



HAL
open science

Image Segmentation and Level Set Method: Application to Anatomical Head Model Creation

Jérôme Piovano

► **To cite this version:**

Jérôme Piovano. Image Segmentation and Level Set Method: Application to Anatomical Head Model Creation. Computer Vision and Pattern Recognition [cs.CV]. Université de Nice-Sophia Antipolis, 2009. English. NNT: . tel-03374002

HAL Id: tel-03374002

<https://inria.hal.science/tel-03374002>

Submitted on 11 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PhD THESIS

prepared at
INRIA Sophia Antipolis

and presented at the
University of Nice-Sophia Antipolis
Graduate School of Information and Communication Sciences

*A dissertation submitted in partial fulfillment
of the requirements for the degree of*

DOCTOR OF SCIENCE
Specialized in Signal and Image Processing

Image Segmentation and Level Set Method: Application to Anatomical Head Model Creation

Jérôme PIOVANO

Advisor	Théodore Papadopoulo	INRIA Sophia-Antipolis, France
Reviewers	Luis Alvarez	University of Las Palmas, Spain
	Renaud Keriven	CERTIS / ENPC, France
Examiners	Michel Barlaud	University of Nice-Sophia Antipolis, France
	Jean-Pierre Cocquerez	University of Technology of Compiègne, France

UNIVERSITÉ NICE-SOPHIA ANTIPOLIS - UFR Sciences

École Doctorale STIC

(Sciences et Technologies de l'Information et de la Communication)

THÈSE

pour obtenir le titre de

DOCTEUR EN SCIENCES

de l'UNIVERSITÉ de Nice-Sophia Antipolis

Discipline: Automatique, Traitement du Signal et des Images

présentée et soutenue par

Jérôme PIOVANO

Segmentation d'Image par Ensembles de Niveaux: Application à la Création de Modèles Anatomiques de la Tête

Thèse dirigée par Théodore PAPADOPOULO

Date prévue de soutenance, 15 janvier 2009

Composition du jury:

<i>Rapporteurs</i>	Luis Alvarez	Université de Las Palmas, Espagne
	Renaud Keriven	CERTIS / ENPC, France
<i>Examineurs</i>	Michel Barlaud	Université de Nice-Sophia Antipolis, France
	Jean-Pierre Cocquerez	Université de Technologie de Compiègne, France

Titre : Segmentation d'Image par Méthode des Ensembles de Niveaux et Application à la Création de Modèles Anatomiques de la Tête

Résumé : L'apparition des techniques d'imagerie par résonance magnétique (IRM) à la fin du XXe siècle a révolutionné le monde de la médecine moderne, en permettant de visualiser avec précision l'intérieur de structures anatomiques de manière non invasive. Cette technique d'imagerie a fortement contribué à l'étude du cerveau humain, en permettant de discerner avec précision les différentes structures anatomiques de la tête, notamment le cortex cérébral.

Le discernement de structures anatomiques de la tête porte le nom segmentation, et consiste à "extraire" des régions dans les IRMs. Plusieurs méthodes de segmentation existent, et cette thèse porte sur les méthodes à base d'évolution d'hyper-surfaces: une hyper-surface (surface en 3D) est progressivement déformée pour finalement épouser les frontières de la région à segmenter.

Un modèle de tête correspond au partitionnement de la tête en plusieurs structures anatomiques préalablement segmentées. Un modèle de tête classique comprends en général 5 structures anatomiques (peau, crâne, liquide céphalo rachidien, matière grise, matière blanche), imbriquées les unes dans les autres à la manière de "poupée russes". Néanmoins de par la complexité de leurs formes, segmenter ces structures manuellement s'avère pénible et extrêmement difficile.

Cette thèse se consacre à la mise en place de nouveaux modèles de segmentation robustes aux altérations d'IRMs, et à l'application de ces modèles pour la création automatique de modèles anatomiques de la tête. Après avoir brièvement survolé un état de l'art des différentes méthodes de segmentation d'image, deux contributions à la segmentation par évolution d'hyper-surface sont proposées. La première constitue une nouvelle représentation et un nouveau schéma numérique pour la méthode des ensembles de niveaux, en utilisant des éléments finis quadrilatéraux. Cette représentation vise à améliorer la qualité et la robustesse du modèle. La seconde contribution constitue un nouveau modèle de segmentation basé sur des statistiques locales, robuste aux altérations présentes dans les IRMs. Ce nouveau modèle vise à unifier plusieurs modèles "état de l'art" en segmentation d'image. Enfin, un cadre pour la création automatique de modèle de tête est proposé, utilisant principalement le précédent modèle de segmentation par statistiques locales.

Mots clés : segmentation, ensembles de niveaux, méthode des éléments finis, statistiques locales, modèle anatomique de la tête

Title: Image Segmentation with Level Set Method and Application to Anatomical Head Model Creation

Abstract: Magnetic Resonance Images (MRI) have been introduced at the end of the XXth century and have revolutionized the world of modern medicine, allowing to view with precision the inside of anatomical structures in a non-invasive way. This imaging technique has greatly contributed to the study and comprehension of the human brain, allowing to discern with precision the different anatomical structures composing the head, especially the cerebral cortex.

Discernment between these anatomical structures is called segmentation, and consist in “extracting” structures of interest from MRIs. Several models exists to perform image segmentation, and this thesis focus on those based on hypersurface evolutions: an hypersurface (surface in 3D) is incrementally adjusted to finally fit the border of the region of interest.

A head model corresponds to the partitioning of the head into several segmented anatomical structures. A classic head model generally includes 5 anatomical structures (skin, skull, cerebrospinal fluid, grey matter, white matter), nested inside each other in the manner of “Russian nested dolls”. Nevertheless because of the complexity of their shapes, manual segmentation of these structures is tedious and extremely difficult.

This thesis is dedicated to the creation of new segmentation models robust to MRI alterations, and to the application of these models in the purpose of automatic creation of anatomical head models. After briefly reviewing some classical models in image segmentation, two contributions to segmentation based on hypersurface evolution are proposed. The first one corresponds to a new representation and a new numerical scheme for the level-sets method, based on quadrilateral finite elements. This representation aims at improving the accuracy and robustness of the model. The second contribution corresponds to a new segmentation model based on local statistics, and robust to standard MRI alterations. This model aims at unifying several 'state-of-the-art' models in image segmentation. Finally, a framework for automatic creation of anatomical head models is proposed, mainly using the previous local-statistic based segmentation model.

Keywords: Segmentation, Level-Sets, Finite Element Method, Local Statistics, Anatomical Head Model

Contents

Résumé	3
Abstract	5
1 Introduction (version française)	11
Contexte	11
Besoin de modeles anatomiques	12
Segmentation d'image	12
Organisation et contribution de cette thèse	13
2 Introduction	17
Context	17
Need for Anatomical Model of the head	18
Image Segmentation	18
Organization and Contributions of this Thesis	19
3 Image Segmentation	21
Overview	22
3.1 Introduction	22
3.2 Discrete Models: Markov Random Fields and Graph Cuts	24
3.2.1 Markov Random Field	24
3.2.2 Graph Cuts	25
3.2.3 Random Walker	25
3.3 Continuous Models: Snakes and Active Contours	26
3.3.1 The Snake Model	26
3.4 Front Representation: Explicit models vs Implicit models	27
3.4.1 Explicit Representation	27
3.4.2 Implicit Representation: The Level-Set Method	28
3.5 Boundary Based Segmentation Models	31
3.5.1 Geodesic Active contour	32
3.5.2 Gradient Vector Flow	34
3.6 Region based segmentation models	35
3.6.1 Functional-based models	37

3.6.2	Bayesian-based models	39
3.7	Conclusion	43
4	Implementing Level Sets using the Finite Element Method	47
	Overview	48
4.1	Introduction	48
4.2	Finite Element Method	50
4.2.1	Variational Formulation	50
4.2.2	Discretization of the solution	52
4.2.3	Discretization of the weak formulation	54
4.3	Finite Element Implementation of the Level Set Method	55
4.3.1	Formulation of the Problem	55
4.3.2	Time Discretization : implicit scheme	56
4.3.3	Variational Formulation	57
4.3.4	Discretization of the Solution & Algorithmics	58
4.3.5	Discretization of the weak formulation	61
4.3.6	Solving the system	63
4.4	Results and Comparisons	65
4.5	Conclusion & discussion	66
5	Image Segmentation using Local Statistics	71
	Overview	72
5.1	Introduction	72
5.2	Region Segmentation using Local Statistics	75
5.2.1	Piecewise Smooth Regions and Local Means	75
5.2.2	Bayesian Formulation and Local Region Statistics	77
5.3	Varying the Neighborhood	80
5.3.1	Small Neighborhood	81
5.3.2	Intermediate Neighborhood	82
5.3.3	Large Neighborhood	84
5.4	Coherence Improvement	84
5.4.1	Coherence Problem	85
5.4.2	Coherence Solution	85
5.5	Multiscale Improvement	88
5.5.1	Homogeneous Regions Problem	89
5.5.2	Multiscale Approach	89
5.6	Efficient Implementation	91
5.6.1	Parallelized Finite-Differences based Level-Sets	91
5.6.2	Recursive Gaussian filter	91
5.6.3	Scale Selection using GPU	93
5.7	Results	94
5.8	Conclusion	96

6	Head Models from Bimodal MRIs using Local Statistics	101
	Overview	102
6.1	Introduction	102
6.2	Anatomical representation of the head model	105
6.3	Creation of Head Model from Bi-modal Anatomical MRIs	106
6.3.1	MRIs Acquisition & Registration	106
6.3.2	Scalp Segmentation	108
6.3.3	CSF Segmentation	109
6.3.4	WM Segmentation	110
6.3.5	GM Segmentation	112
6.3.6	Skull Segmentation	116
6.4	Validation	117
6.5	Results	118
6.6	Conclusion	122
7	Conclusion	125
8	Conclusion (version française)	129
	Appendix	133
A	Details about finite-element level-set implementation	135
A.1	Calculus of basis functions	135
A.2	Q1 basis functions	136
A.2.1	2D basis functions	136
A.2.2	3D basis functions	137
A.3	Calculus of distance constraint system	137
A.3.1	2D case	138
A.3.2	3D case	139
A.4	Calculus of evolution equation system	141
A.5	Conjugate gradient method	142
B	Energy Derivation of the Piecewise Smooth Model	145
B.1	Derivation of the energy	145
B.2	Implementation	147
	Bibliography	149

INTRODUCTION (VERSION FRANÇAISE)

CONTEXTE

Les neurosciences sont un domaine fascinant. Elles ont gagné en popularité au cours des dernières décennies, principalement grâce aux récentes avancées dans le domaine de l'imagerie médicale et fonctionnelle. Les premiers travaux de recherche consistaient à analyser les formes du cerveau pour en déceler certaines caractéristiques, mais n'étaient d'aucune aide pour comprendre l'activité neuronale. Avec l'apparition d'équipements permettant des études fonctionnelles du cerveau, plusieurs méthodes ont émergées pour étudier en temps réel l'activité neuronale (TEP-scan, IRMf, Optical Imaging, EEG, MEG ...). Chacune de ces méthodes est basé sur une hypothèse sous-jacente de cette activité:

- la Tomographie par Émission de Positron (TEP-scan), et l'Imagerie par Résonance Magnétique fonctionnelle (IRMf) se basent sur le fait que les changements de circulation sanguine et d'oxygénation du sang dans le cerveau (connu sous le nom d'*hémodynamique*) sont étroitement liés à l'activité neuronale. Le TEP-scan mesure le flux sanguin à l'intérieur du cerveau, par l'intermédiaire de traceurs radioactifs préalablement injecté dans le sang. L'IRMf se base sur le même principe que les IRMs classiques, mais se différencie en étant plus sensible à l'oxygène. Cette méthode est donc capable de détecter les grandes quantités d'oxygène libérées par le sang au niveau des neurones actifs.
- l'ElectroEncephaloGraphie (EEG) et la MagnetoEncephaloGraphie (MEG) se basent sur le fait que l'activité neuronale produit un courant électrique, en raison de la nature chimique des connexions synaptiques. L'EEG mesure les potentiels électrique au niveau du scalp, tandis que la MEG mesure le champ magnétique induit par cette activité électrique [113].

Ces méthodes ont chacune leurs propres avantages et leurs propres inconvénients. Elles sont caractérisées par leurs *résolutions spatiales* et *résolutions temporelles*,

et visent à minimiser ces 2 grandeurs. Les méthodes fournissant la meilleure résolution temporelle sont celles basées sur la mesure de l'activité électrique neuronale (MEG/EEG).

BESOIN DE MODELES ANATOMIQUES _____

Cependant, la résolution spatiale des méthodes d'EEG/MEG dépend de la précision de reconstruction du *problème inverse*, qui consiste à estimer l'emplacement des sources électriques à l'intérieur du cerveau à partir de la série de mesures sur le scalp. Cette reconstruction est réalisée grâce aux *équations de Maxwell*, qui décrivent des propriétés reliant les champs électriques et magnétiques à leurs sources électriques respectives (on cherche donc à résoudre ces équations "à l'envers"). Toutefois, il est dit dans ces équations que les champs électriques et magnétiques sont très sensibles à la nature des zones qu'ils traversent (surtout pour l'EEG).

La tête humaine est très complexe et comprend divers types de structures anatomiques conduisant différemment l'électricité. Par conséquent, afin de réaliser correctement la reconstruction inverse, l'emplacement des différentes structures anatomiques de la tête doit être connu à l'avance. Estimer l'emplacement de ces structures consiste à créer un *modèle de tête*, qui partitionne la tête suivant la conductivité des tissus. En général, ces modèles comprennent 5 structures anatomiques, qui sont imbriquées les unes et dans les autres à la manière de "poupées russes" (Fig. 1.1). De l'extérieur vers l'intérieur, ces modèles sont constitués de : la peau, le crâne, le liquide céphalo-rachidien (LCR), la matière grise (MG), et la matière blanche (MB) [8].

La structure la plus importante dans ces modèles est sans doute la matière grise, qui constitue *le cortex cerebral*. Anatomiquement, le cortex est une "couche" de quelques millimètres d'épaisseur, fortement convoluée et de topologie sphérique. C'est dans cette couche que reposent tous les neurones du cerveau, la matière blanche ("intérieur" du cerveau) ne contenant que les fibres nerveuses reliant les neurones entre elles. La détection de l'activité neuronale se fait à l'intérieur du cortex, ce qui implique que la qualité de son modèle doit être primordiale.

Plusieurs méthodes existent pour effectuer ces types de modèles, et appartiennent au domaine de la *segmentation d'image*.

SEGMENTATION D'IMAGE _____

La segmentation d'image consiste à partitionner une image en "zones significative". Ces zones correspondent en général à des objets à extraire et peuvent être considérées comme des régions séparées par des frontières distinctes ou des régions caractérisées par des propriétés globales. Dans notre cas, ces zones correspondent

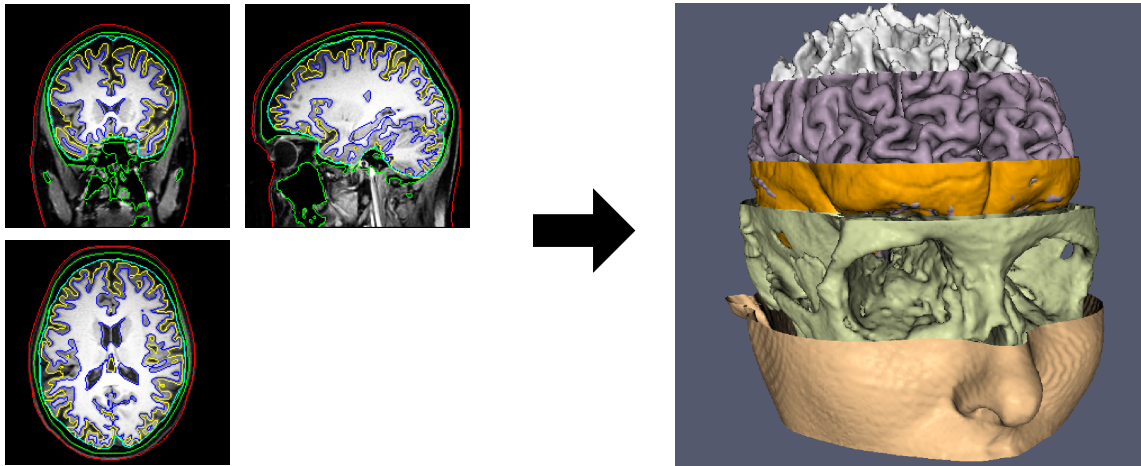


Figure 1.1: Création d'un modèle de la tête à partir d'IRMs.

aux différentes structures anatomiques présentes dans la tête, différenciées entre elles par leur intensité dans les IRMs anatomiques. Toutefois, il est fréquent que les IRMs anatomiques souffrent d'altérations conduisant à des problèmes de *bruits* et d'*inhomogénéités*, rendant la tâche de segmentation non triviale.

Plusieurs méthodes existent pour effectuer des segmentations de régions, et nous nous sommes principalement concentrés sur celles dérivées d'évolutions d'hypersurfaces: une hypersurface (courbe en 2D ou surface en 3D) est initialisée dans une image, et est progressivement déformée pour finalement épouser les frontières de la région à segmenter. Cette méthode permet d'intégrer des connaissances a-priori sur la forme de la structure à segmenter, en particulier des contraintes topologiques requises pour la segmentation du cortex.

ORGANISATION ET CONTRIBUTION DE CETTE THÈSE _____

Ce manuscrit est organisé en 4 chapitres, qui reflètent les différentes contributions de cette thèse, et 2 annexes présentant des détails supplémentaires:

Le **chapitre 3** couvre un état de l'art des méthodes de segmentation d'images. Après un bref aperçu de certains modèles discrets, ce chapitre se concentre sur les méthodes à base de propagation de front pour la segmentation, notamment ceux utilisant la méthode des ensembles de niveaux.

La méthode des ensembles de niveaux consiste à représenter le front comme étant le niveau zéro d'une fonction de dimension supérieure, en général choisie comme *fonction distance* au front. Cette méthode permet de gérer facilement les changements de formes du front, notamment ceux entraînant un changement de topologie.

L'évolution du front est ensuite exprimée à travers une équation aux dérivées par-

tielles, découlant de la minimization d'une fonctionnelle définie:

- sur l'hypersurface elle-même dans le cas des *modèles basés contours*. Ces modèles consistent à faire évoluer librement le front dans les régions homogènes, et à stopper son évolution au niveau des zones de contours dans l'image, correspondant en général à des zones de fort gradient.
- sur les régions délimitées par l'hypersurface dans le cas des *modèles basés régions*. Ces modèles consistent à faire évoluer le front de façon à séparer significativement les 2 régions qu'il délimite. Dans le cas d'une image scalaire, cette évolution revient à séparer les régions claires des régions sombres.

Le **chapitre 4** présente un nouveau schéma numérique pour la méthode des ensembles de niveaux à l'aide d'éléments finis quadrilatéraux. Similairement à l'approche de "la bande étroite", le calcul de la fonction distance est restreint aux éléments qui sont proches du niveau zéro, et peut donc être représentée par un ensemble connexe d'éléments finis. Pour garder une trace du niveau zéro au cours de son évolution, certains éléments sont insérés dans la bande, et d'autres retirés.

Cette implémentation a pour but d'améliorer la robustesse du modèle, en résolvant simultanément l'équation d'évolution et l'équation Eikonale grâce à un schéma en temps implicite. Toutefois, cette implémentation alourdit la charge de calcul, et augmente de façon non négligeable la complexité du modèle. Quelques détails concernant cette implémentation peuvent être trouvés en Annexe A.

Le **chapitre 5** présente un modèle de segmentation rapide basé sur des informations locales. Ce modèle combine des statistiques locales à l'intérieur et à l'extérieur du front pour le faire évoluer. Comme il est basé sur des décisions locales, ce modèle est plus robuste aux variations des régions d'intérêt (en intensité, contraste, bruit, ...). Ces statistiques locales sont exprimées à travers des convolutions Gaussiennes, ce qui permet une implémentation efficace grâce à des filtres récursifs. Ce modèle généralise plusieurs modèles état-de-l'art, grâce à un nouveau paramètre de voisinage dont le rôle est étudié. Ce modèle introduit toutefois de nouvelles difficultés qui sont inhérentes au fait de baser une propriété globale (la segmentation), sur de pure décisions locales. La seconde partie de ce chapitre explore certaines de ces difficultés et propose quelques corrections. Enfin, une implémentation efficace est proposée, en utilisant la puissance de calcul de dispositifs modernes hautement parallèles tels que les GPU (Graphic Processing Units).

Le **chapitre 6** porte sur la création de modèles anatomiques de la tête à partir d'IRMs bi-modales T1 et T2. Après avoir brièvement présenté les différentes structures anatomiques de la tête, un cadre est proposé pour créer automatiquement un modèle de tête en segmentant successivement les différentes structures anatomiques:

- **La peau** est segmentée en 2 phases: une segmentation grossière avec contrainte de régularité importante, puis plus fine une avec moins de régularité pour rentrer dans des détails d'importance.
- **Le liquide céphalo rachidien (LCR)** est ensuite segmenté, et constitue une enveloppe englobante du cerveau ("skull-stripping").
- **La matière blanche** est ensuite segmentée, en partant de la segmentation du LCR et en faisant rétracter le front sur lui même.
- Puis **la matière grise** est segmentée, en partant de la segmentation de la matière blanche et en faisant "gonfler" le front. Une contrainte topologique est proposée pour maintenir une topologie sphérique.
- Finalement **le crane** est finalement segmenté en partant d'une couche créée à partir de la segmentation du LCR, puis en faisant évoluer le front dans la zone comprise entre la peau et le LCR.

L'**Annexe A** présente quelques détails sur l'implémentation de la méthode des ensembles de niveaux grâce aux éléments finis, proposée dans le chapitre 4.

L'**Annexe B** présente la dérivée de la fonctionnelle introduite dans le chapitre 5. Cette dérivée est calculée grâce au *gradient de forme* [3], et inclue quelques termes supplémentaires qui augmentent la robustesse du modèle.

INTRODUCTION

CONTEXT

Neuroscience has always been a challenging and fascinating domain. It has gained in popularity in the last decades mainly due to recent advances in medical imaging and functional imaging. Early researches were based on anatomical and morphological studies of the brain, and were of no help for understanding neural activity. With the apparition of functional imaging devices, several methods have been proposed to study in real-time neural activity (PET-scan, fMRI, optical imaging, EEG, MEG, ...). Each of these methods is based on an underlying assumption of this activity:

- Positron Emission Tomography (PET-scan) and functional Magnetic Resonance Imaging (fMRI) are based on the fact that changes in blood flow and blood oxygenation in the brain (collectively known as hemodynamics) are closely linked to neural activity. PET-scan measures the blood flow inside the brain, by injecting radioactive tracers into the blood stream and detecting them using nuclear imaging. fMRI follows the same principle as MR images, but is more sensitive to oxygen and is thus able to detect large amounts of oxygen released by blood inside active neurons.
- ElectroEncephaloGraphy (EEG) MagnetoEncephaloGraphy (MEG) are based on the fact that active neurons produce some electrical activity, due to the chemical nature of synaptic transmissions. The EEG method measures the electric potential created by the induced electrical field, while the MEG measures the magnetic field created by this electrical activity [113].

Each methods has its own advantages and drawbacks, and are characterized by their *spatial resolutions* and *temporal resolutions*. The methods providing the best temporal resolution are without a doubt those based on electrical activity measurement (MEG/EEG).

NEED FOR ANATOMICAL MODEL OF THE HEAD ---

However, spatial resolution in the MEG/EEG methods strongly depends on the accuracy of the *inverse problem reconstruction*, which consists in computing locations of electrical sources inside the brain from the set of measurements on the scalp. This inverse problem is performed using the *Maxwell's equations*, that describe the properties of the electric and magnetic fields and relate them to their electrical sources. However, it is stated in these equations that both fields are highly sensitive to the *conductivity* of areas they cross (especially for EEG).

The human head is very complex and includes various types of anatomical structure that conduct differently electricity. Therefore, in order to accurately perform the inverse problem, one has to know precisely the different anatomical structures present in the head. Differentiating between these structures consist in creating a *model of the head*, which partition the head according to the local conductivity of these structures. In general, these models include 5 structures, that are nested inside each others in the manner of "Russian nested dolls" (Fig. 2.1). They include, from outer to inner structures : the skin, the skull, the CerebroSpinal Fluid (CSF), the Grey Matter (GM), and the White Matter (WM) [8].

The most important structure in these models is without a doubt the grey matter, that constitutes the *cerebral cortex*. Anatomically, the cortex is a highly convoluted layer of grey matter, with spherical topology. It includes all the neurons of the brain, the white matter ("inside" of the brain) including only the nervous fiber linking these neurons to each others. Neural activity is detected inside the cortex, and accuracy of its model is therefore crucial.

Several methods exist to perform these kind of models, that have been widely studied in the *image segmentation* domain.

IMAGE SEGMENTATION ---

Image segmentation consists in partitioning an image into "meaningful areas". These areas correspond in general to objects to be extracted from images, and can be seen as regions separated by distinct boundaries or regions characterized by some global properties. In our case, these areas correspond to different anatomical structures present in the head, differentiated according to their intensities. However, several alterations coming from the imaging device can perturb the MRI, leading in most of the case to *noise* and *intensity drifts*. These alterations make the segmentation non trivial, and models robust to these alterations are required

Several models exist to perform image segmentation, but we focused on those based on hypersurface evolutions: an initial curve or surface is initialized in the image, and is incrementally adjusted to finally fit the border of the region of interest.

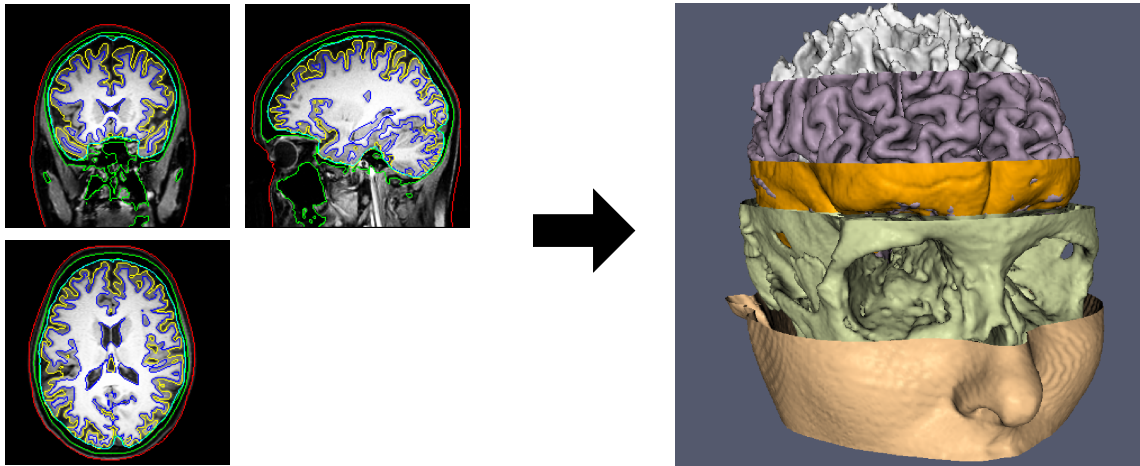


Figure 2.1: Creation of anatomical head model from MRIs.

This method allows to incorporate some knowledge about the shape of the anatomical structure, such as topological constraints in order to segment the cortex.

ORGANIZATION AND CONTRIBUTIONS OF THIS THESIS _____

This manuscript is organized in four chapters, which reflect the different contributions of this thesis, and two appendices presenting some additional details:

Chapter 3 covers state-of-the-art methods for image segmentation. After briefly reviewing some discrete models such as Markov random field and graph cuts, this chapter focuses on front propagation methods for image segmentation, especially on those using the *level-set* method. These models are then classified in two separate categories : *edge-based* segmentation models, and *region-based* segmentation models.

Chapter 4 presents a novel implementation of the level-set method using the finite element method with quadrilateral elements. Similarly to the narrow band approach, the computation of the distance function is restricted to the elements that are close to the zero level-set, and thus can be represented as a sparse set of connected elements. To keep track of the zero level-set during its evolution, some elements are inserted to the set, and others removed from the set depending on the motion of the curve.

Chapter 5 presents a fast and robust segmentation model based on local informations. It combines local statistics of the regions along the contour (inside and outside) to drive the segmentation procedure. Since it is based on local decisions, this model is more robust to local variations of the regions of interest (contrast, noise, blur, ...). In order to avoid the computational burden of a direct estimation, these

local statistics are expressed as the result of Gaussian convolutions. This makes an efficient implementation via recursive filters possible, and gives a complexity of the same order as methods based on global statistics. This model generalizes several state-of-the-art segmentation models, using a neighborhood parameter whose role is investigated. This model also introduces some new difficulties which are inherent to the fact of basing a global property (the segmentation) on pure local decisions. The second part of this chapter explores some of those difficulties and propose some possible corrections. Finally, an efficient implementation is proposed, using the computational power of highly parallel devices such as modern GPUs (Graphic Processing Units).

Chapter 6 presents a method for the creation of anatomical head models from bi-modal T1- and T2-weighted MRIs. After briefly reviewing the anatomical structures present in the head, a framework is proposed to segment successively the different anatomical structures included in the model by using the local-statistics model introduced in the previous chapter.

- **The scalp** is first segmented by starting from a bounding box of the MRI.
- **The CerebroSpinal Fluid (CSF)** is then segmented by initializing the front as a sphere inside the brain.
- **The White Matter (WM)** is then extracted by starting from the previously segmented CSF.
- **The Grey Matter (GM)** is then extracted by starting from the previously segmented WM.
- **The skull** is finally segmented by initializing the front as a layer between the CSF and the scalp.

Appendix A presents some details about the finite-element implementation of the level-set method proposed in chapter 4.

Appendix B presents the derivative of the local-statistics functional introduced in chapter 5. This derivative is performed using *the shape gradient method* [3], and include some additional terms that increase the robustness of the model.

IMAGE SEGMENTATION

Contents

Overview	22
3.1 Introduction	22
3.2 Discrete Models: Markov Random Fields and Graph Cuts . . .	24
3.2.1 Markov Random Field	24
3.2.2 Graph Cuts	25
3.2.3 Random Walker	25
3.3 Continuous Models: Snakes and Active Contours	26
3.3.1 The Snake Model	26
3.4 Front Representation: Explicit models vs Implicit models . .	27
3.4.1 Explicit Representation	27
3.4.2 Implicit Representation: The Level-Set Method	28
3.5 Boundary Based Segmentation Models	31
3.5.1 Geodesic Active contour	32
3.5.2 Gradient Vector Flow	34
3.6 Region based segmentation models	35
3.6.1 Functional-based models	37
3.6.2 Bayesian-based models	39
3.7 Conclusion	43

OVERVIEW

This chapter covers state-of-the-art methods for image segmentation. After explaining the purpose of segmentation, we briefly review some famous discrete models for image segmentation such as *Markov random field* and *graph cuts*. We then focus on front propagation methods for image segmentation, and particularly on those using the *level-set* representation. Finally, we give a detailed insight on segmentation models using the level-set method, by classifying them in two separate categories : *edge-based* segmentation models, and *region-based* segmentation models.

Keywords: Graph-cuts, Snakes, Level-set, Mumford-Shah, Chan-Vese

3.1 INTRODUCTION

Image segmentation is a domain that has been widely studied in the last twenty years, and still remains a challenging task [70, 71]. It consists in partitioning an image into “meaningful areas”. These areas correspond in general to objects to be extracted from images, and can be seen as regions separated by distinct boundaries or regions characterized by some global properties.

Early approaches in image segmentation were based on simple heuristics, and consisted in several different steps [80]. Recent approaches are now mainly based on **optimization methods**, and consist in minimizing a given energy to obtain the desired partition. These energies differ depending on the type of segmentation desired, and are in general formulated as a weighted sum of different terms. Most of the times, these energy include two terms : a *data attachment term*, standing for how the image will be partitioned depending on its properties (intensity, color, texture ...), and a *partition term*, defining some prior knowledge about the shape of the desired partitions (smoothness, topology, global form...). These optimization-based approaches can be classified in two distinct categories : spatially discrete and spatially continuous representations.

Spatially discrete optimization methods rely usually on expressing images as graphs, where the pixels are seen as nodes and adjacent pixels are linked by weighted edges (depending on a given connectivity). These weights denote in general the likelihood of the two linked pixels to belong to the same region. The segmentation is then performed by minimizing a discrete energy defined on the edges of the graph, by using some well-known algorithms from the graph theory domain. For example, one of the most famous discrete optimization method is the *graph cuts* method, first proposed by Greig et al. [40] and recently popularized by Boykov et al. [11]: the segmentation is seen as a *cut* in the graph which has minimal cost.

Spatially continuous optimization methods rely on evolving closed curves or closed surfaces in images, to finally fit the border of the regions to extract.

Curve evolution for image segmentation was first proposed in the seminal work of Kaas et al. [52], often referred as *Snakes* : the evolution is performed by minimizing a continuous energy defined along the contour. This technique was a breakthrough in early development of segmentation models, and have been rapidly extended to overcome some of its important limitations.

These evolving contours were initially represented as **explicit models**, which means they were parameterized by a set of points along the contour, and therefore seen as polygons. Although this representation is simple and efficient, it also brings some important drawbacks, especially concerning the re-parameterization process.

An important step has been reached by using **implicit models** to represent these contours. This representation, also called *level-set* representation, was first introduced by Dervieux and Thomasset [30] and later re-discovered and largely popularized by Osher and Sethian [78, 94]. It consists in expressing the contour as the zero level of a higher dimensional function, most of the time chosen as the *distance function* to the contour. While being computationally more expensive, this representation overcomes some important limitations of explicit models, mostly due to the absence parameterization of the contour.

Several energies have been proposed to make these contours evolve, and can be globally classified into two categories : *edge-based* models, and *region-based* models.

Edge-based models consist in evolving the contour in homogeneous areas, and locally stopping it when it reaches high image gradients [16, 53, 114]. Among these models, the geodesic active contour model [16] is certainly the most elegant one: the segmentation problem is reformulated as finding cyclic geodesics (paths of minimal length) in a Riemanian space whose metric is inversely proportional to the norm of the image gradient.

Region-based models consist in evolving the contour in order to separate significantly the statistics of the regions it delimits [19, 108, 107, 79, 26]. For example, separating bright areas from dark ones, or extracting areas matching a certain histogram [55]. These models take into consideration the whole region and are thus more robust to noise and to initialization than edge-based models.

In the following, we give a detailed insight of all the preceding notions. After briefly reviewing discrete approaches in image segmentation, we will present in detail techniques based on front evolution for image segmentation. We will particularly focus on the level-set method, and how it can be efficiently implemented. Finally, we will review several geometric flows using the level-set method, categorized into two different sets: regions-based and edge-based models.

3.2 DISCRETE MODELS: MARKOV RANDOM FIELDS AND GRAPH CUTS

Although discrete optimization models for image segmentation are not really the scope of this thesis, they have become extremely popular in the last few years, mostly due to the popularization of the *graph cuts* optimization method [10]. Despite some inconvenients, these discrete models have proved to be very powerful in particular cases, and are indeed a true alternative to curve evolution based segmentation methods.

3.2.1 Markov Random Field

Markov Random Field (MRF) based image restoration was first introduced by Geman and Geman [33], and deals elegantly with site interactions. It is based on viewing pixels as *sites* $\mathcal{S} = \{s_1, \dots, s_N\}$, and specifying a *neighborhood system* $\mathcal{N} = \{\mathcal{N}_s \in \mathcal{S}\}$ between these sites.

- A *random variable* X_i is associated to each site s_i , and the set of all these random variable is a **Markov Random Field**, defined by the *Markov property*:

$$P(X_i = x_i | X_j = x_j, i \neq j) = P(X_i = x_i | X_j = x_j, j \in \mathcal{N}_i),$$

with x_i and x_j realizations of the random variables X_i and X_j . This means that the solution at a site i only depends on its neighborhood \mathcal{N}_i .

- A *clique* C is defined as a set of mutual neighbor sites, and $\mathcal{C} = \{C_1, \dots, C_M\}$ is the set of all possible cliques in the field. A *potential function* $V(C, x)$ is then defined, associating to each cliques $C \in \mathcal{C}$ a potential according to the realization $x = \{x_1, \dots, x_N\}$ of the MRF. This potential specifies interactions between sites inside a clique, and defines the problem to solve.
- The *global energy* $U(x)$ is expressed as the sum of the potentials of all cliques:

$$U(x) = \sum_{C \in \mathcal{C}} V(C, x) \longrightarrow \min,$$

and the **Gibbs distribution** of the MRF is defined as:

$$P(X = x) = Z^{-1} \exp(-U(x)) \longrightarrow \max,$$

where Z^{-1} is a normalization term.

In summary, besides mathematical definitions, the main idea is to formulate a given problem in term of local interaction between adjacent pixels. If the cliques are pairwise, the image can then be seen as a weighted graph: pixels are seen as nodes, and adjacent pixels are linked by an edge whose weight depends on the local interaction between them. These interactions define the problem, and the energy to minimize is defined as the sum of these weights.

3.2.2 Graph Cuts

The “*graph cuts*” method is a combinatorial optimization method whose purpose is to minimize a discrete energy defined on a graph. It has been introduced by Greig et al. [40] for the bi-partitioning segmentation case, and has been somehow forgotten for 10 years, to finally become popular in 1999 [10]. Since then, it has been successfully applied to several computer vision problems such as restoration, segmentation, . . .

The main idea is to consider the image as a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where the pixels are seen as a set of nodes \mathcal{V} , and adjacent pixels are linked by a set of weighted edges \mathcal{E} . Two special nodes are also present : a *source* node and a *sink* node, that are in general linked to all pixels of the image.

A **cut** $C \subset \mathcal{E}$ is a set of edges removed from the graph, such that there is no more paths connecting the source node to the sink node. A minimal cut is a cut whose cost (*sum of all its edges*) is minimal. It is then proved that a discrete energy defined on a graph (*ie. sum of all the edges of the graph*) can be minimized by finding a **minimal cut**, which is performed by **maximizing a flow** from the source to the sink (*duality min-cut / max-flow*) using some well known algorithms from the graph theory domain [32, 34].

One of the main advantages of this method is that under certain conditions, a solution corresponding to the global minimum of the energy can be found, which outperforms usual methods finding only local minimas (this may however be an advantage in segmentation). Unfortunately, only a few kind of energies can be minimized this way [10, 56], which restricts the range of application of this method.

For example in the case of segmentation, the two subsets of pixels separated by the cut correspond to the two regions for the bi-partitioning segmentation. Various types of segmentation models have been proposed using this approach [11, 49, 12, 9], and this is indeed a true alternative to active contour based segmentation models.

3.2.3 Random Walker

The random-walker method [38] is another quite popular segmentation method based on graph representation. It consists in labelling manually a set of pixels, and automatically determining the probability that a random walker starting at each unlabelled pixel will first reach one of the prelabelled pixels. Then, each unlabelled pixel is assigned the label for which the greatest probability is calculated. These probabilities are not computed by simulating a finite set of random walks, but are instead computed by solving a *dirichlet problem*. This require solving a linear system with graph Laplacian matrix, that can be performed using an efficient GPU implementation allowing interactivity and real time segmentation [39]. The random walk processes are performed in term of transition probabilities to move from one pixel to another: for a given unlabelled pixel, they consider (infinitely many) random walkers starting from this pixel and all moving according to these transition probabilities.

3.3 CONTINUOUS MODELS: SNAKES AND ACTIVE CONTOURS

In opposition to discrete optimization models, continuous optimization models consider images as continuous functions, and rely in evolving curves or surfaces¹ in images, to finally fit the border of the regions to extract. Evolution of these fronts is done by minimizing a continuous energy defined:

- along the front for **boundary based models**.
- inside the regions the front delimit for **regions based models**.

This method was first introduced by Kaas, Witkins and Terzopoulos with the well-known *snake* model [52], which belong to the boundary-based category.

3.3.1 The Snake Model

Let $I : \Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}^+$ be a scalar image. The *Snake* model is based on evolving a parameterized 2D contour $\Gamma : [0, 1] \rightarrow \Omega$ by minimizing the following functional:

$$E(\Gamma) = \underbrace{\int_0^1 \alpha |\Gamma'(p)|^2 dp + \int_0^1 \beta |\Gamma''(p)|^2 dp}_{\text{internal terms}} - \lambda \underbrace{\int_0^1 |\nabla I(\Gamma(p))| dp}_{\text{external term}}, \quad (3.1)$$

where α , β and λ are parameters balancing between the different terms.

This functional is composed of different terms, each of them contributing to the evolution of the contour. As most of the segmentation models, these terms are classified in two categories :

- The first two terms are regularization terms corresponding to **internals properties** of the contour, and contribute to the desired shape of the segmentation.
- The last term is an **external property** standing for a data-attachment term, and depends on image characteristics.

This energy is intended to be minimum when the contour is smooth enough (internal terms) and overlaps image boundaries corresponding to high image gradient (external term). This model brought several advantages over existing ones, but was quite limited in several aspects and numerous extensions have been proposed.

One of the main limitations is that this model was only restricted to 2D parameterized contours (curve). This type of contour, referred as **explicit** contour, suffers from several inconvenients that are discussed in the next section.

¹In the next sections, we will use the notion of “front” or “interface”.

3.4 FRONT REPRESENTATION: EXPLICIT MODELS VS IMPLICIT MODELS

Before considering the functional defining the segmentation model, one has to choose a type of front representation. Several representations exist in the literature, and each of them has been used in various domains (fluid dynamics, computational geometry, etc. . .). We can globally classify these representations in two separate categories: **explicit** representations and **implicit** representations.

3.4.1 Explicit Representation

These evolving fronts were initially represented *explicitly*, which means they were parameterized by a set of values along the front (Fig. 3.1). This representation allowed simple and efficient implementation, but suffered from several inconvenients:

- Topology changes were not allowed due to the parameterized nature of the front. Several heuristics have been proposed to overcome this issue [67, 68, 69].
- A re-parameterization was needed during the contour evolution, that was likely to perturb position of the contour. To solve this issue, different types of basis functions have been proposed in the literature [121, 84, 5, 27].
- **Generalization to higher dimensions (3D and more) was tedious due to the parameterization.**

However, explicit models are often favored when efficiency is preferred to robustness and theoretical correctness [69], or when *open-contours* are needed. Recent models based on Delaunay triangulation have also been proposed [83] and are quite promising as they benefit from the robustness of the Delaunay triangulation.

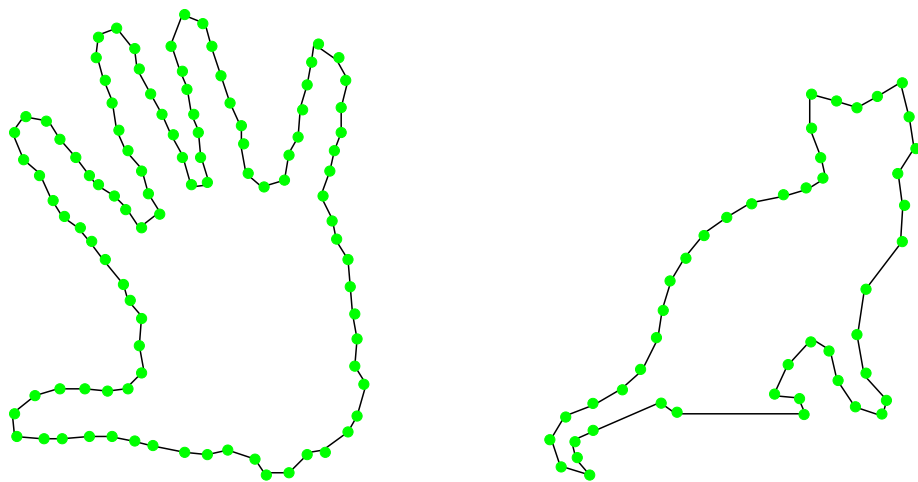


Figure 3.1: Explicit representations of 2D contours.

3.4.2 Implicit Representation: The Level-Set Method

In order to overcome the limitations of explicit models, a new way to represent evolving fronts has been introduced with **the level-sets method**. This method was introduced by Osher and Sethian [78, 94] in 1988, although the same idea was previously used 10 years earlier by Dervieux and Thomasset [30] for fluid dynamics purpose.

This method consists in expressing the interface between two regions as the zero level of a higher dimensional function Φ , defined on the whole image domain. This implicit function is required to be a Lipschitz function, and most of the time is chosen as the *signed distance function* to the front, such as:

$$\Phi(x) = \begin{cases} -\mathcal{D}(x, \Gamma) & x \in \Gamma_{in} \\ 0 & x \in \Gamma \\ +\mathcal{D}(x, \Gamma) & x \in \Gamma_{out} \end{cases}$$

where Γ represents the interface (front) separating two regions Γ_{in} and Γ_{out} , and $\mathcal{D}(x, \Gamma)$ the Euclidean distance from the point x to the contour Γ (Fig. 3.2).

