzanti così ottenuti soddisfano anche il vincolo più stretto rappresentato dalla presenza della funzione di Heaviside, e che il risultato della segmentazione sarà una funzione binaria.

L'indipendenza del metodo GCS dall'inizializzazione dei contorni si può apprezzare in Figura 2.1 dove si cerca ancora una volta di segmentare la risonanza magnetica di un cervello. In basso a sinistra si ha il risultato del processo di segmentazione usando l'algoritmo GCS, che in sole 6 iterazioni raggiunge la convergenza con entrambi i dati iniziali mostrati in alto. La qualità della segmentazione risultante è inoltre elevata. Viceversa, quando si sceglie una curva iniziale esterna all'immagine, il metodo di Chan e Vese rimane bloccato su un minimo locale e la segmentazione, pur svolgendo quasi 500 iterazioni, non riesce ad individuare i dettagli. Ciò non avviene quando il contorno iniziale è interno all'immagine, come si può vedere nella figura a destra.

Tale algoritmo convesso di segmentazione viene ulteriormente perfezionato da Bresson et al. in [16]. Gli autori incorporano nell'espressione dell'energia (2.15) un termine che agisce da edge-detector. Innanzitutto definiamo la norma *Total Variation* (TV) di una funzione $u : \Omega \to \mathbb{R}$ come segue:

$$TV(u) = \int_\Omega |\nabla u(x)| dx.$$

Si può inoltre definire una norma TV pesata con una funzione $g(x)$:

$$TV_g(u) = \int_\Omega g(x) |\nabla u(x)| dx = |\nabla u|_g.$$

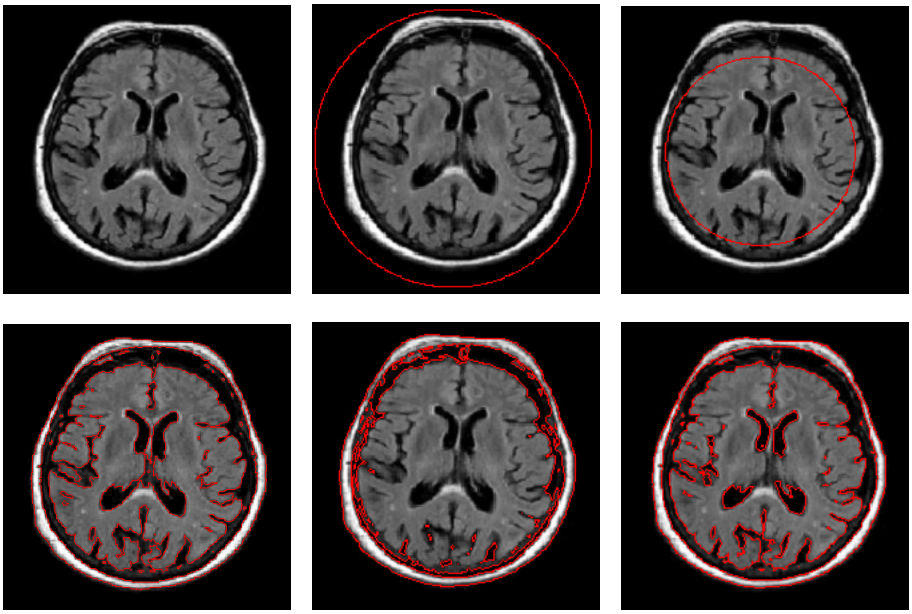


Figura 2.1: In alto l'immagine iniziale e due diverse inizializzazioni del contorno. In basso il risultato della segmentazione con GCS (sinistra) e con Chan-Vese (mezzo e destra) usando i contorni iniziali sovrastanti. Immagine ottenuta con `MATLAB`.

Sostituendo in (2.15) questa norma al posto di quella $L^1(\Omega)$, si ottiene un'espressione dell'energia molto simile a quella originariamente definita da Kass et al. nel loro modello GAC, data da:

$$\min_{0\leq\phi\leq 1} \left\{ |\nabla\phi|_g + \langle \phi, r \rangle \right\}. \tag{2.16}$$

Se g è una funzione edge-detector questa modifica favorisce la segmentazione ad un livello più dettagliato.

Il risultato più importante ottenuto con la convessificazione del funzionale energia consiste nella possibilità di applicare algoritmi rapidi per trovare una soluzione al problema di minimizzazione; a questo proposito citiamo tre diversi approcci.

- In [54] Chan et al. propongono di imporre il vincolo di disuguaglianza usando una funzione di penalizzazione esatta, e di risolvere il problema:

  $$\min_{\phi} \left\{ \|\nabla\phi\|_1 + \langle \phi, r \rangle + \langle \alpha, \nu(\phi) \rangle \right\}$$

  dove $\nu(\xi) = \max\{0, 2|\xi - 0.5| - 1\}$. Gli autori dimostrano che la soluzione di tale problema è equivalente a quella di (2.15) per valori sufficientemente grandi di α.

- Invece Bresson et al. in [16] usano uno splitting della norma TV e, per valori molto piccoli di θ, risolvono, prima per u e poi per v, il problema:

$$\min_{\phi,v}\left\{TV_g(\phi)+\langle\phi,r\rangle+\langle\alpha,\nu(\phi)\rangle+\frac{1}{2\theta}\|u-v\|_2^2\right\}.$$

- Infine in [42] Goldstein, Bresson e Osher propongono l'utilizzo del metodo di *Split-Bregman*, come vedremo nel prossimo paragrafo.

2.3.2 Applicazione a Chan-Vese

Si vuole minimizzare il funzionale in (2.16). Quindi, seguendo le notazioni introdotte nel paragrafo 2.2.2: $\|E(\phi)\| = |\nabla\phi|_g$ e $K(\phi) = \mu\langle\phi, r\rangle$. Per risolvere il problema si introduce la variabile ausiliaria $\vec{d} = \nabla\phi$, e il vincolo di uguaglianza viene imposto tramite un moltiplicatore di Lagrange λ:

$$(\phi^*,\vec{d}^*) = \min_{0\le\phi\le1,\vec{d}}\left\{|\vec{d}|_g+\mu\langle\phi,r\rangle+\frac{\lambda}{2}\|\vec{d}-\nabla\phi\|_2^2\right\}.$$

Analogamente a (2.10)-(2.9) si separano i due termini da ottimizzare ottendo la seguente sequenza di ottimizzazione:

$$(\phi^{k+1},\vec{d}^{k+1}) = \min_{0\le\phi\le1,\vec{d}}\left\{|\vec{d}|_g+\mu\langle\phi,r\rangle+\frac{\lambda}{2}\|\vec{d}-\nabla\phi-\vec{b}^k\|_2^2\right\} \quad (2.17)$$

$$\vec{b}^{k+1} = \vec{b}^k+\nabla\phi^{k+1}-\vec{d}^{k+1}. \quad (2.18)$$

Come spiegato (2.17)-(2.18) si risolve prima ottimizzando rispetto a ϕ e poi rispetto a $\vec{d}$, in entrambi i casi mantenendo fissa l'altra variabile.

L'equazione di Eulero-Lagrange legata a (2.17), per $\vec{d}$ fisso è:

$$\triangle\phi = \frac{\mu}{\lambda}r+\operatorname{div}(\vec{d}-\vec{b}),\ \text{con } 0\le\phi\le1. \quad (2.19)$$

Si consideri il problema della minimizzazione di (2.17) per $\phi^{i,j}$, mantenendo tutti gli altri elementi di ϕ costanti. L'energia in considerazione è quadratica in $\phi^{i,j}$, quindi il suo minimo si trova risolvendo l'equazione (2.19) per $\phi^{i,j}$. Se la soluzione di tale equazione si trova nell'intervallo $[0,1]$, allora questo minimo globale corrisponde col minimo del problema vincolato. In caso contrario si deduce che l'energia, essendo quadratica, è strettamente monotona nell'intervallo $[0,1]$ e che il minimo del problema vincolato si trova nell'estremo più vicino alla soluzione trovata.

Goldstein et al. propongono di discretizzare questa equazione con il metodo delle differenze finite, usando lo stencil a 5 punti per il laplaciano

e le differenze finite all'indietro per l'operatore di divergenza:

$$\begin{aligned}
\alpha_{i,j} &= d^x_{i-1,j} - d^x_{i,j} + d^y_{i,j-1} - d^y_{i,j} - (b^x_{i-1,j} - b^x_{i,j} + b^y_{i,j-1} - b^y_{i,j}) \\
\beta_{i,j} &= \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} - \frac{\mu}{\lambda} r + \alpha_{i,j}) \\
\phi_{i,j} &= \max\{\min\{\beta_{i,j}, 1\}, 0\}.
\end{aligned} \tag{2.20}$$

Si noti che tale discretizzazione può essere assimilata ad un'iterazione del metodo di Gauss-Seidel, ragion per cui nel seguito verrà indicata con la notazione $GS(r^k, \vec{d}^k, \vec{b}^k)$.

Poi, per ϕ fisso, la minimizzazione di (2.17)-(2.18) rispetto a $\vec{d}$ è data da:

$$\vec{d}^{k+1} = shrink\left(\vec{b}^k + \nabla\phi^{k+1}, g\lambda\right).$$

Abbiamo così ottenuto uno schema per risolvere facilmente il problema della segmentazione utilizzando l'algoritmo di Chan e Vese.

2 CHAN-VESE: GCS E SPLIT BREGMAN

while $\|\phi^{k+1} - \phi^k\|_2 > tol$ **do**
- Aggiorna $r^k = (c_1^k - u_0)^2 - (c_2^k - u_0)^2$
- $\phi^{k+1} = GS(r^k, \vec{d}^k, \vec{b}^k)$
- $\vec{d}^{k+1} = shrink(\nabla\phi^{k+1} + \vec{b}^k, g\lambda)$
- $\vec{b}^{k+1} = \vec{b}^k + \nabla\phi^{k+1} - \vec{d}^{k+1}$
- Trova $\Omega^{k+1} = \{x : \phi^{k+1}(x) > \mu\}$
- Aggiorna i valori di $c_1^{k+1} = \int_{\Omega^{k+1}} u_0 dx$ e $c_2^{k+1} = \int_{(\Omega^{k+1})^c} u_0 dx$

end while

2.3.3 Applicazione a RSF

In [72] Osher et al. usano una procedura del tutto analoga a quella vista nei paragrafi precedenti per risolvere le equazioni del modello region-based basato sul region-scalable fitting, presentato nella Sezione 1.4.2.

Analogamente al funzionale studiato nel metodo di Chan e Vese, anche l'energia del modello RSF è concava, e potrebbe avere minimi solo locali. Per eludere questo limite si utilizza ancora l'approccio del metodo GCS.

Considerando l'equazione di flusso (1.23), si prende $\nu = 1$ e si trascura l'ultimo termine:

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon \left[(-\lambda_1 e_1 + \lambda_2 e_2) + \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right]. \tag{2.21}$$

Si può dimostrare ([54]) che la soluzione stazionaria di (2.21) coincide con la soluzione stazionaria di:

$$\frac{\partial\phi}{\partial t} = \left[(-\lambda_1 e_1 + \lambda_2 e_2) + \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right].$$

Risolvere tale equazione di flusso corrisponde a minimizzare il seguente funzionale:

$$E(\phi) = \|\nabla\phi\|_1 + \langle\phi, \lambda_1 e_1 - \lambda_2 e_2\rangle. \tag{2.22}$$

Per garantire l'esistenza di un unico minimo globale è necessario restringere il dominio della soluzione in un intervallo finito $a_0 \leq \phi \leq b_0$ cosicché:

$$\phi^* = \min_{a_0 \leq \phi \leq b_0} E(\phi)$$

Al fine di aumentare la precisione del metodo si incorpora un termine dipendente dalla posizione dei contorni, sfruttando la *edge detector function* $g(s) = 1/(1+\beta s^2)$ già definita in precedenza. Nell'equazione (2.22) si sostituisce a $\|\nabla\phi\|$ la norma TV pesata ottenendo così come espressione finale del problema di minimizzazione:

$$\phi^* = \min_{a_0 \leq \phi \leq b_0} E(\phi) = \min_{a_0 \leq \phi \leq b_0} \left\{ |\nabla\phi|_g + \langle\phi, r\rangle \right\}$$

con $r = \lambda_1 e_1 - \lambda_2 e_2$.

Per applicare la procedura di Split-Bregman si introduce la variabile ausiliaria $\vec{d} = \nabla\phi$ e si aggiunge un termine di secondo grado per imporre il vincolo di uguaglianza:

$$(\phi^*, \vec{d}^*) = \min_{a_0 \leq \phi \leq b_0} \left\{ |\vec{d}|_g + \langle\phi, r\rangle + \frac{\lambda}{2}\|\vec{d} - \nabla\phi\|_2^2 \right\}.$$

Si può ora applicare la procedura di Split-Bregman, analogamente a quanto fatto in (2.17)-(2.18), ottenendo:

$$(\phi^{k+1}, \vec{d}^{k+1}) = \min_{a_0 \leq \phi \leq b_0} \left\{ |\vec{d}|_g + \langle\phi, r\rangle + \frac{\lambda}{2}\|\vec{d} - \nabla\phi - \vec{b}^k\|_2^2 \right\} \tag{2.23}$$

$$\vec{b}^{k+1} = \vec{b}^k + \nabla\phi^{k+1} - \vec{d}^{k+1}. \tag{2.24}$$

Come spiegato precedentemente, questo problema di minimizzazione si risolve fissando alternativamente $\vec{d}$ e ϕ. L'equazione di Eulero-Lagrange legata a (2.23), per $\vec{d}$ fisso è:

$$\triangle\phi = \frac{r}{\lambda} + \operatorname{div}(\vec{d} - \vec{b}), \text{ con } a_0 \leq \phi \leq b_0.$$

Tale equazione viene discretizzata come in (2.20) con un metodo di Gauss-Seidel associato alle differenze finite. Si procede poi con la minimizzazione di (2.24) rispetto a $\vec{d}$ per ϕ fisso, data da:

$$\vec{d}^{k+1} = shrink(\vec{b}^k + \nabla\phi^{k+1}, \frac{g}{\lambda}).$$

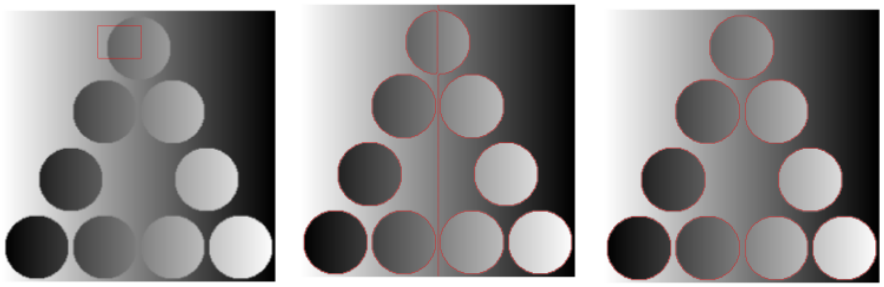
In conclusione, il metodo basato sulla region scalable energy viene risolto con il seguente algoritmo iterativo:

3 RSF: GCS E SPLIT BREGMAN

while $\|\phi^{k+1} - \phi^k\|_2 > tol$ **do**
  Aggiorna $r^k = \lambda_1 e_1^k - \lambda_2 e_2^k$
  $\phi^{k+1} = GS(r^k, \vec{d}^k, \vec{b}^k, \lambda)$
  $\vec{d}^{k+1} = shrink(\nabla\phi^{k+1} + \vec{b}^k, g/\lambda)$
  $\vec{b}^{k+1} = \vec{b}^k + \nabla\phi^{k+1} - \vec{d}^{k+1}$
  Trova $\Omega^{k+1} = \{x : \phi^{k+1}(x) > \alpha\}$
  Aggiorna i valori di e_1^{k+1} e e_2^{k+1}
end while

Al momento dell'inizializzazione la funzione di level set ϕ assume un valore costante b_0 all'interno di una regione e a_0 all'esterno della stessa. La media $\alpha = (a_0 + b_0)/2$ è il valore di soglia, che permette di identificare la posizione del contorno attivo.

In Figura 2.2 si può apprezzare come la risoluzione del problema originale con questo algoritmo conservi le buone proprietà del metodo. Esso riesce infatti a segmentare correttamente un'immagine fortemente disomogenea, obiettivo non raggiunto invece dal metodo di Chan e Vese (Fig. 2.2(b)).



(a) Contorno iniziale (b) Metodo in [42] (c) Metodo in [72]

Figura 2.2: Confronto tra due tecniche di segmentazione entrambe risolte con l'algoritmo di Split-Bregman; tratto da [72].

Capitolo 3

Adattività di griglia

La struttura del presente capitolo è essenzialmente divisa in tre parti. Nella Sezione 3.1 si introduce un problema modello e si fornisce una panoramica di alcune tra le tecniche di adattività più diffuse. Poi nella Sezione 3.2 si presentano le tecniche di stima dell'errore più utilizzate nella comunità scientifica. Successivamente, nella Sezione 3.3, viene introdotto uno tra gli stimatori più usati in ambito ingegneristico, grazie alla sua semplicità teorica e alla sua grande efficienza computazionale: si tratta dello stimatore proposto da Zienkiewicz e Zhu nel 1987 [77].

Tale stimatore è di particolare importanza per questo lavoro in quanto, nei capitoli che seguono, verrà utilizzato per adattare la griglia in alcuni problemi di segmentazione.

3.1 Introduzione

La modellazione numerica della realtà è raramente priva di errore. Tale errore può avere fonti differenti: può, ad esempio, derivare da un errore nella formulazione del modello matematico, oppure dalla discretizzazione del modello continuo, o ancora essere dovuto ad un errore di *roundoff* intrinseco nell'aritmetica di un computer.

In questa sede concentriamo la nostra attenzione sull'errore di discretizzazione, che si definisce come la differenza tra la soluzione esatta del problema differenziale e quella approssimata fornita dallo schema di discretizzazione scelto. In particolare, ci interessiamo al metodo degli elementi finiti ([59]), in cui il problema in esame viene discretizzato su una griglia di calcolo (*mesh*), composta solitamente da triangoli o tetraedri, e ridotto così ad un sistema di equazioni algebriche. La soluzione è infatti approssimata su ciascun elemento della mesh mediante un'opportuna combinazione lineare di funzioni di base (ad esempio di polinomi). Il grado di tali funzioni e le caratteristiche della griglia influenzano l'errore di discretizzazione. La necessità di confrontarsi con questo limite delle simulazioni ha stimo-

lato il proliferare di analisi sull'errore: come si può misurare, controllare e minimizzare l'errore di discretizzazione?

Una tecnica spesso usata per ridurre l'errore nelle simulazioni ad elementi finiti consiste nella costruzione di una griglia di calcolo che tenga conto del comportamento locale della soluzione, cioè sia più raffinata laddove l'errore è maggiore e viceversa. Le caratteristiche della griglia di calcolo sono infatti fondamentali: una buona griglia non solo permette di catturare le peculiarità della soluzione del problema, ma soprattutto di ridurre il costo computazionale. Per questo motivo sono state sviluppate diverse tecniche di **adattività** di griglia che, attraverso algoritmi ricorsivi, costruiscono una mesh efficiente. Al fine di misurare e controllare l'errore di discretizzazione si ricorre inoltre alle cosiddette stime dell'errore, che possono essere a priori o a posteriori.

3.1.1 Errore di discretizzazione

Introduciamo un problema modello definito su un dominio limitato $\Omega \subset \mathbb{R}^2$ con frontiera Lipschitziana $\partial\Omega = \Gamma_N \cup \Gamma_D$, con $\overset{\circ}{\Gamma}_N \cap \overset{\circ}{\Gamma}_D = \emptyset$:

$$\begin{cases} -\triangle u = f & \text{in } \Omega, \\ \dfrac{\partial u}{\partial n} = g & \text{su } \Gamma_N, \\ u = 0 & \text{su } \Gamma_D. \end{cases} \tag{3.1}$$

Si assume la seguente regolarità sui dati: $f \in L^2(\Omega)$ e $g \in L^2(\Gamma_N)$.
La forma variazionale di questo problema è:

$$\begin{gathered} \text{trovare } u \in V = \{v \in H^1(\Omega) : v = 0 \text{ su } \Gamma_D\} \text{ tale che:} \\ B(u,v) = L(v) \quad \forall v \in V \end{gathered} \tag{3.2}$$

dove

$$B(u,v) = \int_\Omega \nabla u \cdot \nabla v d\mathbf{x} \quad \text{e} \quad L(v) = \int_\Omega f v d\mathbf{x} + \int_{\Gamma_N} g v ds.$$

Sia ora V_h una famiglia di spazi dipendente da un parametro positivo h, tali che

$$V_h \subset V, \quad \dim(V_h) = N_h < +\infty \quad \forall h > 0.$$

Allora il problema discreto assume la forma

$$\text{trovare } u_h \in X : B(u_h, v_h) = L(v_h) \quad \forall v_h \in V_h$$

e viene detto *problema di Galerkin*. La funzione u_h è l'approssimazione alla Galerkin della soluzione u di (3.2). Indicando con $\{\varphi_j, j = 1, \ldots, N_h\}$ una

base di V_h, basta che il problema di Galerkin sia verificato per ogni funzione della base, cioè che valga:

$$B(u_h, \varphi_i) = L(\varphi_i), \quad \forall i = 1, \dots, N_h.$$

Una scelta classica per V_h è lo spazio ad elementi finiti di grado r sui singoli triangoli della mesh $\mathcal{T}_h$:

$$X_h^r = \{v_h \in C^0(\bar{\Omega}) : v_h|_K \in \mathbb{P}_r, \quad \forall K \in \mathcal{T}_h\} \quad r = 1, 2, \dots$$

dove $\mathbb{P}_r$ è lo spazio dei polinomi di grado (globale) minore o uguale a r.

Quando si parla di soluzione approssimata di grado r si sceglie dunque $V_h = X_h^r \cap V$, dove l'intersezione con lo spazio V serve per garantire le condizioni al bordo essenziali.

Per migliorare l'approssimazione agli elementi finiti u_h della soluzione u si desidera controllare l'errore di discretizzazione $e_h = u - u_h$. Tale quantità, appartenente allo spazio V, soddisfa le seguenti relazioni:

$$\begin{aligned} &B(e_h, v) = B(u, v) - B(u_h, v) = L(v) - B(u_h, v) \quad \forall v \in V; \\ &B(e_h, v_h) = 0 \quad \forall v_h \in V_h \quad \text{(ortogonalità di Galerkin).} \end{aligned} \tag{3.3}$$

L'errore e_h dipende dalla regolarità della soluzione u, dal grado r di u_h e dal passo h di griglia. Infatti, per $u \in H^{p+1}(\Omega)$, vale la relazione

$$\|e_h\|_{H^1(\Omega)} \leq Ch^s |u|_{H^{s+1}(\Omega)}, \qquad s = \min\{r, p\} \tag{3.4}$$

dove C è una costante indipendente da h.

3.1.2 Tecniche di adattività

Alla luce della stima (3.4), si possono seguire due strade per migliorare la qualità dell'approssimazione u_h:

- ridurre il passo h della griglia $\mathcal{T}_h$;
- aumentare il grado polinomiale r (a patto di avere una soluzione u sufficientemente regolare).

Nel primo caso si parla di *h-adattività*, nel secondo di *p-adattività*. Ovviamente è anche possibile scegliere una tecnica mista, che modifichi contemporaneamente la griglia e il grado della soluzione.

L'adattività di tipo h, detta anche adattività di griglia, fu introdotta per prima ed è ancora adesso la tecnica più adottata nella pratica; per questo motivo la trattazione che segue è incentrata sulla h-adattività.

L'adattazione di griglia è una procedura iterativa in cui si alternano la soluzione del problema discreto e la costruzione della nuova griglia. Più nello specifico, a partire da una triangolazione iniziale del dominio $\mathcal{T}_h^0$, si vuole generare una sequenza di griglie (adattate) $\mathcal{T}_h^k$, con $k > 0$, che

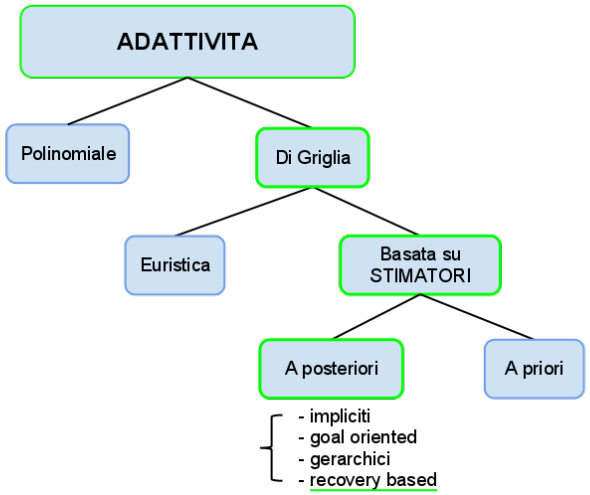


Figura 3.1: Le diverse procedure di adattività; evidenziata in verde quella di maggiore interesse per questo lavoro.

permettano di migliorare la qualità dell'approssimazione u_h mantenendo ridotto il costo computazionale. Come mostrato in Figura 3.1, che esemplifica i principali tipi di adattazione, possiamo essenzialmente distinguere le tecniche di adattazione di griglia in euristiche e teoriche.

