uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Olivier Rousseau
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Mathematics)
GRADE / DEGREE

Department of Mathematics and Statistics
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Geometrical Modeling of the Heart
TITRE DE LA THÈSE / TITLE OF THESIS

Yves Bourgault
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

David Amundsen

Won Sook Lee

Rémi Vaillancourt

Luminata Vese (UCLA)

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Geometrical Modeling of the Heart

Olivier Rousseau

uOttawa

L'Université canadienne
Canada's university

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
University of Ottawa
in partial fulfillment of the requirements for the
Ph.D. degree in the

Ottawa-Carleton Institute for Graduate Studies and Research in
Mathematics and Statistics

Thèse soumise à la
Faculté des études supérieures et postdoctorales
Université d'Ottawa
en vue de l'obtention du doctorat ès sciences à

L'Institut d'études supérieures et de recherche en
mathématiques et statistique d'Ottawa-Carleton

Your file Votre référence
ISBN: 978-0-494-65569-6
Our file Notre référence
ISBN: 978-0-494-65569-6

Acknowledgment

I first thank my supervisor Yves Bourgault. It was a real pleasure to work with him on this project. He is always enthusiastic and supportive. He always finds time to discuss with his students and provide good advices.

I want to thank Charles Pierre for the great time we had working together in Ottawa, for his positive attitude and for his friendship.

Most of all, I want to thank Valérie Poulin for her unconditional support, for the numerous discussions and advices, and for standing by me every day. Big thanks to my son Yohan and his happiness.

Abstract

The heart is a very important human organ, that has a complex structure. Cardiovascular diseases have been the highest cause of death in North America and in Europe for decades. For this reason, a lot of research is made to understand the heart's physiology.

One way to better understand the heart is via theoretical modeling of physiological mechanisms, the main ones being

- trans-membrane potential wave propagation,
- myocardium's contraction and
- blood flow in the cardiac chambers.

These physiological phenomena can be modeled via systems of partial differential equations (PDEs) that are defined on a domain given by the heart's shape. Numerical methods for solving these equations play a crucial role for validating these models. Numerical simulations also serve to make predictions of the organ's reaction to given stimuli. Thereby medical interventions such as the introduction of a pacemaker can be numerically simulated before attempting the surgical implantation.

Such simulations on a complex geometry (the heart muscle or blood chambers) are usually made using the finite element or the finite volume methods. These methods requires a mesh of the computational domain, that is a triangulation of the domain into triangles in 2D or into tetrahedra in a 3D scenario. Thus far, most computations are made on meshes of idealized geometries and there is a lack of accurate 3D geometrical models of the heart. The community is aware of the importance of building accurate 3D models of the heart for understanding its physiology. There is only one realistic heart model that is publicly available.

The main achievement of this project is to have built a precise and complete geometrical model of the human heart. The model consists of

1. An accurate and properly refined mesh of the heart muscle and chambers. The model includes fine features such as the pulpillary muscles (pillars) in the left and right ventricles.

2. The orientation of cardiac fibers.

3. The model is publicly available to the scientific community[1].

Other contributions of this thesis include a careful analysis, of known PDE-based segmentation methods. This is done in Chapter 5. We mostly studied the active contour without edges algorithm. We compared the impact of the choice of discretization on the numerical solutions of the problem. We concluded that some discretizations, while being more natural, do not behave as well as some others. We also evaluated the impact of the initial condition that is chosen on the speed of convergence of the algorithm. We carefully studied the hierarchical method of Gao and Bui [39], and show test cases where it performs better than the original multiphase algorithm of Vese and Chan [117]. We show that the hierarchical segmentation is a more natural framework for segmenting junctions of three segments.

We also proposed modifications of some PDE-based methods to be able to do the heart segmentation. In Chapter 5, we introduced two new types of initial conditions that make the active contour without edges algorithm converge more quickly. Also the hierarchical segmentation algorithm with an L^1 fidelity term is introduced and is shown to be more efficient in some contexts. In Chapter 7, we present a variant of the subjective surface problem introduced by Sarti, Malladi and Sethian [98, 99]. We propose to solve the problem on an annulus around the heart chambers.

During this project, we have developed C++ classes that handles 2D and 3D images. The result is a small PDE image processing toolkit (SPDEIPTK) that is suitable for research use. It features an almost transparent parallel implementation. The toolkit is also publicly available[2].

[1] http://www.mathstat.uottawa.ca/~orous272/

[2] http://www.mathstat.uottawa.ca/~orous272/spdeiptk/index.html

Contents

1

Introduction

The heart is a muscle that pumps the blood through the circulatory system. It brings oxygen-rich blood to the human organs and de-oxygenated blood to the lungs through the pulmonary artery. According to the American Heart Association, cardiovascular diseases have been the number one cause of death in the United States since the year 1900. One out of three adults (about 80 000 000) is affected by cardiovascular diseases. It was responsible for 865 000 deaths across the United States in 2005 only. This represents 35.3% of the total number of deaths that year [59]. These statistics indicate why so many researches are needed to understand the heart's physiology and to understand how cardiovascular diseases affect it.

Figure 1.1: A human heart (from http://www.en.wikipedia.org/wiki/Heart).

During a heartbeat, the heart is crossed by two trans-membrane potential waves. The first one is a depolarization wave that forces the cell contraction. The second is a repolarization wave which brings back the cells to their original state. The depolarization wave is initiated by the sinoatrial node in the right atrium and then transferred to the ventricles by a relay point called the atrioventricular node and the Purkinje fibers. For reference, we include a picture showing the main anatomical features of the heart (Figure 1.2). The myocardial cells (called the myocites) are organized in fibers gathered in sheets. The trans-membrane potential travels at a greater speed in the direction of the heart fibers. The resulting contraction of the ventricles pushes the blood into the blood vessels. The orientation of the fibers also plays an important role in the contraction process, causing a twist in the heart's shape.

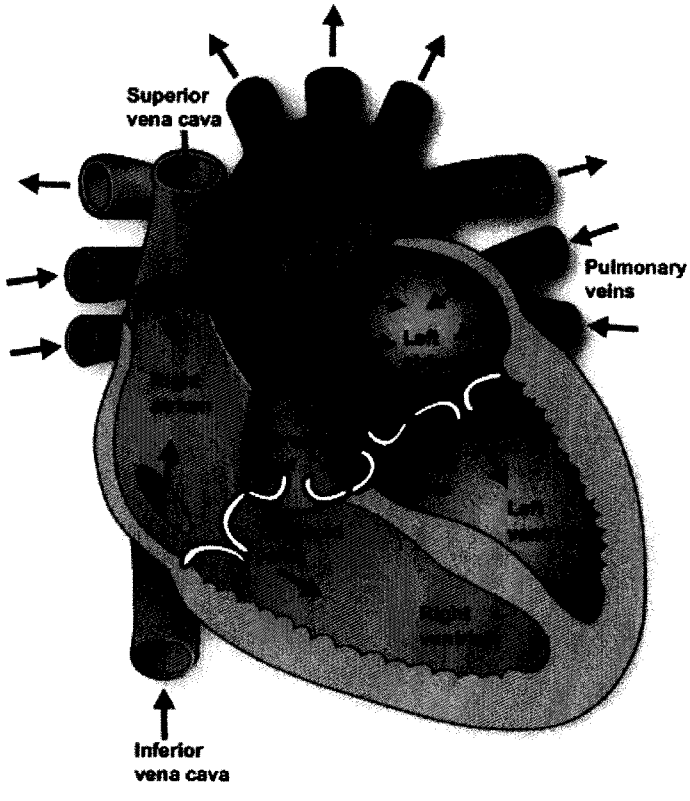


Figure 1.2: Heart anatomy (from http://www.texasheartinstitute.org/hic/anatomy/anatomy2.cfm)

One way to better understand the heart is via the theoretical modeling of these mechanisms, the main ones being

- trans-membrane potential waves propagation,
- myocardium's contraction and
- blood flow in the cardiac chambers.

These physiological phenomena can be modeled via systems of partial differential equations (PDEs) that are defined on a domain given by the heart's shape. Numerical methods for solving these equations play a crucial role for validating these models. Numerical simulations also serve to make predictions of the organ's reaction to given stimuli. Thereby medical interventions such as the introduction of a pacemaker can be numerically simulated before attempting the surgical implantation.

Such simulations on a complex geometry (the heart muscle or blood chambers) are usually made using the finite element or the finite volume methods. These methods requires a mesh of the computational domain, that is a triangulation of the domain into triangles in 2D or into tetrahedra in a 3D scenario. Thus far, most computations are made on meshes of idealized geometries (for example [118]) and there is a lack of accurate 3D geometrical models of the heart. The community is aware of the importance of building accurate 3D models of the heart for understanding its physiology. For instance the *IEEE Transactions on Medical Imaging* dedicated a special issue to this difficult task in 2002 [35].

Only a few realistic models of the heart are available. There are the Auckland canine heart model [57, 105], the INRIA model [75], the Asclepios model [89] and the Auckland porcine model [107]. All these models contain the heart geometry and the orientation of the fibers. In the first models, the fiber orientation was found by dissection, while recent models have fiber orientation derived from diffusion tensor MRI (DTMRI) imaging methods.

Of those, the Asclepios model is the only model publicly available to the scientific community[1]. It consists of the lower part of the left and right ventricles. The surface of the ventricular cavities are smooth and do not contain details such as the pillars. The model is shown in Figure 1.3.

1.1. THE PROJECT

The non availability of accurate 3D models of the human heart motivates this PhD project. The main objective of this project is to build a precise and complete geometrical model of the human heart. The model should consist of

[1] http://www-sop.inria.fr/asclepios/data/heart/index.php

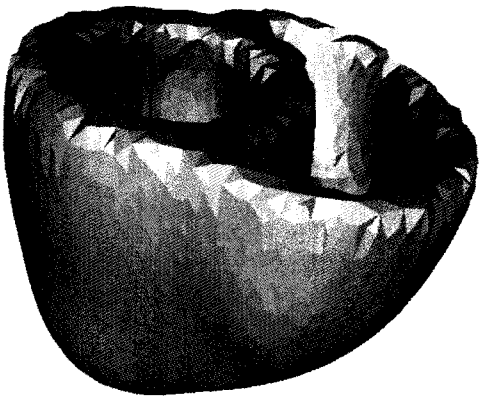


Figure 1.3: The Asclepios model.

1. An accurate and properly refined mesh of the heart muscle and chambers. The model must include fine features such as the pulpillary muscles (pillars) in the left and right ventricles.

2. The orientation of cardiac fibers.

3. The model should be publicly available to the scientific community.

Medical images are about the most reliable sources of data on which to base our construction. The challenge is to extract the heart boundaries and other geometrical characteristics from these medical images. This is called the segmentation process. In a 2D scenario, a human eye can easily detect the contours of objects and draw these contours. It is quite difficult however to get a computer to perform the same tasks, but this computational step is necessary in 3D contexts. Indeed, a human would have a lot of trouble to find a boundary surface in 3D images. Moreover, an automation of the segmentation process would provide a very powerful tool for physicians. Even though automation is not our primary goal, we do favor processes which require little human interaction.

1.2. DATA

Imaging the heart is a difficult task. The main reason for this is that the heart is either constantly moving or, when it is dead, its shape is modified. When the heart beats, the cells contract and make the heart's walls thinner. Two common types of heart imaging

procedures are MRI and CT scans. The Visible Human Project[2] is another source of data but, as the subject is dead, the shape of the heart may not be as realistic as images of living patients. In all cases, the images are given as series of 2D slices. Another cardiac imaging technique is echocardiography, also known as ultrasound. It used to provide 2D views of the heart, but nowadays 3D ultrasound devices are available.

1.2.1. Magnetic Resonance Imaging (MRI)

In magnetic resonance imaging, there is a long sequence of operations that are necessary in order to pass from the acquired data to a real image.

Briefly, the process goes as follows: The patient is placed into a long thick tube. A strong magnetic field affecting hydrogen atoms of the body is produced inside the tube. An hydrogen atom is constituted of a proton and an electron which makes it a dipole. Under the influence of the magnetic field, the dipoles all align in the same direction.

A second device, often shaped like a donut, is placed in the scanner around the region of interest. This device also produces a magnetic field that aligns the hydrogen atoms in a different direction. Everything is then ready for data acquisition: the second magnetic field is stopped and all of a sudden, dipoles come back to their original orientation. When doing so, the atoms release some energy that is captured by the second device. This data is processed in order to reconstruct an image representing the hydrogen density map. Since hydrogen in the human body mostly comes from water, organs can be colored with respect to their water concentration. This is great for heart imaging. It allows to distinguish between the muscle and the cavities of the heart containing blood with a larger water concentration than other tissues. Figure 1.4 shows a slice of a typical MRI data set. The MRI data set to which we have access has resolution $1.2mm \times 1.2mm \times 6mm$ and size $256 \times 256 \times 14$.

During the MRI scan, slices are imaged periodically at the same moment of the beating cycle, usually at the diastole. Hence an image is produced at each heart beat. Typically, the resolution in the x and y directions is between $1mm$ and $2mm$, and the the resolution in the z direction between $2mm$ and $6mm$ This resolution is not high enough to recover fine anatomical details of the heart.

DTMRI scanners (Diffusion tensor MRI) can measure the diffusion of the water in a tissue. In the heart, the privileged direction for the diffusion of the water is along the fibers [101]. This is therefore a good way to compute the orientation of the cardiac fibers. The downside of this technique is that the tissue needs to stay in the scan for a long time

[2] http://www.nlm.nih.gov/research/visible/visible_human.html

Figure 1.4: 3D images of a living patient, $1.2mm \times 1.2mm \times 6mm$. Courtesy of the *Ontario Consortium of Cardiac Imaging*, Sunnybrook Health Sciences Center, University of Toronto

which would be harmful to a living person.

At Johns Hopkins University, in the Center for Cardiovascular Bio-informatics and Modeling, they measured these diffusion tensors from the heart of a dead person and several dog hearts. As the hearts are dead, they may not have a very reliable shape. Nevertheless, these data sets are very interesting regarding the cardiac fibers orientation. A picture of a slice of the human data set together with some fibers is shown in Figure 1.5. The Asclepios team at INRIA Sophia Antipolis created from these data a mean shape of the canine hearts, and a mean fiber orientation from about 20 DTRMI data set [89]. This gives a reasonably clean fiber data set.

1.2.2. Computed Tomography (CT)

Computed tomography uses X-ray technology. As for MRI, the patient is placed into a tube. This tube emits X-rays toward the center of the cylinder. The X-rays pass through the

Figure 1.5: 3D image and fiber information of a dead heart, $0.4mm \times 0.4mm \times 1mm$. The size is $256 \times 256 \times 134$. Courtesy of the Center for Cardiovascular Bio-informatics and Modeling, Johns Hopkins University.

body and the intensity is measured on the other side. Then a reconstruction work is done to actually obtain a 3D image.

A disadvantage of that method is that it distinguishes bones better than organic tissues. The muscle and the cavities of the heart are not well differentiated, both appearing on close gray tones on the CT scan. On the other hand, it offers precise data relatively quickly.

The scan always take the pictures at the same moment of the heart beat. It can capture up to 10 slices of $1.2mm$ per heart beat. On each slice, it is possible to have a resolution of $0.5mm \times 0.5mm$, which is very good. Figure 1.6 shows a slice of a typical CT scan.

Figure 1.6: 3D image of a CT scan, $0.5mm \times 0.5mm \times 1.2mm$. Courtesy of the Heart Institute, University of Ottawa.

This CT scan contains $512 \times 512 \times 199$ voxels and is the best data set that we have at hands at the moment.

1.2.3. Visible Human Project

Another valuable source of data is the one provided by the Visible Human Project [1]. A man and a woman (dead) have been frozen in an ice cube. The frozen bodies were thinly sliced ($1mm$ for the men and $0.33mm$ for the women). A color picture of each slice was then taken. These data sets are very precise and contain a lot of useful anatomic details. The drawback is that the patients were dead, which changes the shape of the heart. The whole data sets are huge. A close up on the man's heart is shown in Figures 1.7. Its size is $440 \times 440 \times 121$.

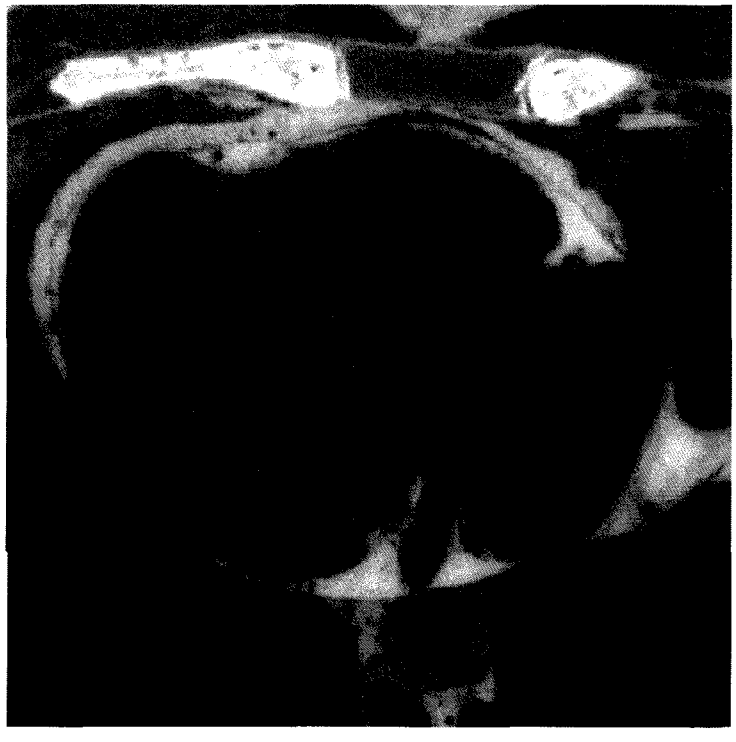


Figure 1.7: Visible Human Male, $0.33mm \times 0.33mm \times 1mm$. Courtesy of the United States National Library of Medicine.

1.2.4. Echocardiography

Echocardiography uses the ultrasound technology to produce 2D images of the heart. It is a fast and safe imaging technique, so that it is now possible to have real-time 3D images of the heart. However, it is often of relatively poor quality as you can see in Figure 1.8. It is mostly used by physicians for quick diagnostics or for real time imaging during a surgical intervention.

Figure 1.8: An example of an echocardiogram (from http://en.wikipedia.org/wiki/Echocardiography).

1.3. HEART SEGMENTATION

Heart segmentation is a difficult task for several reasons. A first reason is the complexity of the organ shape and topology. A second one is the fact that the heart is an organ that is difficult to image. A third reason is that the heart is often occluded by other details of the anatomy such as fat, liver or the chest wall [81]. For all these reasons, the heart is usually segmented using model-based methods. They are segmentation methods that rely on a standard mean shape of the heart, or user defined landmarks in the image.

Model-based techniques yield more robust algorithms, that can perform well on various

images. On the other side, the more robust the algorithm is, the more details will be lost in the segmentation process. An automatic segmentation algorithm will find the overall shape of the heart, but not necessarily the fine details of the heart's anatomy.

To our knowledge, there are very few attempts to segment the myocardium using a method that is not model-based. All of those attempts required careful manual initial condition positioning. The main model-based cardiac segmentation methods are active shape models, active appearance models, and deformable models. A good review is the one of Frangi, Niessen and Viergever [36].

1.3.1. Deformable models in cardiac segmentation

Deformable models have made their first apparition in the snakes introduced by Kass, Witkin and Terzopoulos [51]. Their original idea is to start with a curve (a surface in 3D) and to let the curve evolve until it stops on the contour of the object of interest.

The snake evolution is often driven by a minimization problem: the optimal snake is the one that minimizes a given energy. In general, the energy is made of two terms. The first is an external energy that depends on the underlying image and attracts the snake towards the features of the image. The second is an internal energy that enforces the smoothness of the curve (or surface).

The evolution process can be described in two main ways. The curve can be defined explicitly via a parametrization. The curve is then discretized and a differential equation is solved to find the speed at which the curve should move at each discretization point. The other approach is to define the curve implicitly via a level set method [77]. A partial differential equation needs to be solve on the domain. This approach allows for possible topology changes over the evolution. It is applicable in 2D, 3D or more dimensions without changing the form of the equation. All these methods are described in depth in Sections 4.1-4.5. They have been heavily used for medical image segmentation [36].

Deformable models can segment fine details of the image while enforcing a relative regularity of the segmented contour. This feature also makes the segmentation process less robust to noise or image artifacts. Deformable models are prone to get stuck in local optima due to some local features of the image, or to leak when the object's edges are missing or occluded. Hence, heart segmentation with deformable models requires a careful initialization, with an initial contour that is close to the object of interest. It may also require a user to put landmarks on the image that will guide the segmentation. Because of these characteristics, deformable models are mostly used to segment the left ventricle chamber,

also called the left ventricle's endocardium. The endocardium is the inner surface of the myocardium while the epicardium is the outer surface. The left ventricle endocardium usually has a good contrast with the heart muscle and deformable models can perform well for this segmentation task. It has been done successfully by many people.

Zhukov *et al.* [130] segmented the endocardium surface from MRI images. They take the mesh of a sphere inside the left ventricle chamber as initial condition. The mesh is deformed by minimizing the snake energy (Equation 4.1 below). It is done in an explicit manner in order to preserve the topology of the sphere. Many landmarks are assigned by the user. The mesh stops deforming in the neighborhood of these landmarks.

In the work of Corsi *et al.* [28], the left ventricle's endocardium is segmented from echocardiography images. These are usually very noisy and have many gaps. The segmentation is done via a level set method that has no inflating term (see Section 4.3.1). In this context, a close initial condition is needed. A user takes several slices of the image and on each of these, he marks a polygon located close to the left ventricular chamber boundary. The slice are then gathered together and the initial condition is the polyhedron made of the stacked polygons.

Deformable models are also a good choice to segment 4D images (3D + time), since once a segmented image is found at a given time, it can serve as initial condition for the next time step. It has been done by Gerard *et al.* [41]. They segmented 4D echocardiography images in real time. A first segmentation is given by a coarse mesh that is placed manually. To segment the next images in the time sequence, they first look for a simple affine transformation that brings the initial condition to a better place, then the snake equation is applied explicitly to deform the mesh.

McInerney and Terzopoulos [63] segmented the endocardium surface from 4D CT images. The segmentation is made by deforming the mesh of a sphere that is given inside the chamber. The user can interactively add a string force at some places to attract the mesh to the surface.

There are also some attempts to segment more than just the left ventricle chamber. Santarelli *et al.* [96] segmented the left ventricle with both the endocardium and epicardium surfaces. They used the explicit snake formulation on 2D slices. The first slice is segmented by using a rough curve traced by a user. The snake equation is applied to find a better contour. Once the endocardium contour is found, it is used as initial condition to find the epicardium contour. When the slice is segmented, the algorithm goes to the next slice, taking as initial condition the contours of the last slice.

Park, Metaxas and Axel [79, 80, 81] segmented the endocardium and epicardium sur-

faces of the left and right ventricles in 4D on tagged MRI. This is a special type of MRI image where a grid is overlaid on the MRI image and the grid deforms with the heart movement [128, 11]. In the segmentation process, the volume is first represented by a mesh whose boundaries are ellipsoidal surfaces that roughly fits the first image of the time series. The ellipsoids are deformed very smoothly on the image using the snakes equation. The mesh is then deformed in time using forces derived from the tags.

Von Berg and Lorenz [120] segmented the endocardium and epicardium surfaces of the left ventricle as well as the endocardium surfaces of the right ventricle and atria from CT images. They also included some of the vessels connected to the heart. They used deformable models with user defined landmarks. The initial condition is a model of these surfaces that is built by hand with simple surfaces as spheres, ellipsoids and tubes.

1.3.2. Active shape model and active appearance model

These methods have been introduced by Cootes *et al.* [27, 25]. They use a training set in order to learn the shape and image variability to help the segmentation process. Each image from the training set must be segmented manually by a specialist. Next, N landmark points are placed at strategic locations. The shape of the object is then defined by a single vector x_i in $\mathbb{R}^{3N}$. Principal component analysis (see [34]) is then performed to get the main directions of shape variability.

In the active shape model only information about the shape is used. In the active appearance model, the gray-level variability of the image is also taken into consideration, yielding a more complete segmentation algorithm.

Active shape models and active appearance models are strongly model-based and yield more robust segmentation algorithms. On the other hand, they tend to over-smooth the features. This type of method is suitable for automatic detection of the heart position and overall shape, but not for identification of fine anatomical features.

Active shape models have been used by Van Assen *et al.* [114] to segment left ventricle endocardium and epicardium surfaces from MRI images. A mesh of a mean shape ventricle is first generated and main directions of deformations are obtained from the principal component analysis. Given an image to segment, the model shape is first aligned on the image. It is then deformed using the image intensity and the preferred directions of deformation (see Section 4.6.1).

Frangi *et al.* [37] proposed an automatic method for finding landmarks in images. This can help the process of tagging the training set for the active shape model.

A group from Philips Research has developed over the years an automatic algorithm to segment the endocardium of both ventricles and both atria as well as the epicardium of the left ventricle from various image acquisition techniques [121, 33, 100, 87, 88, 32, 86, 65, 64]. In [33] they addressed the question of how the segmentation via active shape model is affected by the shape variability of the training set.

Zheng *et al.* [129] segmented the same surfaces as the Philips group from CT images. They proposed a fast way to search the shape space for the optimal alignment of the atlas (model shape of the heart). An efficient mix of global and local deformations are also proposed to deform the heart model to fit the image.

Active appearance models have also been used for heart segmentation by Mitchell *et al.* [71] to segment endocardium and epicardium surfaces of the left ventricle from MRI and echocardiography images.

Andreopoulos and Tsotsos [8] proposed a efficient way of matching a heart model on an image using active appearance model. They segmented endocardium and epicardium surfaces of the left ventricle from MRI images.

1.3.3. Other methods

Among other methods are some mix of the methods presented above. For example Fritscher and Schubert [38] combined the idea of active appearance models with deformable models. An atlas is built as for active appearance model and a principal component analysis is made to find the main directions of deformation. Given a new image, the model is first aligned correctly with the image using affine registration [24, 119]. It is then locally deformed with demon registration [110]. It is finally deformed using an extension of the geodesic active contours [15, 58] that uses statistical shape information from the atlas. With this method, they segmented the endocardium of the left and right ventricles from MRI images.

Another example of combined methods is the one proposed by Tobon-Gomez et al. [111]. A 3D active shape model is built on a CT training set to segment lower resolutions images coming from different imaging protocols.

Lorenzo *et al.* [60] segmented the endocardium of the left and right ventricles as well as the epicardium of the left ventricle from 4D MRI images using B-spline registration methods. They first build an atlas from many MRI images that are segmented manually. The atlas is then registered on the image using B-spline registration. It yields a smooth overall segmentation of the surfaces.

Perperidis *et al.* [84] also used a registration method for segmenting the endocardium

and epicardium surfaces of the left ventricle from 4D MRI images.

Santarelli *et al.* [97] proposed a method for segmenting the heart in low resolution images such as CPECT using the segmentation result on a MRI image obtained as in [96].

Some other people use 2D segmentation methods to segment the heart on every slice of the data set and then merge the results to obtain a 3D segmentation [50, 95, 123]. These methods do not make use of all the features of the full 3D images and yields lower quality segmentations.

1.4. PROPOSED FRAMEWORK

As mentioned previously, the main goal of this PhD project is to create a realistic geometrical model of the heart that contains fine anatomical features. We do not intend to devise an algorithm that performs automatic segmentation. Such an algorithm must be robust to noise and shape variability and as such will not segment fine details. However, the proposed method should be easily applicable to many images.

We decided to use the cardiac CT image presented in Section 1.2.2. It is the most precise and clearest cardiac image that we have at our disposal.

We will not consider statistical shape models such as active shape model and active appearance model. These methods are interesting for their robustness for finding the overall shape of the heart. This is perfect for automatic heart segmentation. A drawback of these methods is that they require a training set that consists of many images that are manually segmented. Usually, they require at least 20 images. We do not have access to such a training set. Recall that the efficiency of the active shape methods heavily depends on the diversity of the training set [33].

We use a deformable model approach. However, we will not use a model-based method. There are several reasons justifying this choice. A good model-based method uses an atlas that is built from a given training set. Again, for availability reasons, we do not want to proceed along this avenue. An atlas also introduces a bias in the segmentation result since the algorithm searches for a shape similar to the one of the atlas. This is something that we do not want in our segmentation method.

We create our heart model with the following framework:

1. Segmentation of the full endocardium (both ventricles and atria) via a modified version of the Chan-Vese algorithm.

2. Segmentation of the full epicardium (both ventricles and atria) using a modified subjective surface problem.

3. Mesh generation of the myocardium using DistMesh.

4. Registration of the Asclepios model onto our model to get the fiber orientation in the ventricles using diffeomorphic demons.

Chan-Vese algorithm

The Chan-Vese algorithm is a variant of the Mumford and Shah problem [72]. Let

$$g : \Omega \to [0, 1] \tag{1.1}$$

be an image. Mumford and Shah proposed to consider pairs (u, K), formed of an image u and a compact set K representing edges. They state that the pair (u, K) which minimizes

$$F(u, K) = \int_{\Omega \setminus K} |\nabla u|^2 dx + \int_{\Omega} (g - u)^2 dx + \text{length}(K)$$

gives a good segmentation of the image g. In this context, the set K represents the contours of the objects in g. Moreover, the function u should be such that ∇u is well defined on $\Omega \setminus K$. Such functions belong to the space of functions of bounded variation.

It is a difficult task to prove that this functional has indeed a minimum. To compute this minimum numerically, the problem may be expressed in a simplified form by assuming that the minimal image u has only 2 values. Using this assumption, the problem can be reformulated using level sets. This leads to the PDE, proposed by Chan and Vese [21],

$$\begin{cases} \phi_t = \mu\delta(\phi)\left(\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + (g - u)\right), & \Omega \\ \frac{\partial\phi}{\partial n} = 0, & \partial\Omega \\ \phi(x, 0) = \phi_0, \end{cases} \tag{1.2}$$

where u is a function that takes only two values c_1 and c_2. The steady state solution splits the domain into two regions, namely $\{\phi < 0\}$ and $\{\phi > 0\}$. These two sets represent regions of interest of the image g. It is then possible to consider one of these regions as a new domain, and split it again. This hierarchical process gives an efficient way to segment images. We will study and use an improved version of this algorithm.

Subjective surfaces

The subjective surfaces method has been introduced by Sarti, Malladi and Sethian [98, 99]. It is a variation of the geodesic active contour method of Caselles, Kimmel and Sapiro [16], where all level sets of the level set function are considered. The idea is that all level sets will be attracted by the image edges, creating cliffs. A suitable threshold then yields a segmentation. The subjective surface problem is given as

$$\begin{cases} \phi_t = h(|\nabla g|)|\nabla\phi|\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \nabla h(|\nabla g|)\cdot\nabla\phi & \text{on } \Omega \\ \phi = 0 & \text{on } \partial\Omega, \\ \phi(x,0) = \phi_0 & \text{on } \Omega, \end{cases} \tag{1.3}$$

where h is called an edge stopping function. It is such that

1. h is monotonically decreasing,
2. $h(0) = 1$,
3. $h(x) \overset{x\to\infty}{\longrightarrow} 0$.

We will use a modified version of the subjective surface problem that is more suitable for heart segmentation.

DistMesh

The publicly available Matlab code DistMesh [85, 47] is used to mesh the resulting segmentations. The code has been modified to create meshes that are compatible with the boundary of sub-domains.

Diffeomorphic demons

Diffeomorphic demons will be used to map the Asclepios model onto our segmentation results. A diffeomorphic registration method need to be used in order to be able to map the fibers using its Jacobian. The problem is variational as the diffeomorphism is sought as the minimum of a functional over a space of diffeomorphisms. We used for this the method introduced by Vercauteren *et al.* [116] that was implemented in the Insight Toolkit (ITK) by the same authors [115].

1.5. THESIS OVERVIEW

The structure of this thesis is as follows. In Chapter 2, we present some background material that is relevant for PDE methods in image analysis. We introduce the functional space of functions of bounded variation as well as the main theorems leading to the existence of solutions for most image processing PDE methods. We also present some basics in the calculus of variations and the Mumford-Shah functional.

In Chapter 3, we present the basic PDE methods in image denoising and possible numerical schemes to solve them. We introduce the total variation equation [92].

In Chapter 4, we present an overview of existing segmentation methods for medical images. We give a special care to PDE based methods as they are at the heart of this thesis. We first present the classical snakes method [51]. Next we introduce the level set method [77] that is used to define other snake methods such as geodesic active contours [16] and subjective surfaces [98, 99]. We expand more on the active contours without edges, also known as the Chan-Vese algorithm [21]. We also present how one can introduce a shape prior in each of these methods. We end this chapter by two other segmentation techniques called active shape model and active appearance model [27, 25].

Chapter 5 is dedicated to a fine analysis of the active contour without edges algorithm. We study several aspects of the problem and describe how these influence the results. We first study the different discretizations of the energy functional. This implies the computation of the length of a curve defined implicitly. We then look at the impact of the related discretizations of the Euler-Lagrange equation. Next we study the impact of the initial condition on the problem. We introduce two new types of initial conditions, namely a random initial curve and another curve that is the solution of the active contour without edges with no curvature term. We also analyze the hierarchical method based on active contours without edges introduced by Tsai, Yezzi and Willsky [113] and further investigated by Gao and Bui [39]. We present test cases where this method performs better than the original multiphase method. Finally, we propose to use this method with a L^1 fidelity term instead of the classic L^2 fidelity. This yields an algorithm that is more efficient and less sensitive to noise.

In Chapter 6, we use the analysis of Chapter 5 to get a modified version of the original active contour without edges. We shall use this method to segment 3D cardiac CT images. We introduce a parallel implementation of this algorithm that enables us to solve the related PDE problem in a reasonable time. The heart chambers, ventricles and atria are well segmented and the segmented surfaces contain many fine anatomical details. The epicardium

of the heart is also well segmented, but it suffers from leaking in some regions.

We tackle the problem of the epicardium segmentation in Chapter 7 using subjective surfaces. The original subjective surface problem takes a long time to solve on large images and may also suffer from leaking for the type of images that we are segmenting. For these reasons, we introduce a new subjective surface problem, that is on a smaller domain with boundary conditions that will prevent the curve from leaking. With this method, we obtained nice segmentations of the epicardium of the ventricles and of the atria. The results of Chapters 6 and 7 put together yield a full 3D heart segmentation that is anatomically accurate.

In Chapter 8, we address the problem of mesh generation from the segmentation results obtained. For this task, a library of Matlab code called DistMesh [85, 47] is used. We modified parts of the algorithms in order to fit our applications. We generate 2D and 3D meshes of the heart, heart and torso, the trachea and the carotid.

In Chapter 9, we use the Asclepios model in order to get the fiber orientation over our geometrical model. This is done using the diffeomorphic demons algorithm and an inpainting technique.

In chapter 10 we conclude by stating the scientific contributions of this thesis. We also present some insights on future work and applications.

The code for image segmentation that has been developed during this project is briefly presented in an appendix. It basically consists of four classes that are easy to use to perform serial and parallel image processing operations. The code is gathered in the small PDE image processing toolkit (`SPDEIPTK`).

2

Background material

In this chapter, we shall present some of the background that is necessary to correctly state and solve variational problems that arise in image de-noising and segmentation. This will contain some measure theory, basic notions about distributions and weak derivatives. For proofs, or a more detailed exposition, we recommend for example Rudin [93, 94].

Next, we talk about some fine properties of functions, such as approximate continuity and jump set and introduce spaces of functions of bounded variations, state the main compactness results and go into some proof in order to get a feeling of them. We recommend for this material the excellent and very complete book by Ambrosio [5], although there are several other good references such as [43, 131, 13, 9]

At last, we present some basics of calculus of variations. A more complete treatment can be found for example in Giusti [44] or Struwe [108].

2.1. L^P SPACES

In this section, Ω denotes an open set of $\mathbb{R}^N$. By $L^p(\Omega)$, we mean $L^p(\Omega, \mathscr{L}^N \llcorner \Omega)$, where $\mathscr{L}^N$ is the Lebesgue measure on $\mathbb{R}^N$ and $\mathscr{L}^N \llcorner \Omega$ denotes the restriction of the measure to the set Ω. $u_n \rightharpoonup u$ denotes weak convergence of u_n towards u. Standard theory of L^p spaces is assumed. The statements below are less known and are of interest for variational problems in image processing.

Theorem 1. *Let $1 < p < \infty$. If $\{u_n\}_n \subseteq L^p(\Omega)$ is a bounded sequence, then there exists a sub-sequence*

$$u_{n_k} \rightharpoonup u \in L^p(\Omega).$$

If $p = 1$, the above theorem is not valid, we need an additional condition on the sequence to ensure the existence of a converging sub-sequence, namely equi-integrability.

Definition 2. *$\{u_n\}_n \subseteq L^1(\Omega)$ is **equi-integrable** if and only if*

1. $\forall \epsilon > 0, \exists A \subseteq \Omega$ *such that* $\mu(A) < \infty$ *and* $\forall n, \int_{\Omega \setminus A} u_n dx < \epsilon$,

2. $\forall \epsilon > 0, \exists \delta > 0$ *such that* $\forall E$ *μ–measurable set,* $\mathscr{L}^N(E) < \delta \Rightarrow \forall n, \int_E |u_n| dx < \epsilon$.

Proposition 3. *If Ω is bounded, then $\{u_n\}_n \subseteq L^1(\Omega)$ is equiintegrable if and only if*

$$\{u_n\}_n \subseteq \left\{ f \in L^1(\Omega) : \int_\Omega \phi(|f|) \leq 1 \right\}$$

for some increasing continuous function $\phi : [0, \infty] \longrightarrow [0, \infty]$ satisfying $\lim_{t \to \infty} \phi(t)/t = \infty$.

Theorem 4. *A bounded sequence $\{u_n\}_n \in L^1(\Omega)$ has a weakly convergent sub-sequence if and only if $\{u_n\}_n$ is equiintegrable.*

2.2. RADON MEASURES

In this section X denotes a locally compact separable metric space and $\mathscr{B}(X)$ is the σ-algebra of all Borel sets on X.

Definition 5. *$\mu : \mathscr{B}(X) \longrightarrow \mathbb{R}^N$ is a **Radon measure** if*

1. $\mu(\emptyset) = 0$,

2. $\forall (E_n)_n \subseteq \mathscr{B}(X)$, *pairwise disjoint sets,*

$$\mu\left(\bigcup_{n=0}^{\infty} E_n\right) = \sum_{n=0}^{\infty} \mu(E_n).$$

The set of all Radon measures on X is denoted $\mathscr{M}(X)$.

Remark 6. *For $\mu \in \mathscr{M}(X)$, there is a decomposition $\mu = (\mu_1, ..., \mu_N)$. For $u \in [C_c(X)]^N$ (that is $u = (u_1, u_2, ..., U_N)$ with each $u_i \in C_C(X)$), write*

$$\int_X u d\mu = \sum_{i=1}^{N} \int_X u_i d\mu_i.$$

Definition 7. *Let $\mu \in \mathscr{M}(X)$. The **total variation** of μ*

$$|\mu|(E) = \sup \left\{ \sum_{n=0}^{\infty} |\mu(E_n)| : E_n \in \mathscr{B}(X) \text{ pairwise disjoint }, E = \bigcup_{n=0}^{\infty} E_n \right\}$$

is a finite positive measure on X.

Proposition 8. *Let $\mu \in \mathscr{M}(X)$, then for all open set $A \subseteq X$,*

$$|\mu|(A) = \sup\left\{\int_X u d\mu : u \in [C_c(A)]^N, \|u\|_\infty \leq 1\right\}$$

Remark 9. *The set $\mathscr{M}(X)$ is a Banach space when equipped with the norm*

$$\|\mu\| = |\mu|(X).$$

The space $\mathscr{M}(X)$ can be identified with the dual of the space of continuous functions vanishing at boundaries of X. This space is $C_0(X)$, the closure of $C_c(X)$ in the space $C^0(X)$ of continuous functions on X. It corresponds to the continuous functions which vanish at the boundaries of X.

Theorem 10 (Riesz). *Let*

$$L : [C_0(A)]^N \longrightarrow \mathbb{R}$$

be a linear bounded functional. Then there exists a unique $\mu \in \mathscr{M}(X)$ such that

$$\forall \phi \in [C_0(A)]^N, \quad L\phi = \int_X \phi d\mu.$$

Therefore $\mathscr{M}(X) \cong \left([C_0(A)]^N\right)^*$. Moreover

$$\mu(X) = \|L\| = \sup_{\|\phi\|_\infty \leq 1} L(\phi)$$

In view of this fact, it is possible to define a weak* topology on $\mathscr{M}(X)$. This topology turns out to be very useful for applications.

Definition 11. *Let $(\mu_n)_n \subseteq \mathscr{M}(X)$. Then $\mu_n \overset{*}{\rightharpoonup} \mu$ if*

$$\forall \phi \in [C_c(A)]^N, \quad \int_X \phi d\mu_n \longrightarrow \int_X \phi d\mu.$$

Theorem 12. *Let $(\mu_n)_n \in \mathscr{M}(X)$ be such that $\sup_n |\mu_n|(X) < \infty$. Then there exists a subsequence $(u_{n_k})_k \overset{*}{\rightharpoonup} u \in \mathscr{M}(X)$. Moreover the map $\mu \mapsto |\mu|(X)$ is lower semi-continuous with respect to the weak* convergence. That is*

$$\liminf_n |\mu_n|(X) \geq |\mu|(X).$$

Theorem 13. *Let $(\mu_n)_n \subseteq \mathscr{M}(\Omega)$ such that $\mu_n \overset{*}{\rightharpoonup} u$. If $|\mu_n| \overset{*}{\rightharpoonup} \lambda$ for some positive measure λ, then*

$$|\mu| \leq \lambda.$$

Definition 14. *Let μ be a positive measure on $\mathscr{B}(X)$, and $\nu \in \mathscr{M}(X)$. Then*

1. *ν is **absolutely continuous** with respect to μ ($\nu \ll \mu$) if*

$$\forall E \in \mathscr{B}(X), \quad \mu(E) = 0 \quad \Longrightarrow \quad \nu(E) = 0.$$

2. *ν and μ are **mutually singular** ($\nu \perp \mu$) if there exists $E \in \mathscr{B}(X)$ such that*

$$\nu(E) = 0 = \mu(E^c).$$

Theorem 15 (Radon-Nikodým). *Let μ be a σ-finite measure on $\mathscr{B}(X)$ and $\nu \in \mathscr{M}(X)$. Then there exists $\nu^a, \nu^s \in \mathscr{M}(X)$ such that $\nu^a \ll \mu$, $\nu^s \perp \mu$ and*

$$\nu = \nu^a + \nu^s.$$

Moreover, this decomposition is unique and there is a unique $f \in [L^1(X,\mu)]^N$ such that

$$\nu^a = f\mu.$$

Remark 16. *If μ and ν are mutually singular, then for any $E \in \mathscr{B}(X)$*

$$|\mu + \nu|(E) = |\mu|(E) + |\nu|(E).$$

2.3. APPROXIMATE CONTINUITY

For functions in $L^p(\Omega)$, it is still possible to define some kind of continuity and differentiability.

Definition 17. *Let $u \in L^1_{\text{loc}}(\Omega)$. u has an **approximate limit** at $x \in \Omega$ if there exists $z \in \mathbb{R}$ such that*

$$\lim_{\delta \to 0} \fint_{B_\delta(x)} |u(y) - z| dy = 0,$$

*where $B_\delta(x)$ denotes the ball of radius δ about x and $\fint_B f(x)dx$ denotes the mean value of f over B. If there exists no $z \in \mathbb{R}$ with this property, we say that x is an **approximate***

discontinuity *point, and the* ***approximate discontinuity set*** S_u *is the set of such points.* u *has an* ***approximate jump*** *at* $x \in \Omega$ *if there exits* $z^+ \neq z^- \in \mathbb{R}$ *such that*

$$\lim_{\delta \to 0} \fint_{B_\delta^+(x)} |u(y) - z^+| dy = 0 = \lim_{\delta \to 0} \fint_{B_\delta^-(x)} |u(y) - z^-| dy,$$

where $B_\delta^\pm$ *denote half balls of radius* δ *about* x. *The approximate jump set* J_u *is the set all points having an approximate jump.* ***jump set.*** *Then* $J_u \subseteq S_u$.

The following theorem gathers facts about S_u and J_u that are important.

Theorem 18. *Let* $u \in L^1_{\text{loc}}(\Omega)$. *Then*

1. $\mathscr{L}^N(S_u) = 0$,
2. $\mathscr{H}^{N-1}(S_u \setminus J_u) = 0$,

where $\mathscr{H}^{N-1}$ *stands for the* $(N-1)$*-dimensional Hausdorff measure.*

2.4. DIFFERENTIABILITY

Definition 19. *Let* $\mathscr{D}(\Omega)$ *be the set of* C^∞ *functions with compact support in* Ω. *Then provide* $\mathscr{D}(\Omega)$ *with the following topology: we say that* $\phi_n \overset{\mathscr{D}(\Omega)}{\longrightarrow} \phi$ *if there exists a compact* $K \in \Omega$ *such that for all* n, $\operatorname{supp}(\phi_n) \subseteq K$ *and for all* $\alpha \in \mathbb{N}^N$, $D^\alpha \phi_n$ *converge uniformly to* $D^\alpha \phi$ *on* K.

Remark 20. *The space* $\mathscr{D}(\Omega)$ *cannot be equipped with a norm. It has in fact a family of semi-norms, namely the* $C^k(K)$ *norms, where* $K \subseteq \Omega$ *is compact.*

Definition 21. *The space of distributions* $\mathscr{D}'(\Omega)$ *is the dual of* $\mathscr{D}(\Omega)$, *which means that*

$$\mathscr{D}'(\Omega) = \{L : \mathscr{D}(\Omega) \longrightarrow \mathbb{R} : L \text{ is a continuous linear functional}\}.$$

It is common to write $< L, \phi >_{\mathscr{D}(\Omega)}$ *(or* $(L, \phi)_{\mathscr{D}(\Omega)}$ *or simply* $< L, \phi >$ *or* (L, ϕ)*) for* $L\phi$.

Example 22. *Given* $f \in L^1$, *define the distribution*

$$L_f : \mathscr{D}(\Omega) \longrightarrow \mathbb{R},$$
$$\phi \mapsto (f, \phi)_{\mathscr{D}(\Omega)} = \int_\Omega f\phi \, dx.$$

Suppose f is C^1. Then by Green's theorem and the fact that $\phi|_{\partial\Omega} = 0$

$$(D_i f, \phi)_{\mathscr{D}(\Omega)} = \int_\Omega D_i f \phi \, dx = -\int_\Omega f D_i \phi \, dx = -(f, D_i\phi)_{\mathscr{D}(\Omega)},$$

which suggests:

Definition 23. *The partial **derivative of a distribution** L in the direction x_i is*

$$\begin{aligned} D_i L : \mathscr{D}(\Omega) &\longrightarrow \mathbb{R}, \\ \phi &\mapsto -(L, D_i\phi)_{\mathscr{D}(\Omega)}. \end{aligned}$$

*The **total derivative** of L is $DL = (D_1 L, ..., D_N L)$. For $\phi \in [\mathscr{D}(\Omega)]^N$, we write*

$$(DL, \phi)_{[\mathscr{D}(\Omega)]^N} = (DL_1, \phi_1)_{\mathscr{D}(\Omega)} + ... + (DL_N, \phi_N)_{\mathscr{D}(\Omega)}.$$

Hence remark that

$$(DL, \phi)_{[\mathscr{D}(\Omega)]^N} = -(L, \mathrm{div}\phi)_{\mathscr{D}(\Omega)}.$$

Definition 24 (Sobolev functions). *Let $f \in L^p(\Omega)$. We say that f belongs to the Sobolev space $W^{1,p}(\Omega)$ if its total derivative Df in the sense of distributions belongs to $[L^p(\Omega)]^N$. This means that there exists $g \in [L^p(\Omega)]^N$ such that*

$$\forall \phi \in [\mathscr{D}(\Omega)]^N \quad -\int_X f \mathrm{div}\phi \, dx = \int_X g \cdot \phi \, dx.$$

$W^{1,p}(\Omega)$ is a Banach space with the norm

$$\|f\|^p_{W^{1,p}(\Omega)} = \|f\|^p_{L^p} + \sum_i \|D_i f\|^p_{L^p(\Omega)}.$$

$W^{1,2}(\Omega)$ is even a Hilbert space with the dot product

$$(f, g)_{W^{1,2}(\Omega)} = (f, g)_{L^2(\Omega)} + \sum_i (D_i f, D_i g)_{L^2(\Omega)},$$

and is often denoted by $H^1(\Omega)$.

Definition 25 (Functions of bounded variation). *The function $f \in L^1$ is of **bounded variation** if there exists a finite signed measure μ such that*

$$\forall \phi \in [\mathscr{D}(\Omega)]^N, \quad -\int_\Omega f \mathrm{div}\phi \, dx = \int_\Omega \phi d\mu.$$

Write $BV(\Omega)$ for the space of functions of bounded variation and $D_f = \mu = (\mu_1, ..., \mu_N)$ is the total derivative of f in the sense of distributions. The measure μ_i is the partial derivative of u in the direction x_i. The space $BV(\Omega)$ is a Banach space with the norm

$$\|f\|_{BV(\Omega)} = \|f\|_{L^1} + |Df|(X),$$

where $|Df|$ is the total variation of the Radon measure Df.

