INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Adjustment of Deformable Models to Organ Surfaces in 3D Images

Carlos Jorge Andrade Mariz Santiago

Dissertação para a obtenção de Grau de Mestre em

Engenharia Biomédica

Júri

Presidente: Patrícia Margarida Piedade Figueiredo
Orientador: Jorge dos Santos Salvador Marques
Co-Orientador: Jacinto Carlos Marques Peixoto do Nascimento
Vogal: Arnaldo Joaquim de Castro Abrantes
Vogal: Pedro Manuel Patrício Matos

Outubro 2011

Resumo

A segmentação do ventrículo esquerdo (LV) em imagens ecocardiográficas é uma metodologia vulgarmente utilizada para avaliar o funcionamento cardíaco e para detectar anomalias. Tradicionalmente, os cardiologistas segmentam o LV no fim das fases sistólica e diastólica para determinar a fracção de ejecção. Contudo, a segmentação manual do LV é uma tarefa morosa, o que significa que sistemas automáticos de segmentação proporcionam ferramentas poderosas para aumentar o rendimento das clínicas. Já foram propostos vários sistemas de segmentação do LV, que normalmente utilizam algoritmos de aprendizagem ou modelos com restrições de forma para lidar com a baixa qualidade das imagens de ultrassom e com a grande quantidade de dados 3D. Todavia, estas abordagens podem não revelar anomalias anatómicas no LV. Esta tese propõe um sistema de segmentação 3D robusto que consiste num modelo deformável que usa um modelo de estimação robusta e pesquisa direccional para detectar a fronteira do LV. Os resultados demonstram que o algoritmo obtém bons resultados quer em imagens sintéticas quer imagens ecocardiográficas. As segmentações do LV obtidas foram comparadas com as segmentações realizadas por um especialista, resultando numa média de 4 ± 1 pixéis de distância entre pontos pertencentes a ambas.

Palavras-chave: Ecocardiografia 3D, Ventrículo Esquerdo, Segmentação, Modelos Deformáveis, Extracção de Features, Estimação Robusta

Abstract

The segmentation of the left ventricle (LV) in echocardiographic data has proven itself a useful methodology to assess cardiac function and to detect abnormalities. Traditionally, cardiologists segment the LV border at end-systolic and end-diastolic phases to determine the ejection fraction. However, the manual segmentation of the LV is a tedious and time demanding task, which means automated segmentation systems can provide a powerful tool to improve workflow in a clinical site. Many LV segmentation systems have been proposed, usually using learning methods or models with shape constraints to deal with the low quality of ultrasound images and the large amount of 3D data. But these approaches may conceal unexpected anatomical abnormalities in the LV. This thesis proposes a robust 3D segmentation system consisting of a deformable model that uses a directional feature search and a robust estimation technique to detect the LV border. Results show that the algorithm performs well both in synthetic and real (echocardiographic) data. The obtained LV segmentations were compared with the manual segmentations performed by an expert, yielding an average distance of 4 ± 1 pixel between points from both segmentations.

Keywords: 3D Echocardiography, Left Ventricle, Segmentation, Deformable Models, Feature Extraction, Robust Estimation

Acknowledgements

This work was supported by FCT under contract PTDC/EEA-CRO/103462/2008 (project HEARTRACK)

I would like to thank Prof. Jorge Marques and Dr. Jacinto Nascimento for all the help, guidance, suggestions and most of all patience and dedication, without whom this thesis would not have been possible.

A special thank to Catarina for all the help and assistance she contributed to this work, and for the care and dedication I so much need.

I would also like to thank Osório, for the help and suggestions, and all my colleagues in room E5.16 for their assistance. Thanks to all my friends for their affection and good times. Finally, I must thank my family for all the patience, support and understanding.

Contents

List of Tables

List of Figures

Acronyms

2D	Two-dimensional
3D	Three-dimensional
CT	Computerized tomography
GT	Ground truth
GVF	Gradient Vector Flow
MRI	Magnetic resonance imaging
PDAF	Probabilistic Data Association Filter
S-PDAF	Shape Probabilistic Data Association Filter

Chapter 1

Introduction

1.1 Motivation

Heart diseases are one of the major causes of death in the world. The sedimentary life-style and the bad eating habits that humans have adopted, particularly in developed countries, have significantly increased the incidence and the mortality rates of cardiopathies.

The visualization of the human heart during a cardiac cycle has proven itself a useful diagnostic tool in the clinical site. It allows cardiologists to detect anatomical abnormalities, such as congenital heart disease, and enables the determination of quantitative measures of the heart's performance such as the ejection fraction, which is the amount of blood pumped out of either one of the ventricle chambers in each heart beat. The ejection fraction is computed using the ventricle volume variation throughout the cardiac cycle and it is commonly used to assess cardiac function.

In clinical diagnosis, special interest has been paid to the analysis of the left ventricle (LV) chamber. The LV is responsible for pumping blood through blood vessels to most of the human body and abnormalities in its function can have severe consequences to human life.

Evaluation of the LV is traditionally performed by a cardiologist, who segments the endocardium (inner border) of the LV. Then, these segmentations are used to compute quantitative measures of the heart function such as the ejection fraction. Using this quantitative information, cardiologist also assess the performance of the LV, which is essential to the prognosis of some diseases such as ischaemic heart disease or acute myocardial infarction.

1.2 Echocardiography

There are several ways of visualizing the heart, but the most frequently used is the ultrasound echocardiography. Echocardiography is an imaging technique that uses ultrasound to obtain images of the heart. It is frequently used by cardiologists to assess patients with heart disease [13].

Echocardiography presents some advantages over other imaging techniques (e.g. magnetic resonance imaging (MRI) and computerized tomography (CT)): it is versatile and portable, enabling examinations without the need to move patients; it does not involve any kind of ionizing radiation, which makes it ideally for patients with chronic heart diseases that require frequent analysis; and finally, its lower costs and faster acquisition rates [31] also makes it appealing to clinicians.

Cardiologists use echocardiographic images to assess and diagnose several heart conditions. Depending on the acquisition view, a lot of information can be obtained by ultrasound imaging of the heart. One of the main echocardiography target is the analysis of the left ventricle, due to its relation to the overall performance of the heart as an oxygenated blood provider. However, not all heart diseases have symptomatic effects on that particular chamber. For instance, echocardiographic assessment of the right ventricle is also of great importance for the assessment of myocardial performance, right ventricle failure, among others.

Nowadays, two types of echocardiographic modalities are available. The classic two-dimensional (2D) echocardiographic is still the most widely used by cardiologists. However, technological advances in the ultrasound imaging technique led to the development of a three-dimensional (3D) real-time echocardiography, which is slowly gaining importance in cardiac examinations [13]. This new technique enables an entirely different perspective of the heart anatomy and function that was impossible with the 2D modality.

However, as all other ultrasound images, echocardiography produces low quality images, characterized by several image artifacts, such as: speckle, which is a random interference pattern; edge dropout and shadowing artifacts, due to poor beam penetration; low signal-to-noise ratio; and low contrast between areas of interest (see Fig. 1.1). These artifacts make echocardiography very hard to interpret. Consequently, only experts are able to correctly locate the LV boundary and even among experts the boundary location may not be consensual.

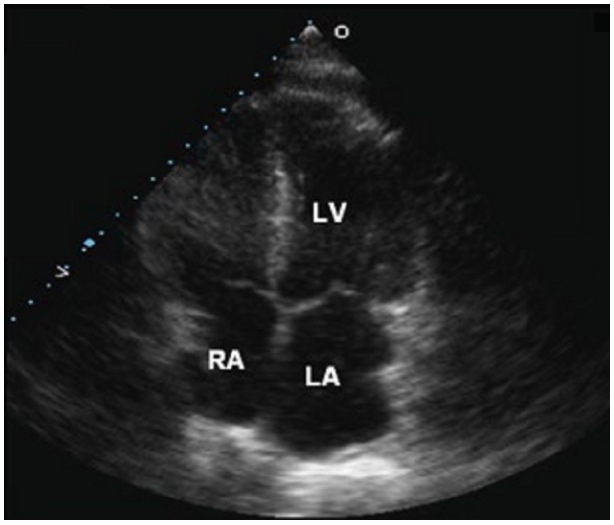


Figure 1.1: Echocardiography - apical four-chamber view [13].

1.3 Left Ventricle Segmentation

The manual segmentation of the LV that cardiologists have to perform presents two difficulties: it is a tedious and time demanding task that can only be performed by a specialized clinician; and repeating the procedure often leads to different results [23]. These issues can be solved using an automatic LV segmentation system, decreasing the segmentations' variability and improving workflow in the clinical site.

Several automatic LV segmentation system have been proposed. Initially, these would focus on single slice segmentation, simulating what cardiologists do. They look at end-diastole (maximum expansion) and end-systole (maximum contraction) frames to compute measures such as the ejection fraction and cardiac output. However, the increasing interest in 3D echocardiography segmentation fostered the development of 3D segmentation systems.

Many approaches have been used successfully, such as deformable curves and learning methods. These first ones use a deformable curve that is attracted to the LV boundary. A deformable curve can either be represented explicitly as points in a contour or using splines. Furthermore, these can be parametric, such as active snakes [32], or geometric, characterized by a level set [50]. Both of them use image information such as intensity features or image gradient to determine the location of the desired boundary and to attract the model towards it. The learning methods [24, 11], on the other hand, address the problem using learning algorithms. Based on a large training set, they learn to identify the location of the LV boundary.

Recently, several 3D automated segmentation system have been proposed. Of these new methods, some still address this segmentation as a 2D problem, analyzing each slice using 2D models and combining all

slices to form a 3D surface. However, these approaches often require additional methods to guarantee the coherence of the obtained 3D surface. The use of deformable surfaces has been avoided since they result in slower algorithms due to the amount of data involved. Nonetheless, deformable surfaces have already been used successfully.

All the mentioned approaches present difficulties. The low quality of ultrasound images leads to the detection of many misleading features (generated by the background). These hamper the performance of most traditional approaches that use deformable contours. And although learning methods are more robust to imaging conditions, they require a large training set, which may not always be available, and they may fail to perform correctly in unexpected situations such as abnormalities in the heart anatomy that are not represented in the training set.

1.4 Thesis Objective and Organization

The objective of this thesis is to develop a system that performs the segmentation of the LV in 3D echocardiography. The developed method is an extension of the Shape PDAF (S-PDAF) [46] to 3D data.

The segmentation system combines a deformable model with a robust shape estimation technique to perform the detection and estimation of the LV boundary and the adaptation of the model towards it.

The system can be decomposed into three main parts:

- The first part concerns the surface model itself. As will be explained in Chapter 3, the deformable surface has the ability to adjust to the image data and it provides a flexible platform for the segmentation of the LV;
- The second part is the detection algorithm, that is responsible for detecting the LV boundary within the image. This part will be discussed in Chapter 4;
- The third part addresses the robust model estimation technique is inspired in the S-PDAF. This task, described in Chapter 5, allows the model to perform well even in the presence of a noisy environment.

In each Chapter, a brief introduction will be provided explaining the reader the difficulties encountered in each part of the developed segmentation system.

Before describing the developed system, Chapter 2 will present an introduction to deformable models and the various approaches used in segmentation systems. It also reviews the state of the art in LV segmentation systems and how the proposed model distinguishes from previously proposed approaches and presents an overview of the proposed system, explaining the idea behind it.

Chapter 6 presents results of the application of the developed segmentation system to synthetic data and results of the application to echocardiography data, as well as a quantitative evaluation of the proposed method. A comparison between the robust estimation approach used and the classic estimation technique is also presented. Finally, Chapter 7 concludes this thesis with final remarks and future work.

1.5 Contributions

Three main contributions are made in this thesis:

1. **3D segmentation system**: a 3D segmentation system is developed, which is an extension of the S-PDAF algorithm [46] to 3D. The segmentation system uses a 3D deformable model and a robust model estimation technique to segment objects in a cluttered environment. A data association scheme is used to assign a confidence degree to features detected in the vicinity of the surface model; all features are used in the adaptation but each with a different weight based on their association probabilities. This approach leads to a flexible deformable model that performs well in 3D cluttered images.
2. **Labeling algorithm**: a new labeling algorithm is proposed for a multi-feature detection situation, in a neighborhood system. The algorithm groups the detected features assigning each a specific label based on their distance from each other and to the surface. This algorithm can be used to cluster low-level features into middle-level features, which helps differentiate them based on their origin.

3. **Application to the LV segmentation**. The 3D segmentation system combined with the proposed labeling algorithm can be used to perform the 3D segmentation of the LV boundary in echocardiographic data. The proposed segmentation system is able to deal with the low-quality of ultrasound images due to: the labeling algorithm, used to build middle-level features from the detected ones; and the robust estimation technique, which improves the robustness of the model, resulting in good 3D LV segmentations.

Chapter 2

Background

Image segmentation is an essential tool in image processing. It consists in finding a curve (or surface) that represents the contour of an object present in an image and it is used in a wide variety of applications.

Deformable models are one of the most popular approaches to perform image segmentation. In this chapter, a brief introduction to deformable models will be presented, followed by a section that reviews the state of the art in LV segmentation systems.

2.1 Deformable Models

Deformable models are curves or surfaces that can deform to fit an object. They can be represented explicitly by a set of points or using splines and their ability to adjust and move towards desired positions makes them very useful in image segmentation systems.

In 1987, Kass *et al.* developed a deformable model, called snakes [32], that was able to accurately fit an object's boundary. Unlike other existing approaches at that time, the snakes did not incorporate all the information available in a single contour, which usually led to meaningless results. Snakes consist of an elastic curve that moves according to an energy minimization scheme. The energy function is defined so that it reaches a minimum when the model fits the desired object.

Most deformable models are defined by a curve (or surfaces) $\mathbf{v}(s) = (x(s), y(s))$, where each $\mathbf{v}(s_i)$ defines a point on the curve. The energy of the curve is typically determined by

$$E = E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s)) \tag{2.1}$$

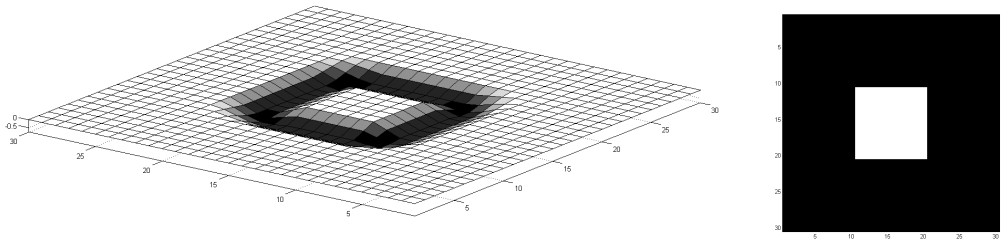
This energy function combines both an internal energy of the curve, where the shape constraints are incorporated, and an external energy that is used to attract the model to the object boundary. The definitions of the term E_{int} depends on the information of the object: if the shape of the object is known, this term can be used to prevent the model from assuming a different shape from the one expected, but it can also be used just to guarantee a smooth and consistent surface. The second term E_{ext} derives from the image data and it reaches a minimum when the deformable model fits a feature of interest, such as the object edge. The energy function minimum is achieved when the curve model lies in the object's boundary without compromising the internal properties of the model.

