

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE

CORSO DI LAUREA MAGISTRALE IN MATEMATICA

**Metodi di tipo level set  
per la segmentazione di immagini di nevi**

TESI DI LAUREA IN ANALISI NUMERICA

RELATRICE:

**CHIAR.MA PROF.SSA  
Elena Loli Piccolomini**

PRESENTATA DA:

**Giovanni Raccichini**

CORRELATORE:

**DOTT. Ivan Tomba**

III SESSIONE

ANNO ACCADEMICO 2012/2013

*A Bruno e Paolo*

*Ringrazio infinitamente il Prof. Massimo Ferri, per la serietà con cui mi ha sempre preso in considerazione e perché senza di lui questo lavoro non sarebbe esistito affatto.*

*Ringrazio la Prof.ssa Elena Piccolomini perché, pazientemente, mi ha accompagnato, sorretto e corretto nel percorso di questa tesi.*

*Ringrazio Ivan Tomba per il tempo speso, la collaborazione fruttuosa e le correzioni puntuali ricevute.*

*Ringrazio la mia famiglia, Caterina e i miei amici, compagni gioiosi, fedeli e infaticabili, segno di una bellezza infinita.*

# Introduzione

L'elaborazione digitale di immagini sta assumendo un ruolo sempre maggiore nelle discipline mediche. Lo studio di processi che aiutino a riconoscere regioni significative di un'immagine (la cosiddetta *segmentazione*) e a ricostruirne parti mancanti, o con disturbate da molto rumore, permette di aggiungere informazioni di notevole importanza per la diagnosi di una malattia. Lo studio di *software* in questa direzione sono quindi un aiuto per il medico per formulare diagnosi più precise.

Uno degli ambiti della medicina in cui l'elaborazione delle immagini può dare un importante contributo è la dermatologia. L'analisi morfologica di un nevo rappresenta lo strumento principale con cui si diagnostica la presenza di un melanoma della pelle. Tale analisi è caratterizzata da una forte soggettività nel giudizio: non solo si hanno segmentazioni diverse a seconda del dermatologo, ma per lo stesso nevo, segmentato dallo stesso dermatologo, a distanza di tempo, ottengono dei risultati differenti. Dunque, è auspicabile cercare un metodo automatico capace di riconoscere autonomamente i bordi di un nevo, che possa aiutare il dermatologo nella diagnosi.

Questa tesi viene incontro a questo bisogno e studia un metodo che possa, in maniera adeguata, riconoscere in modo autonomo il bordo di un nevo. A tale scopo, si è collaborato in maniera diretta con il Dipartimento di Dermatologia dell'Ospedale di Ravenna, con il Prof. Massimo Ferri e con il dott. Ivan Tomba, della ditta CA.MI., direttamente implicato nel progetto per la creazione del software finale. La difficoltà maggiore che si incontra per segmentare immagini di nevi è implementare un algoritmo che possa riconoscere intensità di colore tenui attorno al nevo, molto importanti ai fini diagnostici, e

ottenere dei risultati soddisfacenti in tempi ridotti.

Tra i metodi sviluppati negli ultimi decenni per segmentare immagini, si è focalizzata l'attenzione sui metodi *level set*. Questi si inseriscono in un panorama più ampio, ossia quello dello studio della propagazione di un fronte che si muove con velocità variabile in direzione ortogonale ad ogni suo punto. L'approccio classico a problemi di questo genere è quello di studiare l'andamento del fronte parametrizzandolo e studiando la sua evoluzione tramite la soluzione di un'equazione differenziale ordinaria detta *lagrangiana*. I primi che hanno utilizzato questo metodo per l'individuazione di oggetti presenti in un'immagine sono stati Kass, Witkin e Terzopoulos in [9]. I metodi *level set*, invece, si sono mostrati per la prima volta nel panorama dello studio numerico dell'evoluzione di un fronte nel 1988, con l'articolo [1] di Osher e Sethian. L'idea di fondo dei metodi *level set* è di rappresentare il fronte, o un'interfaccia tra due fluidi, come insieme di livello zero di una funzione definita in dimensioni maggiori, chiamata *level set function* (LSF), e di evolverla risolvendo un'equazione alle derivate parziali (PDE) detta *euleriana*. Questo approccio ha diversi vantaggi rispetto all'approccio legato alla parametrizzazione dell'interfaccia. Infatti, in questa maniera, è possibile gestire fronti con forma complessa e che modificano la propria natura topologica durante l'evoluzione. Inoltre, poiché la LSF resta una funzione su una griglia fissata durante l'evoluzione, è possibile usare metodi numerici di risoluzione efficienti. L'evoluzione della LSF è, quindi, descritta da una PDE che, a seconda delle necessità e delle caratteristiche proprie dell'equazione, può essere risolta con metodi differenti. In questa tesi si sceglie di risolverla con metodi variazionali, ossia attraverso la minimizzazione di un funzionale definito sulla LSF, per i notevoli vantaggi che presenta un tale approccio.

Durante l'evoluzione, per avere risultati stabili ed accurati, risulta però necessario mantenere la LSF vicina a una funzione distanza con segno, funzione chiave in questo tipo di metodi. Tuttavia, non è possibile ottenere una tale regolarità in modo automatico, a causa di problemi intrinseci alla formulazione euleriana dell'evoluzione del fronte; si è ricorsi così, al metodo chiamato *reinizializzazione*, con il quale si ricostruisce periodicamente durante l'evoluzione una LSF che sia una funzione distanza con segno, risolvendo

---

una ulteriore PDE. Questo processo, però, fa sorgere dei problemi non trascurabili legati alla qualità del risultato e all'efficienza dell'esecuzione.

C. Li et al. in [15] propongono un nuovo metodo variazionale, detto *distance regularized level set evolution* (DRLSE), capace di mantenere la LSF approssimata alla funzione distanza permettendo di evitare i problemi sorti con la reinizializzazione.

In questa tesi viene proposto un software, già precedentemente sviluppato da C. Li in [16]. Il *software* è stato applicato qui per la prima volta per segmentare immagini di nevi. Si è lavorato e collaborato con il dott. Ivan Tomba e il prof. Massimo Ferri per migliorare la sua efficienza ed adattabilità: si è cercata la migliore combinazione di parametri per ottenere la segmentazione più efficace possibile in un tempo di esecuzione ridotto e applicabile a un numero di nevi molto elevato. Infatti, uno dei punti di forza di questo software è il comportamento uniforme della segmentazione al variare delle caratteristiche particolari di un nevo: trovando il settaggio adeguato dei parametri, si ottiene una segmentazione esatta sulla stragrande maggioranza dei nevi analizzati.

Nel *Capitolo 1* di questa tesi si descrivono e introducono in modo generale i metodi usati per studiare l'evoluzione di un fronte: si descrive prima la formulazione *lagrangiana* del moto di un fronte e in seguito quella implicita, *euleriana*, caratterizzante i metodi *level set*, dell'evoluzione del fronte mostrandone i punti di interesse.

Nel *Capitolo 2* si prendono in considerazione nel dettaglio gli algoritmi proposti in [15] e in [16]: entrambi formulati in un *framework* variazionale, rappresentano un esempio, rispettivamente, di EBM e RBM. In questi si può individuare un termine comune che regolarizza la LSF e un altro, da definire in base al metodo che si desidera implementare, che guida l'evoluzione della curva verso i bordi degli oggetti. È proprio quest'ultimo l'oggetto matematico in cui si attuano le diverse tecniche di segmentazione.

Nel *Capitolo 3* vengono esposte le prove fatte su nevi campione, applicando il metodo esposto in [16]: si identificano i parametri modificabili nell'algoritmo e gli ordini di grandezza per ottenere dei cambiamenti sensibili nella segmentazione. Infine si propone una combinazione di parametri per avere la segmentazione più accurata possibile del nevo e si confronta il risultato ottenuto con la segmentazione ottenuta tramite *thresholding*.

# Capitolo 1

## Metodi *level set* e il problema del riconoscimento di bordi

I metodi *level set* sono stati introdotti da Osher e Sethian in [1] nell'ambito dello studio della propagazione di un fronte sottoposto a una velocità di propagazione ortogonale ad ogni suo punto e dipendente dalla curvatura del fronte stesso.

I testi e gli articoli di riferimento che si sono presi in considerazione per l'introduzione e la trattazione di questo argomento sono principalmente:

- *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, pubblicato in *Journal of Computational Physics* di S. Osher e J.A. Sethian nel 1988 [1];
- *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science* di J. A. Sethian pubblicato nel 1999 [7];
- *Level Set Methods and Dynamic Implicit Surfaces* di S. Osher e R. Fedkiw pubblicato nel 2003 [5];
- *Mathematical Problems in Image Processing* di G. Aubert e P. Kornprobst pubblicato nel 2002 [17].

## 1.1 Formulazione *lagrangiana* della propagazione di un fronte

Si vuole studiare l'evoluzione di un fronte che si muove con una velocità ortogonale ad ogni punto del fronte con un'intensità che, in generale, è funzione della curvatura. D'ora in poi, per semplicità, considereremo curve in  $\mathbb{R}^2$ . Per perseguire tale scopo il metodo più naturale e immediato è quello di cercare una parametrizzazione del fronte ad ogni istante dell'evoluzione.

Sia  $I = [0, 1] \subset \mathbb{R}$  un intervallo di  $\mathbb{R}$  e sia

$$\Gamma(\cdot, t) : I \longrightarrow \mathbb{R}^2 \quad (1.1)$$

$\forall t \in \mathbb{R}^+$  una parametrizzazione di una curva nel piano e supponiamo che  $\Gamma(\cdot, t)$  sia

- $C^0$  rispetto alla variabile della parametrizzazione, che denotiamo con  $q$ ;
- almeno  $C^1$  nella variabile temporale  $t$ ;
- una curva semplice e chiusa ad ogni istante dell'evoluzione, i.e. semplice  $\forall t \in \mathbb{R}^+ \quad \forall q_1, q_2 \in I$  se  $\Gamma(q_1, t) = \Gamma(q_2, t)$  allora  $q_1 = q_2$  e chiusa  $\Gamma(0, t) = \Gamma(1, t) \quad \forall t \in \mathbb{R}^+$ .

Definiamo il sostegno di una curva a un certo tempo  $t$ , l'insieme

$$\Gamma(t) := \{\Gamma(q, t) | q \in I\} \quad (1.2)$$

La curva dunque è la rappresentazione formale di un fronte a un certo istante di tempo  $t$ . Data un'interfaccia, rappresentata dalla curva iniziale  $\Gamma_0(q) = \Gamma(q, 0)$ , si vuole ottenere una rappresentazione della sua evoluzione nel tempo, sapendo che si muove con una velocità:

- diretta nella direzione normale  $\mathbf{N}$  a ogni punto della curva;
- di intensità  $F$  dipendente dal dato del problema.



Le condizioni sopraelencate si traducono, dunque, nella seguente equazione differenziale ordinaria

$$\begin{cases} \frac{\partial \Gamma(q, t)}{\partial t} = F\mathbf{N} \\ \Gamma(q, 0) = \Gamma_0(q) \end{cases} \quad (1.3)$$

comunemente chiamata *lagrangiana* dell'equazione dell'evoluzione del fronte. Ovviamente, quando si implementa numericamente questo metodo, occorre suddividere il dominio in intervalli ai quali corrisponderanno dei punti sulla curva: si scelgono dei punti, chiamati *marker points*, che verranno interpolati ad ogni passo dell'evoluzione per avere la forma del fronte al tempo desiderato, come nella figura seguente.

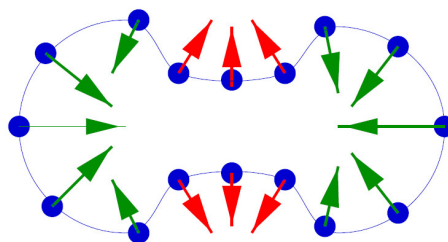


Figura 1.1: Evoluzione di un fronte: la velocità è normale in ogni punto della curva durante tutta l'evoluzione. I punti in evidenza sono i cosiddetti *marker points*

Questo approccio rivela presto la sua inefficacia per diversi motivi.

Per esempio, prendiamo in considerazione un fronte spigoloso come nella figura seguente:

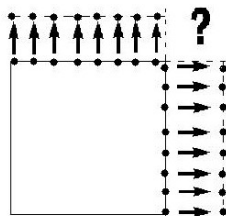


Figura 1.2: Evoluzione di un fronte: esempio di fronte che presenta punti spigolosi dei quali non si può conoscere la soluzione a causa della non derivabilità.

Supponiamo che la velocità con cui si propaga il fronte sia ortogonale alla curva e, per semplicità supponiamola in norma costante uguale a 1 diretta verso l'esterno della curva. Dopo un tempo  $t = 1$ , i punti che appartengono ai lati dello spigolo si troveranno a una distanza 1 dalla curva, ma non possiamo sapere come si evolvono i vertici, essendoci discontinuità nella derivata. Dunque non si è in grado di conoscere l'evoluzione dei punti spigolosi di un fronte, i quali possono presentarsi in maniera naturale in molti studi fisici e matematici, anche tra i più semplici.

Un altro problema è legato alla parametrizzazione del fronte, che, non essendo univoca, può implicare soluzioni diverse a uno stesso problema.

Inoltre, per affrontare il problema relativo al moto di un fronte, occorre usare un metodo che sia in grado di gestire possibili cambi di topologia, come per esempio il caso in cui siano presenti due fiamme separate all'inizio ma che si uniscono con lo scorrere del tempo (figura (1.3)), o al contrario, una bolla che nel tempo si divida formandone due. Tuttavia, in questi casi, il metodo che si sta prendendo in considerazione non è capace di esprimere in modo esatto l'evoluzione di un fronte, in quanto sarebbe necessario riconoscere i punti sovrapposti, rimuoverli e ridefinire il nuovo bordo della curva ogni volta che ci si trovi in una situazione del genere. Risulta evidente che questa operazione è troppo onerosa e complessa, se non impossibile, da attuare, il che ha fatto cercare nuove strade per studiare questo problema.

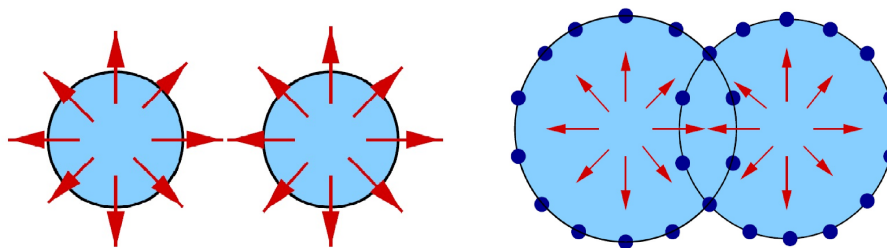


Figura 1.3: Cambio di topologia: simili modifiche della natura del fronte sono molto frequenti e naturali nello studio di problemi di questo genere.

## 1.2 Formulazione *euleriana* della propagazione di un fronte

Uno dei primi testi a dare un notevole contributo alla risoluzione delle difficoltà legate alla formulazione *lagrangiana* del moto di un fronte è stato [1], in cui Osher e Sethian propongono una strada alternativa a quella esposta nella sezione precedente, non usando più una parametrizzazione esplicita ma un'implicita del fronte, guardandolo come insieme di livello zero di una funzione in dimensione maggiore.

### 1.2.1 Una rappresentazione implicita

Si consideri una regione connessa  $\Omega \subset \mathbb{R}^2$  e una curva  $\Gamma$  con sostegno in  $\Omega$  che si evolve secondo la (1.3). Si scelga una funzione  $\phi(x, t) : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}$  detta *level set function* (LSF) caratterizzata dal fatto di avere, ad ogni istante dell'evoluzione, l'insieme di livello zero coincidente con la curva che descrive il fronte a quell'istante di tempo, cioè tale che

$$\phi(\Gamma(q, t), t) = 0 \quad \forall q \in I, \forall t \in \mathbb{R}^+.$$

Per esempio, si consideri una circonferenza di raggio  $r$  centrata nello zero che si vuole rappresentare come insieme di livello zero di una funzione da  $\mathbb{R}^2$  a  $\mathbb{R}$ . A tale scopo può essere scelta come LSF la funzione che esprime la distanza di un punto del piano dalla circonferenza (figura 1.4). Intuitivamente, una tale funzione assumerà valore zero in corrispondenza dei punti appartenenti alla circonferenza, poiché chiaramente i punti che appartengono alla circonferenza sono a distanza zero dalla circonferenza stessa; assegnando poi segno positivo ai valori della distanza dei punti all'esterno della circonferenza e negativo a quelli all'interno, si ottiene una caratterizzazione delle aree del piano tramite la positività della funzione.

Dunque, come suggerisce l'esempio, in generale una funzione adatta per descrivere una

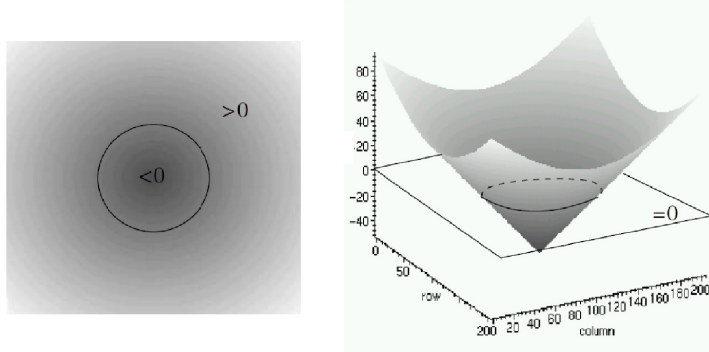


Figura 1.4: *Level set function*: se si vuole rappresentare la circonferenza in  $\mathbb{R}^2$  si può considerare la funzione distanza con segno, questa infatti interseca il piano  $x - y$ , come si vede sopra, proprio nella circonferenza.

curva nel piano in modo implicito è la *funzione distanza con segno* dal sostegno di una curva data.

Per i nostri scopi si focalizza l'attenzione sulla definizione della funzione distanza da un insieme solo nel caso particolare in cui esso sia il sostegno di una curva.

**Definizione 1.1.** Sia  $\Gamma$  il sostegno di una curva come in (1.2), definiamo la *funzione distanza di un punto da  $\Gamma$*  come

$$\begin{aligned} dist(\cdot, \Gamma) : \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ dist(x, \Gamma) &= \inf_{y \in \Gamma} d(x, y) \end{aligned} \tag{1.4}$$

con  $d(x, y)$  la distanza euclidea sul piano.

Se una curva  $\Gamma(\cdot)$  è chiusa il supporto è certamente un compatto; dunque, per il teorema di Weierstrass, si ha che  $\exists x_m \in \Gamma$  t.c.  $dist(x, \Gamma) = d(x, x_m)$ . Pertanto si ha

$$\begin{aligned} dist(\cdot, \Gamma) : \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ dist(x, \Gamma) &= \min_{y \in \Gamma} d(x, y) \end{aligned} \tag{1.5}$$

Da questa definizione si deduce in modo diretto che  $dist(x, \Gamma) = 0 \forall x \in \Gamma$ .

Si definisce ora la *funzione distanza con segno*:

**Definizione 1.2.** Sia  $\Gamma$  come nella definizione (1.1), allora la funzione  $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$  tale che

$$\begin{cases} \rho(x) = 0 & \forall x \in \Gamma \\ \rho(x) = -\text{dist}(x, \Gamma) & \forall x \in \text{int}(\Gamma) \\ \rho(x) = \text{dist}(x, \Gamma) & \forall x \in \text{ext}(\Gamma) \end{cases} \quad (1.6)$$

con  $\text{int}(\Gamma)$  la parte di piano delimitata all'interno dalla curva e  $\text{ext}(\Gamma) = \mathbb{R}^2 \setminus \overline{\text{int}(\Gamma)}$ , è detta *funzione distanza con segno*.

La suddivisione del piano in due componenti connesse  $\text{int}(\Gamma)$  e  $\text{ext}(\Gamma)$  è assicurata dal *teorema della curva di Jordan* che afferma che *una curva, chiusa e semplice, divide il piano in due parti: una limitata, interna alla curva, e un'altra illimitata, esterna ad essa*.

Enunciamo e osserviamo alcune proprietà della funzione definita in (1.6).

Può succedere che ci siano dei punti del piano che abbiano uguale distanza dal sostegno della curva, tali punti formano lo *scheletro* di  $\Gamma$ :

**Definizione 1.3.** Sia  $\Gamma$  come nella definizione (1.1), si definisce lo *scheletro* di  $\Gamma$  come

$$S = \{x \in \mathbb{R}^2 \mid \text{card} \{y \in \Gamma \mid d(x, y) = \text{dist}(x, \Gamma)\} > 1\} \quad (1.7)$$

Si hanno i seguenti risultati sulla  $\rho$ :

**Proposizione 1.2.1.**

- (i) La funzione distanza con segno,  $\rho$ , da una curva chiusa  $\Gamma$  in  $\mathbb{R}^2$  è lipschitziana con costante di Lipschitz 1;
- (ii) Se  $\rho$  è una funzione distanza con segno da una curva chiusa  $\Gamma$  in  $\mathbb{R}^2$  e  $\rho \in C^1(\mathbb{R}^2 \setminus \overline{S}, \mathbb{R})$ , allora si ha che  $\forall x \in \mathbb{R}^2 \setminus \overline{S}$

$$|\nabla \rho| = 1 \quad (1.8)$$

*Dimostrazione.*

(i) Essendo  $\rho(x, \Gamma)$  il minimo della distanza tra  $x$  e tutti gli  $y \in \Gamma$  si ha che

$$\rho(x, \Gamma) \leq d(x, y) \quad \forall y \in \Gamma$$

Consideriamo  $x, x' \in \mathbb{R}^2$ , per la disuguaglianza triangolare si ha che

$$d(x, y) \leq d(x, x') + d(x', y)$$

e dunque, considerando l'estremo inferiore per  $y \in \Gamma$  di  $d(x, y)$  e portando al primo membro il termine  $d(x', y)$ , si ha

$$\rho(x, \Gamma) - d(x', y) \leq d(x, x')$$

Tale disuguaglianza è vera  $\forall y \in \Gamma$ , perciò, resta vera anche per i punti in cui la distanza è minima, ossia vale che

$$\rho(x, \Gamma) - \rho(x', \Gamma) \leq d(x, x')$$

Dunque

$$\|\rho(x, \Gamma) - \rho(x', \Gamma)\| \leq \|x - x'\|$$

ovvero  $\rho$  è lipschitziana con costante di Lipschitz uguale a 1.