Through this representation, geometric properties of the front can easily be expressed as differential terms defined on the level-set function:

- The unit normal vector to the front is $\vec{n} = \frac{\nabla\Phi}{|\nabla\Phi|}$.
- The mean curvature of the front is $\kappa = \nabla \left(\frac{\nabla\Phi}{|\nabla\Phi|} \right)$.

These properties allow to express front evolution as a time-dependent partial differential equation on the level-set function: consider a front Γ evolving under a speed F in its normal direction \vec{n} with no tangential velocity, so that $\frac{\partial\Gamma}{\partial t} = F\vec{n}$.

As the front Γ corresponds to the zero level-set of Φ , we have,

$$\Phi(\Gamma(t), t) = 0 \quad \forall t \quad ,$$

by applying the chain rule, we get

$$\frac{\partial}{\partial t}\Phi(\Gamma(t), t) = \frac{\partial\Phi}{\partial t} + \nabla\Phi \frac{\partial\Gamma}{\partial t} = 0 \quad .$$

In level-sets notation, the evolution equation can be rewritten as,

$$\frac{\partial\Gamma}{\partial t} = F\vec{n} \quad \iff \quad \frac{\partial\Gamma}{\partial t} = F \frac{\nabla\Phi}{|\nabla\Phi|} \quad .$$

Finally, putting all the terms together, we get the evolution equation:

$$\boxed{\frac{\partial\Phi}{\partial t} + F|\nabla\Phi| = 0} \quad . \tag{3.2}$$

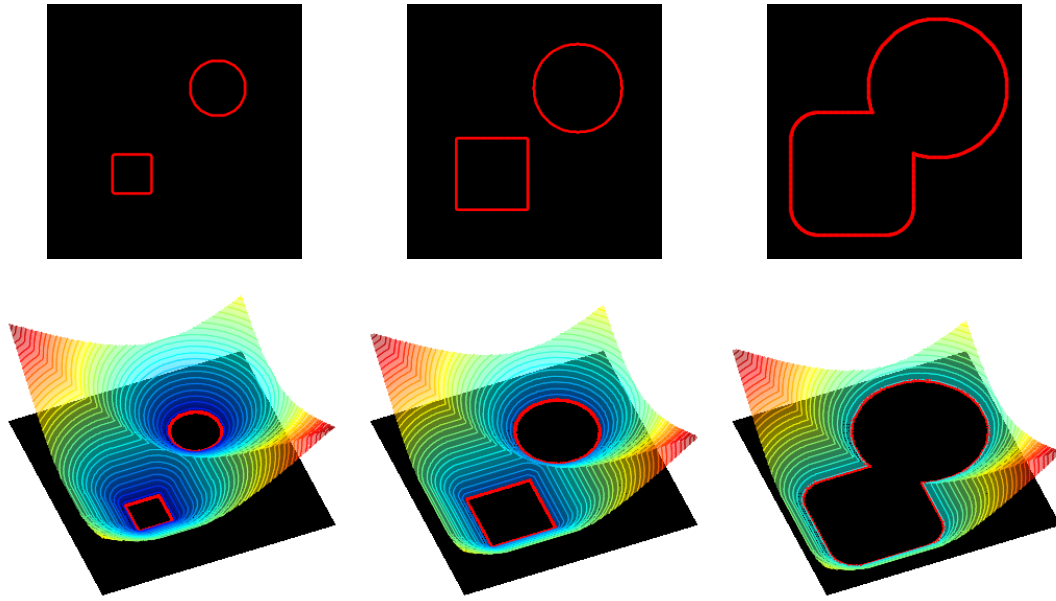


Figure 3.2: Level-sets: Topological changes are automatically handled.

In summary, **the evolving front is embedded in a higher dimensional function, and its evolution is done through the evolution of this function using a time-dependent partial differential equation.**

Since it is not necessary anymore to worry about the parameterization, the level-sets framework has naturally become the standard way of expressing contour & surface evolution techniques [77].

It brings a lot of advantages compared to explicit models:

- **Generalization to higher dimensions** is easy and natural.
- **Topological changes** are automatically handled, since the front is the zero level-set of the level-set function (Fig. 3.2).
- **Theoretical correctness** based on viscosity solution have been widely studied, providing a good and robust mathematical basis [25, 93].
- **Stable and efficient numerical schemes** have been proposed to solve the PDE.
- **Properties of the curve** such as normal vectors, or curvature can be easily and accurately expressed through differential terms.

One of the main inconvenient is that this framework is computationally more expensive than explicit models. Indeed, the evolution of the front is done through the evolution of the level-set function, which is defined on the whole image domain. This requires to compute the solution in the whole image domain, with a computational

cost that not depend on the actual size of the front. Several methods have However been proposed in the literature in order to improve its efficiency.

Narrow Band Implementation

Since the front corresponds the zero level-set of the distance function, computing the solution of the PDE (Eq. (3.2)) far from the zero level-set does not contribute to the evolution of the front. In order to decrease the complexity of the method, an alternative is to compute the solution of the PDE only at the neighborhood of the zero level-set, in a “band” thick enough to keep track of the front; this is known as *the Narrow Band Approach*. This approach decreases dramatically the complexity, and allows computation times close to those obtained using explicit models.

Numerical Schemes

Several numerical scheme have been proposed to solve Eq. (3.2), and the most common choice is to compute spatial derivatives using a **finite differences ENO scheme**, and time derivative with an **explicit scheme**. This choice is driven by the fact that finite differences scheme are quite efficient, and combined with the narrow band approach speeds up considerably the level-set method. However finite difference schemes are nonetheless limited in accuracy, and some problems of stability may arise if CFL conditions² are not satisfied. The chapter 4 investigates a new way to implement the level-set method, using the finite elements method.

Reinitialization

One of the limitations of the level-set method is that the implicit function has to remain a distance function during its evolution, which means it must satisfy a viscosity solution of the Eikonal equation. In general, this is not the case, and the slope of the function may become steep or flat, depending on the speed chosen. The easier way to deal with this limitation is to periodically “reinitialize” the function to a distance function again, without moving its zero level-set (which is not very different from the reinitialization of explicit models). Several methods have been proposed to do so:

- a reinitialization PDE[103] have been proposed to “re-slope” the implicit function, without moving its zero level set:

$$\frac{\partial \Phi}{\partial t} = \text{sign}(\Phi) (1 - |\nabla \Phi|),$$

the complexity of this reinitialization depends on the number of iterations to converge, and is in general relatively expensive.

²Courant-Friedrichs-Lewy conditions: conditions linking the time step to the spatial step

- a re-distancing algorithm, the **fast marching method** [93], has been proposed to recompute a new distance function from the previous one. It is based on initializing seed points on the zero level-set, and create a new distance map using some incremental updates of an evolving front. This algorithm is quite famous in the computer vision community, as it is an easy and efficient way to compute a distance map in Euclidian or Riemanian spaces, and for example to track geodesics of interest. The complexity of this algorithm is in $O(n \log n)$, where n corresponds to the number of points of the grid. However, some algorithms have been proposed to reach a linear complexity in $O(n)$, by using specialized data structures [115]. Let us note that the same idea has been used in the **fast sweeping method** [50], which allows a parallelizable scheme.
- The work of Gomes and Faugeras [37] aims at preserving the distance function during its evolution. It is based on computing the solution of the PDE at the points lying on the the zero level-set, and propagating this speed in the whole function thanks to an advection PDE. This method is however expensive, and a re-distancing method is often favored.

In practice, a mix of the first two method is commonly used: a reinitialization PDE is applied on the points very close to the zero level-set, and a re-distancing with a fast marching is performed to recreate a complete narrow band.

*In this section, we briefly reviewed methods to represent evolving fronts by first introducing explicit models, and then implicit models through the level-set method. In the next sections, we will focus on segmentation models that will drive the front, and characterized by a functional to minimize. These models can be classified in two distinct categories : **Boundary-based** models and **Region-based** models.*

3.5 BOUNDARY BASED SEGMENTATION MODELS

The main idea behind boundary-based segmentation models is:

1. choose a “boundary model” to characterize boundaries in images.
2. move freely the front in image, and locally stop it when it reaches a boundary.

In general, boundaries are chosen as high image gradient, but several models has been proposed in the literature.

Boundary-based models can be seen as simple “closed edge detectors”, and are quite effective on simple objects whose boundaries are well defined. However, these models are “myopic” since they only provide a local criterion to stop the evolution, and do not look at the whole image structure.

The first model of this category was introduced by the *snake* model, where boundaries were seen as high image gradient. Since then, several extensions have been proposed and the most famous one is without a doubt the **geodesic active contour**.

3.5.1 Geodesic Active contour

The **geodesic active contour** model was introduced in [17] and [53, 54], and can be seen as an extension of the classical snake model. Rather than expressing the segmentation functional as a sum of regularization terms and data attachment terms, the authors introduced a functional composed of one single term, as follows:

$$E(\Gamma) = \int_0^L g(|\nabla I(\Gamma(s))|) ds = \int_0^1 g(|\nabla I(\Gamma(p))|) \left| \frac{\partial \Gamma}{\partial p}(p) \right| dp \quad , \quad (3.3)$$

where ds is the arc-length element, and L the Euclidian length of the front Γ .

This functional represent **the length of the active contour in a Riemanian space**, and minimizing this functional implies finding a contour of minimal length, corresponding to a **Geodesic**³. The metric of this space is induced by image characteristics, and has to be strong in homogeneous areas and weak on image boundaries. The function $g(|\nabla I|)$ define the metric, and is a monotonically decreasing function defined on $\mathbb{R}^+ \rightarrow \mathbb{R}^+$, such that $g(0) = 1$ and $\lim_{x \rightarrow +\infty} g(x) = 0$.

A very common choice is given by:

$$g(|\nabla I|) = \frac{1}{1 + |\nabla I|^r} \quad , \quad r = 1 \text{ or } 2 \quad . \quad (3.4)$$

Derivative of this functional is done through *Euler-Lagrange* equations, and minimization of this functional is performed thanks to a steepest gradient descent. This gives us the following contour evolution :

$$\frac{\partial \Gamma}{\partial t} = g\kappa\mathcal{N} - (\nabla g \cdot \mathcal{N})\mathcal{N} \quad , \quad (3.5)$$

where \mathcal{N} denotes the inward normal vector of the front, κ the curvature of the front, and $g = g(|\nabla I|)$ the metric of the Riemanian space.

The functional (3.3) does not depends on a parameterization any more, and can thus be applied with implicit models, which was the principal limitation of the initial snake model. Moreover, this allows to easily extend the model to higher dimensions. The contour evolution can thus be rewritten in level-set notations:

$$\boxed{\frac{\partial \Phi}{\partial t} = g |\nabla \Phi| \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) + \langle \nabla g, \nabla \Phi \rangle} \quad , \quad (3.6)$$

where Φ represent the level-set function.

³Path of minimal length in a Riemanian space.

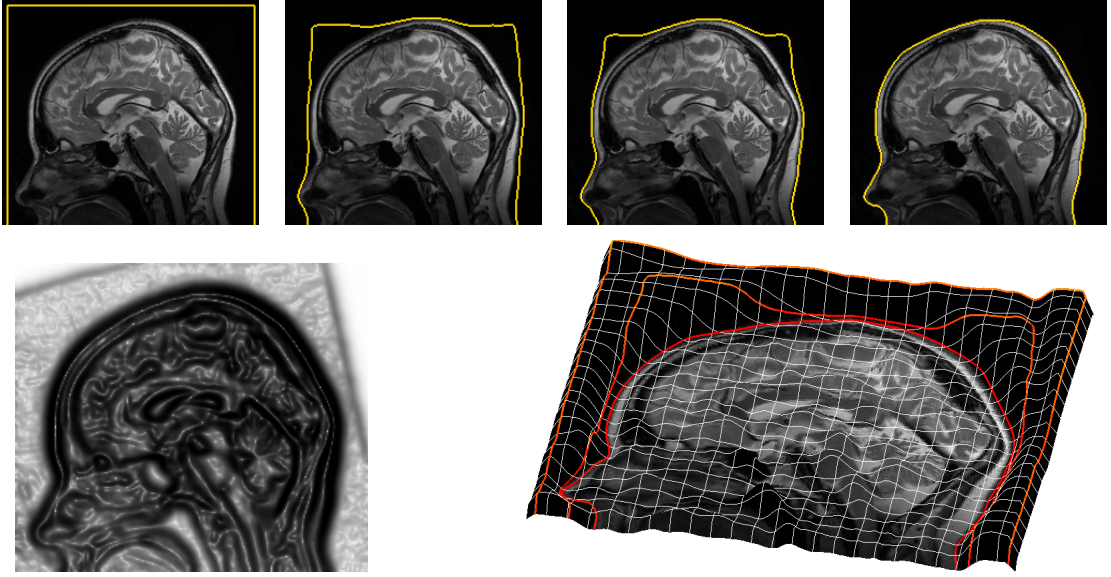


Figure 3.3: Top row: Evolution of the contour with the geodesic active contour model; Bottom row: Metric of the Riemannian space Eq. (3.4), and representation of the image and the contours in that space (the contour remains stuck into some kind of “saddle”).

This evolution is composed of two terms having a different incidence in the evolution of the contour:

- the first term corresponds to a mean curvature motion of the front, that locally stops when $g \rightarrow 0$, ie. when the contour reaches high image gradient. It can be seen as a mix between a regularization term (mean curvature motion) and a data attachment term (stop on image gradient). In order to speed up the convergence, a *balloon pressure force* [23] can be incorporated into this term, such that it becomes $g |\nabla\Phi| \operatorname{div} \left(\frac{\nabla\Phi}{|\nabla\Phi|} + \nu \right)$, where ν is whether positive (contour shrinks) or negative (contour inflates).
- the second term is a less classical term, and plays the role of an *advection term* that speeds up the evolution of the front when it is close to an image boundary. Moreover, it is also a refinement term that pushes back the contour on the boundary if it has slightly missed it. Imagine it is a force that make the contour “fall” into the “saddle” created by the metric of the Riemannian space (Fig. 3.3).

Note that previous boundary models were proposed in [15, 22, 65], but were not based on an optimization framework. The authors proposed an evolution equation for implicit functions that was similar to the first term of Eq. (3.6). The geodesic active contour model is a more interesting model since it is based on an optimization framework, elegantly incorporate a Riemannian space in the formulation, and its derivation introduces a new term that reinforces the robustness of the model.

3.5.2 Gradient Vector Flow

Other boundary-based models have been introduced, but most of them are relatively similar to the geodesic active contour model (GAC), providing only some slight extensions. However, the **Gradient Vector Flow** [114] model introduces a new approach, and overcomes some limitations of the GAC model. It is based on successive steps, defined as follows:

- First, an edge map $f(x, y)$ is computed similarly to the GAC model, by applying a function $g(\cdot)$ on the image gradient, except that it has to be large near image boundaries, and low in homogeneous areas (Fig. 3.4, top row).
- A vector field $\mathcal{V}(x, y) = (u(x, y), v(x, y))$ is then obtained from the edge map $f(x, y)$, by minimizing the following functional:

$$E(\mathcal{V}) = \int \int \mu (|\nabla v|^2 + |\nabla u|^2) + |\nabla f|^2 |\mathcal{V} - \nabla f| \, dx \, dy \quad .$$

This vector field has to be smooth when there is no image boundary, and similar to ∇f close to image boundary. Its purpose is to extend the gradient of the edge map in the whole image (Fig. 3.4, top row).

- Finally, an implicit contour is initialized in the image and is driven by the vector flow $\mathcal{V}(x, y)$ using the following evolution equation:

$$\boxed{\frac{\partial \Phi}{\partial t} = - \langle \nabla \mathcal{V}, \nabla \Phi \rangle} \quad .$$

This equation is an advection equation similar to the second term in the derivative of the GAC model. However, the front will be attracted from a further distance due to the smoothness of the field $\mathcal{V}(x, y)$ (Fig. 3.4, bottom row).

This model is interesting as it is more robust to initialization than the GAC model. However, it is computationally more expensive since the vector fields computation requires a lot of iterations in order to correctly “extend” the edge-map gradient to the whole image.

*In this section, we briefly reviewed boundary-based segmentation models, by first introducing the **Geodesic Active Contour Model**, and then the **Gradient Vector Flow** model. These models are quite effective in simple cases, but become quickly limited as soon as image boundaries are not well defined, when it need to be looked at a more global scope. In order to overcome these difficulties, **region-based** segmentation models have been introduced, and are the scope of the next section.*

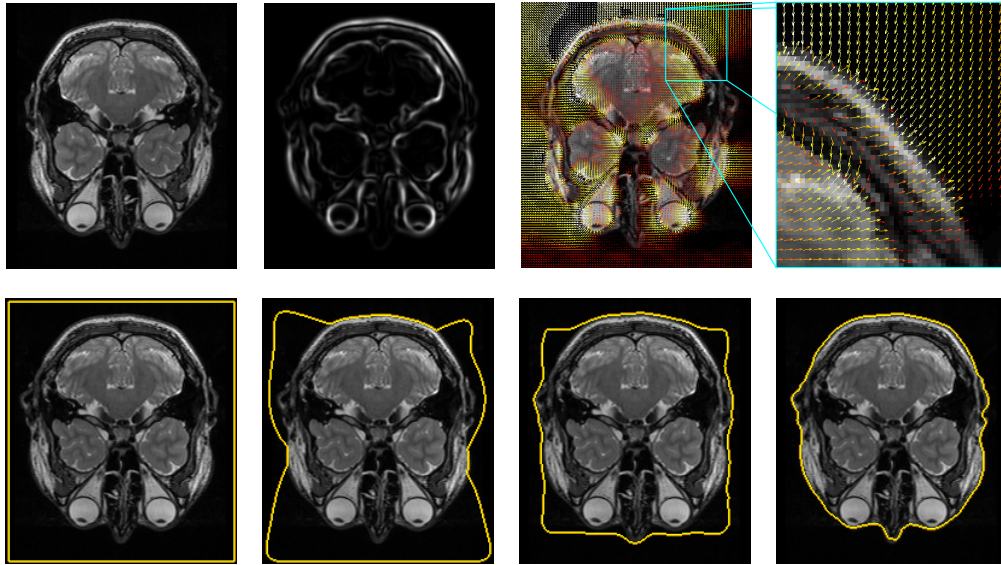


Figure 3.4: Top row: Initial image; Edge map; Gradient vector flow & zoom; Bottom row: Evolution of the contour with the gradient vector flow model.

3.6 REGION BASED SEGMENTATION MODELS _____

Although boundary-based segmentation models are quite powerful to segment simple objects, they suffer from two important limitations:

- These models are based on very local information, and are thus very sensitive to local minima. This is especially a problem when images are noisy: the active contour remains stuck in areas where noise is too important.
- Initialization of the front has to be finely tuned in order to correctly segment the desired object. In most of the cases, the initial contour has to be either completely inside the object, or outside around the object. This is a major drawback especially in the case of segmenting complex 3D structures in medical images.

To overcome these difficulties, **region-based models** have been introduced. The main idea of these models is to take the whole image into account, by expressing the functionals as integrals over the different regions delimited by the front. In the bi-partitioning case of two regions Ω_{in} and Ω_{out} , these functionals will look like:

$$E(\Gamma) = \int_{\Omega_{in}} f_{in}(x) dx + \int_{\Omega_{out}} f_{out}(x) dx \quad , \quad (3.7)$$

where Ω_{in} and Ω_{out} are two regions partitioning the image domain Ω , such that $\Omega_{in} \cup \Omega_{out} = \Omega$ and $\Omega_{in} \cap \Omega_{out} = \emptyset$. Γ is the front separating the two regions, and f_{in} and f_{out} are two “penalty functions” defining a pixel-wise energy for both regions Ω_{in} and Ω_{out} .

If the model has to be applied with implicit representations, a common choice is to reformulate the energy in terms of level-set notations, and to derive it in order to obtain a level-set evolution, using a gradient descent:

$$E(\Gamma) \longrightarrow E(\Phi) \quad \Longrightarrow \quad \frac{\partial \Phi}{\partial t} = - \frac{\partial E(\Phi)}{\partial \Phi} \quad ,$$

where Φ is the implicit representation of the front Γ . This reformulation can be done very easily in the bi-partitioning case, by assigning a “positive” and a “negative” region, using a *regularized heavyside* on the level-set function to differentiate between the regions:

$$E(\Phi) = \int_{\Omega} H_{\alpha}(\Phi(x)) f_{in}(x) + (1 - H_{\alpha}(\Phi(x))) f_{out}(x) dx \quad .$$

The function H_{α} is a regularized heavyside function defined by:

$$H_{\alpha}(x) = \begin{cases} 0 & x < -\alpha \\ \frac{1}{2} \left(1 + \frac{x}{\alpha} + \frac{1}{\pi} \sin \left(\frac{\pi x}{\alpha} \right) \right) & |x| < \alpha \\ 1 & x > \alpha \end{cases}$$

where α is the amount of regularization.

The functional (3.7) can be extended in the case of several regions $\Omega_1, \dots, \Omega_n$ (multi-phase segmentation), by expressing it as a sum of integral across each region with a corresponding penalty function, such as:

$$\begin{aligned} \Omega &= \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_n, & \Omega_i \cap \Omega_j &= \emptyset, \quad \forall i \neq j \\ & \Downarrow \\ E &= \int_{\Omega_1} f_1(x) dx + \int_{\Omega_2} f_2(x) dx + \dots + \int_{\Omega_n} f_n(x) dx \end{aligned}$$

However the level-set formulation is more challenging, since it requires several level-set functions to define these regions. A direct approach is to associate a level-set function to each region, by using only their “positive areas”. Unfortunately, this approach clearly increase the computational complexity, and some heuristics may be needed to forbid overlapping between the region, and suppress the vacuum between them [119].

Another approach introduced by Chan and Vese [19] is to use the overlapping between level-set to define separate regions. This approach is quite interesting, as it decreases the number of level-set to $\log_2(n)$ in order to model n regions. For example, it would require 2 level-set function to define 4 regions.

In the following, we will focus on the case of the simplified bi-partitioning segmentation, with two regions Ω_{in} and Ω_{out} .

In summary, **region-based models consist in choosing adequate penalty functions, that will specify the pixels composing the regions.** Several region-based models have been proposed, but we can again classify them in two categories: those expressed directly through a meaningful **functional**; and those derived from a **Bayesian formulation**. However, these two different approaches can lead in some cases to the same functional, under some specific conditions.

3.6.1 Functional-based models

The first category of region-based models are those whose functional is directly expressed in terms of image properties.

Chan-Vese Model

One of the most famous region-based model is the Chan&Vese model (Active contour without edges) [19], that gained in popularity mainly due to its simplicity and efficiency. The main idea is to see the segmentation as a **piecewise constant partition** of the image. A scalar value is associated to each region, and the penalty functions are expressed as the distance between pixels to their corresponding region-wise scalar value. For the bi-partitioning case, this functional is defined by:

$$E(\Gamma, \mu_{in}, \mu_{out}) = \int_{\Omega_{in}} |\mu_{in} - I(x)|^2 dx + \int_{\Omega_{out}} |\mu_{out} - I(x)|^2 dx + \nu |\Gamma| \quad ,$$

and its corresponding level-set formulation:

$$E(\Phi, \mu_{in}, \mu_{out}) = \int_{\Omega} |\mu_{in} - I(x)|^2 H_{\Phi} + |\mu_{out} - I(x)|^2 (1 - H_{\Phi}) + \nu |\nabla H_{\Phi}| dx \quad ,$$

where μ_{in} and μ_{out} are the two scalars corresponding to regions Ω_{in} and Ω_{out} . $H_{\Phi} = H_{\alpha}(\Phi(x))$ the Heaviside of the level-set function Φ , and ν the amount of regularization of the front⁴.

In order to minimize this functional, derivatives are performed:

- keeping Φ fixed, derivative of this functional with respect to μ_{in} and μ_{out} is straightforward, and gives us the values $\mu_{in}(\Phi)$ and $\mu_{out}(\Phi)$:

$$\left\{ \begin{array}{l} \frac{\partial E(\Phi, \mu_{in}, \mu_{out})}{\partial \mu_{in}} = 0 \quad \Longrightarrow \quad \mu_{in}(\Phi) = \frac{\int_{\Omega} I(x) H_{\Phi} dx}{\int_{\Omega} H_{\Phi} dx} \\ \frac{\partial E(\Phi, \mu_{in}, \mu_{out})}{\partial \mu_{out}} = 0 \quad \Longrightarrow \quad \mu_{out}(\Phi) = \frac{\int_{\Omega} I(x) (1 - H_{\Phi}) dx}{\int_{\Omega} (1 - H_{\Phi}) dx} \end{array} \right. .$$

⁴The length of the front is expressed as the integral of the level-set Heaviside gradient

- keeping μ_{in} and μ_{out} fixed, the derivative of this functional with respect to Φ is done using Euler-Lagrange equations, and gives us the evolution equation:

$$\boxed{\frac{\partial \Phi}{\partial t} = -\frac{\partial E}{\partial \Phi} = \delta_{\Phi} \left(|\mu_{in} - I(x)|^2 + |\mu_{out} - I(x)|^2 + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right),}$$

where $\delta_{\Phi} = \delta_{\alpha}(\Phi(x))$ a regularized Dirac of the level-set function Φ .

The evolution of the front is then performed by alternating between the computation of μ_{in} and μ_{out} which correspond to **average means** of the two regions, and the level-set evolution. The front will likely evolve in order to separate the bright areas from the dark ones, and will not be influenced by the sharpness of image boundaries. This model is particularly suited to segment objects having a different “color” than the background, but will become limited when both share the same average intensity.

Mumford-Shah functional

The Mumford-Shah (MS) functional [73, 74] introduced in 1989 was one of the first model proposed to segment regions. Its initial purpose was to perform both segmentation and restoration, by computing a **piecewise smooth partition** (μ, Γ) of images thanks to the minimization of the following functional:

$$E(\Gamma, \mu) = \underbrace{\int_{\Omega} |\mu(x) - I(x)|^2 dx}_{\text{mean square data term}} + \underbrace{\alpha \int_{\Omega - \Gamma} |\nabla \mu(x)|^2 dx}_{\text{smooth regions}} + \underbrace{\nu |\Gamma|}_{\text{smooth boundaries}},$$

where $\mu(x)$ is a **piecewise smooth function** approximating the image $I(x)$, and Γ a set of discontinuities in $\mu(x)$. This functional is similar to the Chan&Vese functional when the middle term is neglected, in the case of piecewise constant function.

Minimization of this functional is quite difficult, and several approaches have been proposed [1, 18]. Two methods using the level-set framework have been proposed independently in [20] and [107]. For the bi-partitioning case, the MS functional is reformulated in level-set notation:

$$\begin{aligned} E(\Phi, \mu_{in}, \mu_{out}) &= \int_{\Omega} |\mu_{in}(x) - I(x)|^2 H_{\Phi} + |\mu_{out}(x) - I(x)|^2 (1 - H_{\Phi}) dx \\ &+ \alpha \int_{\Omega} |\nabla \mu_{in}(x)| H_{\Phi} + |\nabla \mu_{out}(x)| (1 - H_{\Phi}) dx \\ &+ \nu \int_{\Omega} |\nabla H_{\Phi}| dx . \end{aligned}$$

with the same notations as before, and μ_{in} and μ_{out} two functions defined on Ω . Similarly to the previous approach, derivatives of the functional are performed:

- keeping Φ fixed, derivative of the functional with respect to $\mu_{in}(x)$ and $\mu_{out}(x)$ provide two PDEs, that can be solved using the following gradient descent:

$$\left\{ \begin{array}{l} \frac{\partial \mu_{in}}{\partial t} = \alpha \Delta \mu_{in} + (I - \mu_{in}) \quad \text{in } \Omega_{in} \quad ; \quad \frac{\partial \mu_{in}}{\partial \vec{n}} = 0 \quad \text{on } \Gamma \\ \frac{\partial \mu_{out}}{\partial t} = \alpha \Delta \mu_{out} + (I - \mu_{out}) \quad \text{in } \Omega_{out} \quad ; \quad \frac{\partial \mu_{out}}{\partial \vec{n}} = 0 \quad \text{on } \Gamma . \end{array} \right.$$

- keeping μ_{in} and μ_{out} fixed, the derivative of this functional with respect to Φ gives the following evolution equation:

$$\boxed{\frac{\partial \Phi}{\partial t} = \delta_{\alpha} \left(-|\mu_{in}(x) - I(x)|^2 + |\mu_{out}(x) - I(x)|^2 - \alpha |\nabla \mu_{in}|^2 + \alpha |\nabla \mu_{out}|^2 + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right),}$$

with the same notations as before.

This approach can be seen as an extension to the previous Chan&Vese model, providing segmentation of piecewise smooth regions instead of piecewise constant regions. However this model is not as robust as the Chan&Vese model concerning front initialization. Indeed, the initial front has to be rather close to the object to segment in order to evolve toward it (the same problem is evoked in section 5.5.1).

Another inconvenient is that this approach is computationally expensive, since several PDEs need to be solved at each iteration of the level-set evolution. These PDEs are solved using a gradient descent scheme, and therefore the whole minimization process can be seen as gradient descent (PDEs solution) included inside another gradient descent (level-set evolution). The number of iterations for each descent may be rather large, and this can lead to a very expensive scheme.

In Chapter 5, a new segmentation model that minimize the Mumford-Shah functional is proposed, with a computational complexity largely improved using recursive filters to perform the smoothing process.

3.6.2 Bayesian-based models

The second category of region-based models are those derived from a Bayesian formulation. The main idea is no more to define regions as constant or smooth functions, but to define them in term of **probability density function**. These functions correspond to the probability of a given pixel to belong to the region, and their log-likelihood play the role of the “penalty function” previously used. The *maximum a posteriori* (MAP) segmentation is then obtained by minimizing functionals similar to Eq. (3.7). You can refer to section 5.2.2 for a more detailed insight about Bayesian formulations.

For example, in the bi-partitioning case, we will obtain functionals of the form:

$$E(\Omega_{in}, \Omega_{out}) = \int_{\Omega_{in}} -\log p_{in}(I(x)) dx + \int_{\Omega_{out}} -\log p_{out}(I(x)) dx + \nu |\Gamma| \quad ,$$

where $p_{in}(I(x))$ and $p_{out}(I(x))$ are the probability functions for the pixel $I(x)$ to belong respectively to regions Ω_{in} and Ω_{out} . The functional is then reformulated in terms of level-set notations:

$$E(\Phi) = \int_{\Omega} -\log(p_{in}(I(x))) H_{\Phi} - \log(p_{out}(I(x))) (1 - H_{\Phi}) + \nu |\nabla H_{\Phi}| dx \quad ,$$

and evolution equation is obtained by deriving the functional with respect to Φ :

$$\boxed{\frac{\partial \Phi}{\partial t} = -\frac{\partial E}{\partial \Phi} = \delta_{\Phi} \left(-\log \left(\frac{p_{in}(I(x))}{p_{out}(I(x))} \right) + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right)}, \quad (3.8)$$

The main idea is to consider the front as a set of “trial pixels”, and to compute for each of them the probability to belong to regions Ω_{in} or Ω_{out} . Each pixel is then assigned to the region having the highest probability, and the front will evolve according to these successive classifications.

Bayesian models differ depending on the probability density function chosen to define regions. Two kind of function can be used: **parametric density functions** and **non-parametric density functions**.

Parametric density functions

Parametric density functions rely in expressing distributions with a finite number of parameters. A common, and simple choice is given by the Gaussian distribution [88, 89]:

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad ,$$

defined by 2 parameters : the **mean** μ , and the **variance** σ^2 .

The functional is then expressed according to the parameters of the two regions:

$$E(\Phi) \longrightarrow E(\Phi, \mu_{in}, \mu_{out}, \sigma_{in}^2, \sigma_{out}^2) \quad ,$$

and derivative of the functional,

- with respect to μ_{α} and σ_{α}^2 , $\alpha \in \{\text{in}, \text{out}\}$ gives us the following formulas:

$$\left\{ \begin{array}{l} \frac{\partial E(\Phi, \mu_{in}, \mu_{out}, \sigma_{in}^2, \sigma_{out}^2)}{\partial \mu_{\alpha}} = 0 \implies \mu_{\alpha} = \frac{1}{|\Omega_{\alpha}|} \int_{\Omega_{\alpha}} I(x) dx \\ \frac{\partial E(\Phi, \mu_{in}, \mu_{out}, \sigma_{in}^2, \sigma_{out}^2)}{\partial \sigma_{\alpha}^2} = 0 \implies \sigma_{\alpha}^2 = \frac{1}{|\Omega_{\alpha}|} \int_{\Omega_{\alpha}} (I(x) - \mu_{\alpha})^2 dx. \end{array} \right.$$

- with respect to the Φ , gives us the following evolution equation:

$$\frac{\partial \Phi}{\partial t} = \delta_{\Phi} \left(\frac{(I(x) - \mu_{in})^2}{2\sigma_{in}^2} - \frac{(I(x) - \mu_{out})^2}{2\sigma_{out}^2} + \log \frac{\sigma_{in}^2}{\sigma_{out}^2} + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right)$$

The front will then evolve by alternating between computation of region mean and region variance, and level-set evolution. Computation of the mean and variance can moreover be optimized, by incrementally adding or removing contributions of the pixels moving between the two regions, from one iteration to another.

This model is simple and efficient, and **generalizes the Chan&Vese model [19], which can be identified to the case of constant variances $\sigma_{\alpha}^2 = \frac{1}{2}$** . However, using this density implies that regions follow a Gaussian distribution, which may not be true in all cases. Moreover, some numerical problems may arise in the case of almost homogeneous regions, due to division by zero ($\sigma_{\alpha}^2 \rightarrow 0$).

In Fig. 3.5, the evolution of the front is performed using the Bayesian Gaussian model. The regions-based segmentation model allows for a more flexible initialization, and a common choice is to initialize the region Ω_{in} as the inside of several circle aligned on a regular grid, and the region Ω_{out} as the background located between the circles. The segmentation finally converges by separating a bright region from a dark one.

Note that if the two initial regions share the same distribution, the front will not move. A slight difference in the distributions of the two regions is required, in order to separate them significantly (consequently, the Chan&veese model as well).

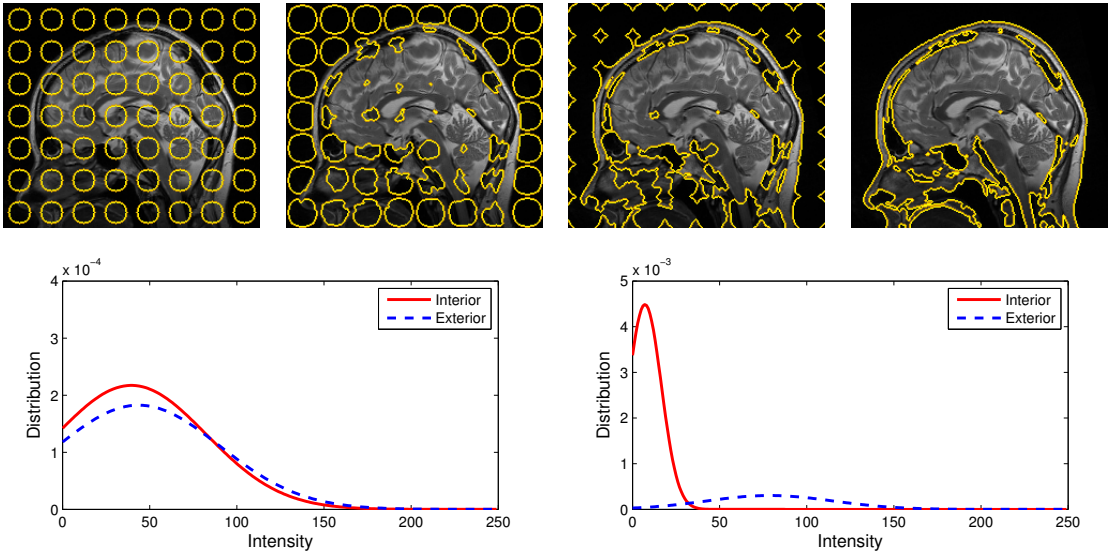


Figure 3.5: Top : Evolution of the contour using the Gaussian model; Bottom : Gaussian distribution of the regions at initialization (left) and convergence (right).

Non-parametric density functions

A more general approach is to consider non-parametric density functions [120, 79, 55]. The main idea is, given an observed dataset (the regions), to estimate an underlying probability density function that does not belong to a family of parametric function anymore. A common choice is to use a **kernel density estimation**, known as **Parzen windows method**.

This method estimates the density functions of the regions by “summing” all the pixels inside regions, with a *kernel function* $K(x)$ applied on each of them. This “summation process” results in convolving region histograms with a kernel function $K(x)$. In the case of generic kernels $K(x)$, the probability density function of a region Ω_α is defined by:

$$\begin{aligned} p_\alpha(x) &= \frac{1}{|\Omega_\alpha|} \sum_{y \in \Omega_\alpha} K(x - I(y)) \\ &= \frac{1}{|\Omega_\alpha|} \sum_{k \in \mathcal{B}} \mathcal{H}_\alpha(k) K(x - k) = \frac{1}{|\Omega_\alpha|} (\mathcal{H}_\alpha * K)(x), \end{aligned}$$

where $\mathcal{H}_\alpha(k)$ is the histogram of the image $I(x)$ inside region Ω_α , composed of a finite set “bins” \mathcal{B} . **The main idea is thus to define probability density functions of regions through their corresponding histograms.**

However, a problem concerning the choice of the kernel function remains open. A common choice is to choose a Gaussian kernel, and therefore to express the probability density function as a simple Gaussian convolution of the region histograms. However the variance of these Gaussian kernel (“kernel size”) has to be chosen a priori, and its incidence on the segmentation is important. For example, using very large Gaussian kernels somehow generalizes the parametric Gaussian model, as the probability density function become similar to Gaussian functions⁵. Several models have been proposed to automatically choose a correct size of kernel, but they suffer from an important increase in computational complexity.

Differentiating the functional using this kind of probability density function is not straightforward, and in general additional terms appear in the evolution equation [55, 26]. In general, these additional terms can be neglected and the evolution is done using Eq. (3.8). Evolution of the front is then performed by alternating between histogram computation, and front evolution.

This model has proved to be very powerfull in a majority of cases, especially on natural images. Discrimination between objects and background through their corresponding histograms appears to be a rather good choice, and this approach has been extensively used in the last few years.

In the *geodesic active region* model [79], Paragios and Deriche proposed to express the segmentation functional as the sum of a region term (non-parametric based

⁵Due to the large amount of “smoothness” on the region histogram

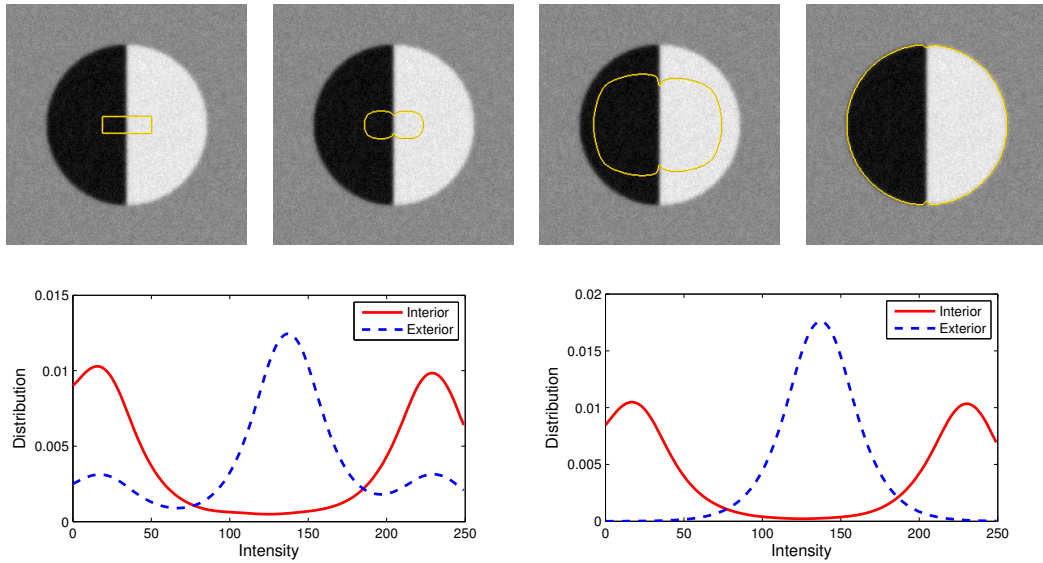


Figure 3.6: Top : Evolution of the contour using a non-parametric Parzen model; Bottom : Distribution of the regions at initialization (left) and convergence (right).

model), and a boundary term (geodesic active contour model). This model combines the advantages of both region and boundary models by adding the contributions of two different evolutions in the final evolution equation. But, this model requires some parameters that may be difficult to adjust.

In Fig. 3.6, evolution of the front using a non-parametric model is performed on a simple synthetic image. Probability density functions of the inside and outside regions are computed using their corresponding histograms (Fig. 3.6, bottom row), and the contour evolves until it separates significantly both regions (Fig. 3.6, top row). Note that the parametric Gaussian model would not have worked in this case (consequently, the Chan&Vese model as well), as both regions share the same global mean.

In Fig. 3.7, segmentation of the inside of the skull (brain + cerebrospinal fluid) is performed on a T2 MRI, using a non-parametric model. The non-parametric model is well suited to segment this kind of anatomical structure, as the histogram of the brain is composed of mainly two bright intensities (white for the CSF, and grey for the brain), while the skull is rather dark.

3.7 CONCLUSION

In this chapter, we presented an overview of different techniques for image segmentation. A diagram of all these notions can be seen in Fig. 3.8. We first briefly reviewed discrete models for image segmentation, that gained in popularity in the last few years, due to the popularization of the graph-cuts optimization method. These methods exhibit several advantages (globals optimas, fast schemes, numerical sta-

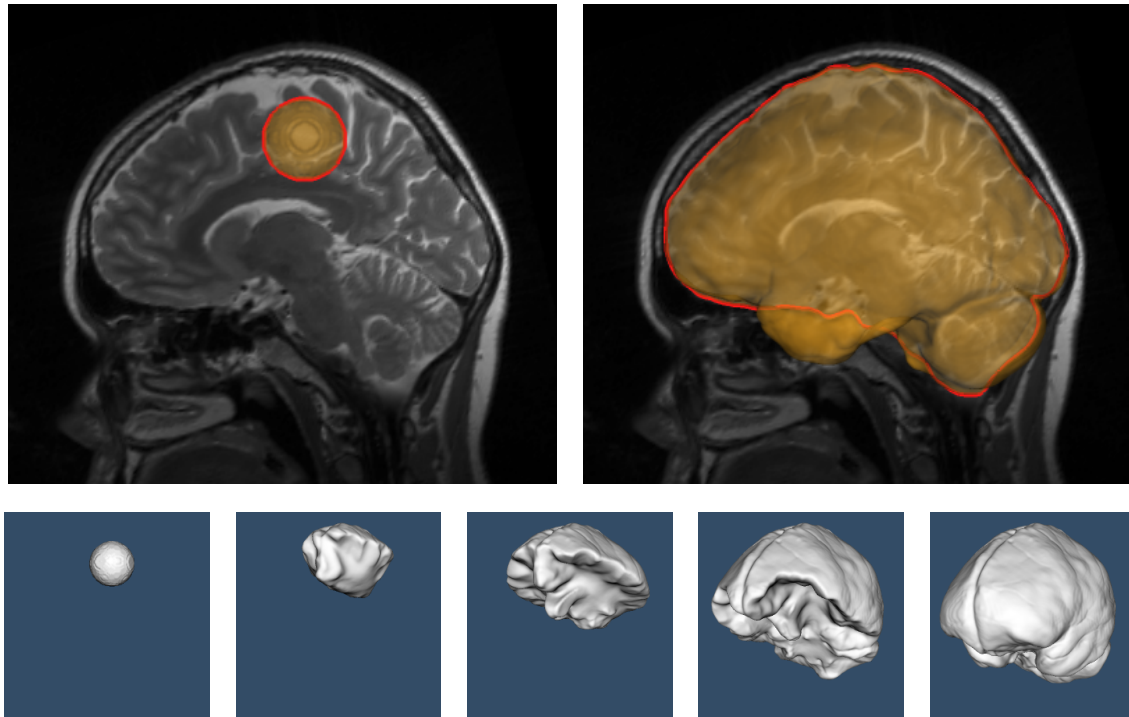


Figure 3.7: 3D Segmentation of the inside of the skull (brain + cerebrospinal fluid) on a T2 MRI, using a non-parametric segmentation model.

bility), but suffer from some drawbacks (memory expensive, restrictions on energy, no shape prior).