A typical external energy designed to lead a deformable model contour towards image edges is the integral of the potential function:

$$P = |\nabla I(x, y)|^2 \tag{2.2}$$

This forms an energy valley along the object edge (see Fig. 2.1), leading the curve towards the valley in order to minimize the energy function.

The minimization of the curve's energy (in the discrete form) can be obtained using a force equilibrium equation:



(a) (b)

Figure 2.1: (a) Example of the edge detection potential function of the image in (b).

$$\mathbf{F}_{int} + \mathbf{F}_{ext} = 0 \tag{2.3}$$

where the internal force $\mathbf{F}_{int}$ is defined with the same purpose of E_{int} (i.e., adding shape constraints to the curve), and the external force $\mathbf{F}_{ext}$ is image dependent and pulls the contour towards the desired image edges.

One of problems of traditional external forces such as the one in (2.2) is that image edges have a short influence, i.e. they only attract the elastic contour when it is close to the potential valley. This motivated a large number of contributions from the image analysis community.

Chenyang Xu, for instance, introduced an upgraded version for the external force of the snakes model, called gradient vector flow (GVF) [61]. One of the ideas behind this new method was to eliminate some of the difficulties in the initialization of deformable contours, associated with the limited range of the commonly used potential functions. As shown in Fig. 2.1, the value of these functions beyond the vicinity of the edge is zero, which means it does not have any influence on the contour. Xu overcame this issue by developing a potential function that extends its influence to the whole image. As a result, the deformable contour no longer requires the initialization to be near the object boundary.

Parametric vs Geometric

Deformable models may be classified as either parametric deformable models or geometric deformable models. Parametric deformable models [32, 15, 18, 63, 61] are the oldest of the two formulations. They are represented explicitly as parameterized contours such as the previously described snakes and use internal and external forces to fit the model to the object boundary.

Geometric deformable models [12, 38, 65, 50], on the other hand, are represented implicitly as level sets and evolve in an Eulerian fashion. These models are based on the curve evolution theory in differential geometry. The level set method was developed by Osher and Sethian in [51] as a method for computing and analyzing the motion of the border of a region Ω. They defined the level set function $\varphi(\mathbf{x})$ that may take the following values: $\varphi(\mathbf{x}) > 0$ if $\mathbf{x}$ is located inside the region Ω, $\varphi(\mathbf{x}) < 0$ if $\mathbf{x}$ is outside Ω and finally $\varphi(\mathbf{x}) = 0$ if $\mathbf{x}$ is located in the border of the region Ω. Then, finding the border of Ω corresponds to finding the curve that yields $\varphi(\mathbf{x}) = 0$.

Geometric models have some advantages over parametric models. First, it is not necessary to add or remove nodes from the contour to keep the nodes evenly distributed. Second, the discretization of the contour in parametric models leads to inaccuracies in the computation of geometric properties such as the normal vector and the curvature of the contour, which does no happen in geometric models. And third, geometric models are able to automatically change its topology, which is very appealing in situations where the object topology is unknown. However, this ability can also be a liability in applications where the object to be segmented has a known topology that must be preserved.

Bottom-up vs Top-down

Deformable models can use either one of two approaches: bottom-up and top-down. These vary according to the prior knowledge about the desired object that is introduced in the model.

Bottom-up approaches are based on the detection of features within the image and associating those feature in order to obtain the segmentation of the object. This type of approach does not use prior knowledge of the object in the feature extraction step and, therefore, requires an image wide search for features. The detected features are grouped according to certain characteristics (such as image intensity) and relying on continuity principles. Difficulties is these approaches arises when noisy images produce many features and the resulting segmentation may be mislead by the background noise.

On the other hand, top-down approaches make use of prior knowledge about the object to guide the segmentation. These often use a local feature search based on the contour location, such as a directional search within the vicinity of the model. This approach deal better with the images' background noise, but relying too much on the prior knowledge often leads to inaccurate results in unexpected various of the object.

2.2 3D LV Segmentation Systems

Several approaches can be used to develop an automated LV segmentation system. The most successful LV segmentation systems are based on the following techniques: active contours [5, 19, 21, 32, 36, 37, 38, 53, 52, 47, 55], deformable templates [14, 30, 42, 67], and supervised learning methods [8, 11, 10, 9, 17, 16, 24, 43, 68, 69]. The first two are part of the deformable models category mentioned in the previous section and have proved good tools to segment the LV in echocardiographic images.

Supervised learning methods are part of a different category that attempts to segment images through learning procedures based on pattern recognition in training sets. Contrary to deformable models, these methods are more robust to image conditions. However, they too present a few challenges: they require a large set of training images for the learning procedure, which usually is not available; and they are not able to perform accurately when presented with a situation no present in the training set [11]. These issues have limited a more extensive exploration of supervised models for the LV segmentation problem.

Initially, LV segmentation systems were developed for 2D echocardiography. When 3D echocardiography became available, one of the most typical approaches was to sequentially apply 2D segmentations to each image plane and assembling them into a 3D structure - as cardiologists manually do in such cases. However, such algorithms usually require additional methods to guarantee the coherency of the 2D contours along the third dimension.

An example of this type of approach is Nillesen *et al.* [48] work, which performs a 3D segmentation of the LV by segmenting each 2D slice of the volume. They tackle the low quality of the ultrasound images using image filtering techniques to reduce image noise and enhance the distinction of the LV border. Then, a deformable contour algorithm is used to segment each plane. To avoid inconsistencies and to guarantee spatial continuity of the segmentation, each contour is adjusted using the surrounding contours.

Much like Nillesen, Scowen *et al.* [56] describes a model that creates the 3D representation of the LV from 2D contours. The 2D contours are obtained by manually segmenting the LV in some slices and then automatically aligning them to form the 3D surface. The automatic alignment is obtained using anatomical features as guiding references.

The need for segmentation systems that perform at acquisition rates has increased, which led to the development of fast 3D segmentation and tracking systems. Comaniciu *et al.* [64], for example, proposes a 3D LV tracking algorithm. It combines both learning methods for the initialization of the model and a tracking algorithm to predict the location of the LV in the next frame. Filtering techniques are used to prevent temporal inconsistencies.

Orderud [49], on the other hand, used the Kalman filter to track structures in volumetric data with deformable models. Combining edge-detection with speckle-tracking, he successfully performs the segmentation and tracking of the LV in real-time.

On a different approach, Juang *et al.* [31] proposes an automatic segmentation system that makes use of the radial symmetry of the LV to transform the volume to the cylindrical coordinate space. This results in an image where the LV boundary divides the image in two parts: the LV chamber on one side and exterior on the other side. Afterwards, they transformed the segmentation back to the Cartesian coordinate space, resulting in a 3D segmentation of the LV.

Xu's GVF, extensible to any dimension, also appealed to some LV segmentation systems researchers. For instance, both Hang *et al.* [28] and Honggang *et al.* [66] developed a 3D segmentation system that consists

in combining the 3D GVF technique with a level set method.

However, the difficulties underlying automatic segmentation systems also led to the development of interactive segmentation systems, such as Boettger *et al.*'s model [7]. He developed a segmentation tool, that interactively enables user correction of the deformable model's development by incorporating a user defined force in the deformable model.

The proposed segmentation system in this thesis uses a different approach from the heretofore mentioned systems. It consists a 3D deformable model that relies on the detection of features in the vicinity of the model. Since ultrasound images often lead to the detection of many undesired features, two approaches are used to increase the system's robustness:

1. All detected features are grouped into patches to form middle-level features; these middle-level features are more robust than the low-level (detected) ones because they represent a portion of a 3D structure the was detected in the volume. This helps discriminate which features were produced by the background and which belong to the LV boundary. Furthermore, it greatly decreases the number of detected features, which improves the performance of the algorithm and its capability;

2. a robust estimation technique is used, inspired in the S-PDAF, that consists in determining all possible interpretations of the middle-level features and using a data association scheme to assign each interpretation a confidence degree. This enables the determination of each feature's weight (strength) in the adaptation procedure, making the model more robust to outliers.

This method does not use prior image filtering techniques to enhance the LV boundary and it ensures that the surface fits the LV even in the presence of unexpected anatomical abnormalities.

2.3 System Overview

The idea behind the developed method was to tackle the difficulties of classic deformable contour methods associated with noisy data (such as echocardiographic volumes) by introducing a robust estimation scheme. This robust estimation scheme is inspired in the S-PDAF [46], developed for shape tracking in cluttered environments, which is rooted in the seminal work of Bar-Shalom [3]. Here we extend it to the context of 3D segmentation.

The proposed segmentation system uses a 3D deformable model to characterize the surface of the segmentation. This deformable surface requires an initialization procedure that ensures it is initialized in the vicinity of the LV boundary, but it provides a flexible platform for the segmentation of the LV.

The adaptation procedure is an iterative process that consists of the following steps: after initialization of the model, an adaptation cycle begins with the detection and extraction of low-level features, searched in the vicinity of the model. Then, these are grouped into middle-level features (patches). Based on the assembled patches, the S-PDAF algorithm determines all possible interpretations of considering a patch valid or invalid and assigns each a confidence degree that is used to define the estimate of the boundary location. The model estimate is then used to fit the surface to the LV boundary, ending an iteration of the adaptation cycle. The process repeats until the surface is considered close to the LV boundary. The following figure shows a diagram of the adaptation cycle.

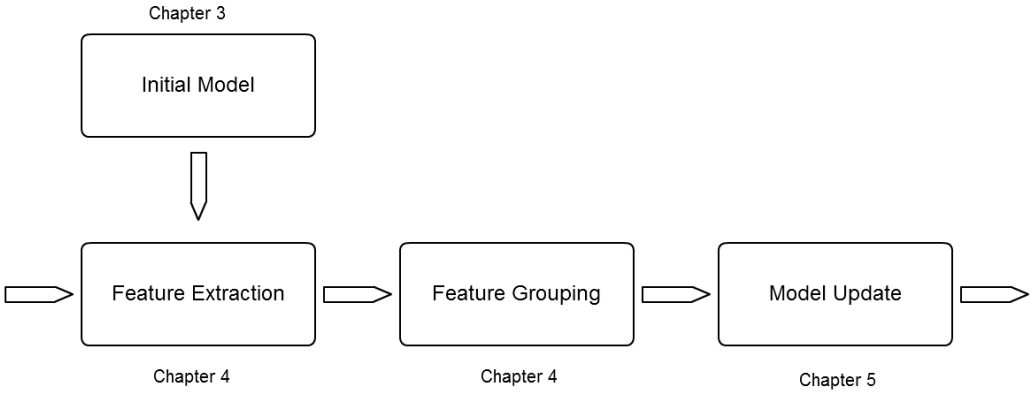


Figure 2.2: Diagram of the proposed segmentation system.

Chapter 3

Surface Model

This chapter addresses the representation of a 3D object. Section 3.1 presents a brief introduction on the various ways of representing a 3D object. Section 3.2 describes the adopted deformable surface and Section 3.4 addresses its initialization.

3.1 Introduction

To represent the segmentation of the LV, a 3D model has to be used. Almost all 3D models can divided into two categories: solid models, that define the set of points inside the object they represent; and boundary models, that represent the surface of the object. Although solid models are more realistic, surface models have been extensively used in segmentation systems because they are easier to work with.

As discussed in Chapter 2.1, one of the most commonly used surface models in segmentation systems is deformable models. These have the ability to move and adapt to fit the data, which makes them extremely useful in segmentation systems. Several types of deformable surface models have been proposed.

In [41], a review of the major deformable models used in medical images is presented. One example is a finite element surface model developed by McInerney and Terzopoulos [40]. This parametric model consists in a finite-element sphere defined by a surface spline. This method represents the surface as a sum of polynomial functions.

Another example is Staib and Duncan's parametric Fourier model [58]. In this case, the geometric surface is defined by a sum of sinusoidal functions. The adaptation scheme is governed by a gradient function that attracts the surface to image edges.

Alternatively, Szeliski used an oriented particle system [59] to model the object's surface. Each particle is a discretization of the surface and they are governed by Newtonian mechanics, i.e., under the influence of external and inter-particle forces. The external forces attracts the particles to the data while inter-particle forces are meant to preserve the structure and coherency of the surface.

Similarly, Delingette's simplex mesh [22] consists of a meshed surface that is defined by vertices and edges. The surface also deforms based on Newtonian mechanics. The external force applied in each vertex results from image features and attracts it towards the object boundary, while the internal force assures that the surface remains smooth and coherent.

The difference between these last two models lies in the structure of the surface. While in Delingette's simplex mesh the structure of the surface is also defined by the edges linking its vertices, in the oriented particle system the structure is not explicitly defined. In other words, simplex mesh has a regular structure, contrary to Szeliski's model. As a result, the definition of some surface parameters such as a tangent plane and its normal is not possible in the oriented particle system.

Table 3.1 presents a summary of the most interesting characteristics of the mentioned models. While the first two models have the advantage of requiring fewer resources to describe the model (since they are fully characterized by the parameters of the surface splines), the non-parametric ones are more intuitive and easy to handle surface models.

Characteristic	Surface	Required Information	Surface normal computation
Finite-element[40]	finite-element	spline parameterization	simple
Parametric Fourier[58]	finite-element	Fourier parameterization	simple
Oriented particle[59]	particles	particle location	complex
Simplex mesh[22]	vertices and edges	vertex location and its neighbors	simple

Table 3.1: Summary of the deformable models' characteristics.

Simplex meshes were considered the best choice for this work. On one hand, the structure of the simplex mesh allows a simple computation of the tangent plane to the surface at any vertex and its correspondent normal vector. On the other hand, the simplicity of the model and its adaptation methodology makes it easier to define the surface and to control its adaptation.

3.2 Simplex Mesh

The simplex mesh, proposed by Delingette [22], is a deformable surface that is capable of representing objects of several topologies. It can be used in different dimensions and it is dictated by few and intuitive concepts. Simplex meshes have been used in several reconstruction and segmentation systems [22, 20, 62, 44] with success, particularly in 3D problems.