(ii) Per dimostrare questo punto si mostra che il  $|\nabla\rho| \leq 1$  e che  $|\nabla\rho| \geq 1$ .

Si dimostra innanzitutto che  $|\nabla\rho(x, \Gamma)| \geq 1$ .

Sia  $x \in \mathbb{R}^2 \setminus \overline{S}$  allora, per la definizione di scheletro,  $\exists! \bar{x} \in \Gamma$  t.c.  $\rho(x, \Gamma) = d(x, \bar{x})$ . Si calcola la derivata direzionale della  $\rho$  nella direzione  $v = \bar{x} - x$  e usando l'osservazione appena fatta si ottiene

$$\frac{\rho\left(x + h \frac{\bar{x} - x}{\|\bar{x} - x\|}, \Gamma\right) - \rho(x, \Gamma)}{h} = \frac{\|x + h \frac{\bar{x} - x}{\|\bar{x} - x\|} - \bar{x}\| - \|x - \bar{x}\|}{h} \quad (1.9)$$

dunque

$$\frac{\|(x - \bar{x})(\|x - \bar{x}\| - h)\| - \|x - \bar{x}\|^2}{h\|x - \bar{x}\|} = \frac{\|x - \bar{x}\|^2 + |h|\|x - \bar{x}\| - \|x - \bar{x}\|^2}{h\|x - \bar{x}\|} = \frac{|h|}{h} \quad (1.10)$$

dunque certamente si avrà

$$|\nabla\rho(x, \Gamma)| \geq 1 \quad (1.11)$$

Si dimostra ora che  $|\nabla\rho| \leq 1$ .

Dire che  $\rho$  è differenziabile vuol dire che comunque presi  $x, x' \in \mathbb{R}^2 \setminus \overline{S}$  si ha

$$\rho(x, \Gamma) - \rho(x', \Gamma) = \nabla\rho(x, \Gamma)(x - x') + \mathcal{O}(\|x - x'\|^2) \quad (1.12)$$

sapendo che  $\rho$  è una funzione lipschitziana con costante di Lipschitz uguale a 1, si ha che  $\forall x, x' \in \mathbb{R}^2 \setminus \overline{S}$

$$\|x - x'\| \geq \|\rho(x, \Gamma) - \rho(x', \Gamma)\| \geq \|\nabla\rho(x, \Gamma)(x - x') + \mathcal{O}(\|x - x'\|^2)\| \quad (1.13)$$

Supponiamo per assurdo che  $\|\nabla\rho\| > 1$ . Poiché la disequazione in (1.13) deve valere  $\forall x, x' \in \mathbb{R}^2 \setminus \overline{S}$ , deve valere in particolare per  $x, x'$  tali che  $\|x - x'\|$  sia sufficientemente piccolo e per  $x'$  preso nella stessa direzione del gradiente. Ma allora per questa scelta di  $x$  e  $x'$  si ha dalla (1.13) che

$$\|x - x'\| \geq \|\nabla\rho(x, \Gamma)\| \|x - x'\| \quad (1.14)$$

da cui

$$\|\nabla\rho(x, \Gamma)\| \leq 1 \quad (1.15)$$

che è in contraddizione con l'ipotesi che  $\|\nabla\rho\| > 1$ . Dunque, unendo le due parti della dimostrazione si ottiene la tesi.  $\square$

Tornando al problema legato all'evoluzione di un'interfaccia, lo scopo ora è quello di capire la legge che governa il cambiamento della LSF con lo scorrere del tempo in modo da esprimere correttamente l'evoluzione della curva sotto l'influsso di una certa velocità  $\mathbf{F}$  data.

Come già detto, per descrivere nella formulazione *level set* l'evoluzione della curva, deve essere soddisfatta la seguente condizione

$$\phi(\Gamma(q, t), t) = 0 \quad \forall t \in \mathbb{R} \quad (1.16)$$

e se  $\phi$  è sufficientemente regolare, si deriva rispetto a  $t$  ottenendo

$$\frac{\partial \phi(\Gamma(q, t), t)}{\partial t} = 0 \quad (1.17)$$

dalla quale, usando la *regola della catena*, si ha

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \frac{\partial \Gamma(q, t)}{\partial t} = 0. \quad (1.18)$$

Sostituendo in (1.18) la velocità data da (1.3) si ha

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot F\mathbf{N} = 0 \quad (1.19)$$

ma la normale, in questa formulazione, può essere espressa come  $\mathbf{N} = -\frac{\nabla \phi}{|\nabla \phi|}$  e la (1.19) può essere dunque riscritta come

$$\frac{\partial \phi}{\partial t} - F|\nabla \phi| = 0. \quad (1.20)$$

In questo modo si è ottenuta un'equazione alle derivate parziali, chiamata formulazione *euleriana* dell'evoluzione del fronte, che descrive in modo implicito, senza il bisogno di una parametrizzazione, la propagazione dell'interfaccia.

Una volta calcolata la  $\phi$  su  $\Omega \times \mathbb{R}^+$  ci sarà solo bisogno di estrarre l'insieme di livello zero per avere il bordo desiderato. Dunque il problema di conoscere l'evoluzione del fronte nel tempo si è trasformato nel problema di risolvere la PDE (1.20) su tutto lo spazio.

Ovviamente si dovrà aggiungere:

- i) una condizione sul bordo, come per esempio che la derivata lungo la normale  $\frac{\partial \phi}{\partial \mathbf{N}}$  sia uguale a zero su  $\partial\Omega$ ;
- ii) una condizione al tempo  $t=0$ : si può scegliere, come precedentemente illustrato, la funzione distanza con segno dalla curva iniziale  $\rho(\mathbf{x}, \Gamma(0))$  come definita in (1.6).



Dunque il modello finale che descrive la propagazione di un fronte è

$$\begin{cases} \frac{\partial \phi(x, t)}{\partial t} = F|\nabla \phi| & \text{per } (x, t) \in \Omega \times ]0, +\infty[ \\ \phi(x, t) = \rho(x, \Gamma(0)) & \text{per } (x, t) \in \Omega \times \{0\} \\ \frac{\partial \phi}{\partial \mathbf{N}} = 0 & \text{per } (x, t) \in \partial\Omega \times ]0, +\infty[ \end{cases} \quad (1.21)$$

### 1.2.2 Legami con l'equazione di *Hamilton-Jacobi*: proprietà e conseguenze

L'equazione (1.20) è un caso particolare dell'equazione di *Hamilton-Jacobi* che, più in generale può essere espressa come

$$\frac{\partial \phi}{\partial t}(\mathbf{x}, t) + \mathbf{H}(\mathbf{x}, \phi(\mathbf{x}), \nabla \phi(\mathbf{x})) = 0 \quad (1.22)$$

con  $\phi \in C^1(\Omega)$  e  $\mathbf{H}$  continua.

Lo studio delle soluzioni dell'equazione (1.20), quindi, è strettamente legato alle soluzioni dell'equazione di Hamilton. Perciò indagando sull'equazione (1.21) si possono trovare delle proprietà interessanti per il nostro scopo.

Seguendo la strada percorsa in [4], si ha che in generale l'equazione di *Hamilton-Jacobi* è strettamente legata all'equazione che esprime la legge di conservazione dell'energia

$$u_t + (H(u(x)))_x = 0. \quad (1.23)$$

Con dei semplici conti e sostituzioni si vede che la soluzione della (1.23) è la derivata di una soluzione dell'equazione di *Hamilton-Jacobi* e, viceversa, una soluzione dell'equazione di *Hamilton-Jacobi* è l'integrale di una soluzione della legge di conservazione. Questa corrispondenza ci dà delle informazioni in più sul problema. Per esempio, le soluzioni dell'equazione di Hamilton possono sviluppare dei punti spigolosi in quanto l'integrale di un punto di discontinuità è uno spigolo e dunque le soluzioni di (1.23) sono in generale continue. Inoltre, poiché la legge di conservazione ammette soluzioni non uniche, c'è bisogno di una condizione cosiddetta di *entropia* per scegliere la soluzione fisicamente più

rilevante.

A livello numerico, per implementare un metodo stabile per l'equazione di Hamilton (e quindi anche della (1.20)), si necessita di metodi *upwind* alle differenze finite, poiché si tratta di un'equazione alle derivate parziali iperbolica. È noto, da risultati classici, che i metodi *upwind*, per essere accurati, devono rispettare il verso dello scorrere delle informazioni nel tempo. Perciò, per avere una soluzione numerica attendibile, deve essere realizzata un'implementazione più sofisticata rispetto agli schemi diretti alle differenze finite centrali. Come espresso in [4], a cui si rimanda il lettore interessato per ulteriori dettagli, esistono almeno tre strategie numeriche per discretizzare e calcolare una equazione del tipo (1.22):

- schemi di Lax-Friedrichs;
- schemi di Roe-Fix;
- schemi di Godunov.

### 1.2.3 La reinizializzazione

Un punto cruciale per la riuscita dei metodi *level set* è di mantenere una LSF regolare lungo il tempo della trasformazione in modo da ottenere risultati accurati. Come visto nella sezione precedente, una buona funzione che assicura tale regolarità è la funzione distanza con segno. Richiedere che la LSF sia una funzione distanza con segno equivale a richiedere che

$$|\nabla\phi| = 1. \tag{1.24}$$

Infatti si ha che se vale la (1.24) allora  $\phi$  è una funzione distanza con segno e viceversa una funzione distanza con segno ha la proprietà (1.24), vedi la sezione 1.2.1. In [4] viene espresso puntualmente il perché sia cruciale questa richiesta: essa assicura la convergenza del metodo, semplifica di molto i conti e permette di identificare univocamente funzione e fronte.

Tuttavia, come mostrato in [5], la soluzione dell'equazione (1.22) non è una funzione distanza con segno. Per risolvere questo inconveniente, è stata usata una tecnica risolutiva *ad hoc*, che ridefinisce, periodicamente durante l'evoluzione, una LSF che sia la funzione distanza con segno che abbia come insieme di livello zero esattamente la curva calcolata al tempo  $t$ . Esistono diversi metodi per attuare questa tecnica, ma il più classico e utilizzato è quello proposto da Sethian in [7], dove si suggerisce di stoppare le iterazioni periodicamente e risolvere l'equazione

$$\frac{\partial \phi}{\partial t} - \text{sign}(\phi_0)(1 - |\nabla \phi|) = 0 \quad (1.25)$$

fino alla convergenza. Questo processo correttivo è chiamato *re-inizializzazione*.

Tale processo incontra presto diversi problemi non trascurabili. Sethian in [7] nota che la reinizializzazione ha l'effetto di far perdere massa (o volume) al fluido di cui stiamo calcolando l'evoluzione, mentre Gomes e Faugeras in [6] evidenziano la discrepanza che emerge tra la teoria del metodo e la sua applicazione: la soluzione dell'equazione dell'evoluzione non è la funzione distanza, ma emerge comunque la necessità di riassegnare alla LSF il valore della funzione distanza durante il calcolo per ottenere un'evoluzione regolare e con risultati soddisfacenti. Restano poi irrisolti i problemi che sorgono nello stabilire esattamente quando e come reinizializzare la funzione; il costo computazionale troppo elevato per la reinizializzazione, inoltre, fa perdere, in termini numerici, efficienza al metodo. Pertanto, come è suggerito in [7], l'uso della reinizializzazione deve essere il più ridotto possibile.

#### 1.2.4 Vantaggi e svantaggi dei metodi *level set*

Possiamo a questo punto elencare sinteticamente i vantaggi e le questioni rimaste ancora aperte della formulazione *euleriana* dello studio della propagazione di un fronte:

- i) un grande vantaggio è che la LSF rimane una funzione finché la velocità  $F$  è abbastanza regolare, ma l'insieme di livello zero, ossia il fronte, può cambiare liberamente topologia, dividersi o unirsi durante l'evoluzione;

- ii) si ha un notevole vantaggio per quanto riguarda l'approssimazione numerica in quanto si può usare una griglia fissa nel dominio spaziale e scegliere uno schema alle differenze finite per approssimare le derivate spaziali e temporali; allo stesso tempo però l'implementazione di schemi *upwind* sempre più complessi ne diminuisce la semplicità e l'efficienza;
- iii) gli elementi intrinseci alla geometria del fronte, come la curvatura e la normale, possono essere espressi facilmente in funzione della LSF;
- iv) la formulazione tramite metodi *level set* può essere estesa e ampliata senza sostanziali mutamenti a ogni dimensione. Per esempio, una superficie può essere espressa come insieme di livello zero di una funzione definita su un volume.
- vi) la reinizializzazione rimane una questione importante da risolvere: è un processo molto oneroso in termini di efficienza computazionale e evidenzia una contraddizione reale che si esprime nei fatti con una strategia *ad hoc*, poco precisa e instabile.

## 1.3 La segmentazione di immagini

Questi metodi possono essere usati, oltre che per lo studio della propagazione dei fronti in problemi di fluidodinamica e dinamica, anche per identificare i bordi di oggetti presenti in un'immagine. Ma come possono essere messi in relazione con un problema di *edge detection*?

### 1.3.1 Una breve introduzione al problema della segmentazione

Un'immagine, che supponiamo essere in scala di grigi, è definita come una funzione  $I : \mathbb{R}^2 \rightarrow \mathbb{R}$  il cui valore in ogni coppia di coordinate è chiamato *intensità dell'immagine* nel punto. Le immagini digitali subiscono una discretizzazione; in linguaggio macchina, quindi, un'immagine è identificata con una matrice che ha per ogni elemento il valore

dell'intensità in quel punto. Gli elementi della matrice vengono detti pixel. Quando si acquisisce un'immagine, si può essere interessati a particolari elementi che la compongono a seconda dello scopo che si vuole raggiungere; dunque, non tutte le regioni presenti in un'immagine hanno uguale importanza. Nel nostro caso, per esempio, un nevo risulta essere molto più importante di un pelo per perseguire i nostri scopi: si pone allora il problema di trovare un metodo numerico in grado di riconoscere automaticamente gli oggetti o le regioni più significative presenti nell'immagine. Sorge allora la domanda: è possibile suddividere un'immagine in regioni significative in modo automatico? Il processo che cerca di rispondere a questo problema viene indicato con il termine, significativo, di *segmentazione*. In particolare, in questa tesi si è interessati ad estrarre da un'immagine la forma di un nevo nel modo più fedele possibile.

### 1.3.2 Metodi di segmentazione

Gli approcci classici al problema della segmentazione sono basati sullo studio delle proprietà che caratterizzano un bordo, ovvero si analizza la variazione di intensità calcolando la derivata prima, o seconda, dell'immagine; si rimanda a [8] per un'analisi approfondita di tali metodi, i cui risultati però, si sono rivelati molto poco accurati. Nel corso degli anni se ne sono sviluppati altri che combinano diverse derivate di diverso ordine, basandosi su considerazioni locali dell'immagine. Lo scopo di questi metodi è quello di identificare i bordi che presentano variazioni nette di intensità. Possiamo suddividere questa nuova categoria di identificatori di bordi in metodi *edge-based*(EBM) e *region-based*(RBM).

***Edge based method*** Come suggerisce il nome, gli EBM sono basati sulle variazioni di colore in prossimità dei bordi degli oggetti senza interessarsi delle proprietà interne.

L'idea è quella di considerare una curva iniziale che contiene all'interno tutti gli oggetti che si vogliono identificare, e di farla evolvere, facendola restringere o espandere finché non si raggiungono i bordi ricercati. Questa idea è stata proposta e sviluppata inizialmente da Kass, Witkin e Terzopoulos in [9] dando vita a metodi conosciuti come *snakes*

o *active contour*, basati su l'evoluzione di una curva parametrizzata che si evolve in base alla minimizzazione di un'energia. In questi primi testi, seppur limitati dall'uso della parametrizzazione, è già presente l'idea di un metodo in cui avviene un'evoluzione attiva della curva tramite una formulazione variazionale del problema. Si risolve un problema di minimo considerando un particolare funzionale che guida il fronte verso i bordi mantenendo una certa regolarità alla curva. Nella formulazione del problema variazionale bisogna necessariamente considerare una funzione che agisca come un indicatore di intensità di colore, cioè una funzione che sia sensibile alla variazione dei colori: si richiede una  $g : \mathbb{R}^2 \rightarrow \mathbb{R}$  che prenda in entrata il gradiente dell'immagine  $\nabla I$  e che assuma approssimativamente dei valori costanti dove la variazione dell'intensità di colore è nulla, e che sia zero nelle regioni in cui il gradiente dell'immagine varia molto, espressione della presenza di bordi. Dunque si richiede che la  $g(s)$  sia:

- i) sempre positiva;
- ii) regolare non decrescente;
- iii)  $g(0) = 1$  e  $\lim_{s \rightarrow +\infty} g(s) = 0$ , ossia si richiede, rispettivamente, che l'indicatore sia costante uguale a uno quando il gradiente è zero, quando cioè non vi sia variazione di intensità, e valga zero in presenza di una grande variazione dell'intensità.

Classicamente si sceglie come funzione soddisfacente queste proprietà la seguente:

$$g = \frac{1}{1 + |\nabla I|^p} \quad (1.26)$$

con  $p \in \mathbb{N}$ .

**Region-based method** Mentre i metodi EBM hanno lo scopo di identificare diverse regioni di un'immagine usando dei descrittori che guidino il moto del fronte, i RBM hanno lo scopo di ottenere la segmentazione di un'immagine trovando un'immagine semplificata che sia unione di regioni con intensità costante. Due esempi classici di metodi RBM sono il metodo di Mumford-Shah[10][11] o quello di Chan-Vese [12].

---

Queste idee possono essere, esattamente come sopra, inserite in un problema variazionale: a partire da un'immagine di partenza  $I_0$ , si cerca la coppia  $(u, K)$  tale che  $u$  sia la più vicina funzione costante a tratti che approssimi  $I_0$  e  $K$  corrisponda all'insieme dei bordi che suddividono le regioni con diverse intensità. In questo modo non si ha più bisogno di imporre alcuna condizione sulla regolarità dei bordi. I metodi RBM, pur essendo più generali e precisi, hanno lo svantaggio di avere un costo computazionale più elevato di quelli EBM ed, inoltre, trovare un indicatore che descriva un criterio di classificazione delle regioni interne a un'immagine risulta essere complicato.

# Capitolo 2

## Il metodo DRSLE

Nonostante i metodi *level set* vengano usati per risolvere un grande numero di problemi nel campo della fisica e dell'ingegneria, il loro utilizzo è limitato a causa delle problematiche legate alla regolarità della LSF. I classici metodi *level set*, infatti, durante l'evoluzione sviluppano delle forti irregolarità che causano errori numerici e distruggono la stabilità dell'evoluzione. Come già visto nel capitolo precedente, per superare tale difficoltà si usa una tecnica chiamata *reinizializzazione*, la quale ricostruisce la LSF periodicamente in modo da ridare regolarità alla funzione durante l'evoluzione. Questa strategia, però, ha degli effetti indesiderati, che non permettono di raggiungere in modo accurato e efficiente la soluzione del problema che si intende risolvere. Negli ultimi anni, sono stati scritti diversi articoli che, da un lato, hanno messo in evidenza in modo puntuale queste difficoltà e, dall'altro, hanno tentato di dare una risposta adeguata.

Gomes e Faugeras in [6] mostrano tre particolare casi problematici legati alla reinizializzazione e propongono una nuova formulazione che utilizza tre PDE: una fa sì che la LSF sia una funzione distanza con segno e le altre due guidano il moto del livello zero della LSF. Tuttavia, in questo articolo è presente alcun fondamento teorico solido che assicuri l'esistenza della soluzione, inoltre, l'implementazione numerica di questa formulazione causa ancora errori che deteriorano la LSF e ne falsano gli insiemi di livello, rendendo necessaria una reinizializzazione parallela.



Weber et al. in [13] propongono una strategia per evitare proprio questo effetto collaterale appena descritto: essi suggeriscono di implementare un modello detto *geodesic active control* (GAC), nel quale l'aggiornamento della LSF viene effettuato ad ogni passo tramite una complessa procedura di risoluzione di un problema di ottimizzazione. In questo caso dunque si sceglie di risolvere un problema di ottimo invece che uno schema iterativo derivato dall'equazione dell'evoluzione, cosa che evidenzia un disaccordo tra la teoria e la pratica.