Remark 26. *Let $u \in BV(\Omega)$ and $\mu = Du$. Then by Riez Theorem,*

$$|\mu|(E) = \sup_{\substack{\phi \in \mathscr{D}(\Omega) \\ \|\phi\|_\infty \leq 1}} \int_E u \operatorname{div} \phi \, dx = \sup_{\substack{\phi \in \mathscr{D}(\Omega) \\ \|\phi\|_\infty \leq 1}} \int_E \phi \, d\mu.$$

Example 27. *Let $u : (-1, 1) \longrightarrow \mathbb{R}$ be the function*

$$u(x) = \begin{cases} 0, & x \leq 0, \\ x^2/2, & x > 0. \end{cases}$$

Therefore

$$u'(x) = \begin{cases} 0, & x \leq 0, \\ x, & x > 0. \end{cases} \qquad u''(x) = \begin{cases} 0, & x \leq 0, \\ 1, & x > 0. \end{cases} \qquad u'''(x) = \delta.$$

Then we have

$$\begin{aligned} u' \in C^0(-1,1) &\implies u \in C^1(-1,1), \\ u'' \in L^1(-1,1) &\implies u' \in W^{1,1}(-1,1), \\ u''' = \delta \in \mathscr{M}(-1,1) &\implies u'' \in BV(-1,1). \end{aligned}$$

Example 28. *Let $E \subseteq \Omega$. Define the **perimeter of** E **in** Ω to be*

$$P(E, \Omega) = |D\mathbb{1}_E|(\Omega),$$

*where $\mathbb{1}_E$ is the indicator function of the set E. It is said that E is of **finite perimeter** in Ω if*

$P(E,\Omega) < \infty$. Now suppose ∂E is C^1. Then by Gauss-Green Theorem and Proposition 8,

$$|D\mathbb{1}_E|(\Omega) = \sup_{\substack{\phi\in\mathscr{D}(\Omega)\\ \|\phi\|_\infty\le 1}} \int_\Omega \phi\, dD\mathbb{1}_E = \sup_{\substack{\phi\in\mathscr{D}(\Omega)\\ \|\phi\|_\infty\le 1}} \int_\Omega \mathrm{div}\phi\, \mathbb{1}_E\, dx$$

$$= \sup_{\substack{\phi\in\mathscr{D}(\Omega)\\ \|\phi\|_\infty\le 1}} \int_E \mathrm{div}\phi\, dx = \sup_{\substack{\phi\in\mathscr{D}(\Omega)\\ \|\phi\|_\infty\le 1}} \int_{\partial E\cap\Omega} \phi\cdot \nu d\mathscr{H}^{N-1} = \mathscr{H}^{N-1}(\partial E\cap\Omega).$$

The last equality seems obvious, but requires some work. The result is still true in a more general context: If E has finite perimeter, then

$$P(E,\Omega) = \mathscr{H}^{N-1}(\partial E\cap\Omega).$$

2.5. THE SPACES $BV(\Omega)$ AND $SBV(\Omega)$

In this section Ω denotes a bounded open set of $\mathbb{R}^N$ with Lipschitz boundary. It turns out that the convergence in the $BV(\Omega)$-norm is too strong for most of the applications. The weak* convergence is more useful.

Definition 29. *Let $\{u_n\}_n \subseteq BV(\Omega)$. Then $u_n \overset{*}{\rightharpoonup} u \in BV(\Omega)$ if*

1. *$u_n \overset{L^1(\Omega)}{\longrightarrow} u$, and*
2. *$Du_n \overset{*}{\rightharpoonup} Du$ as Radon measures.*

It has the following nice properties:

Theorem 30. *Let $\{u_n\}_n \subseteq BV(\Omega)$ and $u \in BV(\Omega)$. Then*

$$u_n \overset{*}{\rightharpoonup} u \iff u_n \overset{L^1(\Omega)}{\longrightarrow} u \quad \text{and} \quad \sup_n \|u_n\|_{BV(\Omega)} < \infty.$$

Theorem 31 (Compactness in $BV(\Omega)$)**.** *Let $\{u_n\}_n \subseteq BV(\Omega)$ be such that*

$$\sup_n \|u_n\|_{BV(\Omega)} < \infty,$$

then there exists $u \in BV(\Omega)$ and a sub-sequence u_{n_k} such that $u_{n_k} \overset{*}{\rightharpoonup} u$.

Using Radon-Nikodým Theorem, there exists a unique $g \in [L^1(\Omega)]^N$ such that

$$Du = D^a u + D^s u = g\mathscr{L}^N + D^s u.$$

So for $\phi \in [\mathscr{D}(\Omega)]^N$,

$$\int_\Omega f \mathrm{div}\phi \, dx = \int_X g\phi \, dx + \int_\Omega \phi dD^s u.$$

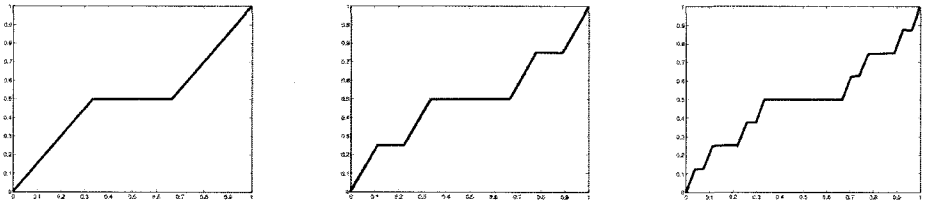
This suggests the notation $\nabla u = g$, which represents the Sobolev part of derivative Du of the function u.

Theorem 32 (Decomposition of derivatives). *Let $u \in BV(\Omega)$, then*

$$Du = D^a u + D^j u + D^c u = g\mathscr{L}^N + (u^+ - u^-)\nu_u \mathscr{H}^{N-1} \llcorner J_u + D^c,$$

*where J_u is the set of approximate discontinuities of u, ν_u is a normal vector to the discontinuity set, $\mathscr{H}^{N-1}$ is the Hausdorff measure and u^+, u^- stand for the limits of u on both sides of J_u. $D^j u$ is the **jump part** of the function u. $D^c u$ is called the **Cantor part** and has Hausdorff dimension $d \in (N-1, N)$.*

Remark 33. *The notation $D^c u$ is motivated by the fact that the Cantor-Vitali function $u \in BV(0,1)$ is such that $Du = D^c u$ and* supp(Du) *is included in the Cantor set. We will consider functions that do not have derivatives of this type.*



C_1 C_2 C_3

Figure 2.1: The Cantor-Vitali function C is the limit of the above sequence of functions. It is continuous, monotonically increasing, $|Du|(0,1) = 1$, but $C' = 0$ almost everywhere. In fact, the support of $D^C u$ is exactly the Cantor set.

Remark 34. *Let $u \in BV(\Omega)$, then $|Du|(\Omega) < \infty$. From Remark 16, it follows that*

$$|\nabla u \mathscr{L}^N|(\Omega) + |D^s u|(\Omega) = |Du|(\Omega) < \infty.$$

Hence both terms on the left hand side are finite.

Definition 35 (Special functions of bounded variation). *Let $u \in BV(\Omega)$, then $u \in SBV(\Omega)$ if $D^c u = 0$. Or equivalently $u \in SBV(\Omega)$ if and only if $D^s u$ has support in a measurable σ-finite set with respect to $\mathscr{H}^{N-1}$.*

Example 36. *Let $u_1, u_2 \in W^{1,1}(\Omega) \cap L^\infty(\Omega)$ and $E \in \Omega$ a subset of finite perimeter in Ω. Then*

$$u = u_1 \mathbb{1}_E + u_2 \mathbb{1}_{\Omega \setminus E} \in SBV(\Omega).$$

Example 37. *An image is some function $g : \Omega \longrightarrow \mathbb{R}$, that is smooth inside the boundaries of the objects and discontinuous on their boundary which have $\mathscr{H}^{N-1}$-finite measure. If $\|g\|_\infty < \infty$, then $g \in SBV(\Omega)$.*

Proposition 38. *The space $SBV(\Omega)$ is closed with respect to the $BV(\Omega)$ norm.*

Proof. Let $u_n \overset{BV(\Omega)}{\longrightarrow} u$, where $u_n \in SBV(\Omega)$. The Sobolev part and the singular part of a function of bounded variation are orthogonal by definition, then

$$|D^a(u_n - u) + D^s(u_n - u)| = |D^a(u_n - u)| + |D^s(u_n - u)| \to 0. \tag{2.1}$$

Now, since $D^s u_n$ is concentrated on B_n which is σ-finite with respect to $\mathscr{H}^{N-1}$, then by the fact that $|D^s u_n - D^s u| \to 0$, we have that $D^s u$ is concentrated on $\cup_n B_n$, which is σ-finite with respect to $\mathscr{H}^{N-1}$. □

However the closure of $SBV(\Omega)$ with respect to the weak* convergence would be more interesting. In fact $SBV(\Omega)$ is not closed with respect to the weak* topology, but it is possible to find sufficient conditions for the weak* limit of a sequence of functions in $SBV(\Omega)$ to be in $SBV(\Omega)$. These will follow from the following result about the chain rule in $BV(\Omega)$.

Theorem 39 (Chain rule in $BV(\Omega)$). *Let $u \in BV(\Omega)$ and $\psi \in C^1(\mathbb{R})$ such that $\psi(0) = 0$, then $\psi \circ u \in BV(\Omega)$ and*

$$D(\psi \circ u) = \psi'(u)\nabla u \mathscr{L}^N + (\psi(u^+) - \psi(u^-))\nu_u \mathscr{H}^{N-1} \llcorner J_u + \psi'(\tilde{u}) D^c u,$$

where $\tilde{u}$ denotes the approximate limit of u. If Ω is bounded, then the hypothesis $\psi(0) = 0$ can be removed.

So if $u \in SBV(\Omega)$, for all $\psi \in C^1(\mathbb{R}) \cap W^{1,\infty}(\mathbb{R})$, we have

$$\begin{aligned} \left|D\psi(u) - \psi'(u)\nabla u \mathscr{L}^N\right| &= \left|(\psi(u^+) - \psi(u^-))\nu_u \mathscr{H}^{N-1} \llcorner J_u\right| \\ &\leq 2\|\psi\|_\infty |D^j u|, \end{aligned}$$

where $\nu_u \leq \mu$ as measures means that for every measurable set E, we have $\nu_u(E) \leq \mu(E)$. It turns out that the reverse holds in the following sense:

Theorem 40. *Let $u \in BV(\Omega)$. If there exists $a \in [L^1(\Omega)]^N$ and μ a finite positive measure on Ω such that $\forall \psi \in C^1(\mathbb{R}) \cap W^{1,\infty}(\mathbb{R})$*

$$|D\psi(u) - \psi'(u)a\mathscr{L}^N| \leq \|\psi\|_\infty \mu,$$

then $u \in SBV(\Omega)$. Moreover $a = \nabla u$ and $\mu \geq \mathscr{H}^{N-1} \llcorner J_u$.

It is then possible to prove the following closure property of $SBV(\Omega)$.

Theorem 41 (Closure of $SBV(\Omega)$)**.** *Let $\Omega \subseteq \mathbb{R}^N$ be a bounded domain and $\{u_n\}_n \subseteq SBV(\Omega)$ such that $u_n \overset{*}{\rightharpoonup} u \in BV(\Omega)$ and*

$$\sup_n \left\{ \int_\Omega |\nabla u_n|^2 \, dx + \mathscr{H}^{N-1}(J_{u_n}) \right\} < \infty.$$

Then $u \in SBV(\Omega)$ and

1. *$\nabla u_n \rightharpoonup \nabla u$ in $[L^1(\Omega)]^N$,*
2. *$D^j u_n \overset{*}{\rightharpoonup} D^j u$.*

Moreover

$$\int_\Omega |\nabla u|^2 \, dx \leq \liminf_n \int_\Omega |\nabla u_n|^2 \, dx$$

and

$$\mathscr{H}^{N-1}(J_u) \leq \liminf_n \mathscr{H}^{N-1}(J_{u_n}).$$

Proof. Note that

$$Du_n = \nabla u_n \mathscr{L}^N + D^j u_n \overset{*}{\rightharpoonup} Du.$$

One needs to show that $\nabla u_n \rightharpoonup \nabla u$ in $L^1(\Omega)$, $D^j u_n \overset{*}{\rightharpoonup} D^j u$ and $D^c u = 0$.

First, take a look at the jump part. By hypothesis, if $\mu_n = \mathscr{H}^{N-1} \llcorner J_{u_n}$, then

$$\sup_n \mu_n(\Omega) = \sup_n \mathscr{H}^{N-1}(J_{u_n}) < \infty.$$

Therefore, by compactness of $\mathscr{M}(\Omega)$ (Theorem 12), we can suppose that

$$\mu_n \overset{*}{\rightharpoonup} \mu \in \mathscr{M}(\Omega) \text{ (up to a sub-sequence).}$$

For the absolutely continuous part, note that

$$\sup_n \int_\Omega |\nabla u_n|^2 \, dx < \infty.$$

Since Ω is bounded, Proposition 3 with $\phi(t) = t^2$ ensures equi-integrability of (∇u_n). Therefore, by Theorem 4, it has a weakly convergent sub-sequence (denoted again by ∇u_n). So

$$\nabla u_n \rightharpoonup a \in [L^1(\Omega)]^N.$$

i.e. $\forall g \in [L^\infty(\Omega)]^N$, $\int_\Omega \nabla u_n \cdot g \, dx \longrightarrow \int_\Omega a \cdot g \, dx$. Therefore $\nabla u_n \mathscr{L}^N \rightharpoonup a\mathscr{L}^N$ in the sense of Radon measures. Theorem 40 is used to establish that $a = \nabla u$ and $\mu \geq D^j u$. Let $\psi \in W^{1,\infty}(\mathbb{R}) \cap C^1(\mathbb{R})$. First, note that

$$\psi'(u_n)\nabla u_n \rightharpoonup \psi'(u)a.$$

Indeed, let $\epsilon > 0$ and $\delta > 0$ be such that $|x - y| < \delta \Rightarrow |\psi'(x) - \psi'(y)| < \sqrt{\frac{\epsilon}{2\mathscr{L}^N(\Omega)}}$. Now, since $u_n \overset{L^1(\Omega)}{\longrightarrow} u$, u_n converges in measure to u, and then there exists a $M > 0$ such that

$$n \geq M \Rightarrow \mathscr{L}^N(\{x : |u_n(x) - u(x)| > \delta\}) < \frac{\epsilon}{4\|\psi'\|_\infty^2}.$$

Set $A_\delta = \{x : |u_n(x) - u(x)| > \delta\}$, then

$$\begin{aligned}
\int_\Omega |\psi'(u_n) - \psi'(u)|^2 \, dx &= \int_{A_\delta} |\psi'(u_n) - \psi'(u)|^2 \, dx + \int_{A_\delta^c} |\psi'(u_n) - \psi'(u)|^2 \, dx \\
&\leq 2\|\psi'\|_\infty^2 \mathscr{L}^N(A_\delta) + \mathscr{L}^N(\mathscr{A}_\delta^c) \frac{\epsilon}{2\mathscr{L}^N(\Omega)} \\
&\leq 2\|\psi'\|_\infty^2 \frac{\epsilon}{4\|\psi'\|_\infty^2} + \epsilon/2 \leq \epsilon/2 + \epsilon/2 = \epsilon.
\end{aligned}$$

So $\psi'(u_n) \overset{L^2(\Omega)}{\longrightarrow} \psi'(u)$. For any $g \in L^\infty(\Omega)$,

$$\int_\Omega \psi'(u_n)\nabla u_n g \, dx = \int_\Omega \nabla u_n g(\psi'(u_n) - \psi'(u)) \, dx + \int_\Omega \nabla u_n(\psi'(u)g) \, dx.$$

But by Cauchy-Schwartz inequality

$$\int_\Omega \nabla u_n g(\psi'(u_n) - \psi'(u)) \, dx \leq \|\nabla u_n g\|_{L^2(\Omega)} \|\psi'(u_n) - \psi'(u)\|_{L^2(\Omega)} \longrightarrow 0,$$

since $\|\nabla u_n g\|_{L^2(\Omega)}$ is bounded. Also, since $\psi'(u)g \in L^\infty(\Omega)$,

$$\int_\Omega \nabla u_n(\psi'(u)g)\,dx \longrightarrow \int_\Omega a\psi'(u)g\,dx.$$

Hence $\forall g \in L^\infty(\Omega)$,

$$\int_\Omega \psi'(u_n)\nabla(u_n)g\,dx \longrightarrow \int_\Omega a\psi' g\,dx,$$

i.e. $\psi'(u_n)\nabla u_n \rightharpoonup \psi'(u)a$.

From Theorem 39 (chain rule formula), it is clear that

$$\sup_n \|u_n\|_{BV(\Omega)} < \infty \quad \Longrightarrow \quad \sup_n \|\psi(u_n)\|_{BV(\Omega)} < \infty.$$

Moreover $\psi(u_n) \overset{L^1(\Omega)}{\longrightarrow} \psi(u)$, so by Theorem 30, $\psi(u_n) \overset{*}{\rightharpoonup} \psi(u)$ in $BV(\Omega)$, i.e. $D\psi(u_n) \overset{*}{\rightharpoonup} D\psi(u)$ in $\mathscr{M}(\Omega)$, which implies that

$$D\psi(u_n) - \psi'(u_n)\nabla u_n \mathscr{L}^N \overset{*}{\rightharpoonup} D\psi(u) - \psi'(u)a\mathscr{L}^N.$$

What is needed is a relation between the absolute values of these measures. Note that

$$\sup_n |D\psi(u_n) - \psi'(u_n)\nabla u_n \mathscr{L}^N|(\Omega) = \sup_n |D^s\psi(u_n)|(\Omega) \le \sup_n |D\psi(u_n)|(\Omega) < \infty.$$

Hence, there is a sub-sequence (that may depend on ψ) such that

$$\left|D\psi(u_{n_k}) - \psi'(u_{n_k})\nabla u_{n_k}\mathscr{L}^N\right| \overset{*}{\rightharpoonup} \lambda.$$

Hence Theorem 13 asserts that

$$|D\psi(u) - \psi'(u)a\mathscr{L}^N| \le \lambda.$$

$$|D\psi(u) - \psi'(u)a\mathscr{L}^N| \le \lambda \le 2\|\psi\|_\infty \mu.$$

Then by Theorem 40, $u \in SBV(\Omega)$. Moreover $a = \nabla u$ and $\mu \ge \mathscr{H}^{N-1}\llcorner J_u$. Hence

$$\nabla u_n \mathscr{L}^N \overset{*}{\rightharpoonup} \nabla u \mathscr{L}^N \quad \text{and} \quad D^j u_n = Du_n - \nabla u_n \mathscr{L}^N \overset{*}{\rightharpoonup} Du - \nabla u \mathscr{L}^N = D^j u.$$

Now, since $\nabla u_n \mathscr{L}^N \overset{*}{\rightharpoonup} \nabla u \mathscr{L}^N$, the lower semi-continuity property

$$\int_\Omega |\nabla u|^2 \, dx \leq \liminf_n \int_\Omega |\nabla u_n|^2 \, dx$$

follows from Theorem 12. Finally,

$$\mathscr{H}^{N-1}(J_u) \leq \mu(J_u) \leq \mu(\Omega) \leq \liminf_n \mu_n(\Omega) = \liminf_n \mathscr{H}^{N-1}(J_{u_n}),$$

where the inequality $\mu(\Omega) \leq \liminf_n \mu_n(\Omega)$ comes from lower semi-continuity of the total variation of Radon measures (see Theorem 12) □

Theorem 42 (Compactness in $SBV(\Omega)$). *Let $\Omega \subseteq \mathbb{R}^N$ be a bounded domain and $\{u_n\}_n \subseteq SBV(\Omega)$ such that $\sup_n \|u_n\|_\infty < \infty$ and*

$$\sup_n \left\{ \int_\Omega |\nabla u_n|^2 \, dx + \mathscr{H}^{N-1}(J_{u_n}) \right\} < \infty.$$

Then there exists a sub-sequence $u_{n_k} \overset{*}{\rightharpoonup} u \in SBV(\Omega)$.

Proof. It is sufficient to show that $\sup_n \|u_n\|_{BV} < \infty$ and apply Theorem 31 to obtain a converging sub-sequence $u_{n_k} \overset{*}{\rightharpoonup} u$.

First, note that

$$\sup_n \|u_n\|_{L^1(\Omega)} = \sup_n \int_\Omega |u_n| \, dx \leq \sup_n \|u_n\|_\infty \mathscr{L}^N(\Omega) < \infty.$$

Also,

$$\begin{aligned}
\sup_n |Du_n|(\Omega) &= \sup_n \left\{ \int_\Omega |\nabla u_n| \, dx + \int_{J_{u_n}} |u_n^+ - u_n^-| \, dx \right\} \\
&\leq \sup_n \left\{ \int_\Omega (|\nabla u_n|^2 + 1) \, dx + 2\|u_n\|_\infty \mathscr{H}^{N-1}(J_{u_n}) \right\} \\
&\leq \sup_n \left\{ \int_\Omega |\nabla u_n|^2 \, dx + \mathscr{L}^N(\Omega) + 2\|u_n\|_\infty \mathscr{H}^{N-1}(J_{u_n}) \right\} < \infty.
\end{aligned}$$

Therefore $\sup_n \|u_n\|_{BV(\Omega)} < \infty$ as needed. □

2.6. CALCULUS OF VARIATIONS

In the Calculus of Variations, one tries to determine the existence of some critical points that optimize functionals. Historically, mathematicians were interested in minima, since the

functionals of interest were deduced from laws of physics where the critical points represent physical solutions, satisfying a minimal energy principle.

Definition 43. *Let V be a Banach space and $F : V \longrightarrow \mathbb{R}$ be a continuous functional. The problem*

$$\inf_{u \in V} F(u) \tag{2.2}$$

*is a **minimization problem**. The goal is to determine the existence and the possible unicity of a minimizer u^*. The minimum of the functional would then be $F(u^*)$.*

If it is possible to define a derivative for F, it is natural to think that at a minimum u^*

$$DF(u^*) = 0.$$

This is the **Euler-Lagrange equation** of the functional F. It is a first order optimality condition. When the problem is unconstrained, this gives a necessary condition for u^* to be a local minimum. Note that, in the general case, $DF(u^*) = 0$ only implies that u^* is a critical point of F. It can be minimum, a maximum or a saddle point. Even, if it is a minimum, it may be only a local one.

Example 44. *Let $\Omega \subseteq \mathbb{R}$, $V = \{u \in \mathscr{C}^1(\Omega) : u|_{\partial\Omega} = h\}$ and*

$$\begin{aligned} F : V &\longrightarrow \mathbb{R} \\ u &\mapsto \int_\Omega L(x, u, u')\, dx, \end{aligned}$$

where $L : \Omega \times \mathbb{R} \times \mathbb{R}$ is a C^2 function. Suppose u is a minimum for F. Then for any $\phi \in C_0^\infty(\Omega)$, $g_\phi(t) = u + t\phi \in V$, and the real function

$$t \mapsto F(g_\phi(t))$$

has a critical point at $t = 0$. Using differentiation, integration by parts and the fact that this is valid for all ϕ, everything boils down to the Euler-Lagrange equation of F:

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial u'}\right) = 0.$$

In general, if V is a space of functions, the Euler-Lagrange equation $DF = 0$ will have the form of a partial differential equation. A classical idea is that if you follow the gradient flow, you must end at a critical point, i.e. a solution of the Euler-Lagrange equation

(under some compactness assumptions). Since minima are of great interest, one may try to follow flow lines in the directions of the negative gradient. These are given by the partial differential equation $u_t = -DF(u)$. Therefore, to find a local minimum, one picks an initial function u_0, and solve the problem

$$\begin{cases} u_t = -DF(u), \\ u(\cdot, 0) = u_0 \end{cases}$$

iteratively to obtain a sequence u_n that could converge to a local minimum. This approach is very convenient to find explicitly a critical point. Note that, if we can prove the convergence of such an algorithm, then using numerical computations the algorithm usually converges to a local minimum, since other critical points are unstable. However the minimum may not be the global minimum. An important issue is the one of the existence of a solution, that will ensure that the sequence f_n converges. A standard way to tackle the existence problem is to use the direct method.

Definition 45. *To solve (2.2) by the **direct method**, first pick a minimizing sequence $(x_n)_n$, i.e.*

$$\lim_{n\to\infty} F(x_n) = \inf_{x\in V} F(x) = c.$$

Then choose a suitable topology τ on V, in the sense that the following steps are feasible.

1. *Show that (x_n) admits a convergent sub-sequence*

$$x_{n_k} \xrightarrow{\tau} x$$

   for the topology τ.

2. *Show that*

$$F(x) = c.$$

   (This step is non trivial because F may not be continuous with respect to τ.)

The way to achieve this highly depends on the space V and on the functional F. A common hypothesis is coercivity.

Definition 46. *A function $F : V \longrightarrow \mathbb{R}$ is **coercive** if*

$$\lim_{|x|_V\to\infty} F(x) = \infty.$$

Proposition 47. *Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ be continuous and coercive. Then the problem*

$$\inf_{x \in V} F(x)$$

admits a solution.

Proof. Let $(x_n)_n$ be a minimizing sequence. By coercivity $(x_n)_n$ is bounded. It admits a converging sub-sequence $x_{n_k} \longrightarrow x$ for the usual topology. Then

$$F(x) = F\left(\lim_{n_k} x_{n_k}\right) = \lim_{n_k} F(x_{n_k}) = \inf_{x \in V} F(x).$$

□

In many case, it is too restrictive to consider only continuous functions. We can relax this assumption by considering lower-semi-continuous functions.

Definition 48. *F is **τ-lower-semi-continuous** (l.s.c.) if*

$$\forall x_n \xrightarrow{\tau} x, \quad \liminf_n F(x_n) \geq F(x).$$

Equivalently

$$f \text{ is l.s.c.} \quad \Leftrightarrow \quad \forall c \in \mathbb{R}, \{F > c\} \text{ is open}$$

Remark 49. *If $V = \mathbb{R}^n$ and τ denotes the usual topology on $\mathbb{R}^n$, then*

$$F \text{ is l.s.c. and coercive} \quad \Leftrightarrow \quad \forall c \in \mathbb{R}, V_c = \{F \leq c\} \text{ is compact.}$$

Proposition 50. *Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ be l.s.c. and coercive. Then the problem*

$$\inf_{x \in V} F(x)$$

admits a solution.

Proof. Let $(x_n)_n$ be a minimizing sequence and $c_n = F(x_n)$. Then

$$V_{c_0} \supseteq V_{c_1} \supseteq V_{c_2} \supseteq \ldots$$

is a decreasing sequence of compacts. There must exist $x \in V$ that belongs to each V_{c_n}, then $\forall n, F(x) \leq c_n$. □

In these examples, it has not been necessary to pick another topology because the space V was of finite dimension. But it appears that in most situations, a weaker topology is needed. Indeed, if the functional is coercive, then minimizing sequence are bounded, but in general one cannot hope that a bounded sequence admits a convergent sub-sequence. Hence, it is necessary to choose a coarser topology τ that has more compact sets than the usual (strong) topology. At the same time τ must be fine enough to have the lower semi-continuity of F.

Theorem 51. *Let V be a reflexive Banach space (that is $V^{**} = V$). Then a bounded sequence $(x_n)_n$ in V admits a weakly convergent sub-sequence*

$$x_{n_k} \rightharpoonup x.$$

This leads to

Theorem 52. *Let V be a reflexive Banach space and $F : V \longrightarrow \mathbb{R}$ be a coercive functional, which is weakly l.s.c., then the problem*

$$\inf_x F(x)$$

admits a solution.

Weak l.s.c. may be hard to prove. The following criterion is often used.

Theorem 53. *Let V be a Banach space and $F : V \longrightarrow \mathbb{R}$ be a convex functional, then*

$$F \text{ is l.s.c.} \quad \Leftrightarrow \quad F \text{ is weakly l.s.c.}.$$

Then Theorem 52 becomes

Theorem 54. *Let V be a reflexive Banach space and*

$$F : V \longrightarrow \mathbb{R}$$

be a coercive convex l.s.c. functional, then the problem

$$\inf_x F(x)$$

admits a solution.

But this result does not hold for functionals on $L^1(\Omega)$, since it is not a reflexive space. It is possible to prove the following results for functional on $L^1(\Omega)$.

Theorem 55. *Let $F(x,u) : \Omega \times \mathbb{R}^s \longrightarrow \bar{\mathbb{R}}$ be a C^1 function and $u_n, u \in L^1(\Omega)$. If one of the following holds*

1. $u_n \overset{L^1(\Omega)}{\longrightarrow} u$ *or*
2. $u_n \overset{L^1(\Omega)}{\rightharpoonup} u$ *and for all $x \in \Omega$, the function $F(x,\cdot)$ is convex,*

then

$$\int_\Omega F(x,u(x))\,dx \leq \liminf_{k\to\infty} \int_\Omega F(x,u_n(x))\,dx.$$

Example 56 (Poisson equation). *A solution of the Poisson equation with homogeneous Dirichlet boundary conditions*

$$\begin{cases} -\Delta u = f & \text{on } \Omega, f \in L^2(\Omega) \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

is a minimum of the following functional defined on $H_0^1(\Omega) = \overline{\mathscr{D}(\Omega)}^{H^1(\Omega)}$

$$F(u) = \frac{1}{2}\int_\Omega |\nabla u|^2\,dx - \int_\Omega fu\,dx.$$

First, this functional admits a minimum. Indeed, $|\nabla u|^2 - fu$ is convex, so that $F(u)$ is a convex functional. Moreover if $u_n \overset{H^1(\Omega)}{\longrightarrow} u$, then $\|\nabla u_n\|^2_{L^2(\Omega)} \to \|\nabla u\|^2_{L^2(\Omega)}$, so that

$$\begin{aligned} |F(u_n) - F(u)| &\leq \frac{1}{2}\left|\|\nabla u_n\|^2_{L^2(\Omega)} - \|\nabla u\|^2_{L^2(\Omega)}\right| + \int_\Omega |f(u_n - u)|dx \\ &\leq \frac{1}{2}\left|\|\nabla u_n\|^2_{L^2(\Omega)} - \|\nabla u\|^2_{L^2(\Omega)}\right| + \|f\|_{L^2(\Omega)}\|u_n - u\|_{L^2(\Omega)} \overset{n\to\infty}{\longrightarrow} 0. \end{aligned}$$

The last inequality comes from Hölder inequality. Hence the functional F is continuous, therefore l.s.c. To prove that the functional is coercive, recall that

$$F(u) = \|\nabla u\|^2_{L^2(\Omega)} - \int_\Omega fu\,dx.$$

It will be necessary to use the Poincare inequality to get

$$\|\nabla u\|_{L^2(\Omega)} \geq \mathscr{C}\|u\|_{H_0^1(\Omega)}$$

Hence the term $\|\nabla u\|^2_{L^2(\Omega)}$ *is at least quadratic in* $\|u\|_{H^1_0(\Omega)}$. *On the other side, by Hölder inequality,*

$$\left|\int_\Omega f u dx\right| \leq \tilde{\mathscr{C}}\|u\|_{L^2(\Omega)} \leq \tilde{\mathscr{C}}\|u\|_{H^1_0(\Omega)},$$

which gives that the second term is less than linear in $\|u\|_{H^1_0(\Omega)}$. *Therefore*

$$\lim_{\|u\|_{H^1_0(\Omega)}\to\infty} F(u) = \infty.$$

So that F is continuous convex and coercive. Since $H^1_0(\Omega)$ *is reflexive, Theorem 54 ensures that F admits a minimum.*

To actually find the minimum, we obtain the Euler-Lagrange equation of F by computing directional derivatives of F. If u is a minimum, all the directional derivatives should vanish. Let $v \in \mathscr{D}(\Omega)$, *then*

$$\begin{aligned}
\frac{\partial F}{\partial v}(u) &= \frac{d}{dt}F(u+tv)\Big|_{t=0} \\
&= \frac{d}{dt}\frac{1}{2}\int_\Omega |\nabla u + t\nabla v|^2 dx - \int_\Omega f(u+tv)dx\Big|_{t=0} \\
&= \frac{1}{2}\int_\Omega \frac{d}{dt}|\nabla u + t\nabla v|^2\Big|_{t=0} dx - \int_\Omega \frac{d}{dt}f(u+tv)\Big|_{t=0} dx \\
&= \int_\Omega (\nabla u + t\nabla v)\cdot\nabla v\Big|_{t=0} dx - \int_\Omega fv\Big|_{t=0} dx \\
&= \int_\Omega \nabla u\cdot\nabla v dx - \int_\Omega fv dx \\
&= (\nabla u, \nabla v)_{\mathscr{D}(\Omega)} - (f, v)_{\mathscr{D}(\Omega)} \\
&= (-\Delta u, v)_{\mathscr{D}(\Omega)} - (f, v)_{\mathscr{D}(\Omega)} \\
&= (-\Delta u - f, v)_{\mathscr{D}(\Omega)} \\
&= 0, \quad \forall v \in \mathscr{D}(\Omega).
\end{aligned}$$

Hence $-\Delta u = f$ *in the sense of distributions. The solutions of that equation correspond to critical points of the functional. In order to solve numerically, it is possible to do a gradient descent which amounts to solving the initial value problem*

$$\begin{cases} u_t = \Delta u + f \\ u(x,0) = g, \end{cases}$$

where g is some initial guess.

2.7. MUMFORD-SHAH FUNCTIONAL

2.7.1. Continuous Mumford-Shah problem

We now present the important minimization problem proposed by Mumford and Shah [72], that we outlined in Section 1.4. Following the same notations as before, they proposed that minimizing the functional

$$F(u,K) = \int_{\Omega\setminus K} |\nabla u|^2 + \int_\Omega (u-g)^2 + \mathscr{H}^{N-1}(K\cap\Omega)$$

over all admissible pairs $(K,u) \in \mathscr{A}$, where

$$\mathscr{A} = \left\{(K,u) : K \subseteq \overline{\Omega} \text{ compact }, u \in W^{1,2}_{\text{loc}}(\Omega\setminus K)\right\},$$

should give a good segmentation of the image $g \in L^\infty(\Omega)$. To prove that this problem admits a solution is really non trivial.

Let us try to solve this problem by the direct method. Take a minimizing sequence (u_n, K_n). We can suppose that $\|u_n\|_\infty$ is uniformly bounded, say by $\|g\|_\infty$, otherwise, we can just truncate the sequence. The first thing we have to do is to provide the space $X = \{K \subseteq \overline{\Omega}, K \text{ compact}\}$ a topology.

Definition 57. *1. For $K \subseteq \overline{\Omega}$, we define $K_\epsilon = \{x \in X : \text{dist}(x,K) \leq \epsilon\}$.*

2. The Hausdorff distance between two compacts K and K' is

$$\delta(K,K') = inf\{\epsilon : K \subseteq K'_\epsilon \text{ and } K' \subseteq K_\epsilon\}.$$

Theorem 58 (Blashke). *(X,δ) is a compact metric space.*

Now, we have the tools to show that the minimizing sequence (K_n, u_n) admits a converging sub-sequence. Since any sequence in X is bounded (Ω is bounded), we can suppose that our sequence K_n converges to some K. Now we wonder if u_n is converging to some function $u \in W^{1,2}_{\text{loc}}(\Omega)$. Let $A \subseteq B \subseteq \Omega$, with A open, B closed such that $B \cap K = \emptyset$. Then clearly there exists N such that for $n \geq N$, $B \cap K_n = \emptyset$. Then for $n \geq N$, $u_n \in W^{1,2}(A)$. Now since $\|u_n\|_\infty$ is uniformly bounded, there exists a sub-sequence converging weakly to $u \in W^{1,2}(A)$. By a diagonal argument, ranging over open subsets A of Ω, we can conclude that there is a sub-sequence of u_n converging weakly to $u \in W^{1,2}_{\text{loc}}(\Omega)$. See Ambrosio [5] for details.

So the sequence (K_n, u_n) has a sub-sequence that is converging to some pair (K, u). The problem is that we cannot show that this pair actually minimizes the functional, since the functional is not lower semi-continuous with respect to this convergence. This comes from the fact that the function

$$K \mapsto \mathscr{H}^{N-1}(K)$$

is not l.s.c. In fact, any K can be approximated in an obvious way by discrete sets, which have $\mathscr{H}^{N-1}$-measure 0.

To bypass this difficulty, we have to go through a weak formulation of the problem. We search for $u \in SBV(\Omega)$ that minimizes the functional

$$F(u) = \int_\Omega |\nabla u|^2 + \int_\Omega (u-g)^2 + \mathscr{H}^{N-1}(S_u),$$

where now ∇u is the approximate gradient of u, ie. $Du = \nabla u \mathscr{L}^n + D^j u$, and S_u is the set of approximate discontinuities of u.

Remark 59. *It is a fact that Cantor functions (ie. a function with $Du = D^c u$) are dense in $L^2(\Omega)$. Those functions have $S_u = \emptyset$ and have zero derivatives almost everywhere. Therefore*

$$\inf_{u \in BV(\Omega)} F(u) = 0;$$

This suggests that we really need to minimize over $SBV(\Omega)$.

The following theorem is crucial (see Ambrosio [5] for details). It shows the equivalence of the two formulations.

Theorem 60.

$$\inf_{u \in SBV(\Omega)} F(u) = \inf_{(K,u) \in \mathscr{A}} J(K,u).$$

The next important step is to prove that the weak formulation admits a solution.

Theorem 61. *Let $g \in L^\infty(\Omega)$. Then there exists $u \in SBV(\Omega)$ such that*

$$F(u) = \inf_{v \in SBV(\Omega)} F(v).$$

Proof. F is bounded from below. So take a minimizing sequence u_n for F. We can suppose without loss of generality that $\|u_n\|_\infty \leq \|g\|_\infty$, since otherwise the sequence of functions $u'_n = min(\|g\|_\infty, u_n)$ is such that $F(u'_n) \leq F(u_n)$. Therefore u'_n is also a minimizing sequence and satisfies the given property.

Since u_n is a minimizing sequence for F, clearly

$$\sup_n \left\{ \int_\Omega |\nabla u_n|^2 dx + \mathscr{H}^{N-1}(S_u) \right\} \leq \infty.$$

Then, using theorem 42, there exists $u \in SBV(\Omega)$ and a sub-sequence u_{n_k} of u_n such that $u_{n_k} \overset{*}{\rightharpoonup} u$. Now the map $v \mapsto \int_\Omega (g-v)^2 dx$ is strongly l.s.c. in $L^1(\Omega)$ by theorem 55. This, together with the second part of the theorem 41 ensures that

$$F(u) \leq \liminf_k F(u_{n_k}).$$

Thus $u \in SBV(\Omega)$ is a minimizer of F. □

2.7.2. Piecewise constant Mumford-Shah problem

In practice, it is often necessary to simplify the problem to be able to compute the minimum numerically. One way is to to minimize the Mumford-Shah functional over piecewise constant functions in $SBV(\Omega)$. It will produce good segmentation results only if the image is close to be constant inside the objects of interest. First, the concept of a piecewise constant function has to be defined carefully (see [5] for details).

Definition 62. *A partition $\{V_k\}_{k\in\mathbb{N}}$ of Ω is a **Cacciopoli partition** of Ω if*

$$\sum_k P(V_k, \Omega) < \infty,$$

*where $P(V_k, \Omega)$ denotes the perimeter of V_k in Ω. The partition is said to be **ordered** if the sequence $(|V_k|)$ is decreasing.*

Definition 63. *A function $u : \Omega \longrightarrow \mathbb{R}$ is said to be **piecewise constant** if there exists a Cacciopoli partition $\{V_k\}$ of Ω and a sequence $\{t_k\}$ of real numbers such that*

$$u = \sum_k t_k \mathbb{1}_{V_k}.$$

Let $PC(\Omega)$ the set of piecewise constant functions over Ω.

Remark 64. *The numbers t_k need not be distinct. But, they can be made distinct by ordering the set of values $\{t_{n_1}, t_{n_2}, ...\}$ of u and taking the Cacciopoli partition $\{F_l\}$ of Ω given by*

$$F_l = u^{-1}(t_{n_l}).$$

We call the $\{F_l\}$ *the* ***level set Cacciopoli partition*** *of* Ω *associated to* u.

It is important to understand how the discontinuity set of a piecewise constant function u can be. It turns out that $\mathscr{H}^{N-1}$-almost everywhere, $S_u \subseteq \bigcup_k \partial V_k \cap \Omega$. Therefore

$$2\mathscr{H}^{N-1}(S_u) \leq \sum_k P(V_k, \Omega).$$

It is clear that the equality need not be attained when the values t_k are not distinct. It turns out that this is the only possibility.

Theorem 65. *Let* u *be piecewise constant function with level set Cacciopoli partition* $\{V_k\}$ *of* Ω. *Then*

$$2\mathscr{H}^{N-1}(S_u) = \sum_k P(V_k, \Omega).$$

Cacciopoli partitions have nice compactness properties.

Theorem 66. *Let* $\{V_{k,n}\}_n$ *be a sequence of Cacciopoli partitions of* Ω *such that*

$$\sup_n \sum_k P(V_{k,n}, \Omega) < \infty.$$

Then there exists a Cacciopoli partition $\{V_k\}$ *of* Ω *and a sub-sequence* n_l *such that for all* k

$$\mathbb{1}_{V_{k,n_l}} \xrightarrow{L^1} \mathbb{1}_{V_k}.$$

With that in mind, it is possible to prove the existence of a minimizer for the piecewise constant Mumford-Shah problem.

Theorem 67. *Let* $g \in L^\infty(\Omega)$. *Then there exists* $u \in SBV(\Omega) \cap PC(\Omega)$ *such that*

$$F(u) = \inf_{v \in SBV(\Omega) \cap PC(\Omega)} F(v).$$

3

Image denoising

An image can be seen as a function

$$g : \Omega \longrightarrow [0, 1]$$

representing the gray tones (0 for black and 1 for white). The domain Ω is in general a rectangle or a cube. The domain is bounded and there is a finite number of pixels in the image, hence it is natural to assume that

$$\|g\|_\infty < \infty.$$

Denoising an image is a complex task that is highly dependent on the image nature. Several types of noise can be found in images and they require different denoising methods. The typical noise found in CT or MRI images is that some pixels, or small packets of pixels, will have a value slightly off the real value they should have. These errors can be large enough to lead to irrelevant segmentation. To repair these pixels, a common strategy is to apply a smoothing algorithm to the noisy image.

We present here several denoising techniques that admit PDE formulations. PDE methods for denoising have proved their efficiency and are widely used for medical images. The PDEs that are involved are closely related to the PDEs that arise for the segmentation problem. There are several good references on the subject, let us only mention the book by Chan and Shen [20] and the one by Aubert and Kornprobst [9].

In this chapter we sketch the different methods as well as their numerical implementations. This will serve later when we get to the segmentation problem in Chapter 4.

3.1. HEAT EQUATION

The best known smoothing PDE is certainly the heat equation:

$$\begin{cases} u_t = \mu \Delta u, & \text{on } \Omega \times (0, T) \\ \frac{\partial u}{\partial n} = 0, & \text{on } \partial\Omega \\ u(\cdot, 0) = g & \text{on } \Omega. \end{cases} \tag{3.1}$$

The physical interpretation of this problem is that Ω can be seen as a body with thermal conductivity μ and temperature distribution given by g at time 0. The solution $u(x, t)$ represents the temperature at position x and time t. Heat diffuses in the body and smoothens temperature differences at neighboring points. To solve the equation, it is necessary to specify boundary conditions. A common choice is to impose $\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$, which means that no heat leaves the domain.

It is easy to solve Problem 3.1 by finite differences when the domain is rectangular. The domain is naturally discretized using the pixels of the image to define an orthogonal grid. Let $u_{i,j}^n$ denote the value of the approximate solution solution $\bar{u}$ at pixel $x_{i,j}$ and time $n\Delta t$. As the spacing between pixel is constant here, we can take $\Delta x = \Delta y = 1$. Then

$$(u_t)_{i,j}^n \approx \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}.$$

The Laplacian discretized by central differences gives

$$(\Delta u)^n = u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n.$$

To solve the equation, it is necessary to choose at what time the Laplacian is to be evaluated. There are 3 main schemes:

1. $\Delta u \approx (\Delta u)^n$ (explicit)
2. $\Delta u \approx (\Delta u)^{n+1}$ (implicit)
3. $\Delta u \approx \frac{(\Delta u)^n + (\Delta u)^{n+1}}{2}$ (Crank-Nicholson)

The first scheme is called explicit since it is possible to express u^{n+1} directly in term of u^n from the discretized equation without factoring any matrix:

$$u_{i,j}^{n+1} = u_{i,j}^n + \mu \Delta t (u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n).$$

The other schemes need the resolution of a linear system of the form

$$Bu^{n+1} = Au^n \quad \Rightarrow \quad u^{n+1} = B^{-1}Au^n,$$

provided B is invertible. These schemes are less sensitive to instability. However, due to the size of the 3D medical images (about 50 000 000 voxels), it is a huge challenge to solve such linear systems.

The desired smoothed image is the solution $u(x, \tau)$ for some time τ not too large. At steady state ($u_t = 0$), we have $u_{i,j}^n = u_{i,j}^{n+1} = u_{i,j}$, thus $\Delta u = 0$ leads to

$$u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}}{4}.$$

This means that the value at a pixel is the mean of the values of the neighboring pixels. This is a discrete version of the property of harmonic functions which says that

$$\fint_{B_r(x)} u(y)dy = u(x)$$

for any ball $B_r(x) \subseteq \Omega$ about x.

One of the problem of that method is that it has the tendency to smooth edges. Another important issue is to determine the stopping time τ that gives the best smoothed image. That will be discussed later on. Increasing the conductivity μ makes the solution to converge to steady state in less time, but the explicit scheme needs smaller time steps to remain stable, generally giving the same computational time.

Figure 3.1 shows images smoothed with the heat equation for different numbers of time steps. The white dotted line in the last 3 images is an important contour taken from the initial image and reproduced in subsequent images. It is possible to see that the corresponding contour in the blurred images tend to slightly shift from the original one as the image gets smoother and smoother.

3.2. INHOMOGENEOUS DIFFUSION

A possible alternative to the preceding method is to consider that the domain's conductivity is not constant. The conductivity could be larger away from edges, and smaller on edges.

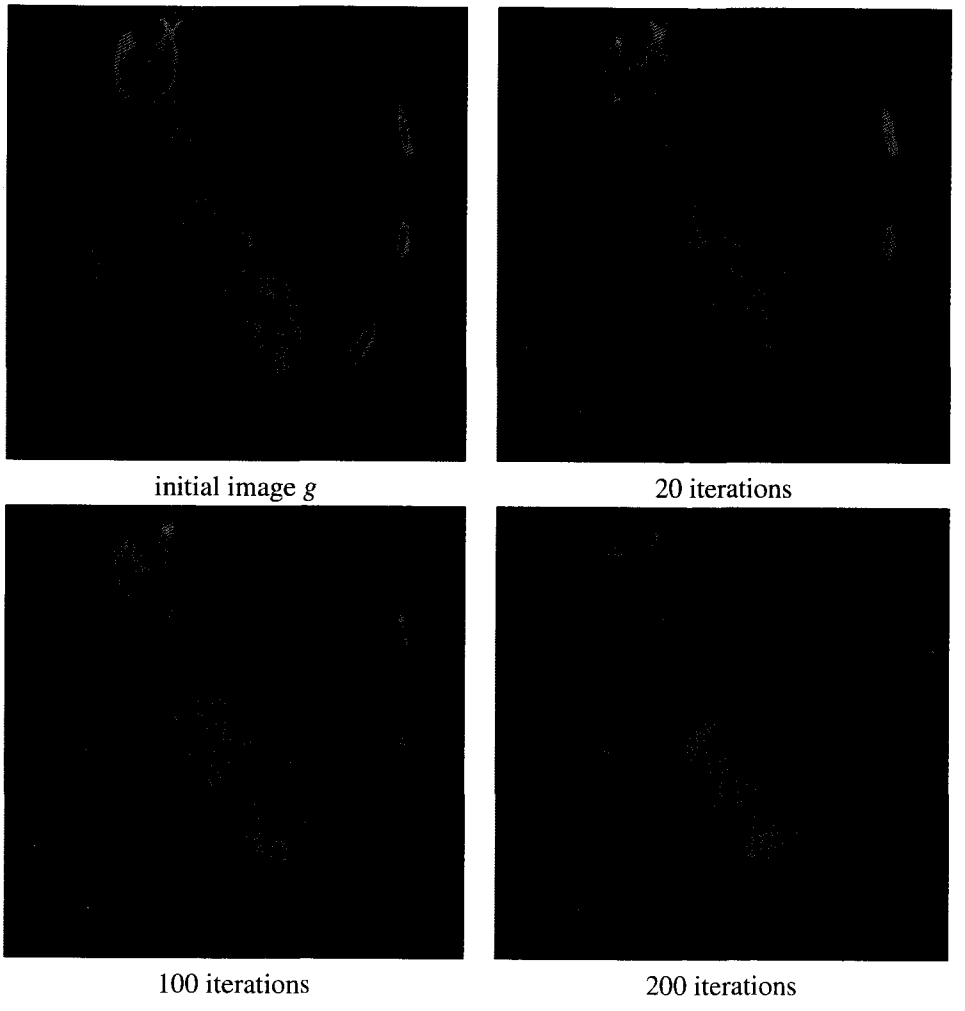


Figure 3.1: Smoothing of a 512×512 slice of a CT image via the heat equation with $\Delta t = 0.1$ and $\mu = 1$.