In the 3D space, simplex mesh is a connected meshed surface where each vertex belongs to three facets, i.e., it forms three edges with its neighboring vertices (see Fig. 3.1). This characteristic defines the whole structure of the surface and it enables the definition of geometrical relations that guide the model's deformation.

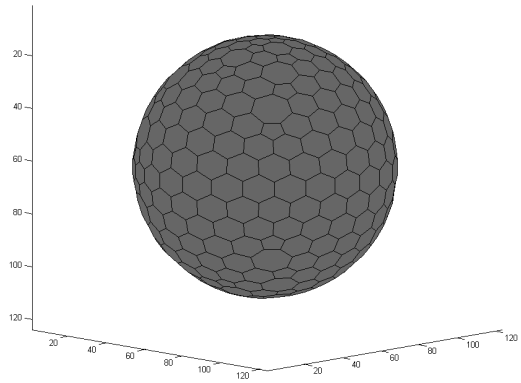


Figure 3.1: Example of a simplex mesh.

Geometry of Simplex Meshes

One particular aspect of the simplex meshes structure is that it allows to define rather simply a tangent plane and, consequently, define the normal to the surface at a given vertex. Let P_i be a vertex of a 3D simplex mesh S and let $(P_i^{N_1}, P_i^{N_2}, P_i^{N_3})$ be the three neighboring vertices of P_i. These neighbors form a triangle that defines a tangent plane to the surface at P_i and the vector $\mathbf{n}_i$ normal to the tangent plane is defined as:

$$\mathbf{n}_i = \frac{(P_i^{N_1} - P_i^{N_2}) \times (P_i^{N_2} - P_i^{N_3})}{||(P_i^{N_1} - P_i^{N_2}) \times (P_i^{N_2} - P_i^{N_3})||} \tag{3.1}$$

Moreover, there are two main geometric relations in a 3D simplex mesh: the simplex angle and the metric parameters. These two geometric relations can be used to define the position of a vertex relative to its three neighbors.

Let S_1 be a circle of center C_i and radius r_i circumscribed by the triangle ($P_i^{N_1}$,$P_i^{N_2}$,$P_i^{N_3}$). Let S_2 be a sphere of center O_i and radius R_i, circumscribed by the four vertices (P_i,$P_i^{N_1}$,$P_i^{N_2}$,$P_i^{N_3}$) (see Fig. 3.2(a)). Projecting these onto the place (O_i, C_i, P_i) enables the visualization of the simplex angle ϕ_i, as shown in Fig. 3.2(b). The computation of the simplex angle, $\phi_i \in [-\pi, \pi]$ is governed by the two equations [22]:

$$\begin{aligned} \sin\phi_i &= \tfrac{r_i}{R_i}\text{sign}((P_i^{N_1} - P_i) \cdot \mathbf{n}_i) \\ \cos\phi_i &= \tfrac{||C_i - O_i||}{R_i}\text{sign}((C_i - O_i) \cdot \mathbf{n}_i)\text{sign}((P_i^{N_1} - P_i) \cdot \mathbf{n}_i) \end{aligned} \tag{3.2}$$

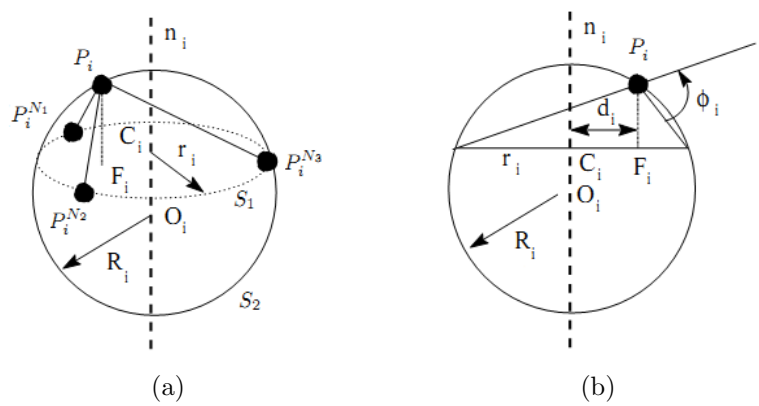


Figure 3.2: Simplex angle [22].

The metric parameters are the other important geometrical quantity. Let F_i be the orthogonal projection of P_i onto the triangle ($P_i^{N_1}$,$P_i^{N_2}$,$P_i^{N_3}$). The metric parameters at a vertex P_i are the barycentric coordinates of F_i [22]:

$$F_i = \epsilon_{1i}P_i^{N_1} + \epsilon_{2i}P_i^{N_2} + \epsilon_{3i}P_i^{N_3} \tag{3.3}$$

$$\epsilon_{1i} + \epsilon_{2i} + \epsilon_{3i} = 1 \tag{3.4}$$

The simplex angle, ϕ_i, is related with the curvature of the surface at P_i, whereas the metric parameters, ϵ_i, are related with the vertex distribution throughout the surface.

3.3 Law of Motion

The deformation scheme used by simplex mesh combines both the influence of the image as well as the relative position between vertices. The final configuration of the vertices is achieved using an iterative process defined by [22]:

$$P_i(k+1) = P_i(k) + (1-\gamma)(P_i(k) - P_i^(k-1)) + \alpha\mathbf{F}_i^{int}(k) + \beta\mathbf{F}_i^{ext}(k) \tag{3.5}$$

where γ, α and β are constants. In order to guarantee the stability of the adaptation process, the parameters α and β must belong to the interval $]0, 1]$. These constants influence the intensity of the internal and external force and, consequently, they determine the speed and smoothness of the process.

The internal and external forces are described in the following subsections.

Internal Force Computation

The internal force adopted in this work is the one proposed in [22]. In each vertex, it can be decomposed in two orthogonal forces: a tangential force $\mathbf{F}_i^T$, parallel to the tangent plane, and a normal force $\mathbf{F}_i^N$, parallel to the normal vector $\mathbf{n}_i$.

The tangential internal force is responsible for the position of the projection, F_i, of the vertex within the tangent plane:

$$\mathbf{F}_i^T = \tilde{F}_i - F_i = (\tilde{\epsilon}_{1i} - \epsilon_{1i})P_i^{N_1} + (\tilde{\epsilon}_{2i} - \epsilon_{2i})P_i^{N_2} + (\tilde{\epsilon}_{3i} - \epsilon_{3i})P_i^{N_3} \tag{3.6}$$

where $\tilde{F}_i = \tilde{\epsilon}_{1i}P_i^{N_1} + \tilde{\epsilon}_{2i}P_i^{N_2} + \tilde{\epsilon}_{3i}P_i^{N_3}$ is the desired position of the F_i. This internal tangential force imposes a particular vertex distribution throughout the surface. Thus, to assure a uniform distribution, in which all the vertices are equally spaced to their neighbors (i.e., in the center of the triangle), the reference parameters should be $\tilde{\epsilon}_{1i} = \tilde{\epsilon}_{2i} = \tilde{\epsilon}_{3i} = 1/3$.

On the other hand, the normal internal force adjusts the curvature at the vertex P_i, by imposing desired simplex angle, $\tilde{\phi}_i$:

$$\mathbf{F}_i^N = (L(r_i, d_i, \tilde{\phi}_i) - L(r_i, d_i, \phi_i))\mathbf{n}_i \tag{3.7}$$

where L is defined as in [22].

The value of the reference simplex angle depends on the desired rigidity of the surface. In this thesis, the reference simplex angle used is an average of the simplex angles of the nine surrounding vertices, as shown in Fig. 3.3. Let $N^{P_i} = \{P^{N_1}, P^{N_2}, P^{N_3}\}$ be the set of neighbors of P_i; the reference simplex angle is defined by:

$$\tilde{\phi}_i = \frac{1}{9}\left(\sum_{P_j \in N^{P_i}} \phi_j + \sum_{\substack{P_k \in N^{P_j} \\ P_k \neq P_i}} \phi_k\right) \tag{3.8}$$

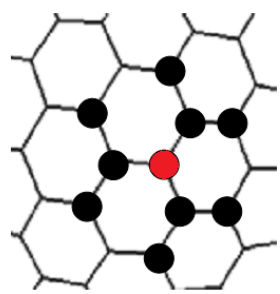


Figure 3.3: Vertices used (black dots) to determined the reference simplex angle of P_i (red dot) [22].

This means that the model tries to smooth the surface (i.e. avoids spikes), but it is flexible enough to avoid withholding the adaptation to a consistent dataset.

The final internal force at P_i is a sum of the tangential and normal forces.

External Force Computation

The purpose of the external force is to pull the vertices of the simplex mesh surface towards points located on the object boundary. This means the external force is a displacement (vector) that points towards these points. The collinearity between the external force and the normal vector is important for the stability of the adaptation process [22]. To avoid surface conflicts, this collinearity will be guaranteed as will be explained

further ahead in Chapter 4. Another way to avoid these conflicts is to used small value of β in Equation (3.5), so that the displacements of the vertices are small in each iteration and allow the internal force to smooth the surface and maintain its structure.

3.4 Model Initialization

The initialization has always compromised many deformable models, since these have to be initialized in a close vicinity of the dataset for the segmentation to perform well. The model used in this thesis is no exception.

The initialization procedure has to meet the following conditions:

1. The initial model should be initialized in the vicinity of the LV boundary;
2. and it should be a simplex mesh.

These two conditions are met using a three step procedure. First, to ensure that the model is initialized in the vicinity of the LV boundary, the user manually defines a coarse outline of the LV in three orthogonal planes. A 3D region is then obtained by space carving [35]. Second, the simplex mesh is initialized as a sphere in the center of the carved volume. Finally, we let the simplex mesh deform until it fits the carved region, which corresponds to the silhouette of the LV boundary, simplifying and initial adaptation of the model.

The creation of the carved volume and the initialization of the spherical simplex mesh will be explained in the next subsections.

Computation of the Carved Volume

The space carving algorithm consists in carving (cutting) a volume with the shape of an object when viewed from specific viewpoints. A simple analogy is forcing a hallow cylinder through a wax cube; the result is two wax pieces: a wax cylinder and the remainder of the cube.

To apply this idea, first we need to determine what is the LV silhouette ("cylinder") in the echocardiographic volume. This is where user interaction is required: the computation of the carved volume begins with a manual segmentation of the echocardiography.

Three different perspective are used, each corresponding to a different reference plane (sagittal, coronal and transverse planes). The best way of visualizing the LV's silhouette in the echocardiography is either by sampling slices of the volume that intersect the LV or by projecting the volume into a 2D image. Then, the user performs the manual segmentation of the LV in that image.

This manual segmentation results in three 2D images with a segmented area. To created a carved volume, these images have to be extruded and their information combined. This means that, for each image, a carved volume is created in which each slice has the segmented area. Then, all three carved volumes are intersected with each other, as shown in Fig. 3.4.

The intersection of the carved volumes results in the creation of a binary volume where the value transition corresponds to the silhouette of the LV. This new binary volume is used by the segmentation system to adjust the initial sphere to a configuration that resembles the LV in the real echocardiographic volume.

Fig. 3.5 shows an example of the manual segmentation of the LV using middle slices of each reference plane of the volume and Fig. 3.6 shows the corresponding 3D carved volume (surface model after fitting the carved volume).

Spherical Simplex Mesh

After the creation of the carved volume, the surface model is initialized as a sphere, where the position of the sphere's center, C, is determined by the centroid of the carved volume, and its radius, r, is proportional to the average of the maximum radius of each of the manually segmented images.

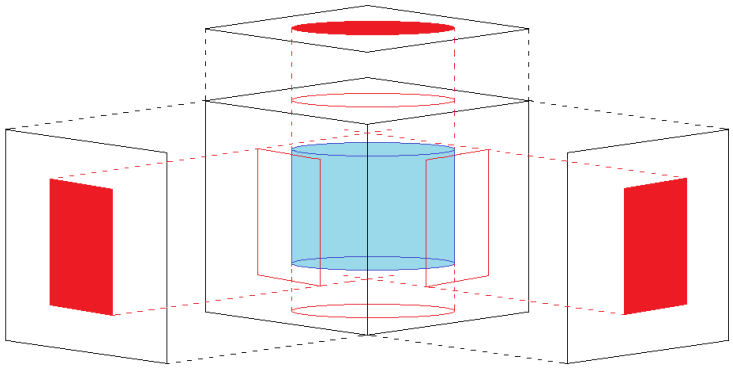


Figure 3.4: Schematic example of the computation of the carved volume (cylinder) (in light blue) by intersection of the segmented projections (in red).

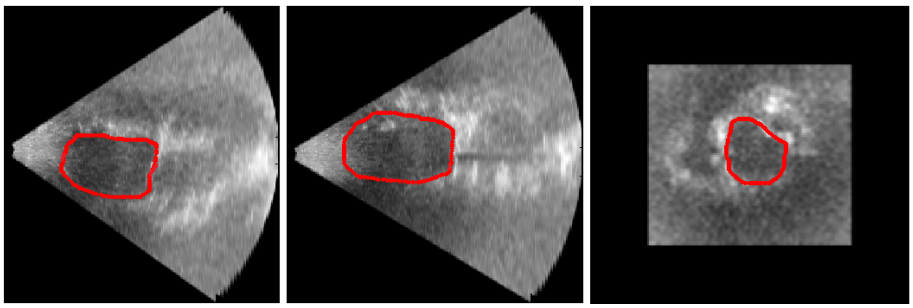


Figure 3.5: Example of the manual segmentation of the LV using middle slices of each reference plane of the volume.

To begin the initialization of the model, a sphere is uniformly sampled along each circle of latitude. Then, the discretization of the sphere has to evolve into a simplex mesh. This could be done using a Voronoi tessellation [2], which consists in creating a tessellation where each edge is equally distant from the two closest vertices. However, this algorithm has only been described for 2D images' tessellation or 3D solid volume tessellation. Since none of these is the case, an alternative had to be used.

This alternative consists in using the convex hull algorithm [4]. After sampling the sphere points, the algorithm is applied, resulting in a triangular mesh on the sphere surface. Then, taking into account the duality of the simplex mesh with triangulations [22] (see Fig. 3.7), an associated simplex mesh can be formed by considering the center of each triangle as vertices and linking each vertex with the center of the three neighboring triangles.

This initialization procedure assures that the number of vertices in each created surface is fixed (equal to the number of triangles that result from the convex hull algorithm) and that the location of the vertices on the surface of the initial sphere is always the same. Furthermore, the proposed radius and center computation guarantees that the sphere is initialized close to the dataset of the carved volume formed by the manual segmentation. Fig. 3.1 is an example of an initial simplex mesh sphere.