Chunming Li et al. in [14][15][16] propongono dei metodi *level set* che definiscono una formulazione variazionale nuova, mostrando una soluzione al problema della reinizializzazione. In questi metodi la regolarità della LSF è intrinsecamente mantenuta durante l'evoluzione degli insiemi di livello. Quest'ultima è ottenuta attraverso un campo vettoriale che minimizza un funzionale, definito come somma di due termini: uno, regolarizzante, mantiene la LSF prossima a una funzione distanza, l'altro, chiamato *energia esterna* guida il moto dell'insieme di livello zero verso le regioni di interesse. Il termine regolarizzante è definito da una funzione, detta *potenziale*, in modo tale che la formulazione variazionale abbia un effetto di diffusione univoco, capace di mantenere la LSF della forma desiderata. In questa tesi ci focalizzeremo proprio su questi tre articoli e in particolare su [16], articolo associato al metodo che verrà usato per segmentare i nevi.

## 2.1 Descrizione del metodo

Negli articoli di Chunming Li et al. [14] e [15] viene presentato un nuovo metodo variazionale chiamato *distance regularized level sets evolution* (DRLSE): l'idea centrale è quella di tenere conto della regolarità della LSF, oltre che della convergenza del fronte al bordo dell'oggetto desiderato, aggiungendo un particolare termine al funzionale da minimizzare, facendo sì che non ci sia più la necessità di reinizializzare la LSF.

Il funzionale considerato per risolvere la PDE (1.21) che descrive il moto di propagazione del fronte, è il seguente:

$$\mathcal{E}(\phi) = \mu \mathcal{R}_p(\phi) + \mathcal{E}_{ext}(\phi) \quad (2.1)$$

dove:

- $\mathcal{R}_p(\Phi)$  è un funzionale che esprime la regolarizzazione dei *level sets*;
- $\mu > 0$  costante;
- $\mathcal{E}_{ext}$  è un funzionale che dipende dall'immagine data e guida il moto dell'insieme di livello zero verso i bordi degli oggetti d'interesse.

Chiameremo  $\mathcal{E}_{ext}$  *energia esterna* perché, come vedremo meglio in seguito, è un funzionale che dipende in generale dall'immagine data; chiameremo invece  $\mathcal{R}_p$  *energia interna* poiché esprime una proprietà della LSF.

Il termine di regolarizzazione viene definito dal seguente funzionale:

$$\mathcal{R}_p := \int_{\Omega} p(|\nabla\phi|) dx \quad (2.2)$$

dove  $p$  è una funzione  $p : [0, \infty) \rightarrow \mathbb{R}$  che chiameremo *potenziale* o *densità*. Il termine di regolarizzazione ha un ruolo chiave in questa formulazione variazionale e rappresenta il cuore dell'innovazione dell'articolo.

A seconda della scelta del potenziale  $p$  si ottengono effetti diversi sulla LSF. Una prima scelta di  $p$  può essere  $p(s) = p_0(s) = s^2$ , che forza  $|\nabla\phi|$  verso il suo unico minimo, lo zero. Tale scelta ha un forte effetto regolarizzante, ma allo stesso tempo tende ad appiattire la LSF facendo scomparire l'insieme di livello zero.

Questo effetto non è desiderabile se si vuole segmentare un'immagine o studiare l'andamento di un fronte, in quanto non permette di ottenere dei risultati significativi. Da questo esempio si capisce che il compito del termine  $\mathcal{R}_p(\phi)$  non sarà solo quello di regolarizzare, ma anche di mantenere la proprietà che caratterizza una funzione distanza con segno, cioè  $|\nabla\phi| = 1$ , durante tutta l'evoluzione. Non si richiede, però, che la LSF sia una funzione distanza con segno su tutto il dominio, ma che lo sia perlomeno nelle vicinanze dell'insieme di livello zero, per assicurarne un calcolo accurato dell'evoluzione almeno.

Questa richiesta può essere soddisfatta usando una funzione potenziale  $p(s)$  che ha minimo in  $s = 1$ , in modo che il termine di regolarizzazione sia minimizzato quando  $|\nabla\phi| = 1$ . Una semplice definizione del potenziale  $p$  con le caratteristiche appena enunciate è

$$p = p_1(s) := \frac{1}{2}(s - 1)^2 \quad (2.3)$$

il quale ha  $s = 1$  come unico minimo. Il termine  $\mathcal{R}_p$  può essere dunque espresso esplicitamente come

$$\frac{1}{2} \int_{\Omega} (|\nabla\phi| - 1)^2 dx \quad (2.4)$$

che evidentemente esprime la deviazione della LSF  $\phi$  da una funzione distanza con segno. Il funzionale definito in (2.4) è quello proposto in [14]; esso mantiene le proprietà proprie della funzione distanza su tutto il dominio  $\Omega$ . Tuttavia, una tale scelta ha in alcuni casi un effetto collaterale sulla LSF che vedremo nello specifico più avanti. Per evitarlo viene introdotta in [15] una nuova funzione potenziale che mantenga la proprietà  $|\nabla\phi| = 1$  solo nelle vicinanze dell'insieme di livello zero e imponga che la  $\phi$  sia costante, ossia che  $|\nabla\phi| = 0$  lontano dall'insieme di livello zero. Quindi, per mantenere un tale profilo della LSF, la funzione potenziale  $p(s)$  deve avere due minimi, uno in  $s = 1$  e un altro in  $s = 0$ . Si vuole dunque minimizzare il funzionale proposto in (2.1) con le caratteristiche appena spiegate, per risolvere l'equazione (1.21). Un metodo standard per minimizzare un generico funzionale energia  $\mathcal{F}(\phi)$  (cfr. [17]) è trovare la soluzione dell'equazione

$$\frac{\partial\phi}{\partial t} = -\frac{\partial\mathcal{F}}{\partial\phi} \quad (2.5)$$

dove  $\frac{\partial\mathcal{F}}{\partial\phi}$  è la derivata di *Gateaux* del funzionale  $\mathcal{F}(\phi)$ .

La (2.5) è l'equazione dell'evoluzione nel tempo di una funzione  $\phi(x, t)$  dipendente da  $t$  e da una variabile spaziale  $x \in \Omega$ , con dato iniziale  $\phi(x, 0) = \phi_0(x)$ . L'evoluzione della funzione  $\phi(x, t)$  è nella direzione opposta della derivata di *Gateaux* ossia  $-\frac{\partial\mathcal{F}}{\partial\phi}$ , la direzione più rapida in cui il funzionale  $\mathcal{F}$  decresce.

Esplicando i calcoli, la derivata di *Gateaux* del funzionale  $\mathcal{R}_p$  può essere scritta esplicitamente come

$$\frac{\partial\mathcal{R}_p}{\partial\phi} = -\text{div}(d_p(|\nabla\phi|)\nabla\phi) \quad (2.6)$$

dove  $div(\cdot)$  indica l'operatore divergenza e

$$d_p := \frac{p'(s)}{s}. \quad (2.7)$$

Dunque da (2.1) e dalla linearità della derivata di *Gateaux* si ottiene che la derivata di *Gateaux* di  $\mathcal{E}$  è

$$\frac{\partial \mathcal{E}}{\partial \phi} = \mu \frac{\partial \mathcal{R}_p}{\partial \phi} + \frac{\partial \mathcal{E}_{ext}}{\partial \phi} \quad (2.8)$$

e dunque l'equazione dell'evoluzione associata al funzionale  $\mathcal{E}(\phi)$  è

$$\frac{\partial \phi}{\partial t} = \mu div(d_p(|\nabla \phi|)\nabla \phi) - \frac{\mathcal{E}_{ext}}{\partial \phi}. \quad (2.9)$$

La PDE (2.9) è l'equazione dell'evoluzione della funzione  $\phi$  presentata nell'articolo [15]. Il grande vantaggio di tale metodo è quello di essere intrinsecamente capace di mantenere la LSF vicino a una funzione distanza con segno a causa dell'azione del termine di regolarizzazione  $\mathcal{R}_p$ . L'equazione ottenuta in [14], invece, è un caso particolare della formulazione proposta in [15], considerando come termine di regolarizzazione quello indicato in (2.4).

L'effetto regolarizzante del DRLSE può essere visto dall'evoluzione del termine  $\mu \mathcal{R}_p(\phi)$ :

$$\frac{\partial \phi}{\partial t} = -\mu \frac{\partial \mathcal{R}_p(\phi)}{\partial \phi} = \mu div(d_p(|\nabla \phi|)\nabla \phi). \quad (2.10)$$

Questo flusso può essere espresso nella forma standard dell'equazione di diffusione

$$\frac{\partial \phi}{\partial t} = div(D\nabla \phi) \quad (2.11)$$

con un tasso di diffusione  $D = \mu d_p(|\nabla \phi|)$ . Perciò, il termine (2.10) ha un effetto diffusivo sulla LSF  $\phi$ , dipendente dal segno del potenziale  $p$ . Quando  $d_p(|\nabla \phi|)$  è positivo la diffusione è in avanti quindi  $|\nabla \phi|$  decresce, quando invece  $d_p(|\nabla \phi|)$  è negativo la diffusione è all'indietro quindi  $|\nabla \phi|$  cresce. Questa diffusione è chiamata *forward-and-backward* (FAB) ed è un tipo di diffusione *adattiva* poiché incrementa e decrementa  $|\nabla \phi|$  in modo da mantenere la  $\phi$  vicina a uno dei punti di minimo della funzione potenziale  $p(s)$ , permettendo alla LSF  $\phi$  di avere la forma desiderata.

Con il potenziale  $p = p_1(s)$  definito in (2.3), abbiamo che  $d_p(s) = 1 - (1/s)$  e in questo caso la PDE (2.9) può essere espressa come

$$\frac{\partial \phi}{\partial t} = \mu \left[ \Delta \phi - \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \right] - \frac{\partial \mathcal{E}_{ext}}{\partial \phi}. \quad (2.12)$$

È da notare che il termine  $\operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|})$  è la curvatura media degli insiemi di livello della funzione  $\phi$ .

Il segno della funzione  $d_p(s) = 1 - (1/s)$  esprime l'effetto del termine di diffusione, infatti si ha che:

- se  $|\nabla \phi| > 1$  il tasso di diffusione  $\mu d_p(|\nabla \phi|)$  è positivo e la diffusione (2.10) è in avanti, quindi  $|\nabla \phi|$  decresce;
- se  $|\nabla \phi| < 1$  il tasso di diffusione  $\mu d_p(|\nabla \phi|)$  è negativo e la diffusione (2.10) è all'indietro, quindi  $|\nabla \phi|$  cresce.

Quindi, la diffusione FAB con potenziale  $p = p_1(s)$  forza  $|\nabla \phi|$  ad avvicinarsi a 1, in modo tale da mantenere la proprietà della funzione distanza.

Tuttavia, questa diffusione FAB non è limitata, infatti il termine responsabile della diffusione  $\mu d_p(|\nabla \phi|) = \mu(1 - \frac{1}{|\nabla \phi|})$  tende a meno infinito per valori di  $|\nabla \phi|$  prossimi allo zero causando dei problemi computazionali che compromettono il risultato del metodo. Questo problema può essere evitato usando un potenziale con una doppia buca che faccia in modo che  $d_p(s)$  sia limitato sul dominio.

Per esempio se si sceglie il seguente potenziale

$$p_2(s) = \begin{cases} \frac{1}{(2\pi)^2} (1 - \cos(2\pi s)) & \text{se } 0 < s \leq 1 \\ \frac{1}{2}(s - 1)^2 & \text{se } s > 1. \end{cases}$$

si vede che esso ha due punti di minimo, in  $s = 0$  e in  $s = 1$ . Si verifica che  $p_2$  è due volte differenziabile in  $[0, \infty)$ , con derivata prima e seconda pari a

$$p_2'(s) = \begin{cases} \frac{1}{(2\pi)} (\sin(2\pi s)) & \text{se } 0 < s \leq 1 \\ s - 1 & \text{se } s > 1 \end{cases}$$

e

$$p_2''(s) = \begin{cases} \cos(2\pi s) & \text{se } 0 < s \leq 1 \\ 1 & \text{se } s > 1. \end{cases}$$

Dunque con un potenziale così definito si ha che

$$|d_p(s)| = \left| \frac{p'(s)}{s} \right| = \begin{cases} \left| \frac{\sin(2\pi s)}{2\pi s} \right| & \text{se } 0 < s \leq 1 \\ \left| \frac{s-1}{s} \right| & \text{se } s > 1 \end{cases}$$

da cui è evidente che

$$|d_p(s)| \leq 1 \quad \forall s \in (0, \infty) \quad (2.13)$$

e che

$$\lim_{s \rightarrow 0} d_p(s) = \lim_{s \rightarrow +\infty} d_p(s) = 1. \quad (2.14)$$

Dunque si ha

$$|\mu d_p(|\nabla\phi|)| \leq \mu \quad (2.15)$$

ovvero il tasso di diffusione per il potenziale  $p_2(s)$  è limitato per ogni valore di  $|\nabla\phi|$ . Grazie a questa proprietà si risolve il problema sorto con la scelta del potenziale  $p_1(s)$  visto in precedenza.

Vediamo come questo nuovo potenziale influisce sul comportamento del segno della funzione  $d_p(s)$  :

- se  $|\nabla\phi| > 1$  il tasso di diffusione  $\mu d_p(|\nabla\phi|)$  è positivo e la diffusione in (2.10) è all'avanti, quindi  $|\nabla\phi|$  decresce;
- se  $\frac{1}{2} < |\nabla\phi| < 1$  il tasso di diffusione  $\mu d_p(|\nabla\phi|)$  è negativo e la diffusione diventa all'indietro, dunque  $|\nabla\phi|$  cresce;
- se  $|\nabla\phi| < \frac{1}{2}$  il tasso di diffusione  $\mu d_p(|\nabla\phi|)$  è positivo quindi la diffusione è all'avanti dunque  $|\nabla\phi|$  decresce.

Perciò, in ultima analisi, si può affermare che la differenza fondamentale tra la diffusione con il potenziale  $p_1$  e quella con il potenziale  $p_2$  è che con la seconda scelta, il tasso di diffusione  $\mu d_p(s)$  è limitato per ogni valore di  $|\nabla\phi|$ , risultato che permette di non perdere precisione durante tutta l'evoluzione.

## 2.2 Applicazione del metodo DRLSE alla segmentazione di un'immagine

Nella sezione precedente è stata sviluppata una formulazione generale del metodo DRLSE senza entrare nella definizione dettagliata del termine  $\mathcal{E}_{ext}$ . È possibile applicare tale metodo alla segmentazione di un'immagine definendo l'energia esterna  $\mathcal{E}_{ext}$  in (2.1) in modo tale che sia possibile guidare l'evoluzione del fronte verso i bordi di un oggetto presente in un'immagine. Ovviamente, a seconda della definizione del funzionale si otterranno metodi di segmentazione diversi. Vediamo qui di seguito due metodi, presentati in [15] e [16], che fanno uso del metodo DRLSE.

### 2.2.1 *Edge-based method*

In [15] il metodo utilizzato per la segmentazione è un *edge-based method*. Si considera la funzione già definita nel capitolo precedente che ha la caratteristica di assumere valori sempre positivi e prossimi allo zero nelle vicinanze di un bordo

$$g := \frac{1}{1 + |\nabla G_\sigma * I|^2} \quad (2.16)$$

con  $G_\sigma$  nucleo gaussiano,  $\sigma$  la deviazione standard e  $I$  la matrice dell'immagine data. La convoluzione con la gaussiana serve per ridurre l'eventuale rumore presente.

Definito un tale indicatore, si definisce un'energia esterna per la LSF  $\phi$

$$\mathcal{E}_{ext}(\phi) := \lambda \mathcal{L}_g(\phi) + \alpha \mathcal{A}_g(\phi) \quad (2.17)$$

con  $\lambda > 0$ ,  $\alpha \in \mathbb{R}$  e i funzionali  $\mathcal{L}_g(\phi)$  e  $\mathcal{A}_g(\phi)$  definiti come segue

$$\mathcal{L}_g(\phi) := \int_{\Omega} g \delta(\phi) |\nabla \phi| dx \quad (2.18)$$

e

$$\mathcal{A}_g(\phi) := \int_{\Omega} g H(-\phi) dx \quad (2.19)$$

con  $\delta$  la delta di Dirac e  $H$  la funzione di Heaviside.

Il termine  $\mathcal{L}_g(\phi)$  calcola l'integrale curvilineo della funzione  $g$  lungo l'insieme di livello zero di  $\phi$ ; infatti, parametrizzando l'insieme di livello zero con la curva  $\mathcal{C} : [0, 1] \rightarrow \Omega$ , il termine  $\mathcal{L}_g(\phi)$  può essere espresso con il seguente integrale curvilineo  $\int_0^1 g(\mathcal{C}(s)) \|\mathcal{C}'(s)\| ds$ , il quale calcola la lunghezza dell'insieme di livello zero nella metrica  $dq = g(\mathcal{C}(s)) \|\mathcal{C}'(s)\| ds$ . La presenza della funzione  $g$  in  $\mathcal{L}(\phi)$  fa sì che questo funzionale sia minimizzato quando l'insieme di livello zero della LSF è situato nei pressi del bordo di un oggetto.

Il termine  $\mathcal{A}_g(\phi)$  è stato introdotto invece per accelerare l'evoluzione dell'insieme di livello zero dal momento che l'insieme di livello zero della  $\phi$  può essere inizialmente molto distante dal bordo dell'oggetto desiderato. Esso misura l'area della regione  $\Omega_{\phi}^- := \{x : \phi(x) < 0\}$  con un peso che dipende dalla funzione  $g$ . Tale fatto è evidente se consideriamo il caso particolare in cui  $g = 1$ : il termine  $\mathcal{A}_g(\phi)$  diventa  $\int_{\Omega} H(-\phi) dx = \int_{\Omega_{\phi}^-} dx$ , ossia esattamente la misura dell'area di  $\Omega_{\phi}^-$ . La funzione  $g$  agisce in  $\mathcal{A}_g(\phi)$  come un peso che rallenta l'evoluzione del fronte nelle vicinanze di un bordo, poiché in presenza del confine di un oggetto essa assume valori molto vicini allo zero. Il ruolo del segno del coefficiente  $\alpha$  in (2.17) invece, è fondamentale perché usando una LSF che assume valori positivi fuori dell'insieme di livello zero e negativi all'interno, il segno di  $\alpha$  fa in modo che il fronte si espanda o si restringa: se  $\alpha > 0$  il fronte si restringe, mentre se  $\alpha < 0$  si espande.

In definitiva, calcolando esplicitamente la derivata di *Gateaux*, l'equazione del moto derivante dalla seguente formulazione dell'energia esterna sarà

$$\frac{\partial \phi}{\partial t} = g |\nabla \phi| \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) + \nabla g \cdot \nabla \phi + \alpha g |\nabla \phi|. \quad (2.20)$$



### 2.2.2 Region-based method

Il metodo proposto in [16] è un *region-based method*. Un'immagine acquisita  $I$ , con diversi metodi come la TAC, MRI o con fotocamera, può essere modellizzata nella seguente maniera

$$I = bJ + n \quad (2.21)$$

dove  $J$  è l'immagine vera,  $b$  è la componente che tiene conto della disomogeneità dell'intensità dovuta a fattori esterni chiamato *bias field* (BF), e  $n$  è il rumore. L'immagine vera  $J$  misura una proprietà fisica intrinseca degli oggetti che saranno mostrati nell'immagine e può essere considerata approssimativamente una funzione costante a tratti. Si suppone inoltre che il BF vari lentamente e il rumore  $n$  possa essere assunto come rumore gaussiano con media zero.

Il metodo è basato perciò sulle seguenti assunzioni:

- il BF varia lentamente e può essere perciò approssimato con una costante in un intorno di ogni punto del dominio dell'immagine;
- l'immagine vera  $J$ , per quanto detto sopra, assume  $N$  valori distinti  $c_1, \dots, c_N$  rispettivamente in  $N$  regioni disgiunte  $\Omega_1, \dots, \Omega_N$  tali che  $\{\Omega_i\}_{i=1}^N$  formino una partizione del dominio dell'immagine.

Facendo tali assunzioni viene proposto un metodo per stimare le regioni  $\{\Omega_i\}_{i=1}^N$ , le costanti  $\{c_i\}_{i=1}^N$  e il BF  $b$ . In [16] si implementa un metodo che permette di ottenere contemporaneamente la segmentazione dell'immagine e la stima del BF. Un altro vantaggio del metodo proposto è quello di poter gestire anche zone dell'immagine con intensità disomogenea, problematica non di seconda importanza nell'elaborazione di immagini mediche.