Le tecniche euristiche si basano su informazioni generalmente geometriche (gradiente, curvatura) della soluzione approssimata. Ne deriva un approccio abbastanza intuitivo: laddove u_h presenta brusche variazioni la griglia viene raffinata e viceversa. Le variazioni della soluzione sono rappresentate dal valore delle sue derivate, quindi due valide alternative per guidare la procedura adattativa sono il gradiente oppure la matrice Hessiana. Proprio sul valore di quest'ultima si basa, ad esempio, il comando `adaptmesh` di *FreeFem++* [45], le cui caratteristiche verranno descritte accuratamente nel Capitolo 5. Esempi di adattazione euristica della griglia nell'ambito del trattamento delle immagini saranno presentati in Sezione 3.4.

Al contrario le tecniche teoriche si basano sui cosiddetti stimatori dell'errore di cui ci occuperemo nel resto del capitolo. Si vuole controllare l'errore di discretizzazione e_h misurato rispetto ad una certa norma. Ad esempio, se ci riferiamo alla norma dell'energia

$$|||e_h|||^2 = \int_\Omega |\nabla u - \nabla u_h|^2 d\mathbf{x}, \tag{3.5}$$

vogliamo controllare $|||e_h|||$ con una stima della forma

$$|||e_h||| \leq \eta = \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2},$$

dove η è lo stimatore globale, mentre η_K rappresenta il corrispondente stimatore locale, ovvero associato all'elemento K della triangolazione $\mathcal{T}_h$.

Basandosi su uno stimatore dell'errore di questo tipo si può procedere con la generazione della griglia adattata. Anche in questo frangente, esistono diverse strategie:

- economicità della griglia : fissato il numero massimo di elementi K, costruisco la griglia che minimizza l'errore di discretizzazione;

- controllo dell'errore : fissata la tolleranza τ su e_h, genero la griglia con il numero minimo di elementi e tale per cui $\eta \simeq \tau$.

Tali strategie non sono equivalenti, e possono dare luogo a risultati sensibilmente diversi. La strategia da seguire viene scelta in base all'obiettivo della simulazione: la prima permette di controllare il costo computazionale, mentre la seconda produce un'approssimazione con un'accuratezza desiderata.

Solitamente queste due tecniche vengono poi combinate con un criterio ben noto nell'ambito dell'adattazione di griglia, ovvero la cosiddetta *equidistribuzione* dell'errore: fissato il numero di elementi N della griglia o, in alternativa, la tolleranza desiderata τ, si crea una griglia tale per cui $\eta_K = \frac{\tau}{N}$ è costante in ogni K.

Ovviamente la procedura alla base dell'adattazione di griglia guidata da uno stimatore è ricorsiva: a partire dalla griglia iniziale $\mathcal{T}_h^0$, ad ogni passo $j > 0$ si calcola la soluzione approssimata u_h^j, usata per valutare lo stimatore η. Si procede con la costruzione della griglia $\mathcal{T}_h^{j+1}$ attraverso una delle tecniche citate. Risolvendo il problema differenziale sulla nuova griglia si ottiene così la soluzione approssimata u_h^{j+1}; il procedimento iterativo continua finché non si ottiene la tolleranza desiderata, solitamente valutata semplicemente come:

$$\|u_h^{j+1} - u_h^j\| \leq \text{ toll.}$$

La costruzione e l'utilizzo degli stimatori ai fini di un'adattazione di griglia sarà argomento delle prossime sezioni: nella Sez. 3.2 si descrivono alcune tra le più diffuse tipologie di stimatori per poi focalizzare l'attenzione, nella Sez. 3.3, sullo stimatore usato nei prossimi capitoli.

3.2 Stima dell'errore

Si è già fatto riferimento al ruolo fondamentale degli stimatori dell'errore nella creazione di una griglia computazionale adattata. È però necessario che uno stimatore dell'errore soddisfi alcune proprietà specifiche per assicurare l'efficacia della strategia adattativa.

Innanzitutto uno stimatore deve essere *locale*, cioè calcolabile a partire da quantità dipendenti solamente dall'elemento K o, al massimo, da un insieme $\tilde{K}$ di elementi ad esso adiacenti. L'insieme $\tilde{K}$ è detto *patch* di elementi

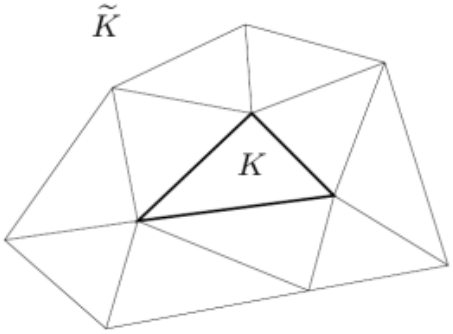


Figura 3.2: Esempio di patch $\tilde{K}$ associato all'elemento K.

associato a K e contiene K insieme a tutti gli elementi che hanno in comune con K almeno un vertice, come mostrato in Figura 3.2.
Un buono stimatore (globale) dell'errore deve inoltre essere *affidabile*, ossia deve

$$\exists C_1 \simeq 1 \text{ tale che } C_1 \, |||e_h||| \leq \eta,$$

ed *efficiente*, cioè:

$$\exists C_2 \simeq 1 \text{ tale che } \eta \leq C_2 \, |||e_h||| \, .$$

Queste due relazioni garantiscono che η tenda a zero con la stessa velocità dell'errore e_h. Data la natura locale dello stimatore, quando tali proprietà valgono a livello globale valgono in maniera analoga anche per lo stimatore locale η_K. Non sempre efficienza e affidabilità sono verificabili contemporaneamente; in caso lo siano, si dice che lo stimatore è *robusto*.

La "bontà" di uno stimatore si misura con il cosiddetto *effectivity index*, definito come:

$$\theta = \frac{\eta}{|||e_h|||}. \tag{3.6}$$

Uno stimatore ottimale dovrebbe avere $\theta \simeq 1$ ma, nelle applicazioni ingegneristiche, spesso si considerano accettabili valori fino a 3.

Si può distinguere tra stimatori *a priori* e *a posteriori*. I primi sono funzioni della soluzione esatta u e della dimensione dell'elemento K; cioè forniscono essenzialmente informazioni sull'ordine di convergenza del metodo numerico utilizzato. Gli stimatori a priori si differenziano dagli stimatori a posteriori che si basano invece sulla soluzione approssimata u_h, calcolabile esplicitamente, e dunque forniscono informazioni quantitative.

I prossimi paragrafi contengono un'analisi approfondita degli stimatori sia a priori (Sez. 3.2.1), sia a posteriori (Sez. 3.2.2).

3.2.1 Stimatori a priori

La tecnica di stima a priori più diffusa si basa su una stima dell'errore di interpolazione, cioè la differenza tra una data funzione e una sua opportuna interpolante.

Sia Π_K^r l'operatore di *interpolazione* locale che, per ogni $r \geq 1$, associa a tutte le funzioni $v \in C^0(\bar{\Omega})$ il polinomio $\Pi_K^r v \in \mathbb{P}_r(K)$, interpolante v nei gradi di libertà dell'elemento $K \in \mathcal{T}_h$. L'operatore di interpolazione globale $\Pi_h^r : C^0(\bar{\Omega}) \to V_h$, $\forall r \geq 1$ si definisce nel seguente modo:

$$\Pi_h^r v \in X : \quad \Pi_h^r v|_K = \Pi_K^r(v|_K) \qquad \forall K \in \mathcal{T}_h. \tag{3.7}$$

In [59] si ricava una stima locale dell'errore di interpolazione caratterizzata dalla seguente forma:

$$|v - \Pi_K^r v|_{H^1(K)} \leq C \frac{h_K^{r+1}}{\rho_K} |v|_{H^{r+1}(K)} \quad \forall v \in H^{r+1}(K)$$

dove ρ_K e h_K denotano rispettivamente la sfericità e il diametro dell'elemento K. La corrispondente stima globale dell'errore di interpolazione è:

$$|v - \Pi_h^r v|_{H^1(\Omega)} \leq C h^r |v|_{H^{r+1}(\Omega)} \quad \forall v \in H^{r+1}(\Omega)$$

essendo $C = C(r, K)$ una costante indipendente da v e da h. Da qui si deduce la seguente stima a priori per l'errore di discretizzazione, valida per $u \in H^{r+1}(\Omega)$ e per u_h ottenuta con il metodo agli elementi finiti di grado r:

$$\|u - u_h\|_{H^1(\Omega)} \leq C \left(\sum_{K \in \mathcal{T}_h} h_K^{2r} |u|^2_{H^{r+1}(K)} \right)^{1/2}. \tag{3.8}$$

Per procedere con la costruzione della griglia adattata, si approssimano le derivate di ordine $r+1$ di u che intervengono nella definizione della seminorma $|u|_{H^{r+1}(K)}$ con le derivate di una soluzione approssimata sufficientemente ricca u_{h^*}, calcolata su una griglia di passo h^*. Solitamente di costruisce u_{h^*} o utilizzando elementi finiti di grado maggiore di r oppure usando una griglia più fitta e provvedendo poi a ricostruire opportunamente le derivate richieste. Infatti, una possibile difficoltà intrinseca della stima (3.8) risiede nella mancata regolarità delle derivate di ordine $r+1$ di u_{h^*} in Ω. Si consideri ad esempio il caso di elementi finiti lineari ($r = 1$): occorre in questo caso procedere con un'adeguata ricostruzione della matrice Hessiana, indicata con $\mathbf{D}_h^2\big|_{K^*}$, $\forall K^* \in \mathcal{T}_{h^*}$. Tale matrice fornisce un'approssimazione costante a tratti delle derivate seconde e gode della regolarità richiesta. Il termine di destra della (3.8) può così essere usato come stimatore, ricorrendo ad una delle tecniche di adattazione precedentemente citate.

3.2.2 Stimatori a posteriori

I pionieri delle tecniche di stima a posteriori dell'errore di discretizzazione sono stati I. Babuška e W.C. Rheinboldt, che nel 1978 hanno compreso l'utilità di questa analisi come base per la costruzione di risolutori adattativi ad elementi finiti [10]. Negli anni successivi sono state introdotte numerose tecniche innovative nell'analisi a posteriori, alcune delle quali sono ricordate nel seguito del paragrafo.

Essendo il panorama degli stimatori a posteriori dell'errore piuttosto ampio, si è scelto di descrivere unicamente le metodologie più note e di spiegare brevemente il loro utilizzo applicandole al problema modello (3.1).

Stimatori espliciti/residuali

Come suggerisce il loro stesso nome, questi stimatori possono essere calcolati direttamente a partire dall'approssimazione ad elementi finiti e dai dati del problema. Sono stati proposti nel 1984 da L. Demkowicz et al. in [29], quasi in contemporanea agli analoghi stimatori proposti da R.E. Bank e A. Weiser [12].

Prima di ricavare l'espressione dello stimatore, introduciamo alcune utili notazioni. Indichiamo con

$$r_h = f + \triangle u_h \text{ in } K,$$

il residuo interno associato al problema (3.1) e con

$$R = \begin{cases} g - \dfrac{\partial u_h}{\partial \mathbf{n}_K} & \text{su } e \cap \Gamma_N, \\ -\left[\dfrac{\partial u_h}{\partial \mathbf{n}_K}\right]_e & \text{su } e \setminus \Gamma_N. \end{cases}$$

il corrispondente residuo di bordo. In queste espressioni $\mathbf{n}_K$ è il versore normale al lato e di K, mentre $\left[\frac{\partial u_h}{\partial \mathbf{n}_K}\right]_e$ è il salto della derivata conormale sul lato e che separa gli elementi K e K', definito come:

$$\left[\frac{\partial u_h}{\partial \mathbf{n}_K}\right] = \mathbf{n}_K \cdot \nabla u_h|_K + \mathbf{n}_{K'} \cdot \nabla u_h|_{K'}. \tag{3.9}$$

Tenendo presente queste notazioni, integrando per parti l'equazione (3.3) su ogni elemento K si ricava:

$$B(e_h, v) = \sum_{K \in \mathcal{T}_h} \int_K r_h v d\mathbf{x} + \sum_{\gamma \in \partial \mathcal{T}_h} \int_\gamma R v ds \quad \forall v \in V.$$

Sfruttando poi l'ortogonalità di Galerkin per ottenere dei termini dipendenti dall'approssimazione $\Pi_h^r v$ di v nel sottospazio V_h e procedendo con

alcune maggiorazioni standard, si ottiene la stima a posteriori dell'errore di discretizzazione cercata, della forma:

$$|||e_h|||^2 \leq C \sum_{K \in \mathcal{T}_h} (h_K^2 \|r_h\|^2_{L^2(K)} + \frac{1}{2} h_K \|R\|^2_{L^2(e)}). \tag{3.10}$$

Il vantaggio di questi stimatori consiste nella semplicità di calcolo della stima dell'errore rappresentata dal termine di destra della (3.10). Talvolta però si obietta che si tratta di stime poco precise e pessimiste in quanto le numerose maggiorazioni effettuate provocano una perdita di informazioni; infatti termini con origini molto diverse sono sommati nell'unica costante C di (3.10) [2]. In realtà si vedrà che quasi tutti gli stimatori dell'errore che incontreremo nelle sezioni successive sono definiti a meno di una costante.

Stimatori impliciti

Una strada alternativa, che pone in parte rimedio all'imprecisione degli stimatori espliciti, consiste nel risolvere un problema ausiliario avente i residui come dati. In questo modo si conserva la struttura dell'equazione originale e si riduce allo stesso tempo il numero di costanti generiche presenti nella stima. Anche questi stimatori sono stati introdotti da Babuška e Rheinboldt [8, 9].

Partendo dall'equazione dei residui (3.3), la si vuole discretizzare al fine di ottenere un'approssimazione $\bar{e}$ della funzione errore. Cercando una soluzione nello spazio ad elementi finiti originale si trova unicamente la soluzione triviale $\bar{e} = 0$. È quindi necessario porsi in uno spazio più ampio. In alternativa si può sostituire il problema globale con una sequenza di problemi locali disaccoppiati, definiti o su un singolo elemento (*element residual method*), o su un patch di elementi (*subdomain residual method*). Sia $\tilde{\Omega}_n$ il patch di elementi associati al nodo x_n:

$$\tilde{\Omega}_n = \{K \in \mathcal{T}_h \text{ t.c. } x_n \in K\}. \tag{3.11}$$

Allora la formulazione del subdomain residual method consiste nel trovare $e_n \in H_0^1(\tilde{\Omega}_n)$ tale che:

$$B_n(e_n, v) = L_n(v) - B_n(u_h, v) \quad \forall v \in H_0^1(\tilde{\Omega}_n) \tag{3.12}$$

dove la forma bilineare B_n e il funzionale L_n sono dati da:

$$B_n(u, v) = \int_{\tilde{\Omega}_n} \nabla u \cdot \nabla v d\mathbf{x} \quad \text{e} \quad L_n(v) = \int_{\tilde{\Omega}_n} f v d\mathbf{x} + \int_{\partial\tilde{\Omega}_n \cap \Gamma_N} g v ds.$$

La soluzione e_n di (3.12) può essere usata per fornire una stima dell'errore di approssimazione sul patch considerato, ovvero

$$\eta_n = |||e_n|||_{\tilde{\Omega}_n},$$

dove $|||\cdot|||_{\tilde{\Omega}_n}$ è la norma dell'energia calcolata sul patch di elementi $\tilde{\Omega}_n$

$$|||e_n|||^2_{\tilde{\Omega}_n} = \int_{\tilde{\Omega}_n} |\nabla u - \nabla u_h|^2 d\mathbf{x}.$$

Lo stimatore globale si ottiene anche in questo caso sommando i contributi locali di tutti i sottodomini:

$$\eta = \left(\sum_{n \in \mathcal{N}} \eta_n^2 \right)^{1/2}$$

essendo $\mathcal{N}$ l'insieme dei nodi della griglia.

Nella pratica questo metodo è raramente usato vista la difficoltà di approssimare l'equazione (3.12) su patch irregolari.

Stimatori *goal-oriented*

Spesso nelle simulazioni ingegneristiche si è interessati al valore di una grandezza fisica $Q(u)$ che dipende dalla soluzione u come, ad esempio, il flusso di u attraverso una porzione di frontiera oppure il suo valore in un punto o in una regione limitata del dominio.

Indichiamo con $Q : V \to \mathbb{R}$ il funzionale lineare e limitato che rappresenta la grandezza di interesse; in questo caso l'errore d'approssimazione che vogliamo valutare e controllare è dato da:

$$Q(e_h) = Q(u) - Q(u_h).$$

Esiste una famiglia di stimatori dell'errore che sfrutta le informazioni fornite da un problema ausiliario, noto come problema aggiunto (o duale), per ottenere una stima dell'errore sul funzionale $Q(e_h)$ [60]. Questo tipo di adattività si dice goal-oriented, e la stima dell'errore di approssimazione si ottiene partendo dalla formulazione debole del problema duale di (3.1):

$$\text{trovare } \phi \in V : \quad \int_\Omega \nabla\phi \nabla v d\mathbf{x} = Q(v) \qquad \forall v \in V. \tag{3.13}$$

Osserviamo che il funzionale d'interesse è legato al termine forzante del problema duale.

Utilizzando l'ortogonalità di Galerkin si ha:

$$\begin{aligned} Q(e_h) &= \int_\Omega \nabla\phi \nabla e_h d\mathbf{x} \\ &= \sum_{K \in \mathcal{T}_h} \left[\int_K r_h(\phi - \phi_h) d\mathbf{x} + \frac{1}{2} \int_e R(\phi - \phi_h) ds \right] \end{aligned} \tag{3.14}$$

dove $\phi_h \in V_h$ è un opportuno interpolante di ϕ, mentre r_h ed R sono i residui interni e di bordo definiti in precedenza.

Applicando la disuguaglianza di Cauchy-Schwarz ad ogni termine dell'integrale (3.14) si ottiene la stima cercata

$$|Q(e_h)| \leq \sum_{K \in \mathcal{T}_h} [\rho_K(u_h) w_K(\phi)]$$

dove ρ_K è il residuo locale (cioè associato all'elemento K)

$$\rho_K(u_h) = h_K \|r_h\|_{L^2(K)} + \frac{1}{2} h_K^{1/2} \|R\|_{L^2(e)},$$

e w_K è il peso locale

$$w_K(\phi) = \max \left(\frac{1}{h_K} \|\phi - \phi_h\|_{L^2(K)}, \frac{1}{h_K^{1/2}} \|\phi - \phi_h\|_{L^2(e)} \right).$$

Mentre il residuo misura come la soluzione discreta approssima il problema differenziale in esame, il peso tiene conto di come tale informazione si propaga nel dominio per effetto del funzionale Q.

Le stime di Q sono spesso utilizzate per effettuare un'adattazione di griglia *goal-oriented*, con degli schemi ad elementi finiti che mirano a ridurre l'errore direttamente per il controllo della grandezza di interesse. Ciò vuol dire che le griglie adattate ottenute in tale modo non saranno ottimali per controllare la soluzione u bensì la quantità $Q(u)$. Questo tipo di adattività si traduce solitamente in una drastica riduzione del costo computazionale, visto che ci si limita a catturare le caratteristiche della soluzione con una forte influenza su Q.

Stimatori gerarchici

Una tecnica efficace per ottenere delle stime dell'errore consiste nel risolvere il problema di interesse usando schemi caratterizzati da un'accuratezza diversa, e poi confrontare i risultati così ottenuti. Questa metodologia è stata introdotta da R.E Bank e R.K. Smith in [11] ed è largamente diffusa grazie alla sua facilità di applicazione ed implementazione.

A tale fine si considerino due spazi ad elementi finiti $X, Y \subset V$, con Y più ricco di X. Si cerca quindi una soluzione approssimata u_h^* del problema modello (3.1) nello spazio $X^* = X \oplus Y$:

$$B(u_h^*, v_h^*) = L(v_h^*) \quad \forall v_h^* \in X^*.$$

Le tecniche più utilizzate per costruire X^* sono due: si può arricchire lo spazio X con funzioni di base di grado superiore, oppure raffinare la griglia di calcolo. Da qui si deduce quello che può essere lo svantaggio principale di questa classe di metodi, che consiste nel costo computazionale necessario per risolvere un problema di complessità maggiore rispetto a quello iniziale.

Un'ipotesi fondamentale per procedere con la stima a posteriori dell'errore è la cosiddetta *saturation assumption* ([12]) con la quale si richiede che:

$$\exists \text{ una costante } \beta \in [0,1) \quad \text{t.c.} \quad |||u - u_h^*||| \leq \beta \, |||u - u_h||| \,. \tag{3.15}$$

Nel caso in cui l'ipotesi sia soddisfatta, l'approssimazione appena calcolata è migliore di quella originale u_h, e la differenza e_h^* tra le due soluzioni può essere utilizzata come stima dell'errore:

$$|||e_h||| = |||u - u_h||| \approx |||u_h^* - u_h||| = |||e_h^*||| \,.$$

Questa approssimazione dell'errore può essere trovata direttamente risolvendo il problema:

$$B(e_h^*, v_h^*) = L(v_h^*) - B(u_h, v_h^*) \quad \forall v_h^* \in X^*. \tag{3.16}$$

Si prova facilmente che, se vale l'ipotesi (3.15), e^* soddisfa le seguenti relazioni:

$$|||e_h^*||| \leq |||e_h||| \leq \frac{1}{\sqrt{1-\beta^2}} \, |||e_h^*||| \,.$$

La difficoltà di questo approccio risiede nel calcolo di e_h^* nello spazio X^*. Il problema può essere riformulato come la ricerca di $e_X^* \in X$ e $e_Y^* \in Y$ tali che $e_X^* + e_Y^* = e_h^*$. Una possibile semplificazione del problema consiste nel trascurare i termini di accoppiamento tra i due spazi. Nella pratica si cerca un'approssimazione $\bar{e}_h = \bar{e}_X + \bar{e}_Y$ di e_h^* risolvendo l'equazione (3.16) separatamente in X e Y:

$$\begin{aligned} &B(\bar{e}_X, v_X) = 0 \quad \forall v_X \in X \\ &B(\bar{e}_Y, v_Y) = L(v_Y) - B(u_h, v_Y) \quad \forall v_Y \in Y. \end{aligned} \tag{3.17}$$

La soluzione della prima equazione è identicamente nulla; si può dunque usare come stimatore $|||e_h||| \approx |||\bar{e}_Y|||$ ottenuto risolvendo (3.17) nel sottospazio Y. L'accuratezza dello stimatore così ottenuto dipende dalla perdita di informazioni causata dall'eliminazione dei termini di accoppiamento.

Esistono anche altre tecniche per trovare la soluzione del problema (3.1) nello spazio X^* (si veda, ad esempio, [33]).

3.3 Stimatori recovery-based

Un'altra importante famiglia di stimatori a posteriori per l'errore di discretizzazione è formata dagli stimatori basati sugli operatori di ricostruzione del gradiente. In vista della loro rilevanza nel seguito di questo lavoro, si è scelto di dedicare alla loro trattazione un'intera sezione.

Il primo stimatore a posteriori di questo tipo fu proposto da O.C. Zienkiewicz e J.Z. Zhu nel 1987, nell'ambito della discretizzazione ad elementi finiti di problemi di elasticità lineare [77]. Dal nome dei due autori, spesso ci si riferisce a questi stimatori come stimatori **ZZ**.

Il gradiente della soluzione dei problemi differenziali è di spiccato interesse in svariate applicazioni ingegneristiche, come ad esempio il calcolo degli sforzi in elasticità o il riconoscimento dei contorni di un'immagine. Da questo presupposto è nata l'idea degli stimatori basati sugli operatori di ricostruzione del gradiente. Infatti, anche se l'approssimazione agli elementi finiti può essere ottimale in termini di regolarità per il problema considerato, il corrispondente gradiente può essere discontinuo attraverso i lati della mesh. Questo aspetto diventa problematico se si è interessati ad un'approssimazione regolare delle quantità rappresentate dal gradiente stesso.

Per porre rimedio a tale irregolarità si possono usare delle tecniche dette di *post-processing*, che permettono di ricostruire un'approssimazione più regolare del gradiente, $G_R(u_h)$, a partire dal gradiente stesso ∇u_h. I gradienti così ricostruiti sono poi usati per ricavare uno stimatore a posteriori dell'errore di discretizzazione nella norma dell'energia, dato dalla differenza tra il valore esatto del gradiente ∇u_h e il valore del gradiente ricostruito $G_R(u_h)$ [77, 78, 79].

3.3.1 Operatori di *recovery*

Per assicurarsi che $G_R(u_h)$ sia una buona approssimazione del gradiente esatto ∇u è necessario che l'operatore di recovery $G_R(\cdot) : V_h \to V_h \times V_h$ soddisfi alcune proprietà specifiche, il cui studio è argomento di questa sezione.