Then, we reviewed continuous models for image segmentation which consist in evolving contours in images. Two kinds of contour representations can be used: explicit models that depend on a parameterization; and implicit models introduced by the level-set method, that overcome some important drawbacks but increase the computational complexity.

Evolution of these contours is done through the minimization of a given functional, defining segmentation models. Two different types of models have been considered: boundary-based models, that provide only a very local criteria to stop the evolution of the contour; and region-based models, that provide a more global criteria in order to separate significantly the two regions delimited by the contour.

Region-based models can furthermore be classified in two separate categories: those whose functional is directly expressed in term of image characteristics, and those derived from a Bayesian formulation. However, several links exist between these two categories (Fig. 3.8, blue curves).

Each model brings its own set of advantages and inconvenients, and no model can pretend to be robust to all alterations present in images (noise, intensity drifts, etc). In chapter 5, a new model based on local statistics is proposed, that aims at unifying both region-based and boundary-based models.

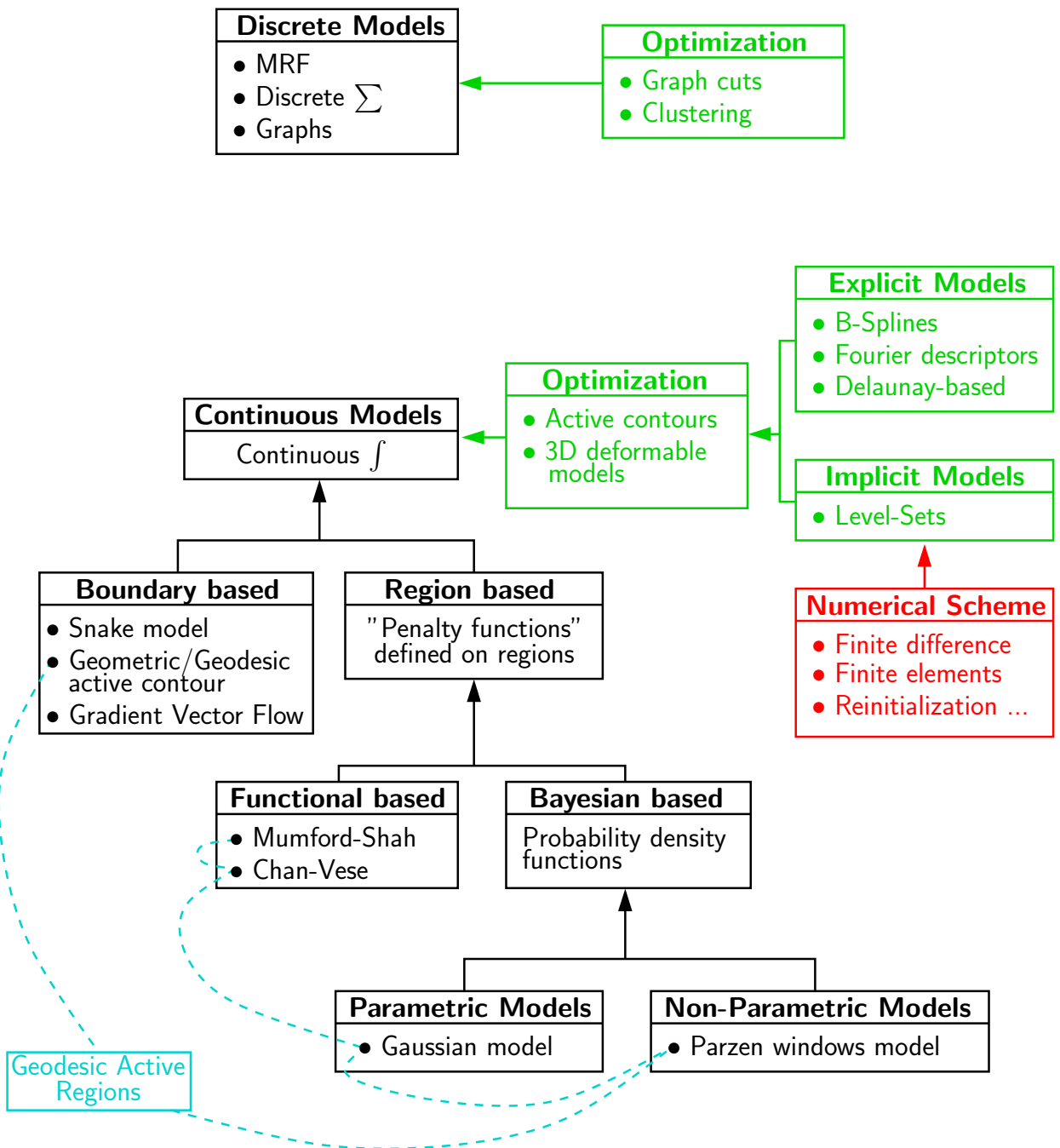


Figure 3.8: Summary of models for image segmentation: there are two separate categories (discrete and continuous) having their own optimization methods (green squares). The continuous models can be seen in a hierarchical way. The blue curves correspond to the possible link between two models.

IMPLEMENTING LEVEL SETS USING THE FINITE ELEMENT METHOD

Contents

Overview	48
4.1 Introduction	48
4.2 Finite Element Method	50
4.2.1 Variational Formulation	50
4.2.2 Discretization of the solution	52
4.2.3 Discretization of the weak formulation	54
4.3 Finite Element Implementation of the Level Set Method . . .	55
4.3.1 Formulation of the Problem	55
4.3.2 Time Discretization : implicit scheme	56
4.3.3 Variational Formulation	57
4.3.4 Discretization of the Solution & Algorithmics	58
4.3.5 Discretization of the weak formulation	61
4.3.6 Solving the system	63
4.4 Results and Comparisons	65
4.5 Conclusion & discussion	66

OVERVIEW

In this chapter, we propose an novel implementation of the level-set method using the finite element method with quadrilateral elements. After briefly explaining the basis of the finite element method and some notions about the type of *elements* and the *discretization*, we show how to solve the level-set evolution PDE using finite elements. Similarly to the narrow band approach (Section 3.4.2), the computation of the distance function can be restricted to the elements that are close to the zero level-set, and thus can be represented as a sparse set of connected elements. To keep track of the zero level-set during its evolution, some elements are inserted to the set, and others removed from the set depending on the motion of the curve. Finally some results are shown in 2D and 3D.

Keywords: Level-set, Numerical scheme, Finite elements method, Mesh, P1 elements, Q1 elements, Galerkin

4.1 INTRODUCTION

The level-set method (Section 3.4.2) is a simple and efficient method used to perform the evolution of a closed contour through a time-dependent partial differential equation (PDE). It thus transform the problem of evolving a curve into the one of solving a specific PDE of the form:

$$\frac{\partial \Phi}{\partial t} + F |\nabla \Phi| = 0, \quad (4.1)$$

where Φ is the level-set function and F the speed of the front in its normal direction.

Several numerical scheme have been proposed to solve this PDE [94, 6], and the most common choice is to use the **finite difference method** (FDM) with an *explicit scheme* (Section 3.4.2). While being relatively fast and simple to implement, the FDM is also one of the least accurate and least stable numerical scheme. It is often required to take relatively small time step in the evolution of the level-set, to ensure the stability of this method.

Solving the evolution PDE alone does not guarantee that the level-set function remains a distance function during its evolution, which is important to accurately compute geometric properties of the contour such as the curvature or the normal. As a consequence, another PDE has to be alternately solved in order to “**reinitialize**” the level-set into a distance function again. This equation, known as *Eikonal equation*, ensures that the slope of the distance function remains 1, and is defined as follows:

$$|\nabla \Phi(x)| = 1 \quad \forall x \in \Omega, \quad (4.2)$$

where Ω the domain of definition of the level-set function Φ .

Solving this equation is equivalent to re-computing a distance map to the active contour, and this can be approximated very efficiently with iterative schemes such as the *fast marching method* or the *fast sweeping method* (Section 3.4.2).

In summary, usual numerical schemes for the the level-set method suffer from two limitation :

- they are based on the FDM with explicit scheme which often produce instabilities if the time step is too big.
- they have to reinitialize the level-set function each k iterations by performing a reinitialization PDE or an iterative redistancing using the fast marching method.

Several methods have been proposed to overcome these limitations.

In [36], the authors proposed a method to evolve the level-set function without the need of a reinitialization process. It is based on computing the speed of the level-set function only at the points neighboring the zero level-set, and *propagating* this speed in the entire function by following the normal direction of the front using an advection PDE. Unfortunately, this framework is still based on the FDM, and thus the stability and the convergence of the level-set may not be guaranteed.

In [61], the authors formulated the energy of the evolution with an additional term which penalize the deviations of the level-set function from a distance function. The derivation of this energy lead to a new term in the evolution equation which is supposed to preserve the distance function without the need of a reinitialization process. Unfortunately this formulation bring another competitive term in the derivative, whose weight has to be accurately chosen. Moreover, being based on an explicit scheme, this method does not allow to choose large time steps in order to preserve the distance function, thus increasing the total number of iterations.

In [109] and [110], the authors proposed a stable and efficient numerical scheme for the level-set method based on the **finite elements method** (FEM) instead of the FDM. They solved both level-set evolution equation and Eikonal equation simultaneously, using an implicit scheme which allow to choose large time step in the level-set evolution without having to reinitialize it. These new method bring several advantages while introducing some new drawbacks. These are investigated in this chapter.

In the following, we explain some basis of the finite elements method, and how it can be used to implement the level-set method. Following [109] and [110], we propose a novel framework for implementing the level-set method using the FEM with quadrilateral elements (Q^1) instead of the triangular elements (P^1) used in these works.

4.2 FINITE ELEMENT METHOD

Several methods exist to find approximate solutions of PDEs. The finite difference method (FDM) is one of the most common choice, and consist in expressing the solution at discrete points on a regular grid and approximating differential operators as finite differences between neighbor points. It has the advantage of being extremely simple to implement and computationally fast, but it is known to be the least stable and the least accurate method. Others methods such as **finite volumes method** or **finite elements method** are a little more complicated to implement, but give significantly better results. f

The **finite elements method** [121] is a powerful numerical scheme for approximating the solutions of PDEs. This method is based on partitioning the domain of the solution into several *elements* of arbitrary forms (polygons in $2D$, polyhedra in $3D$), thus leading to the creation of a *mesh* composed by vertice and elements. The solution of the PDE is then computed at the vertice positions, and interpolated inside each elements, providing a continuous piecewise polynomial approximation of the solution. Because finite elements methods can be adapted to problems of great complexity and unusual geometry, these are extremely powerful tools in the solution of important problems in heat transfer, fluid mechanics, and mechanical systems.

Let's see step by step how to use this method in order to solve a PDE of the form:

$$\left\{ \begin{array}{l} \mathcal{A}(u(x)) = 0 \quad \forall x \in \Omega \\ \underbrace{u(x) = f(x)}_{\text{Dirichlet}} \quad \text{or} \quad \underbrace{\frac{\partial u}{\partial \vec{n}}(x) = f(x)}_{\text{Neumann}} \quad \forall x \in \partial\Omega \end{array} \right.$$

where u is the solution to approximate, \mathcal{A} denotes any linear operator of the function u , Ω the domain of definition of u , $\partial\Omega$ the boundary of Ω , and f a smooth function defined on $\partial\Omega$. A border condition is necessary in order to correctly define the function u , and various type can be chosen (Dirichlet, Neumann, ...).

4.2.1 Variational Formulation

The first step of the FEM is to rephrase the original problem into its weak, or variational form.

This reformulation is based on the following functional analysis theorem :

$$f(x) = 0 \quad \forall x \in \Omega \quad \Leftrightarrow \quad \int_{\Omega} f(x)g(x) dx = 0 \quad \forall g \in H_0^1(\Omega) .$$

If a function f equal zero in a domain Ω , then its scalar product with any differentiable function g defined on Ω equals zero ¹. The second part is called the *weak formulation* of the first part.

¹under certain regularity conditions.

In our case, we get

$$\boxed{\mathcal{A}(u(x)) = 0 \quad \forall x \in \Omega \quad \Leftrightarrow \quad \int_{\Omega} \mathcal{A}(u(x)) w(x) dx = 0 \quad \forall w(x)}.$$

The purpose of this notation is to transform the initial PDE into a system of equation, by choosing a finite number of independent *test functions* w (each of them creating a new equation). Then, the main idea is to discretize the solution u by using a finite number of values, and finding these values by solving a system of n equations (induced by the different test functions w) with n unknowns (the unknown representing the solution u).

In general, PDEs with high differential orders are difficult to solve, as they impose some constraints on the polynomial order of the discretization basis of the solution². Fortunately, the weak formulation² allows to decrease the differential order of the equations, by applying some **integral transformations** based on integration by parts. For example, the **Green theorem** consists in transforming an integral of a second order differential function by two integrals of first order differential functions, and is defined as follows:

$$\int_{\Omega} -(\Delta \mathbf{a} \cdot \mathbf{b})(x) dx = \int_{\Omega} (\nabla \mathbf{a} \cdot \nabla \mathbf{b})(x) dx - \int_{\partial\Omega} \langle \nabla \mathbf{a}(x), \mathbf{n}(x) \rangle \mathbf{b}(x) dx \quad (4.3)$$

where \mathbf{a} and \mathbf{b} are any functions defined on a domain Ω , $\partial\Omega$ referring to the border of Ω , and $\mathbf{n}(x)$ the normal vector to Ω at point x oriented toward the exterior (Fig. 4.1). Moreover, this transformations allows to include the border condition of the PDE in the weak formulation, by replacing the correct values in the boundary integral.

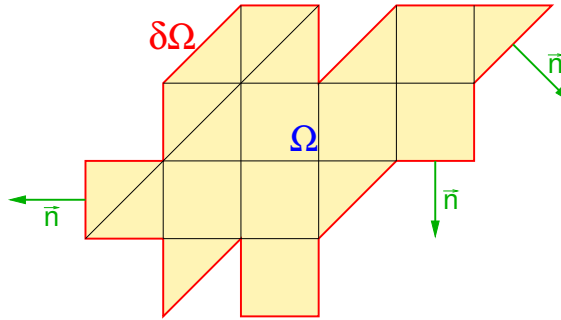


Figure 4.1: $\partial\Omega$ defines the border of a domain Ω , and \vec{n} the normal vector to Ω .

In summary, little to no computation is usually required for this step, and all the transformations are done by hand on paper.

²For example, if the order of derivation of the PDE is *two* (e.g. Laplacian), the polynomial order of the discretization basis has to be at least *two* (e.g. solution piecewise quadratic).

4.2.2 Discretization of the solution

The second step is the discretization of the solution. It raises the following question: how to approximate a continuous function u defined on a specific domain Ω with a finite set of discrete values ?

The most common way is to partition the domain Ω into a **mesh** (Fig. 4.2). Only the values at the vertex position of the mesh are sufficient to approximate the solution. These values are then interpolated inside the elements of the mesh, and the solution becomes a *piecewise polynomial approximation* of u .

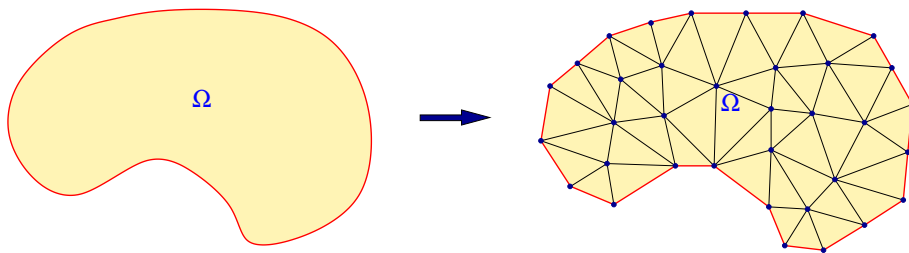


Figure 4.2: Partition of a 2D domain Ω in triangles.

Choice of the type of elements

In general, the partition of the domain is done by using an unique type of element, but it's not necessary. Any type of element can be chosen, but this choice is generally limited between two types: elements of type P^n and elements of type Q^n .

- Elements of type P^n (Fig. 4.3) are triangles in $2D$, tetrahedra in $3D$, and more generally n -simplex in nD . These elements bring several advantages concerning the mesh generation, as usual triangulation methods (*Delaunay*) can be used to perform the mesh. Therefore, these elements are the most adapted if the problem is defined on a geometrically complex domain.

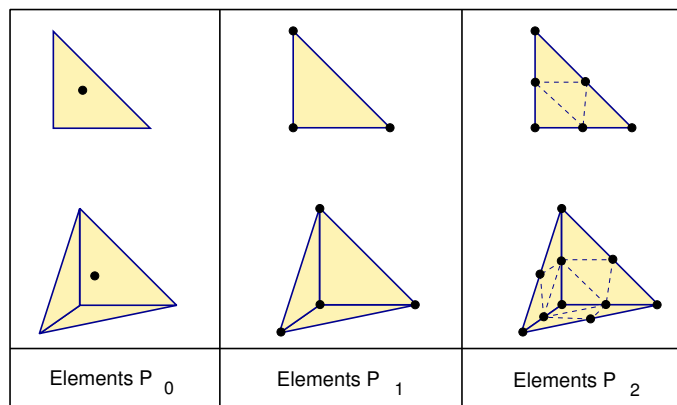


Figure 4.3: Elements of type P^n .

- Elements of type Q^n (Fig. 4.4) are squares in $2D$, cubes in $3D$, and more generally hypercubes in nD . Depending on the problem, these elements can facilitate the mesh creation. For example, if the mesh must have its vertex positioned on a regular grid, the mesh creation using quadrilateral elements becomes trivial.

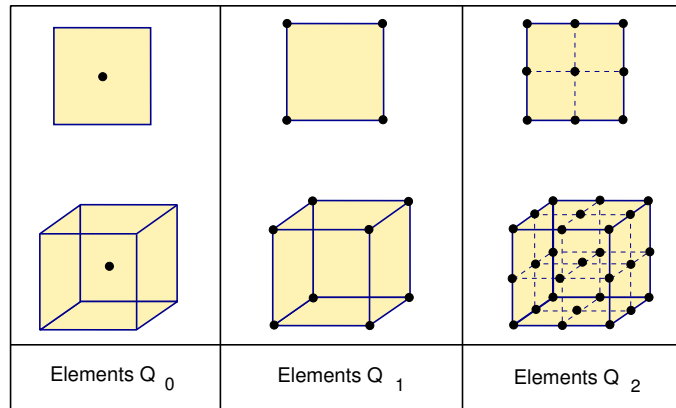


Figure 4.4: Elements of type Q^n .

The ' n ' in the P^n and Q^n notations denotes the polynomial order in each dimensions of the approximation inside the elements. It is linked to the number of **nodal points** (discretization points of the function) related to the element. These nodal points are in general located on the element boundary, but it's not necessary (Fig. 4.3, 4.4). In the following, we will essentially focus on P^1 and Q^1 elements, where each nodal points correspond to a vertex of the mesh.

Basis functions and interpolation

To interpolate the discrete values of u at the nodal points inside the elements of the mesh, **basis functions** are introduced. These functions, often called "*hat functions*" due to their shape, are functions associated to each nodal points of the mesh. The i^{th} basis function φ_i associated to the i^{th} vertex V_i is defined as follows:

- $\varphi_i : \Omega \rightarrow \mathbb{R}$ continuous on Ω
- $\varphi_i(V_j) = \delta_{ij}$ (1 if $i = j$, 0 if $i \neq j$)

In the case of P^1 and Q^1 elements, these functions are respectively *piecewise affine* and *piecewise bilinear* (Fig. 4.5) on the elements of the mesh. Notice that the gradient of these function in the Q^1 case is not constant per element, unlike in the P^1 case (These functions are called "*pagodas*", due to their particular shape). The purpose of these functions is to define a basis of the space where the piecewise polynomial approximations are defined.

Therefore, the piecewise polynomial function \hat{u} approximation of the continuous function u on Ω can be defined as follows:

$$u(x) \approx \hat{u}(x) = \sum_{i=1}^n u_i \varphi_i(x) \quad \forall x \in \Omega,$$

where i are the vertices of the mesh, u_i the values of the solution at vertex i , and $\varphi_i(x)$ the basis function associated to the i^{th} vertex.

In summary, this step consists in choosing a type of element (ie. shape of element with polynomial order), partitioning the domain into a mesh using this kind of element, and computing the basis functions associated to each vertex of the mesh.

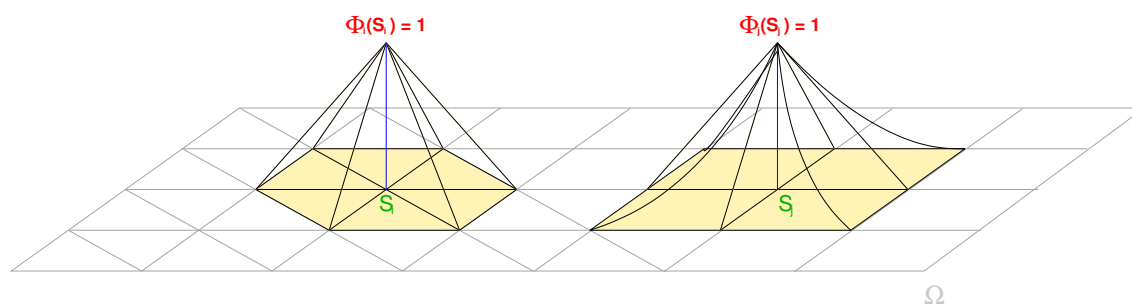


Figure 4.5: Basis functions: piecewise affine functions for triangular P^1 elements (left) and piecewise bilinear functions for quadrangular Q^1 elements (right).

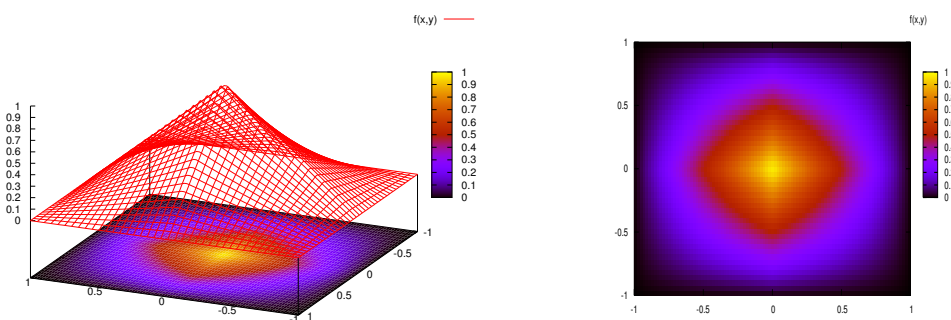


Figure 4.6: Basis functions of Q^1 elements.

4.2.3 Discretization of the weak formulation

The third step consists in discretizing the weak formulation. This commonly means that if u is discretized with n values, finding a set of n independent test functions for the weak formulation in order to transform the PDE into a system of n equations.

with n unknown. Several methods exists (points-subdomain collocation [121], etc...), but the most common discretization scheme is the **Galerkin discretization**: the basis functions associated to each vertex are chosen as the test functions in the weak formulation. Finally, if \mathcal{A} is a linear operator, we obtain the linear system:

$$\mathcal{A}(u(x)) = 0 \quad \Rightarrow \quad \begin{cases} \int_{\Omega} \mathcal{A}(u_1 \varphi_1(x) + \dots + u_n \varphi_n(x)) \varphi_1(x) dx = 0 \\ \int_{\Omega} \mathcal{A}(u_1 \varphi_1(x) + \dots + u_n \varphi_n(x)) \varphi_2(x) dx = 0 \\ \vdots \\ \int_{\Omega} \mathcal{A}(u_1 \varphi_1(x) + \dots + u_n \varphi_n(x)) \varphi_n(x) dx = 0 \end{cases}$$

Basically, each of these equations denotes the local deviation of the solution at the neighborhood of a vertex in the mesh. This leads to a system of n equations with n unknowns, and solving this system ³ will give values u_1, u_2, \dots, u_n of the solutions at specific locations. Interpolating these values inside the elements will finally lead to the piecewise polynomial approximation.

4.3 FINITE ELEMENT IMPLEMENTATION OF THE LEVEL SET METHOD

Let's see how to implement the level-set method using the finite element method.

4.3.1 Formulation of the Problem

Following [109] and [110], we use the finite element method in order to approximate the solution of two equations :

- the level-set evolution equation, for any normal speed F

$$\frac{\partial \Phi}{\partial t} + F(\Phi) |\nabla \Phi| = 0, \quad (4.4)$$

- the Eikonal equation

$$(\nabla \Phi)^2 = 1. \quad (4.5)$$

By solving simultaneously these two equations, the level-set function will remain a distance function during its evolution, and no auxiliary reinitialization scheme will be needed. Solving the Eikonal equation is more like a **constraint** that must be satisfied, thus there is no need of border condition for this PDE. Notice the difference between the reinitialization PDE of the level set (Section 3.4.2), that is formulated as a time dependent PDE and this formulation : in the framework of Section 3.4.2, one has to alternate between *evolution* and *reinitialization*; in our case, both evolution and reinitialization are done simultaneously.

³Using any linear system resolution scheme, such as Gauss, Choleski, or iterative methods like gradient descent.

4.3.2 Time Discretization : implicit scheme

The time derivative appearing in equation 4.4 has to be discretized in a special way. We choose to discretize it using an **implicit scheme**: this means that *the time derivative* is approximated using first order finite difference, thus introducing a time step; and *the spatial derivatives* appearing in the PDEs are calculated at the end of the time step. Compared to **explicit schemes**, where spatial derivatives are calculated at the beginning of the time step, implicit schemes allow to choose much larger time step Δt , and are numerically more stable. However, implicit schemes involve the coupling between unknowns, and therefore require solving a linear system.

Actually, due to our formulation, this implicit scheme is required in order to satisfy the Eikonal equation at the end of the time step. Using an explicit scheme would not have allowed us to include the distance constraint.

So by including time as a parameter, and replacing the continuous function Φ by its discrete approximation u , Eq. 4.4 becomes:

$$\mathbf{u}(x, t + \Delta t) = \mathbf{u}(x, t) - \Delta t \mathbf{F}(\mathbf{u}(x, t + \Delta t)) |\nabla \mathbf{u}(x, t + \Delta t)|, \quad (4.6)$$

where the normal speed F is calculated at time $t + \Delta t$.

Thanks to the constraint imposed by Eq. 4.5, the gradient appearing in the second term of Eq. 4.6 has to be 1, and thus can be removed. Moreover, the “incremental function” that has to be added to the function u at time t in order to get u at time $t + \Delta t$ is actually the **unknown** of this equation. By expressing this unknown as a variable v , we get:

$$\mathbf{u}(x, t + \Delta t) = \mathbf{u}(x, t) + \mathbf{v}(x, t),$$

with

$$\mathbf{v}(x, t) = -\Delta t \mathbf{F}(\mathbf{u}(x, t) + \mathbf{v}(x, t)).$$

By expressing spatial derivatives at the end of the time step, Eq. 4.5 becomes:

$$(\nabla (\mathbf{u}(x, t) + \mathbf{v}(x, t)))^2 = 1. \quad (4.7)$$

Finally, the system of Eq. 4.4 and Eq. 4.5 becomes:

$$\left\{ \begin{array}{l} \bullet \text{ the level-set evolution equation consists in finding the “step function” } v \\ \quad \boxed{\mathbf{v} + \Delta t \mathbf{F}(\mathbf{u} + \mathbf{v}) = 0,} \quad (4.8) \\ \bullet \text{ the Eikonal equation calculated at the end of the time step} \\ \quad \boxed{(\nabla \mathbf{u} + \nabla \mathbf{v})^2 - 1 = 0.} \quad (4.9) \end{array} \right.$$

Solving this system will give us the “step” v , that makes the level-set evolve while preserving it as a distance function. Note that the normal speed F is expressed at the end of the time step.

4.3.3 Variational Formulation

The next step is to write the weak formulation of equations 4.8 and 4.9 and apply some integral transformations to reduce the differential order.

The weak formulation of Eq. 4.9 does not need an integral transformations, since it is a first order differential equation.⁴ Therefore, we get:

$$\boxed{\int_{\Omega} (\nabla \mathbf{u} + \nabla \mathbf{v})^2 \mathbf{w} - \int_{\Omega} \mathbf{w} = 0 \quad \forall \mathbf{w} .} \quad (4.10)$$

The weak formulation of Eq. 4.8 depends on the normal speed F applied of the level-set. We choose to use the Gaussian region based model from section 3.6. By expressing F at time $t + \Delta t$, we get:

$$F(\mathbf{u} + \mathbf{v}) = \epsilon \nabla \cdot \left(\frac{\nabla(\mathbf{u} + \mathbf{v})}{|\nabla(\mathbf{u} + \mathbf{v})|} \right) + \frac{(I - \mu_{in}(\mathbf{u}))^2}{\sigma_{in}^2(\mathbf{u})} - \frac{(I - \mu_{out}(\mathbf{u}))^2}{\sigma_{out}^2(\mathbf{u})} . \quad (4.11)$$

The first term corresponds to the regularization term with a weight ϵ , while the second and third terms stand for data attachment terms. We choose to express the data attachment terms at the beginning of the time step in order speed up the gradient descent. Consequently, our model is a semi-implicit model, and we can note these data attachment terms as a constant α precomputed at the beginning of each time-steps. Moreover, due to the distance constraint imposed by Eq. 4.7, the regularization term can be simplified as a Laplacian.

Therefore, the normal speed (Eq. 4.11) becomes:

$$F(\mathbf{u} + \mathbf{v}) = \epsilon \Delta(\mathbf{u} + \mathbf{v}) + \alpha , \quad (4.12)$$

and the weak formulation of Eq. 4.8 is then defined as

$$\int_{\Omega} \mathbf{v} \mathbf{w} - \Delta t \left(\epsilon \int_{\Omega} \Delta(\mathbf{u} + \mathbf{v}) \mathbf{w} + \int_{\Omega} \alpha \mathbf{w} \right) = 0 \quad \forall \mathbf{w} . \quad (4.13)$$

This weak formulation is problematic since it includes a second order differential term (Laplacian), which prevents us from using first order polynomial elements such

⁴We can use first order polynomial elements (P^1 , Q^1) on first order differential equations.

as P^1 or Q^1 elements. In order to decrease the differential order of this equation, we apply the Green theorem (Eq. 4.1). Thus, the weak formulation becomes:

$$\boxed{\int_{\Omega} \mathbf{v} \mathbf{w} - \Delta t \left(-\epsilon \int_{\Omega} \nabla(\mathbf{u} + \mathbf{v}) \cdot \nabla \mathbf{w} + \epsilon \int_{\partial\Omega} \langle \nabla(\mathbf{u} + \mathbf{v}), \mathbf{n} \rangle \mathbf{w} + \int_{\Omega} \alpha \mathbf{w} \right) = 0.} \quad (4.14)$$

where $\partial\Omega$ represents the boundary of the domain Ω , and \mathbf{n} is the normal to Ω oriented toward the exterior.

4.3.4 Discretization of the Solution & Algorithmics

The discretization of the solution involves choosing a domain for computing the solution, and partitioning this domain using an appropriate type of element.

P^1 elements

In [109] and [110], the authors restricted the domain of computation to a thin band located at the neighborhood of the zero level-set, similarly to the *narrow band approach* (Section 3.4.2). To keep track of the zero level-set during its evolution, they introduced some dynamics to insert elements or remove elements from the band. They used regular P^1 elements (triangles in [109], and tetraedra in [110]) to partition this band, thus approximating the level-set as a piecewise affine function (Fig. 4.7).

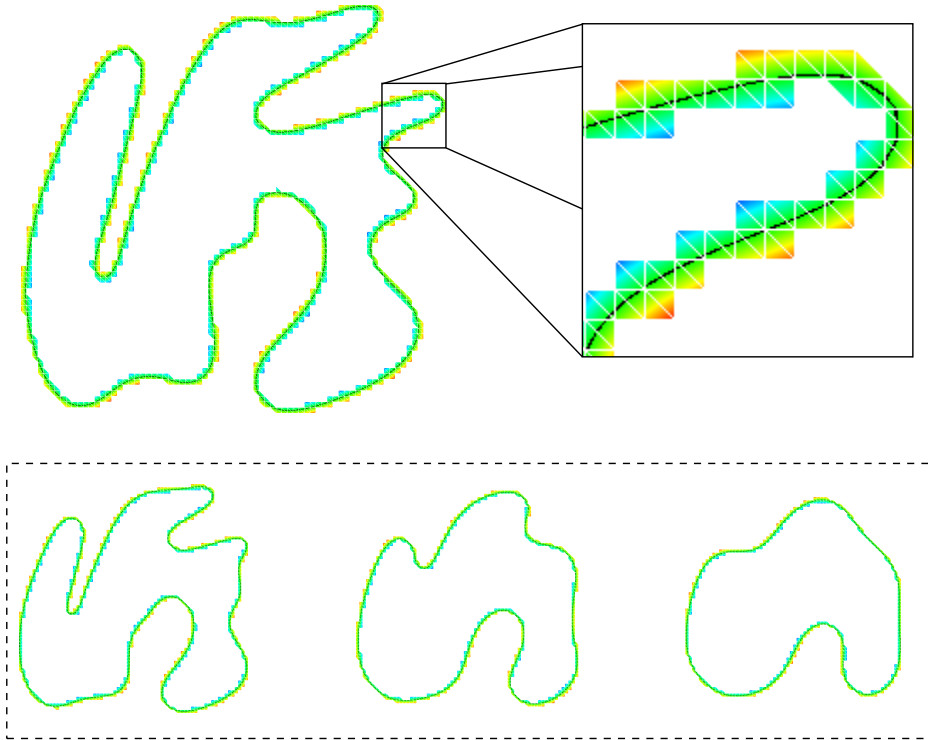


Figure 4.7: P^1 finite element representation of the level set function (top), and its mean curvature motion (bottom).

However, by using such P^1 elements, the authors introduce some problems:

- By using anisotropic meshes, some **bias** may appear in the evolution of the level-set. Indeed, as the Galerkin discretization measures the local deviation of the solution at the neighborhood of a vertex, the accuracy of the solution is directly linked to the “shape“ of this neighborhood. If this neighborhood is not isotropic, the final solution may be affected by this anisotropy, and we may see some slight perturbations in the solution. For example, the evolution of a circle under a mean curvature leads (in the intermediate steps of the evolution) to an ellipsoid oriented by the “mesh aspect” (Fig. 4.8).

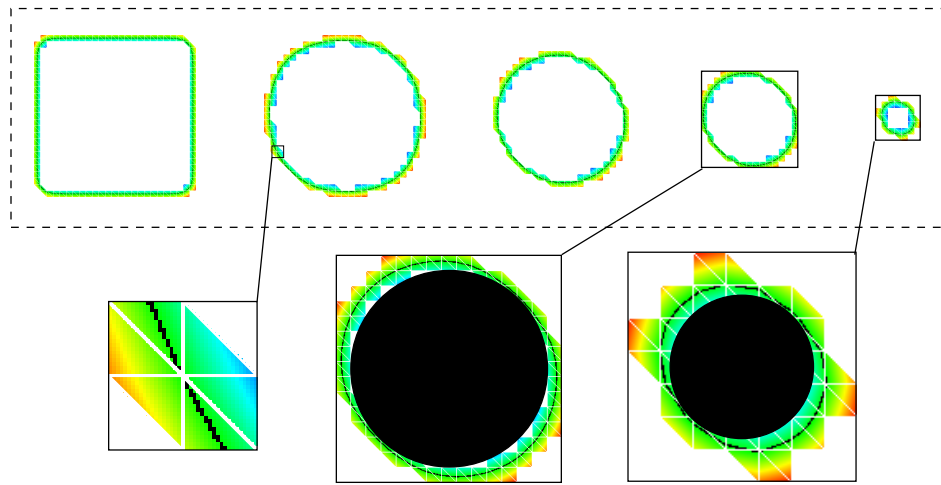


Figure 4.8: top: mean curvature motion of a square ; bottom: anisotropic neighborhood of a vertex, and deviation of the solution from a circle (black discs).

- While the triangulation of a regular grid is trivial in $2D$ as each square produce 2 triangles, it becomes more complicated in $3D$, where each cubes produce 6 tetraedra (Fig. 4.9). This difficulty is even more problematic in our case, as the mesh has to evolve incrementally by adding and removing elements at each iterations.

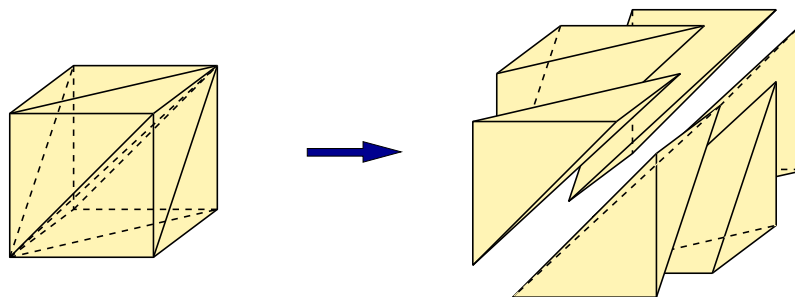


Figure 4.9: 3D Triangulation of a cube: creation of 6 tetraedra per cube.

Q^1 elements

To avoid these problems, we choose to discretize the solution using Q^1 quadrilateral elements. This choice seems more natural in both $2D$ and $3D$, and while it requires more calculations (Cf. Appendix A), it overcomes some important drawbacks of the previous methods [109] and [110]:

- because quadrilaterals tile the image space, there is no need to introduce arbitrary simplicial subdivisions of the image elements (pixels or voxels), thus leading to a simpler method.
- As the vertices of the mesh are in direct correspondence with image pixels, the standard multilinear interpolation scheme can be directly used to provide the continuous interpolant of the discrete solution.
- Using these elements correct the bias induced by anisotropic triangular meshes.

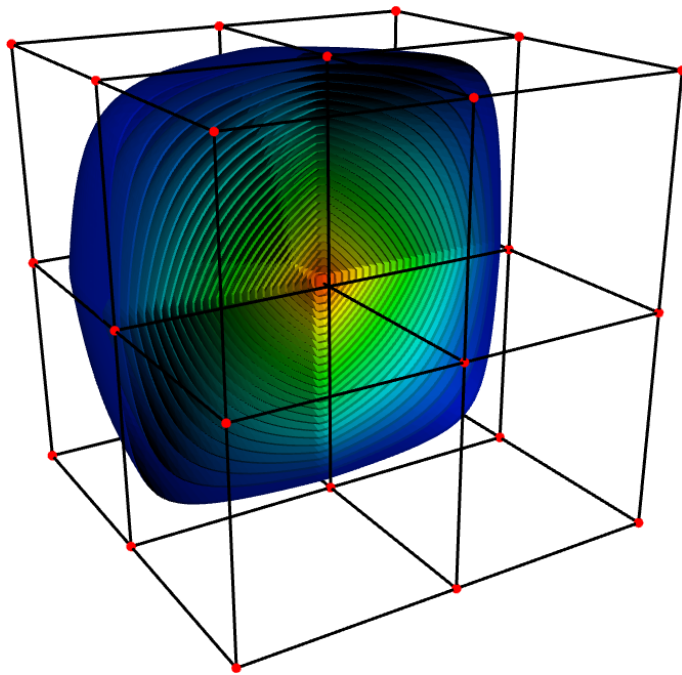


Figure 4.10: Basis functions of a vertex in $3D$ with Q^1 elements.

We use the same dynamics as in [109] and [110], by computing the solution only in a band defined at the neighborhood of the zero level-set, and evolving this band by adding and removing elements to keep track of the zero level-set.

However, adding an element in the band involves extrapolating some values in order to have a good initialization for the next iteration. This extrapolation is performed using a fast marching (Section 3.4.2), but in a simpler way as only a few elements are added at each iterations.

Thanks to this discretization, we can express the discrete approximation $u(x, t)$ of the level-set function Φ , as well as the step function $v(x, t)$ by:

$$\mathbf{u}(x, t) = \sum_{i=1}^n u_i(t) \varphi_i(x) , \quad (4.15)$$

and

$$v(x, t) = \sum_{i=1}^n v_i(t) \varphi_i(x) , \quad (4.16)$$

where $u_i(t)$ and $v_i(t)$ are the values located at the i^{th} vertex at time t of the level-set function and the step function ; and $\varphi_i(x)$ the basis functions associated to the i^{th} vertex and defined on its incident Q^1 elements (Fig. 4.6).

4.3.5 Discretization of the weak formulation

The next step is to discretize both of the weak formulations Eq. 4.10 and Eq. 4.14, by choosing a set of n orthogonal tests functions $w(x)$. We choose the **Galerkin discretization**, which means that we take the basis functions $\varphi_i(x)$ as test functions. This leads to the following two systems of equations.

Eikonal equation

The Galerkin discretization of Eq. 4.10 gives us the system:

$$\int_{\Omega} (\nabla \mathbf{u} + \nabla v)^2 \varphi_i - \int_{\Omega} \varphi_i = 0 \quad i \in \{1, 2, \dots, n\} \quad (4.17)$$

By expressing u and v through their linear combination of basis functions, we get:

$$\nabla \mathbf{u} = \nabla \left(\sum_{a=1}^n u_a \varphi_a \right) = \sum_{a=1}^n u_a \nabla \varphi_a \quad \text{and} \quad \nabla v = \sum_{a=1}^n v_a \nabla \varphi_a ,$$

thus

$$\begin{aligned} (\nabla \mathbf{u} + \nabla v)^2 &= \left(\sum_{a=1}^n u_a \nabla \varphi_a + \sum_{a=1}^n v_a \nabla \varphi_a \right)^2 \\ &= \left(\sum_{a=1}^n (u_a + v_a) \nabla \varphi_a \right)^2 \\ &= \sum_{a=1}^n \sum_{b=1}^n (u_a + v_a) \langle \nabla \varphi_a, \nabla \varphi_b \rangle (u_b + v_b) , \end{aligned}$$

therefore

$$\begin{aligned}
\int_{\Omega} (\nabla \mathbf{u} + \nabla \mathbf{v})^2 \varphi_i &= \int_{\Omega} \left[\sum_{a=1}^n \sum_{b=1}^n (\mathbf{u}_a + \mathbf{v}_a) \langle \nabla \varphi_a, \nabla \varphi_b \rangle (\mathbf{u}_b + \mathbf{v}_b) \right] \varphi_i \\
&= \sum_{a=1}^n \sum_{b=1}^n (\mathbf{u}_a + \mathbf{v}_a) \left(\int_{\Omega} \langle \nabla \varphi_a, \nabla \varphi_b \rangle \varphi_i \right) (\mathbf{u}_b + \mathbf{v}_b) \\
&= (\tilde{\mathbf{u}} + \tilde{\mathbf{v}})^T \mathbf{Q}^i (\tilde{\mathbf{u}} + \tilde{\mathbf{v}}) ,
\end{aligned}$$

and finally, Eq. 4.17 can be reformulated as:

$$\boxed{(\tilde{\mathbf{u}} + \tilde{\mathbf{v}})^T \mathbf{Q}^i (\tilde{\mathbf{u}} + \tilde{\mathbf{v}}) - \int_{\Omega} \varphi_i = 0 \quad i \in \{1, 2, \dots, n\} ,} \quad (4.18)$$

with

$$\tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_n \end{pmatrix}, \quad \tilde{\mathbf{v}} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{pmatrix} \quad \text{and} \quad \mathbf{Q}^i \text{ } n \times n \text{ matrix defined by } \mathbf{Q}_{ab}^i = \int_{\Omega} \langle \nabla \varphi_a, \nabla \varphi_b \rangle \varphi_i$$

Evolution equation

Similarly, the Galerkin discretization of Eq. 4.14 gives us the system:

$$\begin{aligned}
&\forall i \in \{1, 2, \dots, n\} \\
&\int_{\Omega} \mathbf{v} \varphi_i - \Delta t \left(-\epsilon \int_{\Omega} \nabla(\mathbf{u} + \mathbf{v}) \nabla \varphi_i + \epsilon \int_{\partial\Omega} \langle \nabla(\mathbf{u} + \mathbf{v}), \mathbf{n} \rangle \varphi_i + \int_{\Omega} \boldsymbol{\alpha} \varphi_i \right) = 0 ,
\end{aligned}$$

and this system can be reformulated as:

$$\boxed{\mathbf{P}^i \tilde{\mathbf{v}} - \Delta t \left[\epsilon \left(-\mathbf{R}^i + \mathbf{B}^i \right) (\tilde{\mathbf{u}} + \tilde{\mathbf{v}}) + \mathbf{P}^i \boldsymbol{\alpha} \right] = 0 \quad i \in \{1, 2, \dots, n\} ,} \quad (4.19)$$

with

$$\tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_n \end{pmatrix}, \quad \tilde{\mathbf{v}} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{pmatrix} \quad \text{and} \quad \begin{cases} \mathbf{P}^i \text{ vector of size } n \text{ defined by } \mathbf{P}_a^i = \int_{\Omega} \varphi_a \varphi_i \\ \mathbf{R}^i \text{ vector of size } n \text{ defined by } \mathbf{R}_a^i = \int_{\Omega} \langle \nabla \varphi_a, \nabla \varphi_i \rangle \\ \mathbf{B}^i \text{ vector of size } n \text{ defined by } \mathbf{B}_a^i = \int_{\partial\Omega} \langle \nabla \varphi_a, \mathbf{n} \rangle \varphi_i \end{cases}$$

Thanks to the Galerkin discretization, we can compute the step function \mathbf{v} by solving the two systems of equations (4.18) and (4.19).