Consideriamo un intorno sferico di raggio  $\rho$  con centro in  $y \in \Omega$

$$\mathcal{O}_y := \{x : |x - y| \leq \rho\}.$$

Considerando la partizione  $\{\Omega_i\}_{i=1}^N$  del dominio  $\Omega$  si ottiene ovviamente anche una partizione degli intorni  $\{\mathcal{O}_y \cap \Omega_i\}_{i=1}^N$ . Avendo assunto che il BF vari lentamente, possiamo

dire che nell'intorno sferico si avrà una piccola variazione del valore di  $b$ , dunque si può affermare che

$$b(x) \approx b(y) \quad \forall x \in \mathcal{O}_y \quad (2.22)$$

Così le intensità  $b(x)J(x)$  in ogni sottoinsieme  $\mathcal{O}_y \cap \Omega_i$  assumono valori costanti  $b(x)c_i$ , ossia

$$b(x)J(x) \approx b(y)c_i \quad \forall x \in \mathcal{O}_y \cap \Omega_i. \quad (2.23)$$

Allora, sostituendo tali grandezze in (2.21), si ha

$$I(x) \approx b(y)c_i + n(x) \quad \forall x \in \mathcal{O}_y \cap \Omega_i \quad (2.24)$$

dove  $n(x)$  è il rumore additivo gaussiano con media nello zero. Dunque, le intensità nell'insieme  $\mathbf{I}_y^i = \{I(x) \in \mathcal{O}_y \cap \Omega_i\}$  formano un *cluster* con centro  $m_i \approx b(y)c_i$ , che può essere considerato come la media di una distribuzione gaussiana. Di conseguenza, gli  $N$  *cluster*  $\mathbf{I}_y^1, \dots, \mathbf{I}_y^N$ , sono tutti distinti con valori distinti di  $m_i \approx b(y)c_i$ . Il rumore  $n(x)$  può essere considerato piccolo abbastanza da non influire sui valori dei centri del *cluster*. Dunque, dalla proprietà appena enunciata può essere applicato un metodo di classificazione standard come il *K-means clustering*, che serve a classificare le intensità locali. Si considera quindi l'algoritmo *K-means* che può essere scritto nella formulazione continua come

$$F_y = \sum_{i=1}^N \int_{\mathcal{O}_y} |I(x) - m_i|^2 u_i(x) dx \quad (2.25)$$

con  $u_i$  la funzione caratteristica di  $\Omega_i$  e  $m_i \approx b(y)c_i$ . Possiamo perciò riscrivere la (2.25) come

$$F_y = \sum_{i=1}^N \int_{\mathcal{O}_y \cap \Omega_i} |I(x) - b(y)c_i|^2 dx. \quad (2.26)$$

Si può ora definire un criterio per classificare le intensità in  $\mathcal{O}_y$  come

$$\mathcal{G}_y = \sum_{i=1}^N \int_{\mathcal{O}_y} K(x-y) |I(x) - b(y)c_i|^2 dx. \quad (2.27)$$

con  $K(x-y)$  funzione non negativa tale che  $K(x-y) = 0$  per  $x \notin \Omega_y$ . La funzione definita in (2.27) è l'elemento fondamentale di questo RBM; essa classifica le intensità in  $\mathcal{O}_y$  date

dalla partizione  $\{\mathcal{O}_y \cap \Omega_i\}_{i=1}^N$ : più il valore di  $\mathcal{G}_y$  è piccolo e migliore è la classificazione. Cercheremo dunque la migliore partizione  $\{\Omega_i\}_{i=1}^N$  del dominio  $\Omega$  come quella per cui la funzione definita in (2.27) è minimizzata per ogni  $y \in \Omega$ . Per minimizzare (2.27) su tutto il dominio definiamo perciò

$$\mathcal{G} := \int_{\Omega} \left( \sum_{i=1}^N \int_{\Omega_i} K(y-x) |I(x) - b(y)c_i|^2 dx \right) dy \quad (2.28)$$

La segmentazione dell'immagine e il BF possono essere trovate minimizzando il funzionale (2.28) rispetto alle regioni  $\Omega_1, \dots, \Omega_N$ , alle costanti  $c_1, \dots, c_N$  e il BF  $b$ . La scelta della funzione  $K$  è molto flessibile, infatti le uniche condizioni imposte su tale scelta sono

$$\begin{cases} K(x) = a > 0 & \text{per } |x| \leq \rho \\ K(x) = 0 & \text{per } |x| > \rho \\ \int_{\Omega} K(x) = 1 \end{cases} \quad (2.29)$$

Qui viene scelta come  $K$  una gaussiana troncata definita come segue:

$$\begin{cases} \frac{1}{a} e^{-\frac{|x|^2}{2\sigma^2}} & \text{per } |x| \leq \rho \\ 0 & \text{per } |x| \geq \rho \end{cases} \quad (2.30)$$

dove  $a$  è la costante di normalizzazione dell'integrale  $\int_{\Omega} K(x) = 1$ ,  $\sigma$  la deviazione standard e  $\rho$  il raggio dell'intorno  $\mathcal{O}_y$ . Il raggio  $\rho$  va scelto in base al grado di disomogeneità dell'immagine. Il problema che sorge nella minimizzazione di (2.28) è quello di trovare un identificatore delle regioni  $\{\Omega_i\}_{i=1}^N$ , in modo tale da convertire il problema della minimizzazione in un problema *level set*, ossia cercando di identificare le  $N$  aree desiderate con  $N$  LSF differenti. Infatti come visto in precedenza, una LSF ha la proprietà di definire attraverso il suo segno una partizione del dominio in due regioni distinte. Se  $N = 2$  il metodo è chiamato *bifase*, se  $N > 2$  è detto *multifase*. Si prende in considerazione il metodo bifase (quello multifase è leggermente differente, ma il punto fondamentali è lo

stesso di quello bifase). Si è visto che se si prende una LSF  $\phi$  questa suddivide  $\Omega$  in

$$\begin{aligned}\Omega_1 &= \{x : \phi(x) > 0\} \\ \Omega_2 &= \{x : \phi(x) < 0\}\end{aligned}$$

che rappresenta una partizione del dominio. Le regioni  $\Omega_1$  e  $\Omega_2$  possono essere rappresentate attraverso le loro funzioni caratteristiche

$$\begin{aligned}\chi_1(\phi) &= H(\phi) \\ \chi_2(\phi) &= 1 - H(\phi)\end{aligned}$$

con  $H$  la funzione di Heaviside. Tenendo conto di quanto appena definito, il funzionale (2.28) può essere riscritto come

$$\mathcal{G}(\phi, \mathbf{c}, b) := \int_{\Omega} \left( \sum_{i=1}^N \int_{\Omega_i} K(y-x) |I(x) - b(y)c_i|^2 \chi_i(\phi(x)) dx \right) dy \quad (2.31)$$

e scambiando l'ordine d'integrazione diventa

$$\mathcal{G}(\phi, \mathbf{c}, b) := \int_{\Omega} \sum_{i=1}^N \left( \int_{\Omega_i} K(y-x) |I(x) - b(y)c_i|^2 dy \right) \chi_i(\phi(x)) dx \quad (2.32)$$

dove si è indicato in modo sintetico il vettore  $\mathbf{c} = (c_1, \dots, c_N)$ .

In ultima analisi, possiamo scrivere il funzionale nella seguente forma

$$\mathcal{G}(\phi, \mathbf{c}, b) = \int \sum_{i=1}^N e_i(x) \chi_i(\phi(x)) dx \quad (2.33)$$

con  $e_i(x) = \int K(y-x) |I(x) - b(y)c_i|^2 dy$ . Svolgendo i calcoli il termine  $e_i$  diventa

$$e_i = I^2 \int K(y-x) dy - 2c_i I(b * K) + c_i^2 (b^2 * K).$$

Ricordando quanto detto nella sezione precedente, il funzionale definito in (2.28) viene inserito nel metodo DRLSE. Si formula l'energia esterna

$$\mathcal{E}_{ext} = \mathcal{G}(\phi, \mathbf{c}, b) + \nu \mathcal{L}_1(\phi) \quad (2.34)$$

con  $\mathcal{L}_1$  definito come in (2.18) nel caso particolare in cui  $g = 1$ .

Tale energia esterna ci permette di riscrivere la formulazione del DRLSE espressa in (2.1) come segue

$$\mathcal{E}(\phi, (c), b) = \mu \mathcal{R}_p(\phi) + \mathcal{E}_{ext}(\phi, \mathbf{c}, b) \quad (2.35)$$

$$= \mu \mathcal{R}_p(\phi) + \mathcal{G}(\phi, \mathbf{c}, b) + \nu \mathcal{L}_1(\phi) \quad (2.36)$$

con  $\mathcal{R}_p$  definito come in (2.2). Poiché in questa nuova formulazione non si hanno problemi legati alle variazioni troppo repentine dell'intensità del gradiente di  $\phi$  in quanto questo metodo è basato sull'assunzione che le regioni dell'immagine possano essere approssimate con una funzione costante a tratti, si può scegliere il termine regolarizzazione definito in (2.4), che permette di regolarizzare la LSF e approssimarla a una funzione distanza con segno durante tutta l'evoluzione.

Le variabili da minimizzare in (2.35) sono la LSF  $\phi$ , le costanti che compongono il vettore  $\mathbf{c}$  e il BF  $b$ , dunque verranno minimizzate in tre passi diversi. Per minimizzare  $\mathcal{E}(\phi, \mathbf{c}, b)$  rispetto a  $\phi$  si considerano fissati  $\mathbf{c}$  e  $b$ . Si minimizza il funzionale usando il metodo definito in (2.5), ottenendo la seguente formulazione dell'equazione dell'evoluzione della LSF

$$\frac{\partial \phi}{\partial t} = -\delta(\phi)(e_1 - e_2) + \nu \delta(\phi) \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) + \mu \operatorname{div}(d_p(|\nabla \phi|) \nabla \phi) \quad (2.37)$$

dove  $d_p(s)$  è la funzione definita in (2.7) nel caso in cui il potenziale  $p = \frac{1}{2}(s-1)^2$ . Fissati invece i valori di  $\phi$  e di  $b$ , il valore ottimale di  $\mathbf{c}$  che minimizza  $\mathcal{E}(\phi, \mathbf{c}, b)$ , denotato con  $\hat{\mathbf{c}} = (\hat{c}_1 \dots \hat{c}_N)$ , è dato da

$$\hat{c}_i = \frac{\int (b * K) I \chi_i dy}{\int (b^2 * K) \chi_i dy}. \quad (2.38)$$

Per quanto riguarda il BF, si calcola il valore ottimale di  $b$ , e lo si denota con  $\hat{b}$ , come segue

$$\hat{b} = \frac{(IJ^{(1)}) * K}{J^{(2)} * K} \quad (2.39)$$

con  $J^{(1)} = \sum_{i=1}^N c_i \chi_i$  e  $J^{(2)} = \sum_{i=1}^N c_i^2 \chi_i$ . Nell'implementazione di questo metodo gli aggiornamenti della  $\phi$ , delle costanti  $c_i$  e del BF  $b$  vengono effettuati ad ogni passo secondo le equazioni definite in (2.37), (2.38) e (2.39).

### 2.2.3 Implementazione numerica

Il metodo di implementazione di entrambi i metodi esposti risulta semplice e diretto. Per implementare l'EBM presentato nella sezione (2.2.1) non si ha più bisogno di utilizzare i metodi *upwind* a causa della presenza del termine diffusivo. Si usa dunque uno schema alle differenze finite per approssimare le derivate spaziali, mentre la derivata temporale è approssimata con differenze all'avanti. Usando la convenzione secondo cui l'apice indica l'indice della discretizzazione temporale, il pedice l'indice spaziale e  $\Delta t$  il passo temporale per la discretizzazione, l'equazione (2.20) può essere scritta come

$$\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^k}{\Delta t} = L(\phi_{i,j}^k)$$

dove  $L$  è l'approssimazione della parte destra di (2.20). L'equazione alle differenze allora diventa

$$\phi_{i,j}^{k+1} = \phi_{i,j}^k + \Delta t L(\phi_{i,j}^k). \quad (2.40)$$

La condizione da rispettare affinché il metodo sia stabile è la nota condizione di *Courant-Friedrichs-Lewy* (CFL)  $\mu \Delta t < (1/4)$ . Rispettando questa condizione si può scegliere un passo temporale relativamente grande per velocizzare l'evoluzione della curva, a cui corrisponderà un valore di  $\mu$  piccolo proprio per la condizione CFL. Nel secondo metodo proposto, quello nella sezione (2.2.2), si usa lo stesso schema alle differenze finite appena mostrato. La delta di Dirac  $\delta$  e la funzione di Heaviside  $H$  sono approssimate come in [18] da

$$H_\epsilon(x) = \frac{1}{2} \left[ 1 + \frac{2}{\pi} \arctan \left( \frac{x}{\epsilon} \right) \right] \quad (2.41)$$

$$\delta_\epsilon(x) = H'_\epsilon(x) = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + x^2}. \quad (2.42)$$

# Capitolo 3

## Prove sperimentali

Nel seguente capitolo si mostrano e si commentano i risultati ottenuti applicando il metodo proposto in [16], ossia il metodo RBM con DRLSE esposto nel capitolo precedente, a diverse immagini di nevi. Le prove fatte sono state possibili grazie alla collaborazione con il Dott. Ignazio Stanganelli dell'IRST di Forlì e Ravenna e con il Dipartimento di Dermatologia dell'Ospedale di Ravenna, da cui abbiamo potuto ricevere le immagini su cui applicare il metodo esaminato in questa tesi. Le prove sono state compiute con un PC Lenovo Z500, processore Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz, Gpu GeForce GT 635M, 8 GB di RAM, hard disk Samsung MZ-7TE120BW SSD 840 EVO 120 GB e sistema operativo Windows 8.1. Le prove, inoltre, sono state svolte con *Matlab R2013a*.

### 3.1 Obiettivi

L'algoritmo preso in considerazione in questa tesi andrà a far parte di un software che verrà usato in ambito dermatologico, a sussidio dei medici, per rivelare possibili lesioni melanocitiche in un paziente: il software ha il compito di generare un indicatore di rischio di tumore e, allo stesso tempo, di dare uno spettro di lesioni melanocitiche somiglianti, rispetto a determinate caratteristiche, al nevo preso in considerazione. Lo scopo del nostro lavoro, dunque, è consegnare a un secondo algoritmo l'immagine di

un nevo binarizzata, in cui si riconosca in modo chiaro la forma del nevo considerato. L'analisi sull'efficacia del metodo è solo visiva e relativa al costo computazionale. Infatti, segmentando immagini reali non si conoscono a priori i bordi da individuare, per cui, ad oggi, non si è in grado di dare una stima quantitativa dell'errore compiuto. A tale scopo dunque, resta necessario il giudizio di un esperto che convalidi i risultati ottenuti. Si richiede inoltre che l'algoritmo sia capace di:

- segmentare il nevo in maniera non troppo puntuale, poiché una precisione troppo elevata implicherebbe un'eccessiva irregolarità dei bordi e la formazione di molte componenti connesse separate, fatto che va contro le esigenze del processo in cui andrà a svolgere il proprio compito questo algoritmo. Dunque si ha bisogno di una segmentazione precisa, che veda gli aloni eventualmente presenti, ma che non sia troppo fine;
- segmentare il nevo in un tempo dell'ordine della decina di secondi, in quanto il software completo in cui si inserisce questo programma ha bisogno di tempi di utilizzo relativamente rapidi;
- essere il meno influenzato possibile dall'eventuale presenza di peli o di oggetti di disturbo che altererebbero i risultati.



## 3.2 Descrizione delle prove

### Taglio delle immagini

Tutte le immagini dei nevi analizzate, hanno dimensione  $768 \times 586$  pixels. Nella maggioranza dei casi, le immagini contengono una buona parte di informazioni inutili per i nostri scopi: risulta quindi dispendioso applicare l'algoritmo a tutta l'immagine. Visto che la parte più dispendiosa dal punto di vista numerico è il calcolo delle derivate parziali e delle convoluzioni, permettere di calcolarle su una matrice di dimensione minore equivale ad accelerare i tempi e di guadagnare efficienza nella segmentazione. Spesso, la presenza di peli o gocce d'acqua può rendere più difficoltosa la ricerca del bordo del nevo, dunque toglierli o ridurli considerevolmente attraverso il taglio dell'immagine migliora notevolmente il risultato.

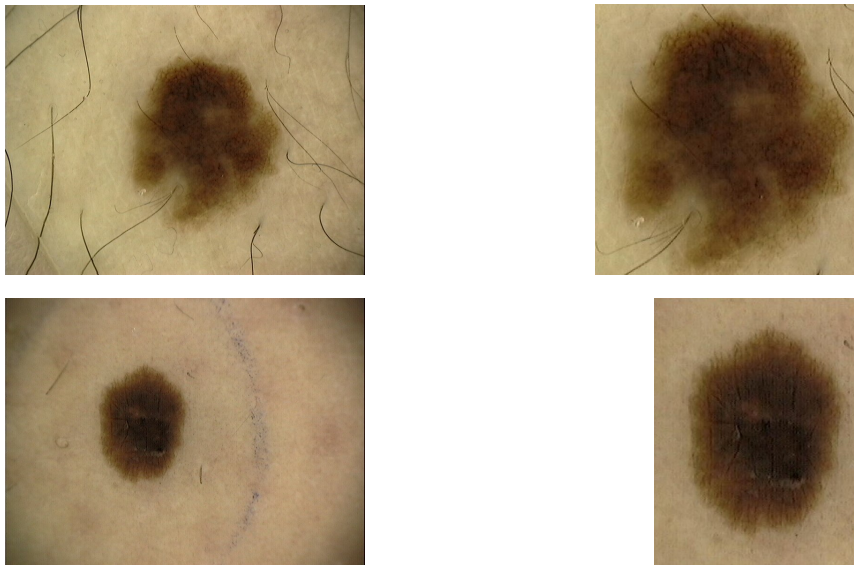


Figura 3.1: Immagine intera a sinistra e tagliata a destra

### Funzione iniziale

Come visto precedentemente, il metodo necessita di una funzione di partenza da cui cominciare l'evoluzione. Come studiato in [16], l'algoritmo non è influenzato dalla funzione iniziale scelta: esso infatti non è dipendente nè dalla funzione iniziale data, nè dalla sua posizione all'interno dell'immagine. In questo c'è una grande differenza rispetto agli altri metodi *level set* EBM, dove è necessario considerare una funzione di partenza che sia una funzione distanza posta all'interno o all'esterno delle zone di interesse, e in cui risulta indispensabile indicare tramite il settaggio di un parametro il verso dell'evoluzione. Nel capitolo precedente si è visto che in [15], non è necessario che la funzione di partenza fosse una funzione distanza con segno, ma l'esito dell'evoluzione del fronte resta comunque influenzato dalla posizione della funzione iniziale. Nel metodo preso da noi in considerazione, invece, si può scegliere una funzione qualsiasi posizionata in un luogo arbitrario dell'immagine. Si è scelto per semplicità di assegnare inizialmente una funzione gradino che sia pari a una costante  $c_0 = 2$  all'esterno di un rettangolo scelto e zero all'interno, posizionata a una distanza di 60 pixel da bordo dell'immagine già ritagliata, come in figura.

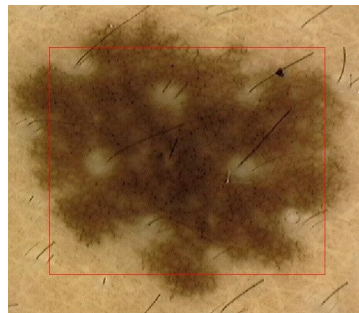
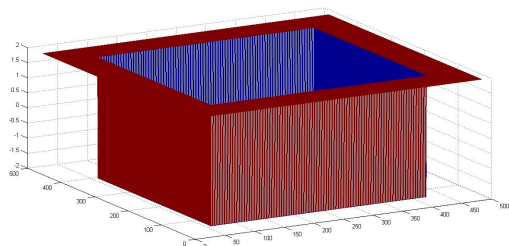


Figura 3.2: A sinistra la funzione iniziale, a destra in rosso il livello zero della funzione gradino

**Numero iterazioni**

L'algoritmo preso in considerazione si tratta di un metodo iterativo e, poiché non è possibile reperire un criterio d'arresto su immagini reali, è necessario porre un limite alle iterazioni da compiere. Per stabilirne uno tale per ottenere dei buoni risultati, sono state fatte alcune prove su nevi campione con un numero di iterazioni molto alto (96 iterazioni esterne) e sono state stampate le immagini ottenute durante il processo ogni 8 iterazioni. In tale modo si è visto, come si può osservare nelle figure successive, che il numero di iterazioni necessarie per raggiungere una buona precisione nella segmentazione è in media intorno alla decina. Dopo aver fatto queste prove iniziali, si è deciso di limitare le iterazioni esterne, cioè il numero di approssimazioni, a 9.