Innanzitutto si richiede che valga una proprietà di **consistenza**, cioè che, in alcune condizioni particolari, il gradiente ricostruito coincida con il gradiente esatto. Si verifica facilmente che questa proprietà è soddisfatta se u è un polinomio di secondo grado e se $u_h \equiv \Pi_h^1 u$, dove $\Pi_h^1 u$ è l'interpolante lineare a tratti di u definito in (3.7). Sotto tali condizioni si ha infatti che

$$G_R(\Pi_h^1 u) = \nabla u, \ \forall u \in \mathbb{P}_2(\tilde{K}),$$

con K elemento generico della mesh e $\tilde{K}$ patch associato all'elemento K. In generale invece si ha che

$$G_R(\Pi_h^p u) = \Pi_h^p(\nabla u), \ \forall u \in \mathbb{P}_{p+1}(\tilde{K}). \tag{3.18}$$

Inoltre si richiede che il gradiente ricostruito sia facilmente gestibile da un punto di vista sia numerico sia analitico. L'operatore deve quindi essere

lineare e **limitato**, ovvero:

$$\begin{aligned} &G_R(u_h + v_h) = G_R(u_h) + G_R(v_h) && \forall u_h, v_h \in V_h \\ &\|G_R(u_h)\|_{L^\infty(K)} \leq C|u_h|_{W^{1,\infty}(\tilde{K})} && \forall u_h \in V_h. \end{aligned} \tag{3.19}$$

Un'ulteriore richiesta di tipo pratico deriva dal desiderio che il gradiente ricostruito sia poco costoso da un punto di vista computazionale. Se fosse necessario ricorrere a calcoli sull'intero dominio, l'elevato costo computazionale richiesto per calcolare $G_R(u_h)$ non giustificherebbe l'utilizzo del gradiente ricostruito nell'analisi a posteriori. Di conseguenza, si chiede che l'operatore sia **locale**: dato un punto $x_0 \in K$, il valore di $G_R(u_h)(x_0)$ deve dipendere unicamente dai valori di u_h nel patch $\tilde{K}$.

Le condizioni fin qui enunciate assicurano che il gradiente approssimato $G_R(u_h)$ sia una buona approssimazione di quello reale, come conferma il seguente

Teorema 3.1

Sia $\mathcal{T}_h$ una partizione regolare del dominio Ω e sia V_h uno spazio ad elementi finiti basato su polinomi di grado p. Se l'operatore G_R soddisfa le condizioni di consistenza, linearità, limitatezza e località e $u \in H^{p+2}(\tilde{K})$, allora si ha

$$\|\nabla u - G_R(\Pi_h^p u)\|_{L^2(\tilde{K})} \leq C h_K^{p+1} |u|_{H^{p+2}(\tilde{K})}$$

dove la costante $C > 0$ è indipendente da h_K ed u.

La dimostrazione si basa essenzialmente sulle proprietà dell'interpolante Π_h^p e si può trovare in [2]. Analogamente si ricava un risultato globale dato dal

Corollario 3.2

Sotto le ipotesi del Teorema 3.1 si ha che:

$$\|\nabla u - G_R(\Pi_h^p u)\|_{L^2(\Omega)} \leq C h^{p+1} |u|_{H^{p+2}(\Omega)}$$

dove $C > 0$ è indipendente da $h = \max\{h_K\}$.

Nella pratica gli interpolanti $\Pi_h^p u$ e $\Pi_h^p(\nabla u)$ sono incogniti, dato che dipendono dalla soluzione esatta. L'operatore di recovery viene quindi applicato all'approssimazione agli elementi finiti u_h, per cui vale la stima classica a priori (3.8).

La cosiddetta proprietà di **superconvergenza** (per un approfondimento sull'argomento si veda [71]) vale per $u \in H^{p+2}(\Omega)$ se

$$\|u_h - \Pi_h^p u\|_{H^1(\Omega)} \leq C(u) h^{p+\tau}, \tag{3.20}$$

con $\tau \in (0,1]$ e $C(u) > 0$ indipendente da h. Allora l'operatore di recovery applicato alla soluzione ad elementi finiti è un'ottima approssimazione del gradiente esatto della soluzione. Vale infatti il seguente

Teorema 3.3

Supponiamo che $u \in H^{p+2}(\Omega)$, che G_R sia un operatore locale che soddisfa le proprietà (3.18)-(3.19) e che valga la proprietà di superconvergenza (3.20). Allora si ha che

$$\|\nabla u - G_R(u_h)\|_{L^2(\Omega)} \leq C(u) h^{p+\tau}$$

con $C(u) > 0$ indipendente da h.

Bisogna evidenziare come le ipotesi che garantiscono la superconvergenza siano valide unicamente in circostanze molto rare; infatti questa proprietà dipende da ipotesi restrittive sulla regolarità della soluzione e della mesh. Nella pratica tali ipotesi non sono quasi mai soddisfatte e, in particolare, non lo sono nel caso di procedure di adattività di griglia. Tuttavia l'accuratezza e la robustezza degli stimatori basati sull'operatore di recovery sono conservate anche quando la proprietà (3.20) non è verificata. La spiegazione di questo fenomeno non è ancora stata dimostrata rigorosamente.

In ogni caso, se valgono le proprietà elencate in questa sezione, il Teorema 3.3 assicura che $G_R(u_h)$ è un'approssimazione di ∇u migliore di quanto lo sia ∇u_h. L'applicazione dell'operatore di recovery alla soluzione agli elementi finiti è quindi più che giustificata, così come l'utilizzo di $G_R(u_h)$ per la costruzione di uno stimatore a posteriori come vedremo in Sezione 3.3.

Tutte le proprietà appena elencate definiscono la struttura degli operatori di recovery:

Lemma 3.4

Se l'operatore G_R soddisfa le condizioni (3.18)-(3.19) e (3.20) allora è della forma:

$$G_R[v] = \sum_{k \in \mathcal{N}} g_k[v] \varphi_k,$$

dove $\{\varphi_k : k \in \mathcal{N}\}$ è la base di Lagrange associata ai nodi $\{x_k : k \in \mathcal{N}\}$ e i funzionali lineari g_k, da definirsi opportunamente, soddisfano le seguenti proprietà:

$$\begin{aligned} &\exists C \text{ t.c., } \forall v \in W^{1,\infty}(\tilde{\Omega}_k), && |g_k[v]| \leq C |v|_{W^{1,\infty}(\tilde{\Omega}_k)} \\ &\forall \text{ polinomio } v \in \mathbb{P}_{p+1}, && g_k[\Pi_h^{p+1} v] = \nabla v(x_k) \end{aligned} \tag{3.21}$$

dove $\tilde{\Omega}_k$ è il patch associato al nodo x_k definito in (3.11).

Il processo di costruzione di un operatore di recovery si riduce quindi a scegliere una procedura di post-processing per i valori dei gradienti che

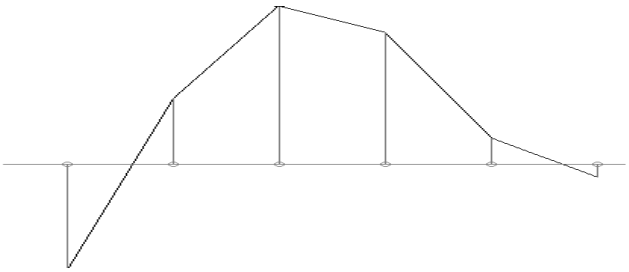
soddisfi le condizioni (3.21).

3.3.2 Tecniche di ricostruzione del gradiente

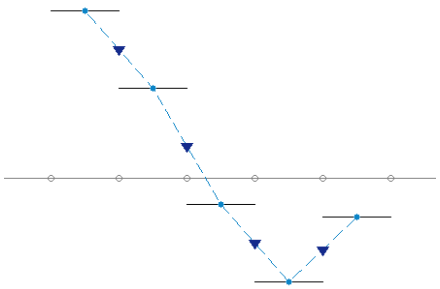
Gli stimatori a posteriori basati su un operatore di recovery differiscono per accuratezza ed efficienza, proprietà che dipendono dalla procedura di ricostruzione utilizzata. Rispettando le proprietà elencate nel paragrafo precedente è infatti possibile definire diversi operatori G_R.

Secondo quanto proposto nei lavori originali di Zienkiewicz e Zhu ([77, 78, 79]), la procedura generale per ottenere il gradiente ricostruito è piuttosto semplice. Nel caso di elementi finiti di grado uno, il valore del gradiente ricostruito nel nodo $x_p \in \mathcal{T}_h$ è dato dalla media pesata dei valori che il gradiente assume sul patch $\tilde{\Omega}_p$. Tale procedura è schematizzata in Figura 3.3 per il caso monodimensionale. Essa consta dei seguenti passi:

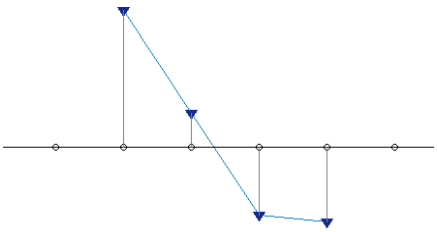
- si parte dalla soluzione approssimata u_h, lineare a tratti, data dal metodo agli elementi finiti (Fig. 3.3(a));
- si calcola il gradiente approssimato (costante a tratti); poi si assegna ad ogni nodo della griglia la media dei valori che ∇u_h assume nei due intervalli a cui appartiene il nodo (Fig. 3.3(b));



(a) Soluzione approssimata u_h



(b) Gradiente di u_h



(c) Gradiente ricostruito $G_R(u_h)$

Figura 3.3: Procedura di ricostruzione del gradiente nel caso monodimensionale

- i nuovi valori nodali vengono reinterpolati utilizzando le stesse funzioni di base di u_h producendo così il ricostruito $G_R(u_h)$; di conseguenza anch'esso sarà lineare a tratti (Fig. 3.3(c)).

Il primo tentativo di ricostruzione del gradiente effettuato da Zienkiewicz e Zhu in [77] è in realtà costituito da due approcci: uno globale, computazionalmente molto costoso, e uno locale, meno dispendioso.

La procedura di ricostruzione globale si basa sulla proiezione L^2 del gradiente della soluzione approssimata sullo spazio ad elementi finiti lineari V_h. Ovvero $G_R(u_h)$ è soluzione del seguente problema:

$$\int_\Omega G_R(u_h) \cdot \mathbf{v} d\mathbf{x} = \int_\Omega \nabla u_h \cdot \mathbf{v} d\mathbf{x}, \quad \forall \mathbf{v} \in V_h.$$

Indicando ancora con $\mathcal{N}$ l'insieme dei nodi della griglia $\mathcal{T}_h$ e con φ_p la funzione di base dello spazio ad elementi finiti associata al nodo x_p, si può scrivere

$$G_R(u_h)(x) = \sum_{x_p \in \mathcal{N}} G_R(u_h)(x_p)\varphi_p(x).$$

Allora i valori nodali $G_R(u_h)(x_p)$ sono soluzione di:

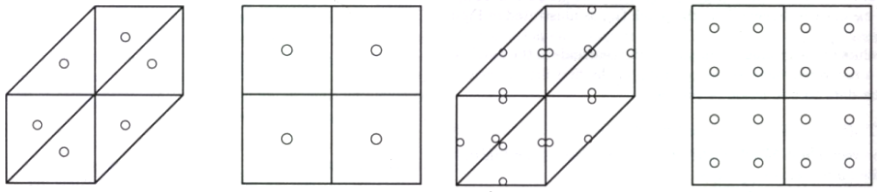
$$\sum_{x_p \in \mathcal{N}} \left(\int_\Omega \varphi_p \varphi_l d\mathbf{x} \right) G_R(u_h)(x_p) = \int_\Omega \nabla u_h \cdot \varphi_l d\mathbf{x}, \quad \forall l = 1, \ldots, N_h.$$

Calcolare $G_R(u_h)$ in questo modo risulta molto costoso: tale procedura infatti costa circa il doppio di quanto costi calcolare l'approssimazione u_h. Di conseguenza gli stessi autori propongono una procedura alternativa computazionalmente più conveniente. Si ricorre alla tecnica del *mass-lumping* al fine di ottenere esplicitamente l'espressione del gradiente ricostruito nei nodi. Si tratta di una procedura locale che approssima la matrice di massa $m_{pq} = \int_\Omega \varphi_p \varphi_q d\mathbf{x}$ con una matrice diagonale applicando,su ogni triangolo, la formula di quadratura dei trapezi. In questo modo si possono ricavare i valori nodali del gradiente ricostruito dati da:

$$G_R(u_h)(x_p) = \sum_{T \ni x_p} \frac{|T|}{|\tilde{\Omega}_p|} \nabla u_h|_T, \tag{3.22}$$

dove $|\cdot|$ indica l'area di un elemento. In pratica si calcola la media pesata (rispetto all'area) dei gradienti della soluzione discreta sul patch $\tilde{\Omega}_p$ associato al nodo p.

La procedura di ricostruzione locale del gradiente è nota invece come *superconvergent patch recovery (SPR)* [78, 79]. Se ne fa largo uso, soprattutto in ambito ingegneristico, grazie alla sua semplicità ed efficienza computazionale. Il valore di $G_R(u_h)$ nel nodo x_p viene ricostruito a partire dal valore dei gradienti di u_h nei baricentri $\mathbf{c}_K$ dei triangoli costituenti il patch



(a) $r = 1$ (b) $r = 2$

Figura 3.4: Punti di campionamento per il calcolo di $G_R(u_h)(x_p)$ usati nella ricostruzione SPR.

$\tilde{\Omega}_p$. Questo metodo può essere applicato a griglie composte da elementi sia triangolari sia quadrangolari, e anche utilizzando elementi finiti di grado superiore al primo (si veda la Figura 3.4).

Per semplicità nel seguito si considera il caso di elementi finiti lineari. Seguendo questo approccio $G_R(u_h)(x_p)$ si ottiene come soluzione di un problema locale ai minimi quadrati sul patch $\tilde{\Omega}_p$. Le componenti del gradiente ricostruito $G_R(u_h)(\mathbf{x}) = (G_R^1(u_h)(\mathbf{x}), G_R^2(u_h)(\mathbf{x}))$ vengono scritte nella forma:

$$G_R^j(u_h)(\mathbf{x}) = \mathbf{p}(\mathbf{x})^T \boldsymbol{\alpha}^j, \quad \text{con } j = 1, 2,$$

dove

$$\mathbf{p}(\mathbf{x}) = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

è il vettore della base dei polinomi $\mathbb{P}_1$, mentre $\boldsymbol{\alpha}^j$ è il vettore dei coefficienti incogniti del polinomio $G_R^j(u_h)$. Si calcola $\boldsymbol{\alpha}^j$ ricorrendo al metodo dei minimi quadrati, minimizzando la distanza tra il gradiente ricostruito e i valori discreti $\nabla u_h(\mathbf{c}_K)$:

$$\boldsymbol{\alpha}^j = \min \mathcal{I}(\boldsymbol{\alpha}^j) \quad \text{per } j = 1, 2,$$

dove

$$\mathcal{I}(\boldsymbol{\alpha}^j) = \sum_{K \subset \tilde{\Omega}_p} \left(\frac{\partial u_h}{\partial x_j}(\mathbf{c}_K) - \mathbf{p}(\mathbf{c}_K)^T \boldsymbol{\alpha}^j \right)^2.$$

Quindi $\boldsymbol{\alpha}^j$ è soluzione del sistema $M\boldsymbol{\alpha} = \mathbf{b}$ dove M è la matrice

$$M = \sum_{K \subset \tilde{\Omega}_p} \mathbf{p}(\mathbf{c}_K) \mathbf{p}(\mathbf{c}_K)^T$$

e $\mathbf{b}$ il vettore le cui componenti sono definite da:

$$\mathbf{b}^j = \sum_{K \subset \tilde{\Omega}_p} \mathbf{p}(\mathbf{c}_K)^T \frac{\partial u_h}{\partial x_j}(\mathbf{c}_K).$$

A partire dai primi lavori di Zienkiewicz e Zhu sono state poi proposte in letteratura diverse tecniche alternative per la ricostruzione del gradiente, tutte guidate da esigenze specifiche intrinseche del problema in esame. Le varie soluzioni proposte possono spesso essere scritte nella seguente forma generica:

$$G_R(u_h)(x_p) = \left(\sum_{T \ni x_p} \omega_T \right)^{-1} \sum_{T \ni x_p} \omega_T \nabla u_h|_T,$$

dove gli ω_T sono dei pesi opportuni. Si possono effettuare diverse scelte per tali pesi, tra cui ricordiamo le seguenti:

- $\omega_T = |T|$ area del triangolo T: si ottiene la formulazione (3.22);
- $\omega_T = \dfrac{1}{\#\tilde{\Omega}_p}$, dove $\#\tilde{\Omega}_p$ denota la cardinalità del patch ([62]);
- $\omega_T = \|x_p - \mathbf{c}_T\|^{-1}$ distanza dal baricentro dell'elemento T ([14]).

3.3.3 Proprietà

Si consideri l'espressione della norma dell'energia dell'errore di discretizzazione (3.5); si osserva che tale norma potrebbe essere calcolata esattamente se si conoscesse il valore del gradiente della soluzione esatta ∇u. Visto che solitamente ciò non accade, l'idea è quella di sostituire a ∇u una sua opportuna ricostruzione $G_R(u_h)$. Lo stimatore a posteriori della quantità in (3.5) è dato dunque da:

$$\eta^2 = \int_\Omega |G_R(u_h) - \nabla u_h|^2 d\mathbf{x} = \sum_{K \in \mathcal{T}_h} \|G_R(u_h) - \nabla u_h\|^2_{L^2(K)}. \tag{3.23}$$

Si nota che è possibile stimare unicamente la norma in energia dell'errore di discretizzazione, e non altre quantità di interesse, come garantito ad esempio dagli stimatori goal-oriented.

Se vale la proprietà di superconvergenza, si dice che lo stimatore a posteriori (3.23) è **asintoticamente esatto**. Si dimostra infatti il seguente

Teorema 3.5

Sotto le ipotesi (3.18)-(3.19) e (3.20) si ha:

$$\lim_{h \to 0} \frac{\eta}{|||e_h|||} = 1,$$

dove il rapporto $\frac{\eta}{|||e_h|||}$ è l'effectivity index θ associato allo stimatore η definito in (3.6).

L'effectivity index dunque tende a 1, valore ovviamente ottimale per un qualsiasi stimatore.

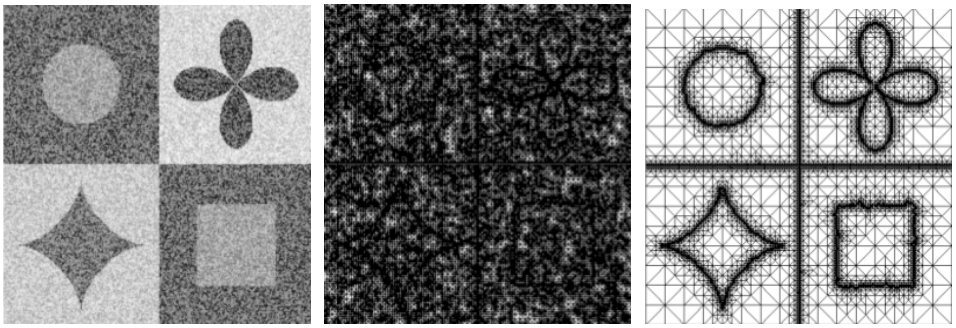
Altri vantaggi di questi stimatori sono la facilità di implementazione e il ridotto costo computazionale; inoltre, a differenza degli stimatori visti nella Sezione 3.2.2, gli stimatori recovery-based sono indipendenti dal problema in esame. Per queste ragioni saranno usati nei capitoli che seguono per adattare la griglia durante la risoluzione dell'equazione differenziale (2.19) dell'Algoritmo 3 di Split-Bregman.

Esiste anche un'interessante e pratica versione anisotropa di tale stimatore, che sarà trattata nel prossimo capitolo.

3.4 Adattività di griglia nel trattamento delle immagini

Il metodo agli elementi finiti è raramente utilizzato nell'ambito del trattamento delle immagini. Infatti la versione discreta delle immagini sembra ideale per ricorrere ad una discretizzazione alle differenze finite, utilizzando come griglia quella individuata dai pixel stessi dell'immagine, introdotta nella Sezione 1.1.

Una tecnica in controtendenza, e quindi di particolare interesse in questo campo, è la strategia di deraffinamento proposta da E. Bänsch e K. Mikula in [17], applicata ad un problema di denoising (si veda la Fig. 3.5). Si tratta infatti di uno dei rari casi, nell'ambito del trattamento delle immagini, in cui una procedura di adattazione di griglia viene effettivamente affiancata alla risoluzione di un problema differenziale. La tecnica di adattazione qui usata è del tutto analoga a quelle euristiche definite nella Sezione 3.1.2. Il problema consiste nel risolvere l'equazione differenziale di Catté et al. per il denoising di immagini ([19]), con l'obiettivo di ricostruire l'immagine originale a partire dalla sua versione sporcata con un rumore gaussiano.



(a) Immagine *noisy* (b) Griglia iniziale $\mathcal{T}_h^0$ (c) Griglia finale

Figura 3.5: Risultato della procedura di adattazione di griglia in [17].

Non entriamo nei dettagli della procedura di *coarsening* della griglia, non essendo tra gli obiettivi di questo lavoro; ci soffermiamo però sul criterio di arresto utilizzato. Data la triangolazione $\mathcal{T}_h^i$ ottenuta al passo i-esimo della procedura adattativa, viene fissato un valore di tolleranza ϵ e si sceglie di deraffinare unicamente i triangoli $T \in \mathcal{T}_h^i$ per cui valga

$$h_T |\nabla u_h^i|_T| \leq \epsilon,$$

dove u_h^i indica la soluzione approssimata del problema differenziale associata al passo i della procedura adattativa (ovvero calcolata sulla griglia $\mathcal{T}_h^i$). Tale condizione è tipica proprio di una tecnica di adattazione euristica: l'intensità del gradiente della soluzione guida le modifiche della griglia.

Il ruolo del gradiente è fondamentale in questo frangente; infatti, agendo da edge-detector, raffina la griglia proprio in corrispondenza dei contorni dell'immagine. I considerevoli vantaggi pratici di tale impostazione potrebbero rappresentare una svolta per la riduzione dell'elevato costo computazionale necessario per trattare grandi quantità di immagini.

Esistono inoltre alcuni filoni nello studio delle immagini il cui interesse è indirizzato proprio verso la creazione di griglie strutturate atte a ricostruire le immagini, sia in due sia in tre dimensioni.

In questo ambito, gli approcci seguiti per costruire una griglia adattata sono essenzialmente due. Una prima alternativa consiste nel partire da una griglia molto fine dove i nodi della mesh corrispondono ai pixel dell'immagine, per poi renderla grossolana nelle aree di minore variazione della soluzione. Come seconda alternativa si utilizza una griglia iniziale lasca che si raffina via via fino a raggiungere la precisione voluta. Questo secondo approccio è ovviamente quello più consistente con le tecniche di adattazione di griglia guidate dagli stimatori dell'errore trattate nelle sezioni precedenti.
Nelle procedure di generazione di griglia per la ricostruzione delle immagini si deve tener conto anche di un altro interessante aspetto: l'immagine di partenza è conosciuta, non è un'incognita del problema! Per questo motivo sono molto inflazionati i metodi che minimizzano la differenza tra l'immagine originale e quella ricostruita attraverso un'interpolazione nei nodi della mesh [40, 41, 27]. In Figura 3.6 si può apprezzare il risultato della costruzione della griglia relativa ad un'immagine, tratta da [27]. Tra i numerosi metodi che seguono l'approccio inverso, procedendo con il deraffinamento di una griglia fine, citiamo a titolo di esempio il lavoro di Ciampalini [24].

(a) Immagine da ricostruire (b) Griglia finale

Figura 3.6: Risultato della procedura di costruzione della griglia in [27].

Probabilmente la scarsa notorietà di cui godono i metodi di adattività di griglia nell'ambito del trattamento delle immagini è dovuta alla loro complessità dal punto di vista sia teorico che implementativo. Vediamo nei prossimi capitoli come tale difficoltà sia però ricompensata da buoni risultati pratici.

Capitolo 4

Adattazione anisotropa di griglia

Questo capitolo si focalizza sull'analisi delle griglie anisotrope, ed in particolare sulla costruzione di tali griglie attraverso procedure di adattazione analoghe a quelle viste nel Capitolo 3. La Sezione 4.2 è dedicata all'introduzione di alcune proprietà fondamentali delle griglie anisotrope, che servono da cornice per le stime a priori e a posteriori presentate nelle Sezioni 4.3 e 4.4. Si presta poi particolare attenzione, nella Sezione 4.5, alla descrizione di un particolare stimatore che generalizza ad un contesto anisotropo uno stimatore recovery-based come quello proposto da Zienkiewicz e Zhu. Infine nella Sezione 4.6 si descrive una procedura pratica di adattazione di griglia basata su tale stimatore.

4.1 Anisotropia

L'anisotropia, per definizione, è la proprietà per la quale un determinato oggetto, materiale o proprietà fisica ha caratteristiche che dipendono dalla direzione lungo la quale vengono considerate. Si può dire che l'anisotropia rappresenta per la direzione quello che la disomogeneità rappresenta per lo spazio. Un materiale è infatti anisotropo se le sue caratteristiche fisiche (conducibilità elettrica o termica, proprietà ottiche) o il suo comportamento meccanico (rigidezza, resistenza, tenacità) sono differenti in direzione longitudinale e trasversale. In particolare, i cristalli mostrano anisotropia per almeno una proprietà fisica, come conseguenza della disposizione ordinata e periodica degli atomi.

In matematica applicata tale concetto può essere associato a diverse entità: può essere anisotropo un modello, così come una funzione o una griglia di calcolo. L'interesse per questo tipo di problemi deriva dalla complessità della realtà fisica. Risulta evidente che l'ipotesi di isotropia per la modellazione dei fenomeni naturali è spesso troppo riduttiva; infatti molti

problemi fisici sono caratterizzati da soluzioni con forti variazioni locali. È questo ad esempio il caso della maggior parte dei problemi di diffusione-trasporto dove si possono sviluppare forti strati limite, ovvero regioni in cui la soluzione è caratterizzata da brusche variazioni del gradiente. In questi casi può essere vantaggioso ricorrere ad un'approssimazione ad elementi finiti sulle cosiddette griglie anisotrope, dove gli elementi sono allineati in modo da seguire la direzionalità della soluzione. A differenza delle griglie isotrope infatti, le griglie anisotrope consentono non solo di modulare la dimensione degli elementi della griglia, ma anche di modificarne forma e orientamento al fine di poter seguire meglio l'andamento della soluzione approssimata.