The heat equation in divergence form is

$$\begin{cases} u_t = \operatorname{div}(\mu \nabla u), & \text{on } \Omega \\ \frac{\partial u}{\partial n} = 0, & \text{on } \partial\Omega \\ u(\cdot, 0) = g & \text{on } \Omega. \end{cases} \tag{3.2}$$

where now μ is some function. Perona and Malik ([83]) initially proposed

$$\mu = e^{-\frac{|\nabla g|^2}{\sigma^2}}$$

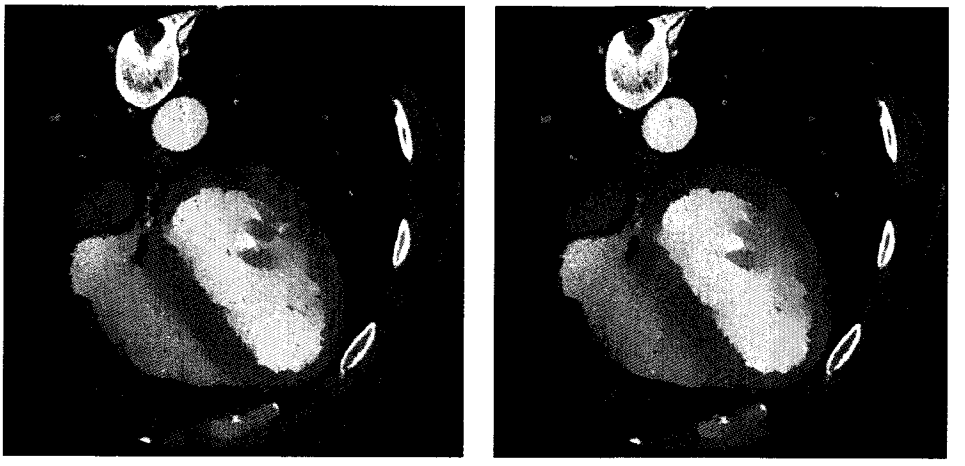
for some constant σ that depends on the actual image. A finite difference discretization of Problem 3.2 is

$$\begin{aligned} u_{i,j}^{n+1} =& u_{i,j}^n + \Delta t(c_{i+1,j}^n(u_{i+1,j}^n - u_{i,j}^n) - c_{i-1,j}^n(u_{i,j}^n - u_{i-1,j}^n) \\ &+ c_{i,j+1}^n(u_{i,j+1}^n - u_{i,j}^n) - c_{i,j-1}^n(u_{i,j}^n - u_{i,j-1}^n), \end{aligned}$$

where $c_{i,j}$ is an approximated value of μ at pixel (i, j). This scheme is directly derived from the divergence form of the equation. At steady state, this gives

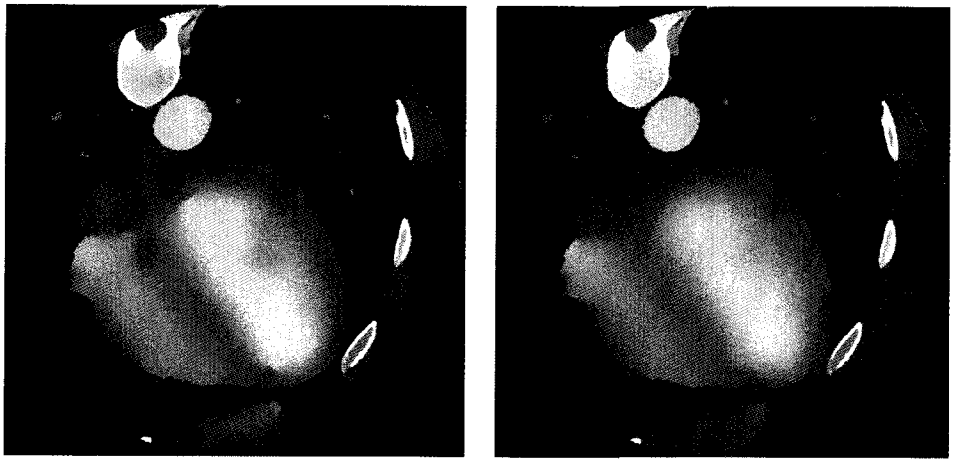
$$u_{i,j} = \frac{c_{i+1,j}u_{i+1,j} + c_{i-1,j}u_{i-1,j} + c_{i,j+1}u_{i,j+1} + c_{i,j-1}u_{i,j-1}}{c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1}},$$

that is, each pixel becomes a weighted average of its neighbors. The diffusion is inhomogeneous. If there is a jump between the pixels $x_{i,j}$ and $x_{i+1,j}$, then $c_{i+1,j}$ will be small and $u_{i,j}$ will depend mostly on the other pixels. It is possible that the steady state is a good smoothing. In practice, this will have the tendency to smooth too much, except edges that are clearly outlined. Figure 3.2 show results of anisotropic diffusion with $\sigma = 10$. The parameter σ is slightly too small since some noise is kept. On the other hand some edges tend to disappear as the one between the cavity of the heart and the myocardium. In Figure 3.3, the same calculation is made with $\sigma = 20$. Now the noise is completely removed but many edges are smoothed.



50 iterations 100 iterations

Figure 3.2: Smoothing via anisotropic diffusion with $\Delta t = 0.1$ and $\sigma = 10$.



50 iterations 100 iterations

Figure 3.3: Smoothing via anisotropic diffusion with $\Delta t = 0.1$ and $\sigma = 20$.

3.3. SOBOLEV SMOOTHING

In this section, we start to investigate variational formulations of the denoising problem. The idea behind variational methods is that the ideal image will minimize a certain functional

$$E : V \longrightarrow \mathbb{R},$$

where V is the vector space of admissible images, that has to be well chosen in order for E to admit a minimum. The choice of the functional will be influenced by the nature of the noise in the image g.

One example is the the Sobolev smoothing functional.

$$E : H^1(\Omega) \longrightarrow \mathbb{R}, \tag{3.3}$$

$$u \mapsto \int_\Omega |\nabla u|^2 dx + \int_\Omega (u-g)^2 dx. \tag{3.4}$$

As mentioned earlier, the function g can always be assumed to be bounded (in $L^\infty(\Omega)$) since it has a finite set of values on the domain Ω. Moreover, as Ω is bounded, the function g is also in $L^2(\Omega)$, so that the second term in the energy is well defined. The functional E is coercive, convex and continuous (hence l.s.c.) on $W^{1,2}(\Omega)$. Therefore E reaches its minimum at some $u \in W^{1,2}(\Omega)$. We compute the directional derivatives of E at u:

$$\begin{aligned}
\frac{\partial E}{\partial v}(u) &= \frac{d}{dt} E(u+tv)\Big|_{t=0} \\
&= \frac{d}{dt} \int_\Omega |\nabla u + t\nabla v|^2 dx + \int_\Omega (u+tv-g)^2 dx \Big|_{t=0} \\
&= \int_\Omega \frac{d}{dt} |\nabla u + t\nabla v|^2 \Big|_{t=0} dx + \int_\Omega \frac{d}{dt} (u+tv-g)^2 \Big|_{t=0} dx \\
&= 2\int_\Omega (\nabla u + t\nabla v)\cdot \nabla v \Big|_{t=0} dx + 2\int_\Omega (u+tv-g)v \Big|_{t=0} dx \\
&= 2\left[\int_\Omega \nabla u \cdot \nabla v dx + \int_\Omega (u-g)v dx\right] \\
&= 2\left[(\nabla u, \nabla v)_{\mathscr{L}(\Omega)} + (u-g, v)_{\mathscr{L}(\Omega)}\right] \\
&= 2\left[-(\Delta u, v)_{\mathscr{L}(\Omega)} + (u-g, v)_{\mathscr{L}(\Omega)}\right] \\
&= 2(-\Delta u + (u-g), v)_{\mathscr{L}(\Omega)} \\
&= 0, \quad \forall v \in \mathscr{D}(\Omega).
\end{aligned}$$

Hence

$$\Delta u = u - g$$

in the sense of distributions, which is the Euler-Lagrange equation of E. The problem associated to the gradient descent method is

$$\textbf{(P3)} \begin{cases} u_t = \Delta u + (g - u), & \Omega \\ \frac{\partial u}{\partial n} = 0, & \partial\Omega \\ u(x, 0) = g. \end{cases}$$

In principle, the initial condition could be any function, but it is reasonable to take g as initial guess. The steady state of that equation will be a solution of $-\Delta u + u = g$, ie. $u = (I - \Delta)^{-1} g$. It corresponds to a local minimum of the functional. This is the desired image.

However, in this ideal image, sharp edges have disappeared. For example, let $\Omega = [-1, 1]$ and $g = \mathbb{1}_{[0,1]}$. g is an ideal image that does not need to be cleaned. Note that for any $g \in L^2([-1, 1])$,

$$(I - \Delta)^{-1}(g) \in H^2([-1, 1]) \subseteq C^1([-1, 1]). \tag{3.5}$$

Hence g is not a solution of the Sobolev smoothing problem. In fact the Sobolev smoothing tends to smooth edges. Indeed, let f_n be the sequence of functions

$$f_n = \begin{cases} 0 & x \in [-1, -1/n) \\ nx + 1, & x \in [-1/n, 0) \\ 1, & x \in [0, 1] \end{cases}$$

As an example, Figure 3.4 shows the function f_5. Now we can compute

$$\int_\Omega |\nabla f_n|^2 = n^2 \frac{1}{n} = n. \tag{3.6}$$

This calculation, shows that the steeper the function f_n is, the higher its Sobolev energy will be. This is not a desirable property since a clean image should have steep gradient on object's boundary. Notwithstanding this fact, if the L^1-norm is used instead, we have

$$\int_\Omega |\nabla f_n| = n \frac{1}{n} = 1. \tag{3.7}$$

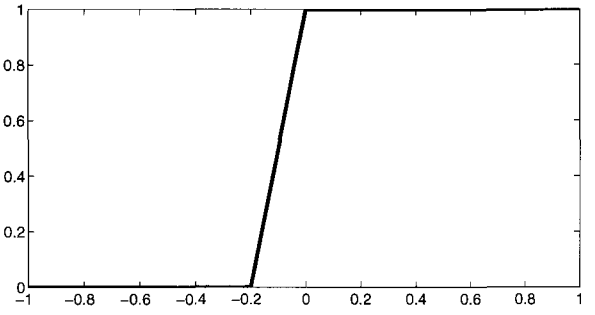


Figure 3.4: The function f_5.

This suggests that the L^1 norm is probably a better choice for conserving clear image edges. This leads to the total variation functional.

3.4. TOTAL VARIATION

Introduced by Osher, Fatemi and Rudin [92], the idea is to minimize the functional

$$E : BV(\Omega) \longrightarrow \mathbb{R},$$
$$u \mapsto E_{TV}(u, g) = |Du|(\Omega) + \lambda \int_\Omega (u - g)^2 dx. \tag{3.8}$$

Note that if $u \in W^{1,1}(\Omega)$, then

$$|Du|(\Omega) = \int_\Omega d|Du| = \int_\Omega |\nabla u| dx.$$

An obvious advantage is that discontinuous functions are admissible as long as their approximate discontinuity set is of finite $\mathscr{H}^{N-1}$ measure. In the last example, the image $g = \mathbb{1}_{(0,1)}$ is a minimizer of $E_{TV}(\cdot, g)$.

Let $v \in \mathscr{D}(\Omega)$. To compute the Euler-Lagrange equation of E_{TV}, the partial derivative

in the direction v is evaluated as

$$\begin{aligned}
\frac{\partial E_{TV}}{\partial v} &= \frac{d}{dt}|Du + t\nabla v|(\Omega)\Big|_{t=0} + 2\lambda(u-g) \\
&= \left(\frac{Du + t\nabla v}{|Du + t\nabla v|} \cdot \nabla v\right)(\Omega)\Big|_{t=0} + 2\lambda(u-g) \\
&= \left(\frac{Du}{|Du|} \cdot \nabla v\right)(\Omega) + 2\lambda(u-g) \\
&= \left(\frac{Du}{|Du|}, \nabla v\right)_{\mathscr{D}(\Omega)} + 2\lambda(u-g) \\
&= \left(-\operatorname{div}\left(\frac{Du}{|Du|}\right), v\right)_{\mathscr{D}(\Omega)} + 2\lambda(u-g) \\
&= \left(-\operatorname{div}\left(\frac{Du}{|Du|}\right) + 2\lambda(u-g), v\right)_{\mathscr{D}(\Omega)} \\
&= 0, \quad \forall v \in \mathscr{D}(\Omega).
\end{aligned}$$

Hence

$$\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) = \lambda(u-g)$$

in the sense of distributions, which is the Euler-Lagrange equation of E_{TV}. The problem associated to the gradient descent method is

$$\begin{cases} u_t = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \lambda(g-u), & \Omega \\ \frac{\partial u}{\partial n} = 0, & \partial\Omega \\ u(x,0) = g. \end{cases} \tag{3.9}$$

Again, the initial condition could be any function, but it is reasonable to take g as initial guess. The steady state of that equation will be the desired image. Note that for a function $u : \Omega \longrightarrow \mathbb{R}$, $\frac{\nabla u}{|\nabla u|}$ is a unit normal vector to the level curves and then $\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$ is simply the mean curvature of the level curves. Therefore the Total Variation algorithm will smooth the level curves of the initial image g. The smaller λ is, the more regular the steady state will be.

The Euler-Lagrange equation is non-linear and exhibits unstable numerical behavior, which forces us to take very small time steps, especially when an explicit time stepping scheme is used. Implicit time-stepping allows us to use larger time steps. On the other hand, since the equation is nonlinear, the equation needs to be linearized and solved via Newton's method, or an equivalent method at each time step. This requires solving several

linear systems and is very demanding for large 3D images. On the other hand, an explicit solver is easily parallelizable.

Another challenging part is to discretize the curvature term $\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$. This term is highly non linear and must be discretized carefully as it may blow up when u is nearly flat. There are several ways to discretize this term. A first approach is to compute

$$\begin{aligned}\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) &= \frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right)+\frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2+u_y^2}}\right) \\ &= \frac{u_y^2 u_{xx}-2u_x u_y u_{xy}+u_x^2 u_{yy}}{(u_x^2+u_y^2)^{3/2}},\end{aligned} \tag{3.10}$$

where the last expression is obtained by computing directly the partial derivatives. Then each of u_x, u_y, u_{xx}, u_{yy}, u_{xy} can be discretized by centered differences. It is a natural choice to use central differences since $\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$ being a diffusive term, there is no preferred direction of propagation.

Recall that the curvature of a curve at a point x is $1/r$, where r is the radius of the largest tangent circle to the curve at x. Then it seems natural to impose that

$$\left|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)\right|_{i,j} < \frac{1}{\Delta x},$$

since we cannot hope to resolve curvature in the image at points where the given radius is smaller than the size of a pixel. The condition can be imposed by truncating the term $\left|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)\right|_{i,j}$, setting it to $\pm\frac{1}{\Delta x}$, when it is too large. It turns out that imposing this condition stabilizes the scheme.

Another approach is direct discretization:

$$\begin{aligned}\frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right)(x_{i,j}) \approx & \frac{\frac{(u_{i+1,j}-u_{i,j})}{h}}{\sqrt{\frac{(u_{i+1,j}-u_{i,j})^2}{h^2}+\frac{(u_{i,j+1}-u_{i,j-1})^2}{4h^2}+\delta}} \\ & - \frac{\frac{(u_{i,j}-u_{i-1,j})}{h}}{\sqrt{\frac{(u_{i,j}-u_{i-1,j})^2}{h^2}+\frac{(u_{i-1,j+1}-u_{i-1,j-1})^2}{4h^2}+\delta}},\end{aligned} \tag{3.11}$$

where δ is some small parameter (typically $\delta = 10^{-10}$) that avoids divisions by zero. With the introduction of this parameter, it is not necessary to test whether $\left|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)\right|_{i,j} < \frac{1}{\Delta x}$ or not.

The partial derivatives are discretized sometimes via forward, sometimes via centered differences. This choice may seem strange, but a completely centered scheme would involve values that are two pixels away from the center $x_{i,j}$, which is a wide stencil.

To avoid the use of a non-centered scheme, it is possible to introduce middle points $x_{i\pm1/2,j}$ and $x_{i,j\pm1/2}$, together with values

$$u_{i\pm1/2,j} = \frac{u_{i,j} + u_{i\pm1,j}}{2} \qquad \text{and} \qquad u_{i,j\pm1/2} = \frac{u_{i,j} + u_{i,j\pm1}}{2}.$$

Then a centered scheme would be

$$\begin{aligned}\frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right)(x_{i,j}) \approx & \frac{\frac{(u_{i+1,j}-u_{i,j})}{h}}{\sqrt{\frac{(u_{i+1,j}-u_{i,j})^2}{h^2} + \frac{(u_{i+1/2,j+1}-u_{i+1/2,j-1})^2}{4h^2}} + \delta} \\ & - \frac{\frac{(u_{i,j}-u_{i-1,j})}{h}}{\sqrt{\frac{(u_{i,j}-u_{i-1,j})^2}{h^2} + \frac{(u_{i-1/2,j+1}-u_{i-1/2,j-1})^2}{4h^2}} + \delta}\end{aligned} \tag{3.12}$$

If we discretize the equation

$$u_t = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$$

with the scheme (3.11) and a forward Euler scheme in time, we get

$$\begin{aligned}u_{i,j}^{n+1} = u_{i,j}^n + \Delta t((&C_{i,j}u_{i+1,j}^n + C_{i-1,j}u_{i,j}^n + D_{i,j}u_{i,j+1}^n + D_{i,j-1}u_{i,j-1}^n \\ &- (C_{i,j} + C_{i-1,j} + D_{i,j} + D_{i,j-1})u_{i,j}^n)),\end{aligned} \tag{3.13}$$

where

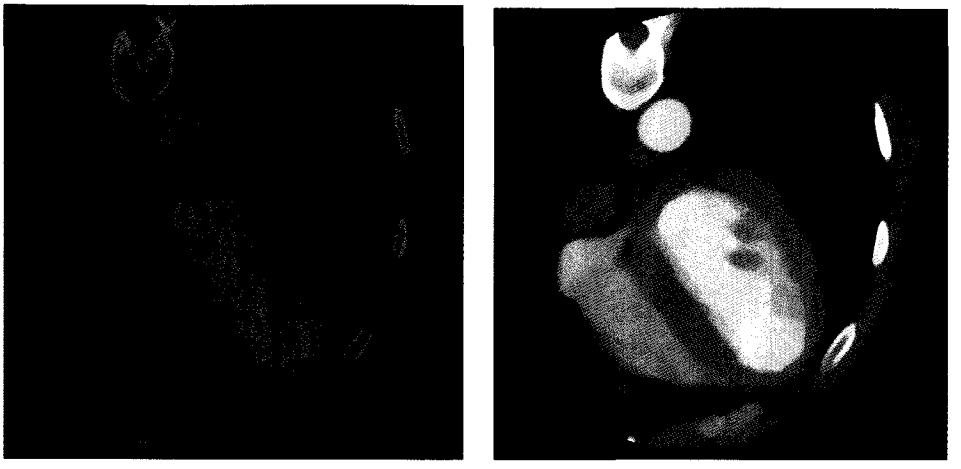
$$C_{i,j}^n = \frac{1}{\sqrt{\frac{(u_{i+1,j}^n-u_{i,j}^n)^2}{h^2} + \frac{(u_{i,j+1}^n-u_{i,j-1}^n)^2}{4h^2}}}$$

and similarly

$$D_{i,j}^n = \frac{1}{\sqrt{\frac{(u_{i,j+1}^n-u_{i,j}^n)^2}{h^2} + \frac{(u_{i+1,j}^n-u_{i-1,j}^n)^2}{4h^2}}}.$$

Smereka [103] suggested that replacing $u_{i,j}^n$ by $u_{i,j}^{n+1}$ in the right hand side of Equation (3.13) improves the stability of the algorithm. This can still be solved explicitly; indeed

$$u_{i,j}^{n+1} = \frac{1}{C}\left(u_{i,j}^n + \Delta t((C_{i,j}u_{i+1,j}^n + C_{i-1,j}u_{i,j}^n + D_{i,j}u_{i,j+1}^n + D_{i,j-1}u_{i,j-1}^n\right), \tag{3.14}$$



Initial image 100 iterations

Figure 3.5: Smoothing via total variation with $\lambda = 1$, $dt = 0.1$.

where

$$C = 1 + \Delta t((C_{i,j} + C_{i-1,j} + D_{i,j} + D_{i,j-1}).$$

This is called a semi-explicit scheme. Experimentally, this scheme has proved his superiority over the others, but to our knowledge, there is no rigorous justification of that fact. It will be the preferred scheme for discretizing curvature terms.

Marquina and Osher [62] remarked that the total variation model may suffer from a so-called stair-casing effect. That is, the solution tend to be piecewise constant in regions where it should be smooth. They propose to multiply the right hand side of Equation (3.9) by $|\nabla u|$. The equation then becomes

$$u_t = |\nabla u| \left[\operatorname{div}\left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(g - u) \right]. \tag{3.15}$$

This has the effect of increasing the stability of the discretized problem as well as avoiding the stair-casing effect [62]. However, the PDE no longer comes from a variational problem. Figure 3.5 shows the result of applying this new equation for smoothing images.

Remark 68. *The variational approach suggests a criterion for what would be a good stopping time when an image is smoothed via the heat equation or by anisotropic diffusion: one should stop when either the Sobolev energy, or the total variation energy is minimal. This corresponds to minimizing the corresponding energy on a one-dimensional family of images parametrized by the time t, namely the solution $u(x,t)$ of the problems in Equations (3.1)*

and (3.2).

The total variation functional given in Equation 3.8 contains a fidelity term that is the L^2 distance between the denoised image u and the original image g:

$$\int_\Omega |g-u|^2 dx. \tag{3.16}$$

It is classical in many minimization problems to use an L^2 fidelity term. It has nice analytical properties and it is easy to handle. The idea of replacing the L^2 fidelity by an L^1 fidelity has first been introduced in signal processing by Alliney [2, 3, 4] and later in image processing by Nikolova [73] and Nikolova, Esedoglu and Chan [74].

The total variation functional with L^1 fidelity takes the form

$$E_{TVL^1}(u) = |Du|(\Omega) + \int_\Omega \lambda |u-g| dx. \tag{3.17}$$

It has been remarked that for TV denoising, the L^1 fidelity is more natural [18], since in this case the problem is scale-invariant.

This means that if the underlying image is g and u^* denotes the minimum of $E_{TVL^1}(\cdot, g)$, then cu^* is the minimum of $E_{TV}(\cdot, cg)$. This property that seems very natural and desirable for a noise removal algorithm is not satisfied by the original total variation algorithm.

Another interesting fact about the L^1 version of the total variation algorithm is that one can make a direct correspondence between the value of the weight parameter λ and the size of details to be kept in the image. Indeed Nikolova, Esedoglu and Chan [74] remarked that given details do not fade out continuously as the value of λ changes. On the contrary, for a given size of detail, there is a critical value of the parameter λ above which the detail is kept, and below which the detail disappears. This property is very interesting if, for example, one is aware about the nature and the variance of the noise in the image.

Total variation methods are very efficient for removing noise amongst relatively uniform objects. There has been many improvements of the method. Actually, the main area of development is to try to incorporate texture information into the models, see, for example, [10] or [56].

4

Segmentation problem

The segmentation problem consists in separating a given image into its significant components. The meaning of significant relies on the segmentation goals. Figures 4.1 and 4.2 shows examples of segmentation problems. In Figure 4.1, the goal is to extract the position of the cameraman in the image. In Figure 4.2, it is the heart muscle and heart cavities that are to be extracted. The two images are of very different nature, as of the objects to detect.

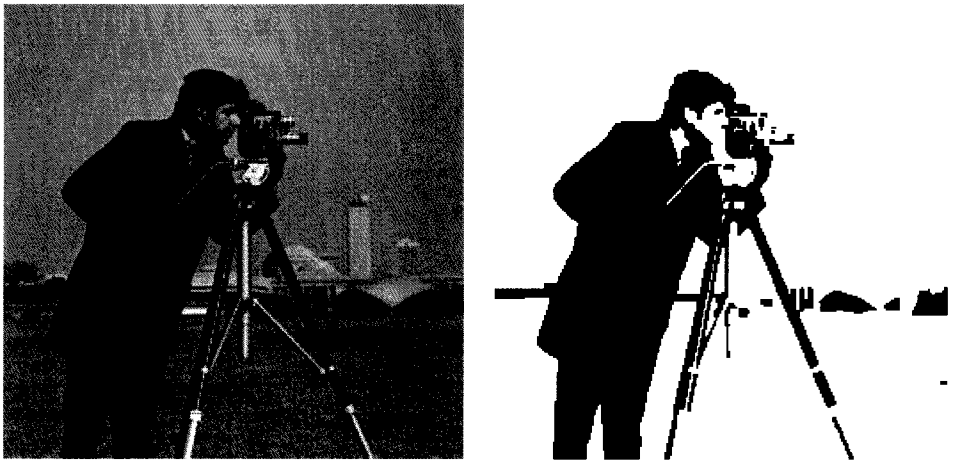


Figure 4.1: Segmentation of the cameraman image.

There exists a multitude of segmentation techniques, of very different nature. Each technique has strengths and drawbacks. Different applications will need different segmentation algorithms. The main factors that will guide the choice of a segmentation method are

1. The level of noise in the image,

2. The shape of the object of interest: its topology, the smoothness of its contours, its

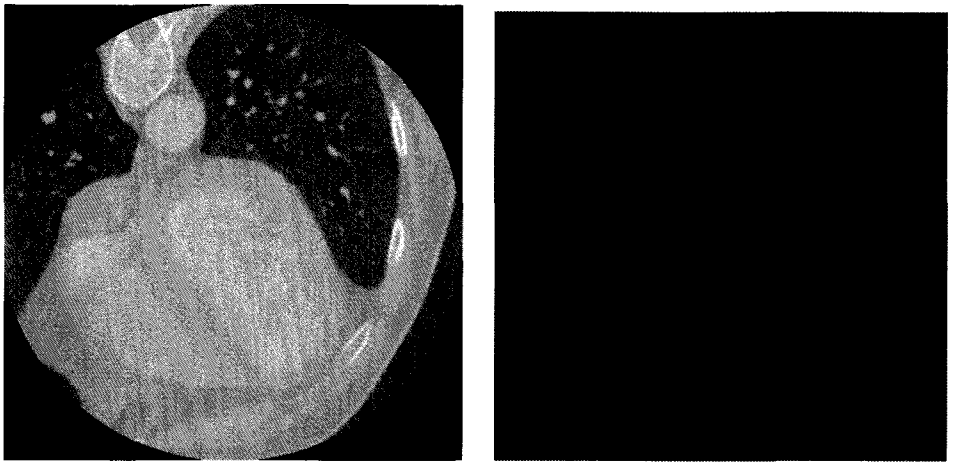


Figure 4.2: Heart segmentation from a 2D slice of a CT scan.

size,

3. The type of texture in the image,
4. The sharpness of the object's contour,
5. The image contrasts.

Among all segmentation methods, active contours (or deformable models) have proved their efficiency for applications to medical image segmentation. The idea is to let a curve evolve on the image until it stops on the edges of the object of interest. There are many variants of the original snake method proposed by Kass, Witkin and Terzopoulos [51]. The motion of the curve is generally driven by a partial differential equation. There are many variants of active contours algorithm. In some cases, the PDE is derived from a variational problem: the location and shape of the contour should minimize a given energy. In some other cases, the PDE is given ad-hoc. Another distinction between active contour methods is whether they are edge-based or region-based methods. In an edge based method, the contour evolves according to the presence of edges in the image. This is a more local method, as edge presence is a local property of the image. In region based methods, the evolution is driven by global properties of the regions delimited by the curve. It is also possible to add shape prior knowledge to a deformable model technique.

Other important methods in medical image segmentation are active shape models and active appearance models. Both methods rely on a training set of images on which important landmarks are tagged. For active shape models, a mean shape is built and a principal

component analysis gives the preferred directions of deformation. For active appearance models, the gray-level variation is also taken into account.

In the following sections, we give more details on the segmentation methods outlined above.

4.1. SNAKES

We begin by describing the active contour model introduced by Kass, Witkin and Terzopoulos [51]. The method is variational and edge based. It is the first appearance of active contour methods in image segmentation. The set of admissible edges is the set V of closed continuous curves in Ω, often called snakes.

$$V = \{v : \bigsqcup_{i=1}^{k} S^1 \longrightarrow \Omega\},$$

where k denotes the number of connected components in the curve. To know which curve in V approximates best the boundaries in the image, a curve energy is defined. The energy of $v \in V$ is defined as

$$E_{\text{snake}}(v) = E_{\text{int}}(v) + E_{\text{ext}}(v). \tag{4.1}$$

$E_{\text{int}}(v)$ is the internal energy, which decreases as v gets more regular. Typically

$$E_{\text{int}}(v) = \int_{S^1} \left|\frac{dv}{ds}\right|^2 ds + \alpha \int_{S^1} \left|\frac{d^2v}{ds^2}\right|^2 ds.$$

The first term is an elasticity term, that penalizes long curves. The second is a rigidity term, that penalizes high curvature regions.

The external energy $E_{\text{ext}}(v)$ is related to the underlying image. It has to be small if v is on the edges of g and large otherwise. There are many possibilities for $E_{\text{ext}}(v)$, but typically edges are characterized by a sudden change of color in the image. Hence pixels where the image gradient is large are likely to be close to relevant edges. A common choice for $E_{\text{ext}}(v)$ is

$$E_{\text{ext}}(v) = -\beta \int_{S^1} |\nabla g(v(s))|^2 ds,$$

but it could be $-\int_{S^1} h(|\nabla g(v(s))|) ds$ for any increasing positive function h.

Remark 69. *The energy does have minimizers, but they need not be unique [51].*

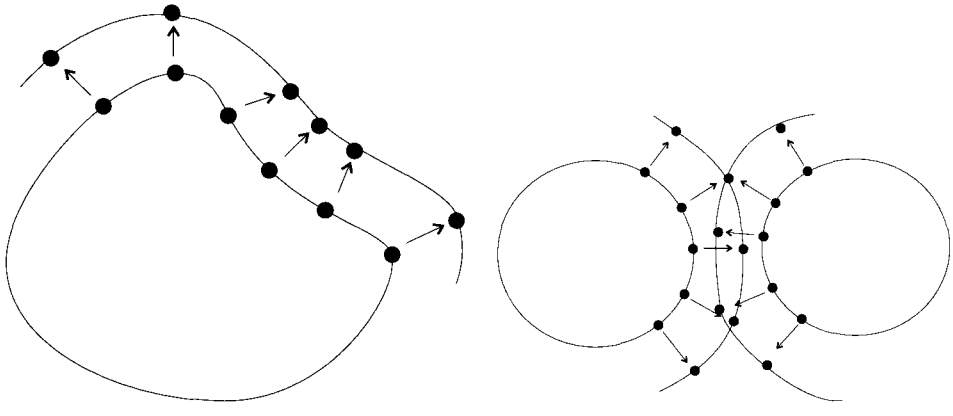
It is then possible to compute the Euler-Lagrange equation of this minimization problem. The problem is usually solved by a gradient descent method, starting from an initial

curve v_0. In general, if the initial curve is close to the object contour that is sought, this approach will give interesting results, since the initial curve is close to the global minimum of the functional.

On the other end, when the initial curve is relatively far, many difficulties may occur. First, the curve can get stuck in a local minimum. Another problem is that we may want the curve to change topology or avoid overcrossings. These operations are hard to do with a parametrized curve. The algorithm to evolve a parametrized curve usually goes as follows

1. Discretize the curve;
2. Advance the front by an explicit or implicit scheme;
3. Re-parametrize the curve.

The re-parametrization is very important since the curve may be stretched or compressed at some places. Figure 4.3 shows how this can be complicated when there is a possibility for a change of topology.



Advance and re-parametrize the curve Reconnecting problems

Figure 4.3: Active contours with a parametrized curve

4.2. LEVEL SET METHOD

The **Level Set method** was introduced in the late 80's by Sethian and Osher [77]. This method gives tools to evolve a curve and manage topology changes, which was hard with the active contour approach of Kass, Witkin and Terzopoulos [51]. The idea is simple: instead of describing the curve via an explicit parametrization $C = \{v(s), s \in S^1\}$, the curve

is described implicitly via a function

$$\phi_0 : \Omega \longrightarrow \mathbb{R},$$

such that

$$C = \{x : \phi_0(x) = 0\}.$$

If the function ϕ_0 changes, then so does its 0-level set. Hence the curve evolution will be describe as a continuous deformation of ϕ_0:

$$\begin{cases} \phi : \Omega \times \mathbb{R}^+ \longrightarrow \mathbb{R}, \\ \phi(x, 0) = \phi_0(x) \end{cases}$$

This can be thought of as a time evolution, such that the curve C_t at time t is simply

$$C_t = \{x : \phi(x, t) = 0\}.$$

Topology changes do not require special care using level set functions. They occur naturally as nothing special happens to the level set function when the topology of its level sets changes.

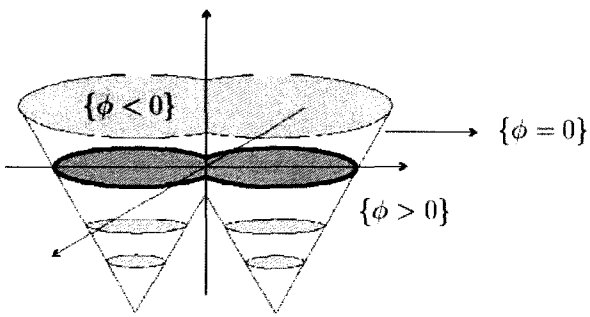


Figure 4.4: The level set method can handle topology changes of the curve. Different slices, that correspond to different times in the curve evolution process are shown.

Example 70. *The equation*

$$\textbf{(P5)} \begin{cases} \phi_t = -V_n |\nabla \phi| \\ \phi(\cdot, 0) = \phi_0 \end{cases}$$

evolves the curve initially described by ϕ_0 in the normal direction at speed V_n. Indeed, if

$v(t)$ is a parametrization of the curve at time t, then

$$0 = \frac{d}{dt}\phi(v(t),t) = \phi_t + \nabla\phi(v(t),t)\cdot v'(t).$$

But as noted before, the normal to the curve is $n = \frac{\nabla\phi}{|\nabla\phi|}$ *and since the normal speed is* $v'(t)\cdot n = V_n$*, we get*

$$0 = \phi_t + \nabla\phi(v(t),t)\cdot v'(t) = \phi_t + |\nabla\phi| n\cdot v'(t) = \phi_t + V_n|\nabla\phi|.$$

The normal speed V_n can depend on various things such as

1. *Position in space;*
2. *External parameters (e.g. medical image g, interface pressure);*
3. *Curvature of the curve.*

There are now many books available on this method, since it revealed itself as a very efficient and robust way to model the evolution of curves and fronts. Two main references are the book by Sethian [102] and the one by Osher [76].

4.3. GEOMETRIC ACTIVE CONTOURS

4.3.1. Level set formulation of classical snakes

The original active contour method of Kass, Witkin and Terzopoulos [51] can be described in a level set framework. For a given level set function

$$\phi : \Omega \longrightarrow \mathbb{R},$$

the internal energy is rewriten as

$$E_{\text{int}}(v) = \mathscr{H}^{N-1}(\{\phi = 0\}) + \alpha \int_\Omega \delta(\phi)\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)dx. \tag{4.2}$$

The first term in the energy computes the length and the second integrates the mean curvature of the curve (the 0 level curve of ϕ). Remark that the integrand of the curve length integral is not an integrable function. It is rather a function of bounded variation. The integral is defined in that respect. Now remark that if $H = \chi_{[0,\infty)}$ denote the Heaviside

function

$$\mathscr{H}^{N-1}(\{\phi = 0\}) = |DH(\phi)|(\Omega) = \int_\Omega d|DH(\phi)| = \int_\Omega \delta(\phi)|\nabla\phi|dx. \tag{4.3}$$

Hence

$$E_{\text{int}}(v) = \int_\Omega \delta(\phi)|\nabla\phi|dx + \alpha \int_\Omega \delta(\phi)\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)dx \tag{4.4}$$

is the level set formulation of the internal energy. As it will be noted in section 4.4, the term associated to $\int_\Omega \delta(\phi)|\nabla\phi|dx$ in the Euler-Lagrange equation has the form

$$\delta(\phi)\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right), \tag{4.5}$$

which accounts for the rigidity term. Hence minimizing length gives rise to a curvature term in the curve evolution process. This suggests some dependence between the two terms in the internal energy. In most cases, only the first term is considered. The second is an elastic energy, that gives rise to a 4^{th} order PDE.

4.3.2. Non variational active contours

Level set methods have first been applied to active contours in the works of Caselles et al. [15] and of Malladi, Sethian and Vermuri [61]. They proposed edge based algorithms, that are non variational.

They rely on an edge detector (edge stopping function), which is a function $h : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ such that

1. h is monotonically decreasing,
2. $h(0) = 1$,
3. $h(x) \overset{x\to\infty}{\longrightarrow} 0$.

Hence, $h(|\nabla g|)$ is a function that has value 1 in flat regions of the image and gets close to 0 close to edges. A typical example of an edge detector is

$$h(x) = \frac{1}{1 + cx^2}. \tag{4.6}$$

A clear edge in an image is a very fine feature. Around an edge, only few pixels will have high gradient magnitude $|\nabla g|$. In order for the edge to have a wider influence, Caselles *et al.* [15] proposed to use $G_\sigma * g$ instead of g, where G_σ is the Gaussian kernel of variance σ^2. $G_\sigma * g$ is a blurred version of the image g. Figure 4.5 shows the difference between $h(|\nabla g|)$

and $h(|\nabla(G_\sigma * g)|)$. Also the influence from the noise is less important using the blurred image instead of the real image. In practice $G_\sigma * g$ is generally computed by applying the heat equation (3.1) for some time steps.

Now, consider the problem

$$\begin{cases} \phi_t = |\nabla\phi| h(|\nabla(G_\sigma * g|), \\ \phi(\cdot, 0) = \phi_0. \end{cases} \tag{4.7}$$

This problem evolves the initial curve $C_0 = \{x : \phi_0(x) = 0\}$, that moves in the normal direction with constant speed in objects with uniform colors, and stops when it reaches regions of high gradient of g (boundary points). The model can be refined by adding a curvature term

$$\begin{cases} \phi_t = |\nabla\phi|(h(|\nabla(G_\sigma * g|) + \epsilon\kappa), \\ \phi(\cdot, 0) = \phi_0. \end{cases} \tag{4.8}$$

This prevents the curve from developing sharp corners. The process is local: the evolving may get stuck if it is blocked by image features. In such a case, the contour may not detect features over these boundaries.

4.3.3. Geodesic active contours

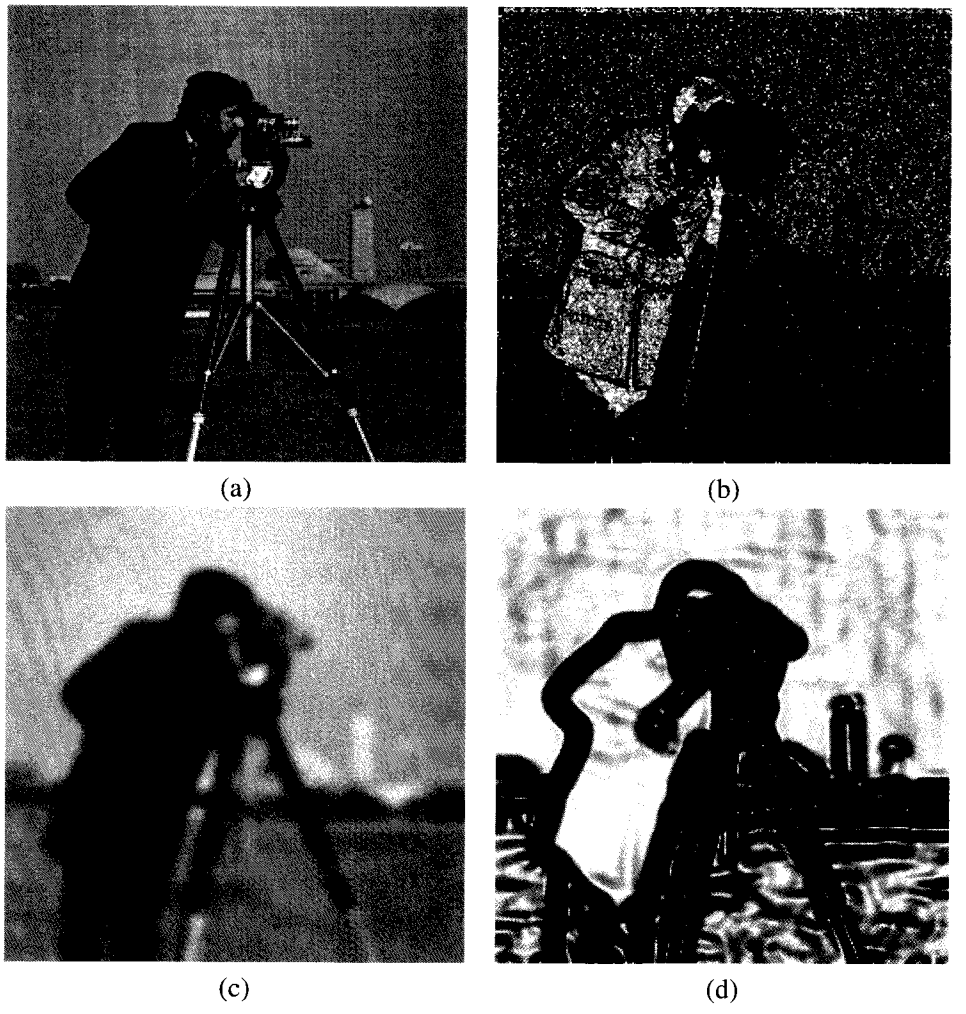
Geodesic active contours have been first introduced by Kichenassamy *et al.* [52] and by Caselles, Kimmel and Sapiro [16]. Their idea was to set up a variational active contour model whose minimizers are geodesics for a particular metric. This metric is given by the stopping function $\bar{h} = h(|\nabla(G_\sigma * g)|)$ in the following sense: for a given curve $c : [0, 1] \to \Omega$, its length is given by

$$\int_0^1 \bar{h}(c(t))|c'(t)|\, dt. \tag{4.9}$$

For a given curve c, they define the curve energy to be

$$E(c) = \alpha \int_0^1 |c'(t)|^2\, dt + \lambda \int_0^1 \bar{h}(c(t))^2\, dt. \tag{4.10}$$

They showed in [16] that minimizers of $E = E(c)$ are geodesics under the metric given by equation (4.9).



(a) (b)

(c) (d)

Figure 4.5: Different stopping images, using blurred and non blurred image. When using the blurred image, the edges have an influence on a wider band, and there is also less contamination by noise and texture. (a) The original image. (b) The stopping image $\frac{1}{1+|\nabla g|^2}$. (c) A blurred version of the image. (d) The stopping image $\frac{1}{1+|\nabla(G_\sigma * g)|^2}$.

In a level set framework, local minimizers are level sets of steady states of the equation

$$\begin{cases} \phi_t = |\nabla\phi| \mathrm{div}\left(\bar{h}\frac{\nabla\phi}{|\nabla\phi|}\right) \\ \quad = \bar{h}|\nabla\phi| \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \nabla\bar{h}\cdot\nabla\phi, & \text{on } \Omega, \\ \frac{\partial\phi}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{4.11}$$

This equation is very similar to equation (4.8). The extra advection term $\nabla\bar{h}\cdot\nabla\phi$ attracts the level sets of ϕ towards the edges, since the vector field $-\nabla\bar{h}$ points towards minima of $\bar{h}$, which correspond to edges. This new feature makes the active contour method more robust. It has been used successfully in many applications, for example [16, 58, 78, 122].

The main difficulty when using this algorithm is that the energy has many local minima. The noise will also affect the number of local minima. If the image is very noisy, it is very likely that the curve will get stuck into a local minimum. In that case, the initial curve needs to be close to the global minimizer.

The advection term in equation (4.11) is discretized using a standard upwind scheme. The diffusion term can be discretized with a simple scheme like the one given in equation (3.11). There is no need to use a more complex scheme, since the time step restriction imposed by the advection term will in general ensure the stability of the simple scheme of equation (3.11). Mikula *et al.* [68, 69] proposed a semi-implicit finite volume scheme that is unconditionally stable and that allows to iterate with large time steps. They also proposed a parallel implementation of that scheme [67].

4.3.4. Subjective surfaces

In a typical level set application, it is the 0 level set of the level set function ϕ that is tracked in time. Sarti, Malladi and Sethian [98, 99] pointed out that the level set equation (4.11) has no preference towards the 0 level set of the function ϕ. In fact, all of its level sets are attracted towards edges of the image g. In this process, steep cliffs are created near edges. Away from edges, the level set function becomes flat. The steady state should then be a piecewise smooth function whose discontinuities lye on edges. This method is called the *subjective surface method*, since it looks at all level curves instead of the sole 0 level set.

They took advantage of this fact to help choosing a good initial condition to equation (4.11). A point x_0 is chosen inside the object of interest, and the distance function $D(\cdot, x_0)$ to this point is computed. A good initial condition is then $\phi_0 = \frac{1}{1+D}$. It has high values just in a neighborhood of the given point x_0 and rapidly goes to 0 away from x_0.

Equation (4.11) is solved with a Dirichlet boundary condition instead of a Neumann boundary condition. Only features that are relatively close to the chosen point x_0 will be captured by the evolving equation.

Figure 4.6 shows the evolution of the subjective surface in the case of a 100×100 image featuring a broken circle. A time step $dt = 0.001$ has been used and 40 000 iterations have been performed. The middle point has been chosen as initial condition. The method has the ability to close missing boundaries.

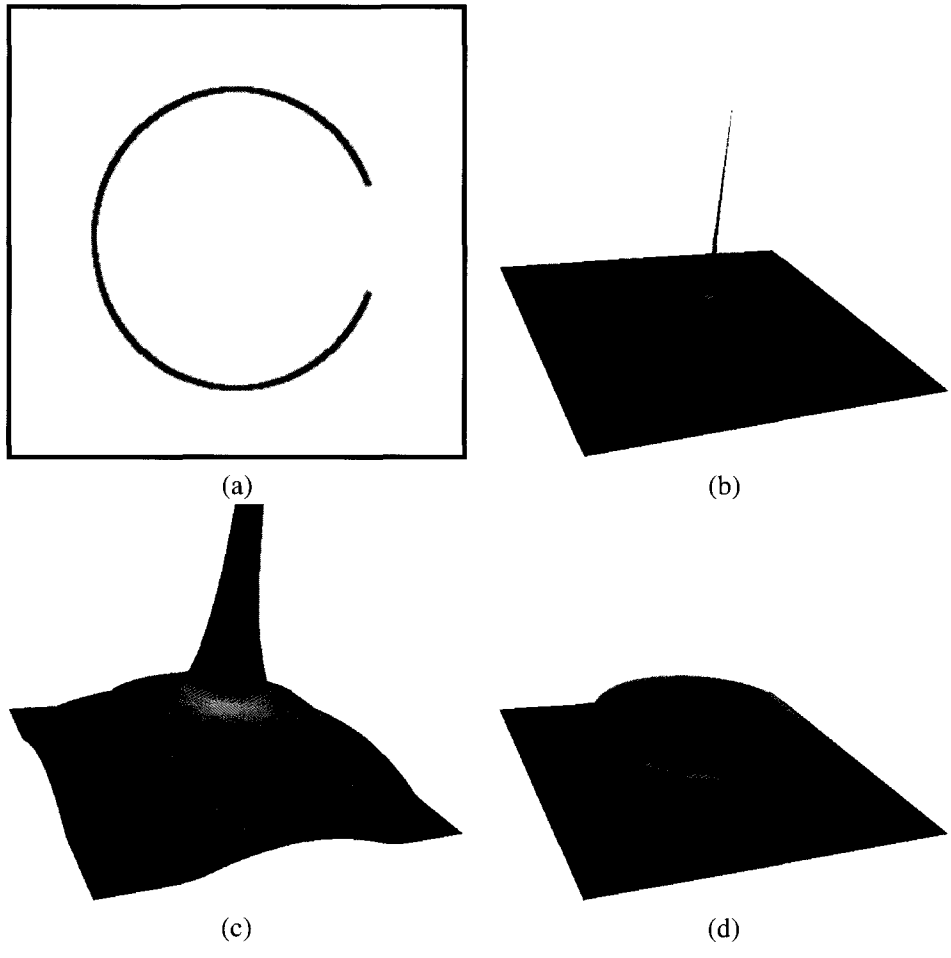


Figure 4.6: The subjective surface method. (a) The synthetic image of a broken circle. (b) The initial condition $\phi_0 = \dfrac{1}{1 + D(\cdot, x_0)}$, where x_0 is point chosen inside the circle. (c) The level set function after 20 000 iterations (d) The steady state.