4.3.6 Solving the system

In order to solve both of the systems 4.18 and 4.19, we use **the conjugate gradient method** [96]. This method is the most popular iterative method for solving large systems of linear equations of the form

$$Ax - b = 0, \quad (4.20)$$

where x is the unknown, b is a known vector, and A is a known, square, symmetric positive-definite matrix. In our case, we first have to reformulate our systems, by expressing them in a similar form as Eq. 4.20. The unknown is the “*step function*” v that needs to be computed in order to update the level-set function. Therefore, we have to solve 2 systems of the form:

$$\begin{cases} A_1 v - b_1 = 0 \\ A_2 v - b_2 = 0 \end{cases} \quad (4.21)$$

where A_1 and A_2 are respectively the matrices of systems 4.18 and 4.19 and b_1, b_2 the corresponding known vectors.

The system 4.18 is not a linear system, as quadratic terms of the unknown v appears in the equations. However, since the deviation is almost affine in the unknown v , the quadratic term $\tilde{v}^T Q^i \tilde{v}$ can be removed, and the system can indeed be linearized. Therefore, the system becomes:

$$\underbrace{2\tilde{u}^T Q^i \tilde{v}}_{A_1^i} + \underbrace{\tilde{u}^T Q^i \tilde{u}}_{-b_1^i} - \int_{\Omega} \varphi_i = 0 \quad \forall i \in \{1, 2, \dots, n\}, \quad (4.22)$$

where A_1^i is a vector corresponding to the i^{th} row of the global matrix A_1 and b_1^i a scalar corresponding to the i^{th} component of the known vector b_1 .

By developing the system 4.19, we find

$$\underbrace{\left[P^i - \Delta t \epsilon \left(-R^i + B^i \right) \right]}_{A_2^i} \tilde{v} - \underbrace{\Delta t \epsilon \left(-R^i + B^i \right) \tilde{u} - \Delta t P^i \alpha}_{b_2^i} = 0, \quad (4.23)$$

where A_2^i is a vector corresponding to the i^{th} row of the global matrix A_2 and b_2^i a scalar corresponding to the i^{th} component of the known vector b_2 .

Unfortunately, both of the matrix A_1 and A_2 are not symmetric, and therefore the conjugate gradient method can not be applied directly. Therefore, instead of solving the linear system 4.21, we solve the following system:

$$\begin{cases} A_1^T (A_1 v - b_1) = 0 \\ A_2^T (A_2 v - b_2) = 0 \end{cases} \quad (4.24)$$

and finally, by combining these two system:

$$(A_1^T A_1 + A_2^T A_2) v - (A_1^T b_1 + A_2^T b_2) = 0 \quad (4.25)$$

Matrices A_1 and A_2 have to be recomputed at each iteration, because they both depend on the level-set function u , constantly updated. Finding the elements composing matrix A_1 and A_2 is purely combinatorial, and actually a simple visit on the elements of the mesh is necessary to compute these values.

In summary, the system 4.25 needs to be solved at each iterations of the level-set evolution, in order to compute the step v that is needed update the level-set function. This step will then make the level-set evolve, while preserving it as a distance function. The computation of this step is performed by using the efficient conjugate gradient method, and therefore is computed in $O(n)$.

The algorithm below explains the different steps of our implementation. It can be seen as a gradient descent performed inside another gradient descent: the inside one is meant to compute the step v needed to evolve the level-set function, while the global one is the evolution of the level set function, which is done by minimizing a given functional.

Algorithm 1 Level-set evolution using Q^1 finite elements

initialize u

repeat

for all elements e **do**

for all edges $\langle i, j \rangle \in e$ **do**

 Compute contribution to matrices A_1 and A_2

 Compute contribution to vectors b_1 and b_2

$v \leftarrow 0$

repeat

$v \leftarrow v +$ conjugate gradient step of system 4.25

until convergence

$u \leftarrow u + v$

for all elements e **do**

if e too far from the zero level-set **then**

remove e from the mesh

insert new elements using fast-marching extrapolation

until convergence

4.4 RESULTS AND COMPARISONS

Actually, as the data attachment term in the evolution equation are computed at the beginning of the time step, our finite element implementation only manages to compute accurately the curvature of the level-set, with a distance preservation constraint. Therefore the following results present mean curvature evolutions with distance preservation using Q^1 elements.

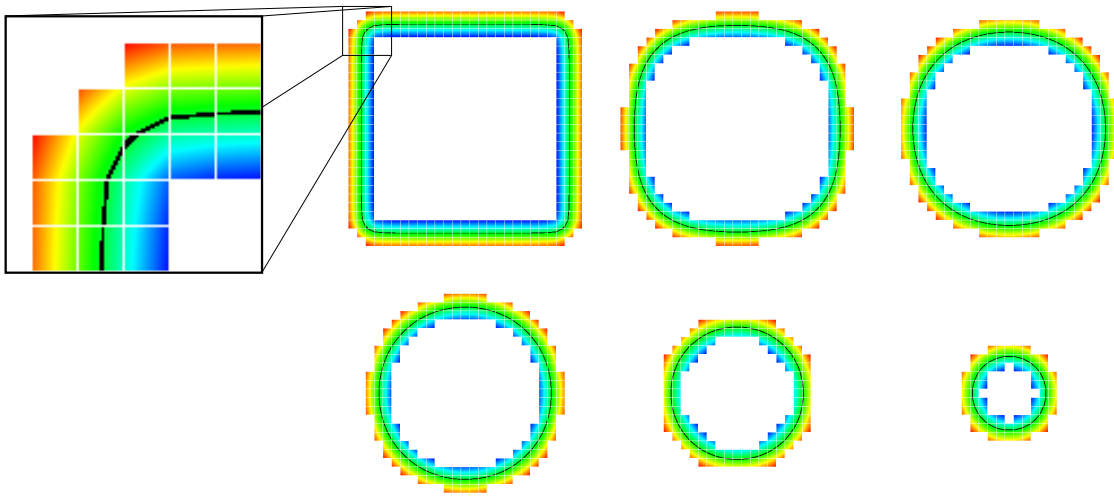


Figure 4.11: 2D mean curvature motion of a square with Q^1 finite-elements.

In Fig. 4.11 and Fig. 4.12 are shown mean curvature motion of a square in 2D, and a cube in 3D. Using Q^1 instead of P^1 improves the sub-voxelic representation of the level-set function, and corrects the bias pointed out in Fig. 4.8. The square and the cube slowly collapse, with an isotropic evolution.

In Fig. 4.13 can be seen the mean curvature motion of a cube using a finite-difference implementation with no reinitializations (top row), and our Q^1 finite-element implementation (bottom row). The finite-difference implementation does not provide intrinsic sub-voxelic representation of the distance function, but some methods exist to extract sub-voxelic iso-contours, such as the marching cube algorithm. By using finite-differences with no reinitializations, the different level of the level-set functions can be seen to get away from each others, meaning that the distance functions is not preserved during its evolution. Moreover, finite-differences are known to be less accurate than finite-elements, and we can indeed see the iso-contours extracted being less regular than the finite element-implementation. Our Q^1 finite-element implementation performs the evolution very well, by providing a good sub-voxelic representation of the solution at very low resolution, and indeed preserves the distance function during its evolution.

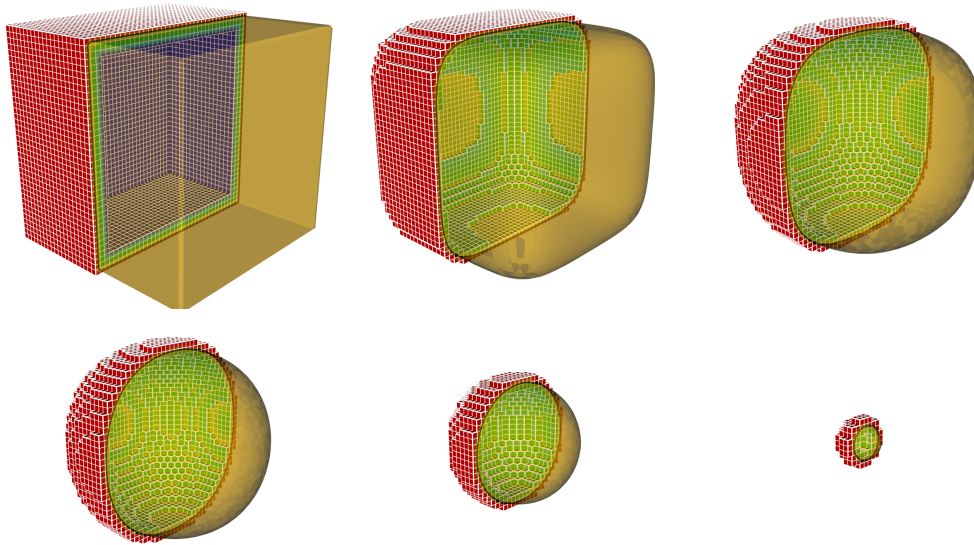


Figure 4.12: 3D mean curvature motion of a cube with Q^1 finite-elements.

In Fig. 4.14 can be seen the good preservation of the distance function during its evolution, with several level-sets staying at equal distance from each others. Without the distance constraint, the level-set of the functions would get away from each other at the corner of the cube, due to higher curvature in these areas. Moreover we can see that, at very low resolution (only a few pixels), we manage to represent very well the zero level-set of the distance function.

4.5 CONCLUSION & DISCUSSION ---

In this chapter, we presented a novel and stable implementation of the level-set method using Q^1 finite elements. This implementation has the advantage of being more accurate than finite-differences based implementation, and provides a good sub-voxelic representation of the zero level-set without the need of auxiliary iso-surface extractions methods. This implementation is optimized by restricting the domain of computation to a thin band located at the neighborhood of the zero level-set. This band is constantly updated to keep track of the front, and this allows to omit useless calculations at positions far from the zero level-set.

Thanks to the Green theorem, the differential order of the evolution equation has been decreased: the computation of the second order differential term has been transformed into first order differential terms. This decreases the error in the approximation by preventing numerical computations of second order derivatives, with all the problems brought by the discretization (ENO schemes, etc...). Moreover, this transformation allows to decrease the thickness of the narrow band. Indeed, finite difference implementations requires a computational stencil relatively large for computing sec-

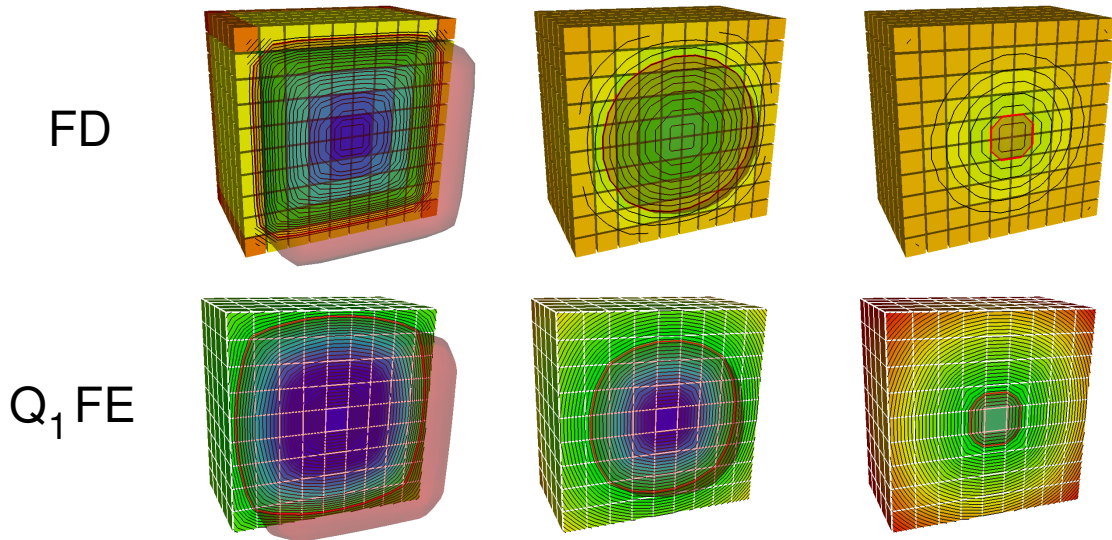


Figure 4.13: Top row: mean curvature motion of a cube with finite difference implementation; Bottom row: mean curvature motion of a cube with Q^1 finite element implementation.

ond order derivatives (stencil of 5 pixels wide), and thus influence the thickness of the narrow band. In our case, no computational stencil is needed, and all the derivatives are computed by integration inside elements. This allows to choose a much thinner band, with a minimum effective thickness of 1 element.

By using an implicit scheme instead of an explicit one, this implementation allows to choose larger time steps in the evolution, while guaranteeing the stability. Moreover, the use of an implicit scheme allows to incorporate a distance constraint in the evolution, in order to preserve the distance function of the level-set at the end of the time step. This allows to skip the auxiliary reinitialization process in the level-set evolution which may perturb the zero level-set, such as reinitialization PDE or fast marching (Section 3.4.2).

By using Q^1 elements instead of P^1 , some bias introduced by the anisotropy of the P^1 elements is corrected, and the sub-voxelic representation of the solution is improved. Moreover, this kind of element facilitate the partition of the space, and the evolving mesh embedding the zero level-set can be updated more easily, with adding new cubes in the mesh by extrapolating some values.

However, this implementation also bring some drawbacks. By using an implicit scheme, a large system of equation need to be solved at each iterations of the level-set evolution. While being done by using an efficient conjugate gradient descent, this is of course a bottleneck in the implementation compared to usual finite difference based implementation. However, while the conjugate gradient method ensures to find the solution of the system in n step, one possible solution would be to stop the

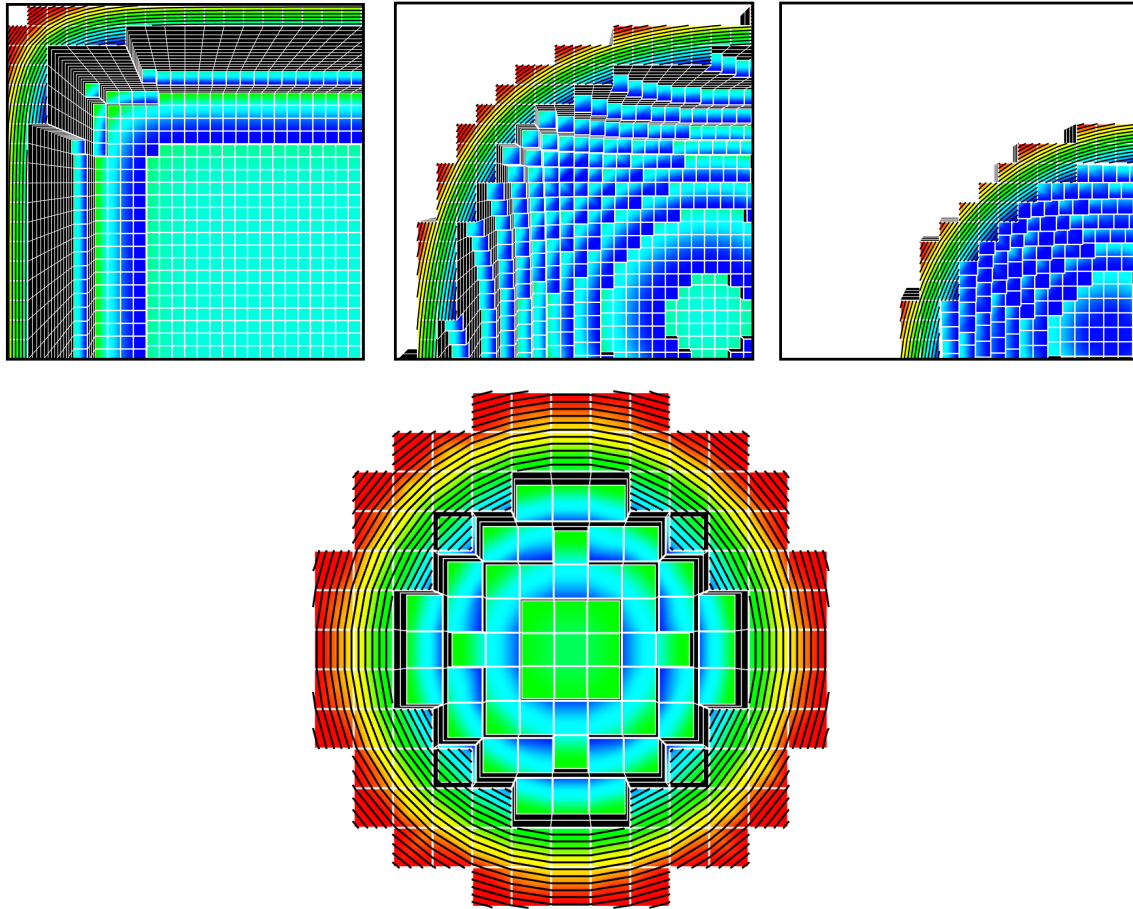


Figure 4.14: Top: 3D mean curvature motion of a cube with Q^1 finite elements; Bottom: result of the evolution at middle convergence, the distance function is well preserved.

descent before convergence, depending on some properties of the quadratic form (energy) of the system. But this introduces some new problems, such as detecting if the approximation is good enough to stop the descent.

Moreover, using an implicit scheme is a bit incoherent with having a thin narrow band ... The implicit scheme allows to choose large time steps in the evolution, while having a thin narrow band prevents from using large time steps in order to keep track of the zero level-set. This implicit scheme is only used to include the distance constraint, and the good stability of this scheme with large time step can unfortunately not be used with thin bands. Of course, one can indeed increase the thickness of the band and choose large time steps, but in this case the conjugate gradient descent computed at each iteration would be very slow, as much more equations (one per vertex) would appear in the system.

While the distance constraint allows us to skip reinitialization process of the level set, a fast marching is indeed needed at each iterations to extrapolate the new values

in the narrow band. This fast marching is less expensive than usual fast marching performed in common finite difference implementations, but requires the same initialization and data structures which are not negligible.

At last, the sub-voxel representation may not be very important, as the grid of image pixel is relatively dense, and the common marching cube method performs iso-surface extraction quite well.

This approach brings some good advantages, but is also quite difficult to implement and requires a lot of calculus in order to accurately compute a Laplacian with distance preservation. It indeed compute accurately the regularization term of the level-set, but common finite-difference methods do not actually suffer from the stability of this computation. Image segmentation is a difficult problem, and the data attachment term in the evolution of the level-set plays a much more important role than accurately computing the regularization term.

IMAGE SEGMENTATION USING LOCAL STATISTICS

Contents

Overview	72
5.1 Introduction	72
5.2 Region Segmentation using Local Statistics	75
5.2.1 Piecewise Smooth Regions and Local Means	75
5.2.2 Bayesian Formulation and Local Region Statistics	77
5.3 Varying the Neighborhood	80
5.3.1 Small Neighborhood	81
5.3.2 Intermediate Neighborhood	82
5.3.3 Large Neighborhood	84
5.4 Coherence Improvement	84
5.4.1 Coherence Problem	85
5.4.2 Coherence Solution	85
5.5 Multiscale Improvement	88
5.5.1 Homogeneous Regions Problem	89
5.5.2 Multiscale Approach	89
5.6 Efficient Implementation	91
5.6.1 Parallelized Finite-Differences based Level-Sets	91
5.6.2 Recursive Gaussian filter	91
5.6.3 Scale Selection using GPU	93
5.7 Results	94
5.8 Conclusion	96

OVERVIEW

In this chapter, we propose a fast and robust segmentation model based on local informations. It combines local statistics of the regions along the contour (inside and outside) to drive the segmentation procedure. Since it is based on local decisions, this model is more robust to local variations of the regions of interest (contrast, noise, blur, ...). In order to avoid the computational burden of a direct estimation, we express these local statistics as the result of Gaussian convolutions. This makes an efficient implementation via recursive filters possible, and gives a complexity similar to methods based on global statistics. The model generalizes several state-of-the-art segmentation models, thanks to a neighborhood parameter whose role is investigated in this chapter. This model nonetheless introduces some new difficulties which are inherent to the fact of basing a global property (the segmentation) on pure local decisions. The second part of this chapter explores some of those difficulties and propose some possible corrections. Finally, an efficient implementation is proposed, using the computational power of highly parallel devices such as modern GPUs.

Keywords: Segmentation, Bayesian formulation, Local statistics, Level-set, Mumford-Shah, GPU

5.1 INTRODUCTION

Several segmentation models based on curve evolution have been proposed, each of them bringing its own advantages and its own drawbacks (Cf. Chapter 3). Active-contours based segmentation models can be classified in two distinct categories: edge-based models and region-based ones.

Edge-based models consist in evolving a contour in homogeneous areas, and locally stopping it when it reaches high image gradients [16, 53, 114]. These models have several advantages like local coherence and robustness to region inhomogeneities, but they also have important drawbacks that make them inefficient on noisy images [26], or when the contour initialization is not completely inside or outside the region to segment. Moreover, they are only able to segment regions with rather sharp edges, and this can result in “leaks” when the region edges are too diffuse.

Region-based models consist in evolving a contour in order to separate significantly the regions it delimits [19, 108, 107, 79, 26]. For example, separating bright areas from dark ones, or extracting areas matching a certain histogram [55]. These models take account of the whole region and are thus more robust to noise and to initialization than edge-based models. However, they are inefficient in handling images with regions which statistics are spatially varying across the domain.

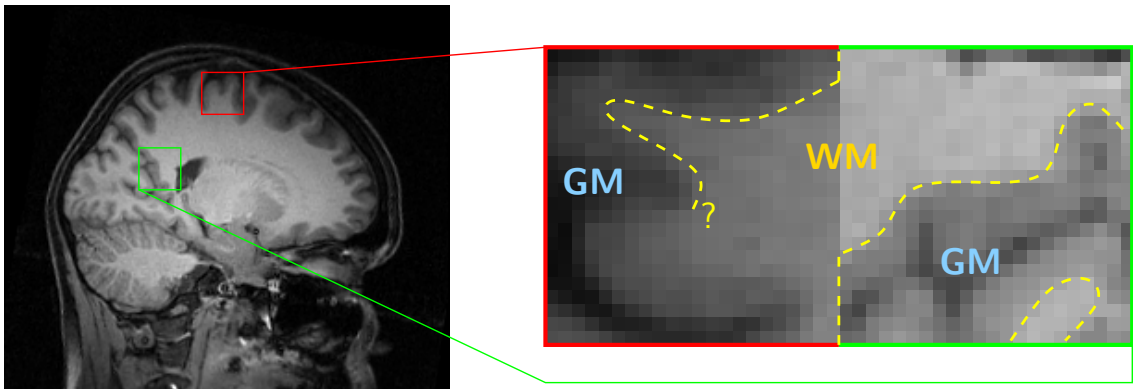


Figure 5.1: MRI difficulties: Intensity of white matter may vary across the image; and contour between white matter and grey matter may be locally diffuse.

One difficulty comes from the fact that segmentation models often relies on global criteria ¹, whereas images are most of the time subject to local variations (in intensity, contrast, noise ...). In opposition, it is admitted that the human visual system is more sensitive to local variations of the information [112], and the Adelson illusion is one of the many example showing this property (Fig. 5.2). This explains why it is often difficult to create segmentation models that are able to extract regions matching our perception.

Medical images are a typical example of image that suffer from alterations that makes usual segmentation models most of the time inefficient. For example, MRIs coming from high field MR devices suffer from biases and intensity drifts, and therefore prevent the direct use of classical region-based models. In addition, MRIs can be very noisy depending on the acquisition parameters, and the partial volume effect coming from the voxelic discretization often produce locally diffuse contours between tissues. These two alteration make classical edge-based models inefficient, as the contour may be stuck in regions with high level of noise, while some leaks may appear in the segmentation where contours are too diffuse (Fig. 5.1).

To overcome these problems, new segmentation models have recently emerged to segment regions by computing their statistics locally, in the neighborhood of the active contour [81, 2, 13, 14, 72, 57, 60, 59, 106]. All these models are somehow similar, and have been developed in parallel. These models bring several advantages like robustness to noise and inhomogeneities, but introduce some new drawbacks that are investigated hereafter.

In this chapter, we propose such a model a segmentation model based on local statistics. It consists in computing regions statistics differently for each point of the contours, inside a Gaussian window of fixed size at the neighborhood of the active

¹Either a contour strength or global distributions of pixel values inside regions.

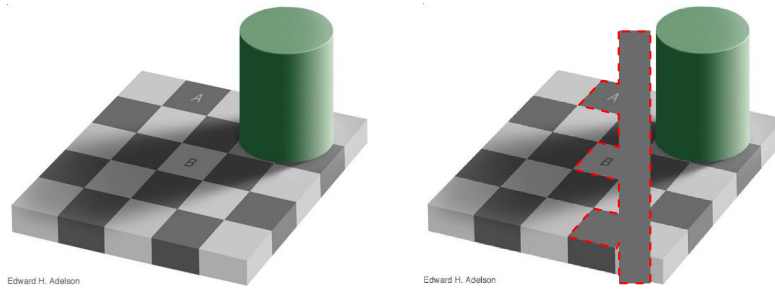


Figure 5.2: The Adelson illusion: the human visual system is more sensitive to local variations of the information than to global grey level values.

contour (Fig. 5.3). This model introduces a new parameter which corresponds to the size of the Gaussian neighborhood, and its effect on the segmentation is very interesting: when a large neighborhood is chosen, the model is equivalent to standard region-based models, whereas a small neighborhood results in a model that behaves like standard edge-based ones. Moreover, choosing an intermediate neighborhood generalizes the well known Mumford-Shah functional, as it provides segmentation for piecewise smooth regions. Thus, this model can be seen as an elegant unification of edge-based and region-based models. Experimental results show that this model is well suited for medical images, especially on those suffering from high intensity drifts.

But basing a segmentation on local decisions brings a whole new set of difficulties: locally the information might be insufficient to make the contour evolve, and it might need to be looked at different scales through a scale space approach; local decisions in neighboring regions might be contradictory, as the notions of *inside* and *outside* might be different depending on the initialization and position of the active contour. The second part of this chapter explores these problems and describes two rules that help in defining a meaningful segmentation procedure based on local contrast information.

Finally, the third part of this chapter describes an efficient implementation for computing these local statistics, based on Gaussian recursive filters, and highly parallel devices such as GPUs (Graphic Processing Units).

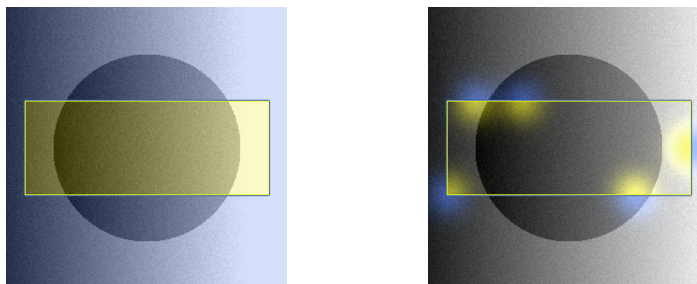


Figure 5.3: Inside (yellow) and outside (blue) statistics: left computed in the whole region; right computed at the neighborhood of the active contour.

5.2 REGION SEGMENTATION USING LOCAL STATISTICS

5.2.1 Piecewise Smooth Regions and Local Means

The proposed model can be seen as a local extension to the Chan & Vese model. Instead of minimizing the distance between regions and their corresponding mean value, we propose minimizing the distance between regions and their local means, different for each point and computed inside a Gaussian window (Fig. 5.3). These local means are expressed through Gaussian convolutions, and correspond to smoothed versions of segmented regions.

Assume that the image is covered by a set of regions Ω_i , $i = 1 \dots n$. Denoting by $\mu_i(x)$ the local mean of the region Ω_i at point x , this quantity is defined by:

$$\mu_i(x) = \frac{\int_{\Omega_i} g_\rho(x-y)I(y)dy}{\int_{\Omega_i} g_\rho(x-y)dy}, \quad (5.1)$$

where I is a given image and g_ρ a Gaussian kernel of standard deviation ρ .

The denominator of this expression is a normalization factor which is important for the points that are close to the region boundaries, when the Gaussian kernel does not overlap completely with the regions (Fig. 5.4).

Based on the Chan&Vese model, this model consists in minimizing the distance between the regions and their corresponding smooth approximation, as follows:

$$E(\Omega_1, \dots, \Omega_N) = \sum_i \int_{\Omega_i} ((I(x) - \mu_i(x))^2 dx + \nu|C|), \quad (5.2)$$

where $\mu_i(x)$ is the local mean of the region Ω_i at point x , and $|C|$ the total length of the interface between regions, which stand for a regularization term.

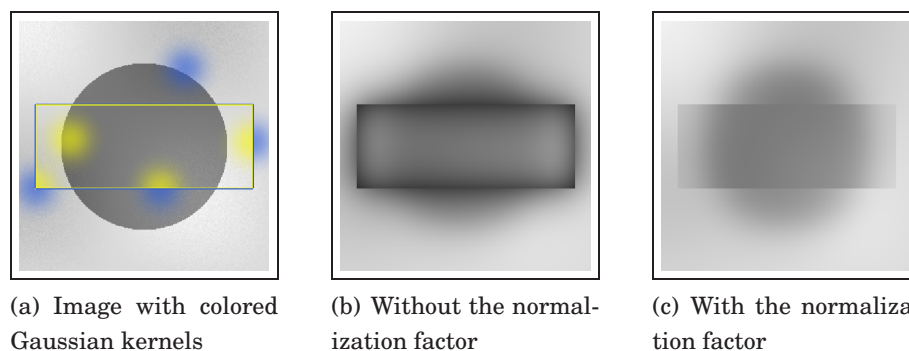


Figure 5.4: Importance of the denominator in the formulation of regions local means.

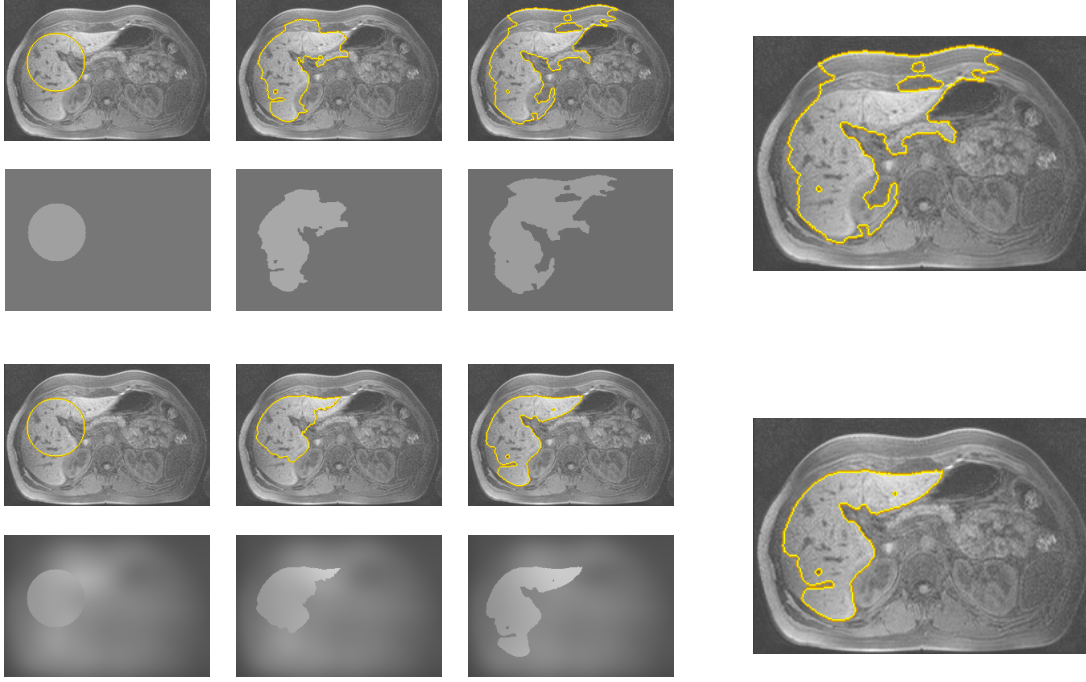


Figure 5.5: Extraction of the liver from 2D real data. Top: Global mean Chan & Vese model ; Bottom: Local means model computed with $\sigma = 16$.

The derivation of this energy can be found in Appendix B, and gives us the following gradient descent², for the bi-partitioning case in level-set notation:

$$\left\{ \begin{array}{l} \mu_{in}(x) = \frac{\int_{\Omega_{in}} g_{\rho}(x-y)I(y)dy}{\int_{\Omega_{in}} g_{\rho}(x-y)dy} \\ \mu_{out}(x) = \frac{\int_{\Omega_{out}} g_{\rho}(x-y)I(y)dy}{\int_{\Omega_{out}} g_{\rho}(x-y)dy} \\ \frac{\partial \Phi}{\partial t}(x) = (I(x) - \mu_{in}(x))^2 - (I(x) - \mu_{out}(x))^2 + \nu \nabla \cdot \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right). \end{array} \right. \quad (5.3)$$

where $\mu_{in}(x)$ and $\mu_{out}(x)$ denote respectively the local means of the inside and outside regions delineated by the zero level of the level-set function Φ .

The minimization of this energy is then performed by alternating contour evolution and local mean computation. Note that the contour is only driven by local means computed at its neighborhood, and those computed inside the region have no incidence in the evolution. This raises some problems which are discussed further.

In Fig. 5.5, comparisons between Chan & Vese model (top), and our model (bottom) is done through the segmentation of the liver from a 2D anatomical MRI. Evolution of the contour can be seen, as well as their corresponding piecewise constant segmentation for the Chan & Vese model, and piecewise smooth segmentation for the local

²Under some simplifications, see Appendix B.

mean model. As the Chan & Vese model is not robust to bias, the liver is not correctly extracted, and the top “leak” in the part of the image where intensity is close to the global mean inside the front. It finally results in separating the bright area from the dark ones, with no preference to sharp edges. Note that non-parametric Bayesian models produce the same result, as they rely on a global region model as well.

On the contrary, our model behaves quite well as the liver is correctly delimited at the local level. The model is more robust to low frequency alterations, and is well suited for this kind of medical images.

5.2.2 Bayesian Formulation and Local Region Statistics

An alternative approach is to model the problem through a Bayesian formulation. It’s a different kind of formulation for image segmentation, but this leads to the minimization of an energy similar to Eq. (5.2). In the following, we note K a partition of the image domain Ω , composed of several regions Ω_i such that

$$K = \{\Omega_1, \Omega_2, \dots, \Omega_n\} \quad \text{with} \quad \bigcup_i \Omega_i = \Omega, \quad \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j. \quad (5.4)$$

We consider $P(K|I)$, the *a posteriori* probability for having the segmentation K , given the observed image I . The optimal segmentation K is then obtained by maximizing this probability according to the associated hypothesis. Thanks to the Bayes rule, we can write:

$$P(K|I) = \frac{P(I|K)}{P(I)} P(K) \quad \longrightarrow \quad \max, \quad (5.5)$$

where:

- $P(I)$ is the probability of having an image I . This term is constant during the segmentation, and can be neglected.
- $P(K)$ is the probability of having a segmentation K . This probability stand for the desired properties of the region borders such as their smoothness, global shape, topology, etc ...
- $P(I|K)$ is the a-posteriori probability for having the image I , given the segmentation K .

We choose to use the segmentation probability as a regularizer, by minimizing the total length of the border between regions. This corresponds to maximize the following probability:

$$P(K) = \exp(-\nu |K|) \quad \longrightarrow \quad \max \quad \iff \quad \nu |K| \quad \longrightarrow \quad \min \quad (5.6)$$

where $|K|$ denote the total length of the interface, and ν the regularization parameter.

Several simplifications need to be done for decomposing the term $P(I|K)$. Assuming that there is no correlations between regions Ω_i , as well as independence between pixels inside each region, we can write,

$$\begin{aligned} P(I|K) &= P(I|\Omega_1) P(I|\Omega_2) \cdots P(I|\Omega_n) \\ &= \prod_{x \in \Omega_1} P(I(x)|\Omega_1) \prod_{x \in \Omega_2} P(I(x)|\Omega_2) \cdots \prod_{x \in \Omega_n} P(I(x)|\Omega_n) \end{aligned}$$

where $P(I(x)|\Omega_i)$ corresponds to the probability of the pixel $I(x)$ to belong to the region Ω_i .

In order to maximize the probability Eq. (5.5), we apply the negative logarithm and minimize the obtained energy using usual gradient descent. Therefore, we get

$$P(K|I) \longrightarrow \max \iff -\log(P(K|I)) \longrightarrow \min \quad (5.7)$$

with

$$-\log(P(K|I)) = \boxed{E = \sum_i \int_{\Omega_i} -\log(P(I(x)|\Omega_i)) dx + \nu|K|}. \quad (5.8)$$

The choice of the distribution $P(I(x)|\Omega_i)$ depends on the type of regions desired in the segmentation. Several distributions have been proposed³, and a classical choice is to model regions using a global Gaussian distribution so that:

$$P(I(x)|\Omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \frac{-(I(x) - \mu_i)^2}{2\sigma_i^2} \quad (5.9)$$

where μ_i and σ_i^2 are respectively the mean and the variance of the region Ω_i . This distribution generalizes the Chan&Vese model, as the energy obtained is similar when $\sigma_i^2 = \frac{1}{2}$.

In order to model regions through local statistics, we choose to use local Gaussian distributions. It takes the position into account, and assumes that each pixel in the region has a different probability, induced by its own neighborhood. This distribution is defined as:

$$P(I(x), x|\Omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i(x)} \exp \frac{-(I(x) - \mu_i(x))^2}{2\sigma_i(x)^2} \quad (5.10)$$

where $\mu_i(x)$ and $\sigma_i(x)^2$ are respectively the local mean and local variance of the region Ω_i at position x .

³See section 3.6.2 for more details.

These local statistics can be computed in a Gaussian windows around each pixel, and thus expressed as Gaussian convolutions:

$$\begin{cases} \mu_i(x) = \frac{\int_{\Omega_i} g_\rho(x-y)I(y)dy}{\int_{\Omega_i} g_\rho(x-y)dy} \\ \sigma_i(x)^2 = \frac{\int_{\Omega_i} g_\rho(x-y)(I(y) - \mu_i(x))^2 dy}{\int_{\Omega_i} g_\rho(x-y)dy}. \end{cases} \quad (5.11)$$

In summary, the probability for each pixel to belong to a given region is modeled through a *Gaussian* distribution, locally computed in a spatially *Gaussian* window.

Putting all the term together, the energy to minimize becomes,

$$E = \sum_i \int_{\Omega_i} \frac{(I(x) - \mu_i(x))^2}{2\sigma_i(x)^2} + \frac{1}{2} \log(2\pi\sigma_i(x)^2) dx + \nu|K|, \quad (5.12)$$

and its derivatives can be found in the same way as previously. This leads to the following gradient descent for the bi-partitioning case,

$$\frac{\partial \Phi}{\partial t}(x) = \frac{(I(x) - \mu_{in}(x))^2}{2\sigma_{in}(x)^2} - \frac{(I(x) - \mu_{out}(x))^2}{2\sigma_{out}(x)^2} + \log \frac{\sigma_{in}(x)^2}{\sigma_{out}(x)^2} + \nu\kappa, \quad (5.13)$$

where κ is the curvature of the front, and $\mu_{in}(x)$, $\mu_{out}(x)$ and $\sigma_{in}(x)^2$, $\sigma_{out}(x)^2$ are respectively the local means and local variance of the inside and outside regions.

This model is similar to the previous piecewise smooth model, with the addition of *the local variance*. It is useful to model the local variation of image intensity, such as local noise, or proximity to an image edge. However, this term might be problematic if the size of the Gaussian windows is too small. Indeed in homogeneous areas, the local variance has to be close to zero, and therefore dividing by this term in the evolution may produce some numerical instabilities.

In Fig. 5.6 can be seen a template for computing these local statistics at each point of the contour. The local means are first computed by blurring each region separately, and then normalized by a blurred version of regions characteristic functions. Local variances are then computed in the same way using the previously computed local means. See section 5.6 for more details about the implementation.

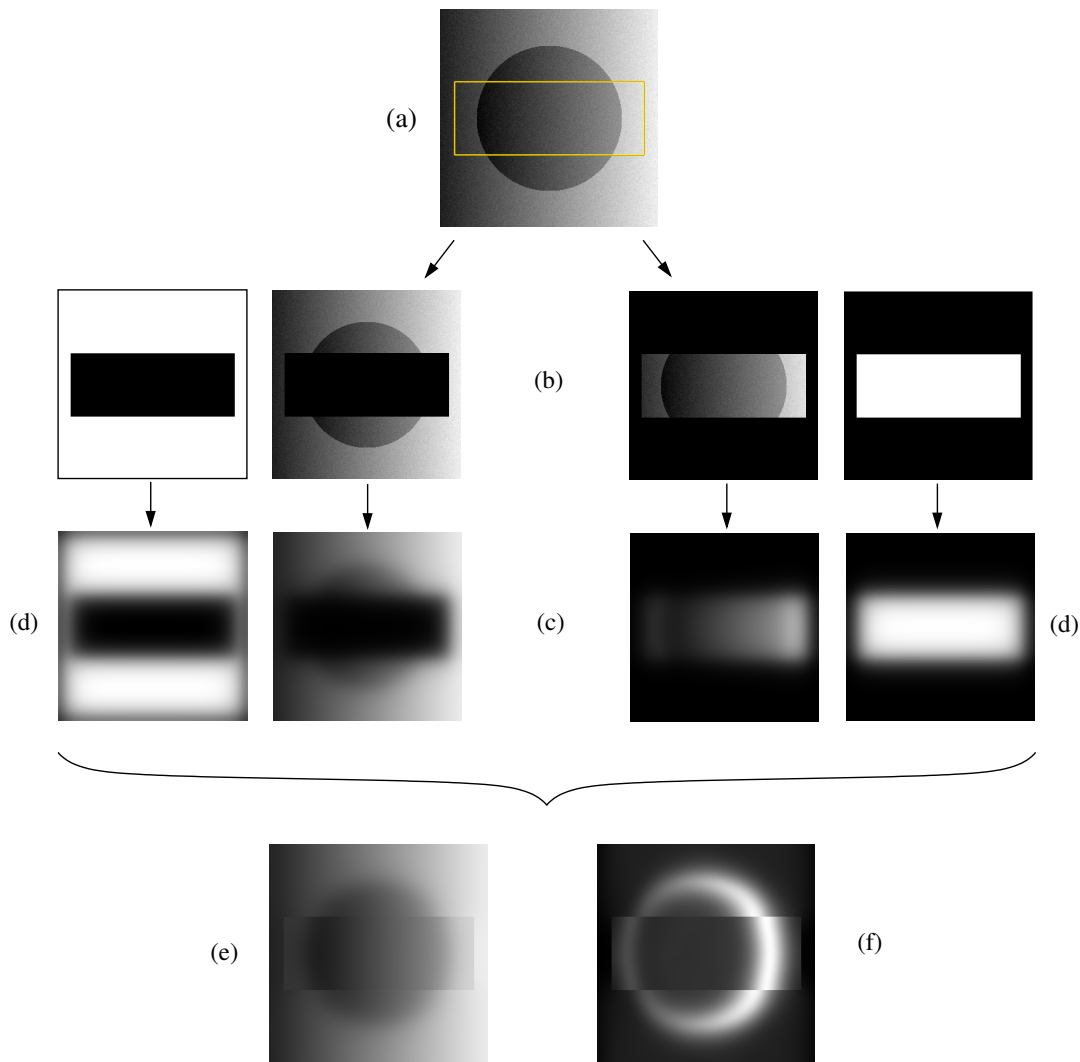


Figure 5.6: Local statistics computation: The regions delineated by the yellow contour (a) are first separated (b), blurred independently (c), and then normalized by a blurred characteristic function of the regions (d). This gives the local mean (e) and the local variance (f) using in this example with a Gaussian kernel of variance 16.

5.3 VARYING THE NEIGHBORHOOD ---

Varying the size of the Gaussian neighborhood impacts significantly the behavior of the curve evolution. However, several generalizations of other segmentation models can be made. 3 cases are investigated :

- **small neighborhood**: the variance of the Gaussian window is close to zero.
- **intermediate neighborhood**: any Gaussian window of medium size.
- **large neighborhood**: the variance of the Gaussian window tends to infinity.

5.3.1 Small Neighborhood

Since the size of the Gaussian window is close to zero, the local variance has to be zero, and this is thus problematic. Therefore, we consider in the following the piecewise smooth model, without the local variance.