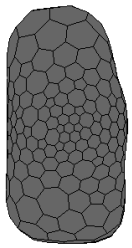


Figure 3.6: Three-dimensional view of the surface after fitting the carved volume.

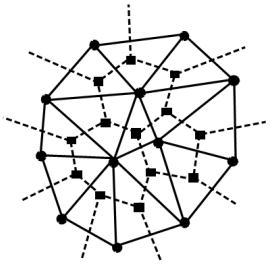


Figure 3.7: Duality between triangulation and simplex mesh [22].

Chapter 4

Feature Extraction

Feature extraction is a crucial step in any deformable model system. It aims to detect 3D points based on certain characteristics in the volume, such as gradient peaks or intensity transitions, that characterize the edge of the desired object. In this thesis, a feature extraction algorithm is used to detect 3D points that correspond to the LV boundary. However, as mentioned before, ultrasound is an imaging technique that produces low-quality images, which means extracting the desired features is a difficult task. In addition, in echocardiographic images there are multiple structures present, which means the feature extraction algorithm may also detect other structures besides the LV.

This chapter describes the feature detection algorithm in Section 4.1 and 4.2 and the middle-level features assemblage in Section 4.3.

4.1 Directional Search

Taking into account that the collinearity between the external forces F_i^{ext} and the normal vectors $\mathbf{n}_i$ is important for the stability of the adaptation procedure (as mentioned in the previous chapter), feature detection is performed along straight lines orthogonal to the surface, as shown in Fig. 4.1. Note that these lines also extend to the interior of the surface.

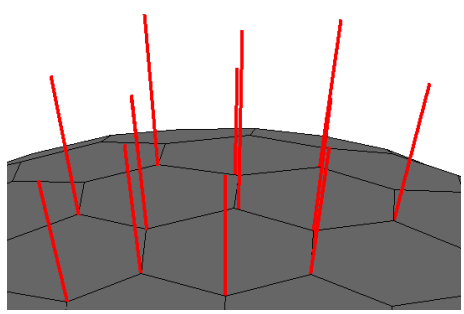


Figure 4.1: Search lines of some vertices in a simplex mesh.

We define the search line signal, $I(n)$, as the discretization of the image intensity I at sample points $(x_j, y_j, z_j) = P_i + c_j\mathbf{n}_i$ within the search line, where $c = \{-d_1, \ldots, 0, \ldots, d_2\}$ are constants; d_1 and d_2 define the length of the search line. However, we only know the values of I in discrete points within the volume, which results in two conclusions:

1. in order to determine the one-dimensional signal $I(n)$ some sort of interpolation is required;
2. the extraction of features from the 3D image now involves the analysis of a one-dimensional signal.

We use a tri-linear interpolation to determine the image intensity $I(n)$ at any 3D point (x, y, z). The tri-linear interpolation algorithm is described in [1] .

The second conclusion means that, instead of performing a volumetric search for features, the model performs several individual feature searches (as many as the number of vertices in the surface) in one-dimensional signals, which simplifies the feature detection problem.

The length of the search interval is determined by the parameters d_1 and d_2. In order to define there, it would useful to know whether the normal vectors $\mathbf{n}_i$ point outwards or inwards the surface. The initialization procedure proposed in Section 3.4 enables the definition of an initial set of normal vectors $\mathbf{n}_i^0$ that are faced outwards the sphere surface. These initial normal vectors can be used to assure $\mathbf{n}_i$ are always faced outwards as well. Consequently, the parameters of d_1 and d_2 define the length of the search inwards and the length of the search outwards, respectively. The values of these parameters have to be chosen in a way that guarantees that the LV boundary is located within the search line.

4.2 Feature Detection

Feature detection is a well-studied subject. In this work, the desired features are the ones belonging to the LV boundary. These correspond to intensity transitions - edges - present in the volume, forming a color/shade discontinuity (see Fig. 1.1). However, detecting them is not always an easy task, particularly in ultrasound data, due to background noise.

As previously defined, the search line signal $I(n)$ is a discrete signal of the image intensity along the search line. An intuitive approach for the detection of features characterized by intensity transitions is the analysis of the first derivative of $I(n)$. Whenever an abrupt transition of intensity exists, the first derivative has a peak (either a local maximum or minimum). However, this methodology does not always detect the LV boundary and, due to the image noise, it often results in the detection of too many undesired features, as shown in Fig. 4.2.

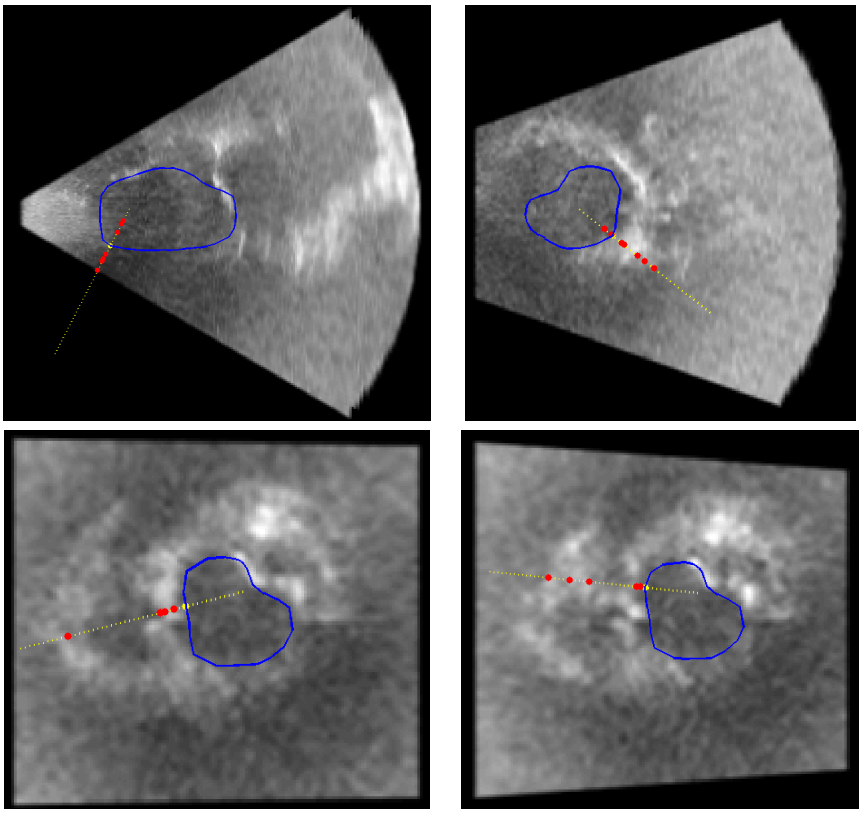


Figure 4.2: Features detected using the analysis of the first derivative of the search line signal.

We will adopt the procedure described in [6]. First the intensity profile, $I(n)$, is filtered using a filter with mask $M(n)$ of dimensions $2 \times d_f$. The filter output is given by:

$$J(n) = \sum_{m=-d_f}^{d_f} M(m)I(n+m) \tag{4.1}$$

$M(m)$ determines what type of features are detected. In this case, the mask used is defined so that it detects edges, since the LV boundary is characterized by edges in ultrasound images. The mask used is defined as:

$$M(m) = \begin{cases} -1 & \text{if } -d_f < m \leq 0 \\ 1 & \text{if } 0 < m < d_f \end{cases} \tag{4.2}$$

In Equation (4.1) and (4.2), d is a constant that defines the length of the mask and, thus, the amount of smoothing of the filter - higher values of d_f result in a convolved signal $J(n)$ less sensible to noise but also in less accurate signal peaks. The ideal value for d_f may be different for different images, depending on its noise intensity.

Maxima of the signal $J(n)$ is extracted using a threshold and a non-maximum suppression method. The threshold is chosen to include all meaningful maxima and to exclude spurious, noise-generated maxima. Fig. 4.3 shows the application of this detection method to an intensity signal of an echocardiography search line. The highest peak in the signal marks the edge of the LV boundary.

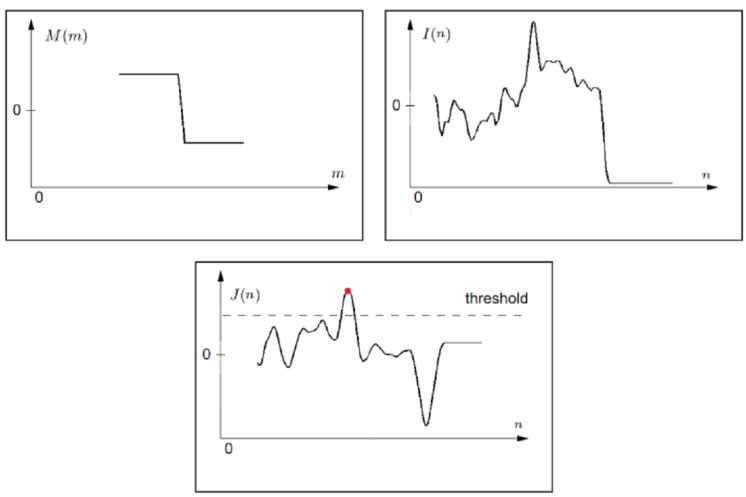


Figure 4.3: Filtered signal, $J(n)$, obtained by application of a filter with mask $M(m)$ to an echocardiographic search line signal, $I(n)$.

Other approaches were available as well, such as the application of a correlation matching algorithm, called matched filter [6]. This is similar to the previous method, but it uses the correlation of the intensity signal with a template to find the most likely locations of that template within the signal. The results would be very similar to the ones obtained using the mask $M(m)$.

As Fig. 4.4 shows, this method may also produces undesired features and it does not always detect the LV boundary. Fig. 4.5 confirms this by showing that the feature extraction algorithm often detects two feature points and sometimes three or even four, where, at best, one of them is the LV boundary. This means that it is difficult to know which feature point corresponds to the LV wall. Therefore, the segmentation system requires sophisticated techniques in order to deal with outlier data and increase the model's robustness.

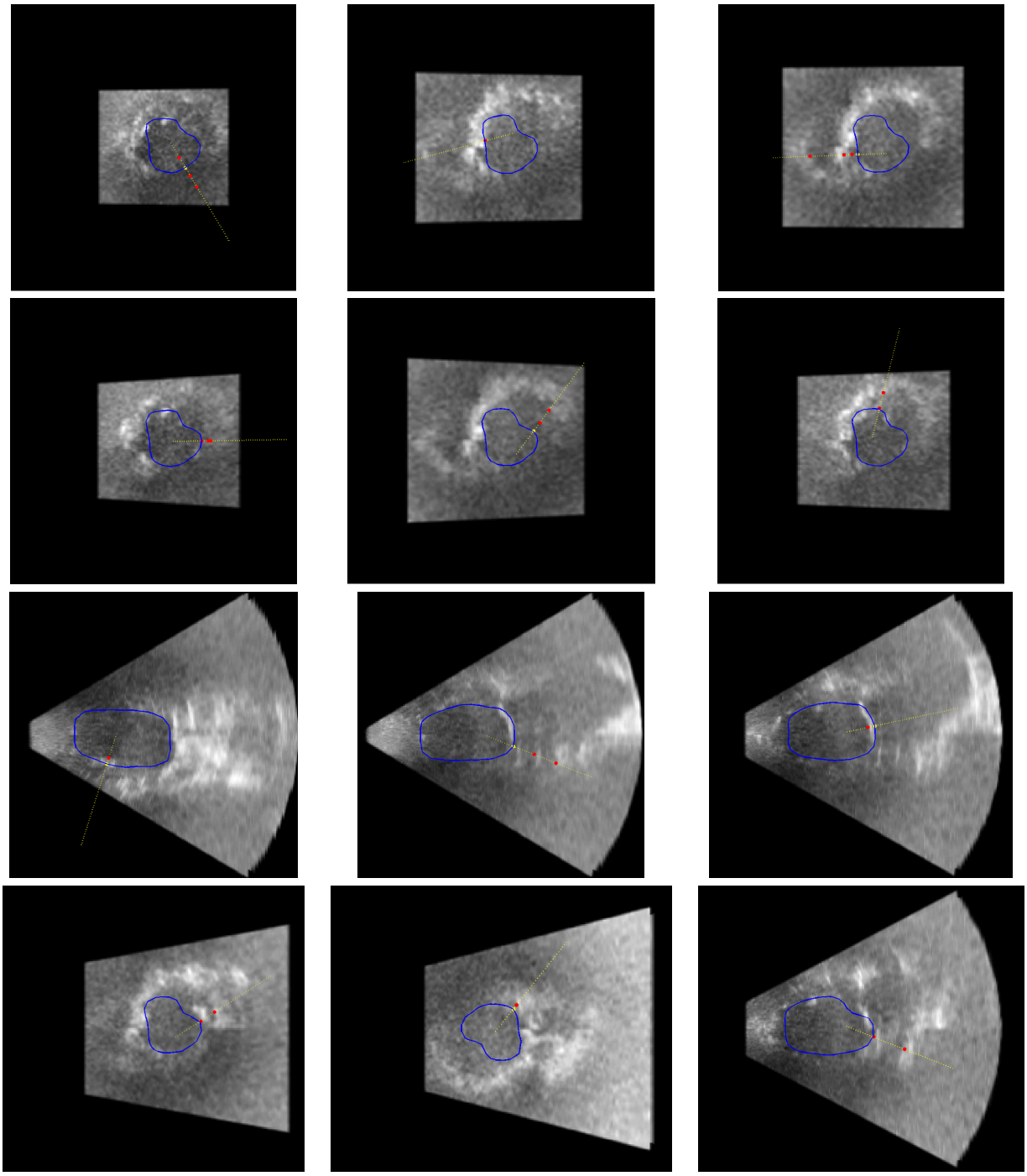


Figure 4.4: Examples of the detected features using the mask $M(m)$.

4.3 Middle-Level Features

One way of increasing the robustness of the segmentation system is by considering middle-level features assembled from the low-level features detected in the volume. This has two major advantages: it significantly reduces the number of features, and makes it simpler discriminate if the detected features belong to the LV boundary or if they were produced by the background.

This process consists in grouping low-level features, according to some criteria, to form 3D patches. Fig.

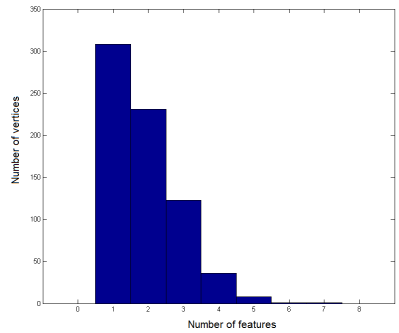


Figure 4.5: Histogram of the number of features detected in each vertex's search line.

4.6 shows an example of what could be the result of building mid-level features from the detected low-level features in a 2D echocardiography.