4.4. ACTIVE CONTOUR WITHOUT EDGES

4.4.1. The original model

The active contour without edges of Chan and Vese [21] is a variational active contour method based on the Mumford-Shah functional

$$J(K,u) = \int_{\Omega\setminus K} |\nabla u|^2 + \lambda \int_{\Omega} (u-g)^2 + \mu \mathscr{H}^{N-1}(K \cap \Omega) \tag{4.12}$$

described in section 2.7. There has been several approaches in order to solve this minimization problem. One consists in approximating the Mumford-Shah functional by a sequence of elliptic functionals [6, 7, 13]. These can be solved more easily with a sequence of compact sets K_n that will Gamma-converge to the set K minimizing the Mumford-Shah functional.

Another approach is to minimize the Mumford-Shah over a restricted domain. Chan and Vese [21], as well as Tsai, Yezzi and Willsky [113] independently proposed ways of solving the minimization problem over the set of piecewise smooth functions. The problem becomes much simpler as the discontinuity set J_u can be described by a level set function ϕ. Hence the minimum is sought among functions of the form $u = u_1 H(\phi) + u_2(1 - H(\phi))$, where u_1 and u_2 are smooth in the regions $\{\phi \geq 0\}$ and $\{\phi < 0\}$ respectively. The Mumford-Shah functional becomes

$$F(\phi,u_1,u_2) = \mu \mathscr{H}^{N-1}(\{\phi = 0\}) + \int_{\Omega} |\nabla u|^2 dx + \lambda \int_{\Omega} (g-u_1)^2 H(\phi) dx + \lambda \int_{\Omega} (g-u_2)^2 (1-H(\phi)) dx, \tag{4.13}$$

where $H = \chi_{[0,\infty)}$ stands for the Heaviside function. Also, remark that

$$DH(\phi) = \delta(\phi)|\nabla\phi|, \tag{4.14}$$

so that

$$\mathscr{H}^{N-1}(J_u) = |DH(\phi)|(\Omega) = \int_{\Omega} \delta(\phi)|\nabla\phi| dx = \int_{\Omega} |\nabla\phi| d\delta(\phi). \tag{4.15}$$

Then

$$F(\phi,u_1,u_2) = \mu \int_{\Omega} \delta(\phi)|\nabla\phi| dx + \int_{\Omega} |\nabla u|^2 + \lambda \int_{\Omega} (g-u_1)^2 H(\phi) dx + \lambda \int_{\Omega} (g-u_2)^2 (1-H(\phi)) dx. \tag{4.16}$$

It is now possible to compute the Euler-Lagrange equation associated to F. Looking at the variation with respect to u_1 leads to the Sobolev smoothing problem of section 3.3. That is, u_1 is a solution of

$$\begin{cases} \Delta u_1 = u_1 - g & \text{on } \Omega_1 = \{\phi \geq 0\}, \\ \frac{\partial u_1}{\partial n} = 0, & \text{on } \partial\Omega_1. \end{cases} \tag{4.17}$$

Similarly, u_2 is a solution of

$$\begin{cases} \Delta u_2 = u_2 - g & \text{on } \Omega_2 = \{\phi < 0\}, \\ \frac{\partial u_2}{\partial n} = 0, & \text{on } \partial\Omega_2. \end{cases} \tag{4.18}$$

The directional derivative of the length term at ϕ in the direction $\psi \in \mathscr{D}(\Omega)$ can be computed as follows

$$\begin{aligned} &\frac{d}{dt}\int_\Omega \delta(\phi + t\psi)|\nabla\phi + t\nabla\psi|\,dx\Big|_{t=0} \\ &= \int_\Omega \delta'(\phi + t\psi)\psi|\nabla\phi + t\nabla\psi| + \delta(\phi + t\psi)\frac{\nabla\phi + t\nabla\psi}{|\nabla\phi + t\nabla\psi|}\cdot\nabla\psi\,dx\Big|_{t=0} \\ &= \int_\Omega \delta'(\phi)\psi|\nabla\phi| + \delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi\,dx. \end{aligned} \tag{4.19}$$

Remark that

$$\operatorname{div}\left(\delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\psi\right) = \delta'(\phi)|\nabla\phi|\psi + \delta(\phi)\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\psi + \delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi \tag{4.20}$$

in the sense of distributions, so that

$$\begin{aligned} &\int_\Omega \delta'(\phi)\psi|\nabla\phi| + \delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi\,dx \\ &\quad = \int_\Omega \operatorname{div}\left(\psi\delta(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) - \int_\Omega \delta(\phi)\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\psi. \end{aligned} \tag{4.21}$$

Moreover

$$\int_\Omega \operatorname{div}\left(\delta(\phi)\psi\frac{\nabla\phi}{|\nabla\phi|}\right) = \int_{\partial\Omega} \delta(\phi)\psi\frac{\nabla\phi}{|\nabla\phi|}\cdot n d\mathscr{H}^{N-1} = 0 \tag{4.22}$$

if we impose $\frac{\partial\phi}{\partial n} = \nabla\phi\cdot n = 0$, which seems a reasonable boundary condition, since this forces the level curves to be transverse to the boundary $\partial\Omega$. Therefore

$$\frac{d}{dt}\int_\Omega \delta(\phi + t\psi)|\nabla\phi + t\nabla\psi|\,dx\Big|_{t=0} = -\int_\Omega \delta(\phi)\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\psi. \tag{4.23}$$

For the fidelity term, we have

$$\begin{aligned}\frac{d}{dt}\int_\Omega (g-u_1)^2 H(\phi+t\psi)dx + \int_\Omega (g-u_2)^2(1-H(\phi+t\psi))\,dx\Big|_{t=0}\\ = \int_\Omega (g-u_1)^2\delta(\phi)\psi - (g-u_2)^2\delta(\phi)\psi dx\\ = \int_\Omega \delta(\phi)\psi[(g-u_1)^2-(g-u_2)^2]dx. \end{aligned} \tag{4.24}$$

Therefore

$$\begin{aligned}\left(\frac{\partial F}{\partial \phi},\psi\right) &= -\left(\delta(\phi)\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right),\psi\right) + \left(\delta(\phi)[(g-u_1)^2-(g-u_2)^2],\psi\right)\\ &= \left(\delta(\phi)\left[-\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)+(g-u_1)^2-(g-u_2)^2\right],\psi\right)\\ &= 0, \quad \forall\psi\in\mathscr{D}(\Omega), \end{aligned} \tag{4.25}$$

which means that

$$\delta(\phi)\left[-\mu\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)+\lambda(g-u_1)^2-\lambda(g-u_2)^2\right]=0 \tag{4.26}$$

in the sense of distributions. In gradient descent form, this gives the initial value problem

$$\begin{cases}\phi_t = \delta(\phi)\left[\mu\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)-\lambda(g-u_1)^2+\lambda(g-u_2)^2\right], & \text{on } \Omega\times(0,T),\\ \frac{\partial\phi}{\partial n}=0 & \text{on } \partial\Omega,\\ \phi(\cdot,0)=\phi_0 & \text{on } \Omega. \end{cases} \tag{4.27}$$

Problem 4.27 is very close to the gradient descent Problem 3.9 for the total variation algorithm of Osher, Rudin and Fatemi [92]. The main difference is that with (4.27) we let evolve a curve while with Osher algorithm it is the initial image g that is deformed. Also, in the last equation, there is a coefficient $\delta(\phi)$ so that only the 0 level set of ϕ moves. In practice, a regularization δ_ϵ of δ is used. The function

$$\delta_\epsilon(x) = \frac{1}{\pi}\frac{\epsilon}{\epsilon^2+x^2} \tag{4.28}$$

is a good choice in many cases.

The algorithm for computing the solution then goes as follows: At each time step,

1. Compute u_1 and u_2 from ϕ^n, solving Problems 4.17 and 4.18.

2. Compute ϕ^{n+1} from ϕ^n, u_1 and u_2, solving Problem 4.27.

Problem 4.27 can be solved using similar discretizations as the ones used for the total variation problem.

In this version of the Mumford-Shah problem, Equations 4.17 and 4.18 need to be solved at each time step. This is still computationally demanding. Chan and Vese also propose a simpler version of the Mumford-Shah problem [21], where the Mumford-Shah functional is minimized over the set of binary functions

$$\{u \in SBV(\Omega) : u \in \{c_1, c_2\}, c_1, c_2 \in \mathbb{R}\}. \tag{4.29}$$

The condition $u \in SBV(\Omega)$ forces the discontinuity set J_u to be of finite perimeter. As in the piecewise smooth case (equation (4.16)), the function u can be written as $u = c_1 H(\phi) + c_2(1 - H(\phi))$ and the Mumford-Shah functional becomes

$$E_{CV}(\phi, c_1, c_2) = \mu \int_\Omega \delta(\phi)|\nabla\phi| dx + \lambda \int_\Omega (g - c_1)^2 H(\phi) dx + \lambda \int_\Omega (g - c_2)^2 (1 - H(\phi)) dx. \tag{4.30}$$

The Euler-Lagrange equations lead to the following gradient descent problem:

$$\begin{cases} \phi_t = \mu\delta(\phi)\left[\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda(g - c_1)^2 + \lambda(g - c_2)^2\right], & \Omega \\ \frac{\partial\phi}{\partial n} = 0, & \partial\Omega \\ \phi(\cdot, 0) = \phi_0, \end{cases} \tag{4.31}$$

that is coupled with the conditions

$$c_1 = \frac{\int_\Omega g H(\phi) dx}{\int_\Omega H(\phi) dx} \quad \text{and} \quad c_2 = \frac{\int_\Omega g(1 - H(\phi)) dx}{\int_\Omega 1 - H(\phi) dx}. \tag{4.32}$$

The algorithm for computing the solution then goes as follows: At each time step,

1. Compute c_1 and c_2 from ϕ^n, using equations (4.32).

2. Compute ϕ^{n+1} from ϕ^n, c_1 and c_2 solving Problem 4.31.

4.4.2. Multiphase active contours without edges

The active contour without edges algorithm separates the given image g in two distinct parts. In many applications, the images to be segmented are complex and contain much more than two distinct parts or phases. In order to segment complex images, Vese and Chan [117]

proposed a multiphase framework to their algorithm. The idea is to have several level set functions at the same time. If two level set functions ϕ_1 and ϕ_2 are used, the domain is split into four different regions, namely

$$\begin{aligned}
\Omega_{11} &= \{x : \phi_1(x) \geq 0\} \cap \{x : \phi_2(x) \geq 0\} \\
\Omega_{12} &= \{x : \phi_1(x) \geq 0\} \cap \{x : \phi_2(x) < 0\} \\
\Omega_{21} &= \{x : \phi_1(x) < 0\} \cap \{x : \phi_2(x) \geq 0\} \\
\Omega_{22} &= \{x : \phi_1(x) < 0\} \cap \{x : \phi_2(x) < 0\}.
\end{aligned} \tag{4.33}$$

This situation is illustrated in Figure 4.7.

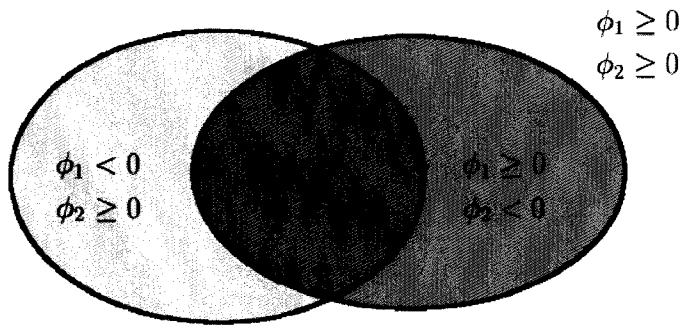


Figure 4.7: Two level set functions ϕ_1 and ϕ_2 split the domain Ω into four distinct phases.

With the help of these level set functions, the Mumford-Shah energy becomes

$$\begin{aligned}
E_{CV} =& \mu \int_\Omega |\nabla H(\phi_1)| + \mu \int_\Omega |\nabla H(\phi_2)| \\
&+ \lambda \int_\Omega (g - c_{11})^2 H(\phi_1) H(\phi_2)\, dx + \lambda \int_\Omega (g - c_{12})^2 H(\phi_1)(1 - H(\phi_2))\, dx \\
&+ \lambda \int_\Omega (g - c_{21})^2 (1 - H(\phi_1)) H(\phi_2)\, dx + \lambda \int_\Omega (g - c_{22})^2 (1 - H(\phi_1))(1 - H(\phi_2))\, dx.
\end{aligned} \tag{4.34}$$

The Euler-Lagrange equation and the gradient descent equation can be derived as in the two phase case. One clearly obtains that c_{ij} is the mean value of g in the region Ω_{ij}. Also, looking at the variations with respect to ϕ_1 and ϕ_2 yields to the following system of

equations,

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)\Big\{\mu \mathrm{div}\left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - \lambda\left[(g - c_{11})^2 - (g - c_{21})^2\right] H(\phi_2) - \lambda\left[(g - c_{12})^2 - (g - c_{22})^2\right](1 - H(\phi_2))\Big\}, \tag{4.35}$$

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)\Big\{\mu \mathrm{div}\left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - \lambda\left[(g - c_{11})^2 - (g - c_{12})^2\right] H(\phi_1) - \lambda\left[(g - c_{21})^2 - (g - c_{22})^2\right](1 - H(\phi_1))\Big\}. \tag{4.36}$$

These equations can be solved in the same way as the far simpler two phase case. However, a problem is that the two equations are coupled. As noted in [39], the algorithm often falls into local minimums, yielding the complicated problem of well seeding the algorithm.

Chung and Vese [23] proposed instead to use only one level set function, but to look at several of its level sets. The number of phases must be given in advance. In the three-phase case, a level set function ϕ splits the domain into the three regions

$$\begin{aligned} \Omega_1 &= \{\phi \geq 1\}, \\ \Omega_2 &= \{\phi \geq 0\} \cap \{\phi < 1\}, \\ \Omega_3 &= \{\phi < 0\}. \end{aligned}$$

Again, the Mumford-Shah energy can be written in terms of this level set function,

$$\begin{aligned} E_{CV} =& \mu \int_\Omega |\nabla H(\phi)| + \mu \int_\Omega |\nabla H(1 - \phi)| \\ &+ \lambda \int_{\Omega_1} (g - c_1)^2 \, dx + \lambda \int_{\Omega_2} (g - c_2)^2 \, dx + \lambda \int_{\Omega_3} (g - c_3)^2 \, dx. \end{aligned} \tag{4.37}$$

This method is very efficient on images where phases are contained one in another, like brain images. However there could be no intersections between phases as all level sets are strictly disjoint from each other for any continuous function ϕ.

Another approach is the hierarchical method proposed by Tsai, Yezzi and Willsky [113], and further investigated by Gao and Bui [39]. The idea is simple. It consists of first segmenting the image with the original two-phase Chan-Vese model. Then both regions can be considered as new domains, on which the two-phase Chan-Vese model can be applied again. This approach has the advantage that it is not necessary to solve equations (4.35) and

(4.36) simultaneously. Instead, three different two-phase problems need to be solved. As noted in in [39] this method solves the problem that the multiphase algorithm often falls into local minima.

4.4.3. Variations on the Chan-Vese model

As for the total variation problem described in section 3.4, the numerical computations could benefit from replacing $\delta(\phi)$ by $|\nabla\phi|$. This was suggested by Marquina and Osher [62]. Here, there is no staircasing effect, however, the numerical schemes have better stability. The gradient descent equation becomes

$$\phi_t = |\nabla\phi|\left[\mu \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda(g-c_1)^2 + \lambda(g-c_2)^2\right]. \tag{4.38}$$

It is interesting to point out that Problem (4.31) admits steady states, but equation (4.38) no longer has steady states. On the contrary, for any initial condition ϕ_0, the function ϕ will tend to $+\infty$ where it is positive, and to $-\infty$ where it is negative. However, the 0 level set of ϕ converges to the exact same contour as the steady state of Problem (4.31).

In [106], Song and Chan decoupled equation (4.27) in order to simplify the resolution. They first smooth the image g to obtain a new image g^*, and then solve the simplified equation

$$\phi_t = (g^* - c_1)^2 - (g^* - c_2)^2. \tag{4.39}$$

They proposed a fast algorithm for solving equation (4.39). In fact, their algorithm corresponds closely to clustering the set $\{g^*(x) : x \in \Omega\}$ of values of g^* into two clusters using a k-means procedure [45]. Osher and He introduced a similar algorithm to solve the multiphase problem [46]. Again, this is closely related to applying a k-means procedure to the set $\{g^*(x) : x \in \Omega\}$ with the right number of clusters k. Independently, Gibou and Fedkiw took advantage of this analogy to propose a hybrid algorithm that alternates between smoothing and k-means [42]. However, it is not known if any of these fast algorithms lead to minima of the Chan-Vese energy for some value of the parameters μ and λ.

Chan, Esedoglu and Nikolova [19] also proposed a way of finding global minimizers for the Chan-Vese model. They proposed to minimize the following functional

$$E_{CVGM} = \mu \int_\Omega |\nabla H(\phi)|\, dx + \lambda \int_\Omega \phi\,(|c_1 - g|H(\phi) + |c_2 - g|(1 - H(\phi)))\, dx, \tag{4.40}$$

under the constraint $0 \leq \phi \leq 1$. This constraint makes the problem non convex. However,

they devised an exact penalty term $v(s)$, where $s = (c_1 - g)^2 + (c_2 - g)^2$ to be added to this functional and that removes the constraint. The gradient descent equation is then

$$\phi_t = \mu \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \lambda\left[-(c_1 - g)^2 + (c_2 - g)^2\right] - \alpha v'(s). \tag{4.41}$$

In theory, the steady state is a function ϕ^* that has values in $[0, 1]$. In practice, they show that the steady state is almost a binary function. For almost any value $c \in (0, 1)$, the threshold function $\phi_c^* = H(\phi - c)$ is a minimizer of the original Chan-Vese energy.

Burger and Hintemüller [14] proved that the functional in equation (4.40) can be minimized using the following gradient descent equation

$$\phi_t = \mu \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \lambda\left[-(c_1 - g)^2 + (c_2 - g)^2\right]. \tag{4.42}$$

It is the same equation as the original equation for the Chan-Vese model except that the factor $\delta(\phi)$ is missing. Nevertheless the function ϕ must be subject to the condition $\phi \in [-1, 1]$ which can be enforced easily at each time step.

In section 3.4, it has been mentioned that the total variation algorithm benefits from using an L^1 fidelity term instead of an L^2 fidelity term. It is natural to ask the same question for the Chan-Vese model. To our knowledge, Kimmel [53] first introduced the L^1 fidelity for this problem. The problem is to minimize

$$E_{CVL1} = \int_\Omega \mu|\nabla H(\phi)|\,dx + \lambda\int_\Omega |c_1 - g|H(\phi)\,dx + \lambda\int_\Omega |c_2 - g|(1 - H(\phi))\,dx. \tag{4.43}$$

A continuous change of contrast is any continuous non decreasing function $f : \mathbb{R} \to \mathbb{R}$. A minimization problem $\min_u E(u, g)$ will be said contrast invariant if

$$\arg\min_u E(u, f \circ g) = f \circ \arg\min_u E(u, g). \tag{4.44}$$

Note that this contrast invariance property is stronger than the scale invariance property obtained for the Problem (3.17) in [18]. Darbon [31] proved that unlike the L^2 problem, Problem (4.43) is contrast invariant. In order to solve this problem via a PDE formulation, it can be formulated in terms of a level set function ϕ as follows:

$$\min_{c_1, c_2, \phi} E_{CV}(\phi, c_1, c_2), \tag{4.45}$$

where

$$E(\phi, c_1, c_2) = \mu \int_\Omega \delta(\phi)|\nabla\phi|\,dx + \lambda \int_\Omega |c_1 - g|H(\phi)\,dx + \lambda \int_\Omega |c_2 - g|(1 - H(\phi))\,dx. \quad (4.46)$$

The gradient descent equation for ϕ is obtained in the same way as for the L^2 case:

$$\phi_t = \delta(\phi)\left[\mu \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda|c_1 - g| + \lambda|c_2 - g|\right]. \quad (4.47)$$

Note that outliers here have a weakened influence since for such points, $(c_1 - g)^2$ will be much bigger than $|c_1 - g|$. But the main difference is in the calculation of the values c_1 and c_2. If ϕ is fixed, we need to find c_1 which minimizes

$$\int_\Omega |c_1 - g|H(\phi)\,dx = \int_{\Omega_1=\{\phi\geq 0\}} |c_1 - g|\,dx.$$

We can use the following proposition. We did not find any proof of this proposition in the literature, so we present here a personal proof.

Proposition 71. *For $g : \Omega \to \mathbb{R}$, if c^* is a minimum of*

$$F(c) = \int_\Omega |g - c|\,dx,$$

then c^* is a median value for g in Ω.

Proof. It is not possible to directly compute the derivative with respect to c because the absolute value is not smooth at the origin. However, we can compute the variation of the energy F as c changes. For $\Delta c \geq 0$, let us compare $F(c + \Delta c)$ with $F(c)$. If μ denotes the usual Lebesgue measure on $\mathbb{R}^n$, we have

$$\int_{g\leq c} |g - (c + \Delta c)|\,dx = \int_{g\leq c} |g - c|\,dx + \Delta c\mu\{g \leq c\},$$
$$\int_{g\geq c+\Delta c} |g - (c + \Delta c)|\,dx = \int_{g\geq c+\Delta c} |g - c|\,dx - \Delta c\mu\{g \geq c + \Delta c\}.$$

Moreover, for the missing part, there is also the following obvious estimate

$$\left|\int_{c<g<c+\Delta c} |g - (c + \Delta c)|\,dx - \int_{c<g<c+\Delta c} |g - c|\,dx\right| \leq \Delta c\mu\{c < g < c + \Delta c\},$$

Hence, we get

$$\frac{\int_\Omega |g-(c+\Delta c)|\,dx - \int_\Omega |g-c|\,dx}{\Delta c} = \mu\{g \le c\} - \mu\{g \ge c+\Delta c\} + \int_{c<g<c+\Delta c} |g-(c+\Delta c)|\,dx - \int_{c<g<c+\Delta c} |g-c|\,dx.$$

Now, when $\Delta c \to 0$, we have $\mu\{g \ge c+\Delta c\} \to \mu\{g > c\}$ and

$$\lim_{\Delta c \to 0} \mu\{c < g < c+\Delta c\} = 0,$$

so that

$$\frac{F(c+\Delta c) - F(c)}{\Delta c} \to \mu\{g \le c\} - \mu\{g > c\}.$$

That is, F is Gâteaux-differentiable along $\Delta c > 0$. If c^* is a minimum of F, this derivative has to be positive, which leads to

$$\mu\{g \le c^*\} \ge \mu\{g > c^*\} = \mu(\Omega) - \mu\{g \le c^*\}.$$

Hence $\mu\{g \le c^*\} \ge \mu(\Omega)/2$. A similar calculation can be performed for $\Delta c \le 0$, which gives as left derivative $-\mu\{g < c\} + \mu\{g \ge c\}$. For a minimum c^*, we also get $\mu\{g \ge c^*\} \ge \mu(\Omega)/2$, which means that c^* is a median value for g in Ω. Moreover, the function F is differentiable in the usual sense at a value c if and only if $\mu\{g = c\} = 0$. □

This means that c_1 and c_2 are the median values of the observed image g in the regions $\{\phi \ge 0\}$ and $\{\phi < 0\}$ respectively. Recall that in the original problem, they were the mean values, which are again more sensitive to outliers than the median value. This is why the median values is desirable for noisy images. If segmentation is performed with L^2 fidelity, a smaller value of the parameter λ will be needed as if the segmentation was made using the L^1 fidelity. This leads to over-smoothing of the segmentation result.

There is also a main difference between the L^1 and the L^2 problems. The L^2-norm Chan-Vese problem tends to split up the image according to major color changes. Hence a relatively small region, whose color is far from the rest of the image could be segmented in a single step. On the other hand, the L^1-norm Chan-Vese problem will tend to split the image into regions of similar area. This version is particularly useful when objects with low contrast differences are to be detected. The choice of the norm should depend on the image to segment and the goal of the segmentation.

4.5. ACTIVE CONTOURS WITH SHAPE PRIORS

Active contours have proved be very efficient for image segmentation. However, in most real applications, image are very complex, and the objects to be segmented are melted in their environment. This makes them very difficult to recognize in the image. In those cases, the active contours will leak, or will fall into a local minimum. In such complex application, a common tactic is to incorporate shape prior information to the model.

The basic idea is to add a term in the evolution that comes from the minimization of a distance between the image to be segmented and a model of the object of interest. This leads to the question of choosing an adequate notion of distance between images. A simple choice is to take the L^2 distance between the segmented image u and a model image g_m:

$$E_{L^2} = \int_\Omega (u - g_m)^2 \, dx. \tag{4.48}$$

This supposes that the two images are aligned. Even then, it turns out not to be an optimal choice. Indeed it depends on the size of the image, which is not natural. There has been a lot of work on defining appropriate metrics on the space of curves and on how to align the model on the image. A non exhaustive part of this work can be found in [127, 125, 70, 22, 66, 124].

There has been many attempts at the use of active contours with shape prior information. Given a metric η on the space of curves, it is natural to add the energy

$$E_\eta = \eta(u, g_m),$$

which is the distance between the segmented image u and the model image g_m. If the active contour method is variational with energy E_{AC}, then one can consider the new variational problem that consists in minimizing

$$E_{\text{Prior}} = E_{AC} + E_\eta. \tag{4.49}$$

This variational problem will lead to a gradient descent equation as usual. This is the approach taken by Cremers and Soatto [30], Riklin-Raviv, Kiryati and Sochen [91] and Chan and Zhu [17]. In each of these cases, they consider the Chan-Vese model as the active contour energy, and they handled the metric part in different manners.

If the active contour problem is non variational one can just add the Euler Lagrange equation of the metric energy to the evolution equation for the active contour. This is the

approach used by Leventon, Grimson and Faugeras [58] and Tsai *et al.* [112].

4.6. ACTIVE SHAPE MODEL AND ACTIVE APPEARANCE MODEL

Active shape model (ASM) [27] and active appearance model (AAM) [25] are model based segmentation algorithm. The algorithm knows what is the standard shape of the object and how it may vary from one image to another.

4.6.1. Active shape models

The active shape model has been introduced by Cootes *et al.* [27]. In the algorithm, the shape of the object of interest must be learned prior to the segmentation process. This is done by looking at a database of a reasonable number of images (say N). On each image, the shape of the object is described by a number of tagged points on the object. The points are to be placed consistently on each image. Points must be placed at strategic locations, for example regions where the shape changes significantly. If each image is tagged with n points, then the general shape of the object in an image is described by a single point in $\mathbb{R}^{2n}$ for 2D images or in $\mathbb{R}^{3n}$ for 3D images.

The N images of the training sets are given by N points x_i in, say, $\mathbb{R}^{2n}$. From these points, a principal component analysis (PCA, see for example [34]) can be applied to get the main modes of variations of the points. It is done as follow. First the mean shape of the object is

$$\bar{x} = \frac{1}{N}\sum_i x_i. \tag{4.50}$$

Then the covariance matrix can be calculated as

$$S = \frac{1}{N}\sum_i (x_i - \bar{x})(x_i - \bar{x})^T. \tag{4.51}$$

The eigenvalues λ_k of S represents the variance of the data set in the direction of the associated eigenvector v_k. The idea of PCA is to approximate the data set variability only with few modes with the highest eigenvalues. Let the eigenvalues λ_k be ordered in decreasing order. Then, if

$$\lambda_T = \sum_{k=1}^{2n} \lambda_k \tag{4.52}$$

is the total variance of the data set, one could choose a number m such that

$$\frac{1}{\lambda_T}\sum_{k=1}^{m}\lambda_k > 0.95. \tag{4.53}$$

In that case, 95% of the variation of the data set are described by the eigenvectors $v_1, v_2, ..., v_m$. Hence, the shape variability of the object to be segmented may be described by only a few eigenvectors, depending on the application.

Given a new image, the segmentation problem consists then in looking for a shape x in the image that is of the form

$$x = \bar{x} + \sum_{k=1}^{m} w_k v_k. \tag{4.54}$$

Chebyshev's inequality suggests that the weights w_k can be chosen in the interval $[-3\sqrt{\lambda_k}, 3\sqrt{\lambda_k}]$. The set of admissible shape can be written as

$$\mathscr{S}_{\text{ASM}} = \left\{ x = \bar{x} + \sum_{k=1}^{m} w_k v_k,\ w_k \in \left[-3\sqrt{\lambda_k}, 3\sqrt{\lambda_k}\right] \right\}. \tag{4.55}$$

The active shape model finds the optimal weights by starting with the mean shape $y_0 = \bar{x}$ and alternating between the following two steps until convergence.

1. Find a better position y_1' for landmark points, that fits better the image characteristics.
2. Update landmarks positions to y_1 that is the closest point to y_1' in $\mathscr{S}_{ASM}$.

4.6.2. Active appearance model

The active appearance model, introduced by Cootes, Edwards and Taylor [25] is an enriched version of the active shape model where the gray level of the image is also taken into consideration. Using the training set as above, the shape vectors x_i and the mean shape vector $\bar{x}$ are formed as above.

In order to formulate statistics about gray level variability, the images of the training set must be standardized in some way. First the image g_i of the training set is warped to fit the mean shape via a transformation that sends its shape x_i to the mean shape $\bar{x}$. Then a mean intensity image $\bar{g}$ can be computed from all the warped images. Each image has to be rescaled to be consistent with the mean image $\bar{g}$. Then a PCA can be performed on the set of normalized images to get eigenvectors u_k corresponding to the m highest intensity modes.

Given a new image h, a segmented image is sought in the set of admissible images $\mathscr{S}_{\text{AAM}}$. An admissible image must be of the form

$$\begin{bmatrix} g \\ x \end{bmatrix} = \begin{bmatrix} \bar{g} \\ \bar{x} \end{bmatrix} + Pb, \tag{4.56}$$

where P is the matrix containing all the eigenvectors v_k and u_k as columns and b is the vector of weights. Each weight b_k is sought in an interval $[-3\sqrt{\lambda_k}, 3\sqrt{\lambda_k}]$.

The segmented image g will be the one that minimizes the L^2 distance between h and $\mathscr{S}_{\text{AAM}}$. A gradient descent method can be applied on the set of weights b to find g. However, it is computationally expensive to do this directly. Some clever iterative techniques have been proposed by Baker and Matthews [12] and Cootes and Taylor [26].

5

Analysis of the Chan-Vese model

In this chapter, we analyze the Chan-Vese model under various factors. These are

1. Influence of the discretization;
2. Influence of the initial condition;
3. Performance of the hierarchical segmentation;
4. Introduction of the hierarchical segmentation with L^1 fidelity.

To our knowledge, the analysis and the remarks made in this chapter are original, except when sources are explicitly cited.

5.1. DISCRETIZATIONS

The Chan-Vese model is variational, based on an energy E_{CV} defined on functions $\phi : \Omega \rightarrow \mathbb{R}$ (see equation (4.30)). A gradient descent equation (equation (4.31)) is derived from this energy. To solve the Chan-Vese model, it is necessary to discretize the gradient descent equation in some way. Conversely, it is possible first to discretize the energy, and then solve the resulting finite dimensional optimization problem. The diagram in Figure 5.1 shows this situation. These two approaches may result in different discretization for minimizing the energy and making the diagram non commutative. In this section, we analyze the impact of these different discretizations.

5.1.1. Length of an implicit curve

To compare the impact of the discretization on the active Chan-Vese minimization problem, it is necessary to compute the energy $E_{CV}(\phi, c_1, c_2)$. However computing the length of an implicitly defined curve is not a trivial task. The length (or area in 3D) is given analytically

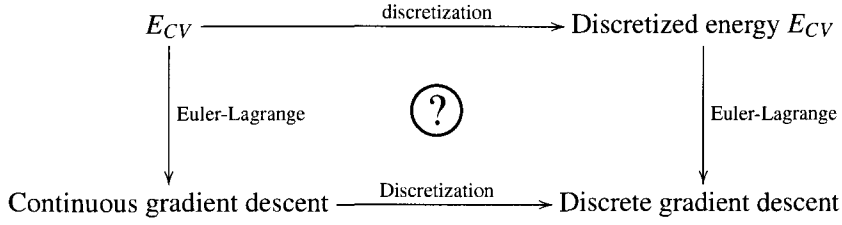


Figure 5.1: The commutative diagram representing the two possible ways to get a discretization.

by the integral

$$\mathscr{H}^{N-1}(\{\phi = 0\}) = \int_{\Omega} |DH(\phi)| = \int_{\Omega} \delta(\phi)|\nabla\phi|. \tag{5.1}$$

Here $|DH(\phi)|$ and $\delta(\phi)$ are measures. One approach in computing the length is to approximate the Dirac delta function by the Gaussian kernel

$$\delta_\beta = \frac{1}{\beta\sqrt{\pi}} e^{(x/\beta)^2}, \tag{5.2}$$

where $\beta > 0$ is a parameter controlling the degree of blurring. Hence in 2D the length can be approximated as

$$\mathscr{H}^{N-1}(\{\phi = 0\}) \approx \sum_{i,j} h^2 \delta_\beta(\phi_{i,j})|\nabla^+\phi_{i,j}|, \tag{5.3}$$

where

$$|\nabla^+\phi_{i,j}| = \sqrt{(D_x^+\phi_{i,j})^2 + (D_y^+\phi_{i,j})^2}, \tag{5.4}$$

D_x^+ and D_y^+ standing for the forward finite difference operators. For a given choice of β, it is possible to compute an approximation to the length.

An alternate approach is to compute numerically the integral $\int_{\Omega} |DH(\phi)|$. In this case,

$$\mathscr{H}^{N-1}(\{\phi = 0\}) \approx \sum_{i,j} h^2 |\nabla^+ H(\phi)_{i,j}|. \tag{5.5}$$

It is a coarse approximation since at every pixel $|\nabla^+ H(\phi)_{i,j}|$ can only take values 0, $1/h$ or $\sqrt{2}/h$. Indeed, much information about the interface present in ϕ is lost in the function $H(\phi)$. However, it has an important advantage over the Gaussian kernel approximation since it has a very narrow support. The support is one pixel wide, for whatever scaling of the function

ϕ, meaning that the length computed remains unchanged while rescaling the function ϕ. Although imprecise, this approximation is the only truly invariant method.

Smereka [104] proposed a modified version of this approximation method. His discrete delta function $\tilde{\delta}$ is given by

$$\tilde{\delta}(\phi_{i,j}) = \tilde{\delta}_{i,j}^{(+x)} + \tilde{\delta}_{i,j}^{(-x)} + \tilde{\delta}_{i,j}^{(+y)} + \tilde{\delta}_{i,j}^{(-y)}, \tag{5.6}$$

where

$$\tilde{\delta}_{i,j}^{(+x)} = \begin{cases} \dfrac{|\phi_{i+1,j} D_x^0 \phi_{i,j}|}{h^2 |D_x^+ \phi_{i,j}| |\nabla_0^\epsilon \phi_{i,j}|} & \text{if } \phi_{i+1,j}\phi_{i,j} \leq 0, \\ 0, & \text{otherwise.} \end{cases}$$

$$\tilde{\delta}_{i,j}^{(-x)} = \begin{cases} \dfrac{|\phi_{i-1,j} D_x^0 \phi_{i,j}|}{h^2 |D_x^- \phi_{i,j}| |\nabla_0^\epsilon \phi_{i,j}|} & \text{if } \phi_{i-1,j}\phi_{i,j} \leq 0, \\ 0, & \text{otherwise.} \end{cases}$$

$$\tilde{\delta}_{i,j}^{(+y)} = \begin{cases} \dfrac{|\phi_{i,j+1} D_y^0 \phi_{i,j}|}{h^2 |D_y^+ \phi_{i,j}| |\nabla_0^\epsilon \phi_{i,j}|} & \text{if } \phi_{i,j+1}\phi_{i,j} \leq 0, \\ 0, & \text{otherwise.} \end{cases}$$

$$\tilde{\delta}_{i,j}^{(-y)} = \begin{cases} \dfrac{|\phi_{i,j-1} D_y^0 \phi_{i,j}|}{h^2 |D_y^- \phi_{i,j}| |\nabla_0^\epsilon \phi_{i,j}|} & \text{if } \phi_{i,j-1}\phi_{i,j} \leq 0, \\ 0, & \text{otherwise.} \end{cases}$$

and

$$|\nabla_0^\epsilon \phi_{i,j}| = \sqrt{(D_x^0 \phi_{i,j})^2 + (D_y^0 \phi_{i,j})^2 + \epsilon}$$

From values of the level set function ϕ given at the vertices of the image's regular grid, it is easy to construct a piecewise linear approximation to ϕ. Then another approach is to compute the length of the 0-contour for this piecewise linear approximation. It is easily done since on a triangle, the 0 level set is a line segment that crosses two of the three sides of the triangles and the intersection points can be found in a straightforward manner.

We tested the different methods for computing the length of the 0-contour with a simple function: the signed distance function ϕ to a circle of radius 100. Table 5.1 shows the results of the use of the different Delta functions. We used the Gaussian approximation with β = 2.5 and β = 10, the Heaviside approximation, the discretization proposed by Smereka [104] as well as the piecewise linear approximation of ϕ. When using the signed distance function ϕ to the circle, all methods perform well, except the one using the Heaviside function $H(\phi)$

Table 5.1: Computation of the circumference of a circle of radius 100 via different discretization methods.

Real length	Gaussian ($\beta = 2.5$)	Gaussian ($\beta = 10$)	Heaviside	Smereka	Piecewise linear
628.3185308	623.0148349	623.0177399	722.2914139	622.7586591	623.0202264

which is off the real length. In real applications, the function ϕ will often not be a signed distance function. In the Chan-Vese algorithm, the level set function quickly becomes very irregular. Hence, the method for computing must be robust to changes in scaling of the function ϕ. We computed the same length using the functions ϕ^d with degrees $d = 3$, $d = 5$ until $d = 11$. The results are shown in Figure 5.2. It is clear from the results that the Gaussian approximation is not robust to changes in the scale of the level set function. This is due to the fact that the Gaussian approximation has a wider support. One advantage of the Heaviside approximation is that it will always give the same result for the length of the curve. The Smereka approximation will tend to get away from the real length, but very slowly. The same phenomenon happens for the piecewise linear approximation. The Smereka approximation remains the best approximation for the length of the circle.

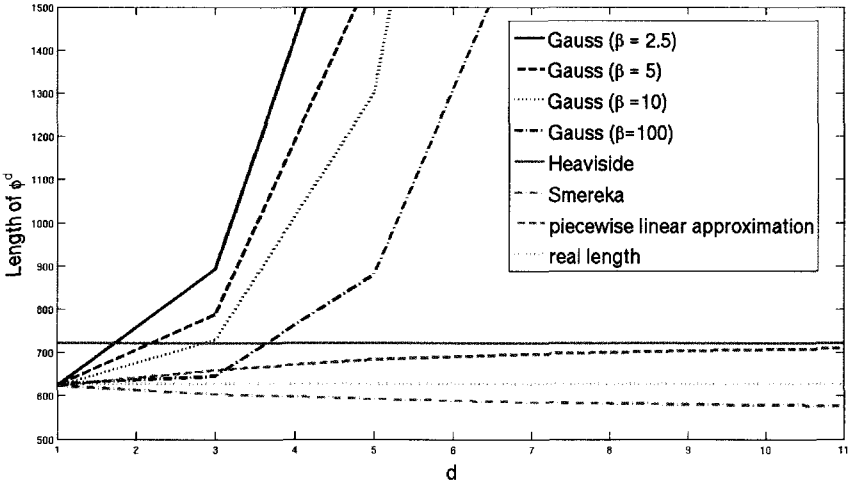


Figure 5.2: Computation of the circumference of a circle of radius 100 via different discretization methods and different rescaled level set functions ϕ^d.

Figure 5.3 shows the length computation when the level set function is multiplied by

a constant m. The same kind of results appears. Again, the wide support of the Gaussian approximation makes it less robust. The Smereka approximation and the piecewise linear approximation are scale-invariant for this type of scaling.

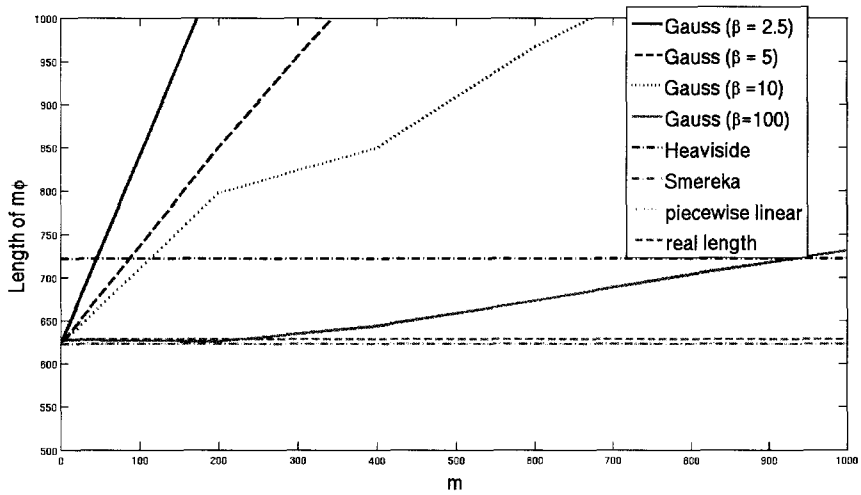


Figure 5.3: Computation of the circumference of a circle of radius 100 via different discretization methods and different rescaled level set functions $m\phi$.

From this study, we conclude that the Smereka approximation is the best choice for computing the length of an implicitly defined curve, although the piecewise linear approximations yields also good results.

5.1.2. Impact of discretization

We now analyze the impact of the choice of the discretization for the Chan-Vese problem. The equation

$$\phi_t = \delta(\phi)\left[\mu \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda(g-u_1)^2 + \lambda(g-u_2)^2\right] \tag{5.7}$$

can be solved numerically in many ways. We propose to study seven different ways of discretizing equation (5.7). The first five discretizations are obtained by discretizing the continuous Euler-Lagrange equation of the energy E_{CV}, that is the down and right paths of Diagram 5.1. For the last two, the energy E_{CV} is first discretized, and then a gradient descent equation is obtained for the finite dimensional minimization problem, which corresponds to going right and down in Diagram 5.1.

The first method of solution consists in replacing the Dirac delta function $\delta(\phi)$ by $|\nabla\phi|$ as explained in section 4.4.3. Then the curvature term

$$\kappa = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$$

is discretized via the simple finite difference scheme of equation (3.10).

The second method replaces $\delta(\phi)$ by $|\nabla\phi|$, but uses the fancier midpoint discretization described in equation (3.12).

The third method also replaces $\delta(\phi)$ by $|\nabla\phi|$ but it uses the semi-explicit scheme given in equation (3.14).

The fourth method implements the algorithm introduced by Chan, Esedoglu and Nikolova [19], presented in section 4.4.3. Here the factor $\delta(\phi)$ is suppressed and a correction term $-\alpha v'(s)$ is added to equation (5.7), where $s = (g - c_1)^2 + (g - c_2)^2$, as described in equation (4.41).

The fifth method replaces the term $\delta(\phi)$ by a regularized version of it

$$\delta_\beta(x) = \frac{1}{\beta\sqrt{\pi}} e^{(x/\beta)^2}. \tag{5.8}$$

The sixth method consists of computing a gradient descent equation for the Chan-Vese energy which is discretized via the Heaviside function. Recall that

$$\mathscr{H}^{N-1}(\{\phi = 0\}) \approx L_h = \sum_{i,j} h^2 |\nabla^+ H(\phi)_{i,j}|. \tag{5.9}$$

The energy functional is then a function of N variables, where N is the total number of voxels in the image. To compute the gradient of the discrete energy, we can simply find the partial derivatives of the energy with respect to all variables $\phi_{i,j}$. In H, only three terms will have non zero partial derivative with respect to $\phi_{i,j}$. Namely the terms

$$\begin{aligned} A &= h^2 \sqrt{D_x^+ H(\phi_{i,j})^2 + D_y^+ H(\phi_{i,j})^2}, \\ B &= h^2 \sqrt{D_x^+ H(\phi_{i-1,j})^2 + D_y^+ H(\phi_{i-1,j})^2}, \\ C &= h^2 \sqrt{D_x^+ H(\phi_{i,j-1})^2 + D_y^+ H(\phi_{i,j-1})^2}. \end{aligned} \tag{5.10}$$

Now

$$\frac{\partial A}{\partial \phi_{i,j}} = h^2 \frac{-D_x^+ H(\phi_{i,j}) - D_y^+ H(\phi_{i,j})}{\sqrt{D_x^+ H(\phi_{i,j})^2 + D_y^+ H(\phi_{i,j})^2}}, \tag{5.11}$$

$$\frac{\partial B}{\partial \phi_{i,j}} = h^2 \frac{D_x^+ H(\phi_{i-1,j})}{\sqrt{D_x^+ H(\phi_{i-1,j})^2 + D_y^+ H(\phi_{i-1,j})^2}}, \tag{5.12}$$

$$\frac{\partial C}{\partial \phi_{i,j}} = h^2 \frac{D_y^+ H(\phi_{i,j-1})}{\sqrt{D_x^+ H(\phi_{i,j-1})^2 + D_y^+ H(\phi_{i,j-1})^2}}. \tag{5.13}$$

Hence the total partial derivative is

$$\frac{\partial L_h}{\partial \phi_{i,j}} = -h^2 D_x^- \left[\frac{D_x^+ H(\phi_{i,j})}{\sqrt{D_x^+ H(\phi_{i,j})^2 + D_y^+ H(\phi_{i,j})^2}} \right] - h^2 D_y^- \left[\frac{D_y^+ H(\phi_{i,j})}{\sqrt{D_x^+ H(\phi_{i,j})^2 + D_y^+ H(\phi_{i,j})^2}} \right]. \tag{5.14}$$

The Euler-Lagrange equation for this problem is then

$$\frac{\partial \phi_{i,j}}{\partial t} = -\frac{\partial L_h}{\partial \phi_{i,j}} + \delta(\phi) \left[-(g - c_1)^2 + (g - c_2)^2 \right]. \tag{5.15}$$

This yields a discretization very close to the one described in equation (3.11), but with the function ϕ replaced by $H(\phi)$. This defines the sixth method. We do not expect that it gives nice results since a lot of information is lost when considering only $H(\phi)$, as for the similar method for computing the length of the 0-contour.

The seventh method is devised by doing the same kind of computations with the discrete length approximation

$$\mathscr{H}^{N-1}(\{\phi = 0\}) \approx \tilde{L}_h = \sum_{i,j} h^2 \delta_\beta(\phi_{i,j}) |\nabla^+ \phi_{i,j}|, \tag{5.16}$$

Again H' contains only three terms whose partial derivatives with respect to $\phi_{i,j}$ do not vanish. These terms are (note that we omit h^2 as it is a factor of all terms):

$$\begin{aligned} \tilde{A} &= \delta_\beta(\phi_{i,j}) \sqrt{D_x^+ \phi_{i,j}^2 + D_y^+ \phi_{i,j}^2}, \\ \tilde{B} &= \delta_\beta(\phi_{i-1,j}) \sqrt{D_x^+ \phi_{i-1,j}^2 + D_y^+ \phi_{i-1,j}^2}, \\ \tilde{C} &= \delta_\beta(\phi_{i,j-1}) \sqrt{D_x^+ \phi_{i,j-1}^2 + D_y^+ \phi_{i,j-1}^2}. \end{aligned} \tag{5.17}$$

Again, we can compute their partial derivatives.

$$\frac{\partial \tilde{A}}{\partial \phi_{i,j}} = \delta_\beta'(\phi_{i,j}) \sqrt{D_x^+ \phi_{i,j}^2 + D_y^+ \phi_{i,j}^2} + \delta_\beta(\phi_{i,j}) \frac{-D_x^+ \phi_{i,j} - D_y^+ \phi_{i,j}}{\sqrt{D_x^+ \phi_{i,j}^2 + D_y^+ \phi_{i,j}^2}}, \tag{5.18}$$

$$\frac{\partial \tilde{B}}{\partial \phi_{i,j}} = \delta_\beta(\phi_{i-1,j}) \frac{D_x^+ \phi_{i-1,j}}{\sqrt{D_x^+ \phi_{i-1,j}^2 + D_y^+ \phi_{i-1,j}^2}}, \tag{5.19}$$

$$\frac{\partial \tilde{C}}{\partial \phi_{i,j}} = \delta_\beta(\phi_{i,j-1}) \frac{D_y^+ \phi_{i,j-1}}{\sqrt{D_x^+ \phi_{i,j-1}^2 + D_y^+ \phi_{i,j-1}^2}}. \tag{5.20}$$

Hence the total partial derivative is

$$\begin{aligned} \frac{\partial \tilde{L}_h}{\partial \phi_{i,j}} = &-h^2 D_x^- \left[\delta_\beta(\phi_{i,j}) \frac{D_x^+ \phi_{i,j}}{\sqrt{D_x^+ \phi_{i,j}^2 + D_y^+ \phi_{i,j}^2}} \right] - h^2 D_y^- \left[\delta_\beta(\phi_{i,j}) \frac{D_y^+ \phi_{i,j}}{\sqrt{D_x^+ \phi_{i,j}^2 + D_y^+ \phi_{i,j}^2}} \right] \\ &+ \delta'_\beta(\phi_{i,j}) \sqrt{D_x^+ \phi_{i,j}^2 + D_y^+ \phi_{i,j}^2}. \end{aligned} \tag{5.21}$$

This is the discretization of the curvature term that is used for the seventh method. The Euler-Lagrange equation will be the same as with the sixth method shown in equation (5.15) with $\frac{\partial L_h}{\partial \phi_{i,j}}$ replaced by $\frac{\partial \tilde{L}_h}{\partial \phi_{i,j}}$.