We have, for the bi-partitioning case of two regions Ω_{in} and Ω_{out} ,

$$\frac{\partial \Phi}{\partial t} = (I(x) - \mu_{in}(x))^2 - (I(x) - \mu_{out}(x))^2 + \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right), \quad (5.14)$$

with

$$\mu_i(x) = \frac{\int_{\Omega_i} g_\sigma(x-y) I(y) dy}{\int_{\Omega_i} g_\sigma(x-y) dy} \quad \text{and} \quad g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-x^2}{2\sigma^2}. \quad (5.15)$$

When $\sigma \rightarrow 0$, we have:

$$\lim_{\sigma \rightarrow 0} g_\sigma(x) = \delta(x), \quad (5.16)$$

where $\delta(x)$ denote the Dirac function.

Therefore, for a point x lying on the boundary of the open subspace Ω_i ,

$$\begin{aligned} \lim_{\sigma \rightarrow 0} \mu_i(x) &= \frac{\int_{\Omega_i} \delta(x-y) I(y) dy}{\int_{\Omega_i} \delta(x-y) dy} \\ &= \lim_{y \rightarrow x} I(y), \quad y \in \Omega_i \\ &= \lim_{\alpha \rightarrow 0} I(x + \alpha \vec{n}), \end{aligned}$$

where \vec{n} denote the normal to Ω_i , oriented toward the interior.

Therefore, as the regions Ω_{in} and Ω_{out} are complementary, we get:

$$\begin{aligned} \lim_{\sigma \rightarrow 0} \frac{\partial \Phi}{\partial t}(x) &= \lim_{\sigma \rightarrow 0} \left((I(x) - \mu_{in}(x))^2 - (I(x) - \mu_{out}(x))^2 \right) + \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \\ &= \lim_{\alpha \rightarrow 0} \left((I(x) - I(x - \alpha \vec{n}))^2 - (I(x) - I(x + \alpha \vec{n}))^2 \right) + \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \\ &= (\Delta_{\vec{n}} I)^2 - (\nabla_{\vec{n}} I)^2 + \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \end{aligned}$$

where \vec{n} is the normal to the region Ω_{in} , and $\Delta_{\vec{n}}$ and $\nabla_{\vec{n}}$ denotes respectively the backward and forward difference derivative in the normal direction \vec{n} .

In continuous areas, both of the backward and forward derivatives are similar, therefore the level-set function Φ will only evolve under a mean curvature motion. If we are infinitesimally close to a discontinuity in the normal direction, one of the two derivatives has to be very large, and the other one can then be neglected.

This leads to

$$\frac{\partial \Phi}{\partial t}(x) = \begin{cases} \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) & x \in \text{continuous areas} \\ \text{fmax}((\nabla_{\vec{n}} I)^2, -(\Delta_{\vec{n}} I)^2) + \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) & x \in \text{discontinuity} \end{cases}$$

where 'fmax' denotes the maximum of the absolute value.

Finally, by taking only one derivative in presence of discontinuity, the directional derivative can be rewritten in term of scalar product:

$$\frac{\partial \Phi}{\partial t} = \begin{cases} \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) & x \in \text{continuous areas} \\ \langle \vec{n}, \nabla I \rangle^2 + \nu \nabla \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) & x \in \text{discontinuity} \end{cases}$$

where ∇I is the image gradient.

If we assume that image edges correspond to discontinuities, the behavior of the model when $\sigma \rightarrow 0$ is equivalent to standard edge-based models.

Indeed, the term appearing in presence of a discontinuity is strongly similar to the advection term coming from the *geodesic active contour*, or the *gradient vector flow* close to image edges: the curve is "attracted" by the areas where the image gradient is very strong, and then remains stuck in the "discontinuity", oscillating around the image edge.

Therefore, with very small Gaussian neighborhoods, the piecewise smooth model is somehow equivalent to an edge-based model: there are no data attachment term contribution in homogeneous areas⁴, and the contour remains stuck at image edge positions.

5.3.2 Intermediate Neighborhood

Choosing an intermediate neighborhood allows the segmentation of piecewise smooth regions. However, one famous model has been defined for segmenting piecewise smooth region : the *Mumford-Shah model* [74]. This model aims at finding a piecewise smooth approximation of a given image by minimizing the given functional:

$$E(\mu, C) = \int_{\Omega} (I - \mu)^2 dx + \lambda \int_{\Omega \setminus C} |\nabla \mu|^2 dx + \nu |C| \quad (5.17)$$

where μ is the piecewise smooth function, C the interface between smooth regions, I the original image, and λ and ν are constant weighting parameters. The minimization of this energy gives the piecewise smooth approximation u , as well as the interface Γ between regions.

⁴One may add an auxiliary balloon force to make the contour evolve in homogeneous areas

As depicted in chapter 3, the minimization of this functional can be done using an active contour framework thanks to the level-set method. However, this minimization is computationally expensive, since a Poisson PDE needs to be solved for each region at each iteration of the contour evolution.

In [13], the authors highlighted a link between the piecewise smooth model (sect. 5.2.1), and the Mumford Shah functional. The main idea is to express the piecewise smooth regions through **the minimization of an energy** instead of **the result of a convolution**. Based on the work of [75, 76, 116], they found the following link:

$$\mu_i(x) = \frac{\int_{\Omega_i} g_\lambda(x-y)I(y) dy}{\int_{\Omega_i} g_\lambda(x-y) dy} \quad \text{with} \quad g_\lambda(x) = \exp(-\lambda x^2)$$

$$\Downarrow$$

$$E(\mu_i) = \int_{\Omega_i} \left((I(x) - \mu_i(x))^2 + \sum_{k=1}^{\infty} \frac{\lambda^k}{k!} \sum_{j_1+j_2=k} \frac{d^k \mu_i}{\partial x^{j_1} \partial y^{j_2}}(x) \right) dx \longrightarrow \min .$$

By “inserting” this energy into the piecewise smooth energy, it can be rewritten as:

$$E(K) = \sum_i \int_{\Omega_i} ((I(x) - \mu_i(x))^2 dx + \nu|C|$$

$$\Downarrow$$

$$E(\mu, K) = \sum_i \int_{\Omega_i} \left((I(x) - \mu_i(x))^2 + \sum_{k=1}^{\infty} \frac{\lambda^k}{k!} \sum_{j_1+j_2=k} \frac{d^k \mu_i}{\partial x^{j_1} \partial y^{j_2}}(x) \right) dx + \nu|C|$$

where $K = \{\Omega_1, \dots, \Omega_N\}$ and $\mu = \{\mu_1, \dots, \mu_N\}$.

Finally, neglecting the penalizers of order $k > 1$ which have a lower incidence, yields

$$E(\mu, K) = \sum_i \int_{\Omega_i} \left((I(x) - \mu_i(x))^2 + \lambda |\nabla \mu_i|^2 \right) dx + \nu|C|$$

which states exactly the Mumford Shah functional.

Therefore, with intermediate Gaussian neighborhood, the piecewise smooth model is somehow equivalent to Mumford Shah functional: both energies are similar under some simplifications, and the piecewise model greatly improves the minimization process by filtering each region with an efficient Gaussian filter scheme (see section 5.6) instead of solving a Poisson PDE for each region and at each iteration.

5.3.3 Large Neighborhood

Using a large neighborhood makes the model behave like usual global regions models. There are no more locality appearing in region statistics, and each point of the contour shares a unique global distribution for each region of the segmentation. Using a Gaussian distribution generalizes the Chan&Vese model, and any global distributions can be used to represent regions.

However, using large neighborhoods makes the model behave like global models, and it thus loses its robustness to inhomogeneities, and its good locality properties.

5.4 COHERENCE IMPROVEMENT

Unfortunately, the local models presented in the previous sections introduce new difficulties. One of these difficulties comes from a problem of *coherence* in the segmentation: as there is no more one single distribution for a given region, the notion of regions is not well defined. Distant points of the contour evolve rather independently, and there is no more coherence in the segmentation of regions. In the following, we focus on the Bayesian model using the local Gaussian distribution (with the local variance).

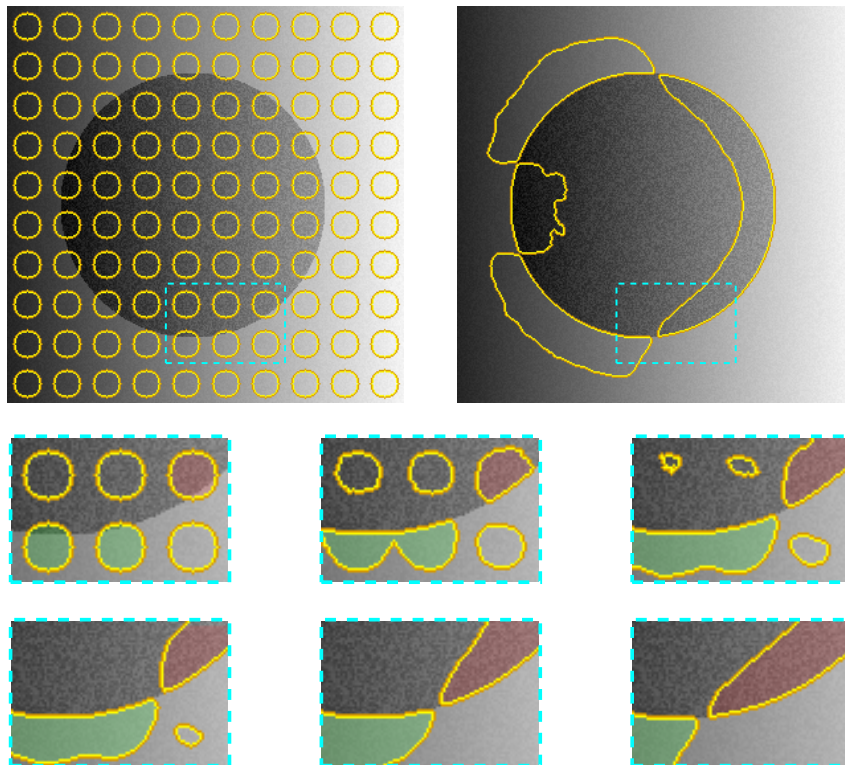


Figure 5.7: Top: Initialization and convergence of the local model ; Bottom: evolution of the contour which lead to the creation of a singularity.

5.4.1 Coherence Problem

For each point of the contour, a local distribution for the inside region and a local distribution for the outside region is estimated. The segmentation model aims at separating these two distributions, and thus distant parts of the contour evolve independently. Later in the evolution, these incoherences lead to the creation of “singularities” corresponding to unwanted local stable minima. When a singularity is formed, no topological change is possible, and the segmentation cannot be turned into something coherent anymore. This is illustrated in Fig. 5.7: the red circle segments the interior of the dark object, while the green circles segment its exterior according to their initialization. No topological change are possible between these two growing regions, and a singularity is indeed created. Other circles collapse under the effect of regularization, because of the absence of contribution due to the data attachment term in homogeneous areas. Note that this problem also occur with simpler initializations, as soon as the contour crosses the edges of the object to segment (Fig. 5.8).

5.4.2 Coherence Solution

A lot of works have been done to avoid the kind of local minima depicted in Fig. 5.7. In [48], the authors added some noise in the evolution to perturb the active contour, and thus to prevent it from being stuck in local minimas. Unfortunately, this stochastic evolution is not well adapted to our problem, as it would create as much incoherence in the evolution.

In [21], [101], the authors smoothed the evolution term of the active contour along the contour, in order to add some correlation in the evolution of neighbor points. This allows the contour to evolve in a more coherent way, avoiding local minima. Unfortunately, the local minima we tackle cannot be avoided in this way, as the evolution can be seen as coherent until a singularity is created.

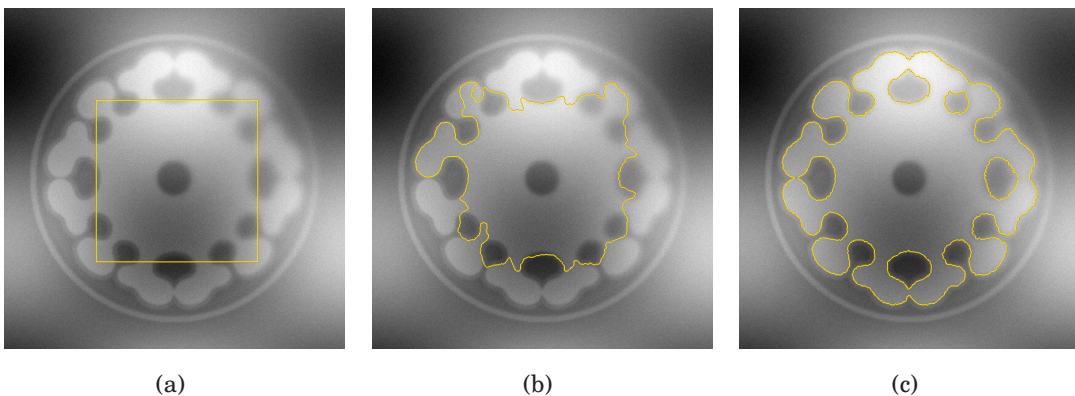


Figure 5.8: (a) Initialization; (b) Convergence without the coherence constraint: the segmentation is “locally correct” but not globally; (c) Convergence with the coherence constraint: intensity change in the same way from the inside to the outside.

However, if we make the assumption that the objects to segment are characterized by boundaries where image intensity changes “in the same way” from the inside to the outside⁵, we can use this constraint to maintain the coherence in the segmentation (Fig. 5.8.c). This constraint can be defined by having, for each point x of the contour, the local interior mean $\mu_{in}(x)$ either inferior or superior to the local exterior mean $\mu_{out}(x)$. Thus, if the contour is locally in contradiction to this constraint, the evolution direction needs to be changed.

This can be explained by looking at the energy of the model. Actually, the final segmentation corresponds to a local minimum of the energy, which strongly depends on the initialization of the contour. If the initial contour is close to a local minimum, then the gradient descent will minimize the energy by making the contour converge toward this minimum. However, if this local minimum is not the one desired, one should instead try to go away from it, in order to find another one. In our case, these local minima are characterized by the local means of the inside and outside regions for all points of the contour.

Fig. 5.9 shows the energy restricted to a point x of the contour Γ . According to its initialization, the contour must evolve in order to minimize the energy by separating the two local regions with $\mu_{in}(x) > \mu_{out}(x)$ (red path). However, if we impose the constraint of having $\mu_{in}(x) < \mu_{out}(x)$, then the contour has to evolve in the opposite direction, thus increasing the energy. When the constraint of having $\mu_{in}(x) < \mu_{out}(x)$ is indeed satisfied, the gradient descent start to minimize the energy by making the contour converge toward the desired local minimum (green path). This behavior can be seen as a local simulated annealing, as the energy is locally increased in order to avoid an undesired local minimum.

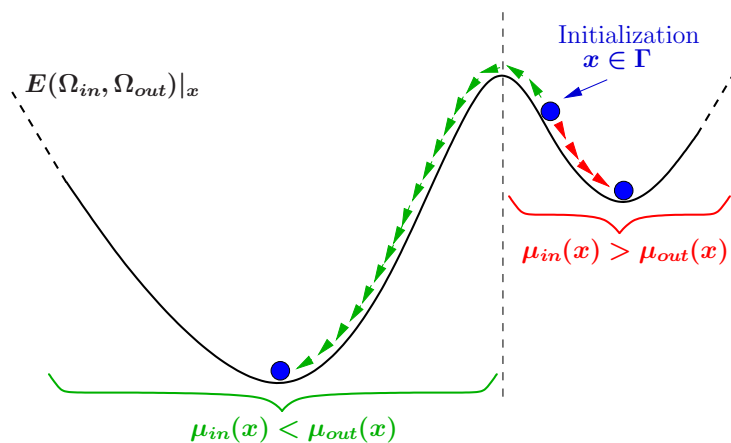


Figure 5.9: Simulated annealing: the energy restricted to a point of the contour is locally increased in order to converge toward the desired local minimum.

⁵The image gradient is globally oriented either toward the inside or the outside of the object, for all the points lying on its boundary.

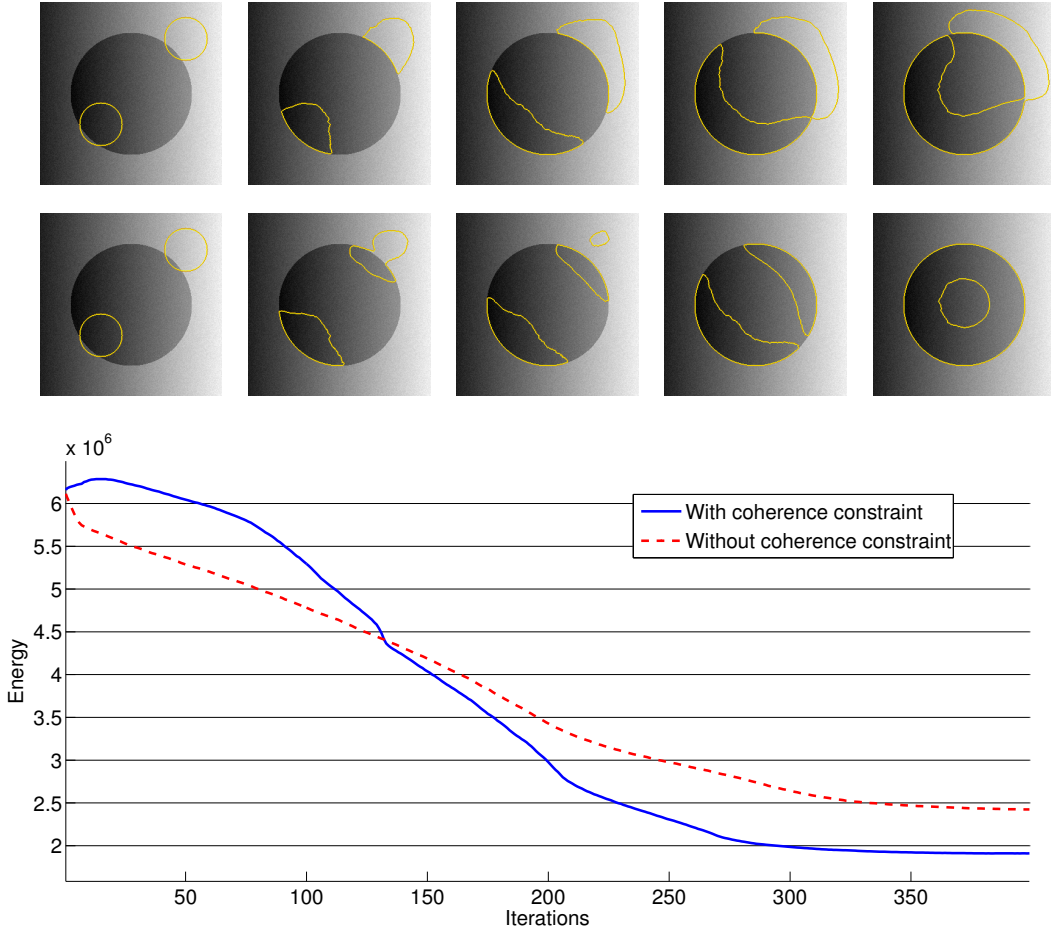


Figure 5.10: Top row: evolution without the coherence constraint; Middle row: evolution with the coherence constraint; Bottom: Energy of the two evolutions.

Therefore, with the incorporation of this behavior, the evolution equation becomes for the bi-partitioning case:

$$\left\{ \begin{array}{l} \frac{\partial \Phi}{\partial t}(x) = \mathcal{S}(x) \left(\frac{(I(x) - \mu_{in}(x))^2}{2\sigma_{in}(x)^2} - \frac{(I(x) - \mu_{out}(x))^2}{2\sigma_{out}(x)^2} + \log \frac{\sigma_{in}(x)^2}{\sigma_{out}(x)^2} \right) + \nu \kappa \\ \mathcal{S}(x) = \pm \text{sign}(\mu_{in}(x) - \mu_{out}(x)) \end{array} \right.$$

where '±' depends on the type of the global constraint. The energy is therefore increased or decreased, depending on the sign of $\mathcal{S}(x)$. Note that this simulated annealing can be used in a similar approach with multiple regions, by increasing the energy if the local features of the contour separating two different regions are not the ones desired.

In Fig. 5.10, the convergence of the active contour can be seen with and without this coherence constraint, on a synthetic image with high intensity drifts. Without

the constraint (Fig. 5.10, top row), the segmentation obtained is "locally correct", as it locally separates the regions of the interior and exterior of the contour. However, it has no sense if we look at the global scope, and this kind of segmentation is of course not acceptable. With the constraint of having the inside region locally darker than the outside region (Fig. 5.10, middle row), the contour finally manages to segment the whole grey circle, even under strong inhomogeneities. By looking at the energy of the evolution with coherence constraint (Fig. 5.10, bottom), the simulated annealing slightly increases the energy at the beginning, and that's why the evolution of the top-right circle is somehow not natural. However, the energy starts decreasing when the evolution become more natural, and finally reaches a minimum of lower energy than without the coherence constraint.

Absence of evolution in locally homogeneous areas is natural since the local statistics of the inside and outside regions are similar. This is another problem introduced by the use of local statistics, and a solution to this problem is proposed in the next section.

5.5 MULTISCALE IMPROVEMENT ---

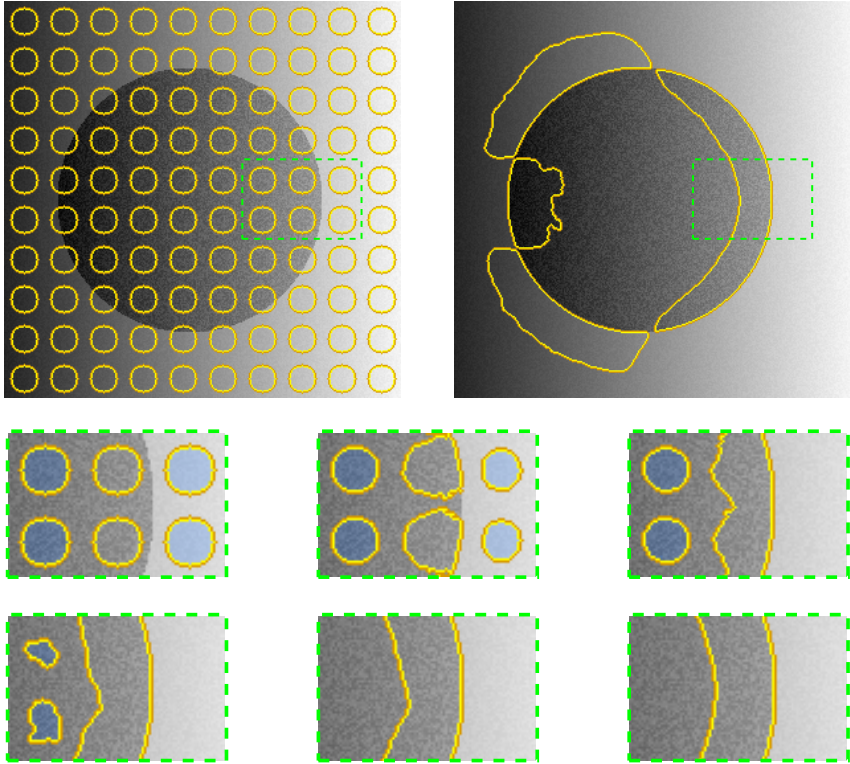


Figure 5.11: Top: Initialization and convergence of the local model ; Bottom: absence of evolution in locally homogeneous areas which leads to the creation of "layers".

5.5.1 Homogeneous Regions Problem

It is easy to understand that there is no data term contribution in homogeneous areas, as the local statistics in the region inside and outside are the same (see Eq. (5.13)).

At convergence, the small circles in homogeneous areas collapse because of the regularization term, which leads to some kind of phantom contour, resulting in “segmentation layers” that are of course not desired (Fig. 5.11.bottom). These layers come from the fact that the only contribution of the data attachment term is close to image edges, where the difference of local region statistics make the contour evolve.

However, it can be observed that the “thickness” of these segmentation layers depends on the size of the Gaussian windows of the model. With a small Gaussian windows, the layers will be thin, whereas a large windows will result in a wider layer. Unfortunately, increasing the size of the Gaussian windows makes the model behave like global models, and it thus loses its good locality properties.

In Fig. 5.11, the left blue circles collapse because they are too far from an image edge. This edge proximity is linked to the neighborhood parameter of the model, and a larger neighborhood would have indeed made these circles evolve. Note that the right circles are collapsing under the effect of the data attachment term, as the “middle layer” influence their local statistics.

5.5.2 Multiscale Approach

As mentioned previously, the thickness of the “segmentation layers” is correlated with the neighborhood parameter of the model, but using large neighborhoods give results equivalent to standard global models (section 5.3.3).

Thus, we would like to use small neighborhoods close to an image edge, and large neighborhoods in homogeneous areas, depending on the proximity of the closest image edge. The question is how to detect this edge proximity.

In [2], the authors introduced a scale selection framework in a local statistics model by using two given scales of local statistics, and combining these scales differently depending on the proximity to an image edge. To detect this proximity, they used a simple edge detector similarly to edges based models. This approach is interesting, however using an edge detector may not give accurate scales for locally diffuse contours.

In [62], the authors tackled similar problems with the purpose of extracting edges and ridges from images. Their method is based on detecting features at various scales of the same image, and keeping the k most salient features depending on a saliency measure. It allows them to extract sharp edges as well as smooth ones by increasing the locality of the estimations of these features.

Our problem is somehow similar. If we assume that the neighborhood parameter is actually a scale used to compute local statistics, the problem becomes the one of finding the most salient scale for each point of the contour to make the con-

tour locally evolve, even in locally homogeneous areas. Actually, the local model of the previous sections is not restricted to have a unique size of Gaussian window for all points, and we can indeed use local window of different size depending on local features. As we want to favor sharp edges over smooth ones, and make the contour evolve in locally homogeneous areas, the scale is chosen as the smallest one inducing an evolution speed superior to a given threshold :

$$\boxed{\rho(x) = \inf_{\rho} \left(\frac{\partial \Phi}{\partial t} \Big|_{\rho} (x) > \epsilon \right) \quad x \in \Gamma} , \quad (5.18)$$

where $\frac{\partial \Phi}{\partial t} \Big|_{\rho}$ is the evolution equation computed with a Gaussian windows of variance ρ , x a point of the contour Γ , and ϵ a constant corresponding to a threshold for the scale selection.

By choosing this scale selection scheme, the contour will evolve even in homogeneous areas. Indeed, at each iteration and for each point, the optimal scale is chosen by slowly increasing the neighborhood from the minimum scale to the maximum, until it reaches some edge of an object, thus giving information about how the contour should locally evolve.

Unfortunately, we removed a parameter (the windows size) by introducing another parameter (the scale selection threshold). However the speed of the level-set is somehow normalized by the local variance Eq. (5.13), and therefore the range of the speed induced by the model is independent of the nature of the images to segment. Experimentations have shown that by fixing this threshold to '1', the behavior of the model is correct, and all the points of the contour evolve even in homogeneous areas.

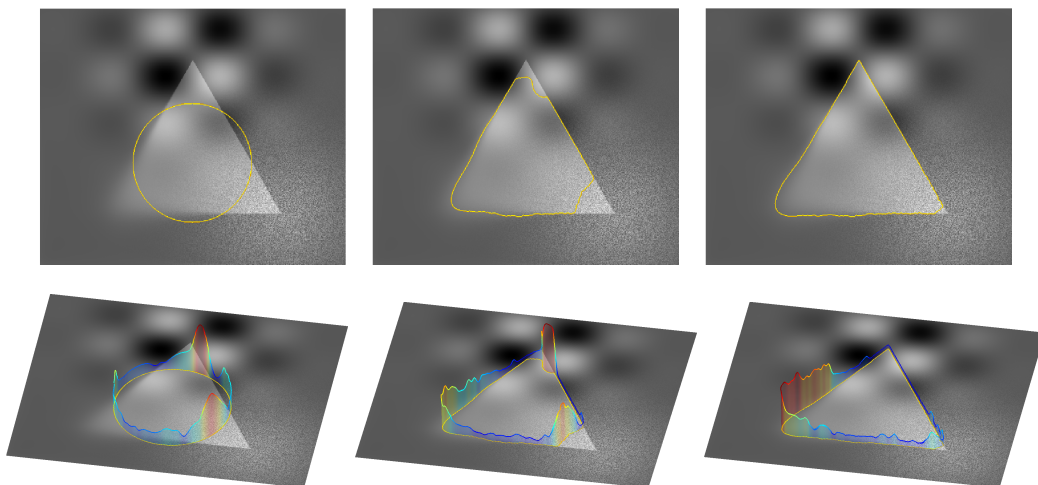


Figure 5.12: Top: initialization and evolution of the contour; Bottom: Scales chosen at each point of the contour, with $\epsilon = 1$ and using 30 scales.

Figure 5.12 shows the evolution of the active contour (top row), as well as the different scales chosen at each point (bottom row). This image is interesting as it shows three fundamental problem in medical image : large intensity drifts (top corner of the triangle), noisy regions (right corner of the triangle) and smooth contours (left corner of the triangle). Where standard edge-based or region-based model would fail to provide good results, the multiscale local model behave quite well, choosing accurately the scales depending on the proximity and the "smoothness" of the triangle edges. The robustness of the model to noisy regions comes from the fact that the speed (Eq. (5.13)) is normalized by the local variance, so the noisier the region is, the smaller the speed is, and the higher the scale chosen will be. This allows to choose larger scales if the contour is in a noisy regions, thus improving the accuracy of computation of local statistics.

5.6 EFFICIENT IMPLEMENTATION ---

5.6.1 Parallelized Finite-Differences based Level-Sets

These local-statistics models are implemented using a finite-difference based level-set method with a narrow band approach. This choice is driven by the fact that finite-difference based numerical schemes are very easy to implement with acceptable accuracy, and are highly parallelizable. Indeed, we parallelized this method using the **OpenMp library**⁶. This library allows us to efficiently distribute the speed computation for each point of the narrow band on the different cores of modern multicore CPUs. The reinitialization of the level-set is performed at each iteration using a fast-marching method, optimized to have a complexity in $O(n)$ (section 3.4.2).

5.6.2 Recursive Gaussian filter

All the local statistics used in the model are Gaussian convolutions, and can be computed very efficiently using *recursive Gaussian filters*.

Several recursive filters have been proposed to perform Gaussian convolutions [105]. We chose the Deriche recursive filter [29] for its efficiency and highly parallelizable implementation. This filter is based on approximating the causal part (i.e. $x \geq 0$) of the Gaussian function by a function of the form:

$$h_{\sigma}^{+}(x) = k \left\{ a_0 \cos\left(\frac{w_0}{\sigma}x\right) + a_1 \sin\left(\frac{w_0}{\sigma}x\right) \right\} \exp\left(-\frac{b_0}{\sigma}x\right) + k \left\{ c_0 \cos\left(\frac{w_1}{\sigma}x\right) + c_1 \sin\left(\frac{w_1}{\sigma}x\right) \right\} \exp\left(-\frac{b_1}{\sigma}x\right)$$

where k , a_0 , a_1 , b_0 , b_1 , c_0 , c_1 , w_0 , w_1 are constants depending on the Gaussian parameter. The same approximation is done for the anti-causal part (i.e. $x < 0$).

⁶www.openmp.org

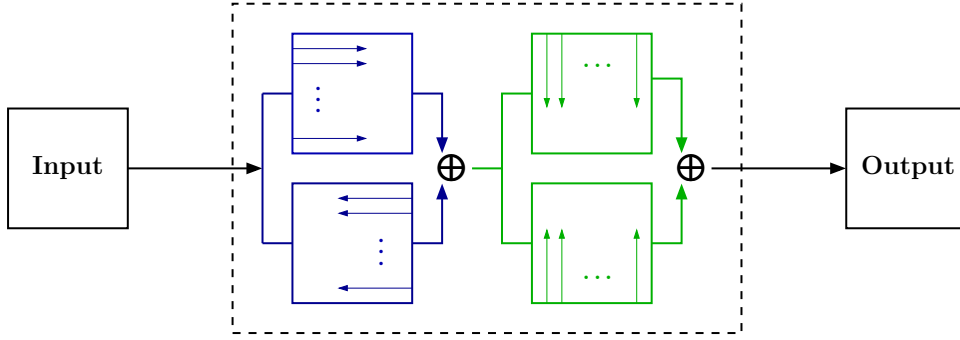


Figure 5.13: Two pass filtering of a 2D image.

The z -transform of these functions is known to be an exact ratio of polynomials, which leads to the following notation for the causal part with a third order approximation:

$$H_{\sigma}^{+}(z) = \frac{n_0^{+} + n_1^{+}z^{-1} + n_2^{+}z^{-2}}{1 + d_1^{+}z^{-1} + d_2^{+}z^{-2} + d_3^{+}z^{-3}} \quad , \quad (5.19)$$

where the coefficients n_i^{+} and d_i^{+} are functions of the parameter k , a_0 , a_1 , b_0 , b_1 , c_0 , c_1 , w_0 , w_1 . The same transformation is done for the anti-causal part.

The convolution of an input $f(x)$ by a kernel $h(x)$ in the spatial domain is written as a multiplication of their z -transform in the z -domain [29]. Therefore, using the previous z -transform of the causal and anti-causal parts of the Gaussian approximation leads to the following recursive equations:

$$\begin{cases} g^{+}(x) = n_0^{+}f(x) + n_1^{+}f(x-1) + n_2^{+}f(x-2) \\ \quad -d_1^{+}g^{+}(x-1) - d_2^{+}g^{+}(x-2) - d_3^{+}g^{+}(x-3) \\ g^{-}(x) = n_0^{-}f(x) + n_1^{-}f(x+1) + n_2^{-}f(x+2) \\ \quad -d_1^{-}g^{-}(x+1) - d_2^{-}g^{-}(x+2) - d_3^{-}g^{-}(x+3) \end{cases} \quad . \quad (5.20)$$

where $g^{+}(x)$ corresponds to the result of the left-to-right recursion, and $g^{-}(x)$ to the result of the right-to-left recursion. The complete result of the filtering $g(x)$ is the sum of the two recursions:

$$g(x) = g^{+}(x) + g^{-}(x) \quad .$$

Due to the separability of the Gaussian function, an image can be filtered by a 2D Gaussian kernel in 2 pass, by first filtering all the rows separately, then filtering all the column separately (Fig. 5.13). This lead to a computational complexity in $O(n)$, where n is the number of pixels.

In addition, this scheme is highly parallelizable since each of the rows and columns of the image can be filtered independently, with for each of them, the

backward and forward recursion being done simultaneously. On highly parallel devices, this can lead to significant gains in speed.

Let's see how many convolutions are needed to compute the local statistics for the bi-partitioning case. By noting $(\cdot * \cdot) |_{\Omega}$ the convolution operator on a domain Ω we have,

- The **local mean** is actually a ratio of convolutions:

$$\mu_i(x) = \frac{\int_{\Omega_i} g_{\rho}(x-y)I(y)dy}{\int_{\Omega_i} g_{\rho}(x-y)dy} = \boxed{\frac{(g_{\rho} * I) |_{\Omega}(x)}{(g_{\rho} * 1) |_{\Omega}(x)}}$$

where 2 convolutions are needed: the numerator corresponds to a filtered image of the region Ω , and the denominator to a blurred characteristic function of Ω .

- The **local variance** can be computed using the previous local mean:

$$\begin{aligned} \sigma_i(x) &= \frac{\int_{\Omega_i} g_{\rho}(x-y) (I(y) - \mu_i(x))^2 dy}{\int_{\Omega_i} g_{\rho}(x-y)dy} \\ &= \frac{\int_{\Omega_i} g_{\rho}(x-y)I(y)^2 dy}{\int_{\Omega_i} g_{\rho}(x-y)dy} - 2\mu_i(x) \left(\frac{\int_{\Omega_i} g_{\rho}(x-y)I(y)dy}{\int_{\Omega_i} g_{\rho}(x-y)dy} \right) + \mu_i(x)^2 \\ &= \frac{\int_{\Omega_i} g_{\rho}(x-y)I(y)^2 dy}{\int_{\Omega_i} g_{\rho}(x-y)dy} - \mu_i(x)^2 \\ &= \boxed{\frac{(g_{\rho} * I^2) |_{\Omega}(x)}{(g_{\rho} * 1) |_{\Omega}(x)} - \mu_i(x)^2} \end{aligned}$$

Only one additional convolution is needed which corresponds to a filtered squared image of the region Ω .

Therefore, the total number of convolutions needed for computing the local statistics of a given region is 3. For the bi-partitioning case, the total number of convolutions needed is 6. However, an extra convolution can be saved by expressing the blurred characteristic function of the complementary region $(g_{\rho} * 1) |_{\bar{\Omega}}$ by $1 - (g_{\rho} * 1) |_{\Omega}$, but special border conditions are needed in this case.

5.6.3 Scale Selection using GPU

With the multiscale approach, various scales are incorporated in our model, so each region has to be blurred several times using different Gaussian kernels to obtain a scale space of local statistics. This allows to retrieve, for each point of the contour, the optimal scale using the criteria of Eq. (5.18).

For efficiency purposes, as the local statistics need to be recomputed at each iteration, the scale space is limited to only a few scales. In the experiments, the

Algorithm 2 Segmentation using adaptive local statistics

```
repeat
  for all scales  $\rho_j \in \{\rho_1, \dots, \rho_n\}$  do
    for all regions  $\Omega_i \in \{\Omega_1, \dots, \Omega_n\}$  do
      Compute local means  $\mu_{i,j}$  of  $\Omega_i$  at scale  $\rho_j$ 
      Compute local variance  $\sigma_{i,j}^2$  of  $\Omega_i$  at scale  $\rho_j$ 
    for all points in the active contour do
      Find optimal scale  $\rho_{opt}$  using Eq. (5.18) with  $\mu_{i,j}$  and  $\sigma_{i,j}^2$ 
      Compute speed with  $\mu_{i,opt}$  and  $\sigma_{i,opt}^2$ 
    Update contour
until convergence
```

small scales are more important, so we decided to choose four scales: 2, 4, 8, 16.

The model is described by the following algorithm:

Finding the correct scale from the set of local statistics is very fast, and the bottleneck remains the creation of the scale space of statistics at each iteration. Although the recursive filter is fast, several blurs are needed at each iteration, depending on the chosen scales.

In Table 5.1, the execution time of 500 iterations of the algorithm is shown on a 2D and a 3D image. It is compared with the single scale model, and with its implementation on a GPU (graphics processing unit), using the NVidia's Cuda Library⁷. The GPU implementation leads to reasonable times for both 2D and 3D image. All the convolutions are parallelized by filtering each row/columns simultaneously. However note that modern GPU have a quite limited on-chip memory, and this prevent the parallelization of all the needed convolutions (a total of 6 and for each scale for the bi-partitioning case).

Images	single scale	4 scales	4 scales GPU
2D (300 ²)	13.17 Sec	56.36 Sec	7.61 Sec
3D (256 ³)	51Mn 20Sec	3h 4Mn	25Mn

Table 5.1: Time execution of 500 iterations. 'single scale' and '4 scales' corresponds to the CPU implementation of the method, and '4 scales GPU' corresponds to the multi-threaded algorithm on a GPU (NVidia's Quadro FX570M).

5.7 RESULTS

⁷www.nvidia.com/cuda

Our model was tested on several synthetic and real images in 2D and 3D.

In Fig. 5.14, our model is compared to the global Gaussian model (Chan & Vese model), on a synthetic image created as a simple cortex model with large inhomogeneities. The kind of initialization used here does not facilitate the global model as the contour crosses the edges of the structure to segment. Moreover, the distribution of intensities of the internal and external regions are the same. Due to its sensitivity to inhomogeneities, the global model (Fig. 5.14 top row) results in separating globally the bright areas from the dark ones, with no preference to sharp edges. Moreover, global non parametric models such as Parzen windows [55] give similar results.

By contrast, our model (Fig. 5.14 middle row) behaves quite well. It extracts accurately the details of the region's shape, it is more robust to inhomogeneities, and improves notably the behavior of the model to initialization. Note that the scales (Fig. 5.14 bottom row) are chosen accurately, depending of the proximity to the closest image edge. Due to the proximity of the initialization to the solution, only a few scales are needed.

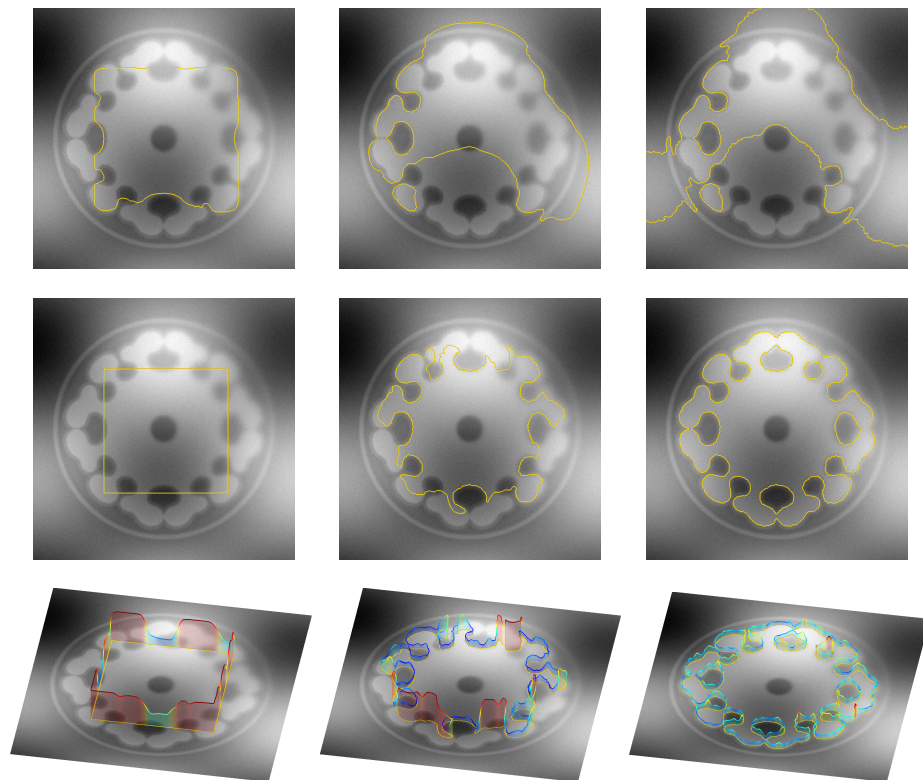


Figure 5.14: Top row: Initialization and evolution of the contour with the global Gaussian model ; Middle row: Initialization and evolution of the contour with our model ; Bottom-row: Scales chosen a each point of the contour.

In Fig. 5.15, we applied the model on the Lena image, with a fixed Gaussian windows of variance 8. The top row represent the mapping of the contour on the

original image, and the bottom row represent the local mean computed inside each region. The first column represent the initialization; the second one the convergence without the coherence constraint of Section 5.4; and the third one the convergence with coherence constraint.

As can be seen, the segmentation without coherence constraint produce some singularities that correspond to unwanted local minima. These singularities are visible on the piecewise smooth image as well, and slightly increase the energy of the segmentation.

As pointed out in section 5.3.2, the local model is similar to the Mumford Shah functional, that's why the result obtained on this image is strongly similar to the results obtained by minimizing the Mumford Shah functional. This gives us a partition of the image into piecewise smooth regions.

However, minimizing the Mumford Shah functional using our model gives us local minimas of the functional, due to active contour framework. This can be an advantage as it allows to give an initialization of the evolving contour, and to focus on a region in particular.

A previous segmentation model has been proposed in [108] to minimize the Mumford Shah functional using an active contour framework. However, this model is computationally expensive as it requires to solve a Poisson PDE inside each region, and at each iteration. Our model greatly improve the complexity, by filtering each region with a complexity in $O(n)$.

In Fig. 5.16, we applied the model on structures of interest in retina. These structures are quite difficult to segment, as they suffer from locally diffuse contour, and inhomogeneities coming from the imaging device. Moreover, large blood vessels crosses these structures, with similar image intensity. As can be seen the local model behave quite well by locally separating the inside region from the outside.

We incorporated in the model a rather simple shape prior, penalizing the points of the contour that are far from the centroid of the region, and thus preventing the contour from being attracted by the blood vessels. This shape prior is formulated in a Bayesian form, by maximizing the probability of a point to be close to the centroid of the region it belongs. This term alone result in the creation of circles, and coupled with the data attachment term prevent the deviation of the region from an regular circle.

The model has been implemented in $3D$, in order to create a complete $3D$ head model from anatomical MRI, and is the basis of the next chapter.

5.8 CONCLUSION

In this chapter, we proposed a segmentation model based on local statistics. This model is positioned between classical boundary models (very local criteria), and classical region models (very global criteria). Since it is based on local decisions, this



Figure 5.15: Equivalence between the piecewise smooth model and the Mumford Shah functional.

model is more robust to local variations of the regions of interest (in contrast, noise, blur ...). In order to avoid the computational burden of a direct estimation, these local statistics are expressed as the result of Gaussian convolutions, with a fixed Gaussian kernel.