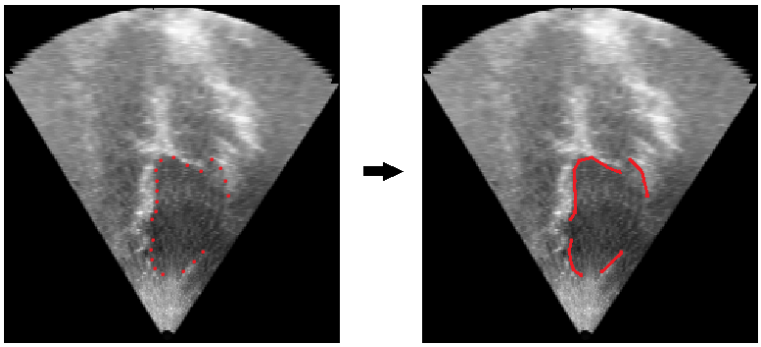


Figure 4.6: Building middle-level features in a 2D echocardiography.

Ideally, all detected features should be differentiated based on the patch they belong to. For instance, features located on the LV boundary should all belong to the same large patch, whereas noise-originated features should belong to small patches of their own. However, performing such differentiation is a difficult task for two reasons: 1) each vertex of the surface model performs an individual feature search from which can result a non-limited number of features; and 2) we do not know which features are noise-originated and which belong to the LV. The only available information is the distance between each other and to the surface and the detected features must be grouped based on this information (see Fig. 4.7).

The following sections will mathematically describe the problem and the proposed solution.

Labeling Problem

Building middle-level features is a labeling problem, meaning each feature receives a specific label and all features with the same label belong to a specific patch. In this particular scenario,

- The surface model is composed of a set of vertices $S = \{P_1, \ldots, P_{m_s}\}$, where m_s is the total number of vertices;
- For each vertex P_i there is a set of features Y_i either empty (no feature detected) or composed of y_i^k features, where $k = 1, ..., m_i$ and m_i is the total number of features associated to P_i;
- The simplex mesh's neighboring system provides a structure that helps the labeling system: we define the set Y_i^N as the set of containing all features associated with the neighborhood of P_i, $N^{P_i} = \{P_i^{N_1}, P_i^{N_2}, P_i^{N_3}\}$.

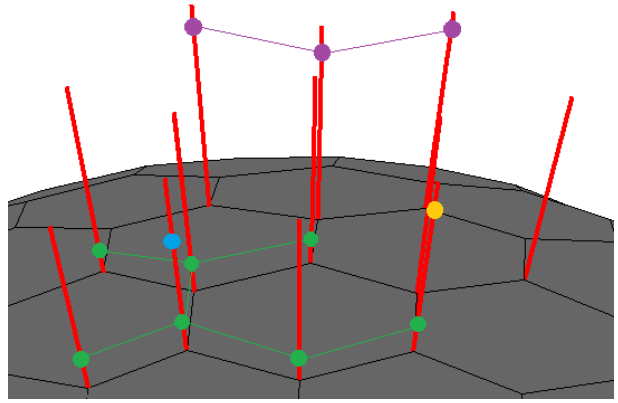


Figure 4.7: Example of middle-level features. Each color represents a different patch.

Three assumptions can be made. First, it can be assumed that each feature y_i^k associated to P_i belongs to a different edge, since it is impossible to detect the LV boundary more than once along a straight line that begins inside the LV and ends outside it. Second, it can also be assumed that every patch must have feature continuity, i.e., if more than one feature has a specific label, each feature y_i with that label has at least one neighboring feature y_j with the same label. And finally, for these neighboring features to be considered as belonging to the same patch, they must not be too far apart; i.e., the distance between y_i and y_j must not exceed a specific labeling threshold: $d = ||y_i - y_j|| < d_{max} = t_l$.

Literature on labeling algorithms that address a problem such as this one is scarce. It was recently addressed in [39], which proposes a labeling algorithm in the presence of multiple observations, similarly to the problem at hand. They want to identify which feature belongs to which sensor (edge), in two different situations: in the first one, they assume that the sensors are characterized by known probability distributions - which is not the case in this thesis; in the second situation, the sensor models are unknown and they only assume space-dependency, i.e., observations generated by the same sensor are close to each other - just as it was assumed previously for this thesis' problem.

They address the second situation using an energy function that promotes data continuity. More specifically, if two neighboring observations were produced by the same sensor, it is likely that their values are similar. The energy of the label configuration L is dictated by each individual label l_i^k assigned to the feature y_i^k. The energy function defined in [39] is:

$$E(L) = \sum_{P_i \in S} \sum_{P_j \in N^{P_i}} \sum_{k=1}^{m_i} \sum_{k'=1}^{m_j} \delta(l_i^k - l_j^{k'}) ||y_i^k - y_j^{k'}|| + E_l(L) \tag{4.3}$$

where δ is defined as $\delta(z) = 1$ if $z = 0$, and $\delta(z) = 0$ if $z \neq 0$; and $E_l(L)$ is a function that prevents repeated labels in a vertex and penalizes large number of different labels. The minimum energy is achieved when the ideal labels are assigned to each feature.

In [39], the algorithm developed to find the minimum of $E(L)$ consists of initializing L by assigning every feature a different label and then repeatedly merging the neighboring labels that maximize the energy decrease. The algorithm stops when the desired number of labels is reached or when the energy decrease indicates that further merges do not significantly change the energy of the label configuration.

In this thesis a similar energy function is used to define energy of the label configuration. However, a different algorithm is proposed to achieve the desired label configuration. The general idea of the developed algorithm is to perform a region growing labeling scheme. The label configuration L is initialized without any labels assigned and the algorithm stops when every feature has been assigned a label. In this scheme, the desired configuration comprehends the following: if y_i and $y_j \in Y_i^N$ are two neighboring features with the same label, then:

1. y_j is Y_j's closest feature to y_i;

2. and $||y_i - y_j|| < d_{max} = t_l$.

The first item leads to the assembly of more spacial continuous patches. Note that the definition works both ways, which means that for y_i and y_j to receive the same label, y_j has to be Y_j's closest feature to y_i, but y_i too has to be the closest feature to y_j in Y_i. The following energy function verifies this:

$$E_1(L) = \sum_{P_i \in S} \sum_{P_j \in N^{P_i}} \sum_{k=1}^{m_i} \sum_{k'=1}^{m_j} \delta(l_i^k - l_j^{k'}) F(y_i^k, y_j^{k'}) \tag{4.4}$$

where δ is defined as previously: $\delta(z) = 1$ if $z = 0$, and $\delta(z) = 0$ if $z \neq 0$; and $F(y_i^k, y_j^{k'})$ is the function:

$$F(y_i^k, y_j^{k'}) = \begin{cases} -1 & \text{if } ||y_i^k - y_j^{k'}|| \leq ||y_i^k - y_j^{k''}|| \quad \forall k'' \neq k' \\ 1 & \text{otherwise} \end{cases} \tag{4.5}$$

The second item assures that features are not carelessly labeled regardless of the distance between them. Features that are too far apart should not receive the same label as it is unlikely that they belong to the same edge in the volume. $E_2(L)$ in the following equation describes this:

$$E_2(L) = \sum_{P_i \in S} \sum_{P_j \in N^{P_i}} \sum_{k=1}^{m_i} \sum_{k'=1}^{m_j} \delta(l_i^k - l_j^{k'}) \kappa(y_i^k, y_j^{k'}) \tag{4.6}$$

where $\kappa(y_i^k, y_j^{k'})$ is defined as:

$$\kappa(y_i^k, y_j^{k'}) = \begin{cases} 0 & \text{if } ||y_i^k - y_j^{k'}|| < t_l \\ \infty & \text{otherwise} \end{cases} \tag{4.7}$$

Finally, in order to guarantee that there are no repeated labels in Y_i, i.e., $y_i^k, y_i^{k'} \in Y_i, k \neq k' \Rightarrow l_i^k \neq l_i^{k'}$, a third energy function is used:

$$E_3(L) = \sum_{P_i \in S} \sum_{k=1}^{m_i} \sum_{k'=1, k' \neq k}^{m_i} \xi(l_i^k - l_i^{k'}) \tag{4.8}$$

where $\xi(z) = \infty$ if $z = 0$ and $\xi(z) = 0$ if $z \neq 0$.

Equations (4.4),(4.6) and (4.8) can be merged together to form the final energy function:

$$\begin{aligned} E(L) &= E_1(L) + E_2(L) + E_3(L) \\ &= \sum_{P_i \in S} \sum_{P_j \in N^{P_i}} \sum_{k=1}^{m_i} \sum_{k'=1}^{m_j} \delta(l_i^k - l_j^{k'}) \left[F(y_i^k, y_j^{k'}) + \kappa(y_i^k, y_j^{k'}) \right] + E_3(L) \end{aligned} \tag{4.9}$$

This energy function assures that the minimum is only achieved when all criteria is met.

Energy Minimization

This thesis proposes an algorithm that achieves the minimum labeling energy by seeding labels one-by-one and expanding them throughout the neighbors. The label configuration is initially empty - no labels are assigned. The first step of the algorithm is to seed a label in a random feature. This label propagates to the neighboring features whenever the labeling criteria is met, which leads to the local minimization of the energy function. The propagation goes on every time the energy function can be decreased. When all propagation possibilities have been analyzed, the algorithm moves on to the next seed (new label) and the process repeats itself until all features have been labeled. Table 4.1 summarizes the proposed labeling algorithm.

Fig. 4.8 shows the result of the algorithm application to a synthetic volume. The volume is composed of three quarter parts of a sphere, each at its own distance from a common center. The gray sphere is the representation of the surface model; on the left, each yellow dot represents a detected feature; on the right,

repeat
Q = {} % labeling queue
C = {} % labeled features
repeat
If Q is empty
    seed a new label l in a random feature $y_i \notin$ C
    add y_i to C
    for each feature $y_j \in Y_i^N$
        if $y_j \notin$ C & labeling y_j with l lowers $E(L)$
            add y_j to Q
Else
    repeat
    y_i = Q(1)
    label y_i with l
    add y_i to C
    remove y_i from Q
    for each feature $y_j \in Y_i^N$
        if $y_j \notin$ C & labeling y_j with l lowers $E(L)$
            add y_j to Q
    until Q is empty
until all features have been labeled

Table 4.1: Labeling algorithm.

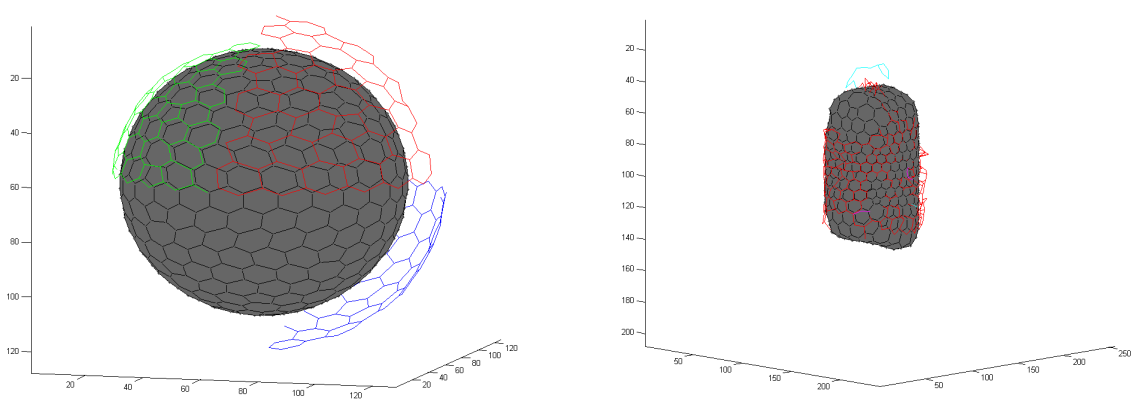


Figure 4.8: Examples of patch extraction (labeling) results in a synthetic volume (on the left) and in an heart ultrasound volume (on the right)

each color represents a different label assigned by the labeling algorithm. The algorithm was able to correctly label each of the three sphere quarters.

Although the proposed algorithm performs well, it is not able to handle patch discontinuities. When assuming some sort of spacial continuity, neighboring features should have the same label if their values are alike (i.e., if they are close to each other). However, if one assumes that there can be no repeated labels along a search line, the labeling procedure face the problem of having structure inconsistencies. A structure inconsistencies exist, for instance, in spiral-like structures and it may happen both in 2D and 3D cases (see Fig. 4.9 and Fig. 4.10, respectively).

Based on the origin of this conflict, it is assumed that it should not happen in ultrasound images of the

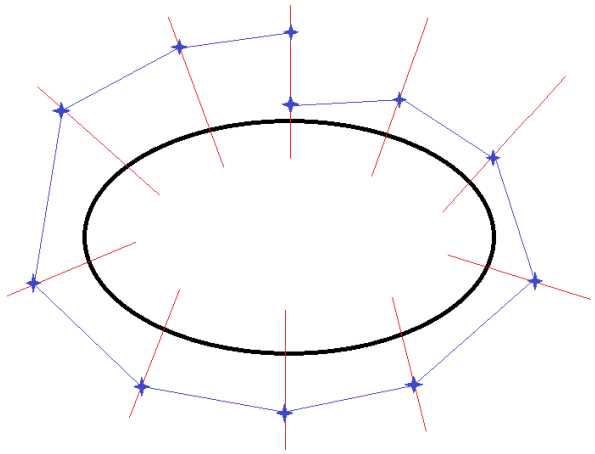


Figure 4.9: Schematic example of a labeling conflict in a 2D image.

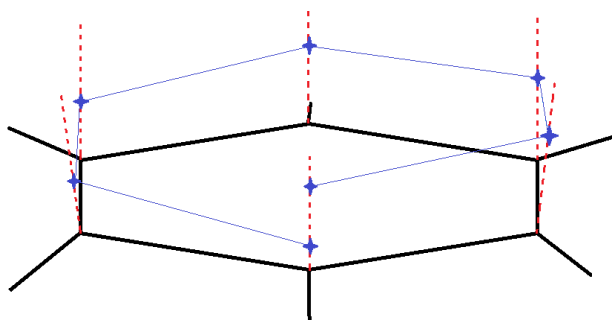


Figure 4.10: Schematic example of a labeling conflict applied to a 3D simplex mesh model.

heart. Thus, no action was taken to tackle this problem.

Chapter 5

Robust Model Estimation

Another way of increasing the robustness of the segmentation system is using robust estimation techniques to determine the most likely location of the LV boundary. These perform better in situations where several features have been detected and most of them are outliers (i.e. they do not belong to the boundary). Since it is impossible to know which feature is valid or invalid, the model estimation has to consider all possibilities.