We compare the use of these seven different methods on the cameraman image. The parameters μ and λ are set in order that the curvature term is dominant in the equation, in order to be able to see the influence of the discretization. We take $\mu = 2$ and $\lambda = 0.00004$, $\beta = 5$ and $\Delta t = 0.1$. The gradient descent method was run for a very long time to be sure that steady state was reached. Figure 5.4 shows the final binary segmentation of the cameraman image with the various discretizations. Figure 5.5 shows the various final level set functions.

In order to determine which segmentation method performs best, we computed the Chan-Vese energy on every segmentation result. To compute the length, the piecewise linear approximation, the Heaviside approximation as well as the Smereka approximations were performing well, so we computed the length in these three ways to compare the results. The energies are displayed in Table 5.2. The final level set function of a segmentation process can be far from being a signed distance function, so we did not compute the length using the Gaussian approximation of the Dirac delta function. It would be possible to reinitialize the level set functions to signed distance functions. However in doing so, there is, in general, a smoothing effect on the zero level set that may perturb the length calculation.

From Table 5.2, we conclude that methods 4 and 5 lead to the solutions with lowest energy. Methods 2 and 3 also yields solutions with low energy. Method 4 has the advantage that the final level set function has a lot of contrast. Hence, there is more freedom on which level curve to take. The final level set function of the method 6 is surprisingly close to the

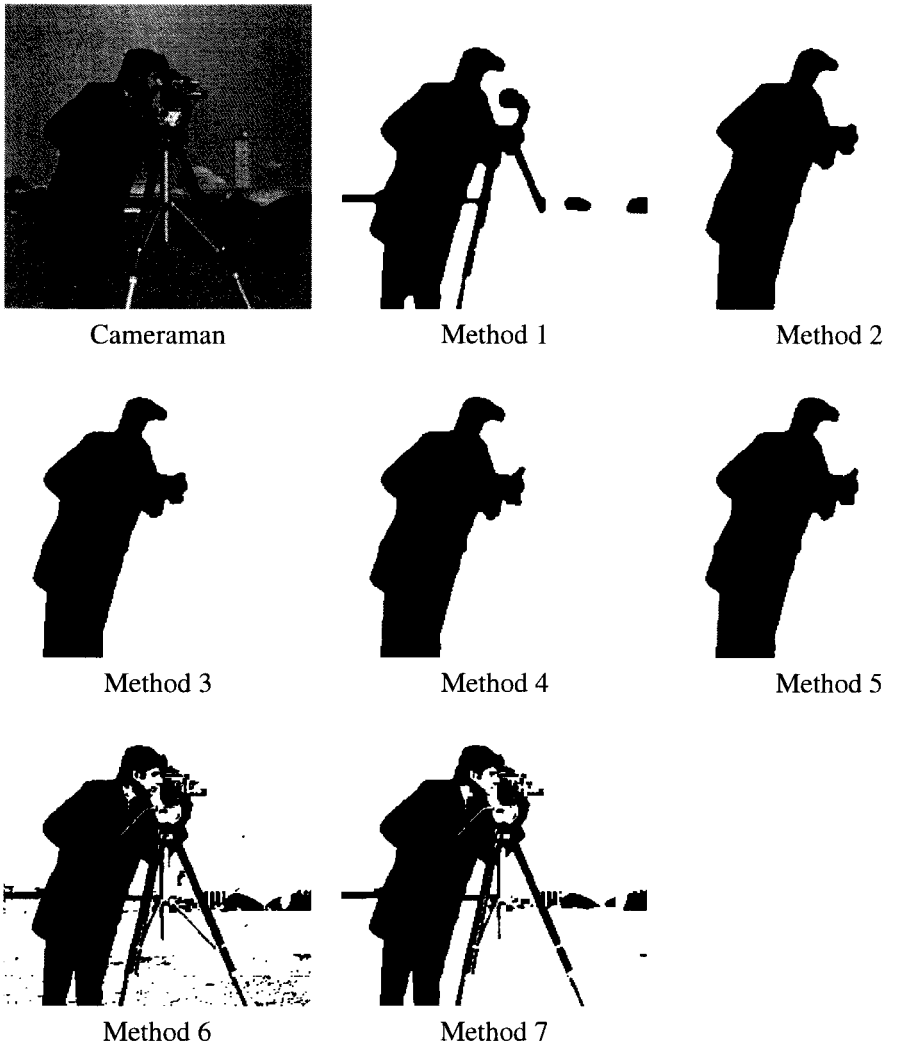


Figure 5.4: Binary segmentations of the cameraman by the different methods.

original image. We believe that this is due to the fact that the curvature term discretized in this way has a smaller weight in the gradient descent equation. In fact, the segmentations obtained by methods 6 and 7 in Figure 5.4 are somewhat similar to the result of the active contour without edges algorithm with a small weight for curvature as seen on Figure 5.8(b), where there is no curvature term. This is a drawback of these approaches.

The fifth method seems to be the one leading to the lowest energy. Hence we assume that its solution — the combination of its 0 level-set and of the values of c_1 and c_2 — is the

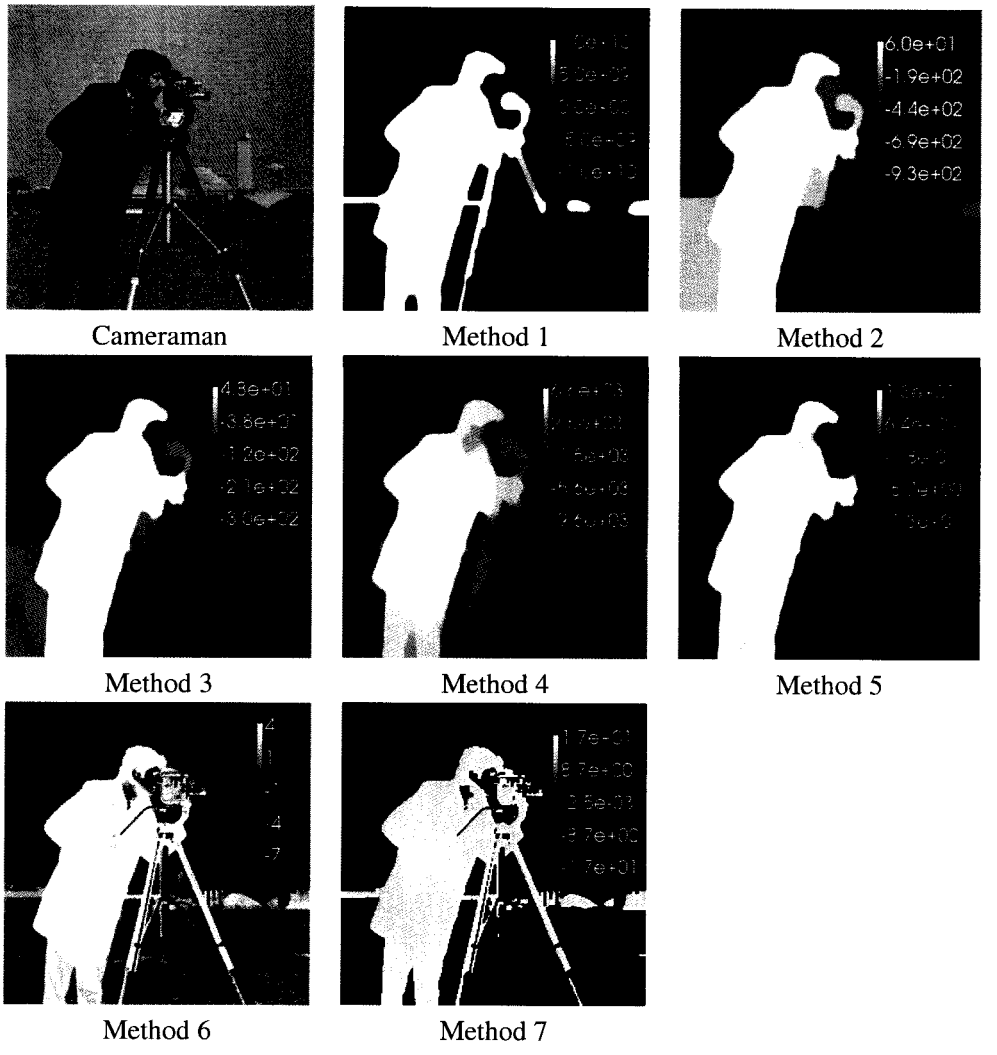


Figure 5.5: Final level set functions for the binary segmentation of the cameraman by the different methods.

global minimum of the Chan-Vese problem. Figure 5.6 shows the L^2 convergence towards this minimum with the various discretizations. If u^* is a level set function corresponding to the global minimizer of E_{CV}, then at each iteration the distance between $H(\phi^*)$ and the function $H(\phi)$ or $H(-\phi)$ is computed:

$$d_{L^2}(\phi^*, \phi) = \min\Big(\#\{x_i : H(\phi^*(x_i)) \neq H(\phi(x_i))\},\ \#\{x_i : H(\phi^*(x_i)) = H(\phi(x_i))\}\Big), \qquad (5.22)$$

Table 5.2: The energies associated with the different segmentation results.

Method	Heaviside	Smereka	Piecewise linear
1	5421.001806	5143.046447	5156.683373
2	5163.946193	5000.949844	5039.889972
3	5092.511354	4950.370264	4973.554848
4	5050.317000	4899.265609	4908.181506
5	5040.670162	4902.036943	4905.685890
6	10220.39522	8887.510197	9184.742464
7	7436.302185	6983.656071	7550.272390

where x_i stands for the midpoints of the image pixels. Again methods 4 and 5 perform better, as seen in Figure 5.6. Nevertheless, method 4 seems to converge more quickly towards the minimum. A surprising fact is that the sixth method initially converges very fast, then it stagnates.

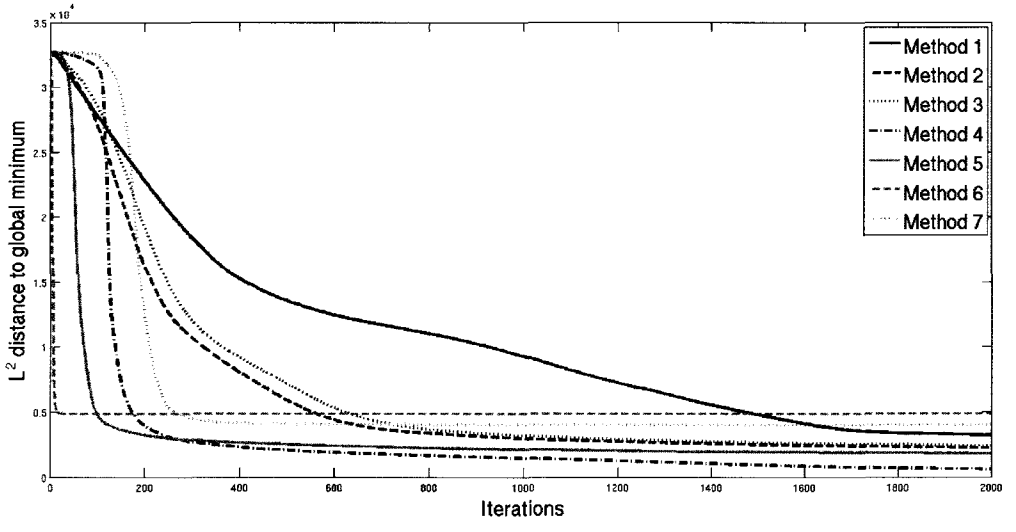


Figure 5.6: L^2 convergence to the global minimum with the various discretization methods

The behavior observed for the various discretization remains unchanged as the parameters λ and μ are varied. The same behavior has also been observed on many different images. In summary, we recommend to use method 4 which leads to a low energy and converges quickly. It is also a method that is very close to the original continuous Euler-Lagrange equation since only the Delta function is regularized.

5.2. INITIAL CONDITIONS

We will study how the problem behaves when different initial conditions are taken. In the seminal work Chan and Vese [21] propose to take as initial condition the signed distance function to a circle. Another choice is to take the signed distance function to a collection of circles spread out on the image. Figure 5.7 shows these choices of initial conditions.

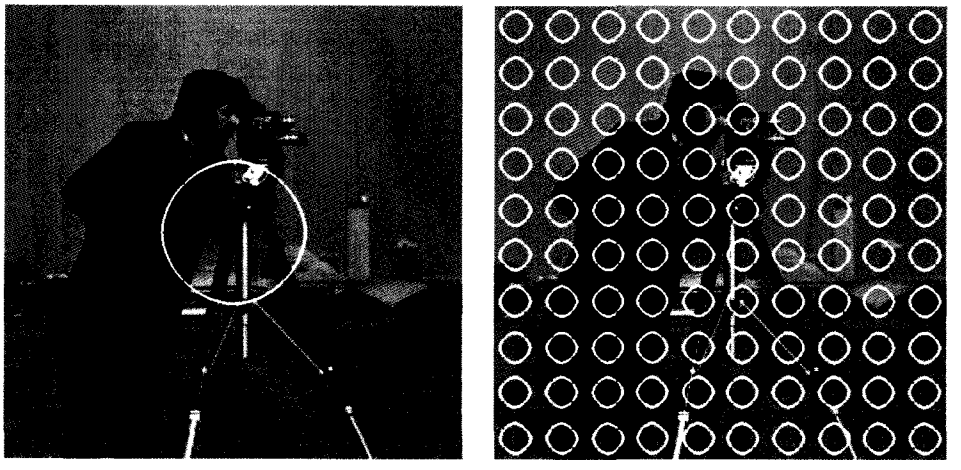
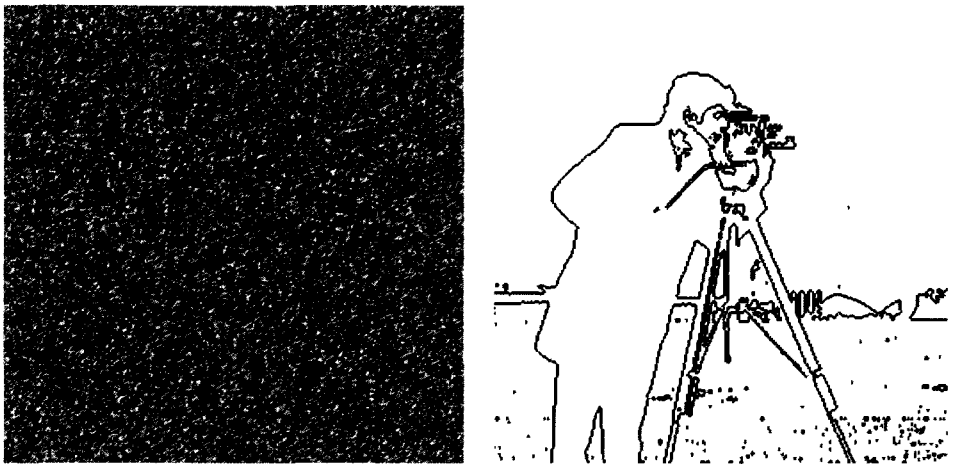


Figure 5.7: Classical choices of initial conditions for the Chan-Vese model.

There are mainly two concerns one might have regarding the choice of an initial condition. First, the energy is non convex so that it might have some local minima. Hence choosing a wrong initial condition may lead to a minimum that is not global. The second concern is the computational time. If the initial condition is far from the minimum, it usually takes more time steps to get to this minimum. For large 3D applications, the difference in CPU time can be very significant.

We propose two original choices of initial conditions. The first one consists in a function with random values uniformely distributed in $[-1, 1]$. It is inspired by the fact that taking a collection of circles instead of one circle increases the performance of the algorithm. The reason for this is that every pixel is closer to the interface, hence it is easier for it to change sign if needed. In a random image, every pixel is very close to the interface, so it can cross it easily. This should save computational time. Also, there are points inside and outside the curve all over the image, which will help capturing all pertinent information in the image and avoid local minima. Figure 5.8(a) shows this initial condition function for the cameraman image.

The other initial condition we propose is to take the solution of the Chan-Vese problem



(a) (b)

Figure 5.8: (a) The random initial condition: for each pixel a value is taken uniformely at random in $[-1, 1]$. (b) The initial condition that is the solution of the problem with $\lambda = 1$ and $\mu = 0$.

without curvature term. It has been mentioned in section 4.4 that if there is no curvature term in the Chan-Vese model ($\lambda = 1$ and $\mu = 0$), the problem can be solved very quickly. It corresponds, in fact, to the k-means problem with two clusters on the set of values taken by the image. This can be solved very quickly. Its solution $\bar{\phi}$ is a good initial condition for the more general problem since it should be close to the solution. It is possible to reinitialize the function $\bar{\phi}$ to a signed distance function to have a more regular initial condition. However this is not necessarily a good choice, since some points may be very far from the interface. Instead, we chose the initial condition

$$\bar{\phi}_0 = H(\bar{\phi}) - H(-\bar{\phi}).$$

Figure 5.8(b) shows the initial curve given by $\bar{\phi}_0$ on the cameraman image.

We now compare the results obtained with the four initial conditions on the cameraman image. The first initial condition is the signed distance function ϕ_C to a circle. The second is the signed distance function ϕ_{100} to a collection of 100 circles evenly spread over the image. The third is the random function ϕ_R and the last is the function $\bar{\phi}_0$ described above. The parameters are $\lambda = 1$ and $\mu = 10000$. The Chan-Vese model is solved for 100 time steps. All four initial conditions eventually lead to the same global minimum shown in Figure 5.10. However, it is not done at the same speed. Figure 5.9 shows the L^2 distance

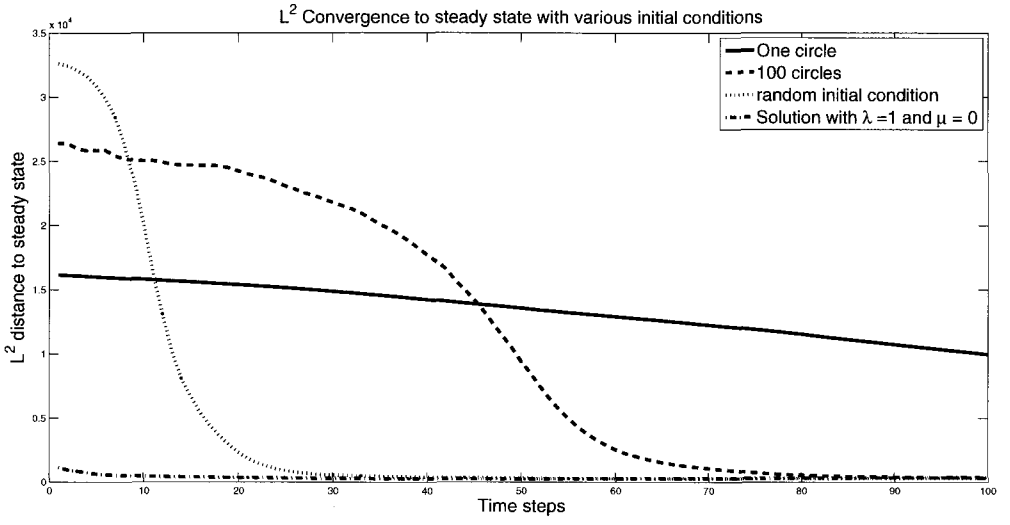


Figure 5.9: The L^2 convergence to the steady state under various initial conditions for the cameraman image.

to the steady state (that is the global minimum) as time evolves. The initial condition $\bar{\phi}_0$ is the one that gets to the steady state the quickest as we could expect. The random initial condition ϕ_R is the second, then the curve ϕ_{100} reaches the steady state. The curve given by the initial condition ϕ_C does not reach the steady state in 100 iterations. It takes about 600 iterations to reach it. Note that the initial condition $\bar{\phi}_0$ can be computed in a time negigible compared to the time for solving the Chan-Vese problem. For this particular problem, it takes 0.4572s. to compute $\bar{\phi}_0$.

We did the same calculations on a different image. It is the synthetic image of shapes with salt and pepper noise added with a probability of 50% shown in figure 5.11. The parameters are $\lambda = 2$ and $\mu = 1$. Due to the high level of noise, the algorithm had to run for about 2000 time steps to reach the steady state shown in Figure 5.12. The L^2 convergence to the steady state is shown in Figure 5.13. Exactly the same behavior as for the cameraman image is observed. We can conclude is that the initial condition $\bar{\phi}_0$ is a very good choice since it reduces considerably the number of iterations necessary to reach the steady state. On the other hand the random initial condition also converges quickly and provides a prior-free initial condition.

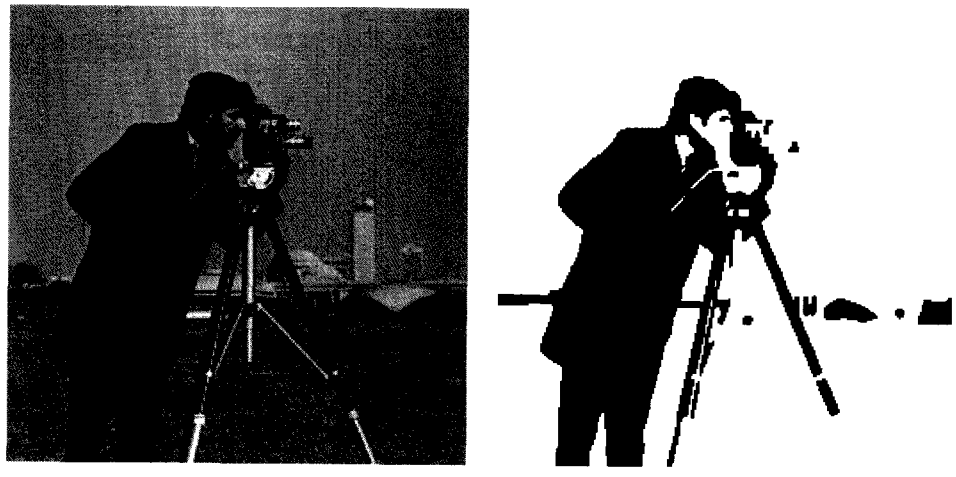


Figure 5.10: The steady state of the Chan-Vese model on the cameraman image with $\lambda = 1$ and $\mu = 10000$.

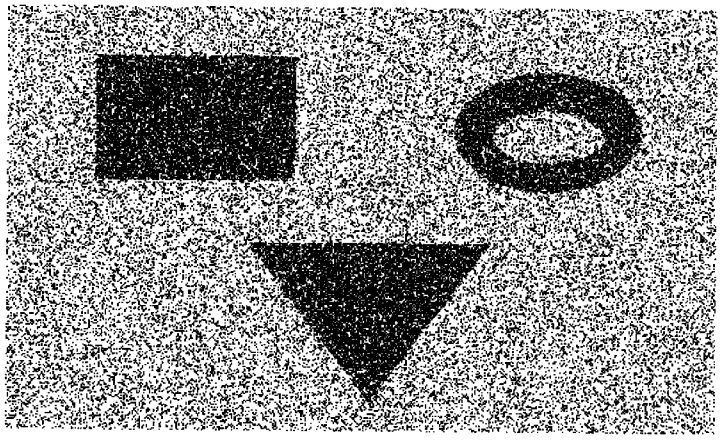


Figure 5.11: A synthetic image with salt and pepper noise added

5.3. HIERARCHICAL SEGMENTATION

We now focus on the hierarchical method proposed by Tsai, Yezzi and Willsky [113] and further investigated by Gao and Bui [39] that we described at the end of Section 4.4.2. The image is first split into two pieces using the Chan-Vese model. Then each of the pieces is considered as a new domain on which the Chan-Vese model is again applied. n steps in the segmentation process will lead to a segmentation into 2^n phases.

This seems very similar to the original multiphase algorithm proposed by Vese and

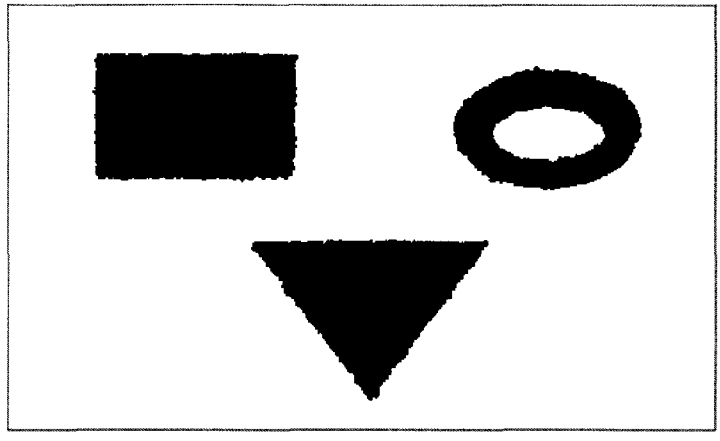


Figure 5.12: The steady state of the Chan-Vese model on a synthetic image with $\lambda = 2$ and $\mu = 1$.

Chan [117], where n level sets are evolved to split the image into 2^n phases (equations (4.34), (4.35) and (4.36) of section 4.4.2). But in doing so, n level set equations have to be solved simultaneously. The PDEs are then coupled and it is known to easily fall into local minima [39]. The hierarchical method is very stable and easy to solve, since at each step, there is only the two phase Chan-Vese problem to solve.

We also remarked that the multiphase algorithm may have some problems when there are triple regions. In order to describe triple junctions with two level set functions, the zero level sets of the two functions must be superimposed in some regions. In this case, if the curvature term is dominant, it may lead to miss-classification in this region, since the curves may not superimpose correctly. Figure 5.14 shows an example of this phenomena. To our knowledge, it is the first time that such artifacts are reported in the literature. We built this test case to emphasize the fact that the multiphase algorithm may behave badly when the curves should be superimposed. In the case of this synthetic image, the hierarchical method performs better as can be seen in Figure 5.15. The first step splits the image into two regions. Then each of these regions is split again into two sub-regions. The second step is computed separately on each of the two regions, and the homogeneous Neumann boundary condition on the boundary forces the contour to be normal to the boundary. Hence, it is not possible for the second contour to be superimposed on the first one since the level curve at the second step of the hierarchical segmentation would then be on the boundary of the domain, violating the Neumann boundary condition.

The hierarchical methods proposed in [113, 39] aim at a complete segmentation of the

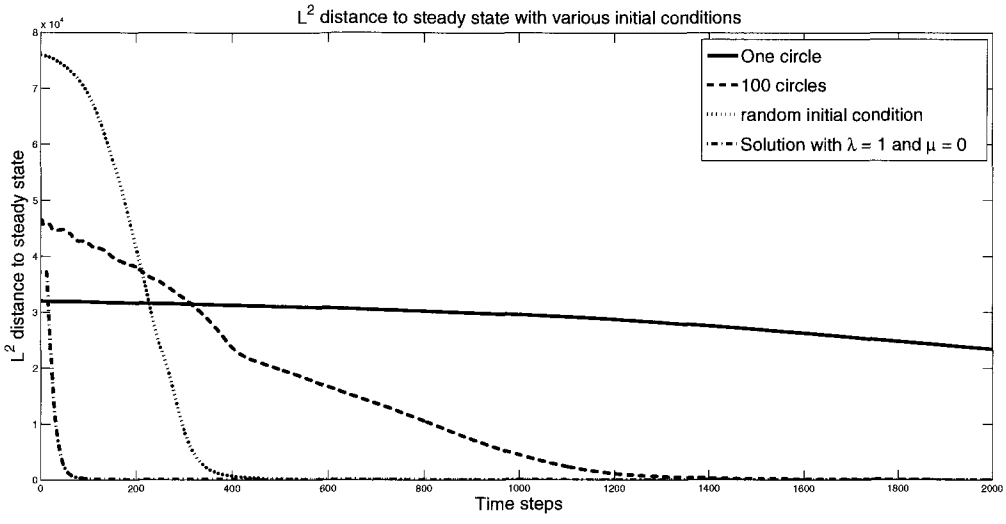


Figure 5.13: The L^2 convergence to the steady state under various initial conditions for a digital image with added salt and pepper noise.

image. Another important remark is that for most real life applications, it is not required to have a full segmentation of the image. For example, with medical images, one is mostly interested by the position and shape of one organ. Hence, the hierarchical segmentation can be done as follows:

1. Split the domain Ω into Ω^+ and Ω^- using the 2-phase Chan-Vese model.

2. Stop if the object is extracted, that is if the object is either Ω^+ or Ω^-. Otherwise, decide which of Ω^+ or Ω^- contains the object of interest, and pick this region as a new domain for step 1.

Figure 5.16 shows the results of the hierarchical segmentation process on a slice of a CT scan of the heart. At each stage, one side of the segmentation is chosen, the other side (in green) is ignored. Convergence is attained when no more pixels change sign.

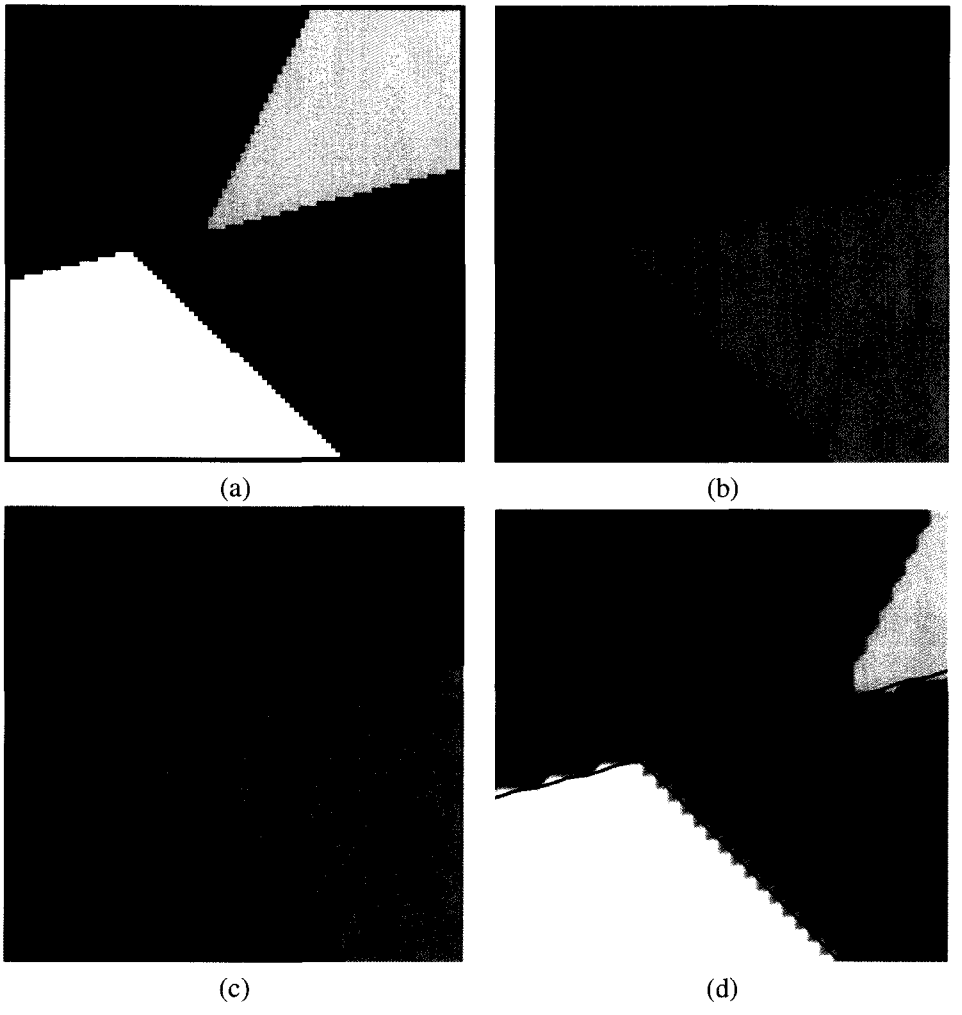


Figure 5.14: An example of an image for which segmentation with two level set function may lead to incorrect segmentation. (a) is the synthetic image to be segmented. (b) shows a correct segmentation of the image, when the curvature term is not dominant ($\mu = \lambda = 1$). (c) is the result if the curvature term is more important ($\mu = 10000$, $\lambda = 1$): there is a miss-classification of some pixels. (d) shows a close-up on the curves where there is miss-classification: it comes from the fact that level curves are only nearly superposed in these regions.

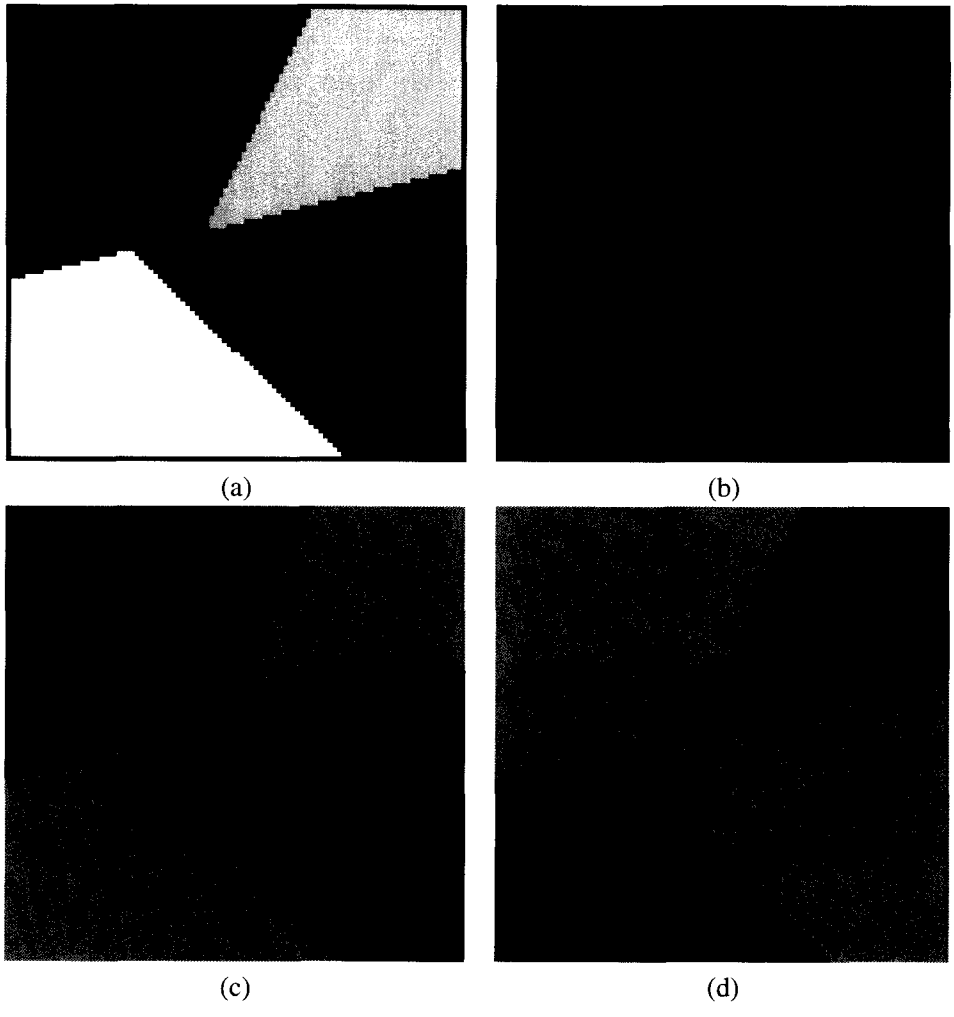


Figure 5.15: Iterative segmentation of the synthetic image (a). (b) shows the two-phase Chan-Vese model applied to the image ($\mu = 10000$, $\lambda = 1$). (c) is the result of the two-phase algorithm applied to the red region of (b) ($\mu = 100000$, $\lambda = 1$) and (d) shows the two-phase segmentation of the blue part of (b) ($\mu = 10000$, $\lambda = 1$). Note that the parameters are different from one iteration to the next since they mainly depend on the size of the fidelity term.

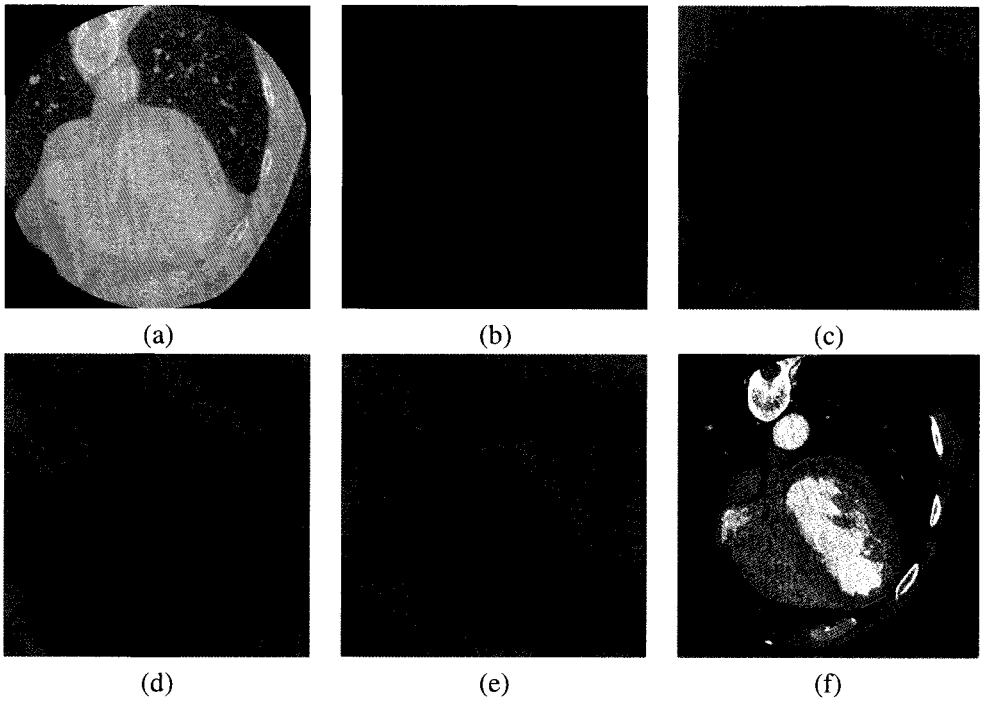


Figure 5.16: 2D Iterative segmentation of the heart from a CT scan image. (a) The image to be segmented (b) the result of the first application of the 2-phase Chan-Vese model, the red part is the region of interest (c) Second step: in green is the region that has been ignored in the segmentation. The blue region will be chosen. (d) Third step, the red region will be segmented further (e) The fourth and final step. (f) shows the connected component containing the heart muscle.

Step	Time step	μ	λ	Nbr. of iterations	CPU time	Cumulative CPU time
1	0.1	0	1	4	1s.	1s.
2	0.1	1	0.01	5	1s.	2s.
3	0.1	1	0.00008	1248	50s.	52s.
4	0.1	1	0.0002	378	10s.	62s.

Table 5.3: Computational times and parameters for the hierarchical segmentation of a 2D slice of a cardiac CT scan. The total computational time is 62s. Steps 3 and 4 take more time since a higher curvature term is needed because of the noisy nature of the image.

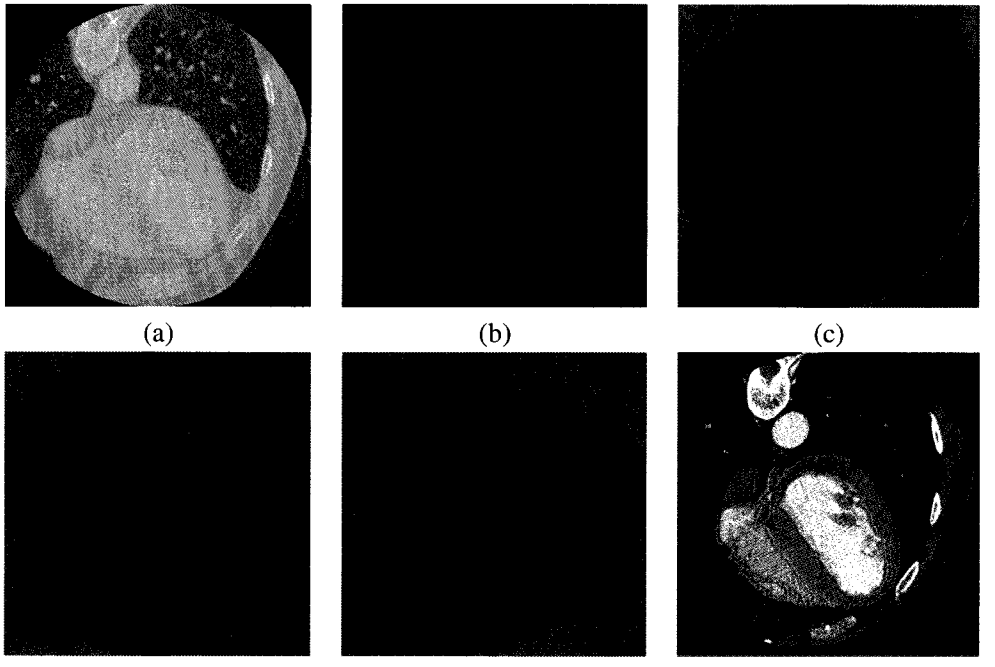
The parameters and computing times for this hierarchical segmentation are given in Table 5.3. It shows a great advantage of this method, which is that different parameters can be chosen from step to step, depending on the nature of the phases to separate. Nevertheless, there are no automatic way to set the parameters; they are tuned by letting them vary and observing the results.

For the first two steps, the segmentation is easy since the phases have sharp boundaries. Hence, only a small influence of the curvature term is needed, which means that the ratio μ/λ is small. For steps three and four, the phases to separate have quite close gray tones and they contain a lot of noise and texture, hence a higher value of the ratio μ/λ is needed in order to obtain the desired segmentation. Note that the fourth step of the hierarchical segmentation in Figure 5.16(e) is not perfect since the cavity of the right ventricle is not well segmented. This is due to the high noise level in this region and the little difference in gray tones between the muscle and the cavity.

To overcome the problems arising from the high noise level, the algorithm can be applied to a blurred version of the original image g. Also, the degree of smoothing can change from step to step in order to help the segmentation. An easy way to blur the image is to simply apply the heat equation (equation (3.1)) for a certain number of time steps. Figure 5.17 shows two different blurrings of the image via the heat equation with conductivity 1 and time step $dt = 0.1$. The first image is after 5 time steps (g_5) and the second after 15 time steps (g_{15}). Figure 5.18 shows the results of the hierarchical segmentation process using these images. If only g_5 is used at the last step, the result is similar to the one shown in Figure 5.16(e). This is the reason why we favored the smoother image g_{15}. Using a blurred version of the image is not an ideal situation since, when smoothing the image, edges can shift as described in chapter 3. Table 5.4 gives the different computational times and parameters to obtain the segmentation shown in Figure 5.18. The overall CPU time is smaller using blurred images since the parameter λ can be taken larger and fewer time steps

(a) g (b) g_5 (c) g_{15}

Figure 5.17: Blurred versions of the original 2D slice of a cardiac CT scan. (a) the original image g, (b) the image g_5 after 5 time steps of the heat equation and (c) the image g_{15} after 15 time steps of the heat equation.



(a) (b) (c)

(d) (e) (f)

Figure 5.18: 2D Iterative segmentation of the heart from a CT scan image using blurred images. (a) The image to be segmented, (b) the result of the first application of the 2-phase Chan-Vese model using the original image g, (c) Second step, obtained using the blurred version of the image g_5, (d) Third step, obtained using g_5, (e) Fourth step, obtained using g_{15}, (f) The final segmentation of the heart muscle over the 2D image.

Step	Time step	μ	λ	Nbr. of iterations	CPU time	cumulative CPU time
1	0.1	0	1	4	1s.	1s.
2	0.1	1	0.01	5	1s.	2s.
3	0.1	1	0.001	76	3s.	5s.
4	0.1	1	0.0002	675	21s.	26s.

Table 5.4: Computational times and parameters' values for the hierarchical segmentation of a 2D slice of a cardiac CT scan using smoothed versions of the image. The total computational time is 26s.

are required to reach convergence.

The results of the hierarchical segmentation suggests that if the image is to be segmented by the original multiphase algorithm of Vese and Chan, four different level sets functions would be necessary. This makes the implementation very difficult, as equations (4.35) and (4.36) needs to include many more terms. Also, when proceeding in this manner, the user need to know at the beginning the number of phases needed, which may be hard for complex images. Given that the multiphase algorithm has shown some weakness with as less as two level set functions on a simple synthetic image (see Figure 5.14), we did not try this approach on the CT scan of the heart.

5.4. HIERARCHICAL SEGMENTATION WITH L^1 FIDELITY

We now modify the hierarchical segmentation method by using an L^1 fidelity term. As discussed earlier, the L^1 fidelity is more natural than the classical L^2 fidelity term. The model gets more robust to noise using L^1 fidelity as can be seen on Figure 5.19. For images (a) and (b), the parameters are the same: $\mu = 1$ and $\lambda = 0.1$, only the fidelity changes. On the result with L^1 fidelity, there is less noise. It is possible to get a similar segmentation with the L^2 fidelity by changing the parameters λ and μ. Taking $\mu = 1$ and $\lambda = 0.0005$ yields a similar segmentation as with the L^1 fidelity (see Figure 5.19(c)). Although it is hard to say in this case, in general some features could be over-smoothed by taking a high ratio μ/λ with an L^2 fidelity term.

Figure 5.20 shows the results of the hierarchical segmentation of the 2D slice of a CT scan of the heart using L^1 fidelity. Using L^1 fidelity, only 3 steps are necessary to segment the heart muscle. This is because the problem with L^1 fidelity tends to split the image into regions of similar size. Note also that a clean segmentation of the heart muscle is attained using only the original image. This was not possible for L^2 fidelity as shown in Figure 5.16. This is because the median is less sensitive to noise. Table 5.5 shows the parameters and the

(a) (b) (c)

Figure 5.19: The results of a segmentation of the cameraman image by the Chan-Vese model with (a) the L^2 fidelity and (b) L^1 fidelity. The parameters $\mu = 1$ and $\lambda = 0.1$ are the same in both cases. In (c), the result of the Chan-Vese model with L^2 fidelity using $\mu = 1$ and $\lambda = 0.0005$.

computational times for this hierarchical segmentation. Note that the overall computational time is similar to the one for L^2 fidelity. This is partly due to the fact that the median value of a set is longer to compute than the mean value. Note that even though step 3 requires more time steps to be solved, it is done faster than step 2 of the hierarchical method. This is due to the fact that the domain to be segmented gets smaller and smaller from one iteration of the hierarchical method to the next. This speeds up the computation time.

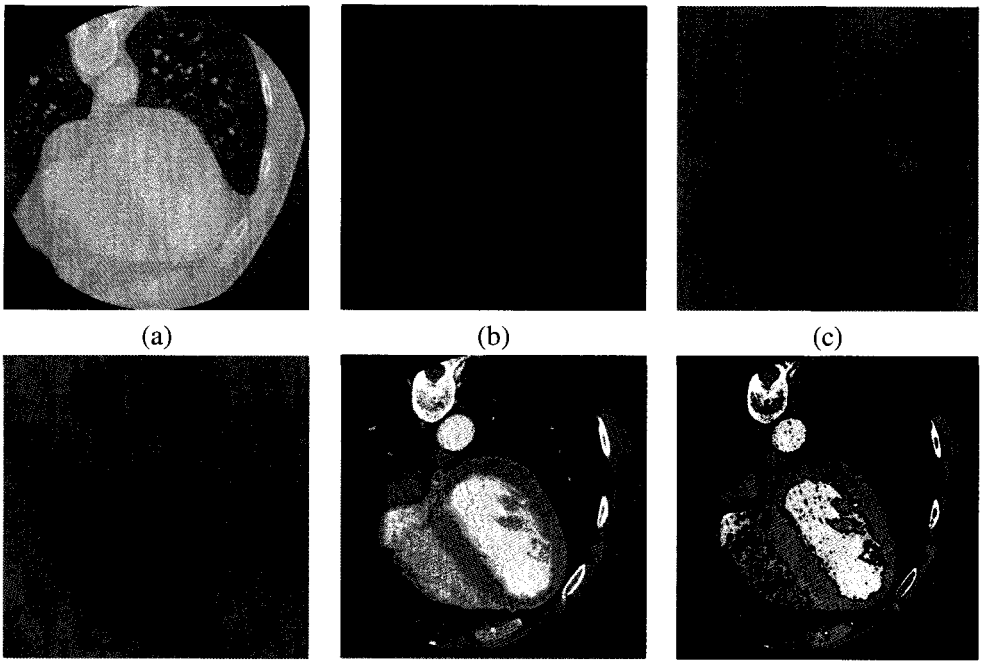


Figure 5.20: Hierarchical segmentation of the heart from a 2D slice of a CT scan using the L^1–fidelity. 3 steps are required rather than 4 for the L^2–fidelity. (a) is the original image and (b) is the result of the first iteration of the hierarchical segmentation. The blue region contains the heart. (c) Second step, the red region will be further segmented. (d) The result of the final step. (e) A full segmentation of the heart muscle is obtained from result (d). (f) shows the difference between the L^1 and L^2 segmentations. The L^1 segmentation is in blue and the L^2 segmentation is in red.