The incidence of the size of the Gaussian windows has been analyzed, and we pointed out that our model generalizes several state-of-the-art segmentation models:

- With a **small Gaussian window**, the model is somehow equivalent to standard edge based models.
- an **intermediate Gaussian window** is similar to minimizing the Mumford-Shah functional.
- using a **large Gaussian window** is equivalent to usual region based models.

The model nonetheless introduce some new difficulties which are inherent to the fact of basing a global property (the segmentation) on pure local decisions. We emphasized two problems in the model, and proposed solutions to both:

- a **problem of coherence** in the evolution, that have been overcome by incorporating a constraint using a local simulated annealing. Although this constraint

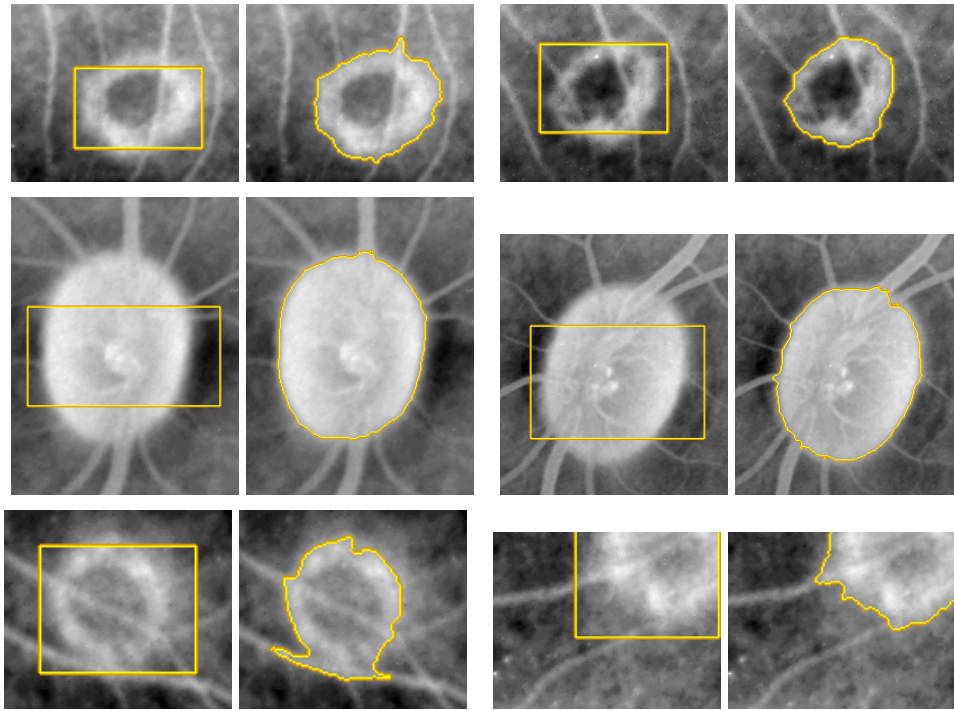


Figure 5.16: Segmentation of diffuse lesions in retina image.

restricts the nature of objects to segment, it is quite effective on objects that satisfy it. Actually this constraint is not so restrictive, as the majority of structures of interest in medical images satisfy it.

- an **absence of evolution** in locally homogeneous areas, that have been overcome by incorporating a scale-space approach in the computation of the speed for all the points of the contour. The scale selection is a computational burden, but an efficient implementation using highly parallel GPU is used.

At last, we proposed an efficient implementation using recursive Gaussian filters to compute the statistics, and efficient parallel library (OpenMP and Cuda), to fully use the potential of modern multicore CPUs and GPUs.

Future work includes using other local distributions of intensities, such as non-parametric distributions. However, incorporating the local variance in the model produced some numerical instabilities if the Gaussian window is too small. Therefore using locally complex distribution may not be meaningful, especially if the area of computation of local statistics is too small.

Another extension is to investigate the advantages of using more than two regions to segment images. However, our local formulations does not include one single distribution per region, so two regions may be sufficient to segment any image composed of several different structures (Fig. 5.15). It has been shown that 4 regions are sufficient to segment any kind of images due to the “four color theorem” [20].

We proposed a simple scale selection which gives more importance to sharp edges, but other scale selections can be used depending of the type of segmentation desired. An optimal scale coming from the derivative of the energy with respect to the scale could be formulated. However, it would certainly require to look at the neighborhood of each point of the contour, hence our scale selection method is not so different.

Finally, an extension of this model to 3D vectorial images is proposed in the next chapter, using registered T1 and T2 MRIs.

CHAPTER **6**

HEAD MODELS FROM BIMODAL MRIS USING LOCAL STATISTICS

Contents

Overview	102
6.1 Introduction	102
6.2 Anatomical representation of the head model	105
6.3 Creation of Head Model from Bi-modal Anatomical MRIs . . .	106
6.3.1 MRIs Acquisition & Registration	106
6.3.2 Scalp Segmentation	108
6.3.3 CSF Segmentation	109
6.3.4 WM Segmentation	110
6.3.5 GM Segmentation	112
6.3.6 Skull Segmentation	116
6.4 Validation	117
6.5 Results	118
6.6 Conclusion	122

OVERVIEW

This chapter presents a method for the creation of anatomical head model from bimodal T1 and T2 MRIs. After briefly reviewing the anatomical structures present in the head, a framework is proposed to segment successively the different anatomical structures included in the model by using the local-statistics model introduced in the previous chapter. The scalp, the CerebroSpinal Fluid, the White Matter, the Grey Matter and the skull are segmented consecutively, with a topological constraint is proposed for the segmentation of the Grey Matter.

Keywords: Segmentation, Head model, Topological constraint, Bimodal MRIs

6.1 INTRODUCTION

Creation of complete and detailed models of the head is extremely useful in the study of brain functionality and anatomy. It consists in automatically delimiting a set of anatomical structures from medical images of the head. The main purpose of these models is to be used for simulating electromagnetic propagation in the head, such as in the MEG/EEG *inverse problem*, where one tries to retrieve locations of neural activity by measuring the electro-magnetic field induced by the electrical activity of the cortex [31, 113]; or for computing the effects of electromagnetic fields induced by cellular phones on the brain. Moreover, the mapping of measurements from positron emission tomography (PET-scan) or functional magnetic resonance imaging (fMRI) on highly detailed cortical models helps a lot in the understanding of neural activity organization. At last, the morphometric study of sulcal and gyral patterns can lead to clinical diagnosis of pathological diseases.

In general, these models include 5 structures nested inside each others like “Russian nested dolls”. They include, from outer to inner structures: the skin, the skull, the CerebroSpinal Fluid (CSF), the Grey Matter (GM), and the White Matter (WM) [8].

In this work, the segmentation of these anatomical structures is performed on couples of co-registered T1- and T2- weighted MRIs. These two images provide rather complementary informations concerning the localization of anatomical structures, but most of the methods create head models from single T1 MRIs, and neglects the T2 MRIs. This chapter reflects the benefits from using both T1 and T2 images, as it allows the segmentation of finer structures, especially concerning the segmentation of the skull. All the couples of T1 and T2 MRIs used in this chapter comes from the *Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany*.

The segmentation of the scalp is relatively easy to find as there is a clear separa-

tion in the T2 MRI between the fat present in the skin and the air. However, there are some areas where the fat layer is too thin, which result in no visible separations between the air and the skull. Fortunately, these areas are very localized (nasals cavities and acoustic canals), and increasing the regularity of the deformable surface prevents potential leaks.

The segmentation of the skull from a single MRI (T1 or T2) is difficult and challenging as such images only provide incomplete informations. Most of the time, the skull is approximated by inflating a convex envelop of the cortex, which may be acceptable for the MEG inverse problem but not for the EEG one.

There exists some approaches based on mathematical morphology that manage to extract the skull from single MR data [7, 85]. Unfortunately, due to the limitation of the skull definition in single MRIs, these approaches are quite approximative and only manage to extract correctly the upper part of the skull.

Several approaches performs the skull segmentation by using CT data instead of MR data. While CT data provide a better definition of skull, it does not give any informations about interfaces between soft tissues. However, an accurate and complete head model can be constructed by extracting the skull from CT data, and other interfaces from MR data [100]. Unfortunately, having two different kind of acquisitions for the same subject is often inconvenient and uncommon, and expose the subject to nocive X-ray radiations. On the contrary, acquiring both T1 and T2 MRIs can be done in a single MRI session, and provide a rather good information about the skull localization.

Extraction of the inner and outer cortical surfaces from T1 MRI data is a difficult task due to its highly convoluted nature [28, 111, 63]. ATLAS based approach do not typically perform well the segmentation of the cortex due to high intersubject versatility. In general, most of the methods are based on successive steps to segment the cortex [31, 41, 95, 66, 24]:

- It is common to first remove non-brain tissue from the MR volume in order to facilitate the segmentation. This approach is called **skull-stripping** and consists in removing the skull and all other extracranial tissue from the brain. Several methods exist to perform this, and most of them are based on morphological operations or atlas-based registrations [99, 95, 41, 91].
- The second step consists in image enhancements and low level segmentations. Given the skull-stripped brain, this task consist in removing inhomogeneities present in MRIs, and classifying the voxels into three different classes: WM, GM and CSF. This **labellization** is in general performed using Markov Random Fields methods (MRF) or some clustering algorithms (EM, fuzzy C-means, ...) [95, 111, 44, 51, 118].

- The last step consists in removing topological inconsistencies from the previous labellization of the brain. In general, this step is performed by evolving **deformable surfaces** [28, 64, 4] **with topological constraints** [42] inside the brain toward the interfaces separating the GM and the CSF. This approach allows to get a sub-voxellic representation of the brain, with the correct topology.

An interesting approach has been proposed in [117] and [63], and later in [35]¹ and [36]: the authors used a system of coupled surfaces initialized as two concentric spheres inside the white matter, to extract inner and outer cortical surfaces simultaneously. They based their methods on the assumption that the cortex is a layer of almost constant thickness, and used a traditional boundary based evolution with incorporation of a force that prevents the coupled surfaces to move away, or being too close to each other. This approach has the advantage of being robust to locally diffuse contours due to the partial volume effect. However, this assumption is not valid for all cases as the cortex thickness may vary a lot from one subject to another.

In addition to the complexity of these structures, some problems arise due to the limitations of medical imaging equipment:

- **Inhomogeneities** are quite common in MRIs due to the non uniformity of the MR field. It results in low frequency alterations, or intensity drifts, that smoothly perturb the intensity of the anatomical structures.
- Most of the MRIs suffer from **noise** coming from the imaging equipment. This noise follow a *Rician* distribution [98], but can be simplified as a *Gaussian* distribution. This noise can be quite strong depending on the acquisition parameters.
- **The partial volume effect** occurs when a single voxel contains a mixture of multiple tissue. It results in locally diffuse boundary between distinct anatomical structures.

Because of these alterations, most of the methods cannot use directly active contours to extract the cortex from MRI and some preprocessing is needed. As discussed in chapter 5, these alterations make classical edge-based and region-based model for active contours inefficient.

In this chapter, we propose a framework for creating a complete head model by using the segmentation method proposed in chapter 5, using bimodal registered T1- and T2-weighted MRIs. Our goal is to segment anatomical structures directly from the initial MRIs, without any intermediate step that would require additional parameters.

¹As pointed out in [82] the authors introduced a variational formulation but made an abuse in the derivation of the energy with respect to the level-sets.

6.2 ANATOMICAL REPRESENTATION OF THE HEAD MODEL

Before presenting our method, let's see a brief overview of the anatomical structures present in the head. All the following anatomical pictures comes from the *Grant's Atlas of Anatomy*, 12th edition. The head can be simplified as a set of consecutively nested anatomical structures (Fig. 6.1). From outer to inner structure, it includes:

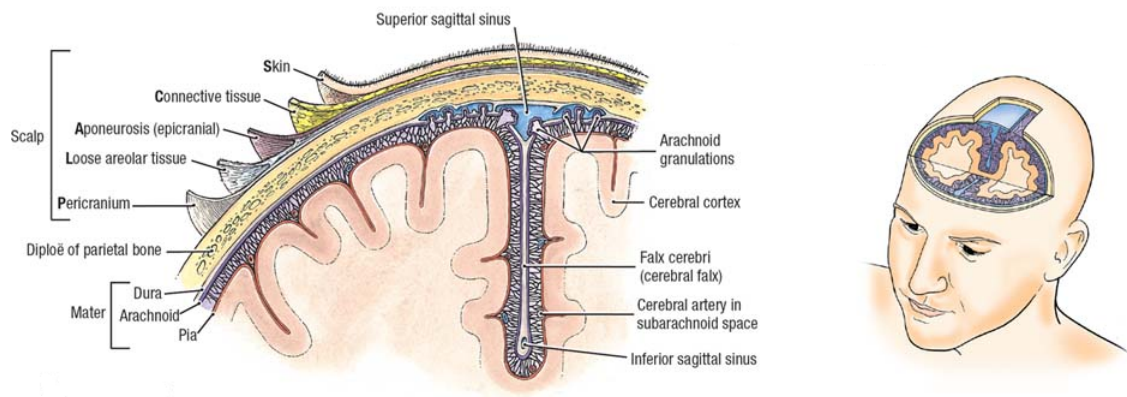


Figure 6.1: The head can be seen as consecutively nested anatomical structures.

- **The scalp**, which is formed of several thin layer besides the skin (Fig. 6.1). It covers the whole head, apart from the nasal cavities and acoustic canals.
- **The skull**, constituted of several bones that englobe and protect the brain (Fig. 6.11). The skull covers entirely the brain and includes one big hole corresponding to the spinal cord, and some tiny holes (called *foramens*) in which nervous fibers and veins pass through.
- **The CSF**, which constitute the area between the brain and the skull. This area is mainly composed of liquid, but includes various type of anatomical tissues (dura, cerebral falx, vascularization, ...). The CSF also goes deep inside the brain, in areas that correspond to the *brain ventricles*.
- **The GM**, which correspond to the *cerebral cortex* (Fig. 6.3). The cortex is a highly convoluted thin layer of grey matter, of 1mm to 5mm thickness, with an average of nearly 3 mm [122, 8]. It is composed of two separate hemispheres, has a spherical topology, and is located between the WM on the inside, and the CSF on the outside [8].
- **The WM**, which is the last structure corresponding to the “inside of the brain” (Fig. 6.3). It is mainly composed of nervous fibers linking the neurons of the cortex to each others. It includes the *corpus callosum* that constitute the only link between the two hemispheres.

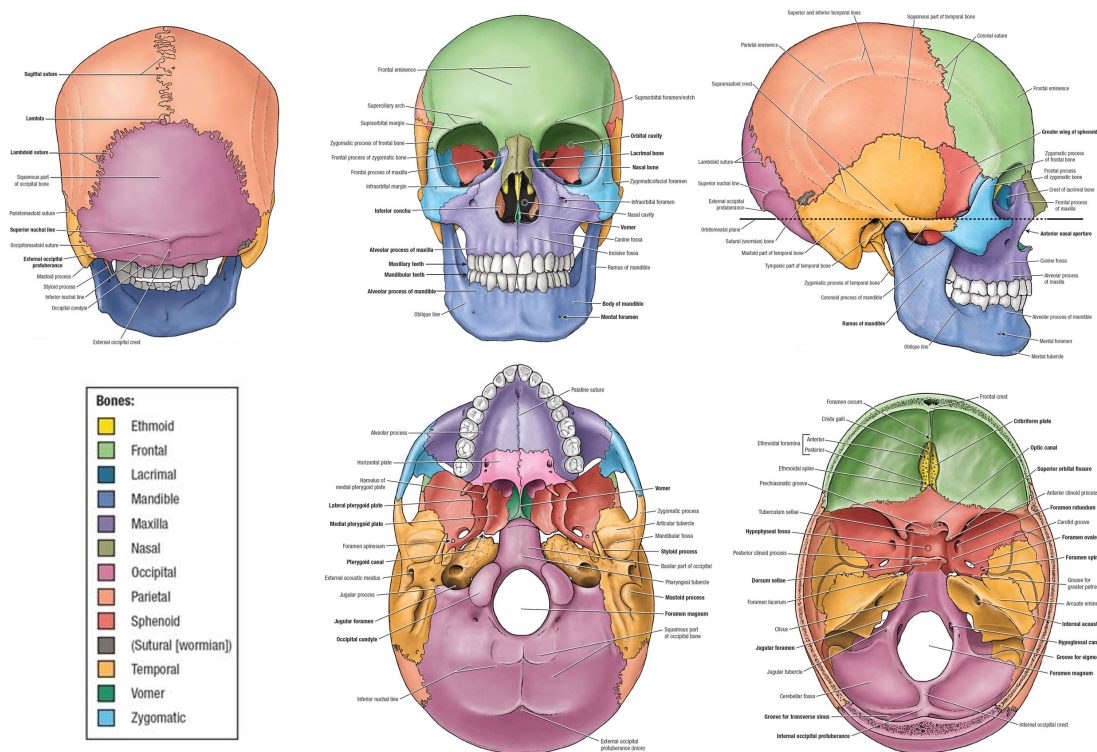


Figure 6.2: Anatomical representation of a human skull.

6.3 CREATION OF HEAD MODEL FROM BI-MODAL ANATOMICAL MRIS

Our method consists in successively segmenting the five anatomical structures present in the model, using bimodal registered T1-weighted and T2-weighted MRIs. This bimodal approach is driven by the fact that a combination of T1-weighted or T2-weighted MRI brings significantly more information about tissues, and especially about the skull interfaces. Indeed, these two images are rather complementary, and while in some places the T2-weighted MRI provide a better localization of inner skull interface, T1-weighted MRI provide a globally better localization of the outer skull interface. Moreover, it does not need much additional efforts to obtain several MRIs of the same subject, with different weightings. Some centers are even starting to systematically acquire both these images.

6.3.1 MRIs Acquisition & Registration

All the couples of T1 and T2 MRIs used in this chapter have been acquired from a SIEMENS Trio 3T imaging device, from the *Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany*.

The T1-weighted MRIs were acquired using a 3D MP-RAGE [47] (magnetization-prepared rapid gradient echo) sequence with selective water excita-

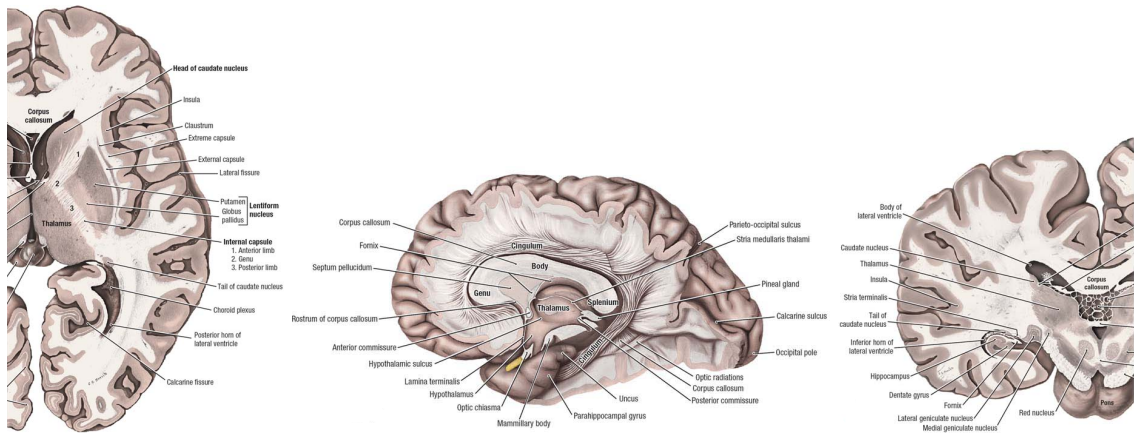


Figure 6.3: Anatomical representation of a human brain.

tion and linear phase encoding. Preparation consists of a non-selective inversion pulse.

Imaging parameters: $TI = 650$ ms; repetition time of the total sequence cycle, $TR = 1300$ ms; repetition time of the gradient-echo kernel (snapshot FLASH), $TR_A = 10$ ms; $TE = 3.93$ ms; $\alpha = 10^\circ$; bandwidth = 130 Hz/pixel (i.e., 67 kHz total); image matrix = 256 x 240; FOV = 256 mm x 240 mm; slab thickness = 192 mm; 128 partitions; 95% slice resolution; sagittal orientation; spatial resolution = 1 mm x 1 mm x 1.5 mm; 2 acquisitions. To avoid aliasing, oversampling is performed in the read direction (head-foot).

The T2-weighted MRIs were acquired using a 3D Turbo Spin Echo sequence, using 8 channel array head coil.

Imaging parameters: $TR = 2000$ ms, $TE = 355$ ms, bandwidth = 355 Hz/pixel; image matrix = 256 x 256; FOV = 256 mm x 256 mm; slab thickness = 176 mm; 176 partitions; sagittal orientation; spatial resolution = 1 mm x 1 mm x 1 mm; 1 acquisition.

The registration between the T1- and T2-weighted MRIs is done as follows: first, the T1-anatomy is aligned with the stereotactical coordinate system proposed by Talairach and Tournoux [104]. The T2 image is then co-registered with the T1-anatomy using rigid-body transformations [46], implemented in FSL². Despite some artifacts, the obtained results are globally satisfactory and the two registered images can indeed be used as a single multimodal image.

A couple of registered T1- and T2-weighted MRIs can be seen in Fig. 6.4 through their coronal, frontal and sagittal views. One of the main difference between those two images is that the cerebrospinal fluid appears very dark in the T1 MRI, while

²<http://www.fmrib.ox.ac.uk/fsl>

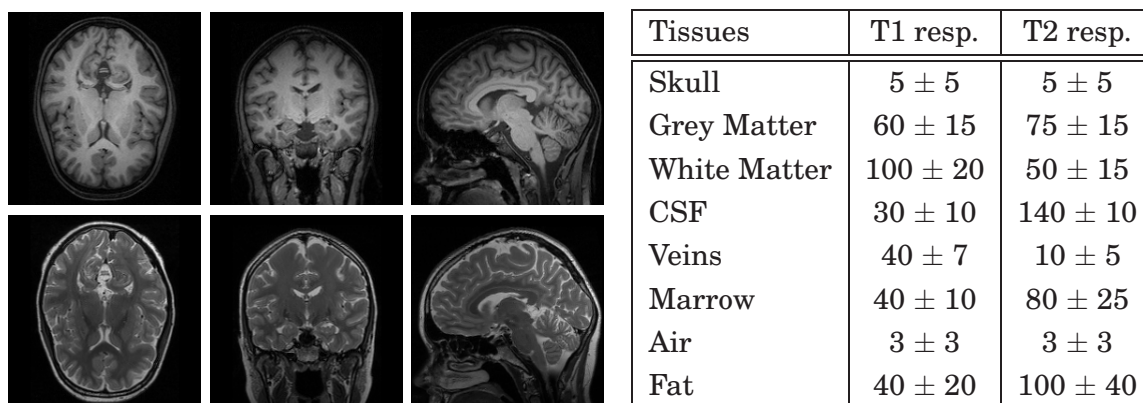


Figure 6.4: Left: T1-weighted anatomical MRI (top) and T2-weighted anatomical MRI (bottom); Right: Tissues response on T1- and T2-weighted anatomical MRI normalized between 0 and 255.

being very bright in the T2. It is thus easier to extract the inner interface of the skull and the outer interface of the cortex from the T2-weighted MRI, as there is a clear separation between the skull, the CSF and the brain. On the other hand, the marrow response in the T2 image is also very strong, and this leads to high contrast between the marrow and the bone. As a consequence, the evolving surface will be stuck in these areas, and the skull won't be segmented correctly.

However the T1 MRI provides a marrow response close to the bone response, and there are in some parts a better definition of the interface between the skin and the skull. Moreover, the contrast between the GM and the WM tissues is much more pronounced than in the T2 MRI. Therefore the T1 MRI is a better candidate to segment correctly the outer interface of the skull and the inner interface of the cortex.

The T1 and T2 MRIs are thus complementary, and a combination of both bring significantly more information about localization of anatomical structures.

6.3.2 Scalp Segmentation

The segmentation of the scalp is done by initializing the deformable surface as a cube around the head, and making it evolve using a simple boundary-based model [16] on the T2-weighted MRI. Indeed, the T2 MRI provide a better definition of the fat present in the skin than the T1 (Fig. 6.4), and thus allows the extraction of a more detailed scalp surface. The cube will slowly shrink on itself, and finally converge toward the skin (Fig. 6.5, left). Note that the regularity constraint of the model is rather large, in order to avoid the surface from being stuck in noisy areas of the background.

The scalp will act as a “mask ” in order to constraint the skull segmentation inside it. Indeed, there are some areas where it is impossible to find a clear separation between the skull and the air, like in the acoustic canals or in the nasal cavities. Consequently, the scalp has to include deep details in order to accurately prevent the

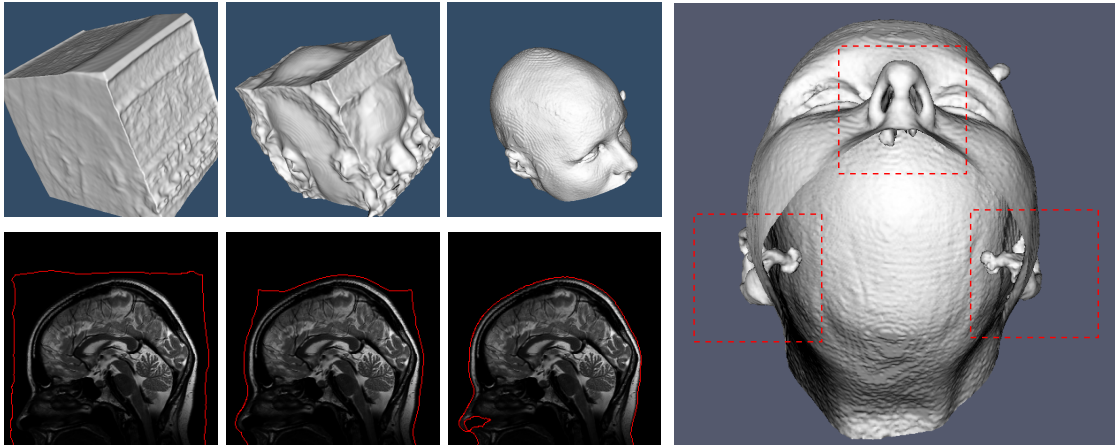


Figure 6.5: Left : 3D evolution of the scalp segmentation, and corresponding slices in the T2 MRI of the evolution. Right : The nasal cavity and acoustic canals are segmented to prevent the skull segmentation from leaking in these areas.

skull from leaking in the air parts. Once the surface has converged toward the skin, some additional iterations³ are made while reducing the regularity constraint and using the local-based model of chapter 5 with a small fixed neighborhood of 2. The surface will then enter in deep parts of the scalp, like the acoustics canals and the nasal cavities (Fig. 6.5, right).

6.3.3 CSF Segmentation

The CSF is segmented by initializing a sphere inside the brain, and evolving it using the local-based model of chapter 5 with scale selection (2 scales: 4 and 16).

To perform the CSF segmentation, the front has to evolve from the WM through the cortex, to finally fit the interface separating the CSF from the skull. However, the cortex is quite visible in both T1 and T2 MRIs, and without any “tweaking” the surface will likely converge toward the WM/GM or the GM/CSF interfaces. Thus, an a-priori parameter need to be fixed in order to define the desired behavior of the front.

The segmentation model of chapter 5 aims at separating the local distributions of the regions inside and outside the front. Since the model is locally-based, there is not one single distribution for each region, hence the a-priori parameter cannot simply be the type of regions desired (the main advantage of the local approach is its robustness to inhomogeneities). In the bimodal case, these distributions are local *bivariate normal distributions*. In order to make the front evolve through the cortex, the local distributions of the GM, WM and CSF have to be as similar as possible to be considered as a single anatomical structure. A simple way to ensure this similarity is

³Only a few iterations are required, otherwise the surface will grow into deep structures of the skull.

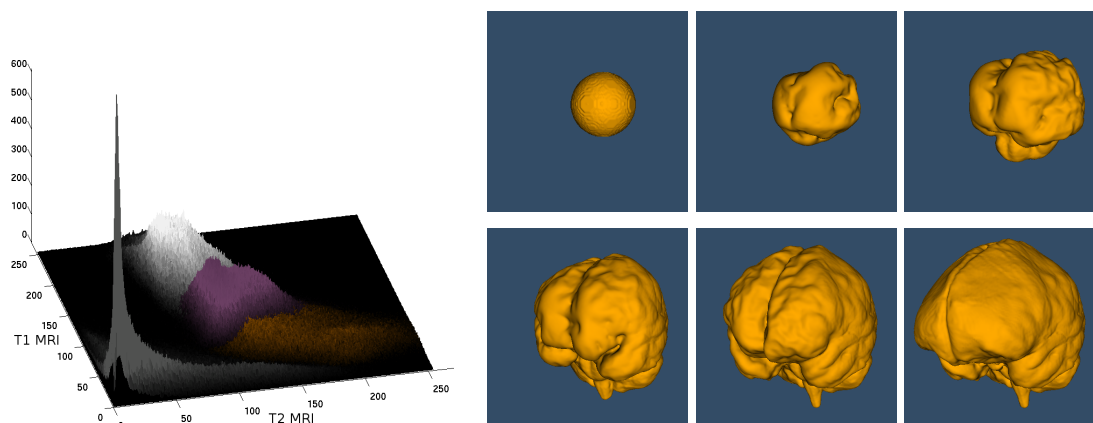


Figure 6.6: Left: Histogram of 4 anatomical structures in a bias corrected bimodal MRI: the skull (Grey), the CSF (Orange), the GM (Purple) and the WM (white); Right: 3D evolution of the CSF segmentation.

to fix a-priori the **local correlation** of the bivariate normal distributions (Fig. 6.6). For example, Fig. 6.6-left shows the histogram of the skull, CSF, GM and WM in a bias corrected bimodal MRI. In order to extract the interface separating the skull from the CSF, the front need to evolve from the WM to the GM to the CSF, without stopping its evolution. By tuning the “orientation” of the distributions (linked to correlation), the GM, WM and CSF can be considered as one single anatomical structure different from the skull. In our case, we computed this correlation from a set of previously segmented head models (good enough to be considered as “ground truth”) by estimating the correlations of the region including the CSF, GM and WM. These correlations do not vary a lot from one image to another (an average of -0.7 ± 0.1), and can indeed be used as an a priori parameter to segment the CSF. The front will finally evolve from the WM through the cortex, and will stop on the interface separating the skull from the CSF (Fig. 6.6, right)

The CSF segmentation plays the role of a “skull-stripping ” that separates the brain from other extracranial structures. There are various method to perform skull-stripping in the literature, and some of them are actually rather complex [99]. Our method manages to segment the CSF (corresponding to the inside of the skull) with a lot of details, especially the veins in the dura and the sagittal sinus (the vein that goes along the separation of the two cortex hemisphere).

6.3.4 WM Segmentation

The WM is then extracted by initializing the front using the previously segmented CSF, and evolving it thanks to the local-based model of chapter 5 with a fixed neighborhood of 8.

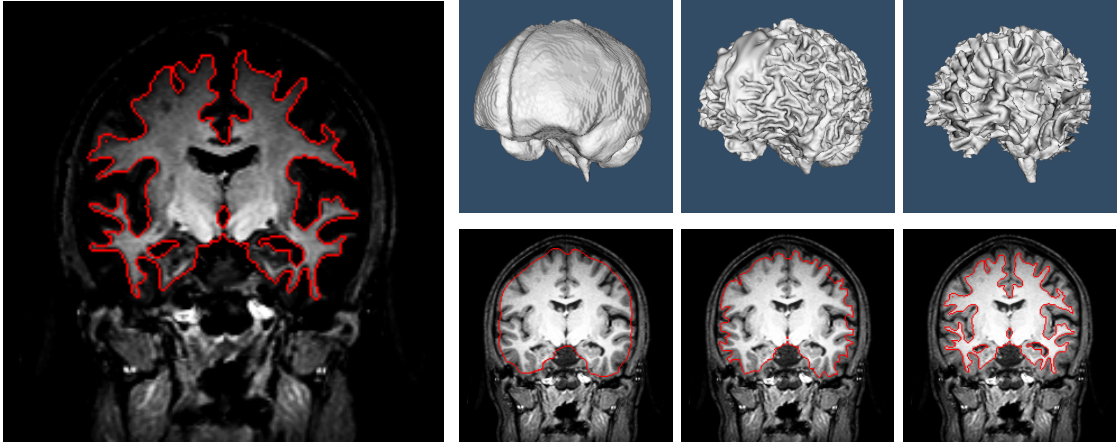


Figure 6.7: Left: projection of the WM segmentation in a slice of the combined I_1 image (Eq.6.1); Right: 3D evolution of the WM segmentation, and corresponding slices in the T1 MRI of the evolution.

The WM is a structure composed of **very sharp gyri** (ridges of the brain), and **very smooth sulci** (fissures of the brain). In order to accurately segment the sharp gyri of the WM, a common way is to initialize the deformable surface outside the brain (eg. skull-stripping), and makes it deflate in order to fit the ridges that might be difficult to reach from the inside (Fig. 6.7).

The interface between the GM and the WM is particularly visible in the T1 MRI, but the front has to evolve from the CSF through the cortex once again, and thus might be stuck on the interface separating the CSF from the GM. As for the CSF segmentation, we cannot simply fix the type of tissue desired due to the locality of the segmentation model. The WM segmentation is performed in **two steps**:

- In order to emphasize the WM from the GM and CSF, the segmentation is first performed on a scalar image I_1 corresponding to the combination of the T1 and T2 MRIs (Fig. 6.4, right), such that:

$$I_1(x) = \max(T1(x) - T2(x), 0) \quad \forall x \in \Omega \quad (6.1)$$

where I is the combined image, $T1$ and $T2$ are the two MRIs, and Ω the image domain. By doing this simple difference, the intensity of the CSF and the GM will be close to zero, while the white matter will appear rather bright (Fig. 6.7, left). The front will thus evolve through the cortex, and get closer to the interface separating the WM from the GM.

- The segmentation is then performed on the T1 MRI alone, with a small fixed neighborhood of 2 in order to slightly refine the front position.

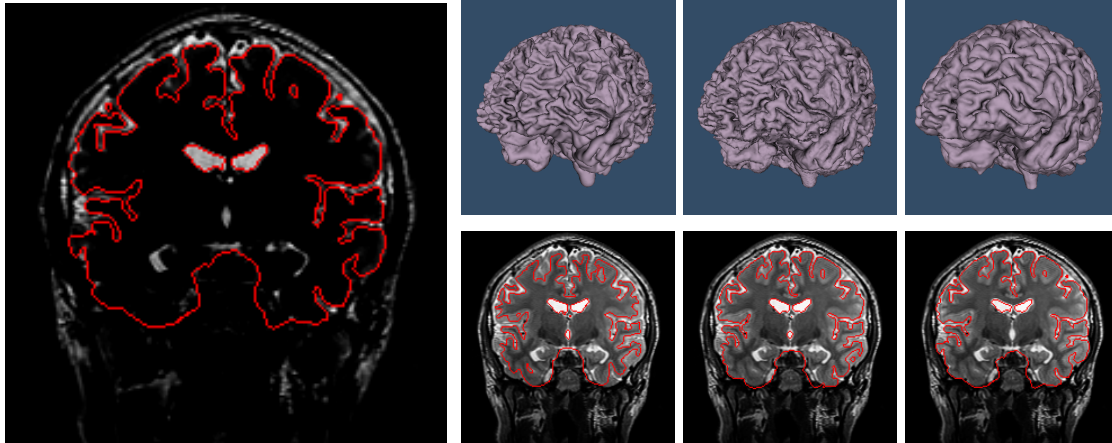


Figure 6.8: Left: projection of the GM segmentation in a slice of the combined I_2 image (Eq.6.2); Right: 3D evolution of the GM segmentation, and corresponding slices in the T2 MRI of the evolution.

6.3.5 GM Segmentation

The GM is then segmented by initializing the front using the previously segmented WM, and evolving it thanks to the local-based model of chapter 5 with a fixed neighborhood of 4.

In opposition to the WM, the GM is a structure composed of **very smooth gyri**, and **very sharp sulci**. In order to accurately segment the sharp sulci of the GM, a common way is to initialize the deformable surface inside the brain, and make it inflate in order to fit the fissures that might otherwise be difficult to reach from the outside.

The interface between the GM and the CSF is particularly visible in the T2 MRI, and the front has to evolve from the WM through the cortex once again. Similarly to the WM segmentation, the GM segmentation is performed in **two steps**:

- In order to emphasize the CSF from the GM and the WM, the segmentation is first performed on a scalar image I_2 corresponding to the combination of the T2 and T1 MRIs (Fig. 6.4, right), such that:

$$I_2(x) = \max(T2(x) - T1(x), 0) \quad \forall x \in \Omega \quad (6.2)$$

where I is the combined image, $T1$ and $T2$ are the two MRIs, and Ω the image domain. Similarly to the WM segmentation, this combination emphasize the CSF from the other tissues, and the front will evolve toward the interface separating the CSF from the GM.

- The segmentation is then performed on the T2 MRI alone, with a small fixed neighborhood of 2 in order to slightly refine the front position.

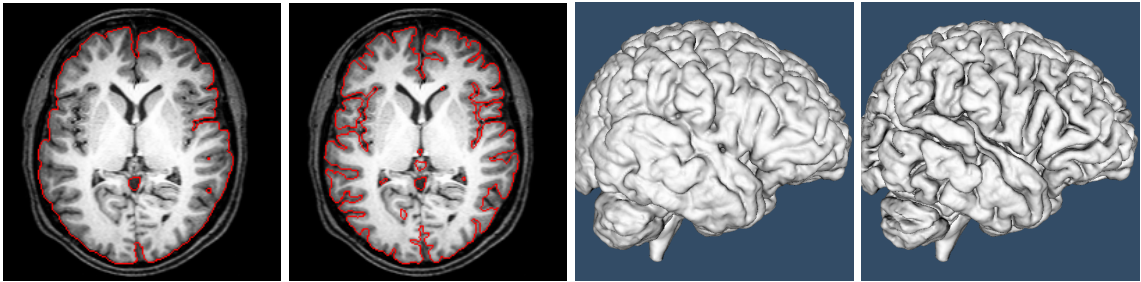


Figure 6.9: Left: projection of the GM segmentation in a slice of the T1 MRI, without (first) and with (second) the topological constraint; Right: 3D segmentation of the GM, without (first) and with (second) the topological constraint.

Topological Constraint

However, one problem concerning the shape of the cortex remains. In some areas, the sulci of the GM are so thin that the CSF lying inside is no more visible in the MRIs, due to the partial volume effect. Therefore, these sulci are not segmented, which impacts significantly the global shape of the brain (Fig 6.9). This is a major drawback concerning functional studies of the brain, since it creates some “proximity” between cortical areas that might be completely wrong. Shape priors based on training sets of brains [90] are not suitable to segment these sulci, due to the high intersubject variability of the cortex.

To overcome this limitation, classical methods incorporate some **topological constraints** in order to preserve the spherical topology of the cortex. These constraints aims at preventing any fusion of the front with itself, thus preserving the sulci and gyri. Note that this goes against one of the two strengths of the level-set method (the other one being the absence of parameterization).

Several methods exist to preserve topology. Most of them are quite simple, based on very local criteria that monitor the points in the level-set that could change the topology of the front (non-simple points)[43, 92]. However, these approach just freeze the front at some specific areas, creating undesired “stair effects“. Recently, new methods have emerged to preserve topology, based on variational formulation of *higher order active contours* [102, 86, 58]. The main idea is, for each point of the front, to integrate over the whole contour in order to look for the points that might get too close. These approaches allow the contour to change its geometry progressively, and no more freezing it ”at the last moment“. They corrects the ”stair case“ effect, and provide results of very good accuracy. However, these methods are computationally expensive, since an integral over the contour is required for each point of the contour.

We propose a novel and efficient method for preserving the topology in level-sets. It is based on detecting **skeletons** of the regions inside and outside the front and

”repelling“ the front if it gets too close to these skeletons. In level-sets function, skeletons correspond to ”shocks“ or singularities of the distance function, that can be detected very easily by looking at the gradient of the level-set (however, more accurate methods exists to detect them [97]). If a shock is detected, a force is created that locally repels the front, and thus preserves its topology.

The evolution is based on the following equations:

$$\begin{aligned} \mathcal{S}_\epsilon(x) &= \text{sign}(\Phi(x)) f_\epsilon(|\nabla\Phi(x)|), \\ \frac{\partial\Phi}{\partial t}(x) &= \alpha\mathcal{S}_\epsilon(x) - |\nabla\Phi(x)|, \end{aligned}$$

with

$$f_\epsilon(x) = \begin{cases} 1 & x < \epsilon \\ 0 & x > \epsilon \end{cases}$$

where \mathcal{S}_ϵ correspond to a “skeleton map” that is 1 on the positive skeleton, -1 on the negative skeleton and 0 elsewhere, and α, ϵ are two non-negative scalars. The skeleton corresponds to areas where the gradient norm of Φ is inferior to a certain threshold ϵ . This skeleton map just plays the role of a “conditional balloon force“.

This topological constraint is rather simple, and corrects the undesired “stair effect” of the previous methods. It worked quite well for $\epsilon = 0.5$, and the weight α has to be chosen rather large, in order to prevail from other evolution terms. The implementation of this constraint is straightforward and efficient, since it does not need to compute the skeletons in the whole image: for a given point x in the narrow band, the only computation needed to detect if it lies in the skeleton is $|\nabla\Phi(x)|$ (with Neumann boundary conditions). If so, a repelling force appear, preventing the front to evolve towards the skeleton direction.

However, this method suffers from some inconvenients:

- The main inconvenient is that some restrictions on the evolution speed are required: the absolute value of the speed can’t be greater than one (one pixel per time step) in order to efficiently preserve topology. A way to overcome this limitation is to incorporate some CFL conditions on the different evolutions terms.
- The two self repelling parts of the fronts are separated by a thin space of one pixel wide corresponding to the detected skeleton (Fig. 6.10). This may be an advantage as well if some methods require this type of constraint (in order to *mesh* some particular domains, etc). A way to overcome this limitation is to detect skeletons and evolve the front in an image of finer scale.

As can be seen in Fig 6.9, this constraint works quite well for the segmentation of the GM, as the sulci are clearly visible in the topology-constrained segmentation (However, these sulci are “thicker” than in reality due to the 1-pixel vacuum evoked).

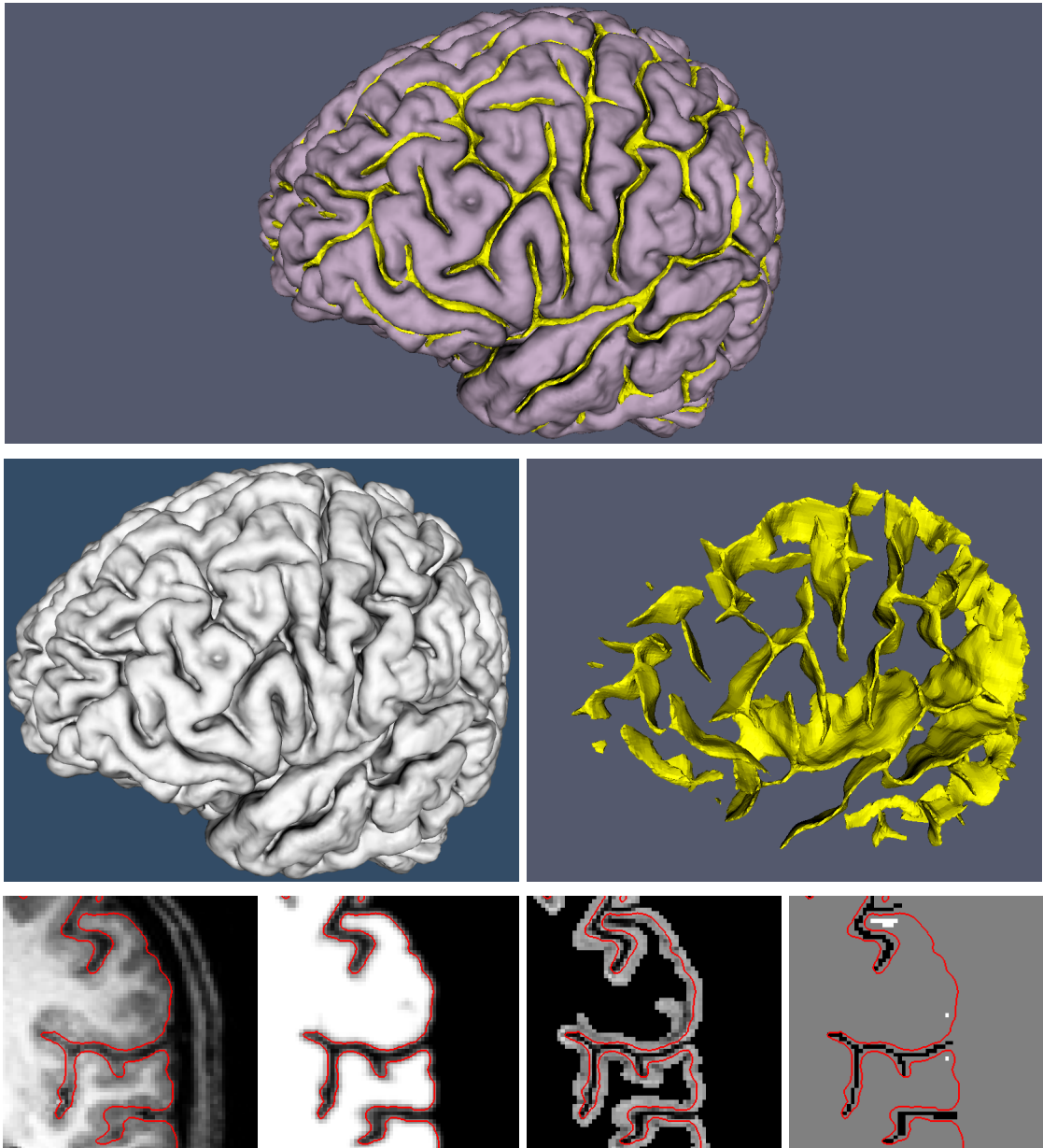


Figure 6.10: Top: GM segmentation (purple) with external skeleton (yellow); Middle: GM segmentation next to external skeleton; Bottom: Projection of the GM segmentation on : the T1 MRI; its corresponding level-set function (LS); gradient norm of the LS (inside narrow band); detected skeleton.