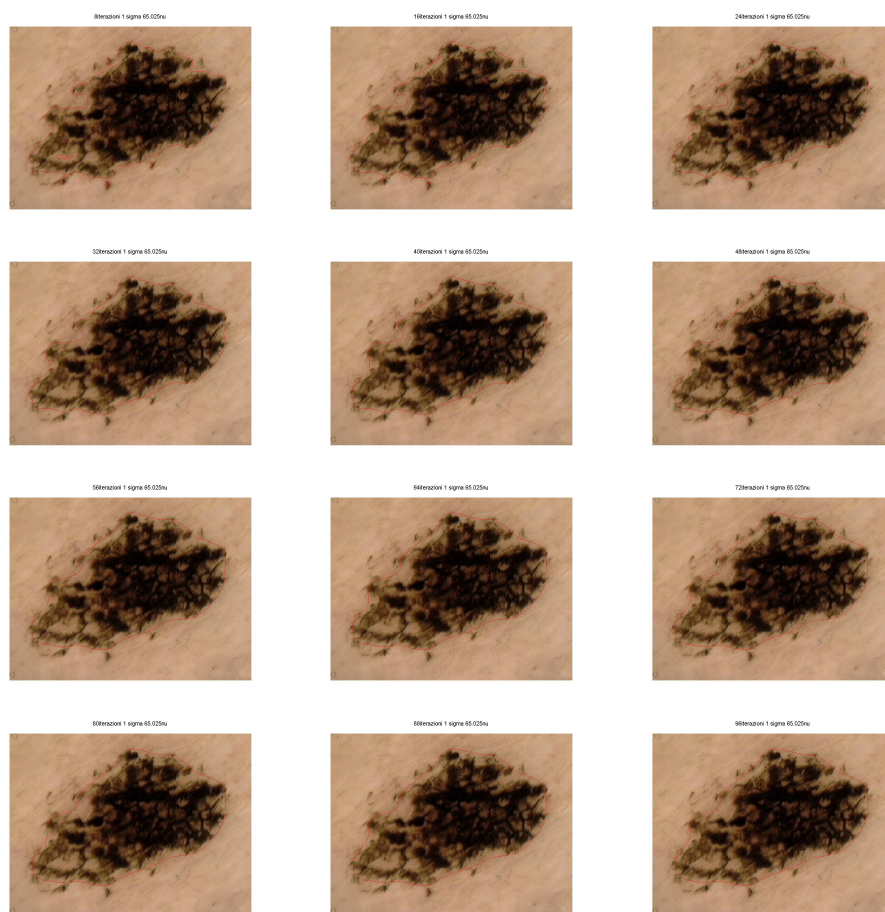


Figura 3.3: Prove compiute su un nevo con 96 iterazioni, i cui risultati sono stati salvati ogni 8 iterazioni: è evidente come già al passo numero 8 sia ben definito il bordo del nevo.

### 3.3 Scelta dei parametri

Come visto nei capitoli precedenti, in particolare facendo riferimento alle equazioni (2.37)(2.38)(2.39), i parametri da modificare per ottenere dei cambiamenti significativi sulla segmentazione e sulle performance dell'algoritmo sono:

- il numero delle iterazioni interne: le iterazioni che vengono compiute ad ogni passo

per evolvere la sola  $\phi$  (2.37);

- il parametro  $\nu$ , che pesa l'effetto di minimizzazione della lunghezza della curva, ossia il livello zero della LSF (2.37).
- il parametro  $\mu$ , che pesa l'effetto regolarizzante del funzionale che approssima la LSF alla funzione distanza con segno (2.37). Questo parametro deve rispettare la condizione CFL,  $\mu\Delta t < (1/4)$ , per assicurare stabilità ed efficienza del metodo. Dunque settando uno dei due parametri  $\mu$  o  $\Delta t$  si definisce l'altro.
- la funzione  $K$ , che deve rispettare le condizioni imposte nelle (2.29) e (2.30). Come precedentemente detto si è presa, come in [16], una funzione gaussiana troncata, in tale modo la si può definire tramite la sola scelta della varianza  $\sigma$ . Per vedere l'azione di questa funzione sul metodo si faccia riferimento alla (2.38) e alla (2.39).  $K$  numericamente è una maschera di dimensione  $\omega \times \omega$ , con  $\omega$  il più piccolo numero dispari tale che  $\omega \geq 4 * \sigma + 1$ . Dunque  $K$  ha l'effetto di restringere l'azione del processo solo sulla maschera: più la varianza è piccola, più piccola è la maschera. Inoltre più la dimensione della maschera è piccola più il tempo necessario per l'aggiornamento del valore di  $c$  e  $b$  sarà ridotto poiché si dovranno compiere convoluzioni meno onerose.

Sono state fatte delle prove per cercare di identificare l'ordine di grandezza dei parametri necessari per avere dei cambiamenti sensibili sui tempi e sulla segmentazione. Con i parametri individuati sono state poi compiute prove, per verificarne l'efficacia su un campione di nevi più vasto. Per verificare la combinazione migliore tra i parametri sono stati fatti dei test sulle immagini in possesso, facendoli variare nella seguente maniera:

- il numero delle iterazioni interne pari a 1, 4, 8, 16;
- $\nu = 10^{-i} \times 255^2$  con  $i = 1, \dots, 4$ ;
- $\Delta t = 10^z$  con  $z = -1, \dots, 2$ . Definendo  $\Delta t$  si definisce di conseguenza, per rispettare la condizione CFL, anche il parametro  $\mu = 0.2/\Delta t$  ;

- $\sigma = 1, 2, 5, 10, 20, 25, 50, 100, 200$ .

Inoltre, per verificare l'efficienza del metodo si è calcolato il tempo necessario per compiere le iterazioni facendo uso della funzione *tic-toc* di matlab e si è fatto stampare il *profiler* dei tempi per individuare quali sono le funzioni che impiegano più tempo.

Nelle considerazioni successive si prenderà come nevo di prova il seguente:



Figura 3.4: Nevo preso in considerazione per analizzare le *performance* dell'algoritmo

### 3.3.1 Variazione di $\nu$

Come è evidente dalle immagini, al variare del parametro  $\nu$  si ottiene una segmentazione molto diversa. Infatti, esso pesa l'effetto del termine del funzionale che minimizza la lunghezza dell'insieme di livello zero della LSF: più  $\nu$  è grande più è importante l'effetto della minimizzazione della lunghezza della curva, che risulterà quindi più regolare. Si considera il nevo prov in figura (??) e si osserva come si modifica la segmentazione al variare di  $\nu$ , per i valori fissati  $\sigma = 10$  e  $\Delta t = 1$  con 8 iterazione interna.

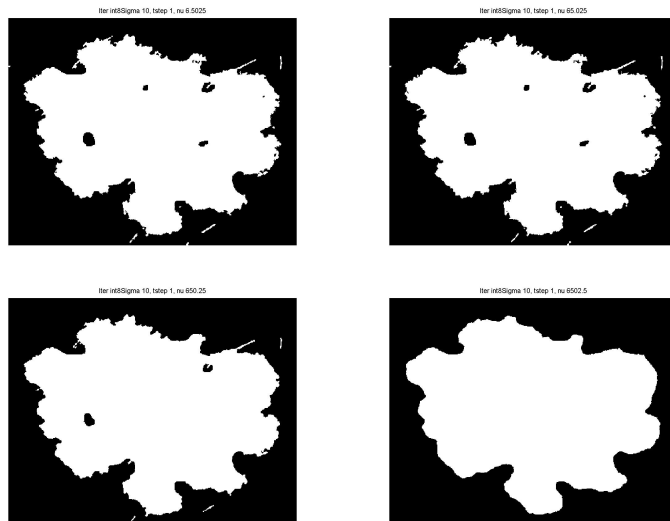
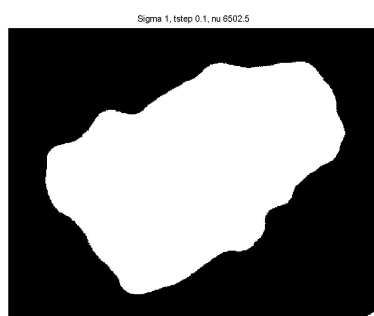


Figura 3.5: Immagini del nevo segmentato con valori fissati  $\sigma = 10$ ,  $\Delta t = 1$  e 8 iterazione interna, al variare di  $\nu = 6,5025$ ,  $65,025$ ,  $650,02$ ,  $6502,5$ , in ordine scorrendo le immagine orizzontalmente

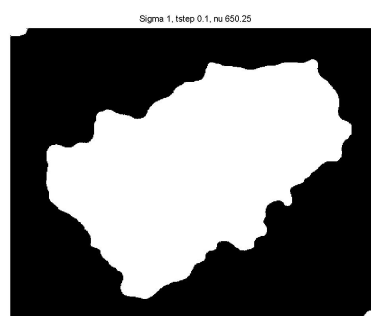
Per  $\nu = 6502,5$  si ha un segmentazione più regolare e anche meno influenzata dai piccoli oggetti, come i peli, e da eventuali piccole variazioni di intensità di colore all'interno del nevo (i buchi che appaiono all'interno del nevo). Tuttavia un valore troppo elevato di  $\nu$  può dare delle approssimazioni troppo grossolane, soprattutto in corrispondenza di particolari combinazioni di parametri: in alcuni nevi, come si può riscontrare nelle immagini che seguono, si perdono delle informazioni importanti.



Figura 3.6: Immagine iniziale di un nevo



(a)  $\nu=6502,5$



(b)  $\nu=650,25$

La segmentazione in figura (3.6(a)) è meno accurata di quella in (3.6(b)): nella prima si perdono informazioni significative sul nevo. Dunque, pur essendo importante per i nostri scopi un'alta regolarità della curva che descrive il bordo del nevo, in questo caso



essa non soddisfa la proprietà di essere aderente al contorno del nevo, come invece è più evidente nella seconda immagine corrispondente al valore di  $\nu = 650, 25$ . Perciò il valore più adatto di  $\nu$  per la segmentazione dei nevi sembra essere 650, 25, per il quale si hanno generalmente risultati ottimi anche in correlazione al cambiamento che subisce la segmentazione variando gli altri parametri. Per quanto riguarda il tempo compiuto non si notano variazioni apprezzabili al variare del parametro, quindi una variazione di  $\nu$  non implica variazioni di tempi consistenti.

### 3.3.2 Variazione di $\Delta t$

Il parametro  $\Delta t$  è il passo temporale dell'implementazione numerica del metodo (2.40). Abbiamo visto come questo sia legato al parametro  $\mu$ , peso relativo al funzionale che regolarizza la LSF. A un passo temporale piccolo corrisponderà un valore di  $\mu$  maggiore e viceversa. Come si nota dalle immagini, variando  $\Delta t = 0.1, 1, 10, 100$ , a valori di  $\sigma$ ,  $\nu$  e numero di iterazioni interne fissati, si ha una segmentazione sempre meno precisa col crescere di  $\Delta t$  per due motivi:

- se  $\Delta t$  cresce, per la condizione CFL,  $\mu$  decresce, e decresce di conseguenza il peso che ha il termine del funzionale responsabile della regolarizzazione della LSF (2.37); si otterrà dunque una LSF meno regolare e quindi una segmentazione meno precisa;
- in secondo luogo, l'approssimazione dello schema alle differenze finite (2.40) con un passo temporale abbastanza grande perde necessariamente delle informazioni, poiché compie dei salti più consistenti durante l'evoluzione, ottenendo un modello meno accurato rispetto a quello ottenuto con valori di  $\Delta t$  minori.

I tempi di esecuzione non variano sensibilmente al variare di  $\Delta t$ , infatti, se si guarda la tabella a pag. 70 c'è una discrepanza dell'ordine dei secondi tra due diversi  $\Delta t$ , quindi, per i nostri scopi, la scelta di  $\Delta t$  è indifferente per quanto riguarda i tempi.

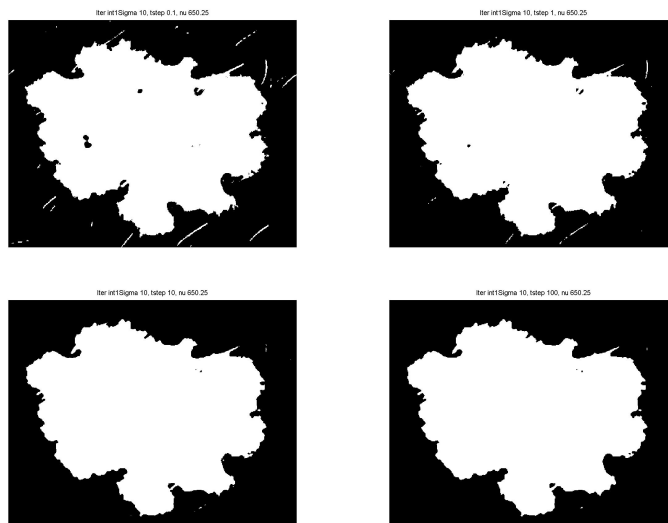


Figura 3.7: Segmentazione di un nevo al variare di  $\Delta t = 0, 1, 1, 10, 100$ , con numero di iterazioni interne pari a 1  $nu = 650, 25$  e  $\sigma = 10$

Per una segmentazione che raggiunga i nostri scopi, il valore di  $\Delta t$  migliore è un numero compreso tra 1 e 10 con il valore di  $\mu$  è definito di conseguenza.

### 3.3.3 Variazione delle iterazioni interne

Un parametro che modifica l'esito della segmentazione è, come abbiamo detto, il numero delle iterazioni necessarie per far evolvere esclusivamente la LSF attraverso l'equazione (2.37). Ciò che ci aspettiamo è che, all'aumentare delle iterazioni aumenti la regolarità del bordo segmentato, ma contemporaneamente aumenti anche il tempo totale del metodo. Questi due effetti sono evidenti dalle seguenti immagini.

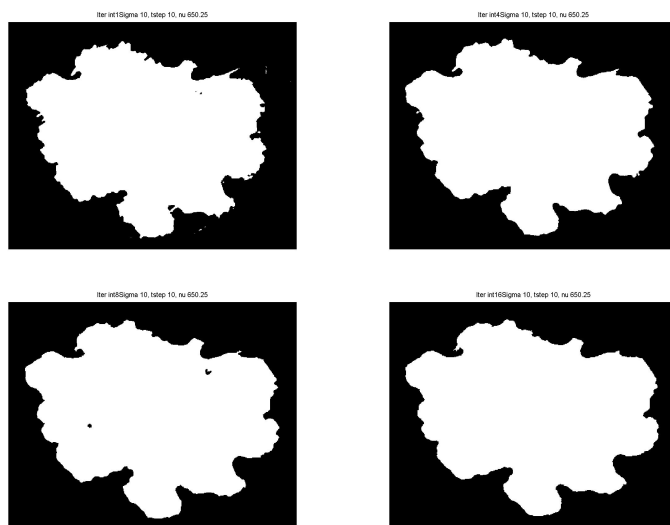


Figura 3.8: Segmentazione di un neve al variare delle iterazioni interne 1, 4, 8, 16 e con restanti parametri fissati  $\Delta t = 10$   $\nu = 650,25$  e  $\sigma = 10$

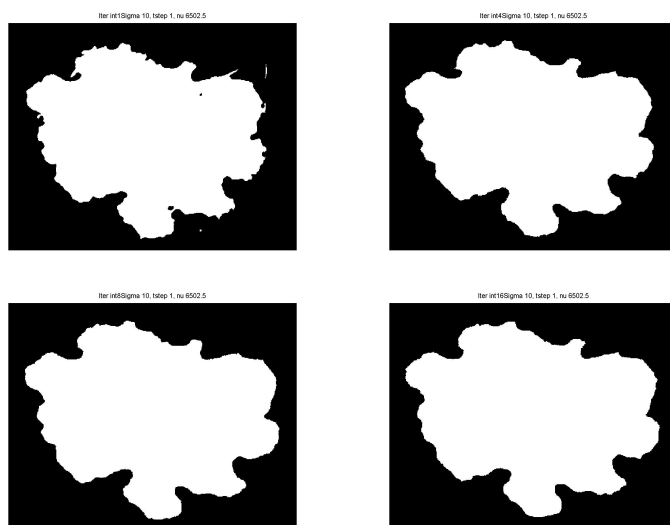


Figura 3.9: Segmentazione di un neve al variare delle iterazioni interne 1, 4, 8, 16 e con restanti parametri fissati  $\Delta t = 1$   $\nu = 6502,5$  e  $\sigma = 10$

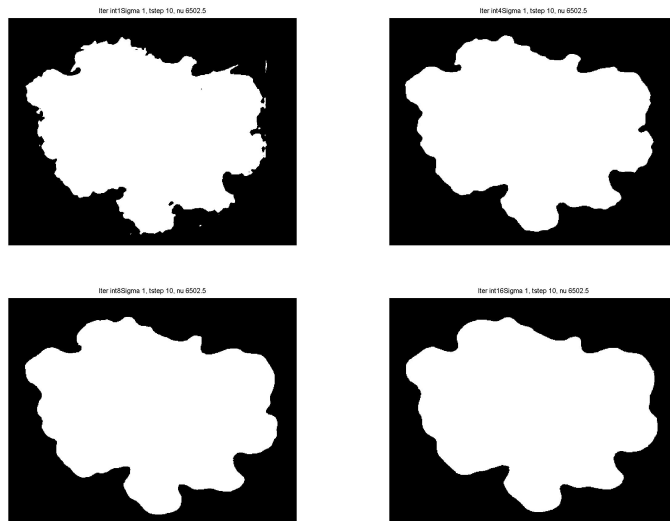


Figura 3.10: Segmentazione di un nevo al variare delle iterazioni interne 1, 4, 8, 16 e con restanti parametri fissati  $\Delta t = 1$ ,  $\nu = 6502,5$  e  $\sigma = 1$

Si nota, che, come previsto, la regolarità della curva aumenta con l'aumentare del numero di iterazioni interne. Questo, come si osserva nelle figure, fa sì che il metodo sia meno sensibile a oggetti di disturbo come peli. Il risultato varia ovviamente al variare degli altri parametri, ma la segmentazione più efficace appare sempre quella relativa a 8 iterazioni interne.

Per quanto riguarda la variazione dei tempi, si è disposto un istogramma che mostra il tempo necessario a compiere la segmentazione (in secondi), al variare del parametro  $\sigma$ : quest'ultimo cambia considerevolmente il costo computazionale perché è il parametro responsabile della dimensione della maschera di convoluzione. La convoluzione tra due matrici ha un costo e un tempo di esecuzione numerica più elevato con l'aumentare della dimensione delle matrici. Questo effetto emerge dal grafico sottostante, dove si può osservare che i tempi crescono all'aumentare delle iterazioni interne, ma ancora di più crescono all'aumentare di  $\sigma$ .

## Istogramma dei tempi

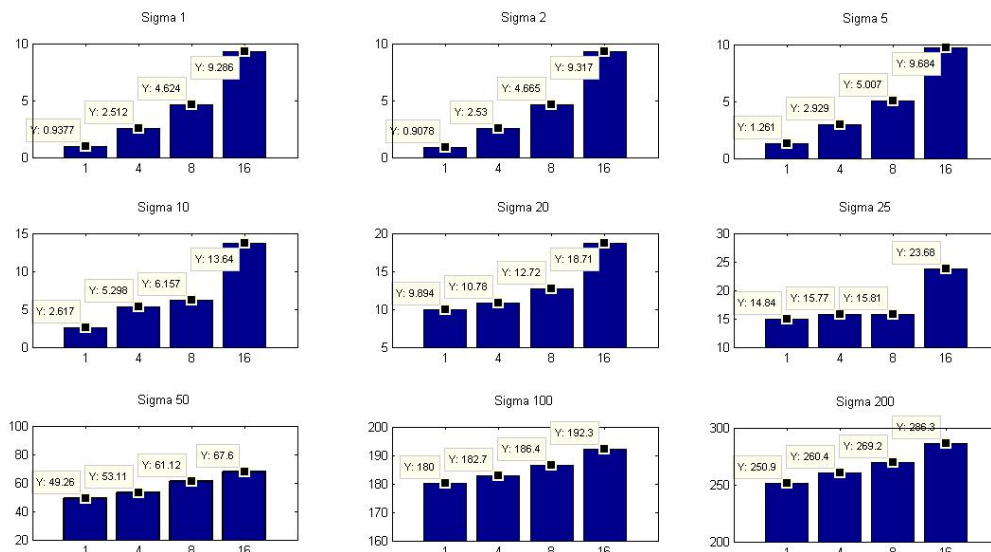


Figura 3.11: Istogramma relativo all'efficienza del metodo, in secondi (ordinate), rispetto alla variazione delle iterazioni interne (ascisse) al variare del parametro  $\sigma$

Si può vedere che fino a  $\sigma = 10$ , i tempi sono molto bassi, sotto i 15 secondi, in particolare si hanno valori sotto ai 6 secondi se si considera un numero di iterazioni interne minori o uguali a 8. Ovviamente col crescere della dimensione della maschera, anche i tempi relativi ad ogni valore del numero delle iterazioni interne crescono. I primi tempi di esecuzione non accettabili sono quelli superiori al minuto, relativi a valori di  $\sigma$  superiori a 100, che dunque perdono di significato per i nostri scopi.

Dunque mediando i risultati ottenuti per la segmentazioni e i tempi di esecuzione, il valore ottimale delle iterazioni interne è 8: la segmentazione è regolare e non perde dettagli importanti e i tempi ottenuti sono dell'ordine della decina di secondi. E' importante dire che i tempi ottenuti sono molto ragionevoli, soprattutto considerando che si sta lavorando con un metodo *level set*, il quale è solitamente molto oneroso dal punto di vista computazionale. La velocità di esecuzione di questo metodo è sicuramente dovuta alla formulazione del DRLSE, precedentemente spiegata nel capitolo 2, che permette di usare

metodi alle differenze finite per calcolare l'evoluzione della LSF e non rende necessaria la reinizializzazione della LSF. Si nota inoltre, che le *performance* dell'algoritmo possono essere migliorate ulteriormente applicando una formulazione *narrow band* all'algoritmo, come espresso in [16] e in [19]: si ridurrebbe di molto il numero delle differenze finite da compiere riducendo sensibilmente il tempo necessario per la segmentazione.