Model estimation has been extensively used in object tracking algorithms, though it can also be used in static problems: while in tracking systems the object moves throughout time $t = \{t_1, \ldots, t_i, \ldots, t_e\}$, in static problems the object does not change its position but the model adapts in an iterative process $k = \{k_1, \ldots, k_e\}$. Thus, each iteration k can be viewed as a different time frame t.

The most commonly used approach is the Kalman filter [60]. It was introduced in 1960 and, since then, it has been subject of extensive research and applications. The next section will be briefly describe this classic estimation technique.

5.1 Kalman Filter

The Kalman filter addresses the general problem of trying to estimate the state $x(k) \in \mathbb{R}^n$ of a dynamic system, corrupted by white noise

$$x(k) = Ax(k-1) + w(k) \tag{5.1}$$

and assuming that the observations $y(k) \in \mathbb{R}^m$ are given by

$$y(k) = Hx(k) + v(k) \tag{5.2}$$

The random variables $w(k)$ and $v(k)$ represent the process and measurement noise, respectively. These are assumed white and they have normal probability distributions with zero mean:

$$w \sim N(0, Q) \tag{5.3}$$

$$v \sim N(0, R) \tag{5.4}$$

where Q and R are the (known) covariance matrices of the distributions.

Fig. 5.1 presents an overview of the Kalman process applied to shape estimation and tracking. The Kalman filter can be decomposed into three blocks: a shape prediction block in which the state estimate $\hat{x}$ is computed based on prior observations; a feature extraction block in which observations are detected; and shape update block where the state estimate is updated based the new observations.

The Kalman filtering procedure has been successfully used in several application. However, it has some limitations when applied to noisy images. In such cases, feature detection algorithms often detect many outliers (see Chapter 4), which may lead to meaningless results using the Kalman filter. Fig. 5.2 shows examples of the lip track using the Kalman filter in the presence of outliers, where the dots represent the

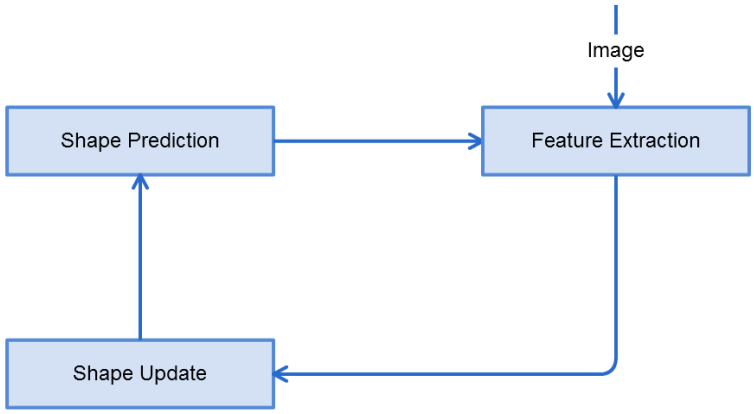


Figure 5.1: The Kalman filter.

detected features. In the absence of outliers, the Kalman filter performs well (on the left), whereas the presence of outliers leads to unexpected and undesired results.

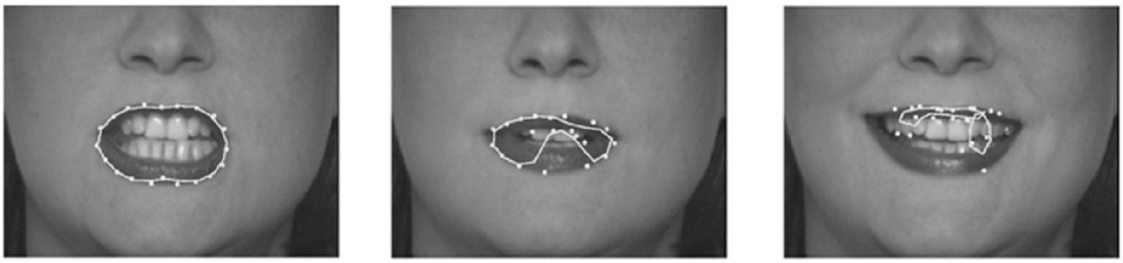


Figure 5.2: Lip tracking with Kalman filter in the presence of outliers [46].

Additional methods have been developed to prevent this. For instance, imposing restrictions to the object shape can prevent the contour from assuming undesired shapes [6, 17]. However, these are not able to discriminate which features belong to the desired boundary an which are produced by the background.

Other approaches have also been proposed, such as: the Nearest Neighbor Filter [54], which replaces the intensity of outlier pixels by an average of the adjacent pixels; and Track Splitting Filter [45], which considers all possible interpretations of valid and invalid features in all iterations k. This last method is the ideal approach but it is not viable when the number of detected features is too large.

5.2 Shape Probability Data Association Filter

An alternative to the classic Kalman filter is the robust shape tracking algorithm proposed in [46], called Shape Probabilistic Data Association Filter (S-PDAF). This algorithm is inspired in the Probabilistic Data Association Filter (PDAF), proposed by Bar-Shalom [3, 33].

The difference between the PDAF and the Kalman filter lies in the data association component, that assigns a weight (confidence degree) to each observed feature. The weights are based on the probability of the observation being generated by the model. Fig. 5.3 shows a block diagram of the PDAF algorithm, in which the difference with the overview of the Kalman filter in Fig. 5.1 is evident.

The PDAF algorithm combines all hypothesis $h_i(k)$, $i = 1, \ldots, m_k$, of considering the feature y_i as valid by defining the state estimate at step k as:

$$\hat{x}(k) = \sum_{i=0}^{m_k} \hat{x}_i(k)\alpha_i(k) \tag{5.5}$$

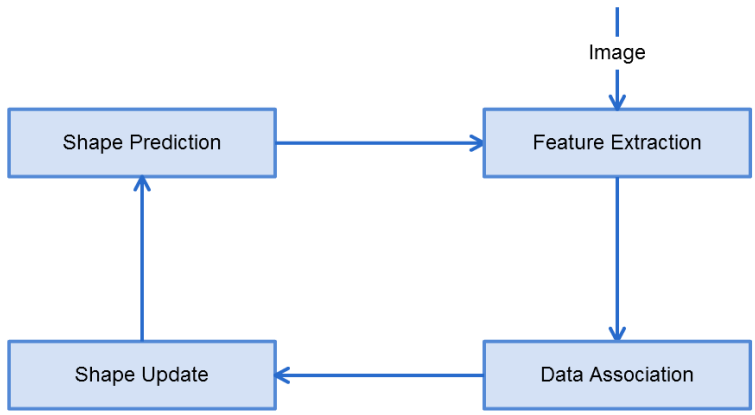


Figure 5.3: PDA filter.

where $\hat{x}_i(k)$ is the traditional Kalman updated state, conditioned on considering the hypothesis $h_i(k)$ true; and $\alpha_i(k)$ is the association probability of that hypothesis, which depends on the set of observations $Y(k)$:

$$\alpha_i(k) = P\{h_i(k)|Y(k)\} \tag{5.6}$$

The S-PDAF differs from the PDAF in two aspects: first, instead of using single features (measurements), it considers middle-level image features such as the patches assembled in Section 4.3. This considerably reduces the number of features used in the estimation procedure. Second, the computation of the association probabilities is adapted to the shape estimation problem.

The S-PDAF approach is the following: assuming each of the m_l detected patch in iteration k can either be true (belongs to the object boundary) or false (produced by the background), a patch interpretation $I_i(k) = \{I_i^1, \ldots, I_i^n, \ldots, I_i^{m_l}\}$ is defined as a sequence of the combination of patches labeled as true ($I_i^n = 1$) or false ($I_i^n = 0$).Therefore, the total number of interpretations is $m_k = 2^{m_l}$.

The correct patch label is unknown and, thus, the correct interpretation is also unknown. However, the computation of the association probabilities (described in the last section of the chapter) assigns high probabilities to the most likely interpretations and low probabilities to all meaningless interpretations.

The following section describes the state estimation used in the S-PDAF algorithm.

State Estimation

In this thesis, the dataset is a static echocardiographic volume. Therefore, it is assumed that the object (LV) shape and position is described in each iteration k by

$$x(k) = x_(k-1) + w(k) \tag{5.7}$$

where $w(k) \sim N(0, Q)$ is white Gaussian noise with normal distribution and zero mean. The state vector $x(k)$ corresponds to the desired position of all the vertices of the simplex mesh.

For each interpretation $I_i(k)$, the features $y_i(k)$ are generated by a different model. If an observation $y_i(k)$ is considered invalid (outlier) in $I_i(k)$, we assume it is generated by uniform distribution. Otherwise, if $y_i(k)$ is considered valid, then we assume it relates to the boundary points $x(k)$ by:

$$y_i(k) = x(k) + v_i(k) \tag{5.8}$$

where $v_i(k) \sim N(0, R_i)$ is a white Gaussian noise with normal distribution associated with the valid features $y_i(k)$ of the interpretation $I_i(k)$.

The state estimate is defined as in Equation (5.5), where $\alpha_i(k)$ now yields:

$$\alpha_i(k) = P(I_i(k)|Y(k)) \tag{5.9}$$

It can be proven (see [46]) that the state estimate given $Y(k)$ in Equation (5.5) may be rewritten as:

$$\hat{x}(k) = \hat{x}(k-1) + \sum_{i=1}^{m_k} \alpha_i(k) K(k) \nu_i(k) \tag{5.10}$$

where $K(k) = P(k-1)(P(k-1) + R_i)^{-1}$ is the Kalman gain, $P(k-1)$ is the covariance matrix at iteration $k-1$, and $\nu_i(k) = y_i(k) - \hat{x}_(k-1)$ is the innovation vector.

Equation (5.10) updates the state vector. The obtained state estimate is then used in the adaptation of the surface model by defining each external force in Equation (3.5), $\mathbf{F}_i^{ext}$, associated with the vertex P_i as:

$$\mathbf{F}_i^{ext} = \hat{x}_i - P_i; \tag{5.11}$$

where $\hat{x}_i$ is the obtained estimate location of the LV boundary for P_i.

The weight of each interpretation is determined by the association probabilities $\alpha_i(k)$ that will be described next.

Association Probabilities

The association probabilities $\alpha_i(k)$ correspond to the probability of $I_i(k)$ being true at each iteration k. This probability depends on two different aspects: the size of the patches it considers valid and invalid, and their distance to the model. Furthermore, Interpretations with overlapping patches will have zero probability - for the same reason that overlapping features receive different labels (see Chapter 4.3). The model will also discard patches that are considerably smaller than the larger ones, since they are most likely noise originated and would only slow the performance of the algorithm.

At each time step (iteration) k, the following variables are defined:

- $I(k) = \{I_1(k), \ldots, I_i(k), \ldots, I_{m_k}(k)\}$ is the set of existing interpretations;
- $Y(k)$ is the set of all detected features in iteration k;
- $L(k) = \{l_1(k), \ldots, l_n(k), \ldots, l_{m_l}(k)\}$ is the set of patches assembled from Y;
- and $\hat{x}(k) = \{\hat{x}_1(k), \ldots, \hat{x}_{m_s}(k)\}$ is the state estimate.

From this point on, we will be omitting the k dependency of these variables to simplify the notation. The *a posteriori* probability of an interpretation I_i is defined by:

$$\alpha_i = P(I_i | Y, L, \hat{x}) \tag{5.12}$$

Using a Bayesian approach, Equation (5.12) can be rewritten as:

$$\alpha_i = \frac{P(Y | I_i, L, \hat{x}) \times P(I_i | L, \hat{x})}{\beta} \tag{5.13}$$

where $\beta = P(Y | L, \hat{x})$ is a normalization constant that does not depend on I_i, $P(Y | I_i, L, \hat{x})$ is the likelihood of the set of features Y and $P(I_i | L, \hat{x})$ is the prior probability of the interpretation I_i conditioned on the patches (i.e., based on its valid and invalid patches). These two individual probabilities are defined in the next subsections.

Defining $P(Y | I_i, L, \hat{x})$

The likelihood of a given set of features can be determined by their distance to the model. In [46], it was assumed that all image features were independently generated, i.e., the probability $P(Y | I_i, L, \hat{x})$ could be calculated as the product of the individual feature probabilities $P(y_j | I_i, L, \hat{x})$. However, in 3D cases such as the one in this thesis, the number of features is considerably larger and computing the product of all the individual probabilities is not computationally feasible. Therefore, a different approach was considered.

Instead of assuming that all image features are independently generated, we assume that all image patches are independently generated. Thus:

$$P(Y|I_i, L, \hat{x}) = \prod_{n=1}^{m_l} P(l_n|I_i, L, \hat{x}) \tag{5.14}$$

Each probability $P(l_n|I_i, L, \hat{x})$ is defined as the probability of having a patch l_n at a certain average distance to model. Thus, the probability density function is either assumed:

1. uniform along the search line, if l_n is considered as invalid ($I_i^n = 0$)

   or

2. normally distributed, if l_n is considered as valid ($I_i^n = 1$)

Combining the two:

$$P(l_n|I_i, L, \hat{x}) \sim \begin{cases} V^{-1} & \text{if } I_i^n = 0 \\ \rho^{-1} N(d; 0, \sigma) & \text{otherwise} \end{cases} \tag{5.15}$$

where:

- $V = d_1 + d_2$ is the length of the search line (see Section 4.1);
- ρ is the normalization constant;
- d is the average distance between the features in of the patch and their corresponding vertices (i.e., the average distance between the patch and the surface);
- and σ is the covariance of the Gaussian distribution.

In order to simplify the problem, σ is defined as being proportional to the length of the search line.

Defining $P(I_i|L, \hat{x})$

As to the prior probability of the interpretations, $P(I_i|L, \hat{x})$, if we assume again that patches are independently generated, it can be rewritten as the product of each individual patch probability:

$$P(I_i|L, \hat{x}) = P(I_i^1|L, \hat{x}) \dots P(I_i^{m_l}|L, \hat{x}) \tag{5.16}$$

The probability $P(I_i^n|L, \hat{x})$ varies depending on the label each patch as received in the interpretation I_i (i.e., $I_i^n = 0$ or if $I_i^n = 1$). The only information used in the definition of the prior probability is the following:

- The probability that a patch belongs to the LV boundary should be proportional to its size, i.e., $P(I_i^n = 1|L, \hat{x})$ should be high if the corresponding patch is large patches and it should be small if the patch is not large;
- On the other hand, $P(I_i^n = 0|L, \hat{x})$ should be small for large patches and it should be high for smaller patches.