In Fig.6.10 can be seen a 3D segmentation of the cortex under this topology constraint. The exterior skeleton (yellow structure) prevents the fusion of the front in sulci areas, and thus preserve the topology of the front. This skeleton can be seen as a “frontier” that cannot be crossed. Moreover, it corresponds to cortical folds, that are of great interest for morphometric studies of the brain.

As can be seen in Fig.6.10-Bottom, this constraint is quite easy to implement, as a simple gradient norm is required for the pixels inside the narrow band.

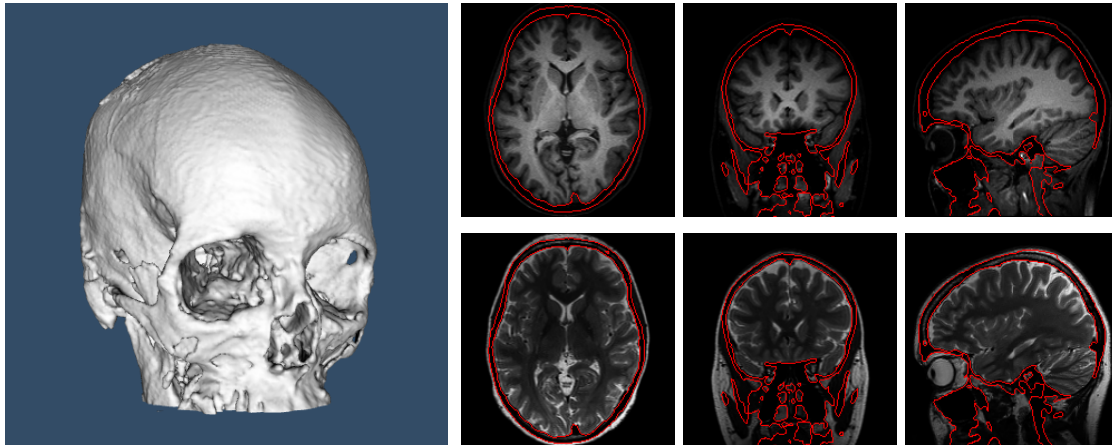


Figure 6.11: Left : 3D view of the segmented skull ; Right : Projection of the skull on the T1 image (top row) and the T2 image (bottom row).

6.3.6 Skull Segmentation

The skull is segmented by initializing the front as a layer between the skull-stripping and the scalp (by inflating the CSF), and evolving it using the local-based model of chapter 5 with a small fixed neighborhood of 2.

The skull is a structure rather difficult to segment since anatomical MRIs (T1 and T2) only provides incomplete informations about its position. However, as stated in Section 6.3.1, the T1- and T2-weighted MRIs are complementary concerning the localization of the skull, and a combination of both images bring more information.

The first approach we used was to combine both T1 and T2 MRI into one scalar image I_3 , such that:

$$I_3(x) = T1(x) \cdot T2(x) \quad \forall x \in \Omega$$

This combination indeed emphasized the skull from other tissues (Fig. 6.4), and its segmentation provided good initial results. Unfortunately, this worked on relatively few subjects, and was not good enough to be included in an automatic framework for head model creation.

Then, we followed the same approach used for the CSF segmentation, by performing the segmentation on the bimodal T1 T2 image, and fixing the correlation to emphasize the skull from the other tissues. This combination provided better results (Fig. 6.11), and have been positively reproduced on different subject (Fig. 6.16).

However, it is still impossible to find a clear separation between the skull and the air in regions where the skin is too thin, like the acoustic canals and nasal cavities. Therefore, we used the previously segmented scalp as a mask to locally stop the evolution of the deformable surface. In Fig. 6.12, the scalp and the skull are shown as well as their projection in the T2 MRI. It can be seen that the skull does not leak in the

acoustics canals or nasal cavities, thanks to the highly detailed scalp segmentation.

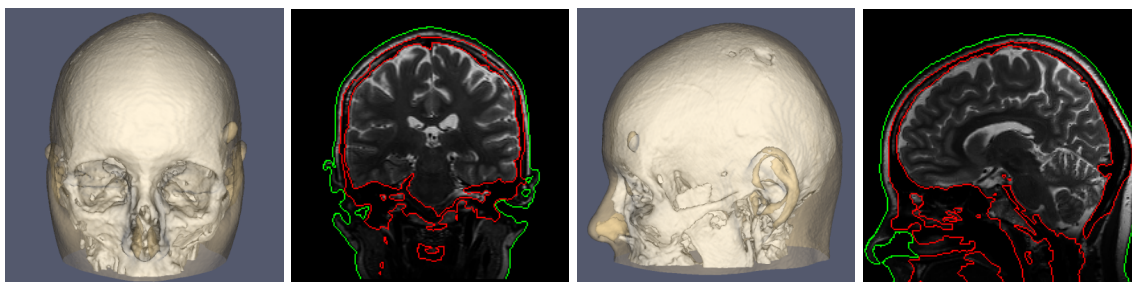


Figure 6.12: The skull does not leak in the background thanks to the scalp.

There are unfortunately some segmented structures that do not belong to the skull, and one of the main drawback is that some veins are also segmented. Indeed they appears very dark in the T2, and a bit brighter in the T1 (Fig. 6.4) and the combination of both image make their intensities close to the one of the skull. As a consequence, the vein close to the inner skull interface (superior sagittal sinus) as well as some carotids are falsely segmented. Note that, for example in Fig. 6.11, right, second column, the zygomatic arch (Fig. 6.11) is a structure that is difficult to segment in single MRI, as it appears to be part of the darker structure on its top in the T1 MRI, and on its bottom in the T2. The combination of both help extracting it, without leaking in other neighboring tissues.

6.4 VALIDATION

We validated our model according to the GM and WM segmentations. We did not manage to validate the skull segmentations since our results include several other anatomical tissues (bone marrow, veins, air in bucal cavity) that strongly perturb the validation process.

In order to validate results of medical segmentations, a **ground truth** corresponding to a labellization is required. Most of the ground truth are done by experts, that manually labellize voxels in MRIs to particular anatomical structures. An interesting and accurate validation method can be performed using the visible human database⁴. It consist in several acquisitions of a complete human (CT, T1 MRI, T2 MRI, etc), as well as true colored photos of a sliced version of the human. These photos are so detailed that differentiation between anatomical structures becomes trivial, and help defining with precision a ground truth.

In our case, we validated our method using the BrainWeb simulations⁵. It consists in simulating MRI acquisitions, given specific parameters and percentage of synthetic

⁴<http://www.nlm.nih.gov/research/visible/visible.human.html>

⁵<http://www.bic.mni.mcgill.ca/brainweb/>

alteration (noise and bias). These simulations starts from a brain phantom, which plays the role of ground truth for the MRIs simulations.

Images	GM Dice coefficient	WM Dice coefficient
0% noise & 0% bias	0.94	0.91
5% noise & 20% bias	0.89	0.91
9% noise & 40% bias	0.85	0.74

Figure 6.13: Dice coefficient of GM and WM on 3 incrementally altered MRIs.

In order to estimate a “ratio of accuracy”, a common choice is to use the **Dice coefficients** related to the Jaccard index, defined as follows:

$$\mathbf{Dice} = \frac{2 \mathbf{TP}}{(\mathbf{FP} + \mathbf{TP}) + (\mathbf{TP} + \mathbf{FN})},$$

where **TP**, **FP** and **FN** correspond respectively to the True Positive (result \in ground truth), False Positive (result \notin ground truth), and False Negative (ground truth \notin result) results (Fig. 6.14). A value of 0 indicates no overlap and a value of 1 indicates perfect agreement. Higher numbers indicate better agreement.

The validation of our model can be found in Fig.6.13. We tested our model on 3 simulated images : one free from alterations (0% noise & 0% bias) ; one slightly altered (5% noise & 20% bias); and one strongly altered (9% noise & 40% bias). We found an average of almost 0,90 in Dice coefficient, which is quite good given the strong alterations that occur in the last image. Possible errors comes from the topology constraint, that require a vacuum voxel between each segmented sulci.

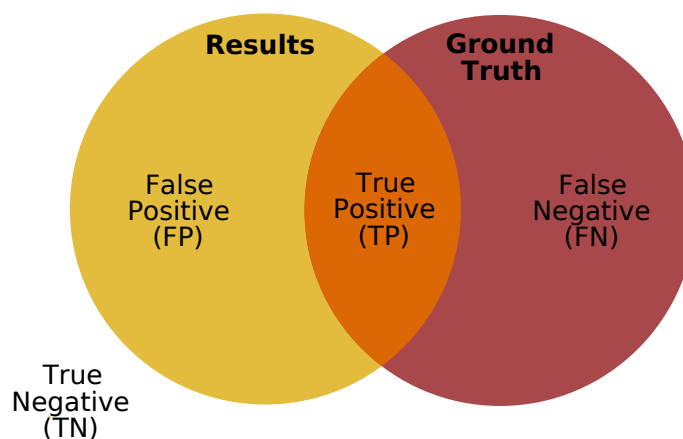


Figure 6.14: Dice coefficient: a ratio between the TP, and the FP and FN is computed.

6.5 RESULTS

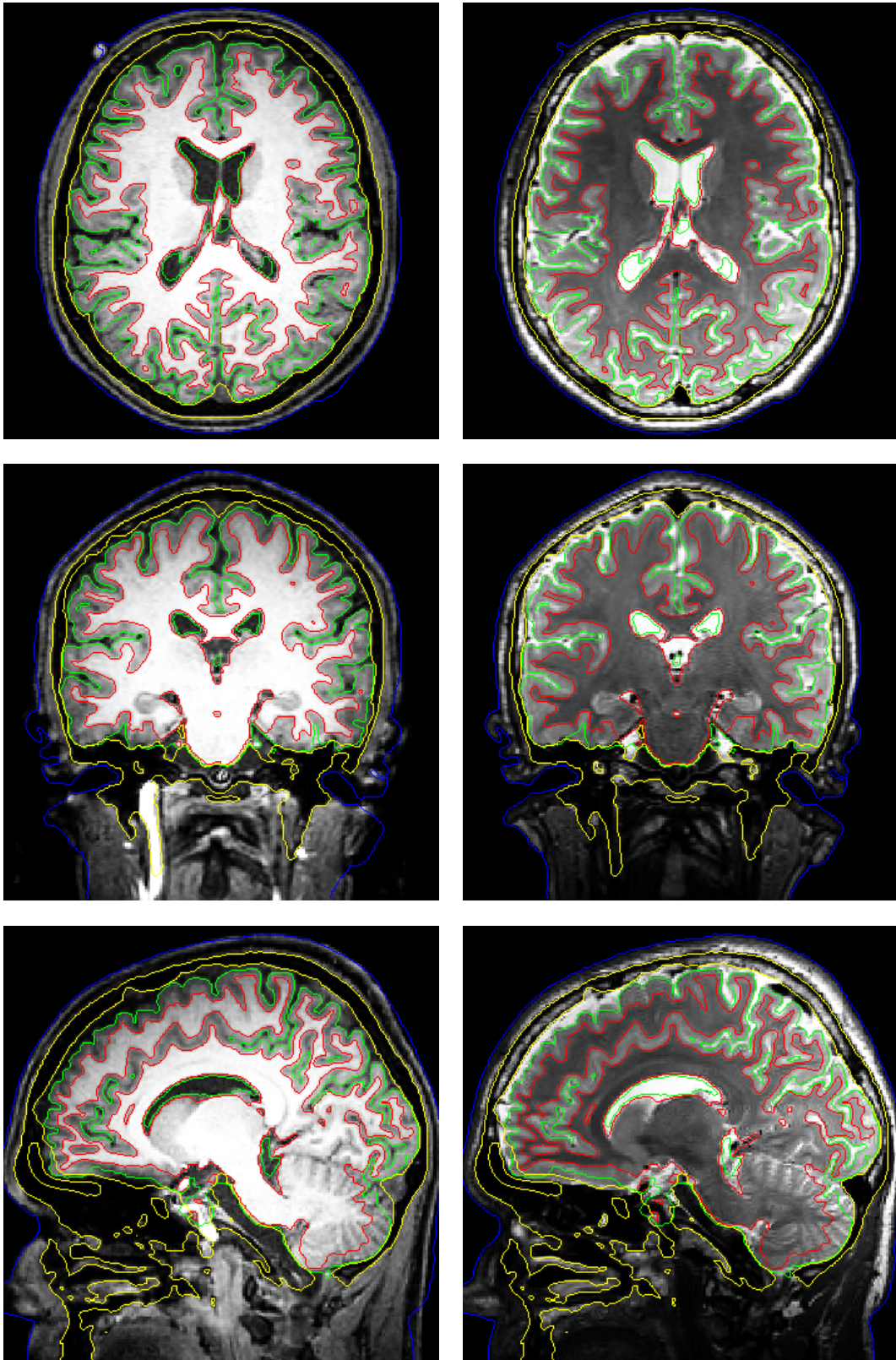


Figure 6.15: Projections (coronal, frontal, sagittal) of the WM, GM, CSF, skull, skin interfaces inside the T1 (Left) and the T2 MRIs (Right).

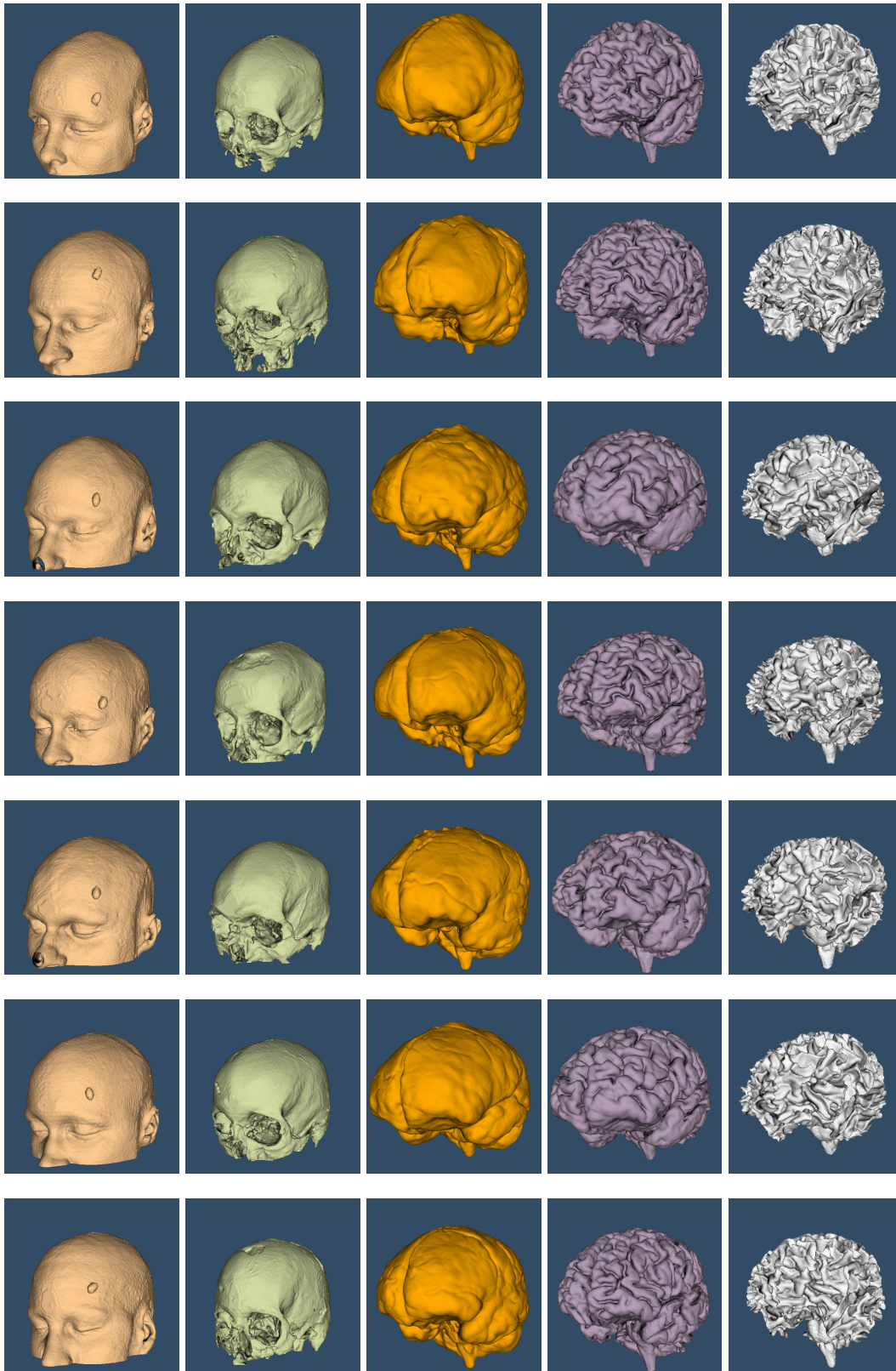


Figure 6.16: 3D Results of 6 different subjects. From left to right : skin, skull, CSF, GM and WM.

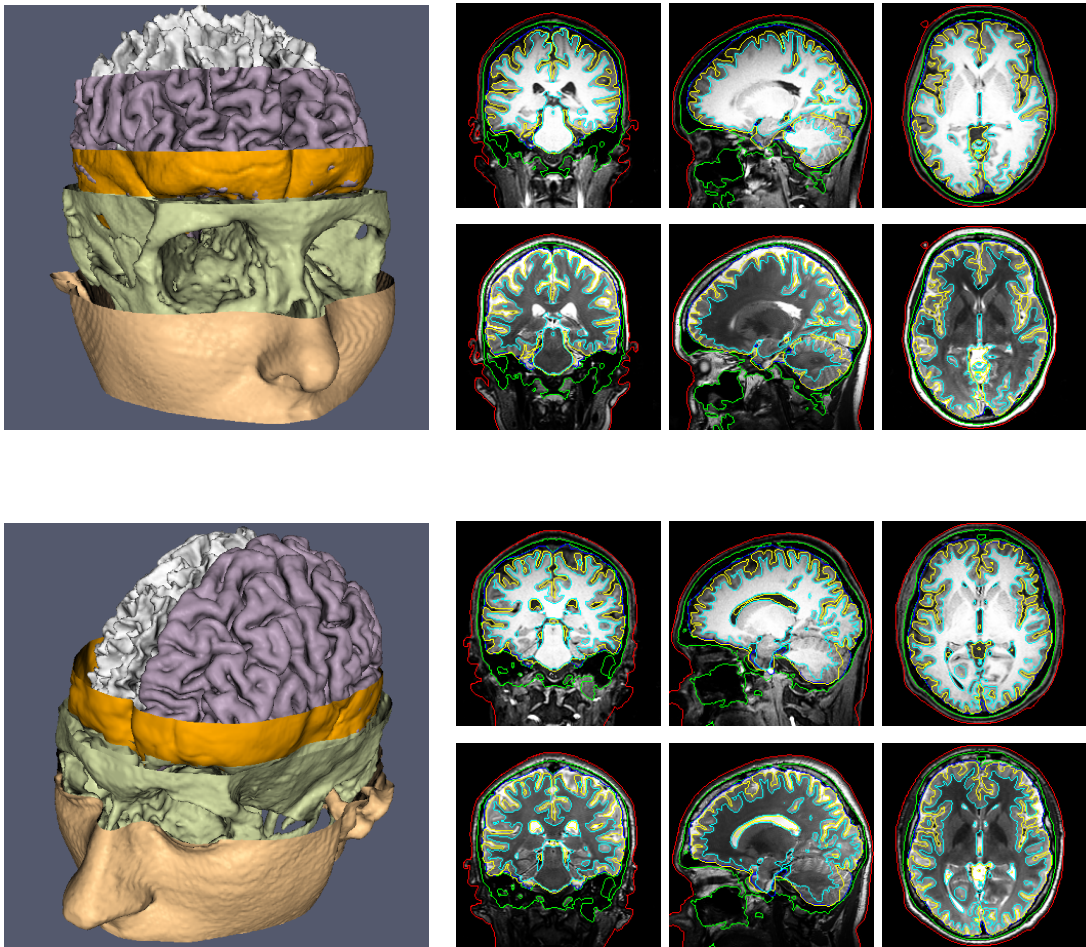


Figure 6.17: Results of 2 different subjects (Top and Bottom), with their 3D representation (Left), and projections on their respective anatomical MRI (Right).

Our framework have been tested on 8 different subjects, and for each of them, the 5 anatomical structures segmented (Fig.6.16). All these segmentations have been done automatically using the same set of parameters. The only interaction corresponds to the initialization of the front inside the brain (for the CSF segmentation): a sphere can be initialized almost anywhere in the brain, and our model will be robust enough to make it evolve correctly. Moreover, since both T1 and T2 MRI are aligned with the Talairach stereotactical coordinate system, a general initialization can be found that will be valid for all the registered subjects. The results are particularly good, especially for the skull that is quite uncommon in standard methods.

All these segmentation benefit from the GPU implementation of the local-statistic based model from chapter 5, allowing very fast computations (less than 1 hour to segment the 5 structures).

In Fig. 6.15 can be seen slices of the 5 segmentations in both T1 and T2 MRIs. The

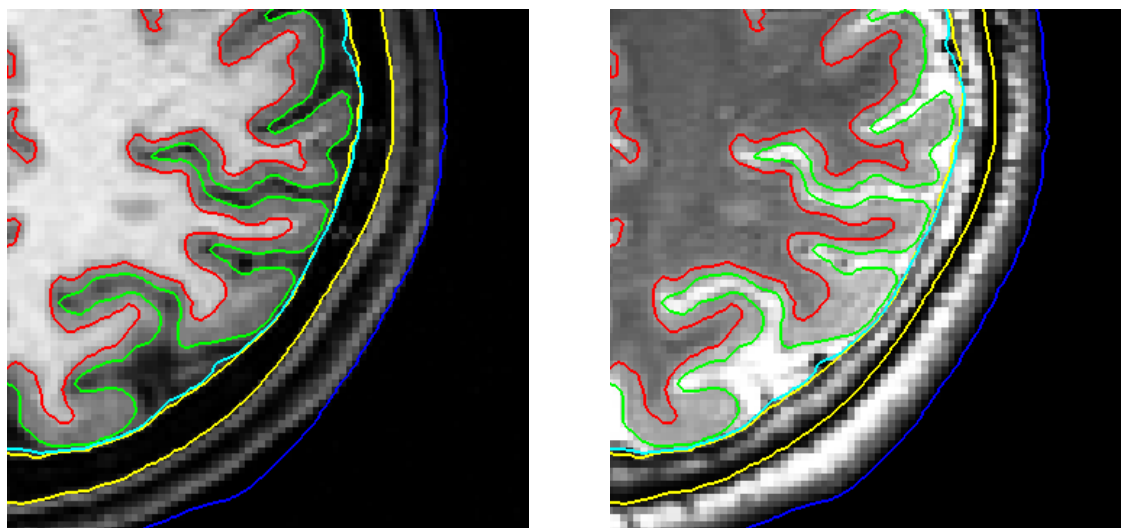


Figure 6.18: Zoom on projections of the 5 segmentations in the T1 MRI (left) and the T2 MRI (right).

5 structures are indeed nested inside each other, and the GM satisfy its spherical topology. In Fig. 6.17 can be seen the 5 interfaces simultaneously, on 2 different subjects. Finally, Fig. 6.18 shows zoom parts of the segmentation on the T1 and T2 MRIs. As can be seen, the topology constraint works quite well, as it prevents self intersections of the front.

6.6 CONCLUSION

This work has shown that rather good head models can be obtained using combinations of T1 and T2 MRIs. These images provide complementary informations and can conveniently be acquired in a single MR session.

The segmentation method is based on local statistics of images, which allows for thin structures extractions and naturally copes with the intensity drifts that arise in MR images. However, instead of a being a global multi-partitioning segmentation model (like EM classifiers for example), this model remains a local bi-partitioning segmentation model, and is not quite suited to segment several regions simultaneously. Hence, our framework is based on consecutive steps to make the front evolve correctly. Our segmentation model works very well in the majority of cases, but is much more powerfull in “refining process“, in order to adjust locally the contour from a close initialization. Therefore, we think our framework would benefit from using first a more global criteria, and then refining the results with our model on bimodal T1- and T2- weighted MRIs. However, avoiding classical preprocessing steps (skull-stripping, classification) is a strong advantage of our approach.

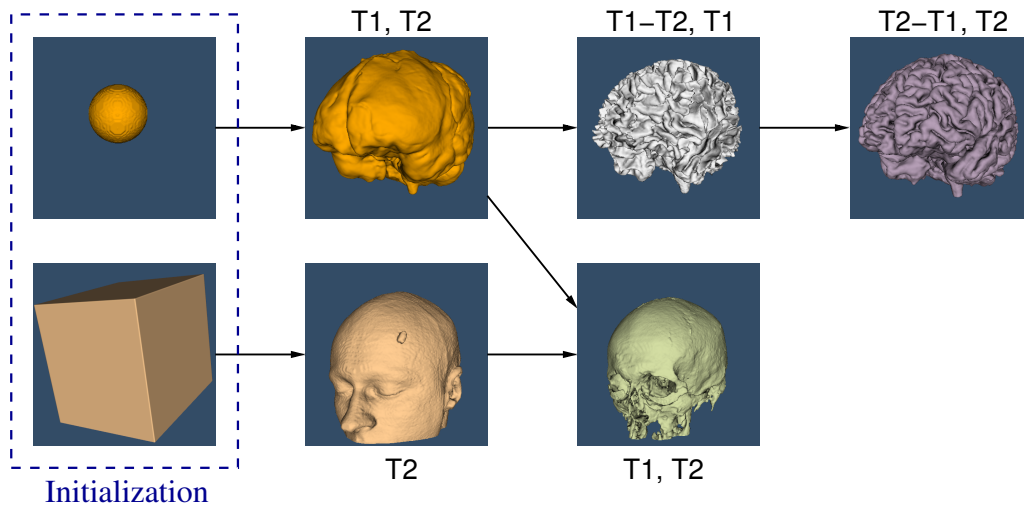


Figure 6.19: Consecutive steps of our model, using different combinations of T1 and T2 MRIs (see text).

In our framework, 5 structures are segmented consecutively: the skin, the CerebroSpinal Fluid (CSF), the White Matter (WM), the Grey Matter (GM), and the skull (Fig.6.19).

- The **skin** has been successfully segmented from the **T2** MRI, and provides highly detailed results especially for thin structures such as the nose and the hears. Segmentation of these structures from the T1 MRIs is rather approximative since they confound themselves with the background.
- The **CSF** has been successfully segmented from a combination of **T1 and T2** MRIs, and provides highly accurate results of the inside of the skull. Very nice details are extracted, especially vascularizations of the brain. This segmentation plays the role of a skull-stripping.
- The **WM** has been segmented from combined “**T1-T2**” first, then on the **T1** alone. Maybe one problem would be that the sharpest gyri are not completely segmented due to the regularity constraint (Fig. 6.18). Otherwise, the results are pretty good.
- The **GM** has been segmented from combined “**T2-T1**” first, then on the **T2** alone. A topological constraint based on skeleton extraction has been proposed, and is efficient and easy to implement. Moreover, it allows the simultaneous extraction of sulcal folds, which are quite important for morphological studies of the brain.
- Finally, the **skull** has been segmented from a combination of **T1 and T2** MRIs. The skull is the most difficult structure to extract and maybe the more interesting for electromagnetic simulations (EEG, MEG, etc). Few models manage

to extract it correctly from MRIs, due to rather incomplete informations. Our method produce very encouraging results, since the skulls are almost perfectly segmented. However, the proposed method still mixes the skull with other compartments in some places (eg air, veins and ligaments). In many cases (air and ligaments around the mouth, veins in the bottom part of the head), these defaults are not very important for the head modelling as they occur far from the area of interest and have thus little influence. It might still be interesting to remove them for the simplicity of the model or for other applications. However, the superior sagittal sinus (the vein that goes along the separation of the two cortex hemisphere) is currently segmented as being part of the skull⁶.

Our framework has been successfully applied to 8 different real acquisitions and shows quite nice details. The timings are quite reasonable thanks to an efficient GPU implementation.

Future works include incorporation of a more global multi-region segmentation for the brain, followed by a refinement process using our local segmentation model. This approach would take in consideration the prior knowledge that the brain is composed of only three separate tissues having a different MR response: WM, GM, CSF; which is somehow neglected in our approach.

An interesting addition would be to incorporate a more elaborate shape prior term, that would characterize the particular shape of the cortex: it would ensure its layered nature (but with a stronger flexibility than classical coupled surface approaches [117, 63, 35]), as well as its spherical topology and the fact that cortical folds are espaced with a rather constant distance (Fig.6.10). The cortical folds would be detected, and evolved in order to take into account this spacing constraint. Their evolutions would influence evolution of the GM segmentation, and thus would produce more accurate results. We think this can be done with a **higher order active contour model** [87, 45], that is quite powerful to elaborate more complex shape constraints.

Finally, the segmentation of the skull provides quite encouraging results, but some artefacts still remain. Correcting these artefacts could be done using some post-processing method, eg an evolution starting with the segmentations obtained with the proposed method but based on other T1 and T2 image combinations. Some preliminary tests show that this seems feasible.

Acknowledgements: We would like to thanks Alfred Anwander from *Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany* for providing all the couples of registered T1 and T2 MRI.

⁶It is interesting to notice that this vein is hardly distinguishable in T1 MRIs, so that most segmentations based on such images wrongly aggregate this head area with the grey matter.

CONCLUSION

In this thesis, we have proposed new models for image segmentation in the purpose of creating anatomical head models from Magnetic Resonance Imaging (MRIs).

We first presented an overview of different methods for image segmentation. Although discrete models (Graph-Cuts, etc), are now widely used in image segmentation, models based on hypersurface evolutions are still the best choice for incorporating some knowledge about the shape of the desired segmentation. In our case, preservation of the topology for the cortex segmentation is crucial, and cannot be done easily using a discrete segmentation model.

We then emphasized the fact that segmentation models based on hypersurfaces are classified in two categories (edge-based and region-based), and none of these categories is robust to all of the MRI alterations (noise, inhomogeneities, diffuse contours).

Then, we proposed a novel implementation of the level-set method using quadrilateral finite elements. This representation aims at improving the robustness and the accuracy of the model while providing a good sub-voxelic representation of the zero level-set. However, its implementation is quite difficult, and largely increases the complexity of the model. We did not use this implementation in the rest of the thesis due to this high complexity, and used instead a classical finite-difference based implementation, with some parallelization routines.

A new segmentation model based on local statistics that aimed in being robust to common MRIs alterations has been proposed. Its formulation unifies classical edge-based and region-based models, and its implementation is quite effective using fast recursive filters and modern GPUs. We then pointed out two major drawback of the model, and proposed two solutions. This model provides good results, and manages to perform both segmentation and restoration, by providing piecewise smooth partition of images.

Finally, we applied the previous segmentation model in the case of anatomical

head model creation. This consists in segmenting five structures of interest from anatomical MRIs: the White Matter (WM), the Grey Matter (GM), the CerebroSpinal Fluid (CSF), the skull and the scalp.

Creation of head models from MRIs has been widely studied in the last decade, and most methods are based on separate consecutive steps. Moreover, standard models cannot segment easily the skull. We propose a semi-automatic framework that segments each of the anatomical structures successively from bi-modal registered T1- and T2-weighted MRIs. This framework is robust to classical alterations in MRIs thanks to the previous segmentation model, and manage to extract the 5 structures of interest with a very good accuracy.

In summary, the main contributions of this thesis are:

- Implementing the level set method using quadrilateral finite elements.
- Proposing a novel and efficient local-statistic based segmentation model robust to MRI alterations. Pointing out two drawbacks of the local model, and proposing two solutions.
- Creating anatomical head model using the previous local-based model. Segmenting the skull from bi-modal MRIs. Proposing a simple and efficient topological constraint for the cortex segmentation.

Some perspective would be:

- Investigating another, more efficient, way to implement level-sets using finite elements (for example, avoiding to include a gradient descent inside another gradient descent).
- Segmenting other kind of images with the local-statistic based model, such as textures data, or high dimensional data.
- Incorporating some global criteria for the segmentation of the brain, and adding some prior shape knowledge for the cortex segmentation, using higher order active contours.

Neurosciences have dramatically improved in the last few years, mainly due to the apparition of more and more detailed images, and new non-invasive functional techniques allowing to distinct with precision the various neural activities in the cortex. Moreover, a very recent imaging technique called **Diffusion Tensor Imaging** has been introduced. It allows to discern with precision the different nervous fibers composing the white matter. Coupling these techniques opens a wide range of new research concerning the understanding of the brain, since it allows to visualize the different actived cortical zones, and to “backtrack” this activity by following their

nervous fibers through the white matter. Head model creation plays a very important role in these research, as it corresponds to the “first step”: localizing the different structures of interest, especially the cerebral cortex. Accuracy of the cortex is crucial to determine precisely in which cortical zone belong a specified neural activity.

Although the brain is without a doubt the most complex anatomical structure of the human body, these neuroscience techniques are very encouraging, and are certainly at the premises of a complete understanding of the brain.

CONCLUSION (VERSION FRANÇAISE)

Dans cette thèse, nous avons proposé quelques contributions à la segmentation d'images, dans le but de créer des modèles anatomiques de la tête à partir d'Images à Résonance Magnétique (IRMs).

Nous avons d'abord brièvement présenté un aperçu des différentes méthodes en segmentation d'images. Bien que les modèles discrets (Graph-cut, etc), sont aujourd'hui largement utilisés en segmentation, les modèles basés sur des évolutions d'hypersurface sont encore le meilleur choix pour incorporer certaines connaissances a-priori sur la forme de la segmentation souhaitée. Dans notre cas, la préservation sphérique de la topologie du cortex est cruciale, et ne peut être mise en place facilement en utilisant des modèle discrets.

Nous avons ensuite insisté sur le fait que les modèles basés sur des évolutions d'hypersurfaces sont classés en deux catégories (basés contours et basés régions), et aucune de ces catégories n'est robuste à l'ensemble des altérations présentes dans les IRMs (bruit, inhomogénéités, contours diffus).

Nous avons ensuite proposé une nouvelle implémentation pour la méthode des ensemble de niveaux, en utilisant des éléments finis quadrilatéraux. Cette représentation vise à améliorer la robustesse et la précision du modèle tout en offrant une bonne représentation sous-voxellique du niveau zéro. Toutefois, sa mise en œuvre est très difficile, et augmente largement la complexité du modèle. Nous n'avons pas utilisé cette implémentation dans le reste de cette thèse en raison de son importante complexité. À la place, nous avons opté pour un modèle classique à base de différences finies, avec quelques routines de parallélisation.

Nous nous sommes ensuite focalisés sur des modèles de segmentations d'image, et avons proposé un nouveau modèle basé sur des statistiques locales, robuste à toute les altérations dont souffrent les IRMs. Sa formulation unifie les modèles basés contours et les modèles basés régions, et son implémentation est particulièrement

rapide à base de filtres récursifs et d'implémentation GPU. Nous avons ensuite mis en évidence deux inconvénients majeurs, et avons proposés deux solutions. Ce modèle a fournis de bons résultats, en accomplissant simultanément la segmentation et la restauration d'images (les résultats produit sont des images lisses par morceaux).

Enfin, nous avons appliqué le précédent modèle de segmentation dans le but de créer des modèles anatomiques de la tête à partir d'IRMs. Cela consiste à segmenter cinq structures anatomiques d'intérêt dans les IRMs: la matière blanche (MB), la matière grise (MG), le liquide céphalo-rachidien (LCR), le crâne et le cuir chevelu.

Les créations automatiques de modèles anatomiques de tête à partir d'IRMs ont largement été étudiés au cours des dernières années, et la plupart sont fondées sur différentes étapes consécutives. En outre, les modèles classiques ne peuvent facilement détecter le crâne à cause d'un manque de donnée concernant sa localisation. Nous avons proposé un cadre semi-automatique qui segmente chaque structure anatomique de manière successive à partir d'IRMs bi-modales T1 et T2 recalées. Ce cadre est robuste aux altérations classiques des IRMs, et permet d'extraire avec une très bonne précision les 5 structures d'intérêt.

En résumé, les principales contributions de cette thèse sont:

- Implémentation de la méthode des ensemble de niveaux à l'aide d'éléments finis quadrilatéraux.
- Introduction d'un nouveaux modèle de segmentation basé sur des statistiques locales. Mise en évidence de 2 inconvénients majeurs dans le modèle, et proposition de 2 solutions.
- Création de modèle anatomique de la tête en utilisant le modele local. Segmentation du crane à partir d'images bi-modales. Introduction d'une nouvelle contrainte topologique simple et rapide pour la segmentation du cortex.

Cette thèse ouvre quelque perspectives, qui sont:

- Implémenter de manière plus rapide et efficace la méthode des ensemble de niveaux à l'aide d'éléments finis (par exemple, éviter d'inclure une descente de gradient à l'interieur d'une autre descente de gradient).
- Segmenter d'autre types d'images à l'aide du modèle de segmentation à base de statistiques locales, comme des images texturées, ou des images à haute dimension.
- Inclure des critères un peu plus globaux pour la segmentation du cerveau. Utiliser des contraintes de formes plus élaborées pour la segmentation du cortex, en utilisant des contours actifs d'ordre superieurs.

Les neurosciences se sont amélioré de façon spectaculaire au cours des dernières années, principalement en raison de l'apparition d'images anatomiques de plus en plus détaillées, et de nouvelles techniques non-invasives permettant de discerner avec précision les différentes activités neuronales du cortex. En outre, une technique d'imagerie appelée **Imagerie par Tenseur de Diffusion** a été très récemment introduite, et permet de discerner avec précision les différentes fibres nerveuses qui composent la matière blanche. Le couplage de ces technique ouvre un large éventail de recherches concernant la compréhension du cerveau, en permettant de visualiser les différentes zones corticales en activité, et de "suivre" cette activité à travers les fibres nerveuses de la matière blanche. Les modeles anatomiques de tête jouent un rôle crucial dans ces recherches, car il permettent de localiser les différentes structures anatomiques d'intérêt, en particulier le cortex cérébral. La précision du cortex est primordiale pour déterminer les zones corticale auxquelles appartiennent les activité neuronales détectées.

Bien que le cerveau soit sans doute la plus complexe des structures anatomiques du corps humain, ces techniques de neurosciences sont très encourageantes et sont certainement les prémices vers une compréhension globale du cerveau.

Appendix

DETAILS ABOUT FINITE-ELEMENT LEVEL-SET IMPLEMENTATION

This appendix describes some details about the implementation of the level-set method using Q_1 finite elements, depicted in chapter 4.

A.1 CALCULUS OF BASIS FUNCTIONS _____

Basis functions are functions associated to each vertice of the mesh. (Fig. 4.5). The i^{th} basis function φ_i associated to the i^{th} vertices V_i is defined as follows:

- $\varphi_i : \Omega \rightarrow \mathbb{R}$ continuous on Ω
- $\varphi_i(V_j) = \delta_{ij}$ (1 if $i = j$, 0 if $i \neq j$)

Therefore, we can restrict the computation of φ_i to the elements adjacent to vertex V_i . This computation can be done using the following method:

- let e be an element of the mesh, defined by vertices V_1, V_2, \dots, V_n
- the restriction of the basis function φ_1 associated to vertex V_1 on the element e is defined as follows:

$$\varphi_1(r)_e = \frac{|r V_2 \dots V_n|}{|V_1 \dots V_n|}$$

where the points r, V_1, V_2, \dots, V_n are expressed in their homogeneous notation.

For example, if the element e is a $2D$ triangle defined by vertices $V_1 = (1, 0)$, $V_2 = (1, 1)$, $V_3 = (0, 1)$, we will have:

$$\varphi_1(x, y)_e = \frac{\begin{vmatrix} x & 1 & 0 \\ y & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}}, \quad \varphi_2(x, y)_e = \frac{\begin{vmatrix} 1 & x & 0 \\ 0 & y & 1 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}} \quad \text{et} \quad \varphi_3(x, y)_e = \frac{\begin{vmatrix} 1 & 1 & x \\ 0 & 1 & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}}$$

A.2 Q_1 BASIS FUNCTIONS

We used the method of appendix A.1 to compute the basis functions of $2D$ (square) and $3D$ (cube) regular Q_1 elements of size 1. Note that the gradient of these basis functions inside the elements is not constant, in opposition to P_1 elements.

The formulation of these basis function will help computing the terms of matrices A_1 and A_2 depicted in Section 4.3.6.

A.2.1 $2D$ basis functions

The $2D$ basis functions φ associated to the vertex $V = (a, b)$ is defined of the 4 elements adjacent to V . The restriction of φ inside these 4 elements is shown in Fig. A.1. These basis functions are characterized by their particular shape, in form of “pagodas” (Fig 4.6).

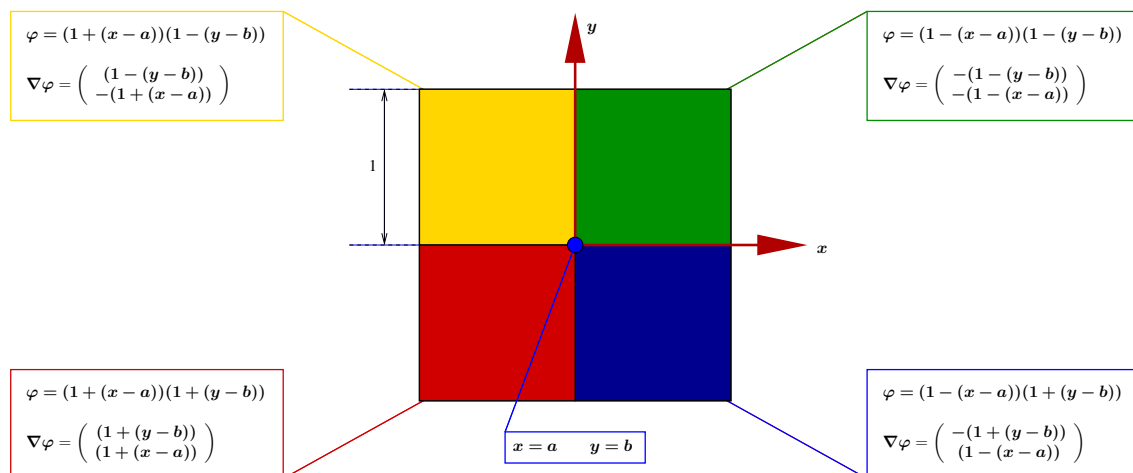


Figure A.1: Basis function of $2D$ regular Q_1 elements associated to vertex V .

A.2.2 3D basis functions

The 3D basis functions φ associated to the vertex $V = (a, b, c)$ is defined of the 8 elements adjacent to V . The restriction of φ inside these 8 elements is shown in Fig. A.2. A representation of this basis function can be seen in Fig. 4.10.