### 3.3.4 Variazione di $\sigma$

La funzione  $K$  è una maschera di dimensione  $\omega \times \omega$  che, come detto precedentemente, si è scelta come una gaussiana troncata. Il suo valore e la sua dimensione sono definiti in modo univoco dalla scelta della varianza  $\sigma$ , poiché la dimensione di  $K$  è definita come il più piccolo numero dispari tale che  $\omega \geq 4\sigma + 1$ . Come l'usuale comportamento di una maschera, al crescere delle dimensioni, aumentano le informazioni ottenute e perciò, l'effetto, nel nostro caso, è quello di avere una segmentazione più dettagliata. Allo stesso tempo, la maschera non può avere dimensioni troppo grandi altrimenti si ha l'effetto contrario a quello appena indicato. Dalle immagini riportate in seguito, che si focalizzano sulla variazione di  $\sigma$ , si nota in modo evidente le conseguenze dell'aumentare delle dimensioni di  $K$ : fino a  $\sigma = 50$  (in altri nevi fino a  $\sigma = 100$ ) si ottiene una segmentazione più precisa, mentre da 100 in poi si ottengono segmentazioni meno definite. Per quanto riguarda l'efficienza del metodo, come precedentemente notato, si vede che il tempo necessario a compiere le iterazioni aumenta al crescere della dimensione. Questo fatto è causa del costo computazionale delle convoluzioni tra l'immagine data e il nucleo  $K$ , che risultano più onerose se la dimensione di  $K$  è maggiore. Dunque, i valori più consoni, compromesso tra efficienza del metodo e esattezza della segmentazione, sono quelli compresi tra 1 e 10.

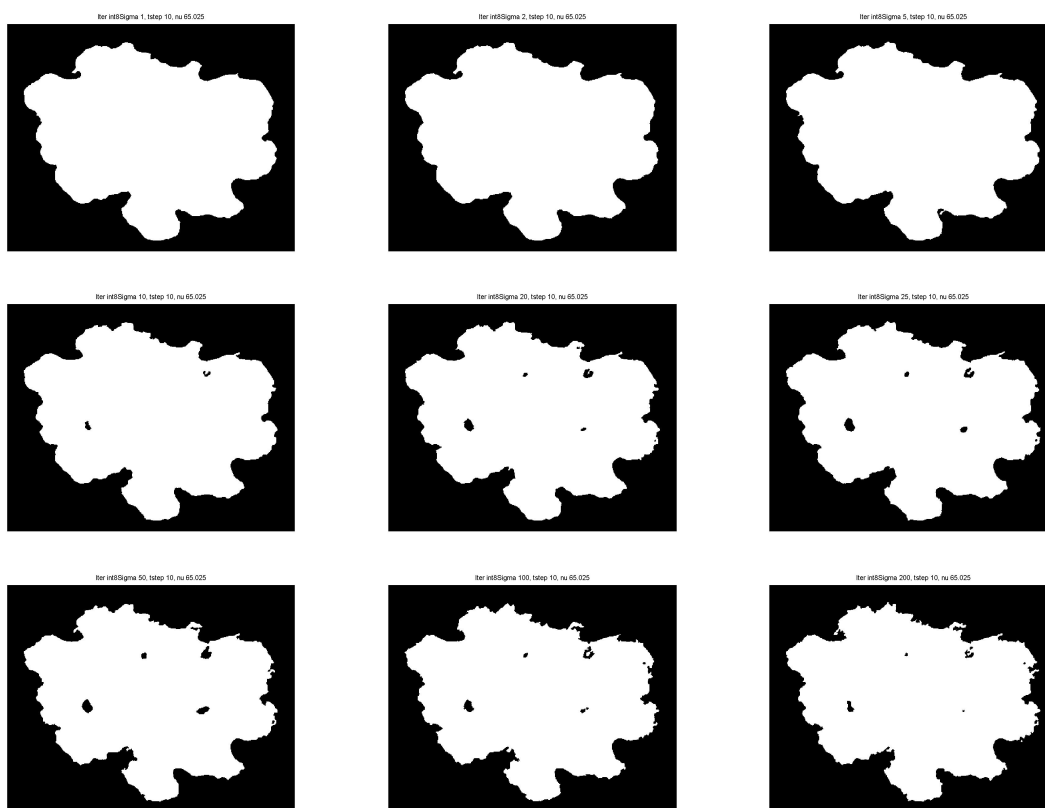


Figura 3.12: Segmentazione di un nevo al variare di  $\sigma = 1, 2, 5, 10, 20, 25, 50, 100, 200$ , 8 iterazioni interne,  $\nu = 65.025$ ,  $\Delta t = 10$

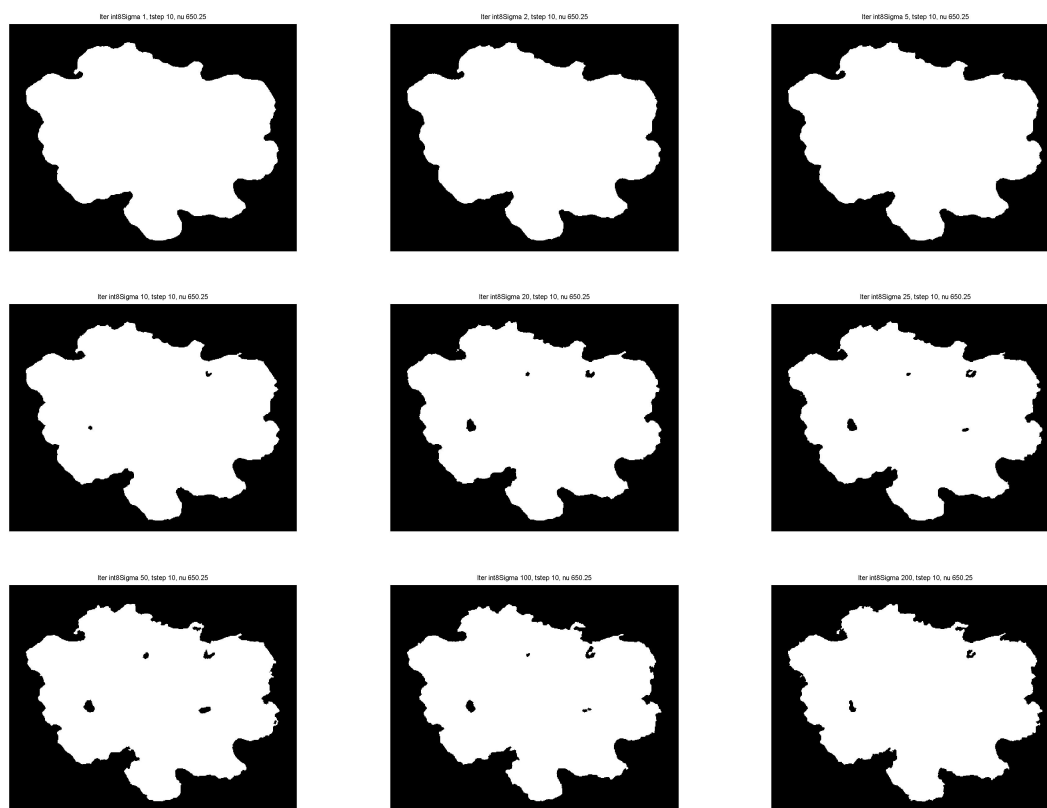


Figura 3.13: Segmentazione di un neve al variare di  $\sigma = 1, 2, 5, 10, 20, 25, 50, 100, 200$ , con 8 iterazioni interne,  $\nu = 650.25$ ,  $\Delta t = 10$



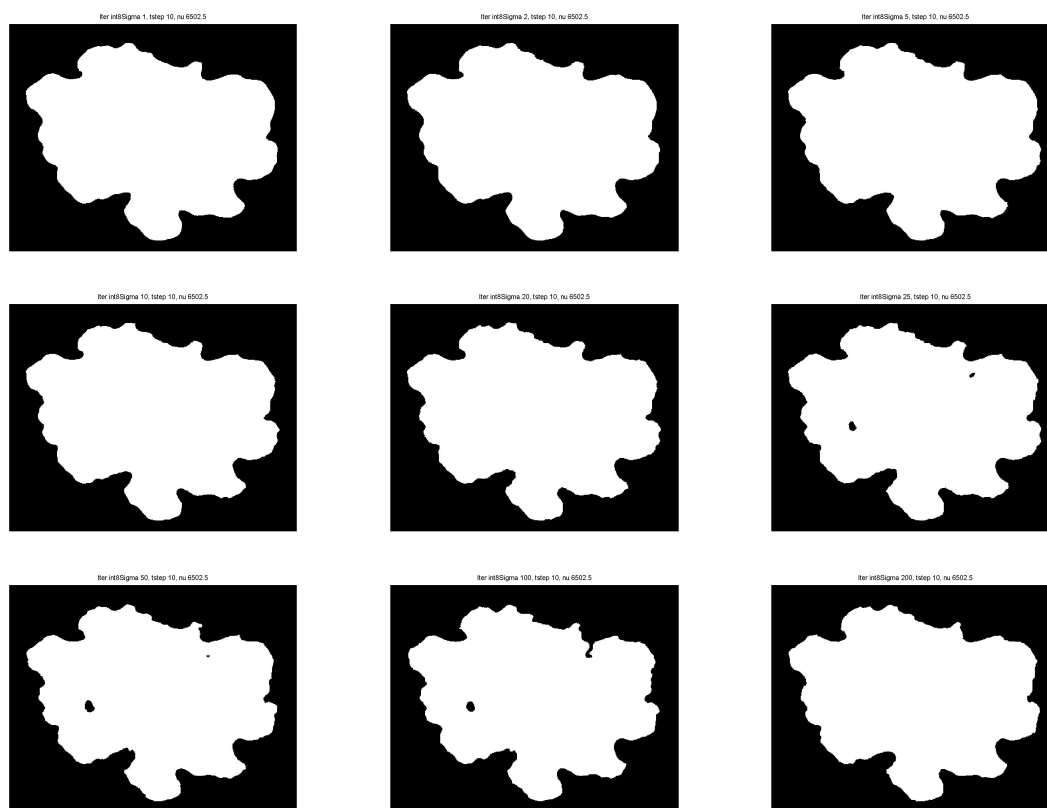


Figura 3.14: Segmentazione di un neve al variare di  $\sigma = 1, 2, 5, 10, 20, 25, 50, 100, 200$ , con 8 iterazioni interne,  $\nu = 6502.5$ ,  $\Delta t = 10$

### 3.3.5 Risultati

Per analizzare le prove fatte si mostrano ora l'immagine di partenza, l'immagine binarizzata e la tabella dei tempi relativi. Con la binarizzazione dell'immagine si evidenzia in modo più chiaro quali sono gli effetti dell'uso dell'algoritmo ed è possibile confrontare il risultato ottenuto con altri metodi.

#### **Immagini segmentate**

Si prende in considerazione in particolare il nevo presentato in figura (3.4) e si mostrano gli esiti della segmentazione, avendo applicato, come detto precedentemente, un taglio all'immagine (la si riduce a una matrice di dimensioni  $430 \times 500$ ) e avendo impostato come LSF iniziale una funzione a gradino.

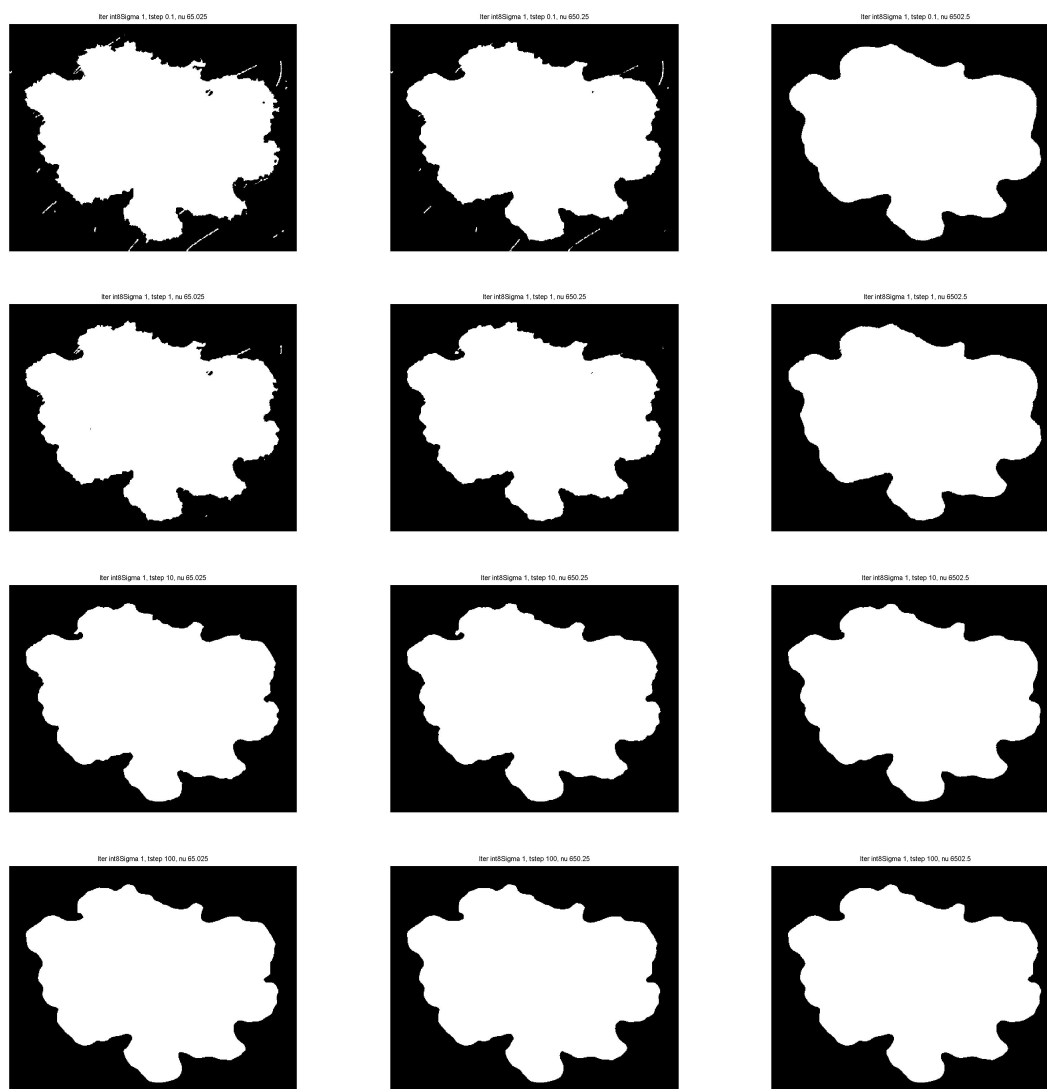


Figura 3.15: Segmentazioni relative al valore  $\sigma = 1$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502, 5$ .

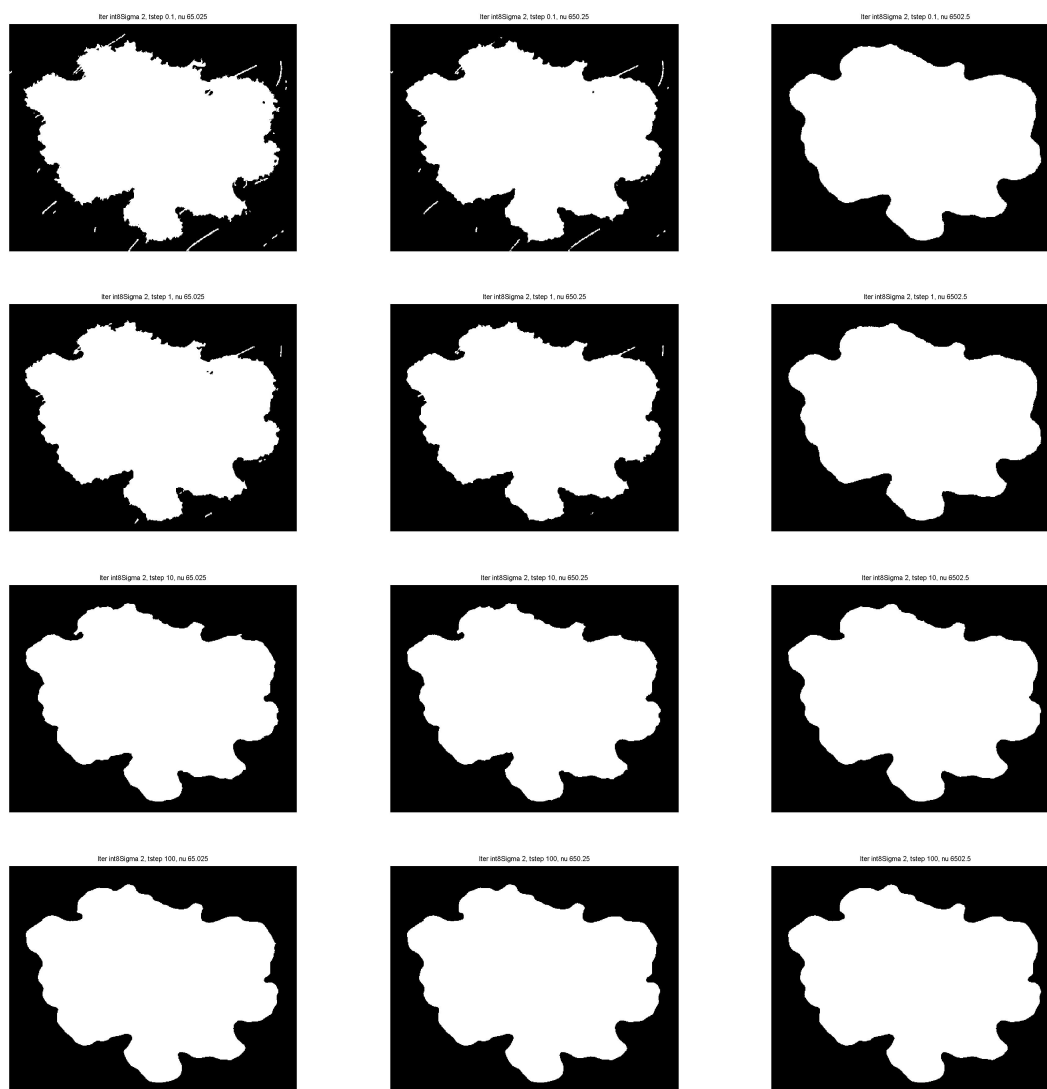


Figura 3.16: Segmentazioni relative al valore  $\sigma = 2$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

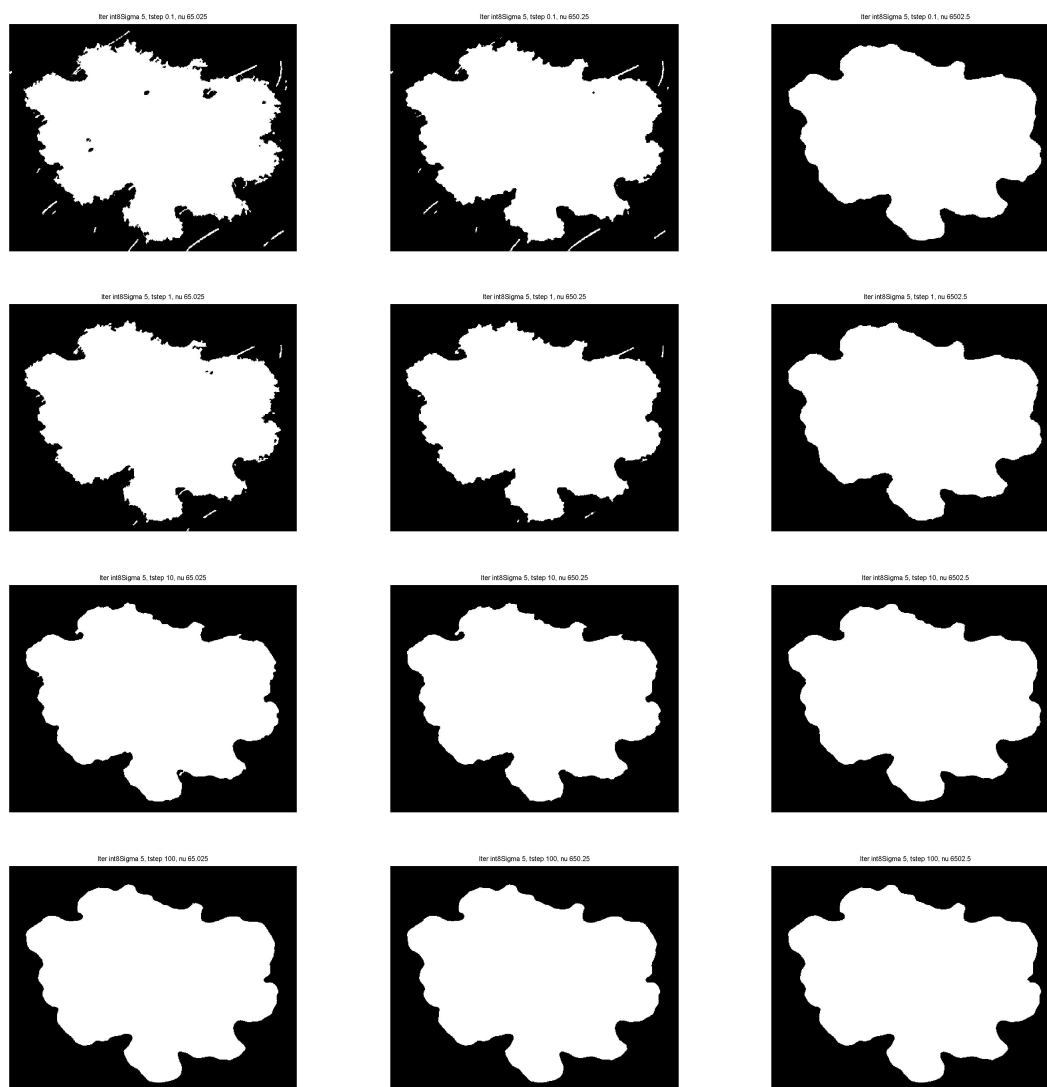


Figura 3.17: Segmentazioni relative al valore  $\sigma = 5$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