L'utilizzo di griglie anisotrope può essere interessante anche per la risoluzione dei problemi di segmentazione delle immagini descritti nei Capitoli 1 e 2. Infatti abbiamo visto che le soluzioni di questi problemi, cioè le funzioni di level set che descrivono il contorno delle immagini, variano bruscamente proprio nell'intorno di tali contorni. La griglia anisotropa può aiutare dunque ad identificare meglio la loro posizione e possibilmente ad avere una riproduzione più regolare del contorno stesso.

4.2 Il contesto anisotropo

Innanzitutto introduciamo il contesto anisotropo a cui facciamo riferimento, ovvero quello proposto in [36]. A questo scopo si consideri una triangolazione conforme $\mathcal{T}_h$ del dominio Ω, composta da triangoli K di diametro $h_K \leq h$. Sia inoltre $T_K : \hat{K} \to K$ la mappa standard affine e invertibile che trasforma il triangolo di riferimento $\hat{K}$ nell'elemento generico K della mesh, come mostrato in Figura 4.1. I risultati che seguono sono indipendenti dalla scelta del triangolo di riferimento; quindi, per como-

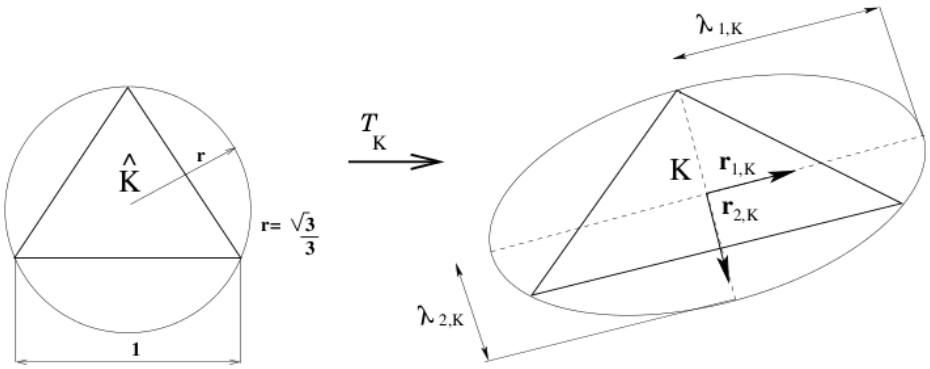


Figura 4.1: Trasformazione affine

dità, scegliamo $\hat{K}$ come il triangolo equilatero di lato unitario e vertici in: $(-1/2, 0), (1/2, 0), (0, \sqrt{3}/2)$.
La mappa affine T_K è definita dalla matrice $M_K \in \mathbb{R}^{2x2}$ e dal vettore $\mathbf{t}_K \in \mathbb{R}^2$ nel modo seguente:

$$\forall \hat{\mathbf{x}} \in \hat{K} \quad \mathbf{x} = T_K(\hat{\mathbf{x}}) = M_K \hat{\mathbf{x}} + \mathbf{t}_K, \quad \text{con } \mathbf{x} \in K.$$

Per dedurre le informazioni su dimensioni, forma e orientamento del triangolo K si sfruttano le proprietà spettrali della mappa T_K. Sia infatti $M_K = B_K Z_K$ la decomposizione polare della matrice M_K, dove $B_K, Z_K \in \mathbb{R}^{2x2}$ sono due matrici, rispettivamente, simmetrica definita positiva e ortonormale. La matrice B_K si può diagonalizzare in funzione dei suoi autovettori $\mathbf{r}_{i,K}$ e autovalori $\lambda_{i,K}$ (con $i = 1, 2$): $B_K = R_K^T \Lambda_K R_K$, dove $R_K = [\mathbf{r}_{1,K}, \mathbf{r}_{2,K}]^T$ e $\Lambda_K = \text{diag}(\lambda_{1,K}, \lambda_{2,K})$. Si assume nel seguito $\lambda_{1,K} \geq \lambda_{2,K}$ senza alcuna perdita di generalità.

Come mostrato nella Figura 4.1, il cerchio circoscritto a $\hat{K}$ viene deformato dalla mappa T_K in un'ellisse circoscritta a K. Ora l'orientamento del triangolo K è univocamente identificato dagli autovettori della matrice B_K, così come le sue dimensioni e la sua deformazione dipendono dal valore degli autovalori $\lambda_{i,K}$, che forniscono la misura dei semiassi dell'ellisse. La deformazione di un generico triangolo K è solitamente misurata in termini del cosiddetto *stretching factor*:

$$s_K = \frac{\lambda_{1,K}}{\lambda_{2,K}} \quad (\geq 1).$$

Una griglia isotropa $\mathcal{T}_h$ è caratterizzata da avere $\lambda_{1,K} = \lambda_{2,K}$, ovvero da $s_K = 1, \quad \forall K \in \mathcal{T}_h$.

Indichiamo con h_K e $h_{\hat{K}}$ i diametri dei triangoli K e $\hat{K}$, mentre con ρ_K e $\rho_{\hat{K}}$ i diametri dei cerchi inscritti a tali triangoli. Il rapporto tra le dimensioni dei due triangoli può essere stimato grazie alle seguenti relazioni:

$$\lambda_{2,K} h_{\hat{K}} \leq h_K \leq \lambda_{1,K} h_{\hat{K}}, \quad \lambda_{2,K} \rho_{\hat{K}} \leq \rho_K \leq \lambda_{1,K} \rho_{\hat{K}},$$

da cui si deduce che

$$\frac{h_{\hat{K}}}{\rho_{\hat{K}}} \frac{\lambda_{2,K}}{\lambda_{1,K}} \leq \frac{h_K}{\rho_K} \leq \frac{h_{\hat{K}}}{\rho_{\hat{K}}} \frac{\lambda_{1,K}}{\lambda_{2,K}}.$$

Partendo da questa caratterizzazione di una griglia anisotropa e in vista della procedura di adattazione di griglia, procediamo con la formulazione di opportuni stimatori anisotropi a priori e a posteriori.

4.3 Stime d'interpolazione anisotrope

In [36] L. Formaggia e S. Perotto derivano delle stime anisotrope per l'errore di interpolazione basandosi sul setting presentato nella sezione precedente. Le informazioni spettrali della mappa affine permettono infatti di

scomporre i contributi dell'errore associati a direzioni differenti. A partire da tali stime di interpolazione è possibile derivare uno stimatore anisotropo a posteriori goal-oriented, come vedremo nella prossima Sezione. Si noti che nel seguito del capitolo ci si limita a considerare problemi in due dimensioni discretizzati con elementi finiti lineari.

In [36] gli autori derivano stime d'interpolazione anisotrope sia per l'operatore d'interpolazione di Lagrange, sia per il meno noto operatore di quasi interpolazione di Clément ([25]); ai nostri fini questo ultimo operatore giocherà un ruolo dominante. Tuttavia, per completezza, forniamo un esempio di stima d'interpolazione anisotropa anche per l'operatore Lagrangiano nella seguente

Proposizione 4.1

Sia $v \in H^2(K)$ e $\hat{v} \in H^2(\hat{K})$ la funzione corrispondente associata al triangolo $\hat{K}$ di referenza. Sia inoltre $\Pi^1_K(v)$ l'interpolante lineare di Lagrange di v definito su K e sia e uno dei tre lati di K. Allora esistono due costanti $C_1 = C_1(\hat{K}, \Pi^1_{\hat{K}})$ e $C_2(\hat{K}, \Pi^1_{\hat{K}})$ tali che

$$\begin{aligned} \|v - \Pi^1_K(v)\|_{L^2(K)} \leq C_1 \Big[& \lambda^4_{1,K} L_K(\mathbf{r}_{1,K}, \mathbf{r}_{1,K}; v) \\ & + \lambda^4_{2,K} L_K(\mathbf{r}_{2,K}, \mathbf{r}_{2,K}; v) + 2\lambda^2_{1,K}\lambda^2_{2,K} L_K(\mathbf{r}_{1,K}, \mathbf{r}_{2,K}; v) \Big]^{1/2}, \end{aligned} \tag{4.1}$$

$$\begin{aligned} \|v - \Pi^1_K(v)\|_{L^2(e)} \leq C_1 \left(\frac{\lambda^2_{1,K} + \lambda^2_{2,K}}{\lambda_{2,K}} \right)^{1/2} \Big[& \frac{\lambda^4_{1,K}}{\lambda^2_{2,K}} L_K(\mathbf{r}_{1,K}, \mathbf{r}_{1,K}; v) \\ & + \lambda^2_{2,K} L_K(\mathbf{r}_{2,K}, \mathbf{r}_{2,K}; v) + 2\lambda^2_{1,K} L_K(\mathbf{r}_{1,K}, \mathbf{r}_{2,K}; v) \Big]^{1/2}, \end{aligned}$$

dove

$$L_K(\mathbf{r}_{i,K}, \mathbf{r}_{j,K}; v) = \int_K (\mathbf{r}^T_{i,K} H_K(v) \mathbf{r}_{j,K})^2 d\mathbf{x} \quad \text{con } i, j = 1, 2.$$

In questa espressione $H_K(v)$ denota la matrice Hessiana associata alla funzione v.

La natura anisotropa di tali stime di interpolazione è rappresentata dalla possibilità di controllare separatamente $\lambda_{1,K}$ e $\lambda_{2,K}$ e le corrispondenti direzioni $\mathbf{r}_{1,K}$ e $\mathbf{r}_{2,K}$. Infatti la quantità definita da $L_K(\mathbf{r}_{i,K}, \mathbf{r}_{j,K}; v)$ altro non è che la proiezione della seminorma $|v|_{H^2(K)}$ lungo le direzioni anisotrope $r_{i,K}$ e $r_{j,K}$. Il risultato della Proposizione 4.1 può essere ridotto alla classica stima di interpolazione isotropa semplicemente ponendo $\lambda_{1,K} = \lambda_{2,K}$.

La stima (4.1) gode di una proprietà molto interessante in vista del suo utilizzo per la generazione di mesh adattate. Infatti non deve rispettare la cosiddetta *maximal angle condition*: in [7] Babuska e Aziz mostrano che una griglia di elementi finiti deve rispettare questa condizione, che richiede

che gli angoli di ciascun elemento non tendano a π. Questo limite sulla costruzione della mesh non è ovviamente compatibile con l'utilizzo di una procedura automatica di adattazione di griglia: è necessario che le stime dell'errore mostrino un comportamento corretto quando l'angolo massimo tende a π. In particolare, il reciproco dell'effectivity index definito in (3.6) deve essere limitato. In [36] si dimostra che la stima anisotropa (4.1) gode di questa proprietà, almeno in alcuni casi altrimenti considerati critici.

Per generalizzare questi risultati a funzioni solamente in $H^1(\Omega)$ bisogna sostituire all'interpolante di Lagrange un operatore più generale; infatti nel caso bidimensionale non è più vero che $H^1(\Omega) \hookrightarrow C^0(\Omega)$. Un operatore adatto a tale scopo è l'operatore di quasi interpolazione proposto da Clément in [25] che può essere definito per funzioni non necessariamente continue.

Sia quindi $I_h^1 : L^2(\Omega) \to V_h$ l'interpolante lineare di Clément tale che, per ogni $K \in \mathcal{T}_h$ e ogni lato e di K, valgono le seguenti relazioni:

$$\|v - I_h^1(v)\|_{L^2(K)} \leq h_K |v|_{H^1(\tilde{K})} \quad \text{e} \quad \|v - I_h^1(v)\|_{L^2(e)} \leq h_e^{1/2} |v|_{H^1(\tilde{e})}$$

dove $\tilde{K}$ e $\tilde{e}$ sono i patch di elementi rispettivamente associati al triangolo K e al lato e. Sia inoltre I_K^1 la restrizione dell'operatore I_h^1 sull'elemento K, per ogni $K \in \mathcal{T}_h$. Prima di fornire un esempio di stima d'interpolazione anisotropa per tale operatore, supponiamo che la cardinalità del patch $\tilde{K}$ e

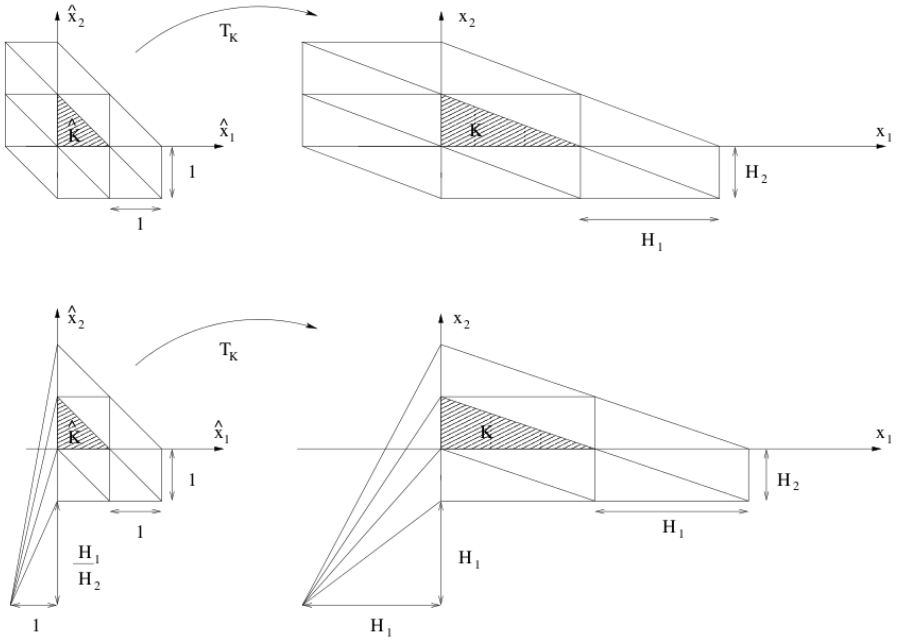


Figura 4.2: Esempio di un patch accettabile (in alto) e non accettabile (in basso) per le stime anisotrope della Proposizione 4.2.

il diametro del patch di riferimento $\hat{\tilde{K}} = T_K^{-1}(\tilde{K})$ siano limitati, ovvero che

$$\text{card}(\tilde{K}) < N \quad \text{e} \quad \text{diam}(\hat{\tilde{K}}) \leq C_\Delta \simeq O(1) \tag{4.2}$$

con $C_\Delta \geq h_{\hat{K}}$. Si veda la Figura 4.2 per esempi di patch accettabili e non accettabili, ovvero che rispecchiano o non rispecchiano le ipotesi (4.2).

Sotto queste ipotesi vale la seguente

Proposizione 4.2

Sia $v \in H^1(\Omega)$. Allora, per ogni $K \in \mathcal{T}_h$, esistono due costanti $C_1 = C_1(N, C_\Delta)$ e $C_2 = C_2(N, C_\Delta)$ tali che:

$$\|v - I_K^1(v)\|_{L^2(K)} \leq C_1 \left[\lambda_{1,K}^2(\mathbf{r}_{1,K}^T G_K(v)\mathbf{r}_{1,K}) + \lambda_{2,K}^2(\mathbf{r}_{2,K}^T G_K(v)\mathbf{r}_{2,K})\right]^{1/2} \tag{4.3}$$

$$\|v - I_K^1(v)\|_{L^2(\partial K)} \leq C_2 h_K^{1/2} \left[s_K(\mathbf{r}_{1,K}^T G_K(v)\mathbf{r}_{1,K}) + s_K^{-1}(\mathbf{r}_{2,K}^T G_K(v)\mathbf{r}_{2,K})\right]^{1/2}$$

dove $G_K(v)$ è la matrice simmetrica semi-definita positiva data da:

$$G_K(v) = \sum_{T \in \tilde{K}} \begin{bmatrix} \int_T \left(\frac{\partial v}{\partial x_1}\right)^2 d\mathbf{x} & \int_T \frac{\partial v}{\partial x_1}\frac{\partial v}{\partial x_2} d\mathbf{x} \\ \int_T \frac{\partial v}{\partial x_1}\frac{\partial v}{\partial x_2} d\mathbf{x} & \int_T \left(\frac{\partial v}{\partial x_2}\right)^2 d\mathbf{x} \end{bmatrix}. \tag{4.4}$$

Per la dimostrazione di questi risultati rimandiamo a [36, 37].

4.4 Stime anisotrope a posteriori

Le stime a priori appena mostrate permettono di ricavare delle stime anisotrope a posteriori. In [37] Formaggia e Perotto studiano un generico problema ellittico analogo a (3.1), dove però la frontiera è ridotta alla sola frontiera di Dirichlet, ossia $\Gamma_N = \emptyset$.

In questo contesto gli autori ricavano sia una stima a priori per la norma dell'energia dell'errore di discretizzazione, sia una stima a posteriori per il problema duale (3.13). Nonostante non si tratti di un risultato rilevante ai fini di questo lavoro, in analogia a quanto fatto nel Capitolo 3, riportiamo tale stima a posteriori.

Innanzitutto ridefiniamo il residuo interno e di bordo scrivendoli in una formulazione locale. Per ogni K in $\mathcal{T}_h$ (ricordando che $e \in \partial K$ indica ognuno dei suoi tre lati), siano

$$r_K(u_h) = (f + \triangle u_h)|_K$$

e

$$R_K(u_h) = \begin{cases} 0, & \forall e \in \partial K \cap \Gamma_D \\ 2\left(g - \dfrac{\partial u_h}{\partial \mathbf{n}_K}\right)\Big|_e & \forall e \in \partial K \cap \Gamma_N \\ -\left[\dfrac{\partial u_h}{\partial \mathbf{n}_K}\right]_e & \forall e \in (\partial K \setminus \partial\Omega). \end{cases}$$

La definizione del salto della derivata conormale $\left[\dfrac{\partial u_h}{\partial \mathbf{n}_K}\right]_e$ è analoga a quella data in (3.9).

Proposizione 4.3

Sia u la soluzione del problema (3.1) e sia u_h la corrispondente approssimazione agli elementi finiti. Sia inoltre e_h il corrispondente errore di discretizzazione. Allora, se la soluzione ϕ del problema duale 3.13 appartiene allo spazio $H^2(\Omega)$, vale la seguente stima:

$$|Q(e_h)| \lesssim \sum_{K \in \mathcal{T}_h} \rho_K(u_h)\omega_K(\phi),$$

dove

$$\rho_K(u_h) = \|r_K(u_h)\|_{L^2(K)} + \frac{1}{2\lambda_{2,K}^{1/2}}\|R_K(u_h)\|_{L^2(e)},$$

e

$$\begin{aligned}\omega_K(\phi) =& (\lambda_{1,K}^2 + \lambda_{2,K}^2)^{1/2}\left[\frac{\lambda_{1,K}^4}{\lambda_{2,K}^2} L_K(\mathbf{r}_{1,K}, \mathbf{r}_{1,K}; v)\right. \\ &\left. + \lambda_{2,K}^2 L_K(\mathbf{r}_{2,K}, \mathbf{r}_{2,K}; v) + 2\lambda_{1,K}^2 L_K(\mathbf{r}_{1,K}, \mathbf{r}_{2,K}; v)\right]^{1/2}.\end{aligned}$$

4.5 Stimatore ZZ anisotropo

S. Micheletti e S. Perotto in [52] ricavano una versione anisotropa dello stimatore recovery-based di Zienkiewicz e Zhu a partire dalle stime di interpolazione (4.3). Analogamente a quanto fatto nella Sezione 3.3, procediamo per passi: innanzitutto descriviamo una procedura per ottenere il gradiente ricostruito, poi usiamo questa quantità per ricavare uno stimatore a posteriori.

Considerando ancora il problema modello (3.1), sia u_h l'approssimazione di Galerkin della soluzione esatta u. Il gradiente ricostruito $G^r_{\tilde{K}}(u_h)$ proposto dagli autori in [52] è di grado r sul patch $\tilde{K}$. Si noti che, nonostante la notazione, tale quantità è strettamente associata al triangolo K, e non

agli elementi del patch $\tilde{K}$. Si cerca allora $G^r_{\tilde{K}}(u_h) \in [\mathbb{P}_r]^2$ tale che

$$\int_{\tilde{K}} (\nabla u_h - G^r_{\tilde{K}}(u_h)) \mathbf{w} d\mathbf{x} = 0 \quad \forall \mathbf{w} \in [\mathbb{P}_r]^2, \tag{4.5}$$

dove $\mathbb{P}_r$ è lo spazio dei polinomi di grado totale inferiore o uguale a r. Nel caso particolare in cui $r = 0$, si può ricavare esplicitamente l'espressione del gradiente ricostruito da (4.5), ottenendo:

$$G^0_{\tilde{K}}(u_h) = \frac{1}{|\tilde{K}|} \sum_{T \in \tilde{K}} |T| \nabla u_h|_T.$$

Per semplicità di notazioni nel seguito consideriamo il caso $r = 0$ e omettiamo il corrispondente apice. Si noti che abbiamo ritrovato l'espressione del gradiente ricostruito definito in (3.22).

Sia ora $e^*_{\tilde{K}} = G_{\tilde{K}}(u_h) - \nabla u_h|_{\tilde{K}}$, approssimazione dell'errore sul gradiente della soluzione calcolata nel patch $\tilde{K}$. Lo stimatore anisotropo locale per la seminorma H^1 dell'errore di discretizzazione è definito in [52] come segue:

$$\eta^2_K = \frac{1}{\lambda_{1,K}\lambda_{2,K}} \sum_{i=1}^{2} \lambda^2_{i,K} (\mathbf{r}^T_{i,K} G_K(e^*_{\tilde{K}}) \mathbf{r}_{i,K}), \tag{4.6}$$

dove la matrice G_K è quella in (4.4). Il corrispondente stimatore globale dell'errore è dato da

$$\eta = \left(\sum_{K \in \mathcal{T}_h} \eta^2_K \right)^{1/2}. \tag{4.7}$$

Lo stimatore a posteriori (4.6)-(4.7) è essenzialmente euristico. Infatti i termini della sommatoria in (4.6) sono suggeriti dalla stima (4.3), prendendo $v = u - u_h$ e sostituendo alle derivate parziali di u le componenti di $G_{\tilde{K}}(u_h)$ corrispondenti.

Il ruolo del fattore di scala $\lambda_{1,K}\lambda_{2,K}$ in (4.6) è quello di garantire la consistenza rispetto al caso isotropo. Infatti, ponendo $\lambda_{1,K} = \lambda_{2,K}$ si ritrova esattamente lo stimatore isotropo di Zienkiewicz-Zhu.

Questo stimatore condivide gli stessi vantaggi di ridotto costo computazionale dello stimatore ZZ originale; le sue proprietà di efficienza e affidabilità sono studiate in [51]. Inoltre questo risultato è stato esteso al caso di problemi tridimensionali in [35].

4.6 Procedura di adattazione anisotropa

In questa sezione si fornisce una procedura pratica per adattare in modo anisotropo la griglia di calcolo basandosi sugli stimatori (4.6)-(4.7). In questo frangente, vogliamo costruire la mesh con il minor numero di elementi che garantisca una tolleranza data sull'accuratezza della soluzione approssimata.

Si usa un approccio basato sul concetto di **metrica** [39]. Ricordiamo che una metrica è un campo tensoriale $\tilde{M} : \Omega \to \mathbb{R}^{2\times 2}$ simmetrico definito positivo. In particolare, per ogni mesh $\mathcal{T}_h$ si può definire una metrica costante a tratti $\tilde{M}_{\mathcal{T}_h}$ tale che $\tilde{M}_{\mathcal{T}_h}|K = \tilde{M}_K = B_K^{-2} = R_K^T \Lambda_K^{-2} R_K$ per ogni $K \in \mathcal{T}_h$, dove le matrici B_K e Λ_K sono esattamente quelle definite in Sezione 4.2. Osserviamo che, rispetto a questa metrica, ogni triangolo K è equilatero e la lunghezza dei suoi lati è unitaria.

Viceversa, mostriamo come ad ogni metrica $\tilde{M}$ è associabile una mesh ottimale attraverso una cosiddetta *matching condition*. A tal fine diagonalizziamo il campo tensoriale: $\tilde{M} = \tilde{R}^T \tilde{\Lambda}^{-2} \tilde{R}$ con $\tilde{\Lambda}$ =diag$(\tilde{\lambda}_1, \tilde{\lambda}_2)$ matrice diagonale positiva e $\tilde{R}^T = [\tilde{\mathbf{r}}_1, \tilde{\mathbf{r}}_2]$ matrice ortogonale. Per ragioni pratiche approssimiamo le quantità $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mathbf{r}}_1$ e $\tilde{\mathbf{r}}_2$ definendo $\tilde{M}$ con delle funzioni costanti a tratti sulla triangolazione $\mathcal{T}_h$, tali che $\tilde{\mathbf{r}}_i|_K = \tilde{\mathbf{r}}_{i,K}$ e $\tilde{\lambda}_i|_K = \tilde{\lambda}_{i,K}$ per ogni $K \in \mathcal{T}_h$ e con $i = 1, 2$. Introduciamo la *matching condition*:

Definizione: *Si dice che una mesh $\mathcal{T}_h$ corrisponde ad una data metrica $\tilde{M}$ se, per ogni $K \in \mathcal{T}_h$ vale:*

$$\tilde{M}|_K = \tilde{M}_{\mathcal{T}_h}|_K \;\; \textit{cioè} \;\; \tilde{\mathbf{r}}_{i,K} = \mathbf{r}_{i,K},\; \tilde{\lambda}_{i,K} = \lambda_{i,K} \;\; \textit{per } i = 1, 2.$$

L'incognita del problema è proprio $\tilde{M}$ e ad ogni iterazione j bisogna considerare tre quantità: la mesh attuale $\mathcal{T}_h^{(j)}$, la nuova metrica $\tilde{M}^{(j+1)}$ calcolata su $\mathcal{T}_h^{(j)}$ e la mesh aggiornata $\mathcal{T}_h^{(j+1)}$ che corrisponde a $\tilde{M}^{(j+1)}$ nel senso della *matching condition*.