In order to guarantee that long valid patches are favored, their areas A_{l_n} are used in the determination of probability of I_i conditioned on the patches. We define the area of a patch as the number of features it comprises. Based on these two premises, the individual probabilities $P(I_i^n|L, \hat{x})$ were defined as follows:

$$\begin{aligned} P(I_i^n = 1|L, \hat{x}) &= mA_{l_n} + c \\ P(I_i^n = 0|L, \hat{x}) &= 1 - (mA_{l_n} + c) \end{aligned} \tag{5.17}$$

where

$$\begin{aligned} c &= P_A \\ m &= \frac{P_B - P_A}{A_{max}} \end{aligned} \tag{5.18}$$

and where $P_A, P_B \in [0,1]$ and A_{max} are constants.

Combining (5.16) and (5.17),

$$P(I_i|L,\hat{x}) = \prod_{l_n:I_i^n=1} mA_{l_n} + c \prod_{l_n:I_i^n=0} 1 - mA_{l_n} - c \tag{5.19}$$

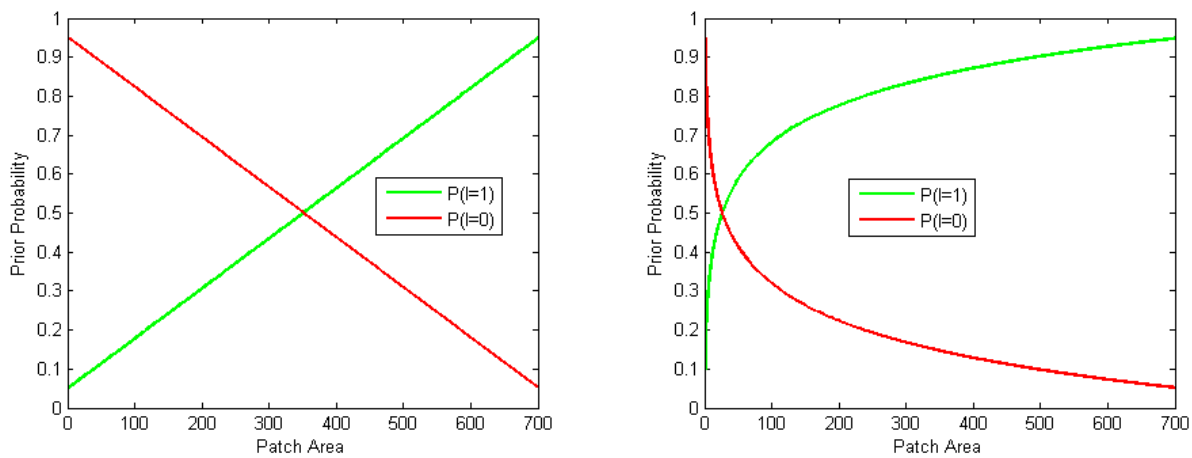
Another way of defining the prior probability of a patch is using an logarithmic function instead of the linear one used in (5.17). Formally:

$$P(I_i|L,\hat{x}) = \prod_{l_n:I_i^n=1} a\log(A_{l_n}+1) + b \prod_{l_n:I_i^n=0} 1 - [a\log(A_{l_n}+1) + b)] \tag{5.20}$$

where:

$$\begin{aligned} a &= \tfrac{P_B - P_A}{\log(A_{max}+1)} \\ b &= P_A \end{aligned} \tag{5.21}$$

The difference between these two definitions lies in the location of the intersection between the probability $P(I_i^n = 1|L,\hat{x})$ and $P(I_i^n = 0|L,\hat{x})$. Comparing Fig. 5.4 (a) with Fig. 5.4 (b), we can see that in the first case the intersection occurs approximately at $A = \frac{A_{max}}{2}$, whereas in the second case it occurs at a considerably smaller A. This intersection marks the patch area for which the model starts to prefer labeling the patch as valid instead of invalid.



(a) (b)

Figure 5.4: (a) Linear prior probability for $I_i^n = 1$ and $I_i^n = 0$. (b) Exponential prior probability for $I_i^n = 1$ and $I_i^n = 0$.

As an example, consider two patches with areas $A_1 = 200$ and $A_2 = 600$. Using the values $P_A = 0.05$ and $P_B = 0.95$, the prior probabilities of each patch (labeled as valid or invalid) are:

	Linear Case	Exponential Case
$P(I^1 = 1\|L,\hat{x})$	0.31	0.78
$P(I^1 = 0\|L,\hat{x})$	0.69	0.22
$P(I^2 = 1\|L,\hat{x})$	0.82	0.93
$P(I^2 = 0\|L,\hat{x})$	0.18	0.07

Table 5.1: Prior probability of the patches with areas $A_1 = 200$ and $A_2 = 600$ in the linear and exponential cases.

	Linear Case	Exponential Case
$P(I_1\|L,\hat{x})$	0.12	0.02
$P(I_2\|L,\hat{x})$	0.57	0.20
$P(I_3\|L,\hat{x})$	0.06	0.05
$P(I_4\|L,\hat{x})$	0.25	0.73

Table 5.2: Prior probability of the interpretations in the linear and exponential cases.

Assuming there are four interpretations: $I_1 = \{0,0\}, I_2 = \{0,1\}, I_3 = \{1,0\}, I_4 = \{1,1\}$, the prior probability of each interpretation would be as shown in Table 5.2.

In the linear case, the highest prior probability is associated with interpretation $I_2 = 0,1$, i.e., validating only the largest patch and considering that the smaller patch should be assigned as invalid. The exponential case, on the other hand, assigns $I_4 = 1,1$ the highest prior probability, which considers both patches valid. Choosing between the two types of prior probability depends on the area A_{l_n} from which the model should start considering patch l_n as valid.

Each interpretation has a data association probability α_i given by the previous equations. As will be shown in Chapter 6, the computation of the assocation probabilities described above yields good results. In synthetic volumes, the estimation model quickly tends to a single interpretation (the correct one), neglecting all other interpretations, except for examples specifically built to created doubt. In the worst case scenario, two interpretations receive high confidence degrees and the surface model tries to fit them both. A comparison between this model and the classic Kalman filter method will also be shown for synthetic images with clutter.

Chapter 6

Results

This chapter presents the results of applying the proposed segmentation system to different datasets. Section 6.4 shows how the system behaves in synthetic volumes. This analysis will be used to test several aspects of the developed system. In Section 6.5 the segmentation system is applied to the segmentation of the LV in echocardiographic data and the respective validation of the results is presented. Section 6.6 compares the performance of the proposed system with a classic estimation techniques. Finally, Section 6.7 demonstrates how parameter variations may lead to different results.

The following two sections explain how to display these results and presents objective evaluation methods used in LV segmentation systems, respectively.

6.1 Model Visualization

The visualization of the segmentation within the echocardiographic volume poses some problems. Traditionally, the visualization of 3D images is done by sampling slices along one direction. However, the segmentation is a 3D surface defined by vertices and edges. Thus, to visualize slices from the volume together with its correspondent segmentation, the surface has to be sliced as well.

The segmentation results can be visualized in two different ways:

1. We may visualize the 3D surface of the segmented region. This provides an overall display of the LV shape but has no information about its location in the echocardiographic volume;

   or

2. we may display volume slices and the 2D curve obtained by intersecting the LV surface with the visualized slice.

The 3D surface is shown by plotting all patches formed by the simplex mesh edges (as shown in Fig. 3.1). Note that the patches are not planar; we use a special Matlab routine called *patch* that automatically displays them given their vertices. This type of visualization allows the viewer to assimilate the whole configuration and structure of the surface model, but it does not give any information of the location of the model within the data volume. Therefore, it cannot be used to visually confirm that the surface model is correctly located and the segmentation was well performed.

On the other hand, showing slices from the data volume and plotting the corresponding 2D segmentation gives a clear information about the location of the surface. However, finding the 2D curve obtained by intersecting the 3D surface with the slice requires some additional work.

The 2D segmentation that corresponds to the limits of the surface can be determined by intersecting the surface model with the plane (slice) that is being shown. However, simplex mesh surface is not explicitly defined. We only know the position of vertices and edges. Therefore, we will determine which edges are intersected by the slice and compute the intersection points.

Let us consider a plane $\mathcal{P}$ defined by $z = ay + z_0$ in $\mathbb{R}^3$, where z_0 defines the height of the plane and a can be used to create an oblique plane containing a specific search line (i.e. containing the normal vector $\mathbf{n}_i$). We first determine which vertices $P = (x_P, y_P, z_P)$ are located above and below that plane:

$$\begin{aligned} P_{below} &= \{P : z_P \leq ay_P + z_0\} \\ P_{above} &= \{P : z_P > ay_P + z_0\} \end{aligned} \tag{6.1}$$

These vertices are then used to find the set of edges $E_{PP'}$ that intersect $\mathcal{P}$. This means searching all edges (P, P') such that:

$$E_{PP'} = \{(P, P') : P' \in N_P \wedge P \in P_{below} \wedge P' \in P_{above}\} \tag{6.2}$$

The exact location of the intersection I_e between the edge and the plane is computed by finding the point for which the edge (P, P') intersects the plane $\mathcal{P}$:

$$\begin{aligned} I_e = P + K(P' - P) \in \mathcal{P} \\ \Leftrightarrow z_{I_e} = z_{\mathcal{P}} \\ \Leftrightarrow z_P + K(z_{P'} - z_P) = a\left[y_P + K(y_{P'} - y_P)\right] + z_0 \end{aligned} \tag{6.3}$$

Solving the previous equation yields:

$$K = \frac{ay_P + z_0 - z_P}{z_{P'} - z_P + a(y_{P'} - y_P)} \tag{6.4}$$

The corresponding 2D location of I_e is $s(x, y) = (x_{I_e}, y_{I_e})$. Therefore, the segmentation is characterized by unordered scattered points s throughout the image (see Fig. 6.2 on the left). To determine the 2D segmentation, these scattered vertices have to be linked according to the limits of the surface.

Since the correct point connection is unknown, the points can be interpreted as a graph where every point s_i is connected with all other points $s_j, j \neq i$. In order to correctly link the vertices, some prior information must be used. In this case, the only known information is that all points belong to the segmentation contour. If there was any assurance that the 2D segmentation would result in a convex figure, the convex hull algorithm [4] could be applied. However, results show that this is unlikely.

To simplify the problem, it is assumed that limit of the surface is determined by the shortest closed path that passes through all the vertices. This way, a minimum spanning tree [34] based on the distance between points can be used to determine the correct path. A spanning tree of a graph is a subgraph that connects all the vertices together. Thus, the minimum spanning tree is the shortest path that connects all vertices.

Consider the set of points S composed of all N_s points s_i that intersect the image plane. To determine the minimum spanning tree, a weight matrix $W = w_{ij}$ where $i, j = 1, \ldots, N_s$ has to be assemble first. The weight matrix, is this case, is determined by the distance between vertices:

$$w_{ij} = ||s_i - s_j|| \tag{6.5}$$

Then, the minimum spanning tree is applied using the weight matrix W. This results in the shortest path that passes through all vertices. Note that the resulting path is not closed, as intended. To close the path, one additional edge has to be added. Finding this edge is easy, since it is formed by the only two vertices that are not part of two edges of the minimum spanning tree (see the arrow in Fig. 6.1). The path is closed by linking these two vertices.

An example of this type of visualization can be viewed in Fig. 6.2. Although the assumption that the limit of the surface is determined by the shortest path proves valid in most cases, it may not always be correct. And iif the assumption is invalid, finding the minimum spanning tree results in a incoherent segmentation (path).

Fig. 4.4 shows examples of the visualization of oblique planes containing a search line. Note that plotting more than one search line simultaneously is impossible since search lines are not coplanar.

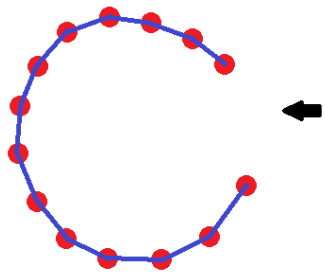


Figure 6.1: Schematic example of the minimum spanning tree of a graph. The red dots represent the graph's vertices and the blue lines are the edges that define the minimum spanning tree.

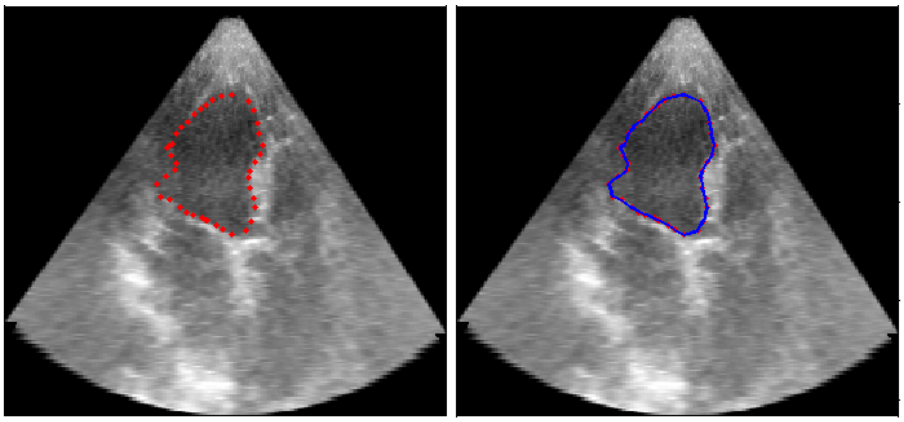


Figure 6.2: (On the left) Intersection of the surface model with a slice of the echocardiographic volume. (On the right) Slice of the echocardiographic volume and the correspondent 2D segmentation.

6.2 Evaluation Methods

Visualization methods allow a qualitative evaluation of the segmentation results in 2D and 3D. However, a quantitative assessment of the segmentation results is needed in order to obtain performance measures, allowing direct comparisons with other approaches. Evaluation methods enable the quantification of a segmentation system's performance based on its results, which is essential for the validation of the system. These can be divided into two metric types:

1. clinical performance analysis in terms of global LV function (volume and mass);

2. and similarity measurements between the proposed segmentation results and the manual segmentation.

The first type is based on global LV functions such as the end-diastolic and end-systolic volumes. The second type compares the LV automatic segmentation with a manual segmentation performed by an expert, the ground truth (GT), and assesses this comparison based on similarity metrics.

In this thesis, the evaluation methods used fit into the second type of metrics. The GT is composed by a set of manual segmentations performed in different slices for each volume (see an example in Fig. 6.3. For each of these slices with a GT, the evaluation methods will be applied and the segmentation evaluation value is computed for the validation of the whole 3D segmentation.