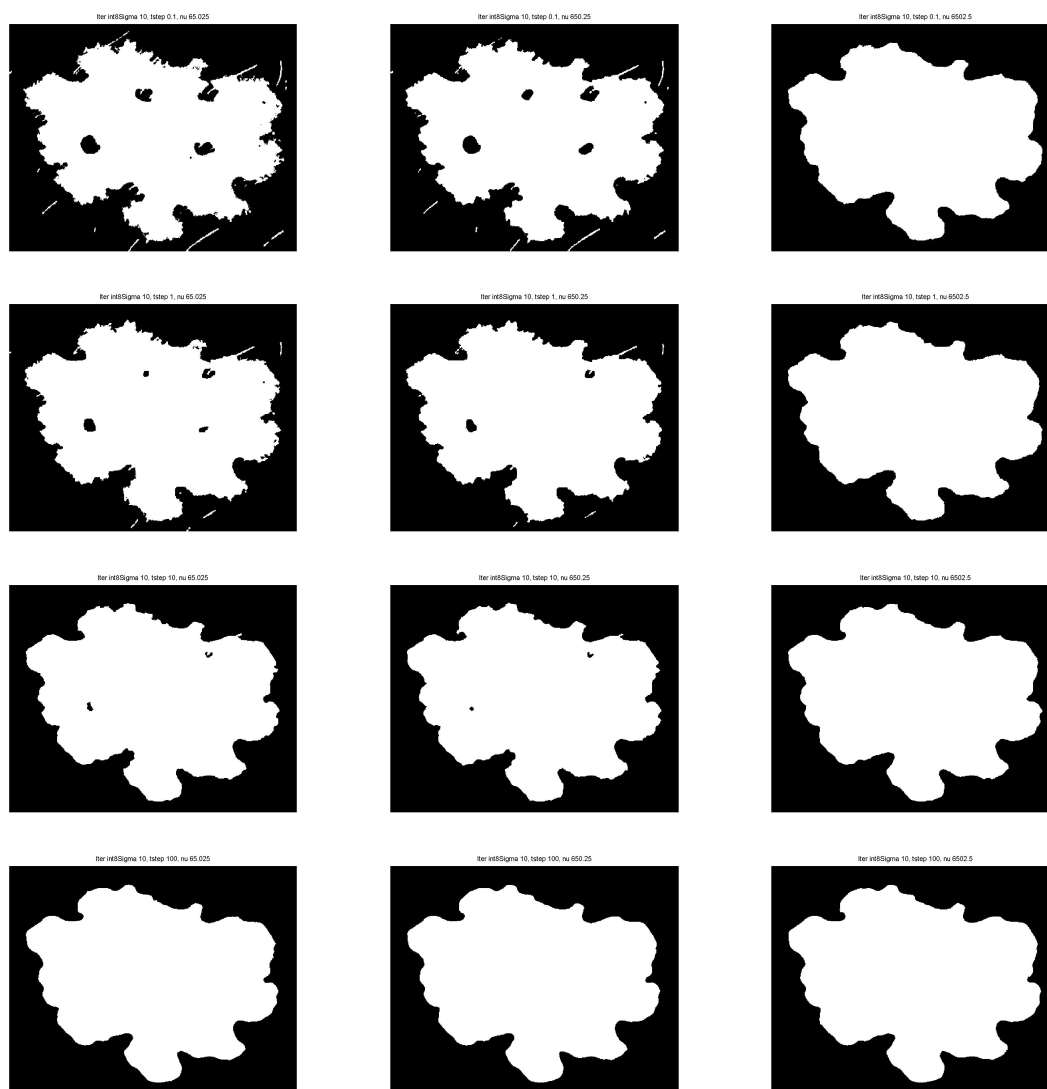


Figura 3.18: Segmentazioni relative al valore  $\sigma = 10$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

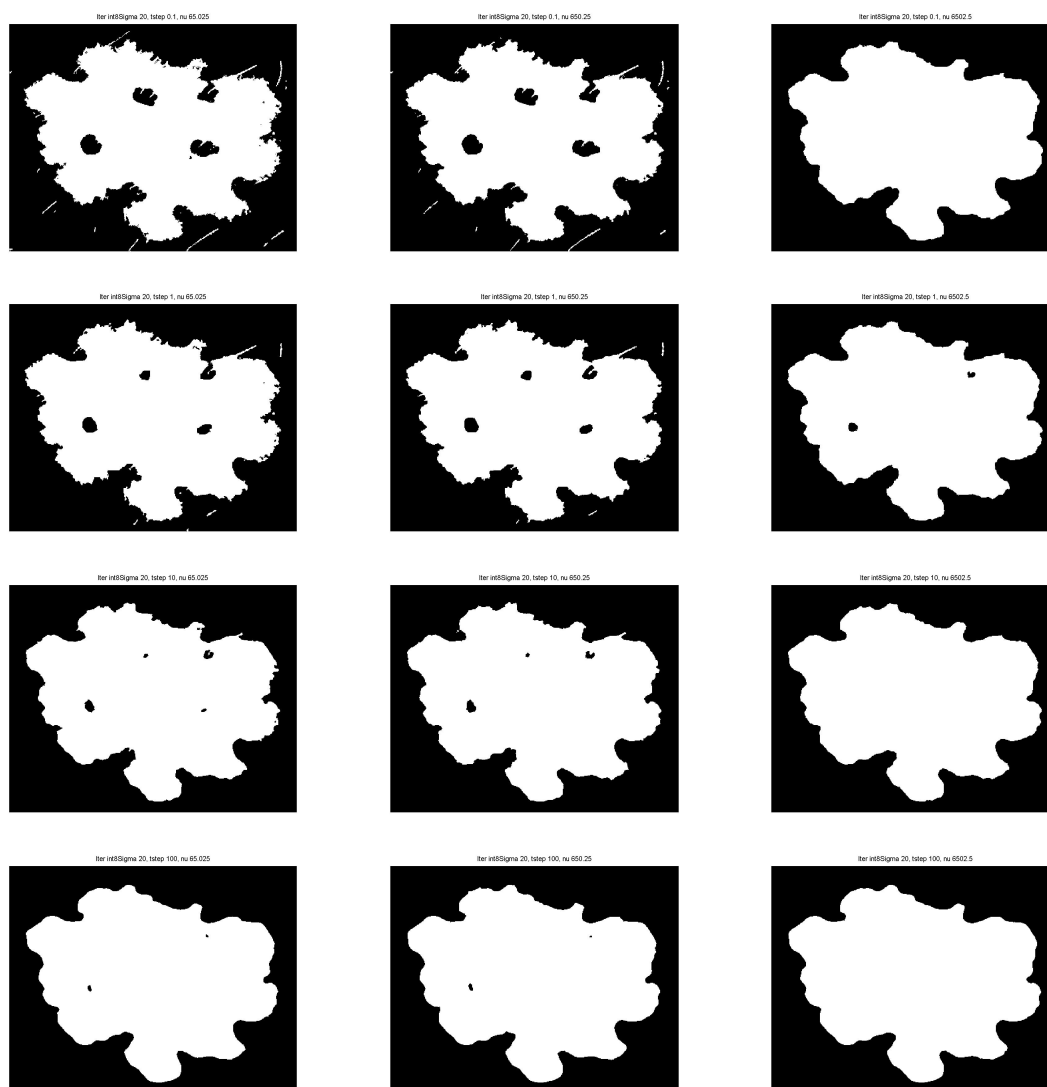


Figura 3.19: Segmentazioni relative al valore  $\sigma = 20$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

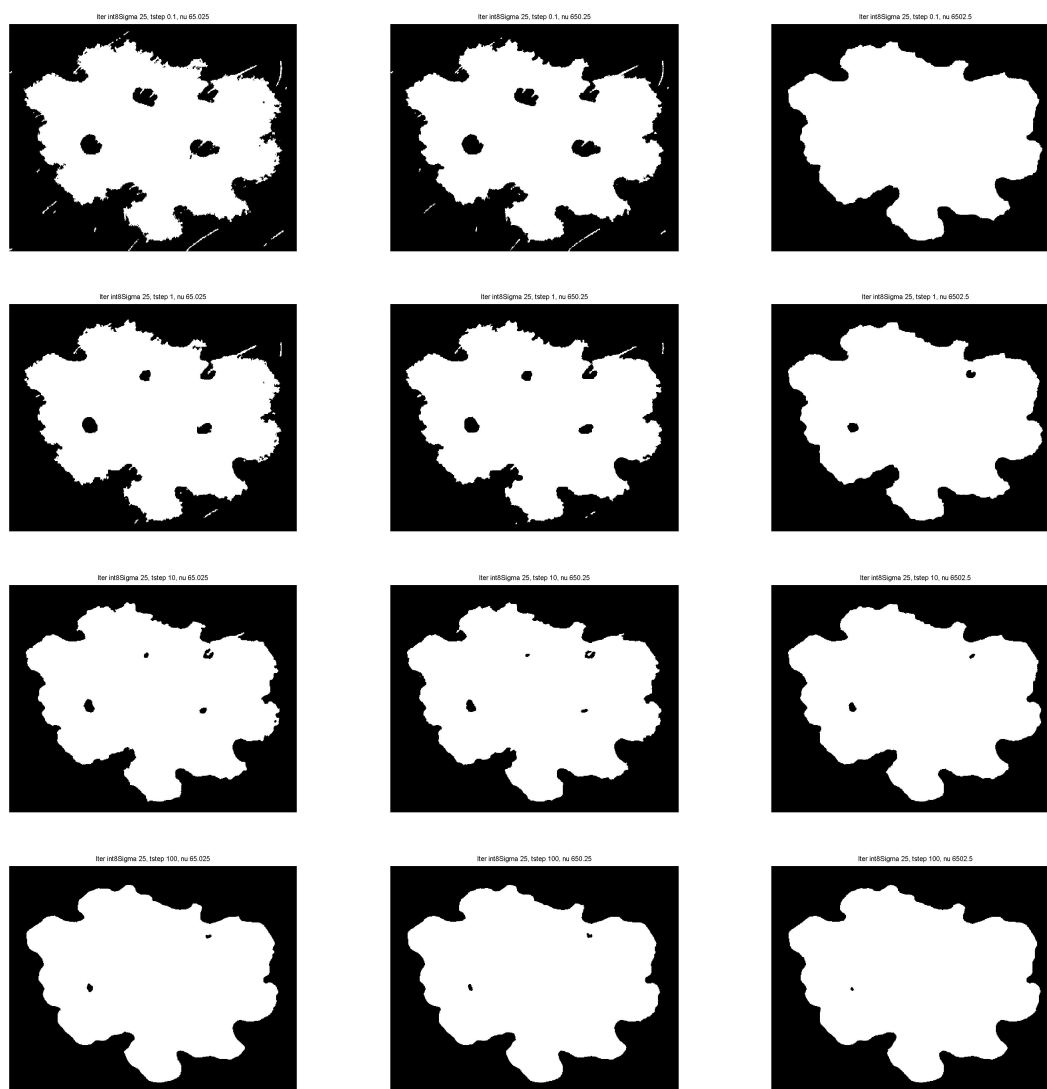


Figura 3.20: Segmentazioni relative al valore  $\sigma = 25$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .



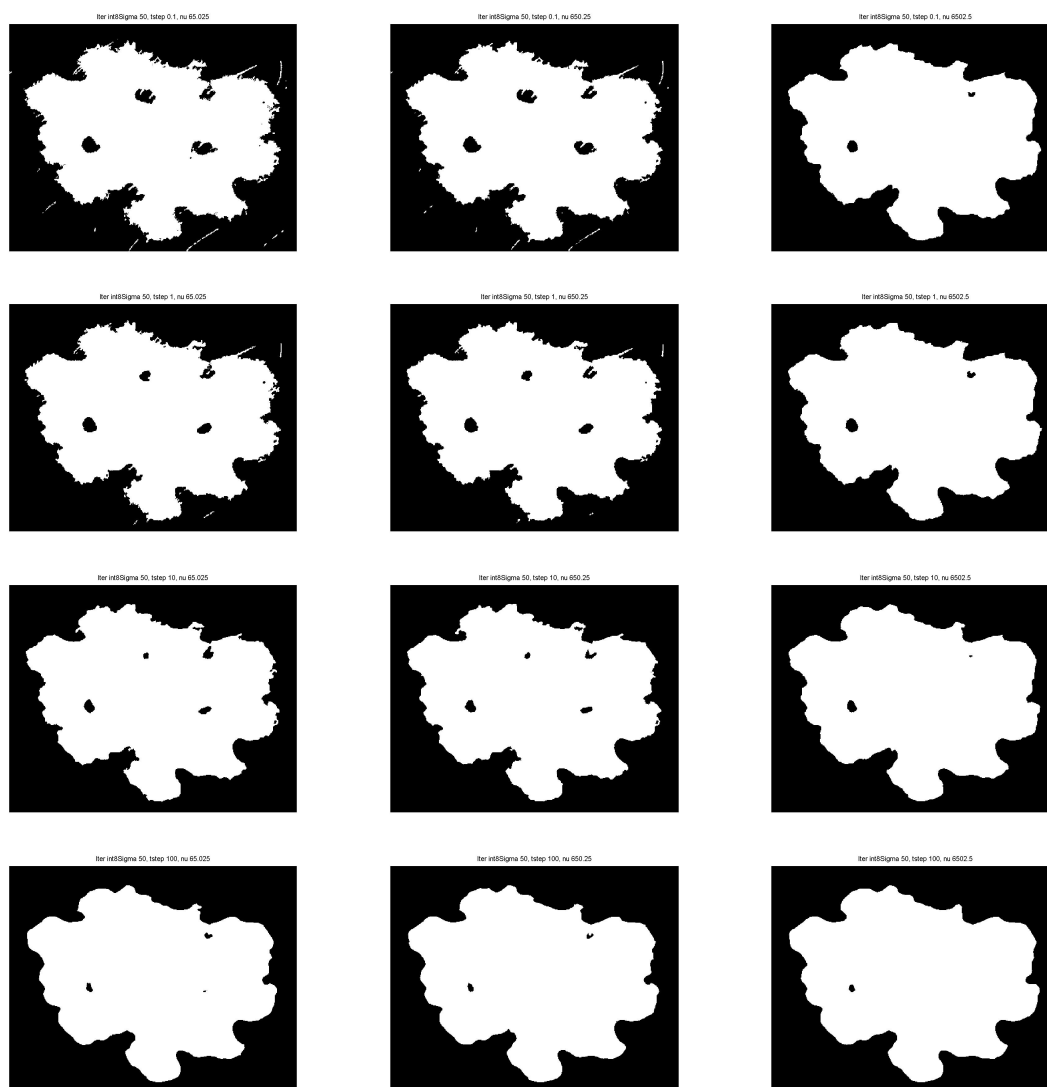


Figura 3.21: Segmentazioni relative al valore  $\sigma = 50$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

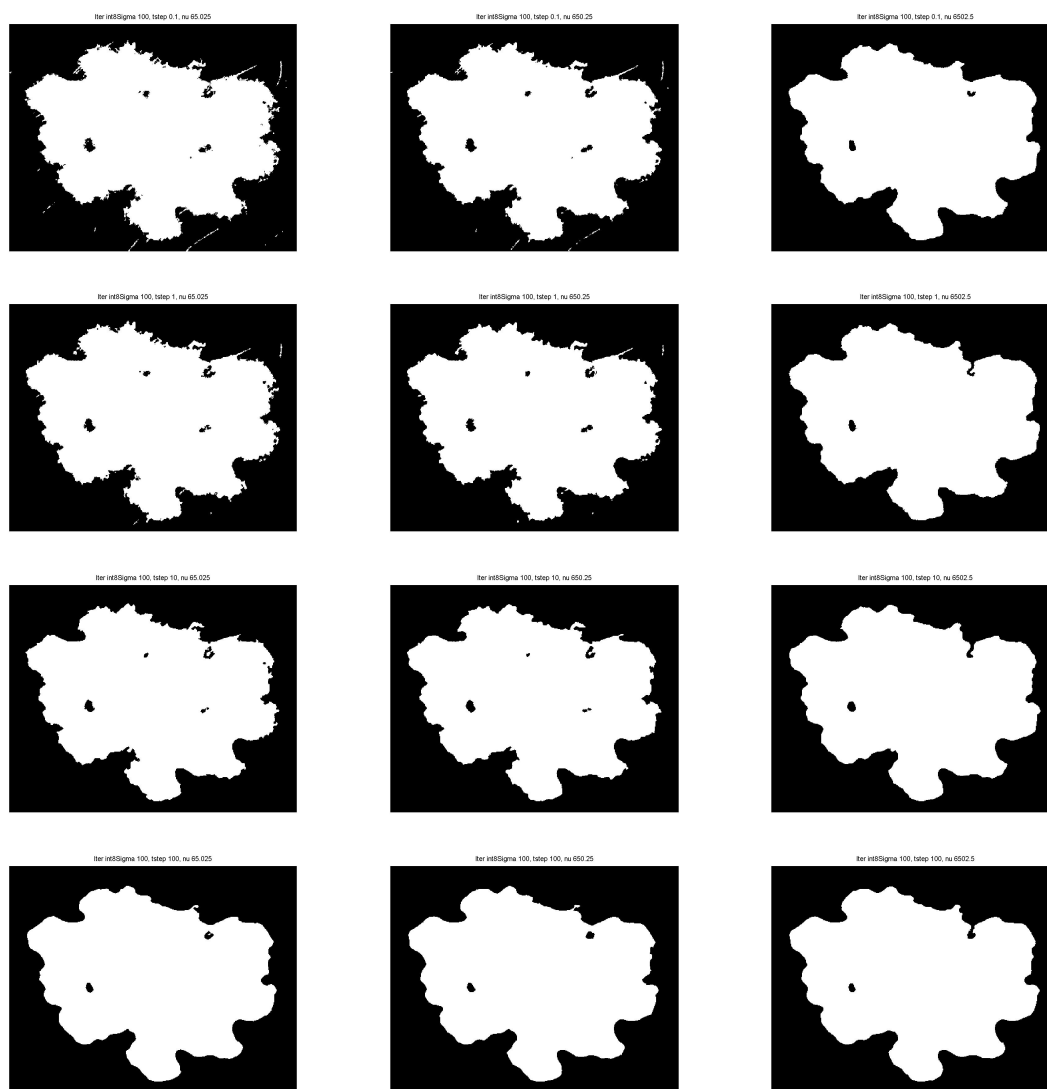


Figura 3.22: Segmentazioni relative al valore  $\sigma = 100$  con 8 iterazioni interne: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

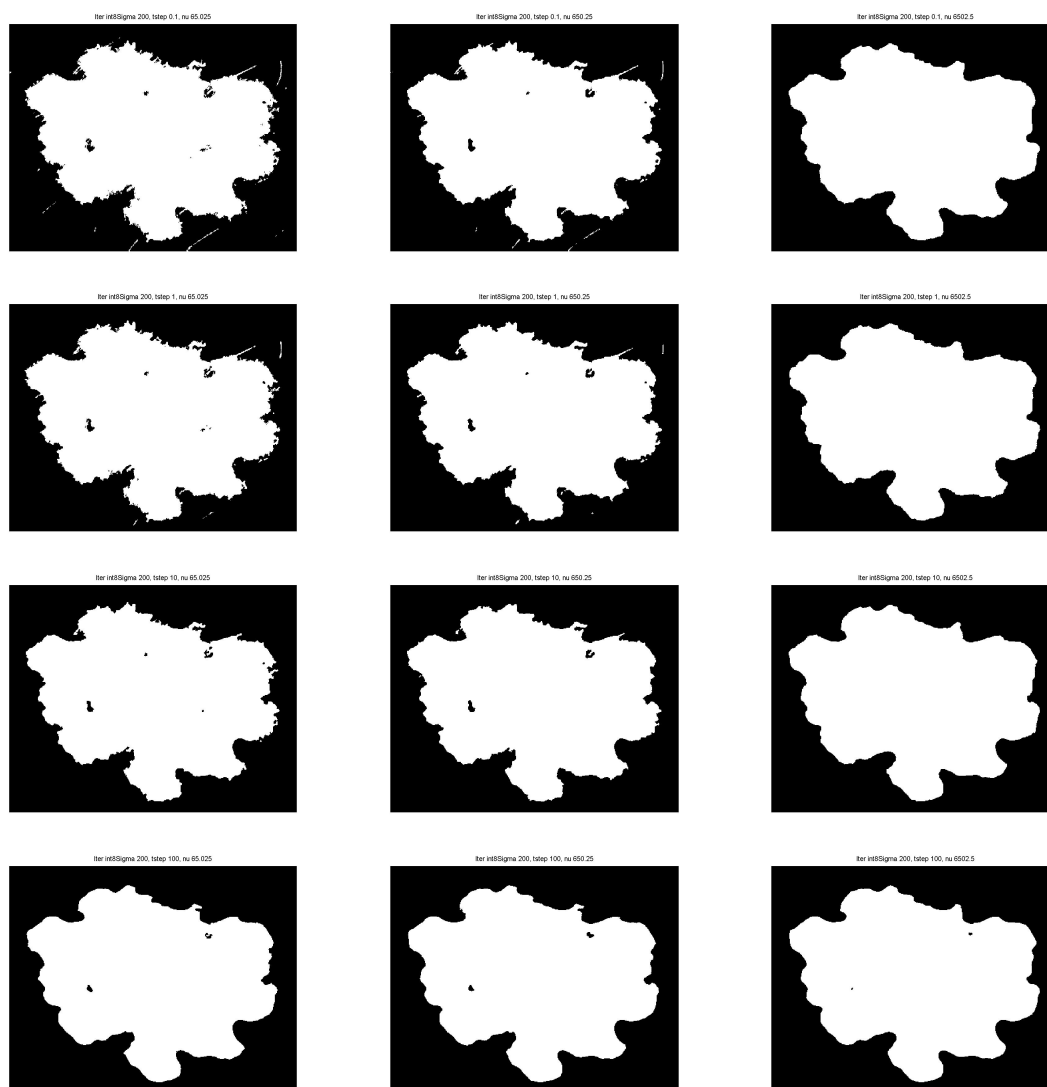


Figura 3.23: Segmentazioni relative al valore  $\sigma = 200$  con 8 iterazioni interni: le righe sono le immagini disposte al variare di  $\Delta t = 0.1, 1, 10, 100$  e le colonne al variare di  $\nu = 65.025, 650.25, 6502.5$ .