La nuova metrica si ottiene prima risolvendo un problema di minimizzazione rispetto agli autovettori $\mathbf{r}_{i,K}$ rappresentanti l'orientazione spaziale di K, poi calcolando i valori dei $\lambda_{i,K}$ tramite un criterio di equidistribuzione dell'errore.

Il problema di ottimizzazione si risolve osservando che minimizzare il numero di elementi corrisponde a massimizzare la loro area, mantenendo così fisso il valore dello stimatore locale η_K definito in (4.6). Quest'ultimo può essere più praticamente riscritto nel seguente modo:

$$\eta_K^2 = \lambda_{1,K}\lambda_{2,K}|\hat{\tilde{K}}| \left[s_K (\mathbf{r}_{1,K}^T \hat{G}_K(e_h^*)\mathbf{r}_{1,K}) + s_K^{-1} (\mathbf{r}_{2,K}^T \hat{G}_K(e_h^*)\mathbf{r}_{2,K}) \right], \tag{4.8}$$

dove $\tilde{G}_K(e_h^*) = \dfrac{G_K(e_h^*)}{|\tilde{K}|}$ e $|\tilde{K}| = \lambda_{1,K}\lambda_{2,K}|\hat{\tilde{K}}|$ è l'area del patch associato al triangolo K. Tale scalatura è stata effettuata con lo scopo di rendere i termini in (4.8) approssimativamente indipendenti dalle dimensioni del triangolo K.

Si cerca quindi un minimo del funzionale

$$J(s_K, \mathbf{r}_{1,K}, \mathbf{r}_{2,K}) = s_K (\mathbf{r}_{1,K}^T \tilde{G}_K(e_h^*)\mathbf{r}_{1,K}) + \frac{1}{s_K} (\mathbf{r}_{2,K}^T \tilde{G}_K(e_h^*)\mathbf{r}_{2,K}). \tag{4.9}$$

La soluzione di questo problema di minimizzazione è espressa nella seguente proposizione.

Proposizione 4.4

Il minimo di (4.9) si ottiene con:

$$s_K = \sqrt{g_1/g_2}, \quad \mathbf{r}_{1,K} = \mathbf{g}_2, \quad \mathbf{r}_{2,K} = \mathbf{g}_1$$

dove $\{g_i, \mathbf{g}_i\}_{i=1,2}$ *sono le coppie di autovalori e autovettori associate alla matrice* $\tilde{G}_K(e_h^*)$*, con* $g_1 \geq g_2 \geq 0$.

Si noti che i valori ottimali forniti dalla Proposizione 4.4 sono tali da rendere uguali i due addendi in (4.9). Infatti $s_K g_2 = s_K^{-1} g_1 = \sqrt{g_1 g_2}$. Ciò significa che il minimo di $j(\cdot)$ non dipende da s_K. Per definire completamente la metrica, restano da calcolare i valori di $\lambda_{1,K}$ e $\lambda_{2,K}$. A tale fine, imponiamo $\eta_K = \tau, \ \forall K \in \mathcal{T}_h$ dove τ è una tolleranza data. Si trova che $\lambda_{1,K}$ e $\lambda_{2,K}$ sono date dalle espressioni

$$\lambda_{1,K} = g_2^{-1/2} \left(\frac{\tau^2}{2\text{card}(\mathcal{T}_h)|\hat{\hat{K}}|} \right)^{1/2}, \quad \lambda_{2,K} = g_1^{-1/2} \left(\frac{\tau^2}{2\text{card}(\mathcal{T}_h)|\hat{\hat{K}}|} \right)^{1/2}. \tag{4.10}$$

La nuova metrica $\tilde{M}^{(j+1)}$ si ottiene, elemento per elemento, dalla relazione

$$\tilde{M}_K^{(j+1)} = \tilde{M}^{(j+1)}|_K = R_K^T \Lambda_K^{-2} R_K$$

con $R_K^T = [\mathbf{r}_{1,K}, \mathbf{r}_{2,K}]$ e $\Lambda_K = \text{diag}(\lambda_{1,K}, \lambda_{2,K})$.

Abbiamo così definito una pratica procedura per calcolare la nuova metrica, e quindi trovare la mesh adattata, che verrà implementata nel prossimo capitolo.

Capitolo 5

Risultati sperimentali

Abbiamo introdotto nei capitoli precedenti le principali tecniche di segmentazione delle immagini, dagli albori rappresentati dal funzionale di Mumford-Shah, fino al recentissimo modello RSFE. Le difficoltà nella risoluzione delle diverse equazioni, nonché i loro limiti pratici, sono stati evidenziati con la massima cura, al fine di rendere il lettore partecipe della complessità del problema.

In questo capitolo discutiamo il problema della discretizzazione di tali equazioni. Innanzitutto si descrive in maniera generale l'implementazione del codice nella Sezione 5.2, poi si studia la sensibilità dei risultati rispetto ad alcuni parametri del modello (Sez. 5.3) e infine si dedica un'ampia sezione ai confronti tra le soluzioni ottenute con griglie differenti (Sez. 5.4).

5.1 Alcune considerazioni introduttive

Come noto un qualsiasi problema differenziale può essere gestito e risolto da un calcolatore solamente dopo essere stato discretizzato attraverso un processo opportuno. Due tra i metodi più utilizzati in questo campo sono il metodo alle differenze finite [64] e il metodo agli elementi finiti [59].

Nel campo dell'analisi delle immagini l'approccio più inflazionato è senza dubbio quello basato sulle differenze finite. Tale scelta segue direttamente dalla particolare struttura delle immagini digitali, descritta nel Capitolo 1. Infatti l'insieme ordinato dei pixel si presta in modo naturale ad essere utilizzato come griglia di calcolo per le differenze finite.

Tuttavia la letteratura scientifica sull'argomento non è priva di lavori in cui si esplora l'utilizzo di schemi ad elementi finiti. Ad esempio in [17, 66, 58] si risolvono diversi problemi legati al trattamento delle immagini con gli elementi finiti, introducendo anche delle tecniche euristiche di adattività di griglia. La possibilità di adattare la griglia al problema differenziale studiato permette, come abbiamo già spiegato in precedenza, di ottenere una soluzione più precisa in un tempo computazionale spesso in-

feriore.
La soluzione di un problema di segmentazione è data da una curva che descrive i contorni interni dell'immagine; una griglia adattata a tale soluzione sarà quindi molto fitta in corrispondenza dei contorni, e permetterà di identificare in modo unifico l'immagine. Risolvere il problema con una procedura di adattazione di griglia permette di ottenere un duplice risultato: innanzitutto la soluzione del problema e poi la ricerca della griglia computazionale. Questa infatti può essere sfruttata per risolvere altri tipi di problemi sulla stessa immagine iniziale (denoising, infitting, ...), riducendo così drasticamente il costo computazionale rispetto ai metodi che si usano la segmentazione come step di *pre-processing* al fine di trovare la griglia. In effetti l'approccio di generazione di griglia più usato, soprattutto nelle applicazioni alle immagini mediche, consiste nel dare i contorni segmentati come input ad un generatore di mesh [20].

Le caratteristiche particolari della griglia adattata per un problema di segmentazione sono chiaramente congeniali all'utilizzo di elementi finiti anisotropi. La possibilità di sfruttare le competenze sulle stime anisotrope a posteriori presentate nel Capitolo 4 è proprio uno dei motivi per cui abbiamo scelto di cimentarci con la risoluzione dei problemi di segmentazione attraverso elementi finiti. Infatti il metodo delle differenze finite permette poca flessibilità per quanto riguarda la struttura della griglia.
In particolare si è scelto di guidare l'adattazione di griglia con uno stimatore di Zienkiewicz-Zhu (si veda la Sez. 3.3). La ragione principale di questa decisione è da cercarsi proprio nella definizione di stimatore recovery-based. Si è visto come esso sia basato sull'approssimazione del gradiente della soluzione approssimata, a differenza di molti altri stimatori che considerano la soluzione stessa. Questo fatto assume una particolare importanza nel nostro caso, visto che la soluzione ϕ del problema di segmentazione è una funzione di level set e come mostrato in Figura 5.1 cambia drasticamen-

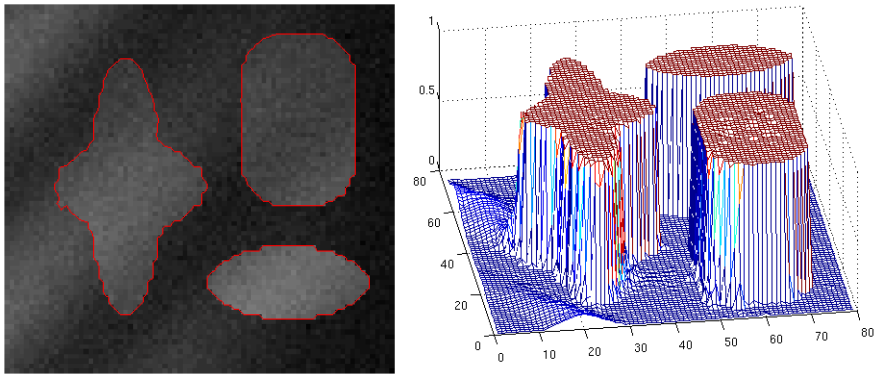


Figura 5.1: Risultato della segmentazione di un'immagine artificiale affetta da rumore (sinistra) e corrispondente funzione di level set (destra).

te valore in corrispondenza dei contorni dell'immagine. Il suo gradiente è quindi una grandezza di particolare importanza nella simulazione.

Un altro motivo che ci ha spinto verso l'utilizzo di questo particolare stimatore sono gli ottimi risultati ottenuti in altre sedi, già citati precedentemente, nonché la semplicità di calcolo che lo caratterizza. Sembra doveroso evidenziare che nel campo della segmentazione delle immagini si tratta di una metodologia completamente innovativa, in quanto a nostra conoscenza gli unici lavori sull'adattazione di griglia in questo ambito sono basati su criteri euristici e non su stimatori a posteriori.

Come sempre, purtroppo, le novità portano con sé anche alcune difficoltà. Infatti si vedrà nel seguito che lo stimatore ZZ permette di generare delle mesh con molti meno gradi di libertà rispetto ad altri tipi di adattazione e la precisione della soluzione è comparabile se non superiore, ma si paga il prezzo di un costo computazionale molto elevato.

Nel seguito si analizzano i risultati ottenuti mettendo un'enfasi particolare sul confronto tra i diversi approcci presentati e sulla sensibilità del test a differenti parametri. Infatti in questo particolare frangente non è possibile effettuare analisi ritenute fondamentali nel campo dell'analisi numerica, come le stime di convergenza o di stabilità. Fortunatamente l'argomento dello studio, cioè le immagini, ci permettono di effettuare dei semplici confronti ad occhio nudo sulla bontà delle segmentazioni!

5.2 Implementazione

Il modello per la segmentazione delle immagini implementato in questo lavoro è quello basato sulla *region scalable fitting energy* descritto nel Capitolo 1. È stato scelto non soltanto perché quello di più recente sviluppo, ma soprattutto perché si tratta di un metodo polivalente che riesce a conciliare una relativa rapidità di calcolo con una segmentazione di buon livello capace di riconoscere i contorni anche di immagini con forti gradienti di intensità. Si implementa quindi l'Algoritmo 3.

Ricordiamo che l'equazione differenziale di Eulero-Lagrange che risolve il problema di minimizzazione in considerazione è

$$\triangle \phi = \frac{r}{\lambda} + \operatorname{div}(\vec{d} - \vec{b}).$$

Nell'articolo originale tale equazione viene risolta come mostrato in (2.20) con il metodo delle differenze finite, usando strategicamente uno schema di Gauss-Seidel per tenere in conto l'avanzamento temporale. Nella pratica

ciò equivale a risolvere il seguente problema parabolico:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} - \triangle \phi = -\dfrac{r}{\lambda} - \operatorname{div}(\vec{d} - \vec{b}) & \text{in } \Omega,\ t > 0 \\ \dfrac{\partial \phi}{\partial \mathbf{n}} = 0 & \text{su } \partial\Omega \\ \phi(0, \cdot) = \phi_0(\cdot) & \text{in } \Omega. \end{cases} \tag{5.1}$$

La risoluzione numerica di tale equazione si effettua con l'ausilio di `FreeFem++`, un software *freeware* sviluppato da F. Hecht e O. Pironneau dell'Université Paris VI [45]. Si tratta di un ambiente di sviluppo integrato (IDE in inglese) per la risoluzione numerica delle equazioni a derivate parziali con il metodo agli elementi finiti. Le sue caratteristiche salienti comprendono un ottimo generatore di mesh e un solutore integrato di equazioni ellittiche. Tale solutore richiede in input la formulazione debole dell'equazione, che è quindi necessario ricavare.

5.2.1 Discretizzazione del problema parabolico

La formulazione debole di (5.1) si ottiene moltiplicando l'equazione differenziale a t fissato per una funzione test $v = v(\mathbf{x})$ ed integrando su Ω. Per ogni $t > 0$ si cerca $\phi(t) \in V$, con $V = H^1(\Omega)$ tale che:

$$\int_\Omega \frac{\partial \phi(t)}{\partial t} v d\mathbf{x} + \int_\Omega \nabla \phi \cdot \nabla v d\mathbf{x} = -\int_\Omega f v d\mathbf{x} \quad \forall v \in V. \tag{5.2}$$

e $\phi(0) = \phi_0$. Si è inoltre posto $\left(\dfrac{r}{\lambda} + \operatorname{div}(\vec{d} - \vec{b})\right) = f$.
Si verifica immediatamente che la forma bilineare è continua e debolmente coerciva, quindi si deduce un risultato di unicità della soluzione debole [59]: il problema (5.2) ammette un'unica soluzione $\phi \in L^2(\mathbb{R}, V) \cap C^0(\mathbb{R}, L^2(\Omega))$, tale che $\partial\phi/\partial t \in L^2(\mathbb{R}, V')$ essendo V' il duale di V.

La semidiscretizzazione di (5.2) nelle sole variabili spaziali si ottiene considerando l'approssimazione di Galerkin della soluzione $\phi_h \in V_h$, dove $v_h \subset V$ è un opportuno spazio a dimensione finita: per ogni $t > 0$ trovare $\phi_h(t) \in V_h$ tale che

$$\int_\Omega \frac{\partial \phi_h(t)}{\partial t} v_h d\mathbf{x} + \int_\Omega \nabla \phi_h \cdot \nabla v_h d\mathbf{x} = -\int_\Omega f_h v_h d\mathbf{x} \quad \forall v_h \in V_h \tag{5.3}$$

e $\phi_h(0) = \phi_{0h}$ essendo ϕ_{0h} una conveniente approssimazione di ϕ_0 nello spazio V_h.

Applicando il θ-metodo, che approssima la derivata temporale in (5.3) con un semplice rapporto incrementale, si ottiene finalmente la discretizza-

zione completa:

$$\int_\Omega \frac{\phi_h^{k+1} - \phi_h^k}{dt} v_h d\mathbf{x} + \int_\Omega \left(\theta \nabla \phi_h^{k+1} + (1-\theta)\nabla \phi_h^k\right) \cdot \nabla v_h d\mathbf{x} = \\ - \int_\Omega f_h v_h d\mathbf{x} \qquad \forall v_h \in V_h.$$

5.2.2 Algoritmo di Split-Bregman

Ottenuta la formulazione debole, vediamo ora gli step fondamentali di implementazione dell'Algoritmo 3.

1. Importazione dell'immagine : si leggono le dimensioni dell'immagine in pixel `nx` e `ny` e i valori di intensità in ogni punto del dominio da un file di testo creato con `MATLAB`.
2. Creazione della mesh : ogni pixel corrisponde ad un vertice della griglia, che viene generata con il comando

   ```
   mesh Th=square(nx,ny,[nx*x,ny*y]).
   ```

3. Definizione dello spazio ad elementi finiti : V_h è uno spazio ad elementi finiti formato da polinomi di primo grado definito sulla mesh `Th`

   ```
   fespace Vh(Th,P1).
   ```

4. Dichiarazione delle funzioni ad elementi finiti e degli altri parametri del problema, tra cui il contorno iniziale ϕ_0 e l'immagine iniziale `Im0` mostrata in Fig. 5.2 insieme alla mesh `Th`.

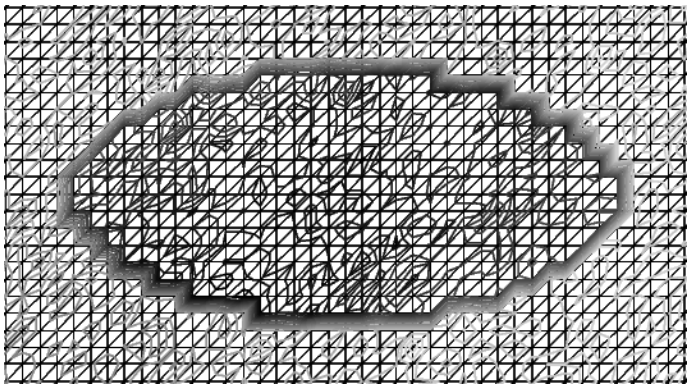


Figura 5.2: L'immagine iniziale viene approssimata con dei polinomi di grado 1 su tutto il dominio; le dimensioni della mesh corrispondono a quelle dei pixel.

5. Definizione del problema differenziale : la formulazione debole (5.3) si traduce in linguaggio `FreeFem++` così

```
problem RSF(phi,v) = int2d(Th)(phi*v) - int2d(Th)(phi1*v)
+ dt*(int2d(Th)(v*(dx(ddx)-dx(bbx)+dy(ddy)-dy(bby)+r/l))
+      int2d(Th)(th*(dx(phi)*dx(v)+dy(phi)*dy(v)))
+      int2d(Th)((1-th)*(dx(phi1)*dx(v)+dy(phi1)*dy(v))))
```

   avendo indicato `phi`=ϕ^{k+1}, `phi1`=ϕ^k e `th`=θ.

6. Ciclo temporale : si risolve l'equazione e si effettuano i passi di aggiornamento del metodo di Split-Bregman finché non si verifica una delle due condizioni del ciclo while:

```
while( iter<maxIter && errnorm[iter]>toll )
```

   dove si è posto

$$\texttt{errnorm[k]} = \frac{\|\phi^k - \phi^{k-1}\|_2}{\|\phi^{k-1}\|_2}.$$

Gli aggiornamenti delle variabili ausiliarie dell'algoritmo di Split-Bregman allo step 6 sono passaggi algebrici semplici, e non sembra necessario dettagliarli ulteriormente. È però interessante notare come sono state calcolate le numerose convoluzioni necessarie ad ogni passo per calcolare il valore della funzione r secondo le equazioni (1.22)-(1.24). In effetti, `FreeFem` non possiede un comando per calcolare esplicitamente il risultato di una convoluzione, e non è altresì possibile farlo definendo l'integrale. Un'astuzia per aggirare tale problema consiste nell'appoggiarsi al noto risultato secondo il quale la soluzione fondamentale dell'equazione del calore

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,\mathbf{x}) - \triangle u(t,\mathbf{x}) = 0 & \mathbf{x} \in \mathbb{R}^2, t > 0 \\ u(0,\mathbf{x}) = u_0(x) & \mathbf{x} \in \mathbb{R}^2 \end{cases}$$

con $u_0 \in L^1(\mathbb{R}^2)$, è data da:

$$u(t,\mathbf{x}) = \int_{\mathbb{R}^2} G_\sigma(\mathbf{x}-\mathbf{y}) u_0(\mathbf{y}) d\mathbf{y} = (G_\sigma * u_0)(\mathbf{x}).$$

Quindi per risolvere ognuna delle convoluzioni è sufficiente definire un problema differenziale con il comando `problem`.

5.2.3 Adattazione di griglia

L'adattazione di griglia in `FreeFem++` si effettua attraverso il comando

```
Th = adaptmesh(Th, f, [parametri] )
```

in cui `Th` è la mesh da modificare, e `f` è la funzione alla quale si vuole adattare la griglia. `Adaptmesh` ammette in ingresso diversi parametri, tra cui:

- `hmin`: dimensione minima dei lati degli elementi finiti;
- `hmax`: dimensione massima dei lati degli elementi finiti;
- `nbvx`: numero massimo di vertici della griglia;
- `err`: errore di interpolazione (di default = 0.01);
- `iso`: valore booleano, se vero impone la generazione di una griglia isotropa (è falso di default);
- `metric`: vettore di dimensione 3 che identifica la metrica.

Questo comando, nella sua versione più semplice, usa un algoritmo automatico di adattazione di griglia basato sulla matrice Hessiana della funzione `f`. Altrimenti l'utente può scegliere di definire una metrica manualmente con il parametro

```
metric = [m11(),m12(),m22()],
```

dove `m11`, `m12` e `m22` sono delle funzioni ad elementi finiti che identificano gli elementi della matrice della metrica M introdotta nella Sezione 4.6:

$$M = \begin{bmatrix} \texttt{m11} & \texttt{m12} \\ \texttt{m12} & \texttt{m22} \end{bmatrix}. \tag{5.4}$$

Vediamo ora come procedere per calcolare la metrica a partire da uno stimatore ZZ a posteriori. Dal punto di vista teorico la costruzione di tale stimatore anisotropo è stata trattata nel Capitolo 4.

Innanzitutto, rispetto al caso base presentato nel paragrafo precedente, è necessario definire un secondo spazio ad elementi finiti, formato da funzioni costanti in ogni triangolo:

```
fespace Nh(Th,P0).
```

Il calcolo effettivo dello stimatore avviene ad ogni passo temporale all'interno del ciclo while inserito nello Step 6 dell'algoritmo base. Entriamo ora nei dettagli del procedimento:

a. Calcolo del gradiente ricostruito : ogni sua componente - `Rgx` lungo x e `Rgy` lungo y - è una funzione ad elementi finiti con un numero di gradi di libertà pari al numero dei vertici. Quindi, contrariamente all'operatore $G_R(u_h)$ definito in (3.22), qui la media pesata si effettua sui triangoli che condividono lo stesso vertice. Sia cioè `Rgx[i]` il valore corrispondente all'i-esimo vertice della mesh:

$$\texttt{Rgx[i]} = \frac{\sum_{T \ni \texttt{i}} |T| \cdot (\texttt{dx(phi)})|_T}{\sum_{T \ni \texttt{i}} |T|}$$

dove `dx(phi)`$= \dfrac{\partial \phi}{\partial x}$ e appartiene allo spazio `Nh`. Analogamente si trova la componente lungo y del gradiente ricostruito.

b. Costruzione della matrice di adiacenza : `A` è una matrice di dimensione [*numero vertici* x *numero triangoli*]. Ogni riga di `A` corrisponde ad un vertice della griglia, e vi si segna il numero di ogni triangolo che condivide quel vertice. Così si riduce il costo computazionale dell'algoritmo, dato che l'associazione tra vertici e i rispettivi triangoli deve essere ripetuta più volte ad ogni iterazione.

c. Costruzione della matrice G : (4.4) ognuna delle 3 componenti della matrice simmetrica è stata definita con una forma variazionale del tipo

```
varf g11K(a,b)=int2d(Th)((Rgx-dx(phi))*(Rgx-dx(phi))*b);
```

cosicché il valore di `G` in corrispondenza dei vertici si ottiene con il comando

```
G11K[]=g11K(0,Nh).
```

Ora è sufficiente sfruttare la matrice di adiacenza appena calcolata per associare ad ogni triangolo della mesh il corrispondente valore di `G`. Dividendo poi per l'area di ogni patch si ottiene la matrice $\tilde{G}_K(e_h^*)$ definita in (4.8). Queste funzioni ad elementi finiti appartengono tutte allo spazio `Nh`.

d. Calcolo di autovettori e autovalori di $\tilde{G}_K(e_h^*)$: secondo la Proposizione 4.4 gli autovalori di tale matrice indicano le direzioni degli assi principali dei triangoli della nuova mesh. Questo calcolo viene svolto tramite una chiamata a `MATLAB`, linguaggio molto efficiente nella risoluzione di problemi di algebra lineare. Il passaggio dei parametri tra i due programmi si effettua con una semplice scrittura e lettura su dei file di testo.

e. Costruzione della nuova metrica : la dimensione degli elementi finiti è data dalle espressioni (4.10). Tali valori sono calcolati ancora in `MATLAB`, così come la matrice M_K della metrica, con le semplici moltiplicazioni matriciali ricordate nella Sezione 4.6.
Il comando `adaptmesh` vuole in input delle matrici con un numero di gradi di libertà pari al numero dei vertici della griglia. È quindi necessario effettuare il passaggio opposto rispetto a quello del punto c., dalla metrica *element-wise* M_K alla metrica *node-wise* M:

$$M[i] = \frac{\sum_{T \ni \mathtt{i}} M_K[T] \cdot |T|}{3 \sum_{T \ni \mathtt{i}} |T|}.$$

Il coefficiente 3 a denominatore serve per ridimensionare il triangolo di riferimento ad un triangolo unitario.