It is also possible to get a segmentation of the muscle using blurred versions of the image g. The results are shown in Figure 5.21. For the first two steps, the blurred image g_5 is used and for the last step the blurred image g_{15} is used.

Step	Time step	μ	λ	Nbr. of iterations	CPU time	Cumulative CPU time
1	0.1	0	1	5	1s.	1s.
2	0.1	1	0.01	493	31s.	32s.
3	0.1	1	0.005	524	22s.	54s.

Table 5.5: Computational times and parameters for the hierarchical segmentation of a 2D slice of a cardiac CT scan using L^1 fidelity. The total computational time is 54s.

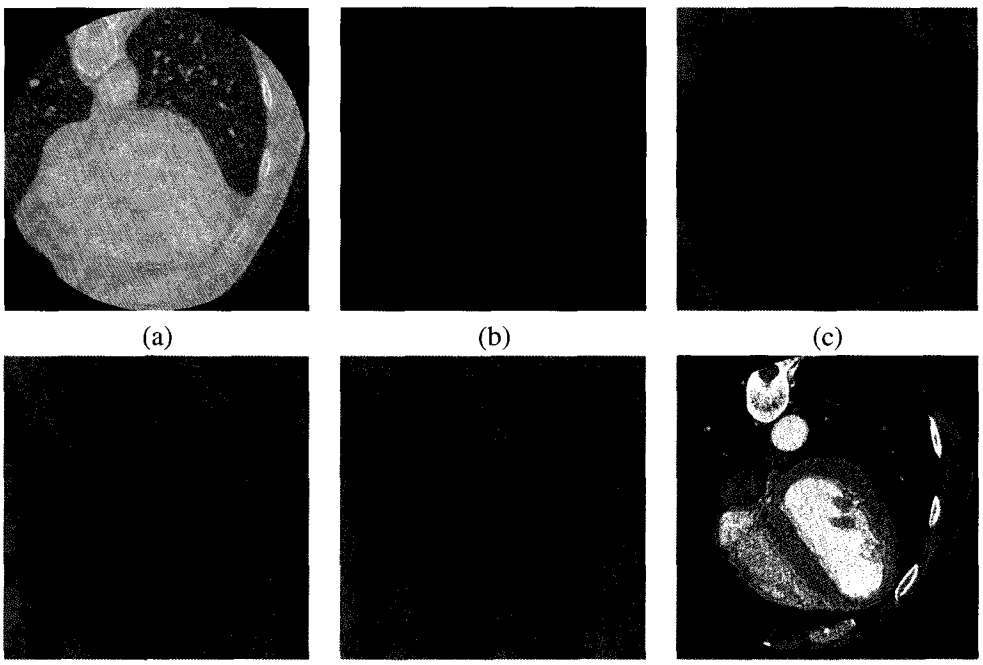


Figure 5.21: 2D Iterative segmentation of the heart from a CT scan image using blurred images and L^1 fidelity. (a) The image to be segmented, (b) the result of the first application of the 2-phase Chan-Vese model using the blurred image g_5, (c) Second step, obtained using the blurred version of the image g_5, (d) Third step, obtained using g_{15}, (e) The final segmentation of the heart muscle over the 2D image.

Step	Time step	μ	λ	Nbr. of iterations	CPU time	Cumulative CPU time
1	0.1	0	1	5	1s.	1s.
2	0.1	1	2	24	2s.	2s.
3	0.1	1	1	57	2s.	5s.

Table 5.6: Computational times and parameter values for the hierarchical segmentation of a 2D slice of a cardiac CT scan using smoothed versions of the image. The total computational time is 5s.

The computational times as well as the parameter values used are shown in Table 5.6. The total computational time is 5s, which is much faster than for the L^2 fidelity with the blurred image which was 26s. In Figure 5.22, the result of this new segmentation is compared with the one obtained without smoothing. As expected, the segmentation using blurred images yields smoother contours, but they might be slightly shifted in some regions.

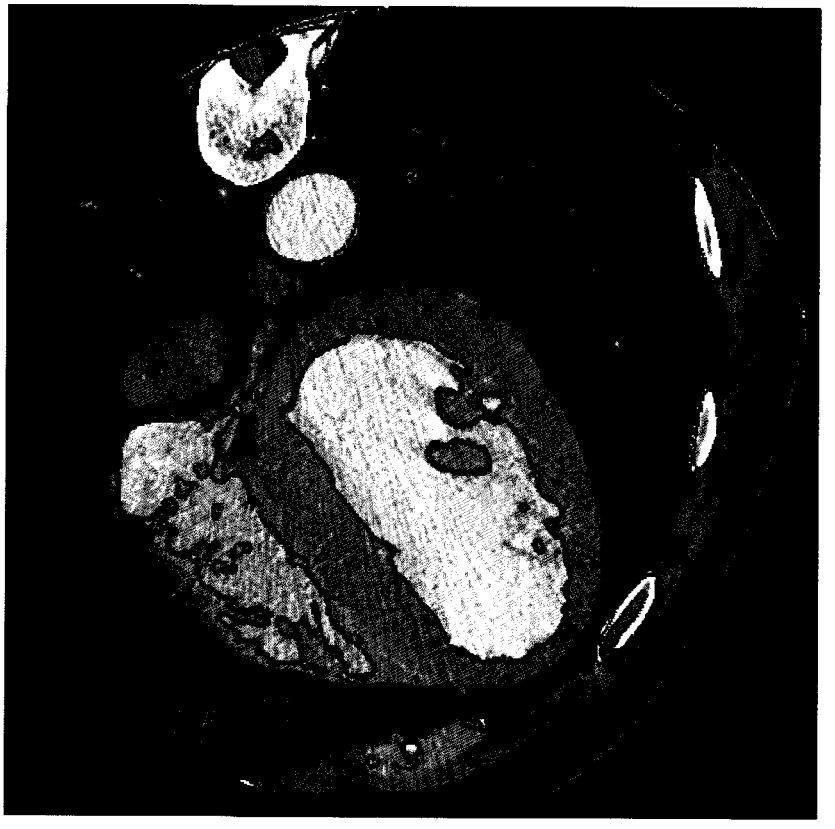


Figure 5.22: Hierarchical method with L^1 fidelity: differences between the smoothed and unsmoothed results. The result with the blurred image is in red while the result with the original image is in blue.

6

Endocardium segmentation

In this chapter, we present how the Chan-Vese algorithm can be used to segment the endocardium from the high resolution CT scan presented in Section 1.2.2. We segment with this technique the endocardial surfaces of the ventricles and of the atria. A parallel implementation of the computations is used to speed up the algorithm. The segmentation is very precise and contains many fine features of the heart's cavities, such as the pillars. The outer surface of the heart is almost completely segmented, but the surface suffers from leaking in several regions.

To our knowledge this is the first application of the Chan-Vese algorithm to cardiac segmentation. The only medical application we know is for brain segmentation [23]. This is also the first time that this algorithm is used on such large images and we believe that we have done its first parallel implementation.

As mentioned in Section 1.3, most of the existing segmentation methods for left ventricle endocardial surfaces will only find the approximate shape of the ventricle. The pillars are rarely segmented. Segmentations similar to the one obtained here can only be found in the work of Fritscher and Schubert [38] on MRI data. They used the maximum a posteriori (MAP) method introduced by Leventon, Grimson and Faugeras [58].

In the light of the previous analysis of the Chan-Vese algorithm made in chapter 5, the segmentation will be done using the L^1 Chan-Vese problem

$$\begin{cases} \phi_t = |\nabla\phi| \left[\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - |g - c_1| + |g - c_2| \right], & \Omega \\ \frac{\partial\phi}{\partial n} = 0, & \partial\Omega \\ \phi(\cdot, 0) = \phi_0, & \\ c_1 = \text{ median value of } g \text{ in } \{\phi >= 0\} & \\ c_2 = \text{ median value of } g \text{ in } \{\phi < 0\}, & \end{cases} \tag{6.1}$$

where the initial condition ϕ_0 is chosen to be a random initial condition as described in

section 5.2. The equation will also be solved with the semi-explicit scheme and space discretization presented in equation (3.14) The hierarchical method studied in section 5.3 is used to assess the multiphase segmentation.

6.1. PARALLEL IMPLEMENTATION

We now focus on how Problem 6.1 can be solved in parallel. The domain is described by the structured grid of the underlying image. A domain decomposition is easily achieved by slicing the image in one direction. We decided to slice the image in the z direction. Then the resulting blocks are distributed on the given processors. For example, the CT image described in Section 1.2.2 is of size 512×512×199. If two processors are used, the image will be split into two blocks: one of size 512×512×100 and a second of size 512×512×99.

Problem 6.1 is solved with a semi-explicit scheme. Starting with the initial condition ϕ_0, the solution is computed forward in time by time stepping. Using the semi-explicit scheme each processor can compute independently the solution ϕ^{n+1} at time $t_{n+1} = t_n + \Delta t$ from the solution ϕ^n. Boundary conditions of each block requires special care.

In the C++ implementation, for each processor the image block is augmented by one in all directions. For example the image block of size 512×512×100 will be contained in the middle of an array of size 514×514×102. Then boundary conditions are easily set. On a face of the image block that corresponds to the boundary of the image, the outside values are set to satisfy a homogeneous Neumann boundary condition. In this case the voxels are set to the same value as the closest voxel that is in the image. The other case is if a face corresponds to a slicing plane of the data set, which means that the image does not end, but voxels belong to another processor. In this situation, the voxel values on the two slices on each side of the slicing plane need to be exchanged between the processors. This is illustrated in figure 6.1 where a 2D domain of 100×100 pixels is split in 2 parts. The domain is extended by on one pixel in every direction. At each iteration, the blue part is set to satisfy the boundary condition. Also each processor will send its yellow slice to the red slice of the processor sharing the slicing line.

To solve the problem 6.1, it is also necessary to compute the values c_1 and c_2 which are the medians of the values of g taken in two different sub-domains. It is not straightforward to write a parallel algorithm that computes the median of a distributed data set.

We first describe the structure of a serial recursive algorithm for finding the k^{th} smallest element in a data set of n elements. This is done by a function $k^{th}\_element(X, k, n)$ where X

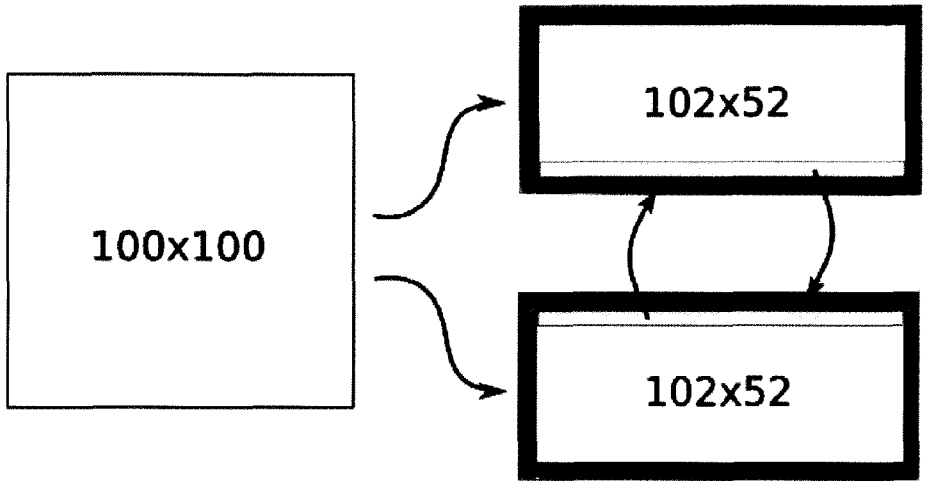


Figure 6.1: An example of a 2D domain decomposition. At each time step the region in blue is set according to the Neumann boundary condition and each processor send its yellow slice to the red slice of the neighboring processor.

is the data set, k is the position of the desired value (it is $n/2$ if the median is wanted), and n is the number of elements in X. The recursive algorithm is implemented as follows.

k^{th}*_element*(X, k, n) :

1. Choose a tentative k^{th} element m.
2. Let $X_m^- = \{x \in X : x \leq m\}$, $X_m^+ = \{x \in X : x \geq m\}$.
   Compute $n^- = \#X_m^-$ and $n^+ = \#X_m^+$.
3. (Stopping criterion) If $n^- \geq k$ and $n^+ \geq n - k$, return m as the median value.
4. Else,
   (a) if $n^- \leq k - 1$, the median is in the set X_m^+. It is the $(k - n^-)^{th}$ element in X_m^+. Therefore return k^{th}*_element*$(X_m^+, k - n^-, n^+)$.
   (b) If $n^- \geq k$, the median is in the set X_m^-. It is the k^{th} element in X_m^-. Therefore return k^{th}*_element*(X_m^-, k, n^-).

Now this algorithm is easily portable to a parallel context where the data is distributed among several processors. The structure of the algorithm is exactly the same as above. The data set X is distributed over the processors. The values of all other variables (k, n, m, $n^{\pm}$) need to be known by every processor. k, n and m need only be shared by all processors. To

find the values of $n^{\pm}$, the processor i can compute a value $n_i^{\pm}$ from his part of the data set (say X_i). The global value is then simply

$$n^{\pm} = \sum_i n_i^{\pm}. \tag{6.2}$$

Then $n^{\pm}$ need just be broadcasted to all processors. Steps 3 and 4 do not require special attention. During the recursive algorithm, it may be worth noticing that even if each processor calls the function $k^{th}\_element(X, k, n)$ with the same values of k and n, each processor only sees a part of the data set X.

In the first step, one needs to choose a tentative value m for the k^{th} element. There are many approaches to choose this value. If m is close to the real k^{th} value, the algorithm stops with fewer iterations. However, a close value may take longer to guess. Hence, there is a trade-off between the accuracy of the initial m and the time needed to compute it. The choice to be made may depend on the applications. There exist clever ways to select m that speeds up the computations. However they may be difficult to implement. In many concrete applications, m is just chosen randomly among the set of values.

Our choice was between fancy algorithms and random choice. If the processor i has n_i elements, then the k^{th} element is around the $(k * n_i/n)^{th}$ element m_i of the processor i. We take m has the mean value of m_i over all processors. The element m_i is simply calculated with the `nth_element` C++ template [48]. It is a linear algorithm for partial sort (it is more efficient than complete sort which is of order n log(n)). On a 52 000 000 values data set, this recursive parallel median algorithm usually terminates within 5 to 10 iterations.

For the global problem 6.1, most of the computations are made locally on one processor. The amount of information to transfer is small (only two slices need to be transferred between each pair of successive processors and a few numbers during the median calculation) compared to the computational load of an iteration. Hence, the speed up of the algorithm is nearly perfect: using n processors divides the time per iteration by n, as depicted in Figure 6.2.

For large problems, most computations are made on a 16 processor SUN cluster (AMD opteron 250, 2.4Ghz, TCP/IP interconnected with distributed memory). Running the code on 6 processors divides the CPU time by almost 6. As the number of processors increases, the time lost in data transmission becomes more important. Note that many computations have been made successfully on a simple dual-core laptop, cutting the CPU time in 2, although the same amount of RAM is required.

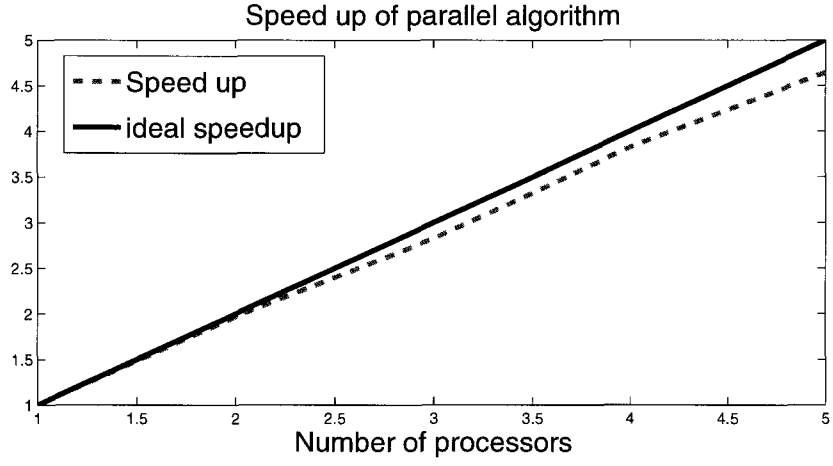


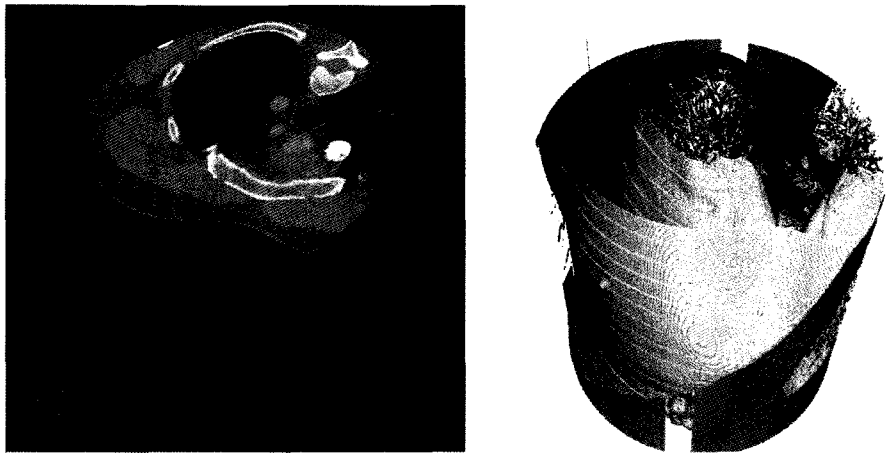
Figure 6.2: The speed up for the parallel algorithm: the time for doing an iteration with one processor divided by the time for doing an iteration with n processors. It is just below the ideal speedup.

With more than 6 processors, this implementation is less effective as the amount of data to be transferred is larger compared to the size of the blocks (the blocks become flat prisms). To solve with more processors, it would be better to split the domain in the 3 different directions, to obtain small cubes. We did not have to do this since the gain from computing with 6 processor is good enough for our applications.

6.2. THE SEGMENTATION

We now apply this algorithm to the high resolution CT scan described in Section 1.2.2. Figure 6.3 shows the 3D CT image as well as the result of the first segmentation step. Following this first segmentation, one side of the surface must be chosen as the new segmentation domain. The interior is the region containing the trachea, the exterior contains the heart. For the first step, as for the 2D case described in section 5.4, there is no need to impose curvature term. Hence, we take $\lambda = 1$ and $\mu = 0$. The parameters and computational times for the three iterations of the hierarchical method are gathered in Table 6.1.

To segment the heart, we need to take the result of the first segmentation and choose the exterior of the surface. To get to the heart, two more segmentation steps are required (as in the 2D case described in Section 5.4). If this would be done with a multiphase method, it would then take 3 different level set functions to isolate the phase obtained in the third step of the hierarchical method from the image. The 3 different PDEs would have to be solved



(a) (b)

Figure 6.3: The first step of the segmentation process. (a) is the CT scan to be segmented. (b) is the result of the first segmentation step.

simultaneously, which is hard to do on images of this size.

From the second stage, we have decided to do the segmentation not on the exact image but on a blurred version of it, due to the high level of noise. To blur the image, we apply the linear heat equation on the image for a few time steps with an explicit scheme. This is also easily done in parallel.

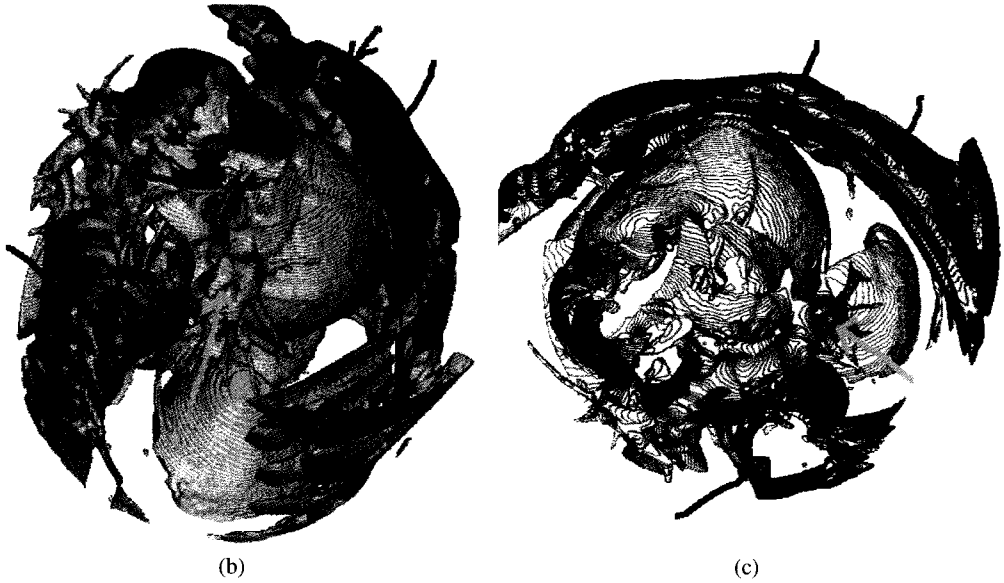
In the 2D example of Section 5.4, we remarked that using the L^1 fidelity allows to obtain a similar segmentation by using only the original image. This is the best way to proceed precision-wise, since some information is lost when blurring the image. However, the number of iterations increases considerably. We decided to go with an in intermediate alternative, by slightly blurring the image to increase efficiency without losing too many fine details.

Figure 6.4a shows the result of the second step of the hierarchical segmentation together with the domain boundary in orange. Figures 6.4b and 6.4c shows two views of the 0-level surface alone. Color arrows point several interesting anatomical regions. The red arrows point to the region containing the heart. The heart is wrapped into a layer of muscles and fat. It is the outer surface of this that is seen on the picture. The yellow arrows point to the pulmonary arteries. The blue ones point to a region containing the spine. Finally, the green arrows point to a region containing the ribs. The spine and the ribs are wrapped with a layer of muscles and fat similar to the one of the heart.

In order to segment the heart, one more segmentation step is needed. We chose the pos-

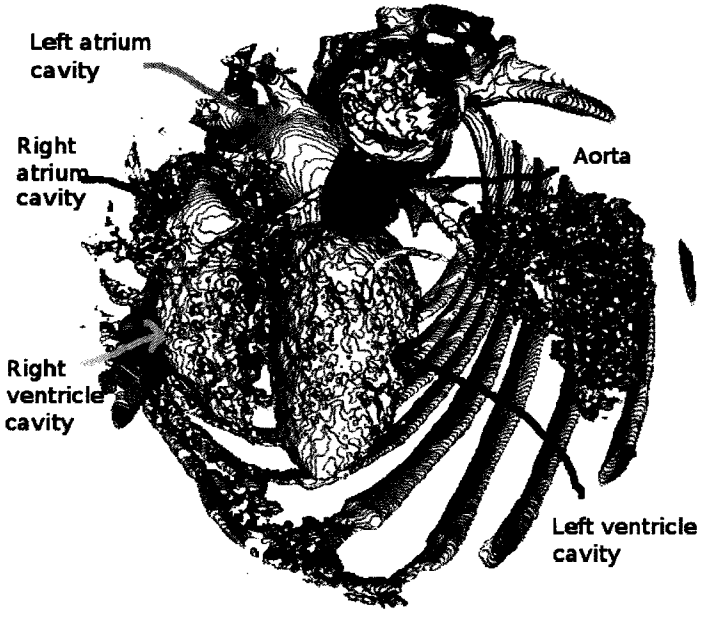
itive side of the previous segmentation. The result of the third step is shown in Figure 6.5. The resulting image u of the segmentation has 3 values: 0 for the region that is ignored from the previous segmentation steps and -1 and 1 for the two regions obtained from the third step. Figure 6.5a shows the level surface $\{u = 0.5\}$ recovering the endocardium while Figure 6.5b shows the level surface $\{u = -0.5\}$ recovering the epicardium.

(a) The result of the second step in white, together with the domain boundary in orange

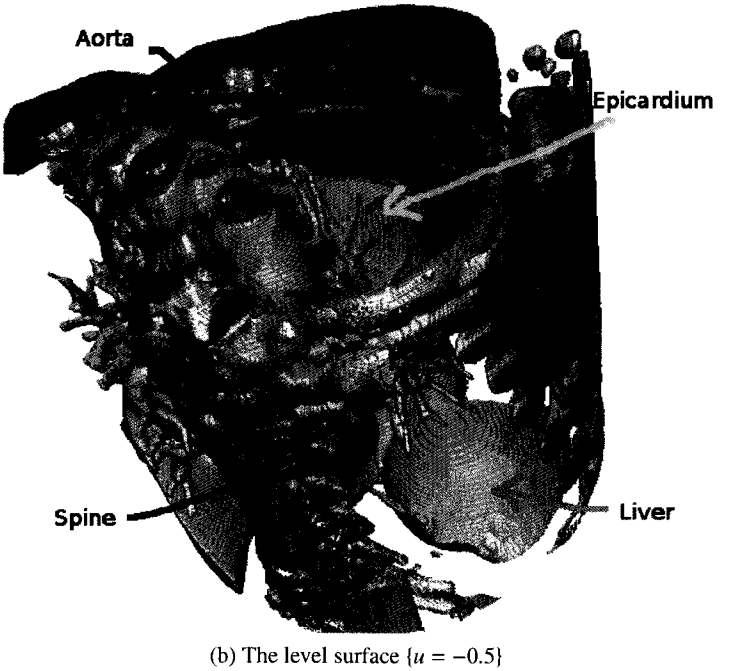


(b) (c)

Figure 6.4: The second step of the segmentation process. (a) shows the 0-level surface of the level set function together with the domain's boundary in orange. In (b) and (c) two views of the 0-level surface alone. The red arrows point on the region containing the heart, the yellow arrows point to the pulmonary arteries, the blue ones point the spine region and the green ones point the region containing the ribs.



(a) The level surface $\{u = 0.5\}$



(b) The level surface $\{u = -0.5\}$

Figure 6.5: The third and final step of the segmentation process.

The various computing time and parameters are gathered into Table 6.1. The behavior is similar to the 2D case studied in Section 5.4. The first step is fast to compute since the curvature term is null. The second and third steps take more iterations. In the first step, the average computing time per time step is 15s. At the second step, it is 8s. while for the last step, the average time is 6.5s. It is normal that this time is getting smaller and smaller since the computational domain also gets smaller and smaller.

Table 6.1: Computational times and parameters for the hierarchical segmentation of a 3D cardiac CT scan using L^1 fidelity. The times are those when the problem is solved in parallel on 6 processors.

Step	Time step	μ	λ	Nbr. of iterations	CPU time	Cumulative CPU time
1	0.1	0	1	11	2m46s.	2m46s.
2	0.1	1	1	666	93min17s.	96m03s.
3	0.1	1	1	1575	179m41s.	275m44s.

The total time for the segmentation is 275min 44s, that is about 4.5 hours. It is a fairly high computational time. Note that this can be considerably reduced if a smoother version of the image is used. As we mentioned earlier, we choose this level of smoothing to keep as many details as possible.

Figure 6.6 shows a slice of the original image with the segmented contour superposed. Cavity contours are very close to the image and contains many interesting details. The pulpillary muscles are well segmented. The exterior boundary of the heart is nicely segmented except in some regions where the curve is leaking. We address this problem in Chapter 7.

The segmentation result shown in Figure 6.5 contains the heart, but also many other details that are not relevant to heart segmentation. On Figure 6.5 are pointed the main anatomic details present in the segmentation. In Figure 6.5a, the heart cavities are shown and identified, as well as the aorta. In Figure 6.5b, the epicardium, the spine, the liver and the aorta are shown.

In order to isolate the heart muscle from this segmentation result, we manually pick a point in the heart muscle, then we take the connected component containing that point.

We have written an algorithm that extract the connected component containing a seeding voxel $x_0 = (i_0, j_0, k_0)$. The algorithm has the following structure.

1. Create an empty vector v. It is a workspace that will contain neighboring points. Insert the point x_0 in the vector.

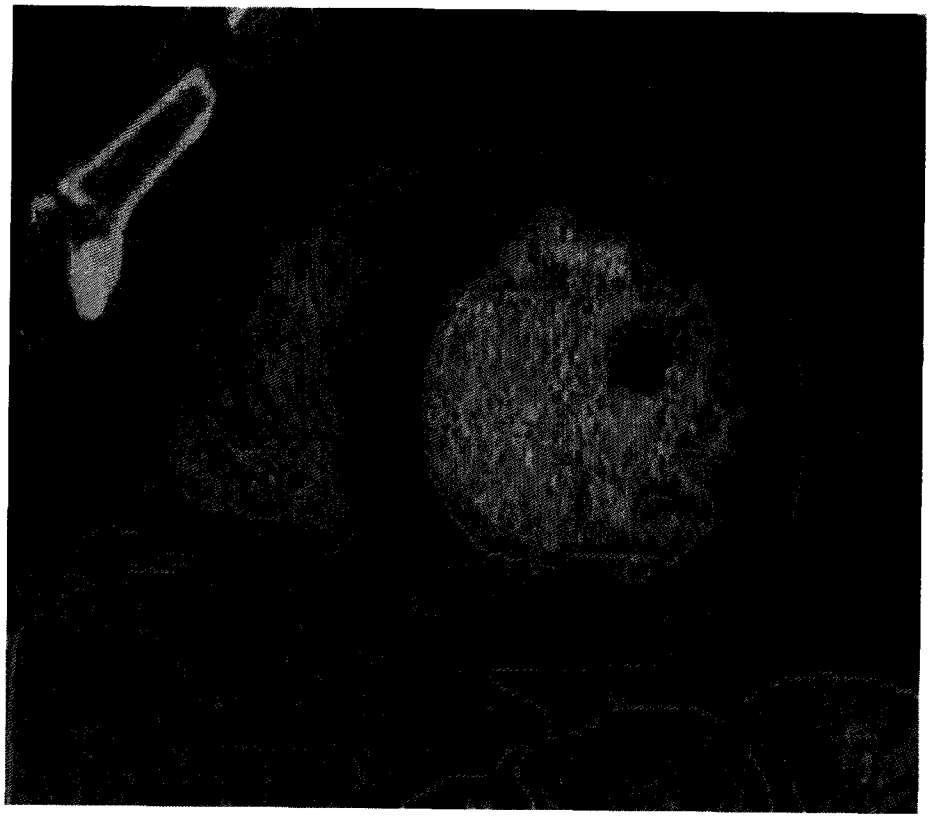


Figure 6.6: The segmented contour over a slice of the original image. The surface is leaking over the liver.

2. If v is empty or if the maximum number of pixels is reached, exit the algorithm. Otherwise, insert at the end of the vector v all neighbors of $v[0]$ in the same component as $v[0]$ that have never been visited. Note that pixels are in the same component if the level set function ϕ as the same sign.

3. Put the element $v[0]$ in a list containing the pixels in the same component as x_0.

4. Delete element $v[0]$ from the vector v and go back to step 2.

To extract the heart, we set a limit of 10 000 000 voxels on the size of the connected component. The output of this is shown in Figure 6.7a.

There are some imperfections in the segmentation of the heart surface near the epicardium, as the surface leaks in some areas. This is due to the fact that the heart touches the liver at that place in the image and that the two organs are of same gray level. This can be

seen in Figures 6.6 and 6.7b. This problem is addressed in Chapter 7.

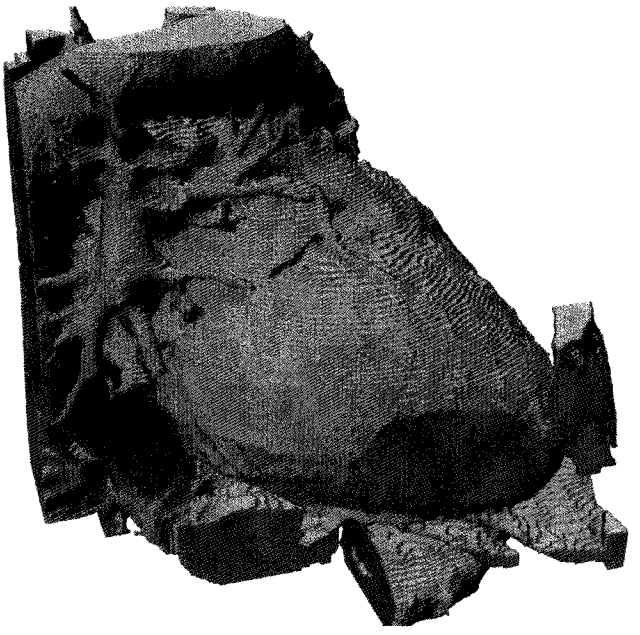
We can also use the algorithm for finding the connected components to extract the cavities of the heart. We picked a point inside the left ventricle cavity and then applied the algorithm, setting a size limit to 3, 500, 000 voxels on the size of the connected component. Figure 6.8 shows the segmented endocardial surfaces. The cavities of the ventricles and the atria are well segmented, especially the one of the left ventricle, which is the most important part in many applications. The occulus of the mitral valve that is between the left ventricle and the left atrium is also precisely captured as shown in Figure 6.9. The general shape of the heart is well represented, ventricles and atria are extracted as well as the aorta. The right ventricle cavity has some imperfections in the surface, but the pulpillary muscles are present in the segmentation.

From this segmentation, the ventricular and atrial cavities can be separated from the segmentation shown in Figure 6.8 by simple plane cutting. The isolated cavities are seen in Figure 6.10.

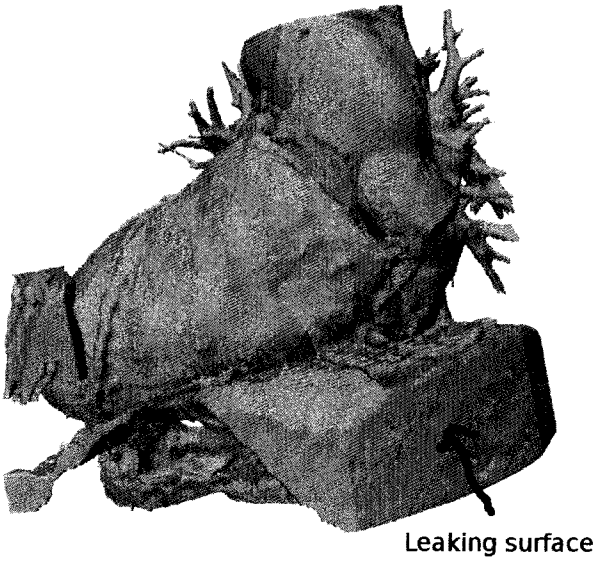
6.2.1. Trachea segmentation

At the first step of the hierarchical segmentation, we have chosen the region that was on one side of the surface in which the heart was located. The region that is on the other side of the surface contains the trachea. The trachea can be segmented by applying the hierarchical segmentation to that region. Only one step is needed to fully segment the trachea.

A high curvature term is needed in order for the curve not to leak in the small air channels of the lungs. As for the heart, we manually chose a point inside the trachea and extracted the full connected component of that point in the segmented image. that corresponds to the trachea. Figure 6.11 shows the resulting segmentation. This segmentation result will be meshed in Chapter 8.

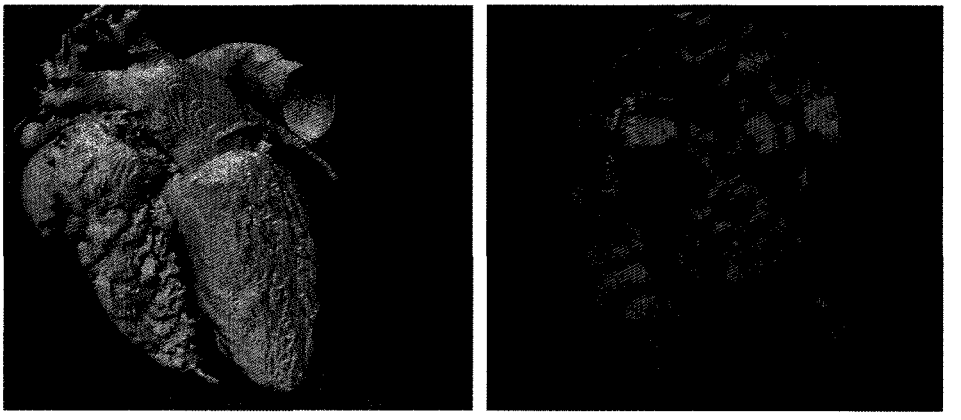


(a) A view of the epicardium



(b) A second view of the epicardium

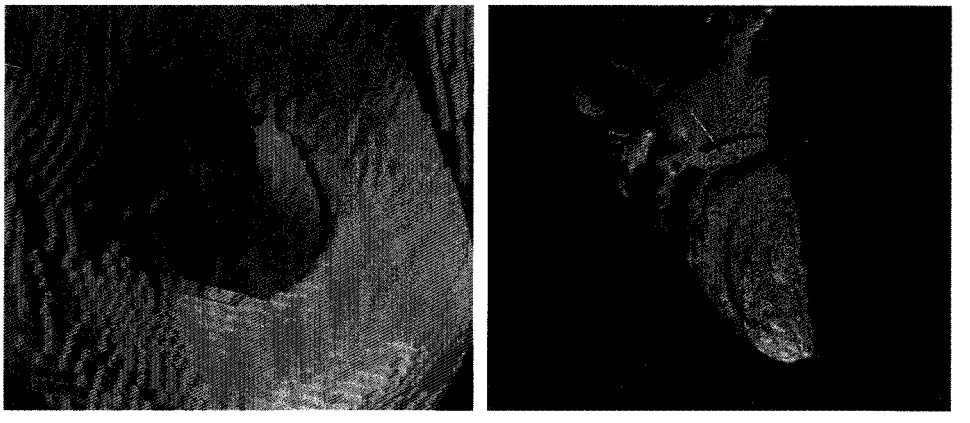
Figure 6.7: Views of the epicardium (outer surface of the heart). The surface is leaking in some regions.



(a) The heart cavities: ventricular and atrial cavities.

(b) The left ventricle cavity

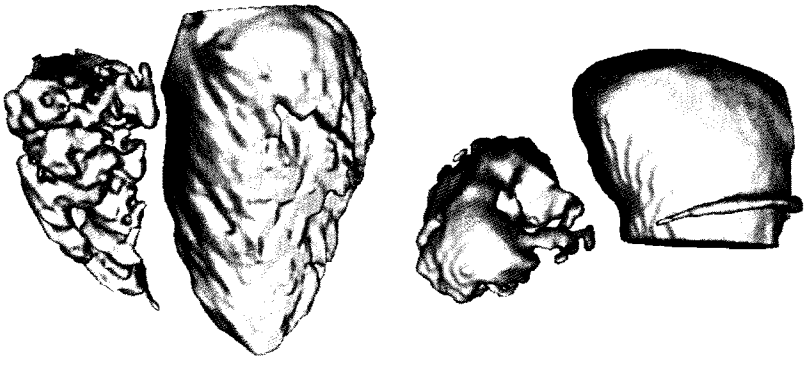
Figure 6.8: Two views of the interior surfaces of the heart in the final segmentation. The pillars in both ventricles are well segmented.



(a) A top view of the position of the mitral valve.

(b) A combined view of the left ventricle and the left atrium and the mitral valve occulus.

Figure 6.9: Two other views of the segmented endocardium.



(a) The isolated ventricular cavities. (b) The isolated atrial cavities.

Figure 6.10: Isolated cavities of the heart.

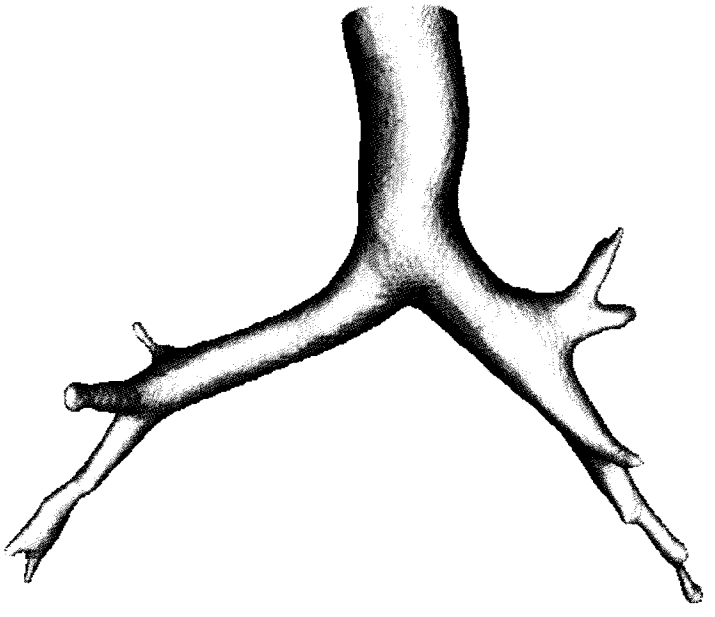


Figure 6.11: The trachea segmented from the 3D CT scan.

7

Epicardium segmentation

In this chapter, we present how a full heart segmentation is obtained from the cavity segmentation of Chapter 6. To do this, we use a problem similar to the one of subjective surfaces. The subjective surface problem presented in Section 4.3.4 requires long computational time to solve for large problems. To help with this, we introduce a new problem that has a smaller domain and boundary conditions adapted for heart segmentation. As in Chapter 6, the problem is solved in parallel.

This method yields a complete 3D segmentation of the heart's epicardial surfaces of both ventricles and atria. This segmentation is very close to the real heart geometry and the resulting geometries should be suitable to do realistic simulations of heart physiology.

The method outlined above, together with the segmentation result itself constitute the original content of this chapter.

7.1. CLASSICAL SUBJECTIVE SURFACES

In Section 4.3.4, we presented the subjective surface problem introduced by Sarti, Malladi and Sethian [98, 99]. It consists in finding a function ϕ that is a steady state of

$$\begin{cases} \phi_t = h|\nabla\phi|\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \nabla h \cdot \nabla\phi & \text{on } \Omega \\ \phi = 0 & \text{on } \partial\Omega, \\ \phi(x,0) = \phi_0. \end{cases} \tag{7.1}$$

In this problem, the initial condition is usually taken as $\phi_0 = \frac{1}{1+D(x)}$, where $D(x)$ is the distance from a point p chosen inside the object of interest. Hence, the function ϕ_0 has highest value 1 at p and decreases quickly to 0. It is also possible to take $D(x)$ to be the distance to a set of points, or the distance to another shape such as a line or a curve. h is an edge stopping function as in Equation 4.6.

The partial differential equation in Problem 7.1 can be solved using a finite difference method. The diffusion term can be discretized as for the total variation problem (see Section 3.4) or the Chan-Vese problem (see Section 5.1). The advection term needs to be discretized using an upwind scheme. The upwind scheme takes an approximation

$$\nabla^{uw}\phi = (D_x^{uw}\phi, D_y^{uw}\phi)$$

of the gradient $\nabla\phi$ using upstream values. In the advection term, the speed is $b = -\nabla h$, hence

$$D_x^{uw}\phi = \begin{cases} D_x^+\phi & \text{if } D_x^0 h \geq 0, \\ D_x^-\phi & \text{if } D_x^0 h < 0. \end{cases} \tag{7.2}$$

Similarly

$$D_y^{uw}\phi = \begin{cases} D_y^+\phi & \text{if } D_y^0 h \geq 0, \\ D_y^-\phi & \text{if } D_y^0 h < 0. \end{cases} \tag{7.3}$$

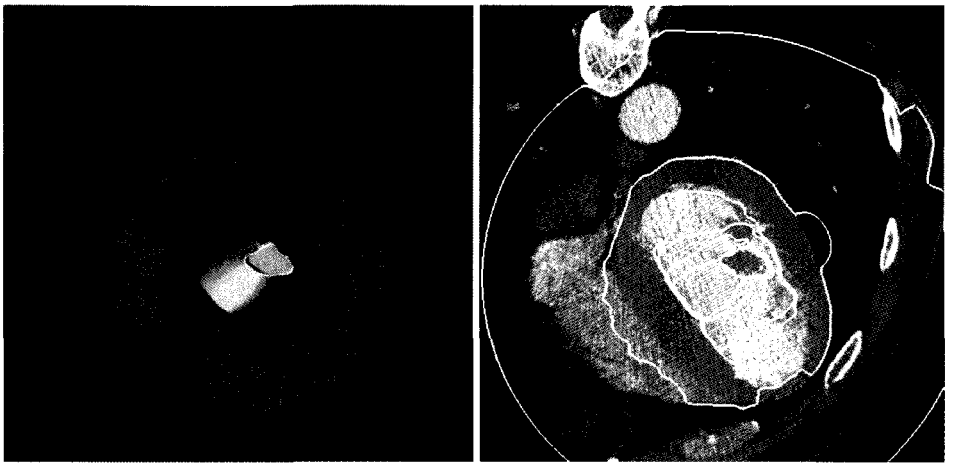
We intend to solve the problem with an explicit time stepping scheme. Without using the upwind scheme, it is not possible to achieve stability even if very small time steps are used (e.g. $\Delta t = 10^{-8}$). Even though the upwind scheme is used, very small time steps still need to be used. Therefore, the curvature term need not be discretized using a fancy scheme. It is more efficient to use the direct discretization described in 5.1, since other schemes double the iteration time.

Figure 7.1 shows the result of the algorithm on a 2D slice of a 3D CT scan. This 2D case study is done on the same slice as the hierarchical segmentation of Section 5.3. A point p inside the left ventricle has been chosen and the initial condition is $\phi_0 = \frac{1}{1+D(x)}$, where $D(x)$ is the distance to that point.

One can see that there is no level curves that can successfully segment the epicardial surface. Moreover, it takes a very large number of time steps, around 100,000 in order to reach this state, which may not yet bet the steady state. The computing time for these iterations is about 33 min 49s, which is very long compared for example to the Chan-Vese method.

7.2. PROPOSED PROBLEM

In order to obtain a better segmentation of the epicardial surface than the one obtained in Section 7.1, we will use the information about the heart's cavities obtained in Chapter 6. In Chapter 6, the endocardial surface of the ventricles have been successfully segmented. The



(a) The steady state of Problem 7.1.

(b) The level curves of the steady state on top of the image.

Figure 7.1: The result of the classical subjective surface problem applied to a 2D slice of a CT scan.

epicardial and endocardial surfaces are somewhat related one to the other and one might think that it would be useful to use the endocardial surface position to infer the epicardial surface.

The idea is that the endocardial surface will be used to restrict the computational domain and will have adequate boundary conditions that help avoiding leakage of the level surfaces.

The main modification we bring to the original subjective problem (Problem 7.1) is to restrict the computational domain. The position and shape of the endocardial surface is known. The epicardial surface has to sit outside the endocardial surface. Also, it cannot be too far away from the endocardial surface as the myocardium has a maximal thickness.

The idea is then to do the computation on points that are outside the endocardial surface and at a distance less than a given number L. For example L could be set to be twice the estimated myocardium wall thickness. The domain is then much smaller than the original domain. We will also modify boundary conditions. We force the function to be 1 at the endocardial surface S and be 0 at the surface S_L that is at a distance L from the endocardial surface. If Ω_L stands for the region between these two surfaces, then the proposed problem

is to find a steady state ϕ of the problem

$$\begin{cases} \phi_t = h|\nabla\phi|\mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \nabla h \cdot \nabla\phi & \text{on } \Omega_L \\ \phi = 1 & \text{on } S, \\ \phi = 0 & \text{on } S_L, \\ \phi(x,0) = \phi_0. \end{cases} \tag{7.4}$$

In practice, we get the surface S_L by computing the signed distance function ψ to the endocardial surface. This is done with the help of the reinitializing equation (Equation 8.3) that will be further discussed in Section 8.2. Then, the domain Ω_L consists of voxels x_i for which $0 \leq \psi(x_i) \leq L$. Figure 7.2 shows an example of the resulting domain.

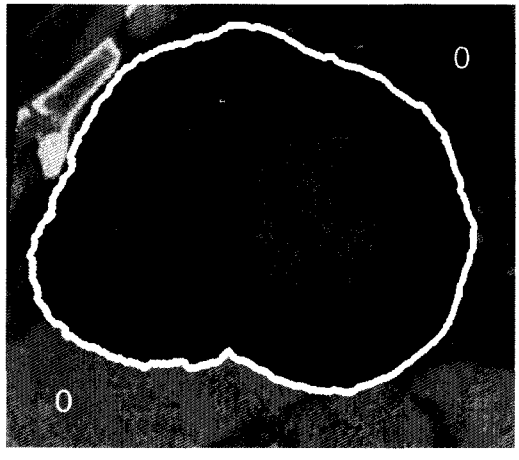


Figure 7.2: The computational domain Ω_L with boundary conditions.

As an example, consider the application to the 3D CT scan presented in Section 6.2. The segmented endocardial surfaces of the ventricles are shown in figure 6.10a. The full computational domain has 52, 166, 656 voxels while a valid restricted domain can contain as few as 4, 415, 284 voxels, which corresponds to 8.4% of the original size. This is a considerable gain.