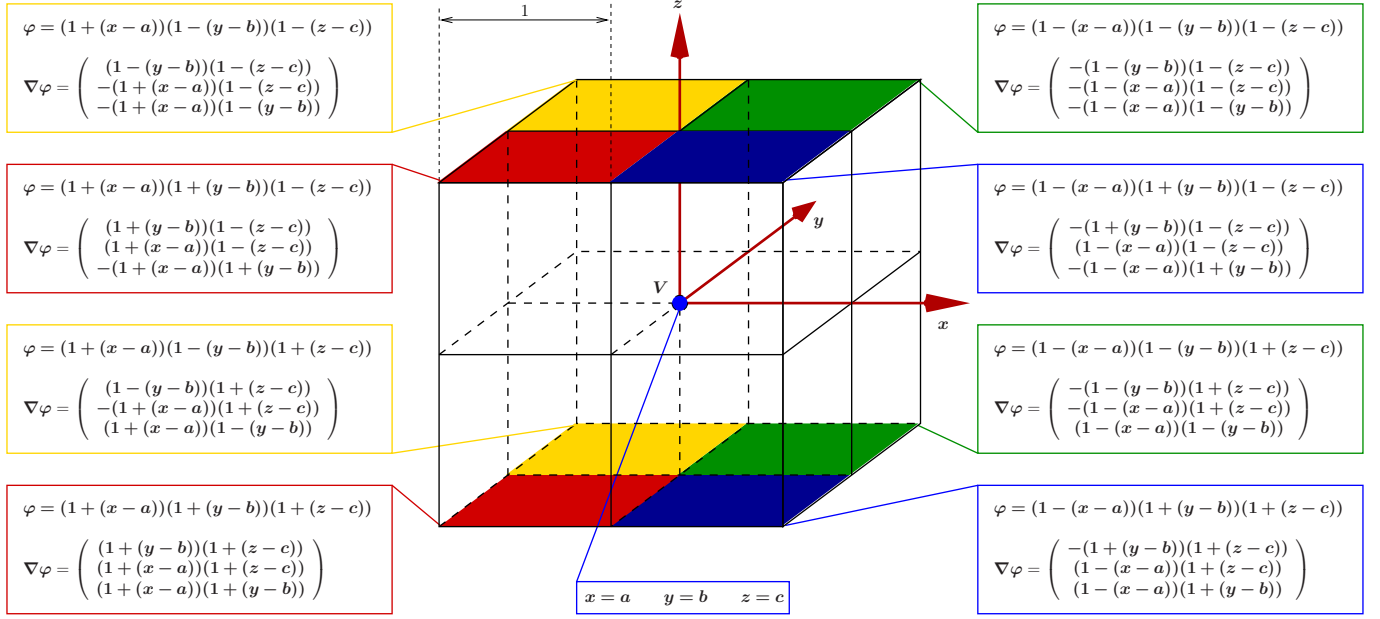


Figure A.2: Basis function of 3D regular Q_1 elements associated to vertex V .

A.3 CALCULUS OF DISTANCE CONSTRAINT SYSTEM

The distance constraint equation:

$$(\nabla\mathbf{u} + \nabla\mathbf{v})^2 - 1 = 0 \quad (\text{A.1})$$

gives us the following system to solve, after some approximations:

$$\underbrace{2\tilde{\mathbf{u}}^T \mathbf{Q}^i \tilde{\mathbf{v}}}_{A_1^i} + \underbrace{\tilde{\mathbf{u}}^T \mathbf{Q}^i \tilde{\mathbf{u}} - \int_{\Omega} \varphi_i}_{-b_1^i} = 0 \quad i \in \{1, 2, \dots, n\}, \quad (\text{A.2})$$

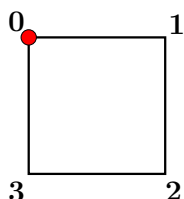
with \mathbf{Q}^i $n \times n$ matrix associated to the vertex i and defined by $\mathbf{Q}_{ab}^i = \int_{\Omega} \langle \nabla\varphi_a, \nabla\varphi_b \rangle \varphi_i$.

The next step is to use the basis functions of appendix A.2 to compute these specific integral. These computations are different for the 2D case and 3D case.

A.3.1 2D case

In the case of a 2D mesh with only one regular Q_1 element (therefore with 4 vertices), the matrices Q^i can be computed using integrals previous 2D Q_1 basis functions. The matrix Q^0 associated to the vertex 0 is defined as follows:

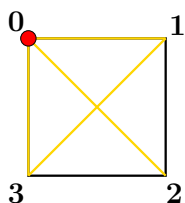
$$Q_{ab}^0 = \int_{\Omega} \langle \nabla \varphi_a, \nabla \varphi_b \rangle \varphi_0$$



$$Q^0 = \frac{1}{12} \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

By developing the term $2\tilde{u}^T Q^0 \tilde{v}$ using the matrix above, it appears that we can reformulate it in terms of difference between all possible pairs of vertex, which is directly linked to **edges** of the element. This formulation is a lot easier to implement, as the system can then be retrieved by looking at all possibles edges of a given mesh. In the following, we will use the notation $u_{ij} = (u_i - u_j)$ and $v_{ij} = (v_i - v_j)$.

For the vertex 0, this lead to the following notation:



$$2\tilde{u}^T Q^0 \tilde{v} = \frac{1}{12} \begin{pmatrix} u_{01} & u_{02} & u_{03} & u_{13} \\ v_{01} & v_{02} & v_{03} & v_{13} \end{pmatrix}^T$$

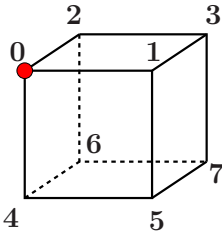
that can be computed by looking at the vertical, horizontal and diagonal edges.

This help us find the term $A_1^0 \tilde{v}$ of system A.2. The first term of b_1^0 is found in the same way and the second term ($\int_{\Omega} \varphi_i$) corresponds to the number of elements adjacent to vertex 0.

These notations are valid for vertex 0, but calculus confirms that this can be generalized for all the other vertices of the mesh, by looking at the same “adjacency” of edges. Moreover, calculus shows that if vertices 0 is adjacent to more elements, each element simply add its contribution in the same way to the equation induced by vertex 0 in the system A.2.

A.3.2 3D case

In the case of a 3D mesh with only one regular Q_1 element (therefore with 8 vertices), the matrices Q^i can be computed using integrals previous 3D Q_1 basis functions. The matrix Q^0 associated to the vertex 0 is defined as follows:

$$Q_{ab}^0 = \int_{\Omega} \langle \nabla \varphi_a, \nabla \varphi_b \rangle \varphi_0$$


$$Q^0 = \frac{1}{288} \begin{pmatrix} 27 & -3 & -3 & -5 & -3 & -5 & -5 & -3 \\ -3 & 15 & -5 & 1 & -5 & 1 & -3 & -1 \\ -3 & -5 & 15 & 1 & -5 & -3 & 1 & -1 \\ -5 & 1 & 1 & 7 & -3 & -1 & -1 & 1 \\ -3 & -5 & -5 & -3 & 15 & 1 & 1 & -1 \\ -5 & 1 & -3 & -1 & 1 & 7 & -1 & 1 \\ -5 & -3 & 1 & -1 & 1 & -1 & 7 & 1 \\ -3 & -1 & -1 & 1 & -1 & 1 & 1 & 3 \end{pmatrix}$$

By developping the term $2\tilde{u}^T Q^0 \tilde{v}$ using the matrix above, the same reformulations as the 2D case can be made, by expressing it in terms of difference between all possible pairs of vertex. In the following, we will use the same notation $u_{ij} = (u_i - u_j)$ and $v_{ij} = (v_i - v_j)$.

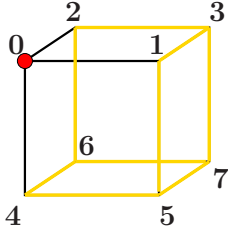
However, the 3D case is more complicated since several type of “edges” contribute differently to the reformulation of the system of equation. In a single element, there is a total of 28 “edges”: 12 vertical & horizontal edges ; 12 diagonal edges inside the 6 faces ; 4 “internal edges”.

For a given vertex, there are 4 types of contributions to its associated equation, depending on a specific “adjacency” of these edges: the 9 horizontal & vertical edges *not adjacent* to it; the 6 diagonal edges *adjacent* to it ; the 6 diagonal edges *not adjacent* to it ; and the other edges (horizontal, vertical, and internal edges). For the vertex 0, This lead to the following notation:

$$2\tilde{u}^T Q^0 \tilde{v} = \frac{1}{144} (-Q_a^0 + 5Q_b^0 + Q_c^0 + 3Q_d^0) ,$$

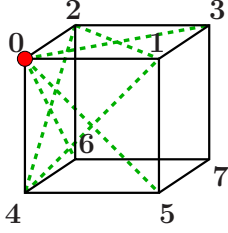
with

- Horizontal & vertical edges not adjacent to vertex 0



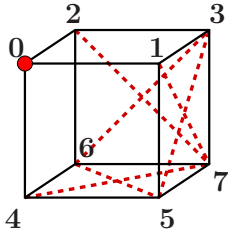
$$Q_a^0 = \begin{pmatrix} u_{13} & u_{15} & u_{23} & u_{26} & u_{37} & u_{45} & u_{46} & u_{57} & u_{67} \end{pmatrix}^T \\ \begin{pmatrix} v_{13} & v_{15} & v_{23} & v_{26} & v_{37} & v_{45} & v_{46} & v_{57} & v_{67} \end{pmatrix}$$

- Diagonal edges of adjacent faces to vertex 0



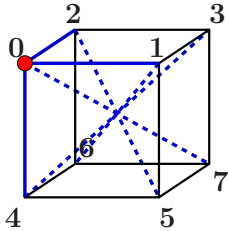
$$Q_b^0 = \begin{pmatrix} u_{03} & u_{05} & u_{06} & u_{12} & u_{14} & u_{24} \end{pmatrix}^T \\ \begin{pmatrix} v_{03} & v_{05} & v_{06} & v_{12} & v_{14} & v_{24} \end{pmatrix}$$

- Diagonal edges of non adjacent faces to vertex 0



$$Q_c^0 = \begin{pmatrix} u_{17} & u_{27} & u_{35} & u_{36} & u_{47} & u_{56} \end{pmatrix}^T \\ \begin{pmatrix} v_{17} & v_{27} & v_{35} & v_{36} & v_{47} & v_{56} \end{pmatrix}$$

- Horizontal & vertical edges adjacent to vertex 0 and internal edges



$$Q_d^0 = \begin{pmatrix} u_{01} & u_{02} & u_{04} & u_{07} & u_{16} & u_{25} & u_{34} \end{pmatrix}^T \\ \begin{pmatrix} v_{01} & v_{02} & v_{04} & v_{07} & v_{16} & v_{25} & v_{34} \end{pmatrix}$$

This help us find the term $A_1^0 \tilde{v}$ of system A.2. The first term of b_1^0 is found in the same way and the second term $(\int_{\Omega} \varphi_i)$ corresponds to the number of elements adjacent to vertex 0.

As for the 2D case, these notations are valid for vertex 0, but calculus confirms that this can be generalized for all the other vertices of the mesh, by looking at the same “adjacency”.

Therefore, all these formulae help us computing the system A.2 very easily, by looking at all elements in the mesh, and for each of them adding its contribution to its 8 vertices they include, by looking at all their possible “edges”.

A.4 CALCULUS OF EVOLUTION EQUATION SYSTEM

The evolution equation:

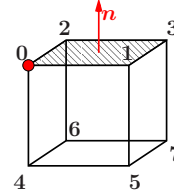
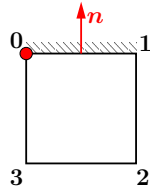
$$v + \Delta t (\epsilon \Delta(u + v) + \alpha) = 0 \quad (\text{A.3})$$

gives us the following system to solve:

$$\underbrace{\left[P^i - \Delta t \epsilon (-R^i + B^i) \right]}_{A_2^i} \tilde{v} - \underbrace{\Delta t \epsilon (-R^i + B^i)}_{b_2^i} \tilde{u} - \Delta t P^i \alpha = 0 \quad \forall i, \quad (\text{A.4})$$

where P^i , R^i and B^i are vectors of size n associated to the vertex i . In the case of only one regular Q_1 element with one of its faces belonging to the boundary $\partial\Omega$ of the domain Ω , the vectors P^i , Q^i and B^i can be computed using the previous Q_1 basis functions. For the 2D and 3D case, the vectors associated to the vertex 0 are defined as follows:

$$P_a^0 = \int_{\Omega} \varphi_a \varphi_0 \quad R_a^0 = \int_{\Omega} \langle \nabla \varphi_a, \nabla \varphi_0 \rangle \quad B_a^0 = \int_{\partial\Omega} \langle \nabla \varphi_a, \mathbf{n} \rangle \varphi_0$$



$$\begin{cases} P^0 = \frac{1}{36} \begin{pmatrix} 4 & 2 & 1 & 2 \end{pmatrix} \\ R^0 = \frac{1}{6} \begin{pmatrix} 4 & -1 & -2 & -1 \end{pmatrix} \\ B^0 = \frac{1}{6} \begin{pmatrix} 2 & 1 & -1 & -2 \end{pmatrix} \end{cases} \quad \begin{cases} P^0 = \frac{1}{216} \begin{pmatrix} 8 & 4 & 4 & 2 & 4 & 2 & 2 & 1 \end{pmatrix} \\ R^0 = \frac{1}{12} \begin{pmatrix} 4 & 0 & 0 & -1 & 0 & -1 & -1 & -1 \end{pmatrix} \\ B^0 = \frac{1}{36} \begin{pmatrix} 4 & 2 & 2 & 1 & -4 & -2 & -2 & -1 \end{pmatrix} \end{cases}$$

The reformulation of this system is more trivial than the previous one since R^i , B^i and P^i are no longer matrices but vectors. Therefore, we won't present in detail their computations. However, we proceed in the same way as for the distance constraint, such as each element in the mesh contributes to the equations induced by its vertices by looking at all the possible pairs of vertex.

However, note that the vectors B^i depends on a vector \vec{n} normal to the boundary, and is thus only defined on the boundary of the mesh. Only the elements having a face included in the boundary of the domain will contribute this term.

Therefore, the system A.4 can be computed very easily, by looking at all elements in the mesh, and for each of them adding its contribution to its vertices, by looking at all its possible "edges".

A.5 CONJUGATE GRADIENT METHOD ---

The conjugate gradient method is an iterative method to approximate the solution of a system of equations by minimizing an energy induced by the system [96].

This method is effective for systems of the form:

$$Ax = b$$

where x is a unknown vector, b is a known vector and A is a square, symmetric, positive definite matrix. If the matrix A is not symmetric, one can still solve the following system:

$$A^T Ax = A^T b$$

The basic idea of gradient descent is to take an initial solution as a “starting point”, and to consecutively improve this solution by updating its values in order to decrease the energy induced by the system.

This update is performed differently depending of the method:

- in **the steepest gradient descent**, this update is performed by going in the opposite direction of the gradient of the energy, with constant steps. By following this direction, the solution will converge toward the minimum of the energy, and thus will solve the initial system.
- **the steepest gradient descent with optimal step** proceeds in the same way, but the difference is that each step are optimally chosen to converge more rapidly to the solution (the further the solution is from the minimum of the energy, the larger the step will be) (Fig A.3).

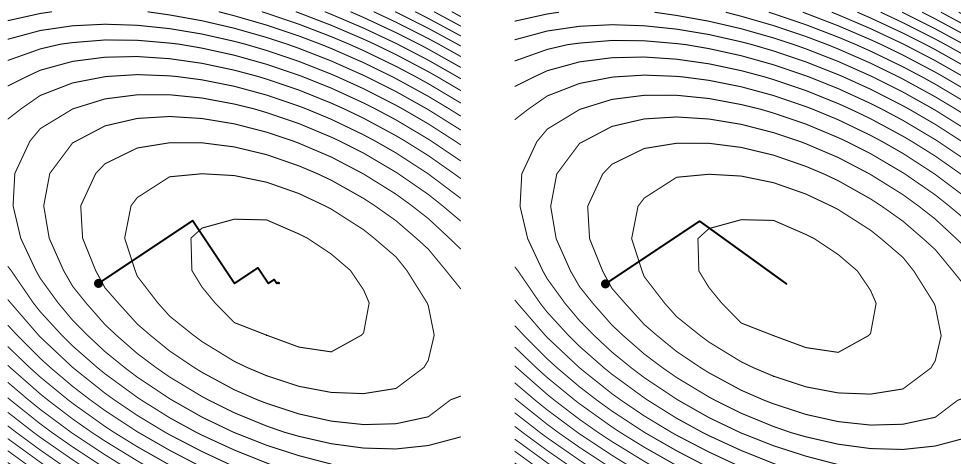


Figure A.3: Minima of the energy obtained with the steepest gradient descent with optimal steps (Left) and with the conjugate gradient method (Right).

- the **conjugate gradient method** proceeds in a similar way to the previous one, but the difference is that the direction is no more the gradient of the energy, but a more complicated direction depending on all the directions already taken in the descent. It has the advantage of converging to the optimal solution in at most n step, where n is the number of equations in the system (Fig A.3).

The conjugate gradient method is the most efficient algorithm, as it guarantee that the solution will be reached in at most n steps.

The algorithm below explains how to implement the conjugate gradient method, in order to solve a system of the form $Ax = b$.

Algorithm 3 Conjugate Gradient Method

initialize solution x_0

initialize direction $d_0 = r_0 = b - Ax_0$

for $i = 0$ to n **do**

compute step $\alpha_i = \frac{r_i^T r_i}{d_i^T A d_i}$

update solution $x_{i+1} = x_i + \alpha_i d_i$

if solution x_{i+1} good enough **then**

break

• $r_{i+1} = r_i - \alpha_i A d_i$

• $\beta_{i+1} = \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i}$

update direction $d_{i+1} = r_{i+1} + \beta_{i+1} d_i$

APPENDIX B

ENERGY DERIVATION OF THE PIECEWISE SMOOTH MODEL

This appendix describes the derivation of the local mean energy proposed in chapter 5, by using the shape gradient method.

B.1 DERIVATION OF THE ENERGY

We consider the following energy, for the bi-partitioning case:

$$E(\Omega, \bar{\Omega}) = \int_{\Omega} f(x, \Omega) dx + \int_{\bar{\Omega}} f(x, \bar{\Omega}) dx,$$

where

$$f(x, \Omega) = (I(x) - \mu_{\rho}(x, \Omega))^2 = \left(I(x) - \frac{G_1(x, \Omega)}{G_2(x, \Omega)} \right)^2, \quad (\text{B.1})$$

and

$$\begin{cases} G_1(x, \Omega) = \int_{\Omega} H_1(x, y) dy & \text{and} & H_1(x, y) = I(y) g_{\rho}(x - y) \\ G_2(x, \Omega) = \int_{\Omega} H_2(x, y) dy & \text{and} & H_2(x, y) = g_{\rho}(x - y) \end{cases} \quad (\text{B.2})$$

We recall the main theorem of shape gradient method presented in [3]:

Theorem B.1.0.1. *The Gâteaux derivative of a functional $J(\Omega) = \int_{\Omega} g(x, \Omega) dx$ in the direction of a vector field \mathbf{V} is:*

$$\langle \mathbf{J}'(\Omega), \mathbf{V} \rangle = \int_{\Omega} g_s(x, \Omega, \mathbf{V}) dx - \int_{\partial\Omega} g(x, \Omega) (\mathbf{V}(x) \cdot \mathbf{N}(x)) da(x)$$

Where $g_s(x, \Omega, \mathbf{V})$ is the shape derivative of $g(x, \Omega)$ in the direction of \mathbf{V} , $\partial\Omega$ is the boundary of Ω , \mathbf{N} is the unit inward normal to $\partial\Omega$ and da its area element.

In our case, for the bi-partitioning problem, we have,

$$\begin{aligned} \langle E'(\Omega, \bar{\Omega}), \mathbf{V} \rangle &= \int_{\Omega} f_s(x, \Omega, \mathbf{V}) dx + \int_{\bar{\Omega}} f_s(x, \bar{\Omega}, \mathbf{V}) dx \\ &\quad - \int_{\Gamma} (f(x, \Omega) - f(x, \bar{\Omega})) (\mathbf{V}(x) \cdot \mathbf{N}(x)) da(x) \end{aligned}$$

with

$$f_s(x, \Omega, \mathbf{V}) = f_{G_1}(x, \Omega, G_1, G_2) \langle G'_1(x, \Omega) \cdot \mathbf{V} \rangle + f_{G_2}(x, \Omega, G_1, G_2) \langle G'_2(x, \Omega) \cdot \mathbf{V} \rangle,$$

where f_{G_1} and f_{G_2} denote the partial derivative of (B.1) with respect to G_1 and G_2 . They can be expressed as:

$$\begin{cases} f_{G_1}(x, \Omega, G_1, G_2) = -\frac{2}{G_2(x, \Omega)} \left(I(x) - \frac{G_1(x, \Omega)}{G_2(x, \Omega)} \right) \\ f_{G_2}(x, \Omega, G_1, G_2) = \frac{2G_1(x, \Omega)}{G_2(x, \Omega)^2} \left(I(x) - \frac{G_1(x, \Omega)}{G_2(x, \Omega)} \right) \end{cases} \quad (\text{B.3})$$

and by using Theorem B.1.0.1,

$$\begin{cases} \langle G'_1(x, \Omega) \cdot \mathbf{V} \rangle = \int_{\Omega} H_{1s}(x, y, \mathbf{V}) dy - \int_{\Gamma} H_1(x, y) (\mathbf{V}(y) \cdot \mathbf{N}(y)) da(y) \\ \langle G'_2(x, \Omega) \cdot \mathbf{V} \rangle = \int_{\Omega} H_{2s}(x, y, \mathbf{V}) dy - \int_{\Gamma} H_2(x, y) (\mathbf{V}(y) \cdot \mathbf{N}(y)) da(y) \end{cases} \quad (\text{B.4})$$

Since H_1 and H_2 do not depend on Ω , we obtain $H_{1s} = 0$ and $H_{2s} = 0$.

Putting all the term together yields

$$\begin{aligned} f_s(x, \Omega, \mathbf{V}) &= f_{G_1} \langle G'_1 \cdot \mathbf{V} \rangle + f_{G_2} \langle G'_2 \cdot \mathbf{V} \rangle \\ &= \frac{2(I(x) - \mu_{\rho}(x, \Omega))}{\int_{\Omega} g_{\rho}(x - y) dy} \int_{\Gamma} g_{\rho}(x - y) (I(y) - \mu_{\rho}(x, \Omega)) (\mathbf{V}(y) \cdot \mathbf{N}(y)) da(y) \end{aligned}$$

and finally we obtain, by changing the integration order,

$$\begin{aligned} \langle E'(\Omega, \bar{\Omega}), \mathbf{V} \rangle &= \int_{\Omega} f_s(x, \Omega, \mathbf{V}) dx + \int_{\bar{\Omega}} f_s(x, \bar{\Omega}, \mathbf{V}) dx \\ &\quad - \int_{\Gamma} (f(y, \Omega) - f(y, \bar{\Omega})) (\mathbf{V}(y) \cdot \mathbf{N}(y)) da(y) \\ &= \int_{\Gamma} \left(q(y, \Omega) - q(y, \bar{\Omega}) - (I(y) - \mu_{\rho}(y, \Omega))^2 \right. \\ &\quad \left. + (I(y) - \mu_{\rho}(y, \bar{\Omega}))^2 \right) (\mathbf{V}(y) \cdot \mathbf{N}(y)) da(y) \end{aligned}$$

with

$$q(y, \Omega) = \int_{\Omega} \frac{2(I(x) - \mu_{\rho}(x, \Omega)) g_{\rho}(x - y) (I(y) - \mu_{\rho}(x, \Omega))}{\int_{\Omega} g_{\rho}(x - z) dz} dx$$

We finally get the following gradient descent:

$$\frac{\partial \Gamma}{\partial \tau}(x) = \left[\left(q(x, \Omega) - q(x, \bar{\Omega}) - (I(x) - \mu_\rho(x, \Omega))^2 + (I(x) - \mu_\rho(x, \bar{\Omega}))^2 \right) \right] \mathbf{N}(x)$$

B.2 IMPLEMENTATION

Each term in the gradient descent can be seen as a convolution by a Gaussian kernel of variance ρ . Therefore, the computation of the evolution can be done in a very fast way by using efficient Gaussian filter scheme.

By noting $(. * .) |_\Omega$ the convolution operator on a domain Ω , we have,

$$\mu_\rho(x, \Omega) = \frac{\int_\Omega g_\rho(x-y) I(y) dy}{\int_\Omega g_\rho(x-y) dy} = \frac{(g_\rho * I) |_\Omega(x)}{(g_\rho * 1) |_\Omega(x)} \quad (\text{B.5})$$

and

$$\begin{aligned} q(y, \Omega) &= \int_\Omega \frac{2(I(x) - \mu_\rho(x, \Omega)) g_\rho(x-y) (I(y) - \mu_\rho(x, \Omega))}{\int_\Omega g_\rho(x-z) dz} dx \\ &= I(y) \int_\Omega \frac{2(I(x) - \mu_\rho(x, \Omega))}{(g_\rho * 1) |_\Omega(x)} g_\rho(x-y) dx \\ &\quad - \int_\Omega \frac{2(I(x) - \mu_\rho(x, \Omega)) \mu_\rho(x, \Omega)}{(g_\rho * 1) |_\Omega(x)} g_\rho(x-y) dx \\ &= I(y) (g_\rho * q_1) |_\Omega(y) - (g_\rho * q_2) |_\Omega(y), \end{aligned}$$

with

$$q_1(x, \Omega) = \frac{2(I(x) - \mu_\rho(x, \Omega))}{(g_\rho * 1) |_\Omega(x)} \quad \text{and} \quad q_2(x, \Omega) = \frac{2(I(x) - \mu_\rho(x, \Omega)) \mu_\rho(x, \Omega)}{(g_\rho * 1) |_\Omega(x)}$$

We can compute the domain convolution for the bi-partitioning case in a level-set formulation:

$$\begin{cases} (g_\rho * f) |_\Omega(x) = (g_\rho * H_\sigma(\Phi) f)(x) \\ (g_\rho * f) |_{\bar{\Omega}}(x) = (g_\rho * (1 - H_\sigma(\Phi)) f)(x) \end{cases} \quad (\text{B.6})$$

where Φ denotes the level set function separating the two regions Ω and $\bar{\Omega}$, and H_σ is a regularized heavyside function.

Experimentations have shown that the auxiliary terms $q(x, \Omega)$ have indeed an influence in the evolution of the contour, especially concerning the energy

minimization. It has been shown that without these terms in the derivative, the energy sometimes slightly increase, but nonetheless converge toward the same local minima. Therefore, for the sake of simplicity and efficiency, these terms can be removed from the derivative.

At last, the bi-partitioning energy can be minimized using the following gradient-descent of coupled equation, in level-set notation:

$$\left\{ \begin{array}{l} \mu_\rho(x, \Omega) = \frac{(g_\rho * H_\sigma(\Phi) I)(x)}{(g_\rho * H_\sigma(\Phi))(x)} \\ \mu_\rho(x, \bar{\Omega}) = \frac{(g_\rho * (1 - H_\sigma(\Phi)) I)(x)}{(g_\rho * (1 - H_\sigma(\Phi)))(x)} \\ \frac{\partial \Phi}{\partial t}(x) = ((I(x) - \mu_\rho(x, \bar{\Omega}))^2 - (I(x) - \mu_\rho(x, \Omega))^2) |\nabla \Phi|. \end{array} \right. \quad (\text{B.7})$$

Bibliography

- [1] L. Ambrosio and V. Tortorelli. On the approximation of free discontinuity problems. *Bolletino della Unione Matematica Italiana*, 6-B:105–123, 1992.
- [2] Jungha An, Mikael Rousson, and Chenyang Xu. γ -convergence approximation to piecewise smooth medical image segmentation. In Nicholas Ayache, S'ebastien Ourselin, and Anthony Maeder, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, volume 4792 of *LNCS*, pages 495–502. Springer, 2007.
- [3] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM Journal of Applied Mathematics*, 63(6):2128–2154, 2003.
- [4] C. Baillard, P. Hellier, and C. Barillot. Segmentation of 3d brain structures using level sets and dense registration. In *MMBIA '00: Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, page 94. IEEE Computer Society, 2000.
- [5] D.C. Ballard and C.M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [6] Timothy J. Barth and James A. Sethian. Numerical schemes for the hamilton-jacobi and level set equations on triangulated domains. *J. Comput. Phys.*, 145(1):1–40, 1998.
- [7] Richard M. Leahy Belma Dogdas, David W. Shattuck. Segmentation of skull and scalp in 3-d human mri using mathematical morphology. In *Human Brain Mapping*, volume 26, pages 273–285, 2005.
- [8] S.M. Blinkov and I.I. Glezer. *The Human Brain In Figures and Tables*. Plenum, New York, 1968.
- [9] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, volume 1, pages 26–33, 2003.

- [10] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.
- [11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), 2001.
- [12] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001.
- [13] T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth Mumford-Shah functional. In F. Sgallari, A. Murli, and N. Paragios, editors, *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*, volume 4485 of *LNCS*, pages 203–213. Springer, May 2007.
- [14] T. Brox and D. Cremers. On local region models and a statistical interpretation of the piecewise smooth mumford-shah functional. *International Journal of Computer Vision*, 2008.
- [15] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.
- [16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. Technical report, HP Labs, September 1994. A shorter version appeared at 5th ICCV'95 - Boston.
- [17] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *The International Journal of Computer Vision*, 22(1):61–79, 1997.
- [18] A. Chambolle and G. Dal Maso. Discrete approximations of the Mumford–Shah functional in dimension two. Technical Report 9820, Université Paris Dauphine, Ceremade, 1998.
- [19] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.
- [20] T. Chan and L. Vese. A level set algorithm for minimizing the Mumford–Shah functional in image processing. *1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 161–168, 2001.
- [21] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven, and O. Faugeras. Generalized gradients: Priors on minimization flows. *The International Journal of Computer Vision*, 73(3):325–344, Jul 2007.

- [22] David L. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106:77–91, 1993.
- [23] L.D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53:211–218, 1991.
- [24] Y. Cointepas, J.-F. Mangin, Line Garnero, J.-B. Poline, and H. Benali. Brain-VISA: Software platform for visualization and analysis of multi-modality brain data. In *Proc. 7th HBM*, page S98, Brighton, United Kingdom, 2001.
- [25] M.G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton–Jacobi equations. *Trans. AMS*, 277:1–43, 1983.
- [26] Daniel Cremers, Mikaël Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.
- [27] Ruppert Curwen and Andrew. Blake. Dynamic contours : Real-time active splines. In Andrew Blake and Alan Yuille, editors, *Active Vision*, chapter III, pages 39–58. The MIT Press, 1993.
- [28] C.A. Davatzikos and R.N. Bryan. Using a deformable model to obtain a shape representation of the cortex. *IEEE Transactions on Medical Imaging*, 15(6):785–795, 1996.
- [29] R. Deriche. Recursively implementing the gaussian and its derivatives. Technical Report 1893, INRIA, Unité de Recherche Sophia-Antipolis, 1993.
- [30] A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh-Taylor instability. *Lecture Notes in Mathematics*, 771:145–159, 1979.
- [31] O. Faugeras, G. Adde, G. Charpiat, C. Chefd’Hotel, M. Clerc, T. Deneux, R. Deriche, G. Hermosillo, R. Keriven, P. Kornprobst, J. Kybic, C. Lenglet, L. Lopez-Perez, T. Papadopoulo, J.-P. Pons, F. Ségonne, B. Thirion, D. Tschumperlé, T. Viéville, and N. Wotawa. Variational, geometric, and statistical methods for modeling brain anatomy and function. *NeuroImage*, 23S1:S46–S55, 2004. Special issue: Mathematics in Brain Imaging - Edited by P.M. Thompson, M.I. Miller, T. Ratnanather, R.A. Poldrack and T.E. Nichols.
- [32] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, 1962.
- [33] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

- [34] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.
- [35] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Cortex segmentation: a fast variational geometric approach. *IEEE Transactions on Medical Imaging*, 21(12):1544–1551, 2002.
- [36] J. Gomes. *Implicit representations of evolving manifolds in computer vision*. PhD thesis, Université de Nice Sophia-Antipolis, 2001.
- [37] J. Gomes and O. Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223, 2000.
- [38] Leo Grady. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, Nov. 2006.
- [39] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation. In J. Duncan and G. Gerig, editors, *MICCAI 2005*, number 2 in LNCS 3750, pages 773–780. MICCAI Society, Springer, 2005.
- [40] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 1989.
- [41] X. Han, D.L. Pham, D. Tosun, M.E. Rettmann, C. Xu, and J.L. Prince. CRUISE: Cortical reconstruction using implicit surface evolution. *NeuroImage*, 23(3):997–1012, 2004.
- [42] X. Han, C. Xu, U. Braga-Neto, and J.L. Prince. Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm. *IEEE Transactions on Medical Imaging*, 21(2):109–121, 2002.
- [43] X. Han, C. Xu, and J.L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.
- [44] K. Held, E.R. Kops, B.J. Krause, W.M. Wells, and R. Kikinis. Markov random field segmentation of brain MR images. *IEEE Transactions on Medical Imaging*, 16(6):878–886, 1997.
- [45] P. Horvath, I. H. Jermyn, Z. Kato, and J. Zerubia. A higher-order active contour model of a ‘gas of circles’ and its application to tree crown extraction. *Pattern Recognition*, 2009.
- [46] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841, October 2002.

- [47] James R. Brookeman John P. Mugler III. Three-dimensional magnetization-prepared rapid gradient-echo imaging (3d mp rage). *Magnetic Resonance in Medicine*, 15(1):152–157, 1990.
- [48] O. Juan, R. Keriven, and G. Postelnicu. Stochastic motion and the level set method in computer vision: Stochastics active contours. *International Journal of Computer Vision*, 69(1):7–25, August 2006.
- [49] Olivier Juan and Yuri Boykov. Active graph cuts. In *CVPR*, volume 1, pages 1023–1029, 2006.
- [50] C.Y. Kao, S. Osher, and Y. Tsai. Fast sweeping methods for a class of static hamilton-jacobi equations. CAM 02-66, UCLA, 2003.
- [51] Tina Kapur, W. Eric L. Grimson, and Ron Kikinis. Segmentation of brain tissue from mr images. In *CVRMed '95: Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, pages 429–433. Springer-Verlag, 1995.
- [52] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *The International Journal of Computer Vision*, 1(4):321–331, 1987.
- [53] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of the 5th International Conference on Computer Vision*, pages 810–815, jun 1995.
- [54] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: from phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996.
- [55] Junmo Kim, John Fisher, Anthony Yezzi, Mujdat Cetin, and Alan Willsky. Nonparametric methods for image segmentation using information theory and curve evolution. In *IEEE International Conference on Image Processing*, pages 797–800, September 2002.
- [56] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.
- [57] Shawn Lankton, Delphine Nain, Anthony Yezzi, and Allen Tannenbaum. Hybrid geodesic region-based curve evolutions for image segmentation. In Jiang Hsieh and Michael J. Flynn, editors, *Medical Imaging 2007: Physics of Medical Imaging*, volume 6510, page 65104U. SPIE, 2007.
- [58] C. Le Guyader and L.A. Vese. Self-repelling snakes for topology-preserving segmentation models. *Image Processing, IEEE Transactions on*, 17(5):767–779, 2008.

- [59] C. Li, R. Huang, Z. Ding, C. Gatenby, D. Metaxas, and J. Gore. A variational level set approach to segmentation and bias correction of medical images with intensity inhomogeneity. In *Proceedings of Medical Image Computing and Computer Aided Intervention (MICCAI)*, volume LNCS 5242, Part II, pages 1083–1091, 2008.
- [60] C. Li, C. Kao, J. C. Gore, and Z. Ding. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Imag. Proc.*, 17(10):1940–1949, 2008.
- [61] C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: A new variational formulation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 430–436. IEEE Computer Society, 2005.
- [62] Tony. Lindeberg. Feature detection with automatic scale selection. *The International Journal of Computer Vision*, 30(2):77–116, 1998.
- [63] D. MacDonald, N. Kabani, D. Avis, and A.C. Evans. Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *NeuroImage*, 12(3):340–356, 2000.
- [64] R. Malladi and J. A. Sethian. A real-time algorithm for medical shape recovery. In *In Proceedings of International Conference on Computer Vision*, pages 304–310, 1998.
- [65] R. Malladi, J.A. Sethian, and B.C. Vemuri. A topology independent shape modeling scheme. *SPIE*, 2031:246, 1993.
- [66] J-F. Mangin, O. Coulon, and V. Frouin. Robust brain segmentation using histogram scale-space analysis and mathematical morphology. In W. M. Wells, A. Colchester, and S. Delp, editors, *Proc. 1st MICCAI*, LNCS-1496, pages 1230–1241, MIT, Boston, October 1998. Springer Verlag.
- [67] T. McInerney and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging*, 18(10):840–850, 1999.
- [68] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73–91, 2000.
- [69] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: Topology, geometry and deformation. *Computer Vision and Image Understanding*, 19(14):1023–1040, 2001.
- [70] J.M. Morel and S. Solimini. Segmentation of images by variational methods: A constructive approach. *Rev. Math. Univ. Complut. Madrid*, 1:169–182, 1988.

- [71] J.M. Morel and S. Solimini. *Variational Methods in Image Segmentation*. Progress in Nonlinear Differential Equations and Their Applications. Birkhäuser, Basel, 1995.
- [72] Benoit Mory, Roberto Ardon, and Jean-Philippe Thiran. Variational Segmentation using Fuzzy Region Competition and Local Non-Parametric Probability Density Functions. In *IEEE International Conference on Computer Vision (ICCV), Rio, Brazil, 2007*.
- [73] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 22–26, San Francisco, CA, June 1985. IEEE.
- [74] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–684, 1989.
- [75] M. Nielsen, L. Florack, and R. Deriche. Regularization and scale space. Research Report 2352, INRIA Sophia-Antipolis, 1994.
- [76] Mads Nielsen, Luc Florack, and Rachid Deriche. Regularization, scale-space and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7(4):291–308, 1997.
- [77] S. Osher and N. Paragios, editors. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer-Verlag, August 2002.
- [78] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
- [79] N. Paragios and R. Deriche. Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation, Special Issue on Partial Differential Equations in Image Processing, Computer Vision and Computer Graphics*, 13(1/2):249–268, march/june 2002.
- [80] W.A. Perkins. Area segmentation of images using edge points. *pami*, 2(1):8–15, 1980.
- [81] Jérôme Piovano, Mikaél Rousson, and Théodore Papadopoulo. Efficient segmentation of piecewise smooth images. In *Proceedings of the Scale Space and Variational Methods in Computer Vision*, volume 4485 of LNCS, 2007.
- [82] Jean-Philippe Pons. *Contributions méthodologiques et appliquées à la méthode des modèles déformables*. PhD thesis, Ecole Nationale des Ponts et Chaussées, 2005.

- [83] J.P. Pons and J.D. Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In *cvpr07*, pages 1–8, 2007.
- [84] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [85] H. Rifai, I. Bloch, S.A. Hutchinson, J. Wiart, and L. Garnero. Segmentation of the skull using deformable model and taking partial volume effect into account. *mia*, 4(3):219–233, September 2000.
- [86] Kelvin Rocha, Ganesh Sundaramoorthi, and Anthony Yezzi. 3d topology preserving flows for viewpoint-based cortical unfolding. In *Mathematical Methods in Biomedical Image Analysis (MMBIA)*, 2007.
- [87] M. Rochery, I. H. Jermyn, and J. Zerubia. Higher order active contours. *International Journal of Computer Vision*, 69(1):27–42, 2006.
- [88] M. Rousson and R. Deriche. A variational framework for active and adaptative segmentation of vector valued images. In *Proc. IEEE Workshop on Motion and Video Computing*, pages 56–62, Orlando, Florida, December 2002.
- [89] Mikaël Rousson, Thomas Brox, and Rachid Deriche. Active unsupervised texture segmentation on a diffusion based space. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 699–704, Madison, Wisconsin, USA, June 2003.
- [90] Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision*, volume 2, pages 78–92, Copenhagen, Denmark, May 2002. Springer–Verlag.
- [91] F. Ségonne, A.M. Dale, E. Busa, M. Glessner, D. Salat, H.K. Hahn, and B. Fischl. A hybrid approach to the skull stripping problem in MRI. *NeuroImage*, 22(3):1060–1075, 2004.
- [92] F. Ségonne, J.-P. Pons, E. Grimson, and B. Fischl. A novel level set framework for the segmentation of medical images under topology control. In *Workshop on Computer Vision for Biomedical Image Applications: Current Techniques and Future Trends*, 2005.
- [93] J.A. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. Technical Report PAM-651, Center for Pure and Applied Mathematics, University of California, Berkeley, August 1995. To appear *Acta Numerica*.

- [94] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.
- [95] D.W. Shattuck and R.M. Leahy. Brainsuite: An automated cortical surface identification tool. *Medical Image Analysis*, 6(2):129–142, June 2002.
- [96] Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
- [97] Yonggang Shi, P.M. Thompson, I. Dinov, and A.W. Toga. Hamilton–jacobi skeleton on cortical surfaces. *Medical Imaging, IEEE Transactions on*, 27(5):664–673, 2008.
- [98] Jan Sijbers. *Signal and Noise Estimation from Magnetic Resonance Images*. PhD thesis, Universiteit Antwerpen, 1998.
- [99] Jean-Christophe Souplet, Christine Lebrun, Pierre Clavelou, William Camu, Stéphane Chanalet, Nicholas Ayache, and Grégoire Malandain. A comparative study of skull stripping methods in relapsing-remitting multiple sclerosis: Emergence of a new automatic segmentation algorithm. In *Congress of the European Committee for Treatment and Research in Multiple Sclerosis (ECTRIMS)*, 2007.
- [100] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Automated 3d registration of truncated mr and ct images of the head. In *BMVC '95: Proceedings of the 1995 British conference on Machine vision (Vol. 1)*, pages 27–36. BMVA Press, 1995.
- [101] G. Sundaramoorthi, A.J. Yezzi, Jr., and A.C. Mennucci. Sobolev active contours. *The International Journal of Computer Vision*, 73(3):345–366, July 2007.
- [102] Ganesh Sundaramoorthi and Anthony J. Yezzi. More-than-topology-preserving flows for active contours and polygons. In *ICCV*, pages 1276–1283, 2005.
- [103] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.
- [104] J. Talairach and P. Tournoux. *Co-planar Stereotaxic Atlas of the Human Brain*. Thieme, 1988.
- [105] S. Tan, J. Dale, and A. Johnston. Performance of three recursive algorithms for fast space-variant gaussian filtering. *Real-Time Imaging*, 2003.

- [106] Maxime Taron, Nikos Paragios, and Marie pierre Jolly. Border detection on short axis echocardiographic views using a region based ellipse-driven framework. In *Medical Image Computing and Computer-Assisted Intervention – MIC-CAI 2004*, LNCS. Springer, 2004.
- [107] A. Tsai, A.J. Yezzi, and A.S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.
- [108] L.A. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *The International Journal of Computer Vision*, 50(3):271–293, 2002.
- [109] M. Weber, A. Blake, and R. Cipolla. Sparse finite elements for geodesic contours with level-sets. In *ECCV*, 2004.
- [110] M. Weber, A. Blake, and R. Cipolla. Sparse finite element level-sets for anisotropic boundary detection in 3d images. In *Scale-Space*, 2005.
- [111] W.M. Wells, E.L. Grimson, R. Kikinis, and F.A. Jolesz. Adaptive segmentation of MRI data. *IEEE Transactions on Medical Imaging*, 15(4):429–442, 1996.
- [112] Adrien Wohrer. *Model and large-scale simulator of a biological retina with contrast gain control*. PhD thesis, University of Nice Sophia-Antipolis, 2008.
- [113] C. H. Wolters, A. Anwander, X. Tricoche, D. Weinstein, M. A. Koch, and R. MacLeod. Influence of tissue conductivity anisotropy on EEG/MEG field and return current computation in a realistic head model: A simulation and visualization study using high-resolution finite element modeling. *NeuroImage*, (3):813–826, 2006.
- [114] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.
- [115] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro. $O(n)$ implementation of the fast marching algorithm. *J. Comput. Phys.*, 212(2):393–399, 2006.
- [116] A. L. Yuille and N. M. Grzywacz. A computational theory for the perception of coherent visual motion. *nat*, 333:71–74, may 1988.
- [117] X. Zeng, L.H. Staib, R.T. Schultz, and J.S. Duncan. Segmentation and measurement of the cortex from 3D MR images using coupled surfaces propagation. *IEEE Transactions on Medical Imaging*, 18(10):100–111, 1999.
- [118] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1), 2001.

- [119] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.
- [120] S. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.
- [121] O.C. Zienkiewicz and R.L. Taylor. *The Finite element method, 4th ed.* Mac Graw Hill, 1987.
- [122] K. Zilles. Cortex. In G. Paxinos, editor, *The Human Nervous System*, pages 757–802. Academic Press, 1990.