La matrice M così trovata è quella definita in (5.4). Ora è sufficiente chiamare `adaptmesh` con i parametri appena calcolati per ottenere la griglia adattata.

5.3 Parametri del modello

Come si può dedurre dalla trattazione teorica nel Paragrafo 1.4.2, i risultati di segmentazione ottenuti applicando il modello RSF dipendono da svariati parametri. Nella pratica si verifica facilmente che la loro regolazione è tutt'altro che facile, dipendendo sia dal tipo di immagine che si segmenta, sia dal risultato cercato.

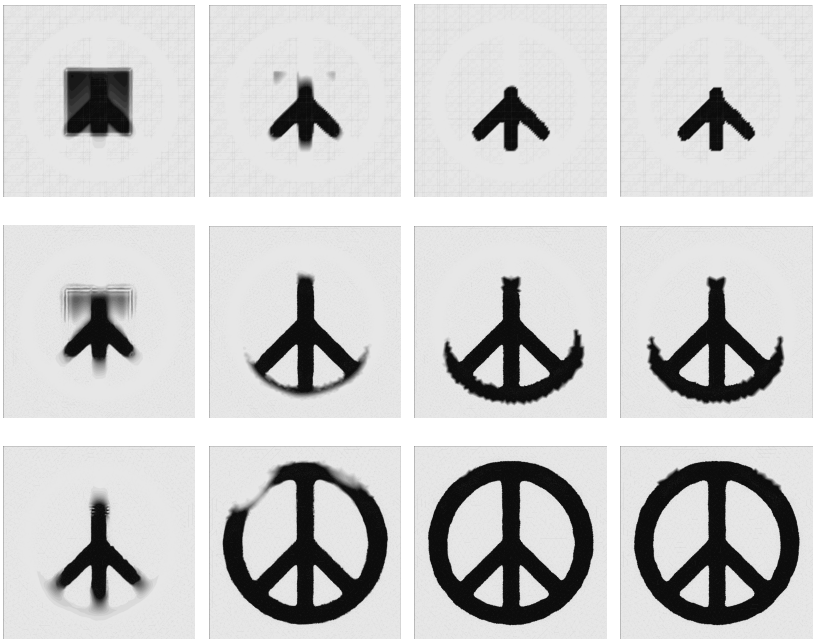
In questo paragrafo si vuole spiegare il significato e il ruolo di ogni parametro, attraverso esempi pratici e richiami alla teoria, cercando di fornire uno strumento euristico per la scelta degli stessi. A nostro parere la mancanza di una tale esplicazione è una grave lacuna dei lavori citati sull'argomento, che solitamente scelgono i parametri ad hoc per ogni simulazione, senza fornire una visione globale della loro utilità pratica.
Analizziamo quindi il ruolo di un parametro alla volta, mantenendo fissi gli altri in modo da semplificare la comprensione.

Moltiplicatore di Lagrange λ
Questo parametro deriva dall'imposizione del vincolo di uguaglianza $\vec{d} = \nabla\phi$ nell'algoritmo di Split-Bregman, e agisce da moltiplicatore di Lagrange per la forma quadratica associata.

Il ruolo di λ nell'evoluzione della funzione ϕ si può dedurre dall'analisi dell'equazione (5.1). Il termine $-\frac{r}{\lambda}$ è di fondamentale importanza per l'evoluzione del contorno attivo. Infatti si ha che $r = \lambda_1 e_1 - \lambda_2 e_2$, espressione che deriva direttamente dall'energia di fitting, in cui e_1 (rispettivamente e_2) è un integrale che agisce da media quadratica pesata della differenza tra il valore di fitting f_1 (f_2) e l'intensità dell'immagine all'esterno (interno) della curva in evoluzione.

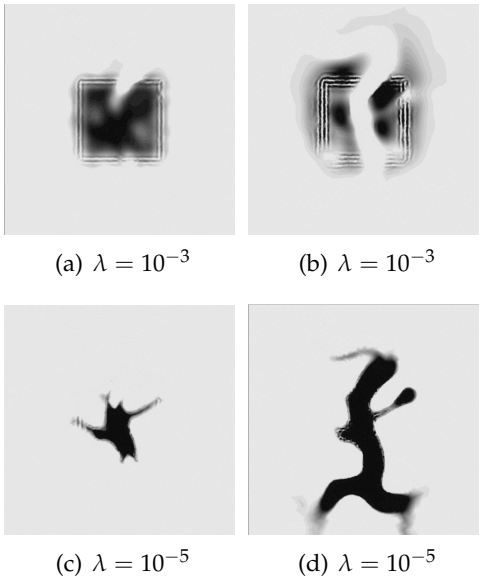
Un valore alto di λ riduce l'importanza attribuita a r e rallenta così l'evoluzione del contorno, come confermano le immagini in Fig. 5.3. Si può anche notare che per valori molto bassi di λ non si hanno più variazioni significative del contorno, infatti ϕ evolve in maniera identica per $\lambda = 10^{-3}$ e per $\lambda = 10^{-5}$.

Questo fenomeno è probabilmente dovuto all'intervallo di esistenza di ϕ: infatti a valori diversi del moltiplicatore di Lagrange corrispondono soluzioni diverse dell'equazione differenziale, ma trovandosi tali soluzioni al di fuori dell'intervallo $a_0 \leq \phi \leq b_0$ vengono subito uniformate.



(a) $\lambda = 10^{-1}$ (b) $\lambda = 10^{-2}$ (c) $\lambda = 10^{-3}$ (d) $\lambda = 10^{-5}$

Figura 5.3: Regione segmentata per diversi valori del parametro λ: dopo 1, 3 e 7 iterazioni.



(a) $\lambda = 10^{-3}$ (b) $\lambda = 10^{-3}$

(c) $\lambda = 10^{-5}$ (d) $\lambda = 10^{-5}$

Figura 5.4: Estensione di ϕ dopo 3 (sin.) e 8 (dx.) iterazioni, al variare di λ. In alto per $\lambda = 10^{-3}$ si può notare un'inversione di valore, che non avviene per $\lambda = 10^{-5}$ in basso.

Un altro fenomeno interessante si può notare in Fig.5.4, in cui si segmenta un'immagine di maggiore complessità. Si può vedere che, per valori grandi di λ, l'algoritmo di segmentazione riconosce la parte esterna dell'immagine, invece di quella interna. Questo effetto è dovuto ad un cambiamento nel segno della soluzione.

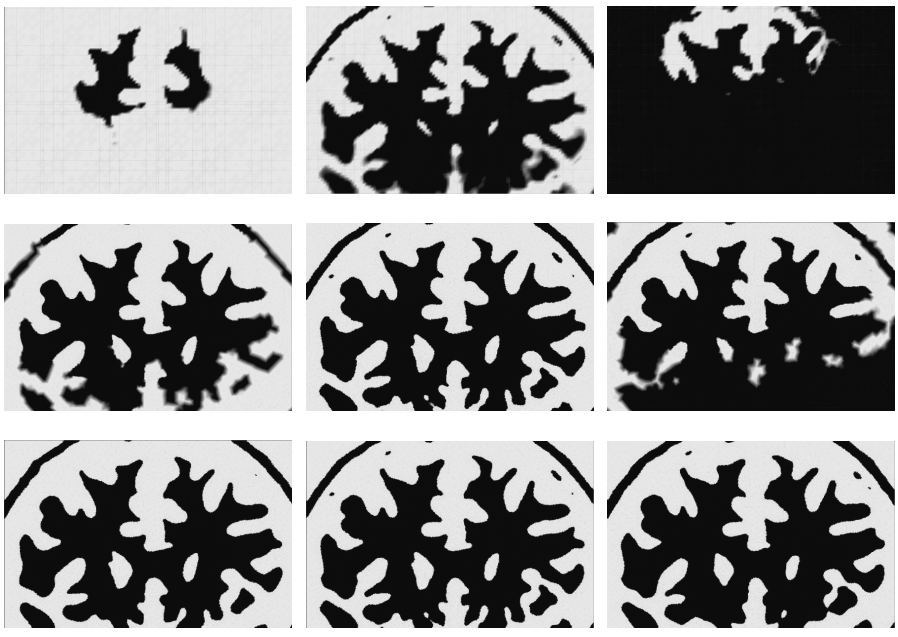
Confrontando i risultati con quelli della Figura 5.3, si può effettuare anche un'ulteriore deduzione: la segmentazione di immagini con contorni poco definiti risulta più semplice usando valori di λ più piccoli. In generale si può concludere che i valori accettabili del parametro λ sono da cercare nell'intervallo:

$$\boxed{10^{-3} \leq \lambda \leq 10^{-5}.}$$

Pesi: esterno/interno λ_1 e λ_2

Le costanti λ_1 e λ_2 rappresentano, nella definizione dell'energia di fitting (1.18), i pesi degli integrali rispettivamente sulla regione esterna e interna rispetto alla curva in evoluzione.

Innanzitutto cerchiamo di stabilire l'ordine di grandezza di tali costanti, studiando il loro ruolo nel problema differenziale (5.1). L'intensità dell'immagine originale in ogni punto del dominio assume valori appartenenti all'intervallo $[0, 255]$; affinché l'influenza del fattore r/λ nell'equazione



(a) $\lambda_1 > \lambda_2$ (b) $\lambda_1 = \lambda_2$ (c) $\lambda_1 < \lambda_2$

Figura 5.5: Regione segmentata dopo 1, 3 e 6 iterazioni al variare delle costanti λ_1 e λ_2.

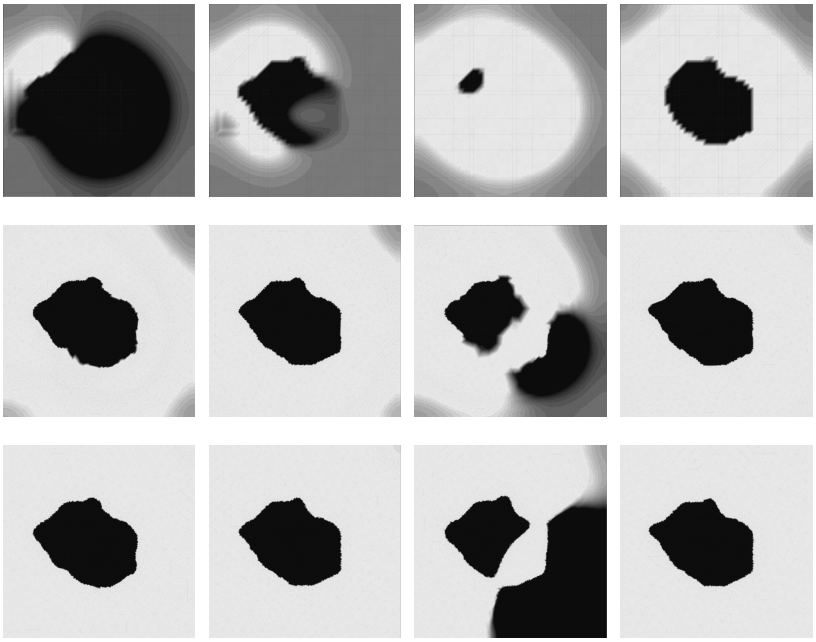
differenziale sia significativa è necessario che esso assuma dei valori dello stesso ordine di grandezza. Quindi per bilanciare il parametro λ, il cui campo di variazione è stato definito nel paragrafo precedente, si scelgono valori dell'ordine di 10^{-5}.

Nella maggior parte dei casi si imposta $\lambda_1 = \lambda_2 = 10^{-5}$, situazione corrispondente ad un equilibrio tra le regioni interne ed esterne rispetto al contorno di livello zero. Al contrario, quando $\lambda_1 \neq \lambda_2$, la penalità che si impone sui due integrali in (1.18) è differente. Ad esempio, se $\lambda_2 > \lambda_1$ la penalità imposta sull'integrale sulla regione interna è maggiore; così si penalizza l'aumento dell'area della regione interna e si previene la formazione di nuovi contorni, che aumenterebbero tale area. In particolare nelle regioni dell'immagine lontane dal contorno, dove $e_1 \approx e_2$, il segno di r è negativo e ϕ non cambia segno, non formando nuovi contorni.

Questo effetto risulta evidente dall'analisi della Figura 5.5, in cui un'immagine raffigurante una RMN del cervello umano viene segmentata con diversi valori di λ_1 e λ_2. In particolare nella prima colonna $\lambda_1 = 1.1e-5$ e $\lambda_2 = 1e-5$, e si nota che l'evoluzione della regione segmentata è lenta, ma precisa. Qui infatti la segmentazione si limita a riconoscere le aree principali del cervello, al contrario di quanto avviene nella seconda e nella terza colonna (rispettivamente $\lambda_1 = \lambda_2 = 1e-5$ e $\lambda_1 = 1e-5, \lambda_2 = 1.1e-5$), in cui si nota la presenza di piccole macchie esterne.

Nel caso appena presentato può sembrare che il valore di questi parametri influenzi poco il risultato finale del processo di segmentazione. Sfortunatamente, questo è vero solamente se l'immagine sotto esame è semplice e soprattutto se il contorno iniziale è ben centrato rispetto alle aree da segmentare. Consideriamo ad esempio quanto avviene in Fig. 5.6. Nelle prime tre colonne si usa un contorno iniziale ϕ_0 posizionato in alto a sinistra, differentemente dal caso standard in cui esso è centrato rispetto all'immagine. Mentre nei casi (a) e (b) tale impostazione non crea nessun problema in vista del risultato finale, quando $\lambda_1 > \lambda_2$ si privilegia talmente tanto la formazione di nuovi contorni, che se ne crea uno esterno all'immagine! Ciò non avviene scegliendo un contorno iniziale centrato, come mostrato in Fig. 5.6(d).

La creazione di nuovi contorni è in realtà considerata un vantaggio di questo metodo, infatti ne aumenta la velocità oltre a permettere il riconoscimento dei contorni interni. Bisogna però fare attenzione a limitare questo effetto allo stretto necessario: capita spesso di imbattersi in situazioni come quella appena descritta, in cui il nuovo contorno disegna l'esterno dell'oggetto invece dell'interno.



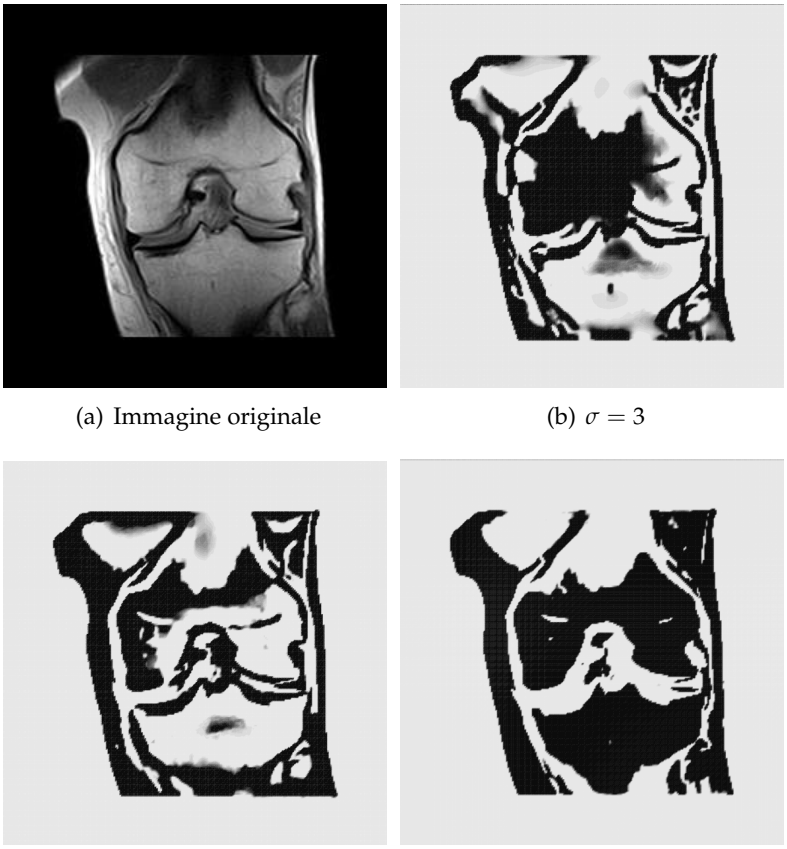
(a) $\lambda_1 < \lambda_2$ (b) $\lambda_1 = \lambda_2$ (c) $\lambda_1 > \lambda_2$ (d) $\lambda_1 > \lambda_2$ contorno centrato

Figura 5.6: Regione segmentata dopo 1, 3 e 6 iterazioni al variare delle costanti λ_1 e λ_2. Nelle prime tre colonne ϕ_0 è scelta in alto a sinistra, mentre nell'ultima è centrata.

Parametro di scala del nucleo gaussiano σ

La scalabilità in spazio rappresenta il vantaggio e l'innovazione principale del modello RSF: la scelta del parametro σ permette di decidere quante informazioni sull'intensità dell'immagine sfruttare, da piccoli intorni all'intero dominio. Infatti le funzioni $f_1(\mathbf{x})$ ed $f_2(\mathbf{x})$, la cui espressione è data da (1.22), approssimano l'intensità dell'immagine in una regione centrata nel punto $\mathbf{x}$ la cui dimensione dipende dal valore di σ. In particolare, per valori bassi di σ, l'energia di fitting (1.17) è calcolata basandosi unicamente su informazioni di intensità presenti in un piccolo intorno di $\mathbf{x}$.

La scelta del valore del parametro di scala non è banale. Infatti σ piccolo permette di ottenere una migliore precisione nella localizzazione dei contorni dell'immagine. Ma per la maggior parte delle immagini reali, la cui intensità è solitamente omogenea, è possibile usare valori abbastanza grandi di σ ottenendo dei risultati soddisfacenti; così aumenta anche l'indipendenza della soluzione dalla posizione iniziale del contorno.
Questo secondo effetto è dovuto all'analogia con il metodo di Chan e Vese (Sec. 1.3), che può essere considerato il limite del modello RSF per $\sigma \to \infty$. A tale limite infatti le funzioni di fitting f_1 ed f_2 coincidono con la media



(a) Immagine originale (b) $\sigma = 3$

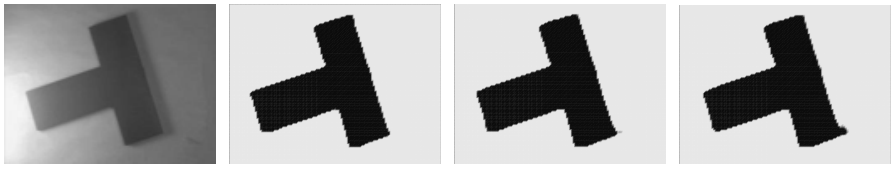
(c) $\sigma = 10$ (d) $\sigma = 70$

Figura 5.7: Risultati della segmentazione al variare di σ, a convergenza dell'algoritmo.

dell'intensità nelle regioni $\{\phi > 0\}$ e $\{\phi < 0\}$, e corrispondono alle intensità medie che compaiono nell'espressione dell'energia (1.16) nel modello di Chan e Vese.

Dal punto di vista del costo computazionale, un parametro di scala più grande aumenta il costo computazionale di ogni convoluzione, ma riduce il numero di iterazioni necessarie per ottenere l'immagine segmentata. Quindi il costo può essere considerato comparabile.

La possibilità di variare σ permette di segmentare con successo immagini molto diverse. Il criterio di scelta di questo parametro è, ancora una volta, di tipo euristico: dato che esso influenza l'ampiezza delle regioni in cui si calcola la media delle intensità, bisogna cercare di considerare una regione abbastanza significativa dell'immagine. Ciò risulta particolarmente complicato con immagini in cui l'intensità varia rapidamente all'interno dell'oggetto da segmentare senza la presenza di contorni ben definiti, co-



(a) Immagine originale (b) $\sigma = 3$ (c) $\sigma = 10$ (d) $\sigma = 20$

Figura 5.8: Risultati della segmentazione al variare di σ, a convergenza dell'algoritmo.

me ad esempio nella Figura 5.7(a) rappresentante la RMN di un ginocchio. In questo caso il calcolo di f_1 e f_2 in punti distanti dell'immagine può dar luogo a dei risultati contraddittori, in quanto la tonalità predominante può passare rapidamente dal chiaro allo scuro. Così ϕ assumerà valori opposti in corrispondenza dello stesso colore, ottenendo come risultato della segmentazione incoerente come quella delle Fig. 5.7(b)-5.7(c). Si può notare come in questo caso l'algoritmo non riesca ad individuare il femore e la tibia nonostante i loro contorni siano definiti. Al contrario nella Figura 5.7(d), in cui $\sigma = 70$, l'immagine risultante dalla segmentazione corrisponde a quella originale, nonostante si siano persi alcuni dettagli.

La modifica del parametro σ non influenza invece il processo di segmentazione di quelle immagini in cui gli oggetti sono ben definiti e non sono soggetti a variazioni di intensità al loro interno, come ad esempio quella mostrata in Figura 5.8(a). Qui si può notare come il risultato della segmentazione sia più preciso per valori bassi del parametro di scala (Fig. 5.8(b)), infatti in tale modo si riesce a limitare l'ampiezza della regione presa in considerazione. Al contrario nelle Figure 5.8(c) e 5.8(d) il contorno tende a spingersi verso l'ombra dell'oggetto geometrico.

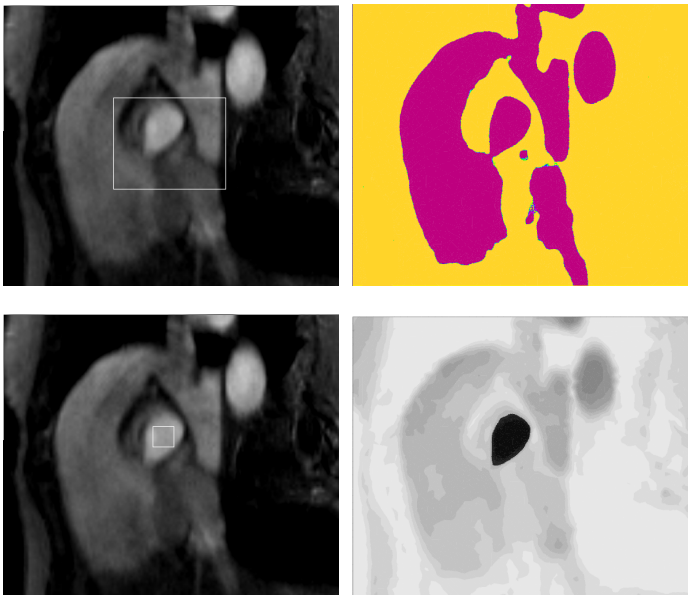
Contorno iniziale ϕ_0

Abbiamo spesso evidenziato l'importanza della scelta del contorno iniziale nel processo di segmentazione, e spiegato come due condizioni iniziali differenti possano portare a diversi risultati. Il modello RSF qui considerato è poco sensibile al contorno iniziale, tanto che tutte le simulazioni presentate in questo capitolo sono state ottenute con la stessa inizializzazione del contorno:

$$\phi_0(x,y) = \begin{cases} 1, & (x,y) \in \left[\frac{\mathtt{nx}}{3}, \frac{2\mathtt{nx}}{3}\right] \times \left[\frac{\mathtt{ny}}{3}, \frac{2\mathtt{ny}}{3}\right], \\ 0, & \text{altrimenti} \end{cases}$$

dove `nx` e `ny` sono le dimensioni dell'immagine in pixel.

È però possibile sfruttare in modo furbo alcune differenze derivanti da diverse scelte del contorno iniziale, come mostrato in Figura 5.9. L'imma-



(a) Contorno iniziale (b) Risultato della segmentazione

Figura 5.9: Risultati della segmentazione al variare di ϕ_0, a convergenza dell'algoritmo.

gine mostra una sezione del cuore, in cui si possono riconoscere l'aorta, la vena cava e una valvola aperta all'interno. Si è interessati sia a segmentare l'immagine nel suo complesso, sia ad estrarre unicamente il contorno della valvola aperta: ciò è possibile scegliendo in maniera opportuna il contorno iniziale.

5.4 Validazione del modello

La validazione del modello di adattazione di griglia esposto in questa sede sarà effettuata in due step. Innanzitutto si mostrano i vantaggi dell'utilizzo dell'adattazione di griglia euristica (cioè con il comando `adaptmesh`) rispetto alla risoluzione con elementi finiti sulla griglia fissa iniziale. In seguito si confrontano i risultati della segmentazione ottenuti con l'adattazione euristica e con l'adattazione basata sullo stimatore a posteriori.

Queste analisi verranno effettuate in maniera qualitativa, in quanto risulta difficile stimare in maniera rigorosa l'efficacia di un processo di segmentazione. Nella pratica ci si baserà sui seguenti elementi per definire la bontà di una segmentazione rispetto ad un'altra: la precisione, il numero di elementi della griglia, le iterazioni necessarie per raggiungere la convergenza, oltre al tempo computazionale. In questa Sezione validiamo quindi l'utilizzo dell'adattazione di griglia attraverso alcuni esempi pratici.