Using the Problem 7.4 yields more than just a reduced domain size: the solution contours must remain between the surfaces S and S_L. Hence there will be less leaking of the level set surfaces. The problem is then solved in parallel using the same framework as the one described in Section 6.1.

7.3. THE RESULTS

We used the new subjective surface problem given in equation 7.4 for two different applications. First, it has been applied for ventricle segmentation, then for atria segmentation.

7.3.1. Ventricle segmentation

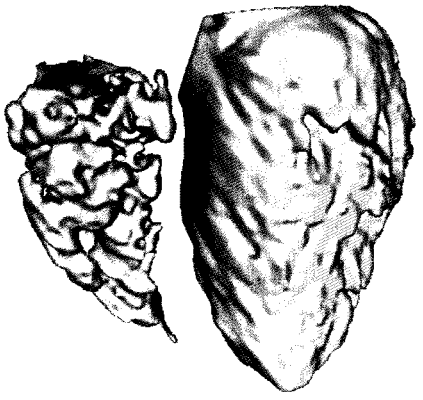
We started with the segmentation results obtained in section 6.2, that are shown in Figure 6.2a. The computations are made on a restricted domain of $4,415,284$ voxels. The equation is solved explicitly with a time step $\Delta t = 0.01$. This is done in parallel on 6 processors for $1,000,000$ iterations. The total computation time is 168 hours, which is close to 7 days. The final result is a function that is between 0 and 1. A level surface needs to be chosen that will yield the epicardial surface. As the steady state should be close to a piecewise constant function, there should be a large interval of values that would give the same level surface.

Figure 7.3 shows the results of the segmentation when choosing the surface of level 0.5 from the steady state. Figure 7.3c and 7.3d show front and back views of the surfaces on top of a 2D slice of the CT image. As it can be seen there, the myocardium is very thin in the right ventricle. In general, this is a major problem for segmenting the whole heart and for this reason, all right ventricle segmentation of which we are aware only segment the epicardial surface of the right ventricle. The modified subjective surface problem used here forces the myocardium to have some width everywhere. Indeed, the heart chambers have fixed value 0 and the epicardium can be chosen as the level surface 0.5 for example.

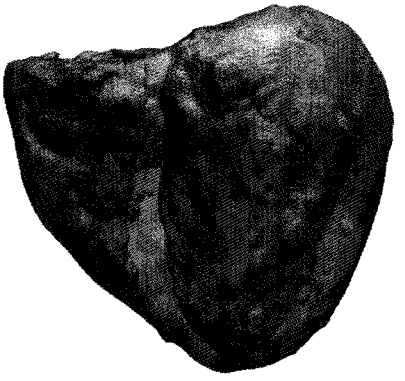
Figure 7.4a shows a 2D slice of the CT image with the segmented epicardial surface restricted to that slice on top. As can be seen, the contour is very close to the epicardium in the CT image. Again, it can be seen in Figure 7.4a how thin is the myocardium in the right ventricle region. Figure 7.4b shows a relief map of the steady state function restricted to the same 2D slice. The function is almost piecewise constant, hence many level surfaces generate the same epicardium surface. This also indicates that the final level set function is close to being the steady state.

With this new subjective surface problem, we have been able to get an accurate segmentation of the epicardial surface. This can be seen by restricting the segmentation to 2D slices as in Figures 7.4a, 7.3c, d and in Figure 7.5. In each of these figures, the segmented epicardial contour follows closely the epicardial contour that a human eye distinguishes in the image.

Now, this result can be combined with the segmented endocardial surface to obtain a

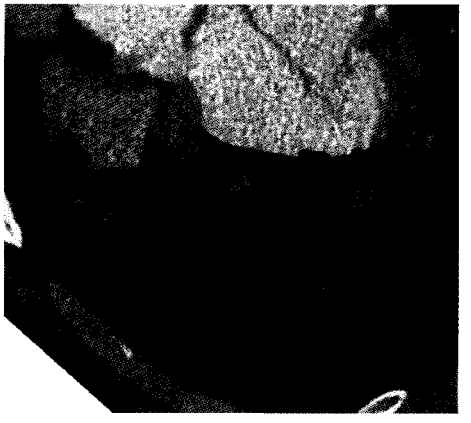


(a) The endocardial surface of the ventricles.



(b) The epicardial surface.

(c) Back view of the endocardium and epicardium on a 2D slice of the CT image

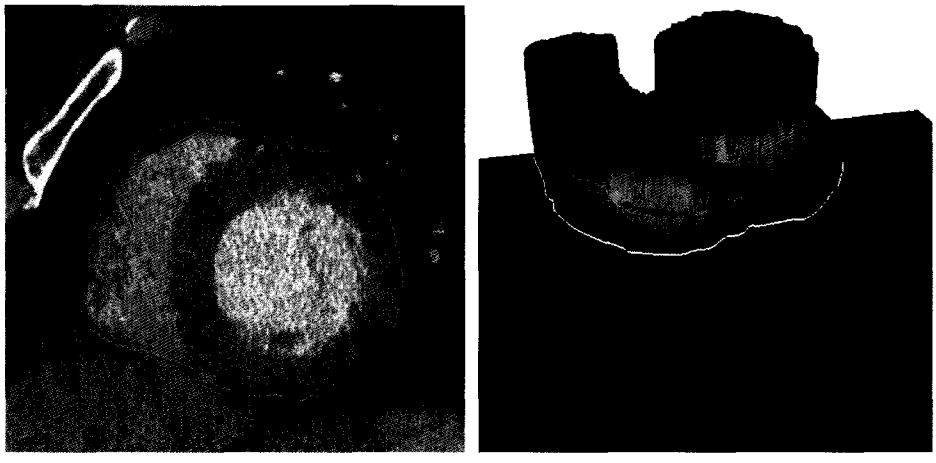


(d) Front view of the endocardium and epicardium on a 2D slice of the CT image

Figure 7.3: The result of the myocardium segmentation.

complete segmentation of the myocardium walls. As remarked previously in Section 6.2, the endocardial surface of the left ventricle is nicely segmented, it is the right ventricle that suffers from some leaking.

To improve even further this segmentation, we will use the fact that we now have a precise segmentation of the epicardial surface. We can then use a hierarchical Chan-Vese algorithm as described in Chapter 6 on the region inside the epicardial surface. This region now contains only the myocardium and the heart chambers. Applying the two phases Chan-Vese method on that domain, yields a better segmentation of the endocardial surface in the right ventricle.



(a) (b)

Figure 7.4: The final epicardium segmentation restricted to a 2D slice of the CT image. (a) The epicardial contour superposed on a slice of the original CT image. (b) A relief map of the steady state solution restricted to the same slice with the chosen contour.

The combined result of the epicardium segmentation using subjective surfaces and the new endocardium segmentation of the endocardium using Chan-Vese gives a full myocardium segmentation. In order to appreciate the quality of the segmentation, we show in Figure 7.5 the segmented myocardium contours on 2D slices of the CT image. It can be seen that the contours closely follow the real myocardium contours.

Figure 7.6 shows several view of the complete segmented myocardium. Figure 7.6d is an explanatory diagram of the four visible orifices. Note that the orifice of the aorta and the one between the left atria and the left ventricle are not completely disjoint.

(a) (b)

Figure 7.5: The final myocardium segmentation restricted to 2D slices. (a) A frontal view of the final segmentation. (b) A top view of the final segmentation.

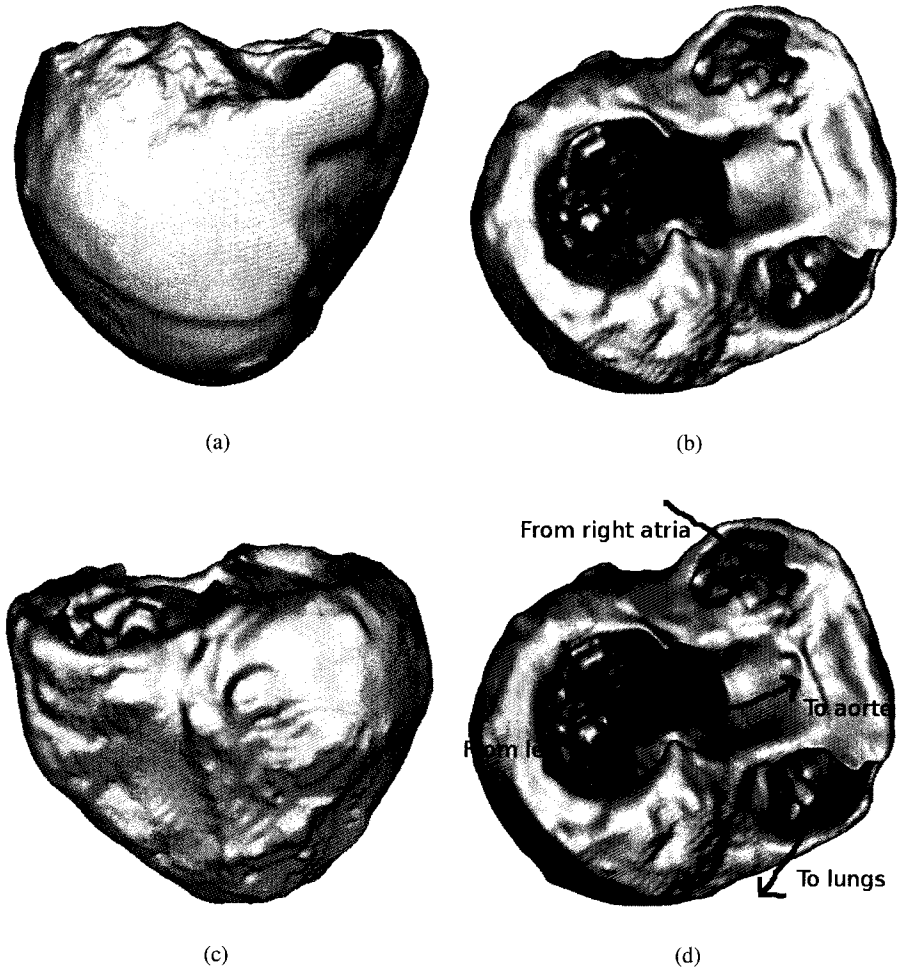


Figure 7.6: The final myocardium segmentation. (a) A front view of the final segmentation. (b) A top view of the final segmentation. (c) A back view of the final segmentation. (d) Explanatory diagram of the orifices.

7.3.2. Atria segmentation

The same subjective surface method can be applied to the segmented atria cavities shown in Figure 6.2b. Again, the domain is restricted and the equation is solved explicitly with a time step $\Delta t = 0.01$. It is solved in parallel using 6 processors for 1 000 000 iterations. The total computation time is 3 598 min, that is 2.5 days.

As in Section 7.3.1, the segmented outer surface can be joined with the cavities. We did not use the Chan-Vese algorithm in this case to improve the cavities. We simply smoothed the inner surfaces with a total variation algorithm as described in Section 3.4.

For the CT scan used for this segmentation, a tracer has been used to highlight the pulmonary arteries. This can be seen as the lightest region in 2D slices shown in Figure 7.8. This presence of the tracer complicates the right atrium segmentation. Using the segmented epicardial surface and the smoothed endocardial surface yields a 3D segmentation where some walls are very thin. These are too thin to be meshed accurately. For this reason, we introduced an artificial width between these surfaces. This is easily done using the signed distance functions to both surfaces and choosing surfaces at distance 1 away from the segmented surfaces. It is about twice the smallest width in the initial segmentation.

Figure 7.7 shows the resulting atria cavities together with the epicardial surface. Annotations have been added to distinguish between the left and the right ventricle.

Figure 7.8 shows the atria endocardial and epicardial contours on 2D slices of the CT image. The effect of the artificial width can easily be seen: the contours are somewhat shifted from the original image contours. Nevertheless, the overall shape of both endocardium and epicardium is well captured.

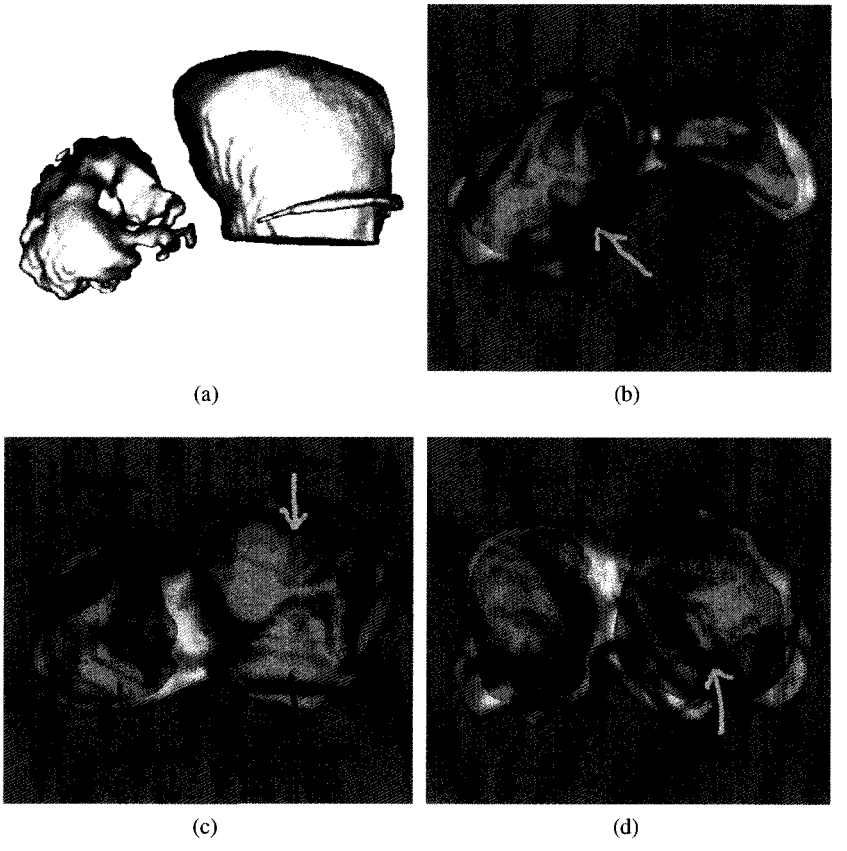


Figure 7.7: The result of the atria segmentation with annotations. (a) The endocardial surface of the atria. (b) A top view of the segmented atria. (c) A front view of the segmented atria. (d) A back view of the segmented atria.

(a) (b)

Figure 7.8: The final segmentation on 2D slices. The contours are close to the real endocardial and epicardial surfaces, although they are somewhat shifted due to the smoothing step and to the artificial width introduced. (a) A top view of the final segmentation. (b) A front view of the final segmentation.

8

Mesh generation

The results of the segmentation methods described in chapters 6 and 7 are given as a level set function whose 0-level set is the contour of the object of interest. The ultimate goal of this work is to be able to perform numerical simulations on segmented organ geometries. A standard finite element method requires a mesh of the given geometry. In this chapter, we present how a volumetric 3D mesh of the heart is generated from the segmentation results of Chapters 6 and 7.

8.1. MESH GENERATORS

There exists plenty of 2D and 3D mesh generators that are publicly available. We have been looking for software that can mesh geometries that are defined implicitly. We also want to produce meshes that are 2D and 3D. There are not many meshers that satisfy these two criteria. Besides, most mesh generators involve using complicated libraries. Robert Schneiders keeps up to date a list of all commercial and non commercial mesh generators[1]. Among those, there is a small function set called DistMesh that captured our attention.

8.2. DISTMESH AND ITS MODIFICATIONS

DistMesh is a set of Matlab functions written by Persson [85, 47]. It is very simple to use and very efficient compared to many mesh generators. It is designed to produce meshes of domains that are defined explicitly via a level set function ϕ. It is based on the Delauney criterion to assess mesh quality and it considers edges as a network of springs with properly chosen forces to obtain mesh uniformity. The novelty of the algorithm is in the way domain boundaries are handled. Points close to the boundary are projected on it, in order to a obtain a nice smooth boundary. Another interesting feature of this code is that it works in 2D and 3D, with some minor changes. The general structure of the mesh generator is as follows.

[1] http://www-users.informatik.rwth-aachen.de/ roberts/

1. The user creates a length function that specify at each point in space the desired diameter of an element. It can be uniform or not.

2. A random initial vertex distribution is created, according to the length function.

3. Vertices outside the domain are rejected.

4. Delauney triangulation is calculated with the given vertices.

5. Forces related to the length function are calculated for each edge and vertices move according to the resulting force.

6. Points near the boundary are projected on the boundary with the help of the level set function.

7. Steps 4-6 are repeated until the mesh is of sufficient quality.

It is in step 6 that the user has some freedom. The distance between a vertex and the boundary is simply given by the value of the level set function at this vertex. The level set function is usually numerically given by values on the underlying image grid, hence it has to be interpolated to get the value at a specific vertex. Next, one must define a notion of proximity to the boundary. In the original code, it is suggested to project only the vertices that lie outside the domain. However, we remarked that the boundary was better preserved when some points inside the domain were also projected on the boundary. This had to be tuned up in each case, but in general we projected vertices at a distance less than $h/2$, where h stands for the element diameter in that region.

The code as it is only meshes the region where the level-set function is negative. In some applications, we had to mesh a domain, that contains sub-domains whose interfaces also need to be respected by elements. This is the case of the example of a 2D torso mesh (see Figure 8.1) where there are 4 sub-domains: the myocardium, the heart cavities, the lungs and the remaining of the torso. In this case, we calculated four different level set functions, one for each of the sub-domains. Then we had only to modify step 6 of the algorithm, to project on the boundaries of all sub-domains, using the same technique.

The projection method of step 6 is simple. If a vertex v is such that $\phi(v) > -c$ for some small positive value c, then we project the node v on the point $T(v)$ on the interface and given by

$$T(v) = v + \phi(v)\nabla\phi(v). \tag{8.1}$$

This works if ϕ is a signed distance function, which means $|\nabla\phi| = 1$ everywhere. However, we happened to have some problems since in some cases the function ϕ is just approximately a signed distance function. In those cases, the projection should be

$$T(v) = v + \phi(v)\frac{\nabla\phi(v)}{|\nabla\phi(v)|}, \tag{8.2}$$

by taking a special care that the numerical gradient is never zero. This is a modification that we made to the original code. These kinds of problems arose when we generated signed distance functions with the help of the `itk` library [126] [2]. For this reason, we preferred to use the reinitialization equation in order to get signed distance functions. It consists in solving

$$\begin{cases} \phi_t = S(\phi_0)(1 - |\nabla\phi|), \\ \phi(0,\cdot) = \phi_0, \\ \frac{\partial\phi}{\partial n} = 0 \qquad\qquad \text{on } \partial\Omega, \end{cases} \tag{8.3}$$

where $S(x)$ is the sign function as suggested in [109]. In practice a regularization $\bar{S}$ of S is used:

$$\bar{S}(\phi) = \frac{\phi}{\sqrt{\phi^2 + (\Delta x)^2}}. \tag{8.4}$$

Peng et al. [82] also suggested to take

$$\tilde{S}(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2(\Delta x)^2}}. \tag{8.5}$$

In our applications, we did not remark significant differences between the regularizations $\bar{S}$ and $\tilde{S}$.

The term $|\nabla\phi|$ needs to be discretized using an upwind scheme. The flow always goes away from the zero 0 level set. In consequence, the pixels inside the curve ($\{\phi < 0\}$) will be treated differently that the pixels outside the curve ($\{\phi \geq 0\}$). The simplest stable scheme is an upwind scheme [76]. At a pixel $x_{i,j}$ inside the curve, the norm of the gradient can be approximated by

$$|\nabla\phi(x_{i,j}| \approx \sqrt{\min((D_x^-\phi)_{i,j}, 0)^2 + \max((D_x^+\phi)_{i,j}, 0)^2 + \min((D_y^-\phi)_{i,j}, 0)^2 + \max((D_y^+\phi)_{i,j}, 0)^2}. \tag{8.6}$$

[2] `SignedDanielssonDistanceImageFilter` [49]

Similarly, for a pixel $x_{i,j}$ outside the curve,

$$|\nabla\phi(x_{i,j}| \approx \sqrt{\max((D_x^-\phi)_{i,j},0)^2 + \min((D_x^+\phi)_{i,j},0)^2 + \max((D_y^-\phi)_{i,j},0)^2 + \min((D_y^+\phi)_{i,j},0)^2}. \tag{8.7}$$

Doing so, the equation can be solve with a Euler explicit scheme with a time step as big as half the size of the spatial step size.

There may be one drawback for using Distmesh in 3D applications as Delaunay criteria in 3D does not prevent the creation of slivers (tetrahedra with null volume and nearly equal edge lengths, see for example [40]). We have overcome this problem by smoothing the resulting 3D meshes with the software Stellar [54].

8.3. APPLICATIONS FOR MESHING

8.3.1. 2D heart and thorax

In section 5.4, we obtained a segmentation of the heart muscle from a 2D slice of a CT scan. The goal is now to generate a mesh that is suitable for 2D numerical simulations of electrocardiograms.

For this application, we added the segmentation of the lunges and of the thorax boundary. For each of these sub-domains, a signed distance function has been computed using the reinitialization equation. The whole thorax region is meshed, but we forced the mesh to respect the boundaries of the lunges, the heart, and heart chambers.

We defined a mesh size function that is dictated by the application. For electrophysiology simulations, a very fine mesh is required in the heart in order to get realistic transmembrane wave propagation. In the lunges and thorax, the mesh may be much coarser. The mesh size function varies continuously – using the distance function to the heart muscle – from a very small value in the heart muscle to the larger size in the thorax. We also remarked that the sub-domain interfaces are better preserved if the mesh size is decreased by a factor of 2 while approaching them. We have done this for every sub-domain using their signed distance functions. Figure 8.1 shows the result of the meshing process. This mesh has 82 000 vertices and 163 000 triangles. It has been successfully used for simulations in electrophysiology [29, 90].

8.3.2. Trachea

As mentioned earlier, one of the great advantage of DistMesh is that it works in 3D as easily as in 2D. In Section 6.2, we obtained a 3D segmentation of the trachea. In order to create

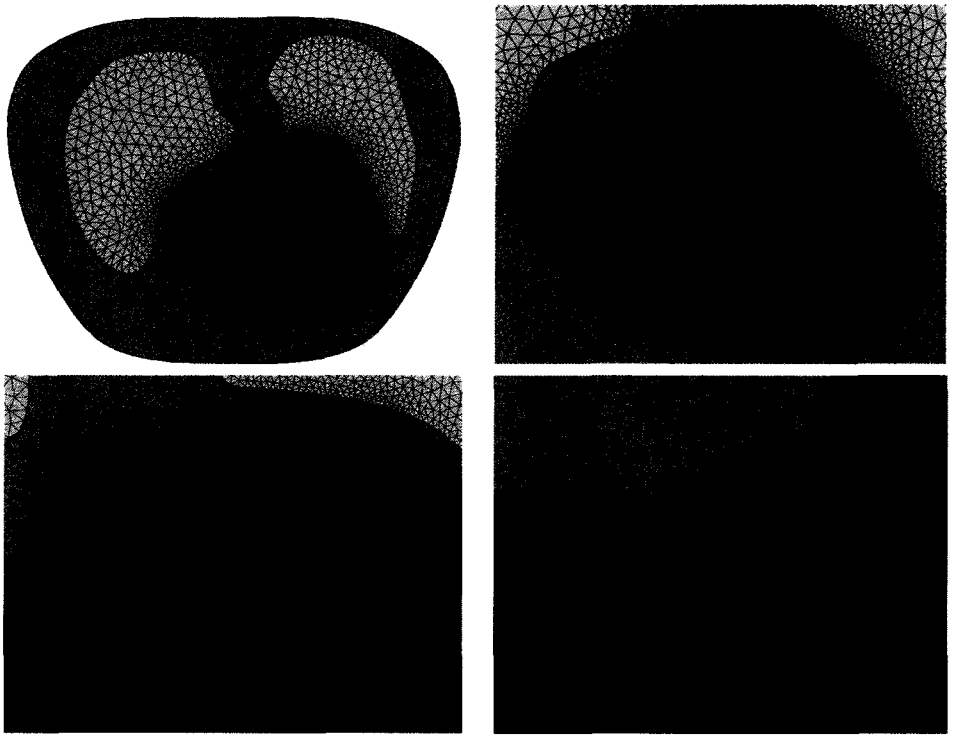


Figure 8.1: Different views of a 2D mesh of the torso. Four sub-domains are represented: the myocardium, the heart cavities, the lungs, the remaining of the torso.

a 3D mesh, we computed the signed distance function to the trachea. Figure 8.2 shows a coarse 3D mesh of the trachea. It contains about 3 400 vertices and 16 000 tetrahedra. In Figures 8.3 are two different views of a finer mesh containing around 52 000 nodes and 290 000 cells.

8.3.3. Carotid

We present here an application to the carotid. We segmented the carotid from a MRI image provided by André Garon[3]. The segmentation was done using a mix of simple segmentation methods such as threshold and classical snakes for which we do not give details. The segmentation has then been smoothed by the total variation equation (equation (3.15)). The result of this is displayed in Figure 8.4. Looking at this segmented carotid, we had a surprise as it has a non standard configuration. On a normal patient, there are three main

[3]from École Polytechnique de Montréal

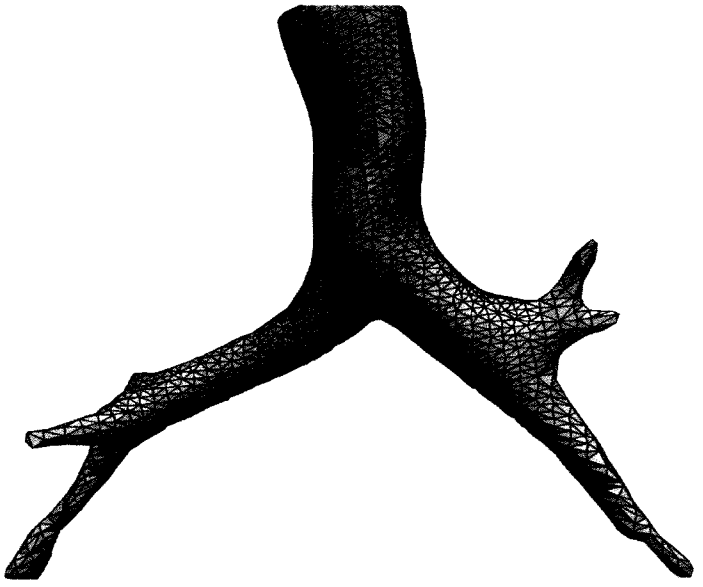
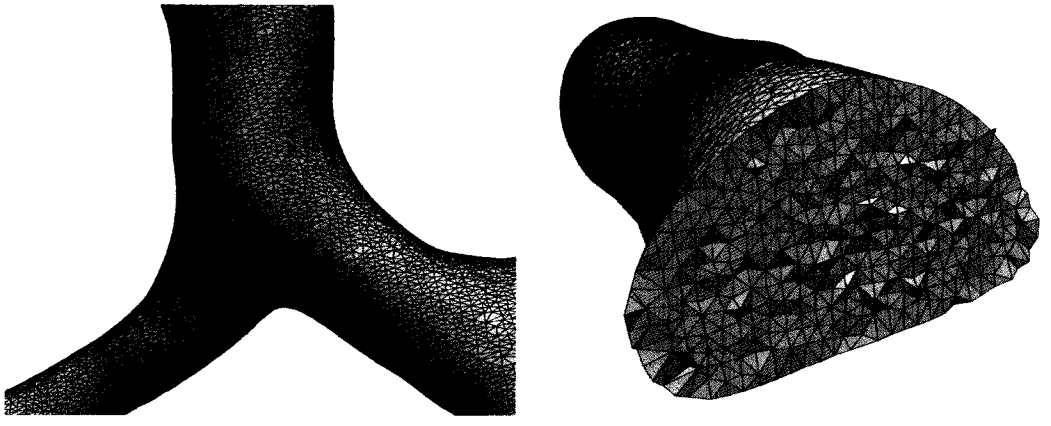


Figure 8.2: A coarse 3D mesh of the trachea: 3 400 vertices and 16 000 tetrahedra.



(a) (b)

Figure 8.3: A fine 3D mesh of the trachea: 52 000 vertices and 290 000 tetrahedra. (a) A view of the mesh skin. (b) a view on interior cells.

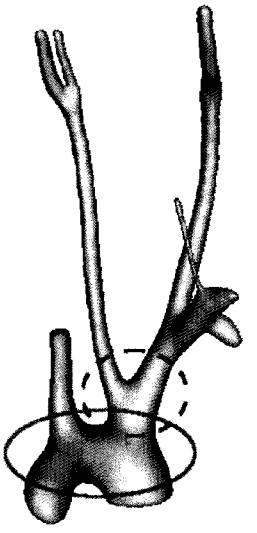


Figure 8.4: The segmented carotid. There are only two junctions in the plain circle instead of the three usually seen on a normal patient. The missing junction is lumped in the dashed circle.

artery junctions with the aortic arch, the region in the plain circle in Figure 8.4. On this patient, there are only two junctions as two of the junctions are lumped together further up in the dash-circled region.

We computed the signed distance function to this carotid and generated 3D meshes of it. The meshes are intended to serve for 3D blood flow simulations. Accordingly we defined the mesh size function to have smaller elements close to the boundary to capture the boundary layer of the blood flow resulting from the no-slip velocity on the arterial walls. The mesh is cut at the end of each vessel to obtain in-flow and out-flow boundaries that are perpendicular to the artery's central axis. Figure 8.5 encompass several views of the resulting mesh. The colors represent the different labels put on the nodes. There is one label per entrance/exit region and per entrance/exit region's boundary (a curve). An extra label is given for points on the rest of the surface.

8.3.4. 3D heart

In Section 7.3.1, we obtained a full segmentation of the myocardium. Again, the signed distance function to the interface can be calculated using the reinitialization equation. With this distance function, we created meshes of the heart muscle. These meshes are intended

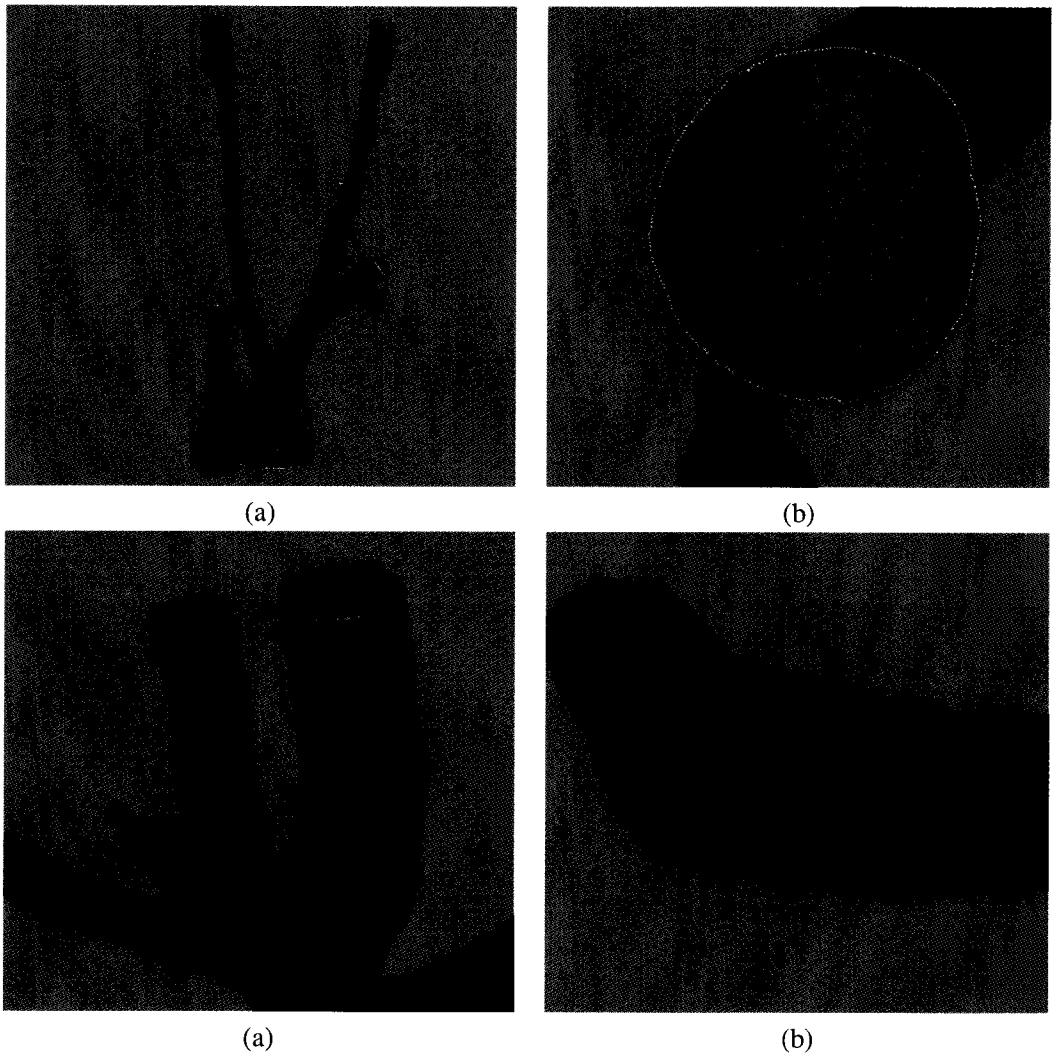


Figure 8.5: A fine 3D mesh of the carotid: 120 000 nodes and 540 000 cells.

to serve for electrophysiology simulations, hence we chose a uniform mesh size function. We only decrease the cell size by a factor of two while approaching boundaries. As we remarked earlier, this helps having a smooth boundary. Figures 8.6 shows a mesh obtained in this way. It has about 247 000 vertices and 1 256 000 cells. Coarser or finer meshes can easily be generated using the same code. In Figure 8.7 views of the skin of the mesh are displayed, of the interior of the right ventricle on top, and of the interior of the left ventricle below.

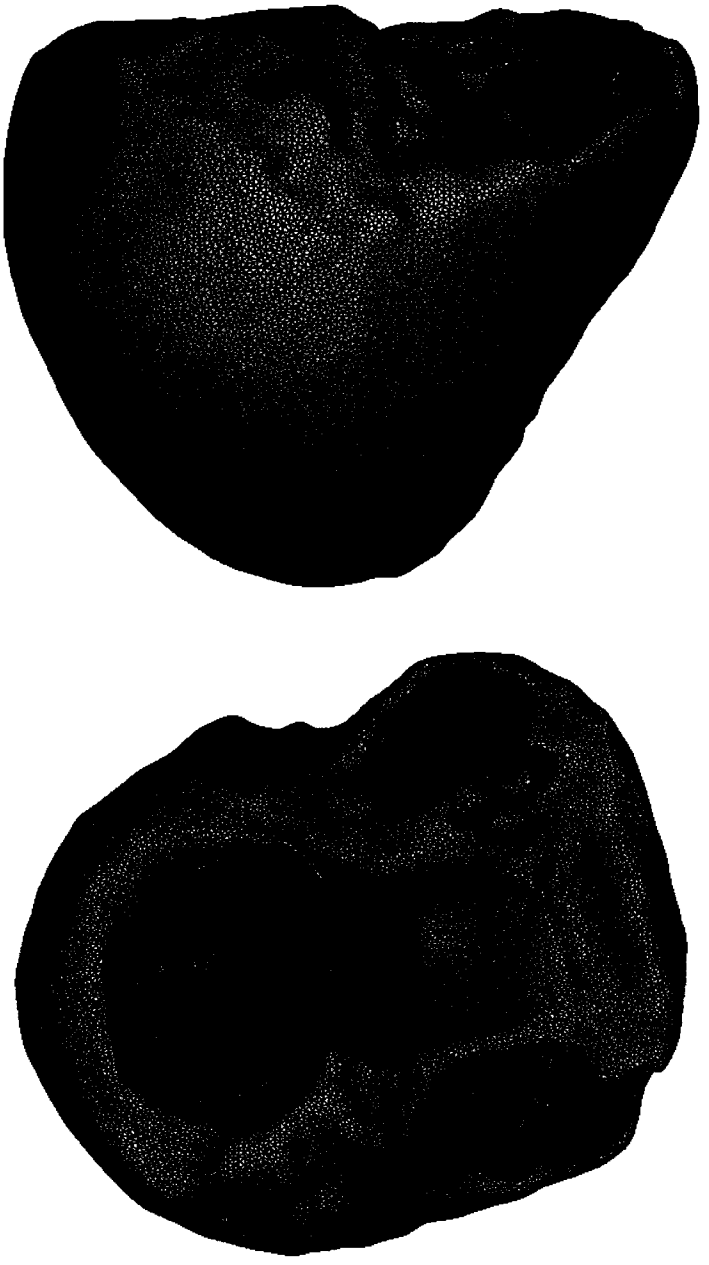


Figure 8.6: Different views of a 3D mesh of the heart containing 247 000 vertices et 1 256 000 cells.

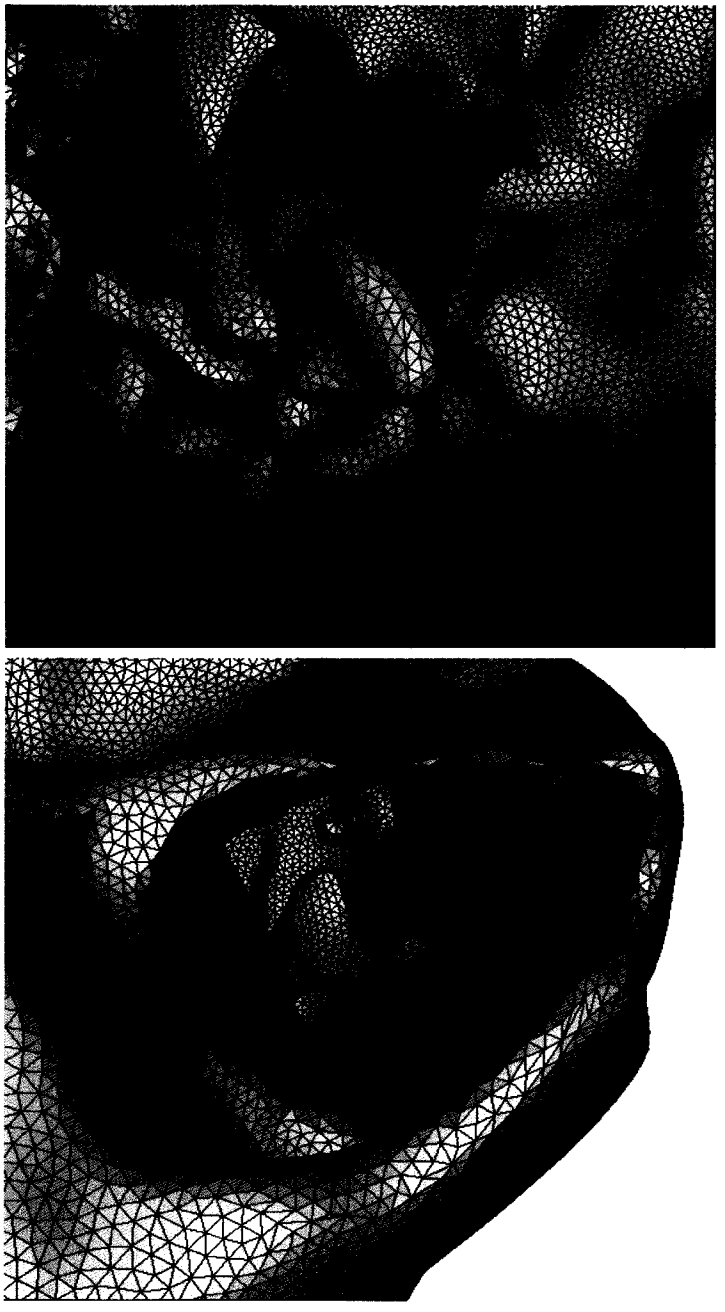
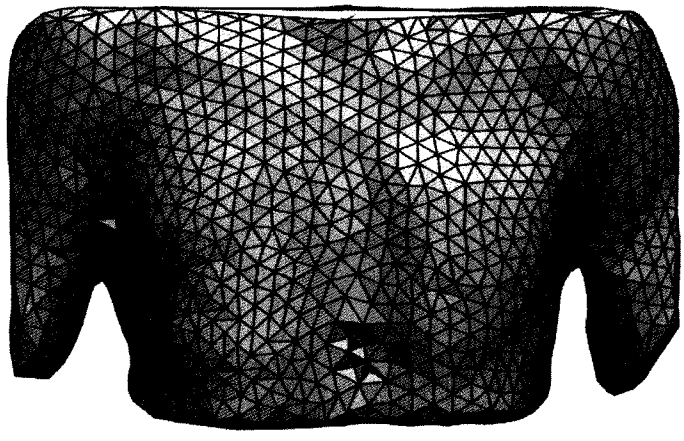


Figure 8.7: Inside views of a 3D mesh of the heart containing 247 000 vertices et 1 256 000 cells. The top view image is looking at the inside of the right ventricle while the bottom one shows the inside of the left ventricle.

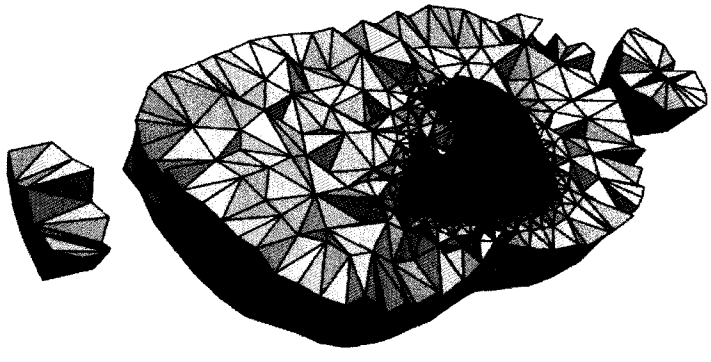
8.3.5. 3D heart and torso

In order to do 3D numerical simulations of electrocardiograms, the torso need to be meshed together with the heart. The CT image from which we segmented the heart does not contain the whole thorax shape. Therefrom, we chose to segment a torso from a lower resolution MRI image of a different patient. The heart as been placed manually inside the segmented torso.

The mesh is created using the signed distance to the torso to define the domain. The signed distance function to the heart is used so that the resulting mesh follows the heart's surfaces. Figure 8.8 and 8.9 show an example of a mesh created in this way. It contains in total 288 000 vertices and 1 724 000 tetrahedra. The sub-region corresponding to the myocardium has 267 000 vertices and 1 389 000 tetrahedra.



(a)



(b)

Figure 8.8: Different views of a 3D mesh of the torso and heart containing 288 000 vertices et 1 724 000 cells. (a) An outside view. (b) Inside view with the heart.

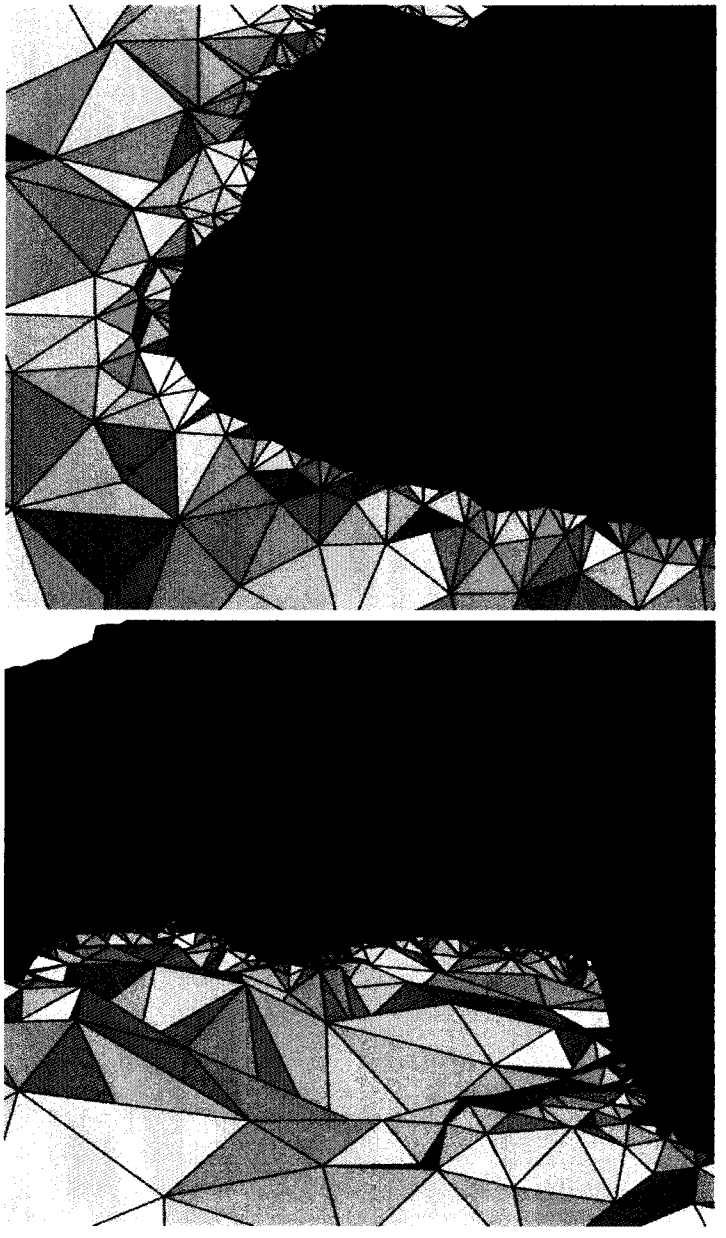


Figure 8.9: Two closeup views of a 3D mesh of the torso and heart containing 288 000 vertices et 1 724 000 cells.

9

Fiber Orientation

In Section 7.3, we obtained a precise and realistic segmentation of the human heart from a CT image. In Section 8.3.4, we presented how the segmented geometry can be meshed. The last step towards a model that can be used for realistic numerical simulations of heart's physiology consists of adding the fiber orientation information to the model. As we described in the introduction, the fibers are very important in the trans-membrane potential wave propagation as well as in the heart's contraction.

The heart muscle is structured in sheets that are wrapped around to form the heart. The sheets are attached one to another by a small layer of collagen. Each sheet is made of aligned fibers. At any point in the heart, the direction of the highest conductivity is along the cardiac fibers, the second is the direction normal to the fibers that is tangent to the sheet. The direction of smallest conductivity is the direction normal to the sheets. Nevertheless, in many numerical simulations, only the fiber direction is considered. For example in electrophysiology, it is common to assume that the conductivity in plane normal to the fibers is isotropic. Thus we will concentrate on recovering only the fiber orientation.

There have been many studies of the heart fibers geometry. For example it has been studied by dissection of a dog heart by LeGrice, Hunter and Smaill [57]. The development of the DTMRI technology that we briefly outlined in Section 1.2.1 provides the community with the best source of data. Kocica *et al.* [55] provide a very complete review of attempts to model the fiber geometry.

Only a few labs have the DTMRI technology and to our knowledge, only the Center for Cardiovascular Bio-informatics and Modeling at the John Hopkins University has this data available for public download. There are seven different data sets of canine hearts and one data set of a human heart available.

The data sets contain the images as well as the diffusion tensor field. The highest eigenvalue of the tensor corresponds to the direction of the highest conductivity, that is the direction of the cardiac fibers. The second eigenvector is believed to be the direction of

(a) (b) (c)

Figure 9.1: The atlas built from the seven canine hearts (a) the geometry (b) and (c) the fibers.

the heart sheets. As we mentioned in Section 1.2.1, hearts can not be imaged by DTMRI in a living patient. The available data sets correspond to ex-vivo hearts that are artificially maintained in life. Their shape may not correspond to those of in-vivo hearts. The human heart imaged by the John Hopkins group cannot be used since its shape is not realistic.

To compensate for the several anatomical anomalies of the data set as well as the noise in the diffusion tensor information, Peyrat et al. [89] used the seven data sets of canine hearts to construct an atlas. This atlas consists of a mean heart shape, and a mean orientation of the fibers. This is the data set that we will use to deduce the fiber direction on our model. The geometry of the heart is given as a coarse volumetric mesh. The geometry and the fibers of the Asclepios model is depicted in Figure 9.1.

9.1. THE DIFFEOMORPHIC DEMONS REGISTRATION METHOD

To get the fiber orientation on our model, we need to be able to map the Asclepios model onto our heart model. The process of mapping a model to another is a registration process. This mapping needs to be a diffeomorphism, i.e. a differentiable map with differentiable inverse, so that the transformed fiber orientations are consistent with the orientation on the Asclepios model.

To do so, we used pre-coded classes in the Insight Toolkit (ITK) for diffeomorphic registration. The main class is called `itkDiffeomorphicDemonsRegistrationFilter`. This class has been written by Vercauteren et al. [115] and implements the diffeomorphic demons algorithm introduced by the same authors in [116]. It allows to control the degree of smoothness of the diffeomorphism.