**Tempi**

Si riportano di seguito i tempi, calcolati in secondi, necessari per compiere l'aggiornamento della LSF con 9 iterazioni esterne. Le righe indicizzano i valori del passo temporale  $\Delta t$ , le colonne invece indicizzano i valori di  $\nu$

SIGMA 1				SIGMA 2			
0.9377	0.8557	0.8649	0.8620	0.9078	0.9101	0.9092	0.8958
1.0951	0.9340	0.8663	0.8678	0.9074	0.9055	0.9096	0.9071
1.0922	0.8692	0.8620	0.8659	0.9095	0.9060	0.9095	0.9080
1.0910	0.9942	0.8624	0.8671	0.9056	0.9078	0.9068	0.9108
SIGMA 5				SIGMA 10			
1.2605	1.2734	1.4833	1.4213	2.6168	2.6767	2.6874	2.8317
1.2625	1.2492	1.2549	1.2586	2.9552	3.1467	3.1286	3.1658
1.2457	1.2687	1.2432	1.2447	3.0359	3.3048	3.2496	3.1924
1.2473	1.2586	1.2502	1.2525	2.8282	3.2154	3.1728	3.1808
SIGMA 20				SIGMA 25			
10.7474	10.5557	10.2008	10.2803	15.2023	15.1655	14.8933	16.3064
9.8943	9.7068	10.0147	9.7666	14.8390	14.9652	15.0746	14.8642
9.7185	10.0514	10.0158	9.8685	14.7096	14.9671	14.8430	14.5044
9.9798	9.8502	9.8178	9.7510	14.8156	14.6069	14.6419	14.6280
SIGMA 50				SIGMA 100			
57.9724	56.6085	56.3351	54.3140	182.6834	182.5862	183.0202	199.2932
54.1995	54.6117	54.5042	54.2379	183.6359	185.6727	183.8342	185.0958
53.7375	53.9566	54.4757	54.3782	180.2081	180.1563	183.5350	181.9512
53.3611	53.9216	53.8265	53.6768	183.8632	180.6550	180.1565	179.9784

### 3.4 Selezione definitiva dei parametri e confronto con il *thresholding*

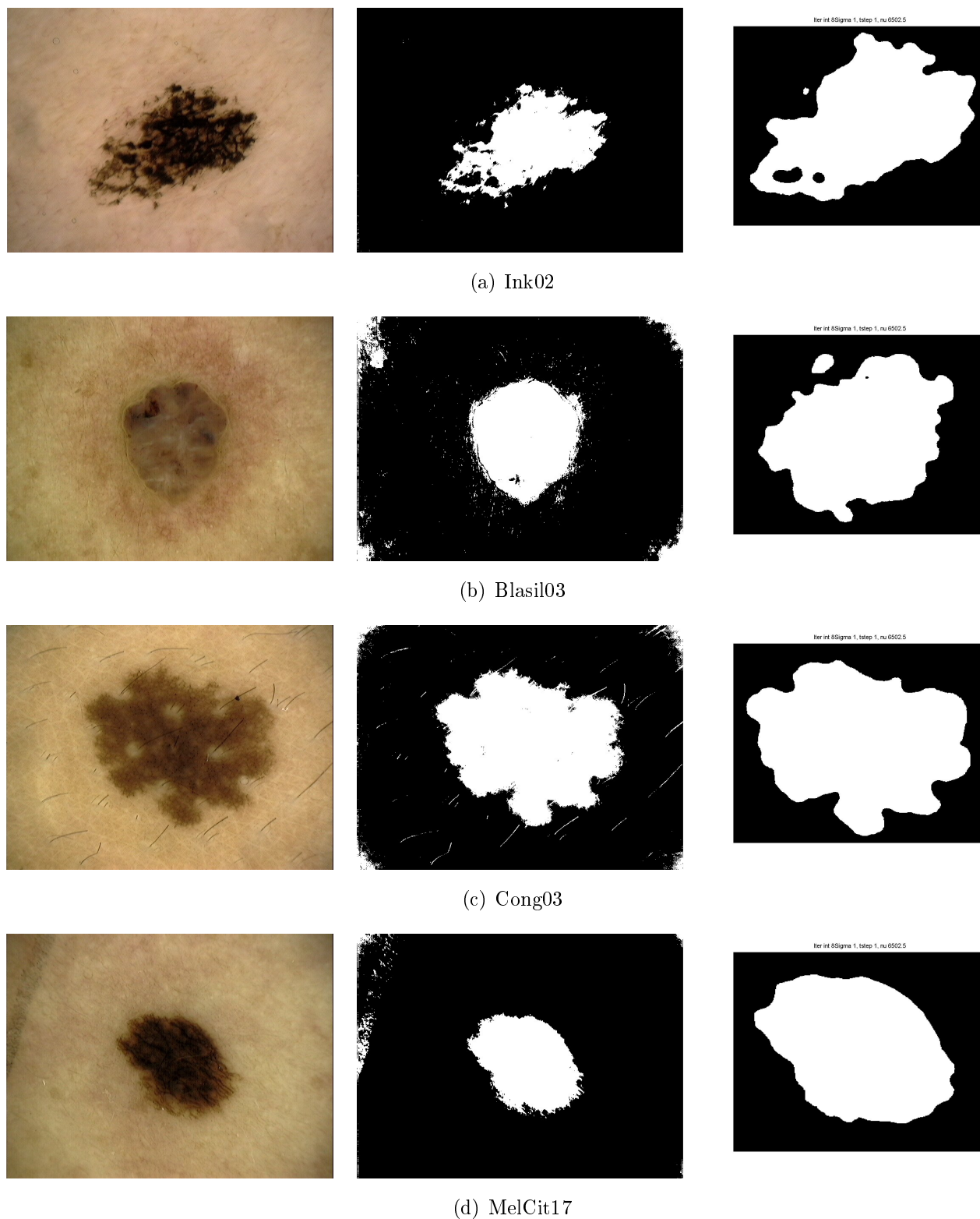
Si è visto dunque che i parametri ottimali per avere una segmentazione regolare, precisa, poco sensibile agli oggetti di disturbo come i peli, che riconosca il più possibile eventuali aloni e che abbia un tempo di esecuzione rapido, dell'ordine dei 5 secondi, sono i seguenti:

- 8 iterazioni interne;
- $\sigma$  compreso tra 1 e 10;
- $\nu = 650.25$  o  $\nu = 6502.5$ ;
- $\Delta t$  tra 1 e 10.

Si sono confrontati i risultati ottenuti con i seguenti valori dei parametri,  $\sigma = 1$ ,  $\nu = 6502.5$ ,  $\Delta t = 1$ , 8 iterazioni interne, con una segmentazione ottenuta tramite *thresholding*. La segmentazione ottenuta tramite il nostro metodo, risulta più efficace, in quanto:

- si ottengono delle segmentazioni molto più regolari;
- a differenza del *thresholding*, il metodo è sensibile alla presenza di aloni attorno alla parte centrale del nevo; è più sensibile, ma non è impeccabile, in quanto, come si vede dal nevo nella riga (b) dell'immagine (3.24), l'alone che compare attorno alla parte centrale del nevo non è segmentato in maniera adatta; è da notare che in relazione a questo nevo, il *thresholding* non riconosce in alcun modo la zona di colore di intensità più debole all'esterno della parte centrale. Si nota invece che in (3.25(a)) e in (3.25(c)) le informazioni ottenute con il nostro metodo sono qualitativamente più significative di quelle ottenute tramite *thresholding*;

- i tempi del *thresholding* sono minori di quelli del metodo preso in considerazione, anche se quest'ultimo ammette dei tempi di esecuzione molto bassi per un metodo *level set*. L'efficacia, per quanto riguarda la segmentazione dei nevi, ripaga a pieno il costo computazionale e temporale necessario per attuarlo.

Figura 3.24: Confronto con *thresholding*

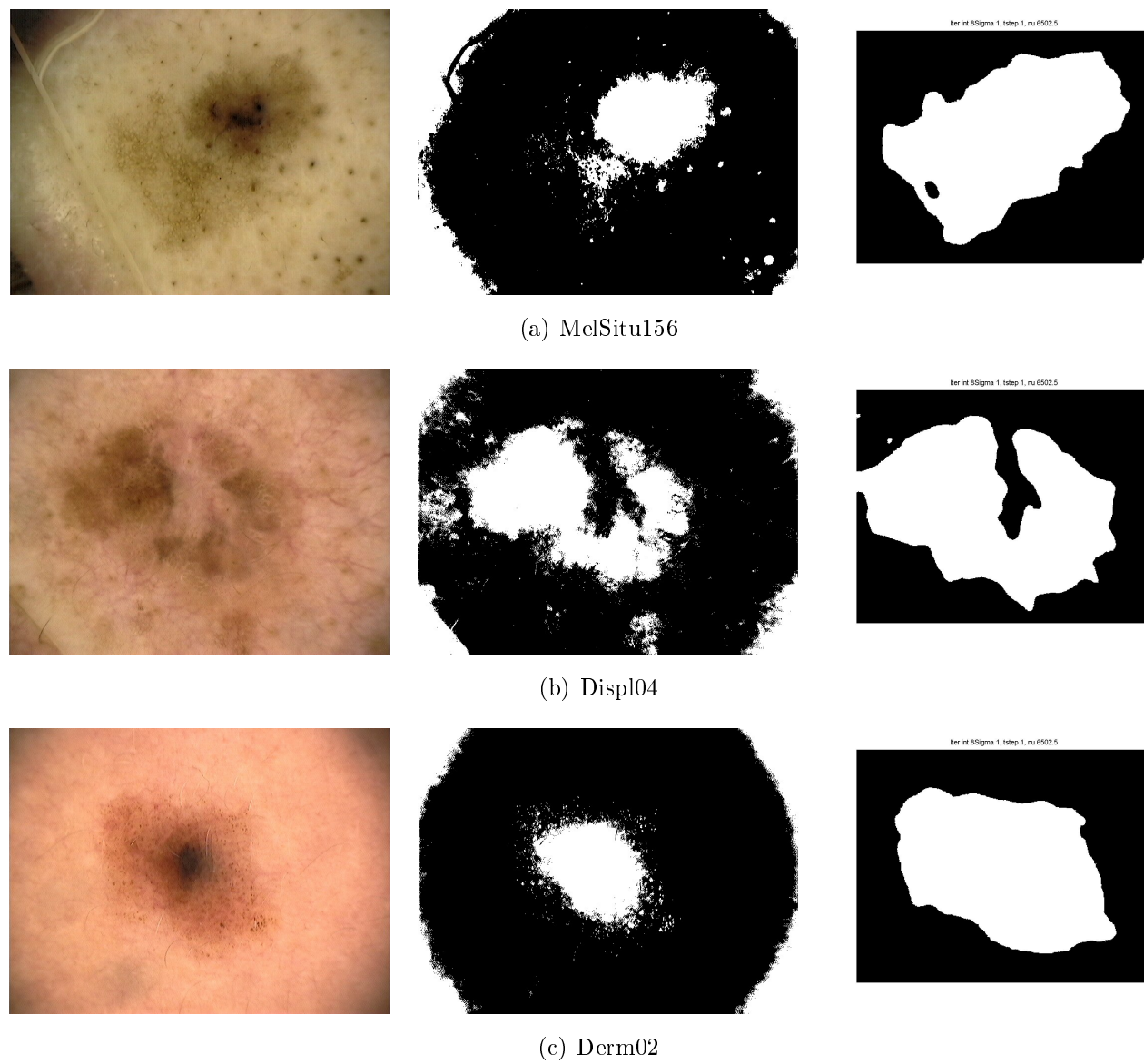


Figura 3.25: Confronto con *thresholding*



# Conclusioni

Il metodo proposto in questa tesi risulta essere adeguato agli obiettivi proposti: sono infatti soddisfacenti i risultati ottenuti sulla maggioranza dei nevi testati. Se da una parte rimangono delle difficoltà ad individuare aloni molto deboli attorno al centro di un nevo, dall'altra il *software* si è dimostrato adattabile e funzionante per la segmentazione di nevi con caratteristiche molto diverse. Sebbene si tratti di un metodo *level set*, i tempi e il costo computazionale sono molto bassi e questa caratteristica risponde alle necessità, esposte all'inizio di questa tesi, di avere una segmentazione adatta in tempi inferiori alla decina di secondi.

Possibili sviluppi futuri sono l'implementazione *narrow band* del metodo, che ridurrebbe ulteriormente i tempi di esecuzione, e una possibile ricostruzione 3D del nevo usando metodi *level set* in dimensioni maggiori.

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Metodi <i>level set</i> e il problema del riconoscimento di bordi</b>	<b>4</b>
<b>Metodi <i>level set</i></b>	<b>4</b>
1.1 Formulazione <i>lagrangiana</i> della propagazione di un fronte . . . . .	5
1.2 Formulazione <i>euleriana</i> della propagazione di un fronte . . . . .	8
1.2.1 Una rappresentazione implicita . . . . .	8
1.2.2 Legami con l'equazione di <i>Hamilton-Jacobi</i> : proprietà e conseguenze	14
1.2.3 La reinizializzazione . . . . .	15
1.2.4 Vantaggi e svantaggi dei metodi <i>level set</i> . . . . .	16
1.3 La segmentazione di immagini . . . . .	17
1.3.1 Una breve introduzione al problema della segmentazione . . . . .	17
1.3.2 Metodi di segmentazione . . . . .	18
<b>2 Il metodo DRSLE</b>	<b>21</b>
<b>Un modello variazionale senza reinizializzazione</b>	<b>21</b>
2.1 Descrizione del metodo . . . . .	22
2.2 Applicazione del metodo DRLSE alla segmentazione di un'immagine . . .	28
2.2.1 <i>Edge-based method</i> . . . . .	28
2.2.2 <i>Region-based method</i> . . . . .	30
2.2.3 Implementazione numerica . . . . .	35

---

<b>3 Prove sperimentali</b>	<b>36</b>
<b>Prove sperimentali</b>	<b>36</b>
3.1 Obiettivi . . . . .	36
3.2 Descrizione delle prove . . . . .	38
3.3 Scelta dei parametri . . . . .	41
3.3.1 Variazione di $\nu$ . . . . .	43
3.3.2 Variazione di $\Delta t$ . . . . .	46
3.3.3 Variazione delle iterazioni interne . . . . .	47
3.3.4 Variazione di $\sigma$ . . . . .	51
3.3.5 Risultati . . . . .	55
3.4 Selezione definitiva dei parametri e confronto con il <i>thresholding</i> . . . . .	66
<b>Conclusioni</b>	<b>70</b>
<b>Bibliografia</b>	<b>75</b>

# Elenco delle figure

1.1	Formulazione <i>lagrangiana</i> . . . . .	6
1.2	Fronte spigoloso . . . . .	6
1.3	Cambio di topologia del fronte . . . . .	7
1.4	Esempio di <i>level set function</i> . . . . .	9
3.1	Cut di un'immagine . . . . .	38
3.2	Funzione iniziale . . . . .	39
3.3	Iterazioni esterne . . . . .	41
3.4	Nevo prova . . . . .	43
3.5	Variazione di $\nu$ . . . . .	44
3.6	Immagine iniziale di un nevo . . . . .	45
3.7	Variazione di $\Delta t$ . . . . .	47
3.8	Variazione iterazioni interne (1) . . . . .	48
3.9	Variazione iterazioni interne (2) . . . . .	48
3.10	Variazione iterazioni interne (3) . . . . .	49
3.11	Istogramma dei tempi . . . . .	50
3.12	Variazione di $\sigma$ (1) . . . . .	52
3.13	Variazione di $\sigma$ (2) . . . . .	53
3.14	Variazione di $\sigma$ (3) . . . . .	54
3.15	Segmentazione nevo $\sigma = 1$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	56
3.16	Segmentazione nevo $\sigma = 2$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	57

---

3.17	Segmentazione nevo $\sigma = 5$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	58
3.18	Segmentazione nevo $\sigma = 10$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	59
3.19	Segmentazione nevo $\sigma = 20$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	60
3.20	Segmentazione nevo $\sigma = 25$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	61
3.21	Segmentazione nevo $\sigma = 50$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . . .	62
3.22	Segmentazione nevo $\sigma = 100$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . .	63
3.23	Segmentazione nevo $\sigma = 200$ , 8 iterazioni interne, al variare di $\nu$ e $\Delta t$ . .	64
3.24	Confronto <i>thresholding</i> (1) . . . . .	68
3.25	Confronto <i>thresholding</i> (2) . . . . .	69

# Bibliografia

- [1] S. Osher e J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., vol. 79, no. 1, pp. 12-49, Nov. 1988.
- [2] V. Caselles, F. Catte, T. Coll. e F. Dibos, *A geometric model for active contours in image processing*, Numer. Math., vol. 66, pp.1-31. 1993
- [3] R. Malladi, J.A Sethian e B.C. Vermuri, *Shape modeling with front propagation: a level set approach*, IEEE Trans. Patt. Anal. Mach. Intell., vol. 17, pp.158-175, 1995.
- [4] S. Osher e Ronald P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, New York: Springer-Verlag, 2002
- [5] G. Barles, H.M Soner, e P.E. Souganidis. *Front propagation and phase field theory*, SIAM J. Control and Optimization, 31(2): 439-469, March 1993
- [6] J. Gomes e O. Faugeras, *Reconciling distance functions and level sets*, J. Vis. Commun. Image Represent, vol. 11, no. 2, pp. 209-223, Jun 2000
- [7] J. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge, U.K.: Cambridge Univ. Press, 1999.
- [8] R.C. Gonzalez e R.E. Woods, *Digital Image Processing*, Prentice-Hall, 2002

- 
- [9] M. Kass, A. Witkin e D. Terzopoulos, *Snakes: Active contour models*, First International Conference on Computer Vision, pp. 259-268, London, June 1987
- [10] D. Mumford e J. Shah, *Boundary detection by minimizing functional*, Proceedings of the International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, June 1985. IEEE
- [11] D. Mumford e J. Shah, *Optimal approximations by piecewise smooth functions associated variational problems*, Communications on Pure and Applied Mathematics, 42: 577-684, 1989
- [12] T. Chan e L. Vese, *Active contours without edges*, IEEE Transactions on Image Processing, 10(2):266-277, Febbraio 2001
- [13] M. Weber, A. Blake and R. Cipolla, *Sparse inite elements for geodesic contour with level sets*, Proc. Europ. Con. Comput. Vis., 2004, pp. 391-404
- [14] C. Li, C. Cu, C. Gui and M.D. Fox, *Level Set Evolution Without Re-initialization: A New Variational ormulation*, Proc. IEEE Conf. Comput. Vis. Pattern Recognition, 2005, Vol. 1, pp.430-436
- [15] C. Li, C. Xu, C. Gui and M.D. Fox, *Distance Regularized Level Set Evolution and its Appllication to Image Segmentation*, IEEE trans. Image Process., vol. 19, no.12, pp. 3243-3254, Dec.2010
- [16] C. Li, R. Huang, Z. Ding, J. Chris Gatenby, D.N. Metaxas, J.C. Gore, *A Level Set Method for Image Segmentation in the Presence of Intensity Inhomogeneities With Application to MRI*, IEEE Trans. Image Progress., vol 20, no.7, July 2011
- [17] G. Aubert e P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential equations and Calculus of Variation*, New York: Springer-Verlag, 2002
- [18] T. Chan and L. Vese, *Active contours without edges*, IEEE Trans. Image. Process., vol. 10, no. 2, pp. 266-277, Feb. 2001

- 
- [19] C. Li, C. Xu, K.M. Konwar, M.D. Fox, *Fast distance preserving Level Set Evolution fo Medical Image Segmentation*, in Proc. 9th Int. Conf. Control autom. Robot. Vis., 2006, pp. 1-7