5.4.1 Adattazione euristica

Basiamo la nostra analisi sulle immagini in Fig. 5.10, Fig. 5.11 e Fig. 5.12. Si noti innanzitutto che si tratta di immagini di natura completamente diversa, scelte appositamente per mostrare pregi e difetti dell'algoritmo utilizzato e la sua complessiva affidabilità.

Prima di addentrarci nello specifico dell'analisi comparativa tra la segmentazione con e senza adattazione di griglia, sembra necessario fare qualche considerazione sulla bontà delle segmentazioni presentate nelle figure delle prossime pagine. In Figura 5.10 si vuole segmentare la fotografia di un vortice di Kármán formato da un flusso di acqua circolante attorno ad un cilindro circolare. Le linee di flusso sono molto sottili e in alcuni punti si sovrappongono in maniera particolarmente confusa, ad esempio nella parte sinistra dell'immagine, al momento della creazione del vortice. Per questo motivo l'algoritmo di segmentazione non riesce a riconoscere i dettagli delle linee di flusso e tende piuttosto a confonderle in un tutt'uno. Bisogna comunque apprezzare la precisione con cui vengono segmentati alcuni particolari dell'immagine, come ad esempio la forma particolare dell'ultimo vortice sulla destra.
La Figura 5.11, raffigurante la sezione di un cervello, rappresenta un'altra sfida abbastanza impegnativa nell'ambito della segmentazione. Infatti in tale immagine si alternano delle zone chiare e scure, i cui contorni non sono definiti in modo preciso. Si è arbitrariamente deciso di ottenere una segmentazione che comprendesse anche la meninge: perché ciò sia possibile il parametro σ deve essere scelto abbastanza grande. Con una scelta diversa, sarebbe possibile ottenere una segmentazione che comprenda unicamente la materia grigia interna. Purtroppo questa scelta crea alcune macchie visibili in tutte e tre le immagini.

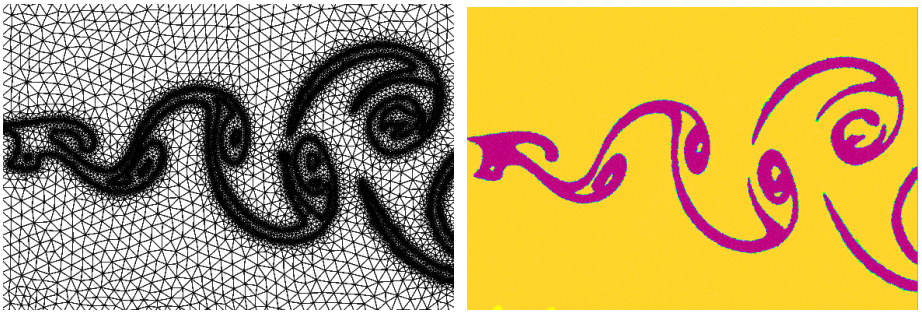
La difficoltà nella segmentazione della Figura 5.12, quadro dell'artista Joan Miró, deriva principalmente dalla presenza di macchie di colore diverso. Questa immagine è stata scelta perché Potendo suddividere l'immagine in solo due regioni, non è possibile distinguere alcuni particolari: ad esempio i cerchi concentrici di colore rosso e nero nella parte destra dell'immagine vengono ridotti ad unico cerchio di tonalità scura. Ancora più difficoltosa è la segmentazione dell'elemento in basso a destra, originariamente colorato di blu, rosso e nero: in nessuna delle segmentazioni mostrate si riesce a distinguere i colori, azione resa ancora più complicata dall'utilizzo di un algoritmo specificatamente definito solo per le immagini in bianco e nero. In ogni caso si nota come la soluzione calcolata con gli elementi finiti riproduca in maniera fedele il quadro originale.
In Figura 5.13 si considera un'immagine estremamente impegnativa da segmentare, rappresentante le famose linee di Nazca, ossia degli estesi geoglifi (linee tracciate sul terreno) situati nel deserto di Nazca, nel Perù meridionale. In questo caso le regioni da segmentare si riducono alle sottilissime linee che compongono il disegno, di difficile individuazione. La precisione della segmentazione con adattazione di griglia euristica è addirittura superiore di quella senza tale adattazione (si osservi per esempio i dettagli della coda a spirale nelle figure 5.13(b) e 5.13(f)).

In ognuno dei casi test si confrontano tre procedimenti:

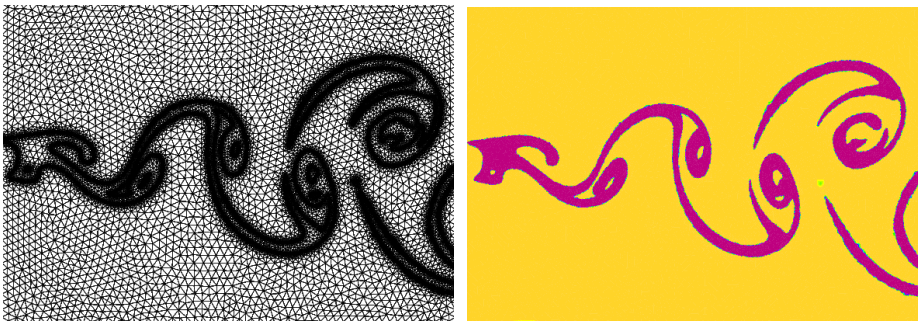
- **a.** Segmentazione senza adattazione di griglia (prima riga);
- **b.** Segmentazione con adattazione di griglia, fissando il valore del parametro `hmin=0.5` (seconda riga);
- **c.** Segmentazione libera con adattazione di griglia (terza riga).

Come primo parametro del confronto tra i tre procedimenti appena citati consideriamo la precisione dei risultati ottenuti: in tutte e tre le figure si nota ad occhio nudo come il risultato della segmentazione effettuata senza adattazione di griglia sia molto sgranato. In effetti la griglia di calcolo rimane quella pixelizzata originale, e l'algoritmo altro non può fare che adattarsi a tale griglia e fornire un risultato costante a tratti e particolarmente poco realistico. Al contrario, le immagini segmentate attraverso il comando `adaptmesh` sono più *smooth* e molto più simili a quelle originali, senza perdere in alcun modo in accuratezza. Questa considerazione di fondamentale importanza sarebbe sufficiente per preferire un approccio basato sull'adattazione di griglia; per completezza analizziamo anche gli altri aspetti citati nella parte introduttiva di questa sezione.

(a) Immagine originale (*200 x 151 pixels*) (b) Senza adattazione



(c) `adaptmesh: hmin=0.5` - griglia (d) `adaptmesh: hmin=0.5`

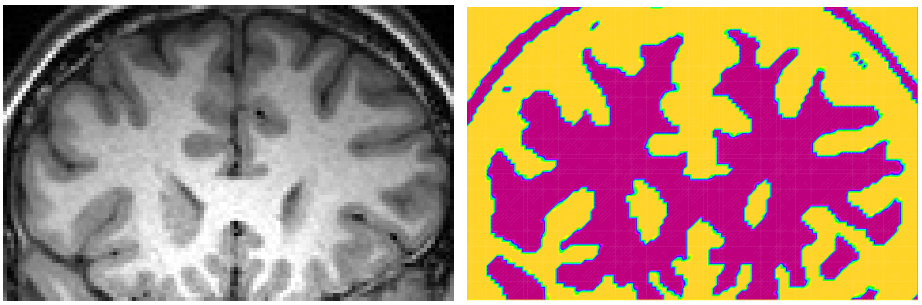


(e) `adaptmesh: hmin=0.01` - griglia (f) `adaptmesh: hmin=0.01`

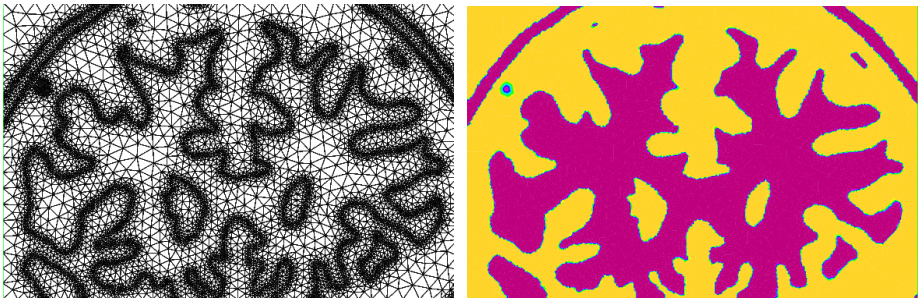
Figura 5.10: Risultati della segmentazione con e senza adattazione [$\beta = 100, \sigma = 70, \lambda = 0.001, \lambda_1 = 10^{-5}, \lambda_2 = 1.1 \cdot 10^{-5}$]. Immagine originale tratta da [31].

Figura	# triangoli	# iter.	tempo (s)
5.10(b)	60400	6	54.6
5.10(c)	28935	8	69
5.10(e)	192738	8	221.3

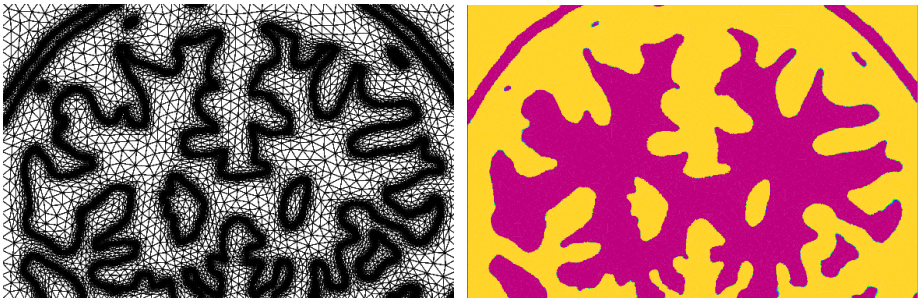
Tabella 5.1: Caso test in Figura 5.10



(a) Immagine originale (*119 x 78 pixels*) (b) Senza adattazione



(c) `adaptmesh: hmin=0.5` - griglia (d) `adaptmesh: hmin=0.5`



(e) `adaptmesh: no hmin` - griglia (f) `adaptmesh: no hmin`

Figura 5.11: Risultati della segmentazione con e senza adattazione [$\beta = 100, \sigma = 70, \lambda = 0.001, \lambda_1 = 1.1 \cdot 10^{-5}, \lambda_2 = 10^{-5}$]. RMN del cervello: sezione frontale.

Figura	# triangoli	# iter.	tempo (s)
5.11(b)	18172	7	20.7
5.11(c)	19779	7	32.1
5.11(e)	1321000	8	62.4

Tabella 5.2: Caso test in Figura 5.11

(a) Immagine originale (*200 x 128 pixels*)

(b) Senza adattazione

(c) `adaptmesh: hmin=0.5` - griglia

(d) `adaptmesh: hmin=0.5`

(e) `adaptmesh: no hmin` - griglia

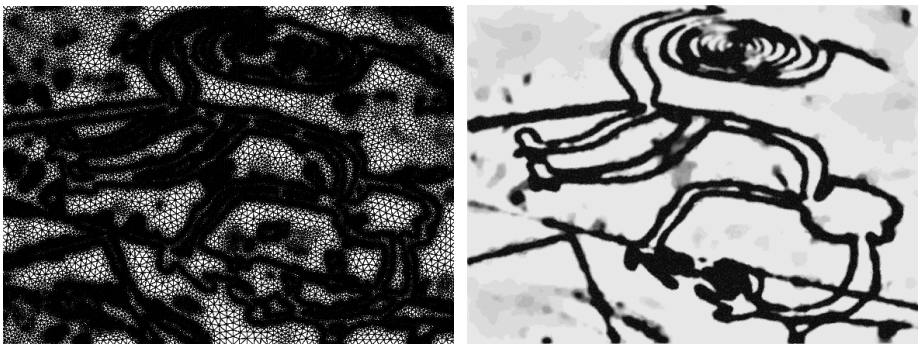
(f) `adaptmesh: no hmin`

Figura 5.12: Risultati della segmentazione con e senza adattazione [$\beta = 100, \sigma = 70, \lambda = 10^{-4}, \lambda_1 = 1.1 \cdot 10^{-5}, \lambda_2 = 10^{-5}$]. Joan Miró, Composizione astratta, 1975.

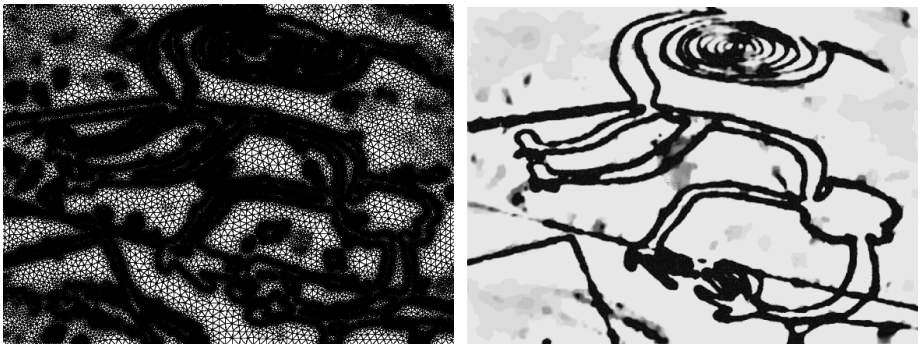
Figura	# triangoli	# iter.	tempo (s)
5.12(b)	50546	6	57.5
5.12(c)	47324	11	121.5
5.12(e)	273681	11	691

Tabella 5.3: Caso test in Figura 5.12

(a) Immagine originale (*200 x 150 pixels*) (b) Senza adattazione



(c) `adaptmesh: hmin=0.5` - griglia (d) `adaptmesh: hmin=0.5`



(e) `adaptmesh: hmin=0.1` - griglia (f) `adaptmesh: hmin=0.1`

Figura 5.13: Risultati della segmentazione con e senza adattazione [$\beta = 100, \sigma = 10, \lambda = 5 \cdot 10^{-3}, \lambda_1 = 10^{-5}, \lambda_2 = 1.05 \cdot 10^{-5}$]. Linee Nazca, Perù.

Una considerazione generale riguarda un limite intrinseco del comando `adaptmesh` di `FreeFem++`, che tende a sovrastimare il numero di triangoli necessari per ottenere un buon risultato. Nelle Figure 5.10, 5.12 e 5.13 non risulta possibile procedere con un'adattazione di griglia senza fissare la dimensione minima dei lati della mesh. Infatti, nonostante si tratti di immagini di piccole dimensioni, se non si assegna un valore al parametro `hmin`, `FreeFem++` esaurisce la memoria a sua disposizione e dà un segnale di errore. Si è allora scelto un valore abbastanza piccolo di `hmin`, con l'obiettivo di ottenere un risultato il più aderente possibile a quello che si avrebbe avuto lasciando il valore di default di tale parametro.

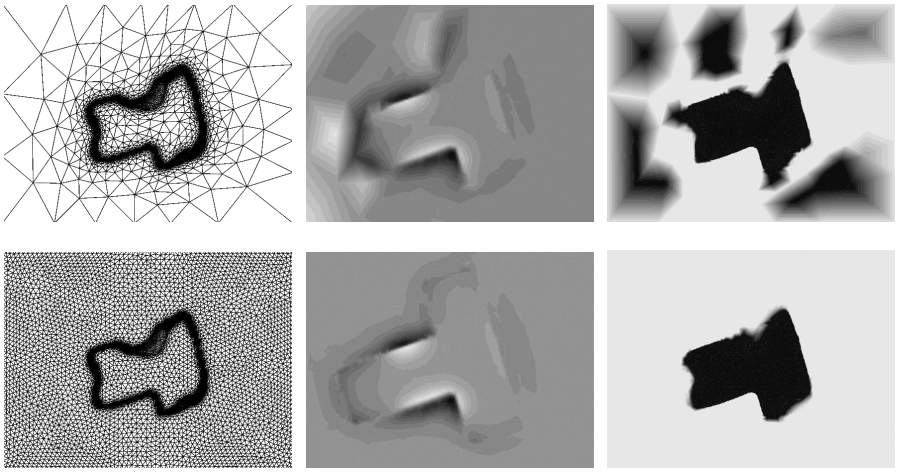
È di fondamentale importanza anche un'altra osservazione relativa al valore del parametro `hmax`. In effetti se non si impone ad `adaptmesh` un limite superiore alla dimensione degli elementi finiti, la griglia sarà molto lasca nelle zone lontane dalla posizione del contorno attivo. In questo modo l'evoluzione della curva viene stravolta in quanto tutte le funzioni che necessarie per calcolare ϕ sono approssimate con dei polinomi di primo grado su ogni triangolo: più le dimensioni dei triangoli sono grandi, più esse saranno imprecise. Per avere un'idea di tale fenomeno si confrontino le immagini in Figura 5.14. Tenendo presente che la griglia iniziale ha i lati di dimensione unitaria, nella pratica si è trovato che un valore appropriato per la maggior parte delle applicazioni è:

$$\texttt{hmax} \simeq 5.$$

Si vedano ora le Tabelle 5.1, 5.2, 5.3 e 5.4 per un confronto tra tempo computazionale, numero triangoli e numero iterazioni nei vari casi.
Si nota innanzitutto che le immagini nella seconda riga di ogni caso test, in cui il valore di `hmin` è fissato a 0.5 cioè dimezzato rispetto alle dimensioni degli elementi nella mesh originale, è possibile ridurre il numero di elementi della griglia base. Ciò avviene sia nei casi presentati in Fig. 5.10(c)-5.10(d), sia in quelli di Fig. 5.12(c)-5.12(d). Il primo caso è particolarmente degno di nota, in quanto il numero di triangoli della griglia risulta addirittura dimezzato rispetto al caso base. Al contrario, le immagini ottenute con una segmentazione libera (o scegliendo `hmin` più piccolo possibile) sono calcolate su una griglia con molti più elementi di quella di partenza.
Confrontando le immagini segmentate nella seconda e terza riga dei casi test non si nota alcuna differenza sostanziale di accuratezza. Si può in-

Figura	# triangoli	# iter.	tempo (s)
5.13(b)	60000	11	104.01
5.13(c)	60671	13	340.52
5.13(e)	988889	13	2721

Tabella 5.4: Caso test in Figura 5.13



(a) mesh (b) funzione r (c) funzione ϕ

Figura 5.14: Seconda iterazione del processo di segmentazione: in alto il valore di `hmax` è quello di default, in basso `hmax=4` [$\beta = 100, \sigma = 10, \lambda = 10^{-4}, \lambda_1 = 1.1 \cdot 10^{-5}, \lambda_2 = 10^{-5}$].

travedere solamente una differenza di spessore nei contorni dell'immagine segmentata (si ricorda che tale contorno coincide proprio con il contorno attivo): più le dimensioni della griglia sono piccole, più tale contorno è leggero e meglio definito. Le relative griglie riflettono queste caratteristiche: sono complessivamente analoghe, ma quelle della riga inferiore sono più fitte e allo stesso tempo hanno i contorni leggermente più sottili.

In tutti e tre i casi test presentati usando la procedura adattativa si soffre di un aumento del costo computazionale. Ciò è dovuto a due fattori: innanzitutto la procedura adattativa (ossia il comando `adaptmesh`) è piuttosto costosa, inoltre, come si è appena visto, l'adattazione può implicare un aumento del numero di elementi della griglia. Può verificarsi anche un aumento del numero di iterazioni necessarie per la convergenza; questo fenomeno ha una spiegazione intuitiva, infatti la griglia stessa può rallentare l'evoluzione del contorno. La griglia viene aggiornata alla fine di ogni passo computazionale, quindi la soluzione allo step successivo verrà calcolata con una griglia adattata rispetto alla soluzione al tempo precedente. È interessante analizzare l'evoluzione della mesh, che segue precisamente l'evoluzione del contorno come mostrato in Figura 5.15.

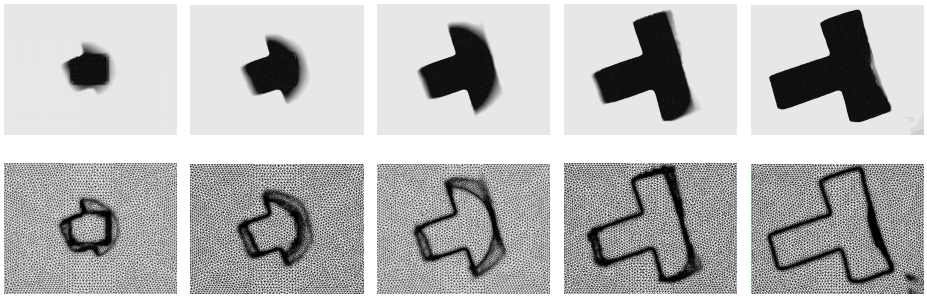


Figura 5.15: L'evoluzione della griglia di calcolo segue l'immagine [$\beta = 100, \sigma = 50, \lambda = 0.001, \lambda_1 = 10^{-5}, \lambda_2 = 1.1 \cdot 10^{-5}$].

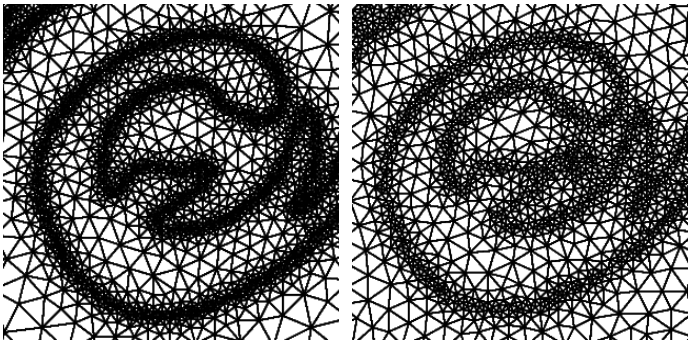
Iso vs Aniso

Come detto in precedenza, attraverso il parametro `aniso` di `adaptmesh` è possibile scegliere di utilizzare elementi finiti isotropi o anisotropi. Il valore di default corrisponde alla costruzione di griglie anisotrope, e gli esempi mostrati fino ad ora sono stati ottenuti in questo modo. In Tabella 5.5 sono dettagliati il numero di elementi necessari per generare delle griglie isotrope nel caso **b** per la segmentazione delle tre casi test appena visti. Si trova che i due casi sono comparabili in termini di numero di elementi. Però, come si può vedere nella Figura 5.16 la precisione nel caso di elementi finiti anisotropi è decisamente superiore.

I risultati presentati in questo paragrafo permettono di concludere che una procedura di adattazione di griglia euristica può ridurre la complessità di un problema di segmentazione risolto con il metodo agli elementi finiti.

Figura	# triangoli (aniso)	# triangoli (iso)
5.10(c)	28935	40058
5.11(c)	19779	21349
5.12(c)	47324	51234

Tabella 5.5: Confronto tra griglie isotrope e anisotrope.



(a) Griglia anisotropa (b) Griglia isotropa

Figura 5.16: Dettaglio della segmentazione dell'immagine in Figura 5.10(a).

Tale adattazione deve essere però indirizzata scegliendo i giusti valori dei parametri per il comando `adaptmesh`. Inoltre si è trovato che le griglie anisotrope sono più performanti rispetto a quelle isotrope, in quanto a numero di elementi. D'altronde uno dei motivi per cui si è seguita l'inusuale strada degli elementi finiti per la risoluzione del problema di segmentazione è proprio l'esistenza delle stime anisotrope ricordate nel Cap. 4. Vedremo nel prossimo paragrafo come il loro utilizzo possa migliorare la procedura di segmentazione, superando i limiti di `adaptmesh`.

5.4.2 Confronto tra due tecniche di adattazione

L'obiettivo dell'utilizzo di uno stimatore nel processo di adattazione della griglia dovrebbe essere duplice: aumentare la precisione del risultato calcolato e diminuire il numero di elementi. Vediamo come questo obiettivo viene raggiunto grazie all'implementazione dello stimatore a posteriori anisotropo presentato nella Sezione 3.3. Anche in questo caso, come nel paragrafo precedente, si baserà l'analisi su alcuni casi test.

Purtroppo in questa analisi ci si scontra con un limite di `adaptmesh`: se si definisce una metrica attraverso il parametro `metric`, i valori di `hmax` e `hmin` vengono considerati in maniera molto marginale. In particolare si trova che il limite inferiore per la grandezza della griglia viene talvolta ridotto di un fattore 100; non è quindi possibile effettuare un'analisi come quella del paragrafo precedente, basata sulla dimensione minima degli elementi finiti.

Si consideri ora la Figura 5.17, dove si mostra il risultato della segmentazione guidata dallo stimatore a posteriori, con una scelta della tolleranza $\tau = 50$. Si nota innanzitutto che in Figura 5.17(b) non ci sono più le macchie che invece si notavano nelle segmentazioni della Figura 5.11. Tuttavia questo aumento macroscopico di precisione è bilanciato da una maggiore difficoltà nel cogliere i dettagli, come si può vedere dal confronto delle

immagini 5.17(e) e 5.17(f). Si nota anche che gli elementi della griglia riproducono l'andamento della soluzione meglio di quelli ottenuti con il comando `adaptmesh` senza l'imposizione della metrica, come tipico delle griglie adattate con uno stimatore a posteriori.

La Tabella 5.6 evidenzia chiaramente il miglioramento fondamentale che si ottiene usando uno stimatore ZZ, cioè la riduzione di più di un ordine di grandezza nel numero di triangoli. I parametri riguardanti la dimensione minima e massima dei lati h degli elementi della griglia mostrano la grande discrepanza di dimensioni rispetto all'utilizzo di `adaptmesh` senza metrica.