By fixed image, we mean the image that we want to register. In the context of this work,

it will be the image containing the segmentation result. The moving image is the one that is deformed to fit the fixed image in the registration process. Here it will be the atlas image that will be deformed. Given a fixed image g and a moving image u the diffeomorphic demons methods consists in minimizing an energy $E(c, s)$ given by

$$E(c, s) = \frac{1}{\sigma_i^2}\text{Sim}(g, u \circ c) + \frac{1}{\sigma_x^2}\text{dist}(s, c)^2 + \frac{1}{\sigma_T^2}\text{Reg}(s), \tag{9.1}$$

$$\text{Sim}(g, u \circ c) = \frac{1}{2}\|g - u \circ c\|. \tag{9.2}$$

The variable c is an hidden variable that corresponds to the direct transformation, without any smoothing constraint. As such, it is a diffeomorphism that is given as a vector field of pixel correspondence. s is a blurred version of the diffeomorphism c. s is the diffeomorphism required to map the fibers. In general $Reg(s) = \|\nabla s\|^2$ and $\text{dist}(s, c) = \|c - s\|$. $\text{Sim}(g, u \circ c)$ measures the distance between the fixed image g and the warped image $u \circ c$.

The problem is solved by an iterative optimization method [116]. At each step, it computes a new diffeomorphism c and then set $s = K_\sigma * c$, where K_σ is a convolution kernel. The parameter σ is a parameter that is related to the three parameters σ_i, σ_x and σ_T. It controls the degree of regularization needed for the diffeomorphism. For details, the reader is referred to [115, 116].

9.2. THE REGISTRATION

Before attempting the diffeomorphic registration, it is a good idea to run an affine registration on the images in order to align the two hearts. In order to do so, we created binary images, having the same domain, that represent the two hearts. In both images, a pixel has value 1 if the pixel is in the heart muscle, and value -1 otherwise. We used a standard affine registration method implemented in ITK using the class `itkAffineTransform`. Let again u denote the moving image (the Asclepios model) and g denote our heart segmentation. The algorithm then looks for an affine transformation T minimizing

$$E(T) = \|T(u) - g\|_{L^2}^2, \tag{9.3}$$

which is simply the L^2 distance between g and $T(u)$. A gradient descent method is used for minimizing E, starting from the initial transformation $T = I$. Figure 9.2 shows two views of the aligned hearts.

Having the actual affine transformation $T(x) = A + Bx$, the fibers vectors can be trans-

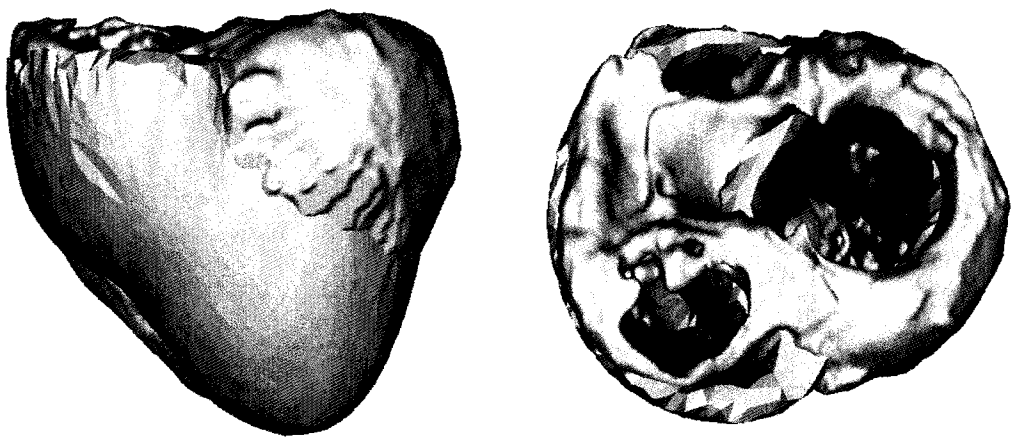


Figure 9.2: An affine alignment of the atlas (in yellow) and the segmented heart (in gray)

formed by multiplying them by the matrix B. It may change the vector size, but it is not a concern since only their direction is of interest.

With this affinely transformed atlas, it is now possible to run the diffeomorphic demons algorithm. There is a choice to make on the degree of smoothness of the diffeomorphic. Figure 9.3 shows the result of the registration algorithm with a low smoothing parameter $\sigma = 2$. The diffeomorphism s_2 was obtained in 186 min on an AMD Opteron 64 bit, 2.4 Ghz computer. The atlas is mapped to our segmented heart with great fidelity.

In Figure 9.4 are shown the result of the same registration with a higher smoothing parameter $\sigma = 10$. The diffeomorphism s_{10} was obtained in 257 min. The resulting mapped atlas is smoother and some details like the pillars are almost gone. This causes no problem since the pillars are made of a different type of tissue than the heart walls and there is no physiological reason for mapping the fibers onto the pillars.

In order to compare the two registrations, one can look at how the regular image grid is deformed using the diffeomorphisms s_2 and s_{10}. Figure 9.5 shows the two grids on a plane normal to the heart. It can be seen from this figure that the diffeomorphism s_{10} is much smoother than the diffeomorphism s_2. In the first registration, only points near the surfaces of the atlas have moved. The diffeomorphism s_{10} is more rigid: interior points are attracted by the moving surfaces. This helps in mapping the fibers in a manner that is physiologically relevant.

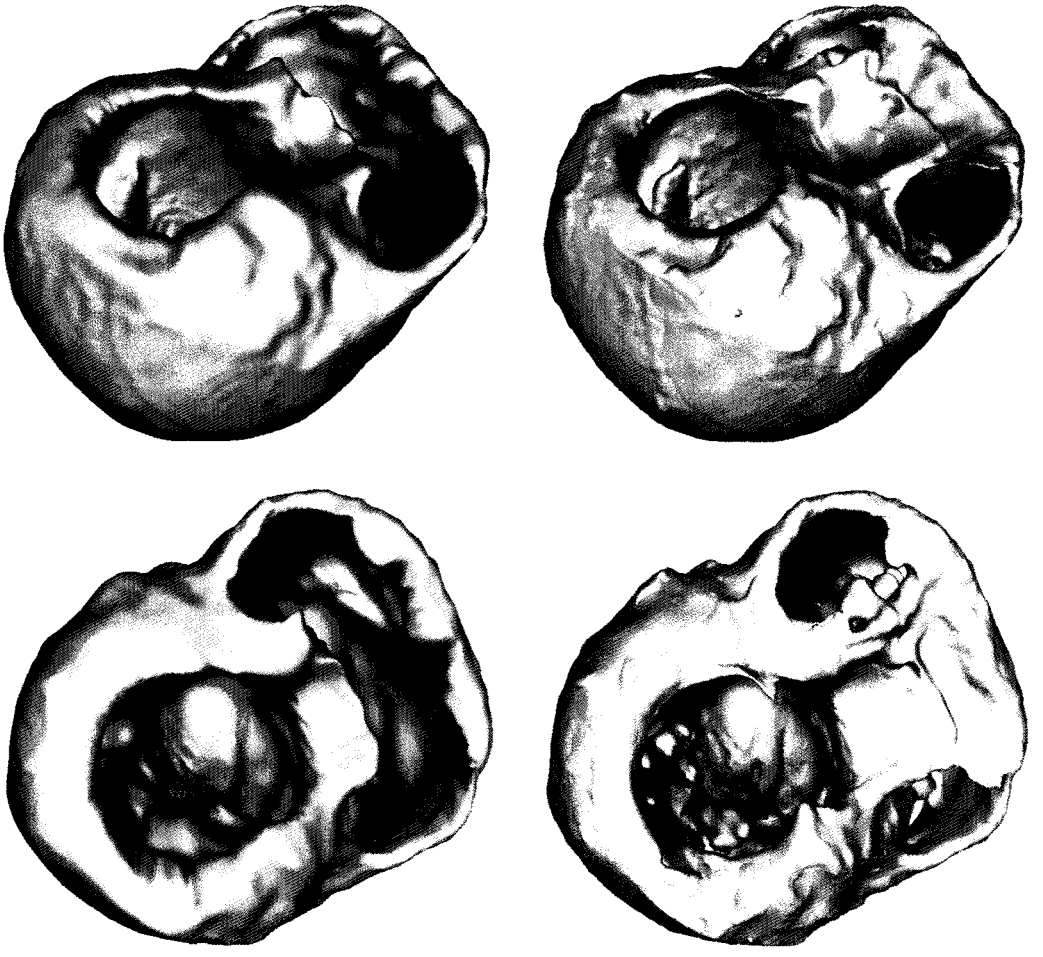


Figure 9.3: Two views of the Asclepios atlas mapped with the diffeomorphism s_2 ($\sigma = 2$). For each view, the warped atlas is on the left. On the right, the segmented heart (in gray) is shown overlaid with the warped atlas (in yellow)

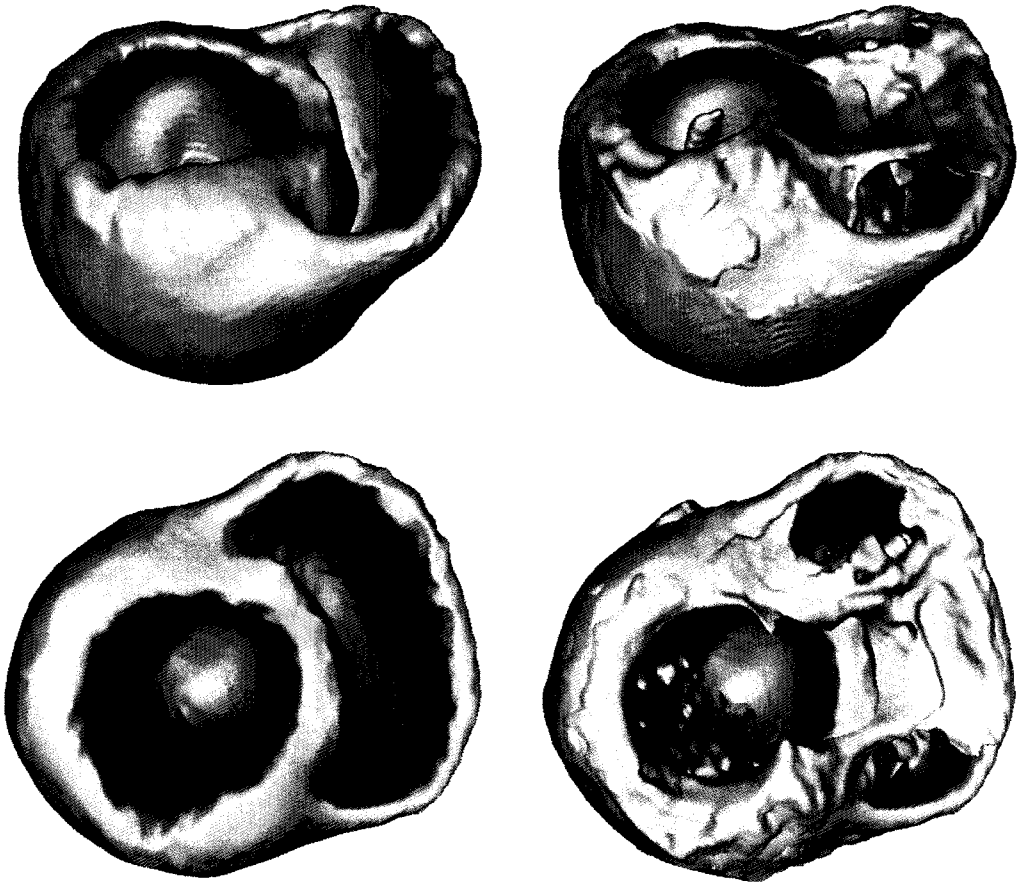
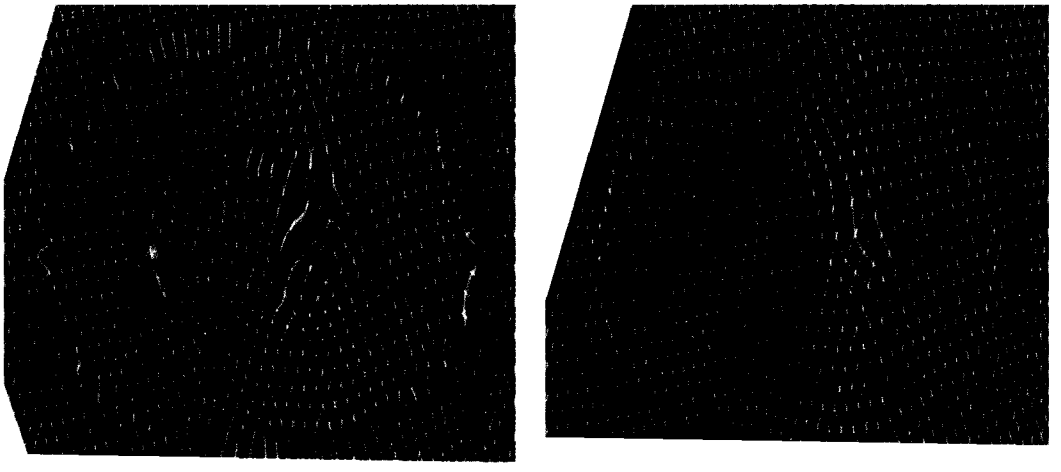
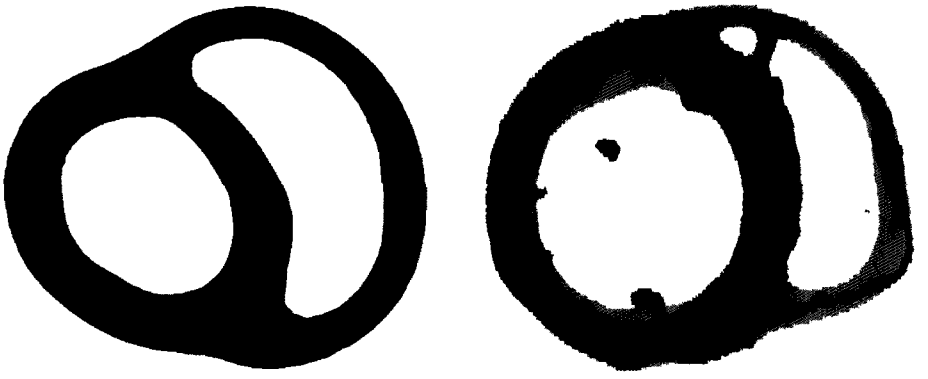


Figure 9.4: Two views of the Asclepios atlas mapped with the diffeomorphism s_{10} ($\sigma = 10$). For each view, the warped atlas is on the left. On the right, the segmented heart (in gray) is shown overlaid with the warped atlas (in yellow)



(a) (b)

Figure 9.5: An image of atlas image regular grid after transformation. (a) When $\sigma = 2$, the grid is mostly locally deformed. (b) When $\sigma = 10$, the grid is more uniformly deformed.



(a) (b)

Figure 9.6: A 2D view of the registration result on a horizontal cut plane. (a) the atlas before the transformation. (b) the warped atlas overlaid with the segmentation result. Deep blue corresponds to the common region, pale blue to the region of the warped atlas that is not in the segmented heart and black region is the region of the segmented image that is not covered by the warped atlas.

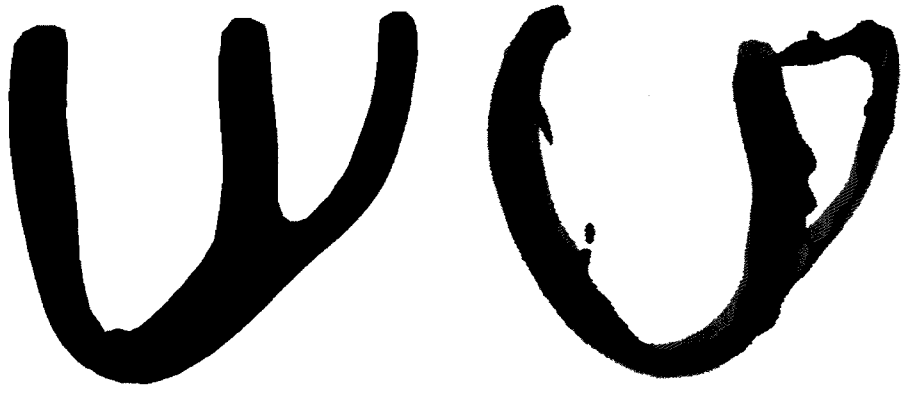
For these reasons, we decided to map the fibers using the diffeomorphism s_{10}. Assume that the fiber direction at a point P on the Asclepios model is represented by a vector $f(P) \in \mathbb{R}^3$. At a point Q in the segmented heart, the fiber direction is then

$$\tilde{f}(Q) = J_{s_{10}}(s^{-1}(Q)f(s^{-1}(Q)), \tag{9.4}$$

where $J_{s_{10}}(P)$ is the Jacobian matrix of s_{10} at point P.

This transform yields the fiber orientation on the segmented heart. It can be calculated on an image grid, or on a 3D mesh of the myocardium like the ones obtained in Section 8.3.4.

Figure 9.6 and 9.7 show two 2D views on cut planes through the 3D domain, of the mapping of the Asclepios atlas on the segmented image using the diffeomorphism s_{10}. On the left is the atlas before the transformation. On the right, the warped atlas is shown overlaid with the segmentation result. The warped atlas is shown in blue. The region in deep blue is the one that is common with the segmented image while the pale blue one is the region of the warped atlas that is not in the segmented heart. Conversely, the black region is the region of the segmented image that is not covered by the warped atlas.



(a) (b)

Figure 9.7: A different 2D view of the registration result on a vertical cut plane. (a) the atlas before the transformation. (b) the warped atlas overlaid with the segmentation result. Deep blue corresponds to the common region, pale blue to the region of the warped atlas that is not in the segmented heart and black region is the region of the segmented image that is not covered by the warped atlas.

9.3. INPAINTING

The fact that there exist regions Ω_{missing} of the segmented image that are not covered by the warped atlas may cause problems since in this region, there will be no fiber direction assigned. On has to provide a fiber orientation to the voxels in the region Ω_{missing}. The problem that consists in recovering missing parts of an image is called the image inpainting problem. Let Ω_{seg} denote region of the segmented myocardium. Ω_{seg} is the domain of interest in this application. There is a sub-region of the domain where the fiber orientation is known via the Equation 9.4. Let Ω_{fibers} denote this region. The fiber orientation is then missing on the other sub-domain Ω_{missing} and

$$\Omega_{\text{seg}} = \Omega_{\text{fibers}} \cup \Omega_{\text{missing}} \tag{9.5}$$

Recall that the total variation model for denoising looks for steady state of the following equation

$$u_t = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \lambda(g - u), \tag{9.6}$$

where g is the image to be denoised. An easy solution to the inpainting problem consists in solving Equation 9.6 with particular boundary conditions. Let g denote the image of

Figure 9.8: An example of a region belonging to Ω_{missing} (in grey). Dirichlet boundary conditions are used on the black line and homogeneous Neumann boundary conditions are used on the white line.

the fibers. Then g has accurate values in Ω_{fibers} and can be set to 0 on Ω_{missing}. Then the following problem is solved.

$$\begin{cases} u_t = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \lambda(g-u), & \text{on } \Omega_{\text{missing}} \\ u = g & \text{on } \partial\Omega_{\text{missing}} \cap \partial\Omega_{\text{fibers}} \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega_{\text{missing}} \cap \partial\Omega_{\text{seg}} \end{cases} \tag{9.7}$$

As an example, Figure 9.8 shows in gray a region belonging to Ω_{missing}. The total variation problem is solved with a Dirichlet boundary condition on the black boundary of the gray region and with the homogeneous Neumann boundary condition on the white boundary of the gray region. The inpainting problem is as easy to solve in parallel as the Chan-Vese and the subjective surfaces problem. Figure 9.9 shows the region before and after inpainting. The use of the total variation equation prevent the inpainted region from being too blurry.

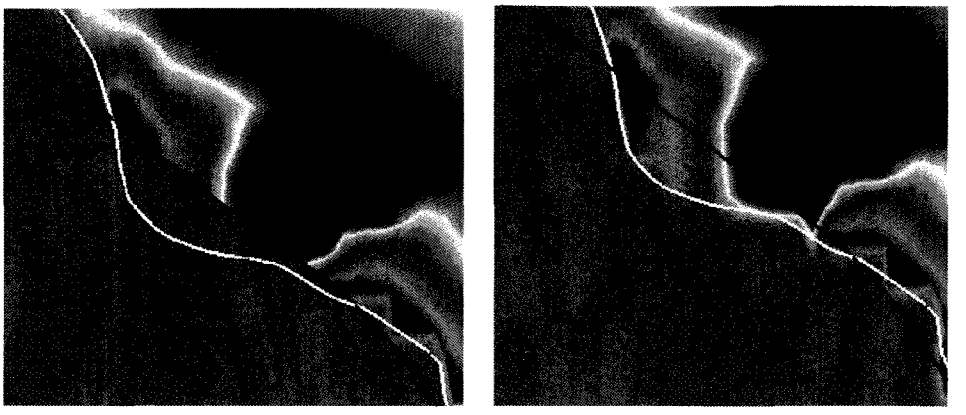


Figure 9.9: The result of inpainting. (a) The region before inpainting. (b) The region after inpainting.

Figure 9.10: Two views of the mapped fibers together with the segmented heart surface. The arrows are colored according to their orientation.

9.4. THE FINAL RESULT

Having solved the inpainting problem, the fibers are now defined over the whole myocardium (the region Ω_{seg}). Figures 9.10 and 9.11 show the resulting fibers with and without the segmented heart surface.

Figure 9.11: Two views of the mapped fibers without the segmented heart surface.

10

Conclusion

10.1. SCIENTIFIC CONTRIBUTIONS

This thesis has scientific contributions of different nature. First, we provided in this thesis careful analysis of known segmentation methods. Next we introduce variants of these algorithms to address our segmentation problem as well as computing issues. With our results, we provide the scientific community with a realistic 3D model of the human heart that is anatomically accurate and has fiber orientation information. Finally we made available to the community a small C++ toolkit named *SPDEIPTK* for PDE methods in image processing. This toolkit is intended as research oriented: it is easy to learn and use and it features an almost transparent parallel implementation. We expand on each of these aspects below.

10.1.1. Algorithm analysis

In Chapter 5, we present a careful analysis of the active contour without edges algorithm. To our knowledge, the aspects considered there constitute original material.

First we compared the impact of different discretizations of the Mumford-Shah energy and of its Euler-Lagrange equation on the numerical solution of the problem. We concluded that some discretizations, while being more natural, do not behave as well as some others.

We also evaluated the impact of the initial condition that is chosen on the speed of convergence of the algorithm.

We studied carefully the hierarchical method of Gao and Bui [39], and show test cases where it performs better than the original multiphase algorithm of Vese and Chan [117]. We show that the hierarchical segmentation is a more natural framework for segmenting junctions with three segments.

10.1.2. Proposed variants

In Chapter 5, we introduced two new types of initial conditions that make the active contour without edges algorithm converge more quickly. We defined a random initial curve, that has values between -1 and 1. We also proposed the use of an initial curve $\bar{\phi}_0$ that is the solution of the problem when there is no curvature term. We tested the initial conditions on many images and in all cases the initial condition $\bar{\phi}_0$ converges more quickly. However, the random initial, which still converges quickly, has no bias and is less prone to be stuck in a local minimum.

In this chapter, we also introduced the hierarchical segmentation algorithm with a L^1 fidelity term. The L^1 fidelity is less sensitive to noise and tends to split the image into regions of relatively close size. Using this fidelity term, the hierarchical segmentation of the heart is done in less computational time than with the original L^2 fidelity. It also requires fewer hierarchical steps.

This analysis of the active contour without edges is then applied in Chapter 6 to the segmentation of the heart chambers. To the best of our knowledge, this is the first time that the method of active contour without edges is applied to cardiac segmentation. In fact, to our knowledge, it is the first time that it is applied to such a large data set. The new parallel implementation made this computation affordable.

In Chapter 7, we present a variant of the subjective surface problem introduced by Sarti, Malladi and Sethian [98, 99]. We propose to solve the problem on an annulus around the heart chambers. The problem is solved with Dirichlet boundary conditions: 1 on the chambers side, 0 on the exterior side. This choice of domain and boundary conditions prevent the surface from leaking and yields a nicely segmented epicardial surface.

The full heart segmentation is the first success of a heart segmentation that is not using a model based approach.

10.1.3. 3D heart model

The 3D heart model that is presented constitutes the main contribution of this thesis. The model is made from a 3D CT cardiac image of a living patient. In principle, all the work done to build this model can be reproduced with other images. This could be used to build patient specific geometries and to perform simulations that could assess for example the feasibility of a surgical operation for a given patient.

The model consists in a high quality mesh, for which there is an associated fiber orientation vector on each node. The mesh has been generated from segmentation results obtained

with a combination of the Chan-Vese method and of the subjective surface method. The fiber orientation was obtained by mapping the Asclepios atlas on the segmented heart with a diffeomorphic demons registration algorithm. Since the 3D heart contains an orientation of the fibers, it is suitable for simulating electrophysiological waves and the mechanical deformation of the heart.

It is the most precise and accurate model of the heart that is publicly available for researchers. Also, it is among the few, if not the sole model, publicly available or not, that has the geometry of the pillars included.

The plan is to add features to this model in the future, to make it even more complete.

10.1.4. SPDEIPTK

During this project, we have developed C++ classes that handles images that are in the standard meta-image format (.mhd). It is easy to learn and use. The classes are designed so that they can be used in parallel computing with very little effort.

The result is a small image processing toolkit that is suitable for research use. It is mostly centered on PDE methods in image processing but the parallel implementation would also accommodate filters that use minimal global information on the image.

The toolkit is publicly available from the website:
`http://www.mathstat.uottawa.ca/~orous272/spdeiptk/index.html`.

10.2. PERSPECTIVES

10.2.1. Applications

The resulting 3D model of the heart will be used to perform numerical simulations in electrophysiology. Such simulations have already been made on the 2D version of the heart model shown in Figures 8.1. This has been done by Coudière et al. [29] and Pierre, Rousseau and Bourgault [90]. There will also be attempts to numerically simulate transmembrane potential wave propagation using mesh adaptation on this model.

The model may also serve for numerical simulations of the coupling between the mechanical deformation of the heart and the blood flow inside the heart. In that case the cardiac chambers would also need to be meshed.

10.2.2. Improvement of the model

The model could be improved in many ways. First the model would gain if the method was applied successfully to multiple data sets. In that eventuality the shape variability of the heart and of the recovery of fine anatomical features could be evaluated. Also, the impact on numerical simulations of heart physiology could be assessed.

The model would gain interest from adding more anatomical structures. In the future, we plan on adding both atria and the aorta to the model. These structures are already segmented, and they just need to be meshed together with the ventricle. We also plan on building a valve geometry that could be added to the model. The valve geometry would consists of three triangular surface meshes. With this information, issues of leaking valves can be studied.

10.2.3. Methodology

In Section 7.2 we proposed a modification of the subjective surface algorithm that takes into account the cavity segmentation. It performs the computation only in a narrow band around the cavities. The evolution is made using the geodesic active contour equation. We believe that it would be possible to add a term to the evolution equation that would favor the level curves to be parallel to the cavities. This could probably be done in the same flavor as the work of Kimmel [53].

Solving the subjective surface problem was the part of the model building that required longest computing times. However, it would be possible to solve it with a semi implicit time stepping scheme like the one introduced by Mikula, Sarti and Sgallari [68, 69]. Using that scheme decreases considerably the computing time. Mikula and Sarti [67] even proposed a parallel implementation of that method.

A

Code

During this thesis, a considerable amount of C++ code has been written for solving PDE problems in image processing. The code has been gathered in a small set of classes and functions that are intended to be used mainly for PDE methods in image processing, but not restricted to this.

The main interesting feature of the code is that it may process large 3D images in parallel. For most parallelizable image processing algorithm, the algorithm can be written in its standard form, and called by a parallel code in an almost transparent way.

This toolkit is to be used for research purposes and provided as is with no guarantee that it is bug free. There exists toolkits far more complete and stable among which is the excellent ITK (Insight Toolkit) [49].

There are several reasons why the ITK has not been used for this project. Programming efficient filters in ITK is an art. Hard coding algorithms within the ITK framework is easy, but leads to slow algorithms. A simple C++ implementation can increase the speed by a factor of 2 to 5 depending on the algorithm. The other main reason is that ITK does not have parallel capabilities.

The hope is that this set of code could be useful for someone testing image processing algorithms by providing a simple serial and parallel code set.

The input and output format available can easily be created or viewed by the visualization software Paraview[1].

In this Toolkit, there are four main classes:

1. `Image2D` : a class that handles 2D images. It reads and outputs .vtk format.

2. `Image3D` : a class that handles 3D images. It reads and outputs meta-image format (.mhd).

[1] http://www.paraview.org

3. `PImage3D` : A class that inherits of Image3D. It is the parallel version of Image3D. All functions of Image3D that require processor interactions are redefined.

4. `Mesh3D` : A class that handles 3D volumetric meshes. It reads and writes .vtk legacy (ASCII) format. This class is more rudimentary and does not have a lot of functions associated with it.

Along with those, there are many sample codes provided that implement standard algorithms. For example:

- the Chan-Vese algorithm (also known as active contours without edges) in 2D, 3D, and parallel 3D.
- The total variation denoising algorithm, again in 2D, 3D and parallel 3D.
- The subjective surface algorithm in 2D, 3D and parallel 3D.
- The reinitialization equation in 2D and parallel 3D.
- The heat equation in 2D and parallel 3D.
- Many codes that performs simple operations.

The code is freely available from the website:
`http://www.mathstat.uottawa.ca/~orous272/spdeiptk`.

Bibliography

[1] M.J. Ackerman. The visible human project. *Proceedings of the IEEE*, 86(3):504–511, 1998.

[2] S. Alliney. Digital filters as absolute norm regularizers. *Signal Processing, IEEE Trans. Signal Process.*, 40(6):1548–1562, 1992.

[3] S. Alliney. Recursive median filters of increasing order: a variational approach. *Signal Processing, IEEE Trans. Signal Process.*, 44(6):1346–1354, 1996.

[4] S. Alliney. A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. *IEEE Trans. Signal Process.*, 45(4):913–917, 1997.

[5] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford University Press, 2000.

[6] L. Ambrosio and V.M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Comm. Pure Appl. Math*, 43(8):999–1036, 1990.

[7] L. Ambrosio and V.M. Tortorelli. On the approximation of free discontinuity problems. *Boll. Un. Mat. Ital*, 6:105–123, 1992.

[8] A. Andreopoulos and J.K. Tsotsos. A Novel Algorithm for Fitting 3-D Active Appearance Models: Applications to Cardiac MRI Segmentation. In *Proc. Scandinavian Conf. Image Analysis*, pages 729–739. Springer, 2005.

[9] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer, 2002.

[10] J.F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-Texture Image Decomposition, Modeling, Algorithms, and Parameter Selection. *Inter. J. of Computer Vision*, 67(1):111–136, 2006.

[11] L. Axel. Tagged MRI-based studies of cardiac function. *Lecture Notes in Computer Science*, pages 1–7, 2003.

[12] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1. IEEE Computer Society; 1999, 2001.

[13] A. Braides. *Approximation of Free-Discontinuity Problems*. Springer, 1998.

[14] M. Burger and M. Hintermuller. Projected gradient flows for BV/level set relaxation. *PAMM*, 5(1), 2005.

[15] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer. Math.*, 66(1):1–31, 1993.

[16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *Inter. J. of Computer Vision*, 22(1):61–79, 1997.

[17] T. Chan and W. Zhu. Level set based shape prior segmentation. In *IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, volume 2, 2005.

[18] T.F. Chan and S. Esedoglu. Aspects of total variation regularized L1 function approximation. *SIAM J. Appl. Math.*, 65(5):1817–1837, 2005.

[19] T.F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. on Appl. Math.*, 66(5):1632–1648, 2006.

[20] T.F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial Mathematics, 2005.

[21] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[22] G. Charpiat, O. Faugeras, and R. Keriven. Approximations of shape metrics and application to shape warping and empirical shape statistics. *Foundations of Computational Mathematics*, 5(1):1–58, 2005.

[23] G. Chung and L.A. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. *Lecture Notes in Computer Science*, 3757:439–455, 2005.

[24] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multimodality image registration based on information theory. *Information processing in medical imaging*, pages 263–274, 1995.

[25] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[26] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. *World Wide Web Publication, February*, 2001.

[27] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[28] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti. Left ventricular volume estimation for real-time three-dimensional echocardiography. *IEEE Trans. on Medical Imaging*, 21(9):1202–1208, 2002.

[29] Y. Coudiere, C. Pierre, O. Rousseau, and R. Turpault. 2D/3D Discrete Duality Finite Volume (DDFV) scheme for anisotropic-heterogeneous elliptic equations, application to the electrocardiogram simulation. *Inter. J. on Finite Volumes*, 6(1), 2009.

[30] D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. In *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 169–176, 2003.

[31] J. Darbon. A Note on the Discrete Binary Mumford-Shah Model. *Lecture Notes in Computer Science*, 4418:283, 2007.

[32] O. Ecabert, J. Peters, H. Schramm, C. Lorenz, J. Von Berg, M.J. Walker, M. Vembar, ME Olszewski, K. Subramanyan, G. Lavi, et al. Automatic model-based segmentation of the heart in CT images. *IEEE Trans. on Medical Imaging*, 27(9):1189–1201, 2008.

[33] O. Ecabert, J. Peters, and J. Weese. Modeling shape variability for full heart segmentation in cardiac computed-tomography images. In *Proceedings of SPIE*, volume 6144, page 61443R, 2006.

[34] B. Flury. *Multivariate Statistics: a Practical Approach*. Chapman & Hall, Ltd. London, UK, 1988.

[35] A. Frangi, D. Rueckert, and J.S. Duncan. Three-dimensional cardiovascular image analysis. *IEEE Trans. on Medical Imaging*, 21(9):1005, 2002.

[36] A.F. Frangi, W.J. Niessen, and M.A. Viergever. Three-dimensional modeling for functional analysis of cardiac images: A review. *IEEE Transactions on Medical Imaging*, 20(1):2–5, 2001.

[37] A.F. Frangi, D. Rueckert, J.A. Schnabel, and W.J. Niessen. Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling. *IEEE Trans. on Medical Imaging*, 21(9):1151–1166, 2002.

[38] K.D. Fritscher and R. Schubert. 3D image segmentation by using statistical deformation models and level sets. *International Journal of Computer Assisted Radiology and Surgery*, 1(3):123–135, 2006.

[39] S. Gao and T.D. Bui. Image segmentation and selective smoothing by using Mumford-Shah model. *IEEE Transactions on Image Processing*, 14(10):1537–1549, 2005.

[40] P.L. George and H. Borouchaki. *Delaunay triangulation and meshing: application to finite elements*. Kogan Page, 1998.

[41] O. Gerard, A.C. Billon, J.M. Rouet, M. Jacob, M. Fradkin, and C. Allouche. Efficient model-based quantification of left ventricular function in 3-D echocardiography. *IEEE transactions on medical imaging*, 21(9):1059–1068, 2002.

[42] F. Gibou and R. Fedkiw. A fast hybrid k-means level set algorithm for segmentation. *4th Annual Hawaii International Conference on Statistics and Mathematics*, pages 281–291, 2005.

[43] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Birkhauser, 1984.

[44] E. Giusti. *Direct Methods in the Calculus of Variations*. World Scientific, 2003.

[45] J.A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc. New York, NY, USA, 1975.

[46] L. He and S. Osher. Solving the Chan-Vese Model by a Multiphase Level Set Algorithm Based on the Topological Derivative. *Proceedings of the 1st International Conference on Scale Space Variational Methods in Computer Vision*, 2007.

[47] http://www-math.mit.edu/ persson/mesh/.

[48] http://www.sgi.com/tech/stl/nth_element.html.

[49] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, second edition, 2005.

[50] M.P. Jolly. Automatic segmentation of the left ventricle in cardiac MR and CT images. *International Journal of Computer Vision*, 70(2):151–163, 2006.

[51] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Interna-*

tional Journal of Computer Vision, 1(4):321–331, 1988.

[52] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 810–815, 1995.

[53] R. Kimmel. Fast edge integration. *Geometric Level Set Methods in Imaging, Vision, and Graphics*, 2003.

[54] B.M. Klingner. Improving tetrahedral meshes. 2008.

[55] M.J. Kocica, A.F. Corno, F. Carreras-Costa, M. Ballester-Rodes, M.C. Moghbel, C.N.C. Cueva, V. Lackovic, V.I. Kanjuh, and F. Torrent-Guasp. The helical ventricular myocardial band: global, three-dimensional, functional architecture of the ventricular myocardium. *European J. of Cardio-Thoracic Surgery*, 29:21–40, 2006.

[56] T.M. Le and L.A. Vese. Image Decomposition Using Total Variation and div(BMO). *Multiscale Modeling & Simulation*, 4(2):390–423, 2005.

[57] I.J. LeGrice, P.J. Hunter, and B.H. Smaill. Laminar structure of the heart: a mathematical model. *Amer. J. of Physiology- Heart and Circulatory Physiology*, 272(5):2466–2476, 1997.

[58] M.E. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2000.

[59] D. Lloyd-Jones, R. Adams, M. Carnethon, G. De Simone, T.B. Ferguson, K. Flegal, E. Ford, K. Furie, A. Go, K. Greenlund, et al. Heart Disease and Stroke Statistics–2009 Update. A Report From the American Heart Association Statistics Committee and Stroke Statistics Subcommittee. *Circulation*, 2008.

[60] M. Lorenzo-Valdes, G.I. Sanchez-Ortiz, R. Mohiaddin, and D. Rueckert. Atlas-based segmentation and tracking of 3D cardiac MR images using non-rigid registration. *Lecture Notes in Computer Science*, pages 642–650, 2002.

[61] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[62] A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM J. Sci. Comput.*, 22(2):387–405, 2001.

[63] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, 1995.

[64] C. Meyer, O. Ecabert, J. Peters, R. Kneser, R. Manzke, R.C. Chan, and J. Weese. A multi-modality segmentation framework: application to fully automatic heart segmentation. In *Proceedings of SPIE*, volume 7259, page 72594L, 2009.

[65] C. Meyer, R. Manzke, J. Peters, O. Ecabert, R. Kneser, V.Y. Reddy, R.C. Chan, and J. Weese. Automatic Intra-operative Generation of Geometric Left Atrium/Pulmonary Vein Models from Rotational X-Ray Angiography. In *Pro-*

ceedings of the 11th International Conference on Medical Image Computing and Computer-Assisted Intervention, Part II, pages 61–69. Springer, 2008.

[66] P.W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *Arxiv preprint math.DG/0312384*, 2003.

[67] K. Mikula and A. Sarti. Parallel Co-Volume Subjective Surface Method For 3d Medical Image Segmentation. *Deformable Models Theory and Biomaterial Applications*, page 123, 2007.

[68] K. Mikula, A. Sarti, and F. Sgallari. Co-volume level set method in subjective surface based medical image segmentation. *Handbook of Medical Image Analysis: Segmentation and Registration Models (J. Suri et al., Eds.), Springer, New York*, pages 583–626, 2005.

[69] K. Mikula, A. Sarti, and F. Sgallari. Co-volume method for Riemannian mean curvature flow in subjective surfaces multiscale segmentation. *Computing and Visualization in Science*, 9(1):23–31, 2006.

[70] W. Mio and A. Srivastava. Elastic-string models for representation and analysis of planar shapes. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, 2004.

[71] S.C. Mitchell, J.G. Bosch, B.P.F. Lelieveldt, R.J. Van der Geest, J.H.C. Reiber, and M. Sonka. 3-D active appearance models: segmentation of cardiac MR and ultrasound images. *IEEE Transactions on Medical Imaging*, 21(9):1167–1178, 2002.

[72] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math*, 42(5):577–685, 1989.

[73] M. Nikolova. Minimizers of Cost-Functions Involving Nonsmooth Data-Fidelity Terms. Application to the Processing of Outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002.

[74] M. Nikolova. A Variational Approach to Remove Outliers and Impulse Noise. *Journal of Mathematical Imaging and Vision*, 20(1):99–120, 2004.

[75] S. Ordas, E. Oubel, R. Sebastian, and A.F. Frangi. Computational anatomy atlas of the heart. In *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, pages 338–342, 2007.

[76] S. Osher and R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[77] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed- Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[78] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[79] K. Park, D. Metaxas, and L. Axel. A finite element model for functional analysis of 4D cardiac-tagged MR images. *Lecture Notes in Computer Science*, pages 491–498, 2003.

[80] K. Park, D.N. Metaxas, and L. Axel. LV-RV shape modeling based on a blended

parameterized model. *Lecture Notes in Computer Science*, pages 753–761, 2002.

[81] K. Park, A. Montillo, D. Metaxas, and L. Axel. Volumetric heart modeling and analysis. *Communications of the ACM*, 48, 2005.

[82] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155(2):410–438, 1999.

[83] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[84] D. Perperidis, R. Mohiaddin, P. Edwards, and D. Rueckert. Segmentation of cardiac MR and CT image sequences using model-based registration of a 4D statistical model. In *Proceedings of SPIE*, volume 6512, page 65121D, 2007.

[85] P.O. Persson and G. Strang. A Simple Mesh Generator in MATLAB. *SIAM Review*, 46(2):329–345, 2004.

[86] J. Peters, O. Ecabert, C. Lorenz, J. von Berg, M.J. Walker, T.B. Ivanc, M. Vembar, M.E. Olszewski, and J. Weese. Segmentation of the heart and major vascular structures in cardiovascular CT images. In *Proceedings of SPIE*, volume 6914, page 691417, 2008.

[87] J. Peters, O. Ecabert, C. Meyer, H. Schramm, R. Kneser, A. Groth, and J. Weese. Automatic whole heart segmentation in static magnetic resonance image volumes. *Lecture Notes in Computer Science*, 4792:402, 2007.

[88] J. Peters, O. Ecabert, H. Schramm, and J. Weese. Discriminative boundary detection for model-based heart segmentation in CT images. In *Proceedings of SPIE*, volume 6512, page 65120H, 2007.

[89] J.M. Peyrat, M. Sermesant, X. Pennec, H. Delingette, C. Xu, E.R. McVeigh, and N. Ayache. A computational framework for the statistical analysis of cardiac diffusion tensors: Application to a small database of canine hearts. *IEEE Trans. on Medical Imaging*, 26(11):1500–1514, 2007.

[90] C. Pierre, O. Rousseau, and Y. Bourgault. Segmented medical images based simulations of Cardiac electrical activity and electrocardiogram: a model comparison. *HAL preprint hal-00378841*, 2009.

[91] T. Riklin-Raviv, N. Kiryati, and N. Sochen. Unlevel-sets: Geometry and prior-based segmentation. *Lecture Notes in Computer Science*, pages 50–61, 2004.

[92] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[93] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

[94] Walter Rudin. *Functional analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill Inc., New York, second edition, 1991.

[95] G.I. Sanchez-Ortiz, G.J.T. Wright, N. Clarke, J. Declerck, A.P. Banning, and J.A. Noble. Automated 3-D echocardiography analysis compared with manual delineations and SPECT MUGA. *IEEE Trans on Medical Imaging*, 21(9):1069–1076, 2002.

[96] MF Santarelli, V. Positano, L. Landini, and A. Benassi. A new algorithm for 3D automatic detection and tracking of cardiac wall motion. *Computers in Cardiology 1999*, pages 133–136, 1999.

[97] MF Santarelli, V. Positano, P. Marcheschi, L. Landini, P. Marzullo, and A. Benassi. Multimodal cardiac image fusion by geometrical features registration and warping. *Computational Cardiology. pp. 277-280. 2001*, 2001.

[98] A. Sarti, R. Malladi, and J.A. Sethian. Subjective surfaces: a method for completing missing boundaries. *Proceedings of the National Academy of Sciences*, 97:12, 2000.

[99] A. Sarti, R. Malladi, and JA Sethian. Subjective surfaces: a geometric model for boundary completion. *International Journal of Computer Vision*, 46(3):201–221, 2002.

[100] H. Schramm, O. Ecabert, J. Peters, V. Philomin, and J. Weese. Towards fully automatic object detection and segmentation. In *Proceedings of SPIE*, volume 6144, page 614402, 2006.

[101] D.F. Scollan, A. Holmes, R. Winslow, and J. Forder. Histological validation of myocardial microstructure obtained from diffusion tensor magnetic resonance imaging. *American Journal of Physiology- Heart and Circulatory Physiology*, 275(6):2308–2318, 1998.

[102] J.A. Sethian et al. *Level set methods and fast marching methods*. Cambridge University Press Cambridge, 1999.

[103] P. Smereka. Semi-Implicit Level Set Methods for Curvature and Surface Diffusion Motion. *Journal of Scientific Computing*, 19(1):439–456, 2003.

[104] P. Smereka. The numerical approximation of a delta function with application to level set methods. *Journal of Computational Physics*, 211(1):77–90, 2006.

[105] N.P. Smith, D.P. Nickerson, E.J. Crampin, and P.J. Hunter. Multiscale computational modelling of the heart. *Acta Numerica*, 13:371–431, 2004.

[106] B. Song and T. Chan. A fast algorithm for level set based optimization. *UCLA CAM Report*, 2(68), 2002.

[107] C. Stevens, E. Remme, I. LeGrice, and P. Hunter. Ventricular mechanics in diastole: material parameter sensitivity. *Journal of biomechanics*, 36(5):737–748, 2003.

[108] Michael Struwe. *Variational methods*, volume 34 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer-Verlag, Berlin, third edition, 2000. Applications to nonlinear partial differential equations and Hamiltonian systems.

[109] 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.

[110] J.P. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[111] C. Tobon-Gomez, C. Butakoff, S. Aguade, F. Sukno, G. Moragas, and AF Frangi. Automatic construction of 3D-ASM intensity models by simulating image acquisi-

tion: application to myocardial gated SPECT studies. *IEEE Transactions on Medical Imaging*, 27(11):1655–1667, 2008.

[112] A. Tsai, A. Yezzi Jr, W. Wells III, C. Tempany, D. Tucker, A. Fan, WE Grimson, and A. Willsky. Model-based curve evolution technique for image segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 2001.

[113] A. Tsai, A. Yezzi Jr, 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.

[114] H.C. Van Assen, M.G. Danilouchkine, A.F. Frangi, S. Ordás, J.J.M. Westenberg, J.H.C. Reiber, and B.P.F. Lelieveldt. SPASM: A 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data. *Medical Image Analysis*, 10(2):286–303, 2006.

[115] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons using ITK's finite difference solver hierarchy. *The Insight Journal*, 2007.

[116] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. *Lecture Notes in Computer Science*, 4792:319, 2007.

[117] L.A. Vese and T.F. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *Inter. J. of Computer Vision*, 50(3):271–293, 2002.

[118] F.J. Vetter and A.D. McCulloch. Three-dimensional analysis of regional cardiac function: a model of rabbit ventricular anatomy. *Progress in Biophysics and Molecular Biology*, 69(2-3):157–183, 1998.

[119] P. Viola and W.M. Wells III. Alignment by maximization of mutual information. *Inter. J. of Computer Vision*, 24(2):137–154, 1997.

[120] J. von Berg and C. Lorenz. Multi-surface cardiac modelling, segmentation, and tracking. *Proc. Functional Imaging and Modeling of the Heart*, pages 1–11, 2005.

[121] J. Weese, M. Kaus, C. Lorenz, S. Lobregt, R. Truyen, and V. Pekar. Shape constrained deformable models for 3D medical image segmentation. *Lecture Notes in Computer Science*, pages 380–387, 2001.

[122] C.F. Westin, L.M. Lorigo, O. Faugeras, W.E.L. Grimson, S. Dawson, A. Norbash, and R. Kikinis. Segmentation by adaptive geodesic active contours. *Lecture Notes in Computer Science*, pages 266–275, 2000.

[123] I. Wolf, M. Hastenteufel, R. De Simone, M. Vetter, G. Glombitza, S. Mottl-Link, C.F. Vahl, and H.P. Meinzer. ROPES: A semiautomated segmentation method for accelerated analysis of three-dimensional echocardiographic data. *IEEE transactions on medical imaging*, 21(9):1091–1104, 2002.

[124] A. Yezzi and A. Mennucci. Metrics in the space of curves. *Arxiv Preprint math.DG/0412454*, 2004.

[125] A.J. Yezzi and S. Soatto. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. *Inter. J. of Computer*

Vision, 53(2):153–167, 2003.

[126] T.S. Yoo, M.J. Ackerman, W.E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, and R. Whitaker. Engineering and algorithm design for an image processing API: A technical report on ITK-the Insight Toolkit. *Studies in health technology and informatics*, pages 586–592, 2002.

[127] L. Younes. Computable elastic distances between shapes. *SIAM J. Appl. Math.*, pages 565–586, 1998.

[128] E.A. Zerhouni, D.M. Parish, W.J. Rogers, A. Yang, and E.P. Shapiro. Human heart: tagging with MR imaging–a method for noninvasive assessment of myocardial motion. *Radiology*, 169(1):59–63, 1988.

[129] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Fast automatic heart chamber segmentation from 3D CT data using marginal space learning and steerable features. In *Proc. ICCV*, volume 2007, pages 1–8. Citeseer, 2007.

[130] L. Zhukov, Z. Bao, I. Guskov, J. Wood, and D. Breen. Dynamic deformable models for 3D MRI heart segmentation. In *SPIE Medical Imaging*, volume 4684, pages 1398–1405, 2002.

[131] William P. Ziemer. *Weakly differentiable functions*, volume 120 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1989. Sobolev Spaces and Functions of Bounded Variation.