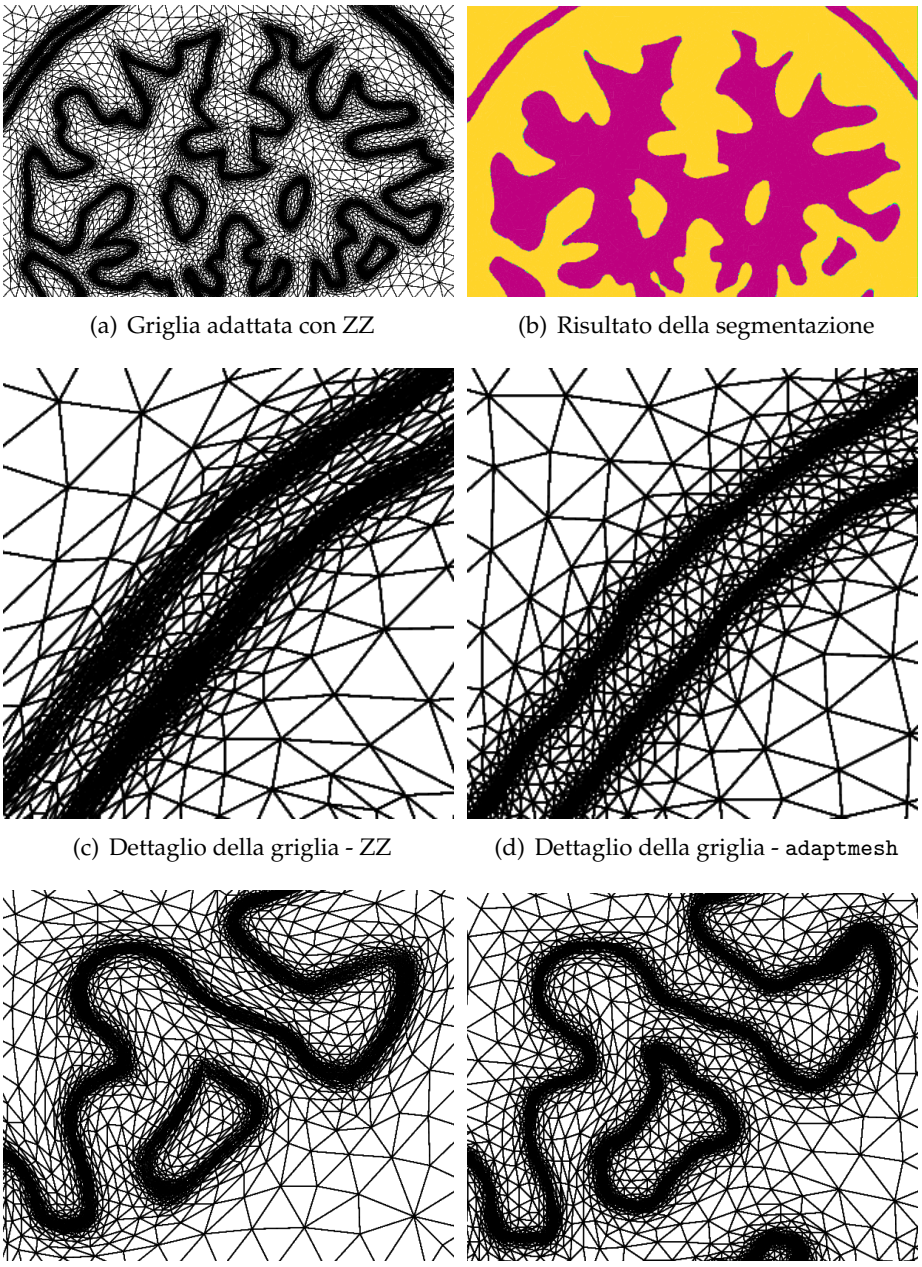
Questo importante guadagno si riflette però in un aumento del costo computazionale. Ciò avviene principalmente per una ragione: il comando `adaptmesh` è ottimizzato per avere delle buone performance sula piattaforma di `FreeFem ++`, mentre per calcolare lo stimatore a posteriori con la tecnica implementata in questa sede è necessario effettuare vari cicli su tutti gli elementi della griglia, procedura notoriamente lunga.

Risultati analoghi si trovano analizzando la Figura 5.18 e la corrispondente Tabella 5.7. Si tratta dei risultati della segmentazione della valvola cardiaca della Figura 5.9. Qui si modifica il valore della tolleranza τ dell'adattazione di griglia anisotropa per mostrare che questo parametro non influenza in maniera sostanziale il risultato.
I contorni della valvola vengono individuati in maniera corretta da tutti i casi analizzati anche se, come ci si poteva aspettare, il grado di precisione del contorno aumenta al diminuire della tolleranza τ. Come nella risonanza magnetica del cervello appena considerata, anche qui l'utilizzo dello stimatore a posteriori per l'adattazione della griglia permette di ridurre le imprecisioni della segmentazione: come si vede anche nelle rispettive griglie, l'algoritmo di limita a riconoscere i contorni della valvola, senza individuare la presenza delle pareti del cuore. Ovviamente una diversa scelta dei parametri permetterebbe l'individuazione dell'immagine completa, ma non è l'obiettivo desiderato.

In termini di numero di elementi il guadagno derivante dall'utilizzo dello stimatore ZZ è anche in questo caso rilevante, in quanto si riducono gli elementi finiti della griglia di più di un ordine di grandezza (per tutti i valori di τ considerati). Si noti che l'immagine di partenza ha una dimensione di 197×165 pixels, cioè una mesh iniziale di 65010 triangoli: mentre `adaptmesh` raddoppia il numero di elementi finiti per segmentare un pic-

Figura	# triangoli	tempo (s)	*hmin*	*hmax*
5.17(b)	50546	350	0.027	4.75
5.11(e)	1321000	62.4	0.0012	4.12

Tabella 5.6: Caso test in Figura 5.17



(a) Griglia adattata con ZZ

(b) Risultato della segmentazione

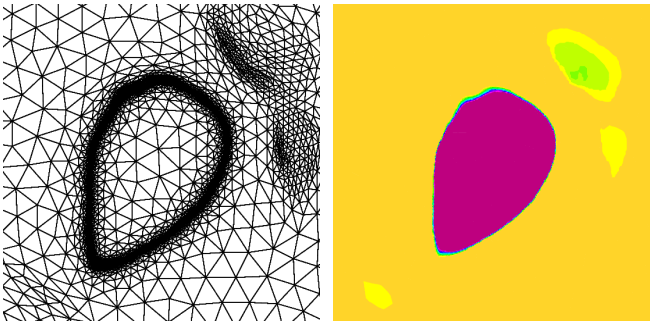
(c) Dettaglio della griglia - ZZ

(d) Dettaglio della griglia - `adaptmesh`

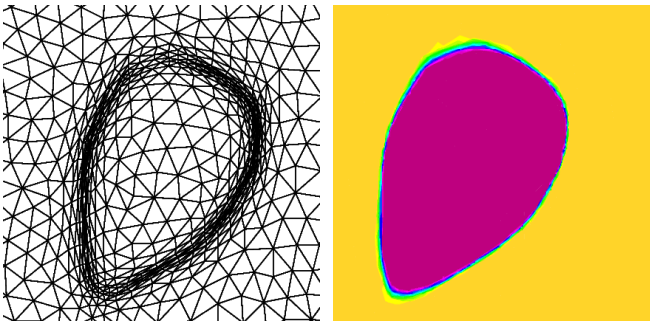
(e) Dettaglio della griglia - ZZ

(f) Dettaglio della griglia - `adaptmesh`

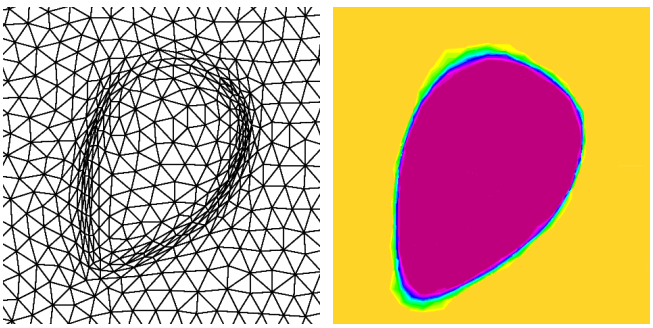
Figura 5.17: Segmentazione della Figura 5.11(a): confronto tra l'adattazione euristica e ZZ.



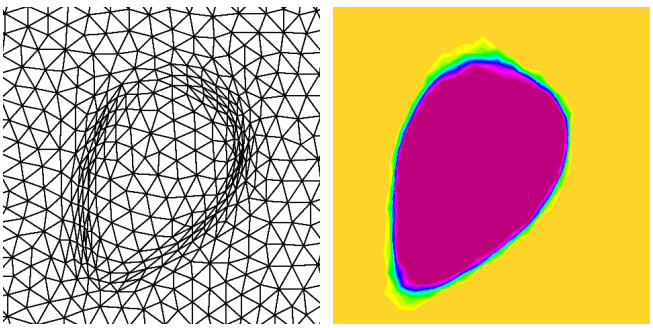
(a) Adattazione euristica (b) Adattazione euristica



(c) $\tau = 25$ - mesh (d) $\tau = 25$



(e) $\tau = 50$ - mesh (f) $\tau = 50$



(g) $\tau = 60$ - mesh (h) $\tau = 60$

Figura 5.18: Segmentazione dell'immagine mostrata in 5.9: si confronta la resa della valvola con un'adattazione euristica e al variare della tolleranza dello stimatore a posteriori.

Figura	# triangoli	tempo (s)	*hmin*	*hmax*	iterazioni
5.18(a)	116150	350	0.0046	4.33	7
5.18(c)	7202	3100	0.159	4.78	7
5.18(e)	7688	2721	0.46	4.53	7
5.18(g)	8702	2650	0.427	4.41	6

Tabella 5.7: Caso test in Figura 5.18

colo particolare dell'immagine, il nostro stimatore lo riduce di un ordine di grandezza.
Anche in questo caso il tempo computazionale richiesto per effettuare la segmentazione con lo stimatore a posteriori è molto più alto rispetto all'adattazione euristica; le ragioni sono quelle spiegate precedentemente.

I risultati sperimentali analizzati in questa sede permettono di concludere che l'adattazione di griglia guidata da uno stimatore a posteriori del tipo ZZ è estremamente efficiente in termini di riduzione del numero di elementi finiti. Per quanto riguarda la precisione, è sicuramente comparabile a quella ottenuta con l'adattazione euristica, se non superiore. Il tempo computazionale rappresenta un limite di questo approccio, anche se si è già spiegato che può sicuramente essere ridotto ottimizzando il codice (e magari usando una piattaforma diversa da `FreeFem++`).

Conclusione e sviluppi futuri

Nello sviluppo di questa tesi ci siamo prefissi come obiettivo quello di analizzare gli eventuali vantaggi apportati da tecniche di adattività di griglia ad un problema di segmentazione di immagini risolto con il metodo agli elementi finiti. Ricordiamo che si tratta di un approccio piuttosto innovativo in tale ambito; infatti non solo in letteratura sono pochi i lavori in cui si ricorre ad una adattività di griglia, ma addirittura scarseggiano gli approcci basati su una discretizzazione agli elementi finiti in generale.

Attraverso diversi casi test abbiamo verificato, in maniera qualitativa, l'effettivo interesse di un approccio adattativo per la segmentazione di un'immagine, sia dal punto di vista della precisione dei risultati così ottenuti (i contorni a tratti che seguono la geometria dei pixel tipici di un approccio senza adattività vengono completamente lisciati), sia per quanto riguarda la riduzione del costo computazionale (l'adattività di griglia, a maggior ragione se anisotropa, tende a ridurre il numero di gradi di libertà).

Nel corso dell'elaborazione di questo lavoro di tesi abbiamo trovato inoltre diversi spunti che potrebbe essere interessante sviluppare in futuro.

Un miglioramento necessario e di fondamentale importanza è rappresentato da un'ottimizzazione del nostro al fine di ridurne il costo computazionale. Abbiamo infatti riscontrato che i tempi di calcolo da questo richiesti possono essere a volte considerevoli.

Inoltre abbiamo mostrato come il modello RSFE sia estremamente sensibile ai numerosi parametri che lo descrivono. Un'indagine più approfondita sull'influenza di tali parametri potrebbe risultare utile ai fini della semplificazione o dell'automatizzazione del processo di segmentazione.

Non ultimo, potrebbe essere interessante applicare analoghe tecniche di adattività di griglia ad altri settori legati all'analisi delle immagini, così come estendere la procedura di segmentazione ad immagini 3D.

Elenco delle figure

Bibliografia

[1] R. Adams e L. Bischof, *Seeded region growing*, IEEE Trans. Pattern Anal. Mach. Intell., 1994:641–647.

[2] M. Ainsworth e J. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure and Applied Mathematics, John Wiley, 2000.

[3] L. Ambrosio, *A compactness theorem for a new class of functions of bounded variation*, Bollettino della Unione Matematica Italiana, **VII**, 1989:857–881.

[4] L. Ambrosio e V. M. Tortorelli, *Approximation of functionals depending on jumps by elliptic functionals via γ - convergence*, Comm. Pure Appl. Math., **XLIII**, 1990:999–1036.

[5] G. Aubert e L. Blanc-Féraud, *Some remarks on the equivalence between 2D and 3D classical snakes and geodesic active contours*, Int. J. Comput. Vision, **34**(1), 1999:19–28.

[6] G. Aubert e P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, vol. 147 of *Applied Mathematical Sciences*, Springer-Verlag, 2nd ed., 2006.

[7] I. Babuska e A. Aziz, *On the angle condition in the finite element method*, SIAM J. Numer. Anal., **13**(2), 1976:214–226.

[8] I. Babuška e W. C. Rheinboldt, *A-posteriori error estimates for the finite element method*, Int. J. Numer. Meth. Eng., **12**(10), 1978:1597–1615.

[9] I. Babuška e W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., **15**, 1978:736–754.

[10] I. Babuška e W. C. Rheinboldt, *Analysis of optimal finite element meshes in* $\mathbb{R}^1$, Math. Comp., **33**, 1979:435–463.

[11] R. Bank e R. Smith, *A posteriori error-estimates based on hierarchical bases*, SIAM J. Numer. Anal., **30**, 1993:921 – 935.

[12] R. Bank e A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comput., **44**, 1985:283 – 301.

[13] A. Bonnet, *On the regularity of the edge set of Mumford-Shah minimizers*, Prog. Nonlin. Diff. Equat. and Appl., **25**, 1996:93–103.

[14] C. Bottasso, G. Maisano, S. Micheletti e S. Perotto, *On some new recovery based a posteriori error estimators*, Comput. Methods Appl. Mech. Engrg., **195**(37-40), 2006:4794–4815.

[15] L. M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Comput. Maths. Math. Phys., **7**, 1967:200–217.

[16] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran e S. Osher, *Fast global minimization of the active contour/snake model*, J. Math. Imaging Vis., **28**, 2007:151–167.

[17] E. Bänsch e K. Mikula, *A coarsening finite element strategy in image selective smoothing*, Comput. Visual. Sci., **1**, 1997:53–61.

[18] V. Caselles, R. Kimmel e G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vision, **22**, 1997:61–79.

[19] F. Catte, P. L. Lions, J. M. Morel e T. Coll, *Image selective smoothing and edge detection by nonlinear diffusion*, SIAM J. Numer. Anal., **29**(1), 1992:182–193.

[20] J. Cebral e R. Lohner, *From medical images to CFD meshes*, in *Proc. 8th Int. Meshing Roundtable, South Lake Tahoe*, 1999, 321–331.

[21] A. Chambolle, R. DeVore, N. Lee e B. Lucier, *Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage*, IEEE Trans. Image Processing, **7**, 1996:319–335.

[22] T. Chan e L. Vese, *An active contour model without edges*, in *Int. Conf. Scale-Space Theories in Computer Vision*, 1999, 141–151.

[23] T. Chan e L. Vese, *A multiphase level set framework for image segmentation using the Mumford and Shah model*, Int. J. Comput. Vision, **50**(3), 2002:271–293.

[24] A. Ciampalini, P. Cignoni, C. Montani e R. Scopigno, *Multiresolution decimation based on global error*, Visual Comput., **13**, 1997:228–246.

[25] P. Clément, *Approximation by finite element functions using local regularization*, RAIRO Anal. Numér., **9**, 1975:77–84.

[26] L. D. Cohen, *On active contour models and balloons*, CVGIP: Image Underst., **53**, 1991:211–218.

[27] A. J. Cuadros-Vargas, L. G. Nonato, R. Minghim e T. Etiene, *Imesh: an image based quality mesh generation technique*, Graphics, Patterns and Images, SIBGRAPI Conference, 2005:341–348.

[28] E. De Giorgi, *Free discontinuity problems in calculus of variations*, Frontiers in Pure and Applied Mathematics Coll Pap Ded JL Lions Occas 60th Birthday, 1991:55–62.

[29] L. Demkowicz, J. Oden e T. Strouboulis, *Adaptive finite elements for flow problems with moving boundaries. I - Variational principles and a posteriori estimates*, Comput. Meth. Appl. Mech. Eng., **46**, 1984:217 – 251.

[30] D. Donoho e I. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, **81**, 1994:425–455.

[31] M. V. Dyke, *An Album of Fluid Motion*, Parabolic Press, 1982.

[32] C. Ebmeyer1 e J. Vogelgesang, *Finite element approximation of a forward and backward anisotropic diffusion model in image denoising and form generalization*, Numer. Meth. Partial Diff. Equations, **2**, 2007.

[33] V. Eijkhout e P. Vassilevski, *The role of the strengthened Cauchy-Buniakowskii-Schwarz inequality in multilevel methods*, SIAM Rev., **33**(3), 1991:405–419.

[34] M. Elad, B. Matalon, J. Shtok e M. Zibulevsky, *A wide-angle view at iterated shrinkage algorithms*, in *in SPIE (Wavelet XII)*, 2007, 26–29.

[35] P. Farrell, S. Micheletti e S. Perotto, *An anisotropic Zienkiewicz-Zhu-type error estimator for 3D applications*, Int. J. Numer. Meth. Eng., **6**(85), 2011:671–692.

[36] L. Formaggia e S. Perotto, *New anisotropic a priori error estimates*, Numer. Math., **89**(4), 2001:641–667.

[37] L. Formaggia e S. Perotto, *Anisotropic error estimates for elliptic problems*, Numer. Math., **94**(1), 2003:67–92.

[38] M. Fornasier, R. Ramlau e G. Teschke, *The application of joint sparsity and total variation minimization algorithms to a real-life art restoration problem*, Adv. Comput. Math., **31**(1-3), 2009:301–329.

[39] P. Frey e P.-L. George, *Mesh Generation: Application to Finite Elements*, John Wiley & Sons, 2 ed., 2008.

[40] M. García, A. Sappa e B. Vintimilla, *Efficient approximation of gray-scale images through bounded error triangular meshes*, in *ICIP (1)*, 1999, 168–170.

[41] T. Gevers e A. W. M. Smeulders, *Combining region splitting and edge detection through guided Delaunay image subdivision*, in *Proc. of the International Conference on Computer Vision and Pattern Recognition*, IEEE Press, 1997, 1021–1026.

[42] T. Goldstein, X. Bresson e S. Osher, *Geometric applications of the split Bregman method: segmentation and surface reconstruction*, J. Sci. Comput., **45**, 2010:272–293.

[43] T. Goldstein e S. Osher, *The split Bregman method for l^1 - regularized problems*, SIAM J. Img. Sci., **2**, 2009:323–343.

[44] L. Grady, *Random walks for image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., **28**(11), 2006:1768–1783.

[45] F. Hecht, *Freefem++*, www.freefem.org/ff++/index.htm, 2011, université Pierre et Marie Curie, Paris.

[46] S. Horowitz e T. Pavlidis, *Picture segmentation by a directed split-and-merge procedure*, in *Proc. Second Intern. Joint Conf. on Pattern Recognition*, 1974, 424–433.

[47] M. Kass, A. Witkin e D. Terzopoulos, *Snakes: active contour models*, in *First Intern. Conf. on Computer Vision*, 1987, 259–268.

[48] F. Z. Kettaf e J. P. Asselin de Beauville, *A comparison study of image segmentation by clustering techniques*, in *IEEE Intern. Conf. on Signal Processing*, vol. 2, 1996, 1280–1283.

[49] C. Li, C. Kao, J. Gore e Z. Ding, *Minimization of region-scalable fitting energy for image segmentation*, IEEE Trans. Image Process., **17**(10), 2008:1940–1949.

[50] G. Maso, *An Introduction to Γ-convergence*, Prog. Nonlin. Diff. Equat. and Appl., Birkhäuser, 1993.

[51] S. Micheletti e S. Perotto, *Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator*, Comput. Methods Appl. Mech. Eng., **9-12**(195), 2006:799–835.

[52] S. Micheletti e S. Perotto, *Anisotropic adaptation via a Zienkiewicz-Zhu error estimator for 2D elliptic problems*, in *Proceedings of ENUMATH 2009*, Springer-Verlag, 2010, 645–653.

[53] D. Mumford e J. Shah, *Optimal approximation by piecewise smooth function and associated variational problems*, Comm. Pure Appl. Math., **42**, 1989:577–684.

[54] M. Nikolova, S. Esedoglu e T. F. Chan, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., **66**(5), 2006:1632–1648.

[55] S. Osher, M. Burger, D. Goldfarb, J. Xu e W. Yin, *An iterative regularization method for total variation-based image restoration*, Simul, **4**, 2005:460–489.

[56] S. Osher e J. Sethian, *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulation*, J. Comput. Phys., **79**, 1988:12–49.

[57] D. L. Pham, C. Xu e J. L. Prince, *A survey of current methods in medical image segmentation*, Annu. Rev. Biomed. Eng., **2**, 1998s:315–338.

[58] T. Preußer e M. Rumpf, *An adaptive finite element method for large scale image processing*, J. Visual Commun. Image Represent., **11**(2), 2000:183–195.

[59] A. Quarteroni, *Modellistica Numerica per Problemi Differenziali*, Springer-Italia, 4 ed., 2008.

[60] R. Rannacher e W. Bangerth, *Adaptive Finite Element Methods for Solving Differential Equations*, Birkhäuser Verlag, 2003.

[61] C. Revol-Muller, F. Peyrin, Y. Carrillon e C. Odet, *Automated 3D region growing algorithm based on an assessment function*, Pattern Recogn. Lett., **23**(1-3), 2002:137 – 150.

[62] R. Rodríguez, *Some remarks on Zienkiewicz-Zhu estimator*, Numer. Meth. Partial Diff. Equations, **10**(5), 1994:625–635.

[63] E. Salinelli e F. Tomarelli, *Modelli Dinamici Discreti*, Springer-Italia, 2 ed., 2008.

[64] S. Salsa, *Equazioni a Derivate Parziali. Metodi, Modelli e Applicazioni*, Unitext / La Matematica Per Il 3+2, Springer Verlag, 2010.

[65] S. Sapna Varshney, N. Rajpal e R. Purwar, *Comparative study of image segmentation techniques and object matching using segmentation*, in *Proc. Intern. Conf. on Methods and Models in Computer Science*, 2009, 1–6.

[66] C. Schnörr, *A study of a convex variational diffusion approach for image segmentation and feature extraction*, J. Math. Imaging Vision, **8**, 1998:271–292.

[67] M. Sezgin e B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*, J. Electron. Imaging, **13**(1), 2004:146–168.

[68] J. Shi e J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., **22**(8), 2000:888–905.

[69] C. Steger, M. Ulrich e C. Wiedemann, *Machine Vision Algorithms and Applications*, Weinheim: Wiley-VCH, 2008.

[70] L. Tonelli, *Fondamenti di Calcolo delleVariazioni*, University Microfilms International, 1921.

[71] L. Wahlbin, *Superconvergence in Galerkin Finite Element Methods*, Lecture notes in mathematics, Springer, 1995.

[72] Y. Yang, C. Li, C.-Y. Kao e S. Osher, *Split Bregman method for minimization of region-scalable fitting energy for image segmentation*, in *Proceedings of the ISVC'10 (2)*, Springer-Verlag, 2010, 117–128.

[73] W. Yin, S. Osher, D. Goldfarb e J. Darbon, *Bregman iterative algorithms for l^1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., **1**, 2008:143–168.

[74] K. Zhang, L. Zhang, H. Song e W. Zhou, *Active contours with selective local or global segmentation: A new formulation and level set method*, Image Vision Comput., **28**(4), 2010:668–676.

[75] H. Zhao, T. Chan, B. Merriman e S. Osher, *A variational level set approach to multiphase motion*, J. Comput. Phys., **127**, 1996:179–195.

[76] Z. Zheng e X. Mei, *MRI head space-based segmentation for object based volume visualization*, in *Computer Science and Information Technology, 2008. ICCSIT '08. International Conference on*, 2008, 691–694.

[77] O. Zienkiewicz e J. Zhu, *A simple error estimator and adaptive procedure for practical engineering analysis*, Int. J. Numer. Meth. Eng., **24**(2), 1987:337–357.

[78] O. C. Zienkiewicz e J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique*, Int. J. Numer. Meth. Eng., **33**, 1992:1331–1364.

[79] O. C. Zienkiewicz e J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity*, Int. J. Numer. Meth. Eng., **33**, 1992:1365–1382.

[80] D. Ziou e S. Tabbone, *Edge detection techniques - an overview*, Int. J. Pattern Recognit. Image Anal., **8**, 1998:537–559.