Five error metrics will be used to characterize the similarity between the obtained result and the GT: (1) Hammoude metric [25], d_{HMD}; (2) average metric, d_{AV}; (3) Hausdorff metric [29], d_{HDF}; (4) mean sum

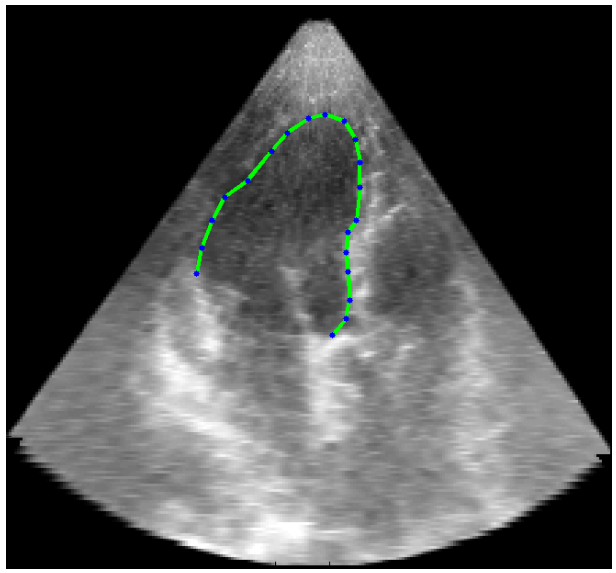


Figure 6.3: Example of a manual segmentation of the LV (ground truth).

of squared distance metric, d_{MSSD}; and (5) mean absolute distance metric, d_{MAD}, which will be briefly described next.

Consider R_Ψ as the region delimited by model segmentation and R_Ω as the region delimited by the GT. The Hammoude metric is defined by:

$$d_{HMD}(\Psi, \Omega) = \frac{\#((R_\Psi \cup R_\Omega) - (R_\Psi \cap R_\Omega)}{\#((R_\Psi \cup R_\Omega))} \tag{6.6}$$

This error metric corresponds to the percentage of area between the two contours using a XOR operator. Low values of d_{HMD} indicate high similarity between both regions.

Now consider the border of the model segmentation defined by the points $\Psi = \{\psi_1, \ldots, \psi_{N_\psi}\}$ and the border of the GT $\Omega = \{\omega_1, \ldots, \omega_{N_\omega}\}$. The average metric between two contours is defined as the average distance to the closest points, $d(\psi_i, \Omega)$,

$$d_{AV} = \frac{1}{N_\Psi} \sum_{i=1}^{N_\Psi} d(\psi_i, \Omega) \tag{6.7}$$

where N_Ψ is the length of Ψ and

$$d(\psi_i, \Omega) = \min_j ||\omega_j - \psi_i|| \tag{6.8}$$

The Hausdorff metric is defined as the maximum value of $d(\psi_i, \Omega)$ between the two contours:

$$d_{HDF}(\Psi, \Omega) = \max\left(\max_i\{d(\psi_i, \Omega)\}, \max_j\{d(\omega_j, \Psi)\}\right) \tag{6.9}$$

Finally, the metrics MSSD and MAD are defined by:

$$d_{MSSD}(\Psi, \Omega) = \frac{1}{N} \sum_{i=1}^{N} ||\psi_i - \omega_i||^2 \tag{6.10}$$

and

$$d_{MAD}(\Psi, \Omega) = \frac{1}{N} \sum_{i=1}^{N} ||\psi_i - \omega_i|| \tag{6.11}$$

which correspond, respectively, to the squared distance's average and to the absolute distance's average between corresponding points in the boundaries.

Section 6.5 presents results of the evaluation methods in different echocardiographic volumes.

6.3 Parameter Definition

Before applying the segmentation system, a few variables have to be defined first.

Adaptation Process

In Equation (3.5), the parameters α and β have to be defined so that the surface adaptation is smooth. Several values were tested (within the interval $]0, 1]$):

- high values of α and β, which led to sharp displacements of the surface (due to the strength of the external forces) and occasional conflicts caused by poorly distributed vertices;
- small values of α and β, that resulted in a slow adaptation process, with almost no internal force influence;
- a combination of small α and high β and vice-versa.

The parameters that proved to work better were a combination of high α and small β. Higher values of α lead to a simplex mesh that maintains its smoothness and good distribution of the vertices throughout the surface. On the other hand, small values of β assure that the external forces does not originate sharp displacements, which may cause mesh problems, and allows the internal force to keep the simplex mesh structure and thus guaranteeing the stability of the adaptation. The parameters α and β utilized throughout the next sections were $\alpha \in [0.4, 0.6]$ and $\beta \in [0.02, 0.1]$.

As to the parameter γ of Equation (3.5), the damping factor, results showed that higher values led to faster adaptations. Therefore, all applications used $\gamma \in [0.7, 0.9]$.

The stopping criterion of the adaptation's iterative process is another parameter that has to be defined. Several criteria may be used, such as the maximum displacement of the vertices in the surface, or the number of iterations. We chose to used the average displacement has the stopping criteria: the adaptation process stops when the average displacement of the vertices is lower than k_{stop}. The value of k_{stop} used in the following sections was $k_{stop} = 0.005$.

Feature Detection

In the feature detection process, three parameters need to be defined: the length of the search line, defined by the parameters d_1 and d_2, and the filter length, d_f. The values used for all the following examples were:

$$d_1 = \frac{d_2}{3} d_2 = 60 d_f = 10 \tag{6.12}$$

Other parameters that need to be defined are the detection threshold t_f and the labeling threshold t_l. These correspond, respectively, to the maxima detection threshold used in Section 4.2, and the maximum distance allowed between low-level features when assigning labels in the middle-level features assembly (Section 4.3). Both of them influence the final result. For instance, the detection threshold has of major importance to guarantee that the LV boundary is one of the detected features and to keep the number of detected features as small as possible. We defined the threshold as a percentage of the convolution maximum, c_{max}. Fig. 6.4 shows three histograms of the number of detected features *per* vertex using (a) $t_f = 0.2c_{max}$, (b)

$t_f = 0.5c_{max}$ and $t_f = 0.9c_{max}$. The influence of the threshold on the number of detected features *per* vertex is clear. Furthermore, the total number of features detected by each thershold was: (a) 2257 features, (b) 1336 features and (c) 807 features (note that the surface has 700 vertices).

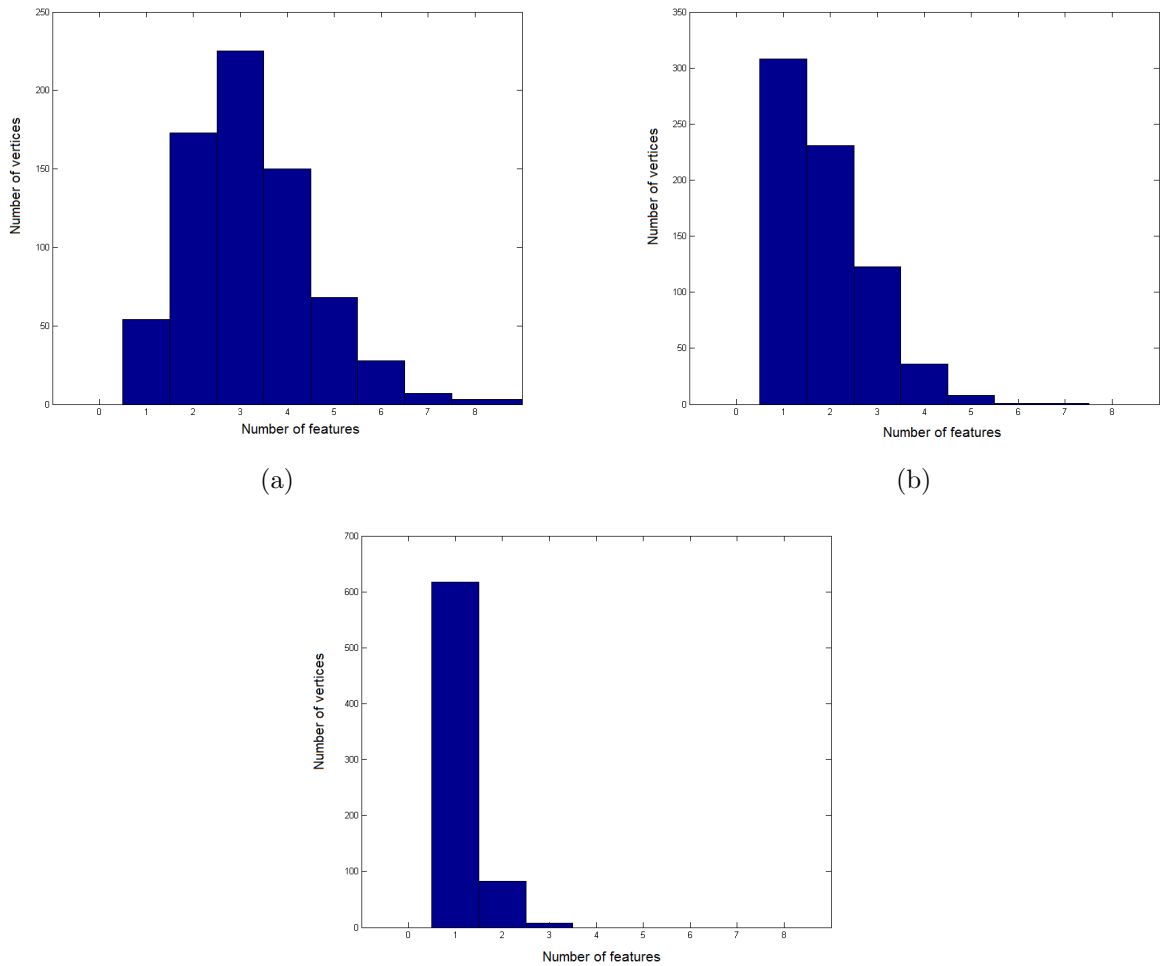


Figure 6.4: Histograms of the number of detected features *per* vertex, using (a) a low, (b) a medium and (c) a high detection threshold.

We can see that small values of t_f result in the detection of many features, with an average of three detected features *per* vertex. This is not desirable, since ideally only the LV boundary feature should be detected. t_f too high is also troublesome because most vertices detect only one feature but there is not guarantee that the detected feature is the LV boundary. Knowing that the algorithm is unable to determine which is the correct feature, it should be able to extract more than one feature in order to consider more possibilities. Using a value between 0.5 and 0.9 yields this result. The value used in all the results shown was 0.7.

A similar analysis can be performed for the labeling threshold, although the consequential results are only visible in the final configuration of the surface. Therefore, different results obtained using different t_l values will be presented farther ahead in Section 6.7. The same goes for variations in the initialization procedure, which will also be addressed in Section 6.7.

Association Probability

Finally, some parameters in the association probabilities also need to be determined. In the prior probability, we will use the exponential case because it yields better results for all scenarios. The values of P_A, P_B and A_{max} used were:

- $P_A = 0.05$;
- $P_B = 0.95$;
- and A_{max} = number of vertices in the surface = 700.

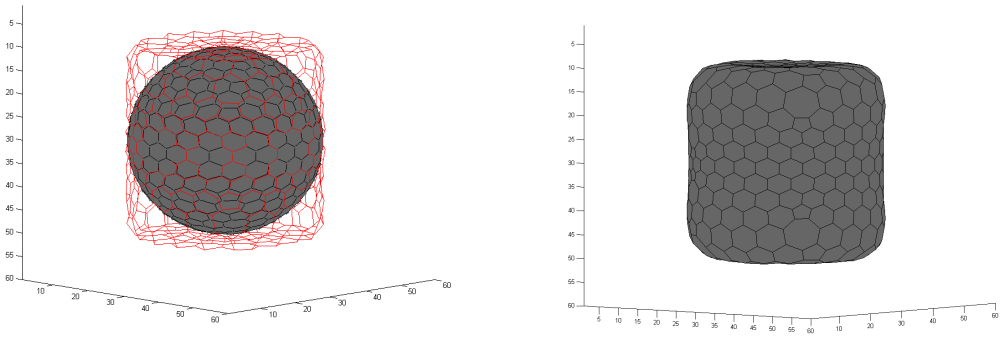
6.4 Segmentation of Synthetic Data

The proposed algorithm was applied to several synthetic data sets in order to illustrate its performance. We will now present four of these experiments and discuss their outputs. All synthetic data consists in binary volumes representing a structure and each of the synthetic volumes used in this section had the intention of evaluating some particular aspects of the system.

The first two address the behavior of the surface model in different scenarios. These scenarios contain simple structures without noise, clutter or misleading structures. The last two, on the other hand, contain a volume corrupted with white noise and a volume with misleading structures, respectively. Therefore, results of the model estimation will only be presented in the final two volumes.

Cylinder

The first volume aimed to test the functionality of the surface model. Using a synthetic volume containing a cylinder, the idea was to evaluate the adaptation procedure. Fig. 6.5 (a) shows the patches detected after initialization of the surface model as a spheric simplex mesh.



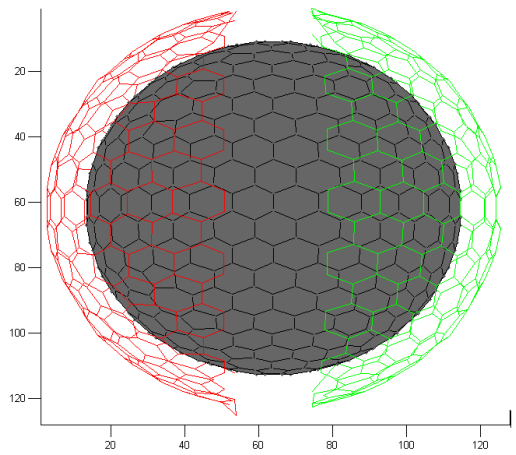
(a) (b)

Figure 6.5: (a) Initial surface and patch detection in a synthetic cylinder. (b) Final configuration of the surface.

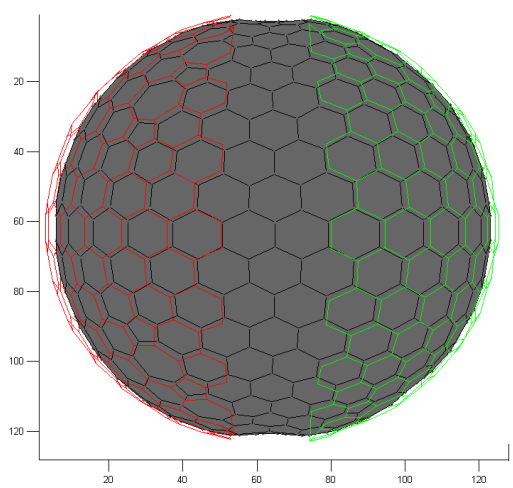
The final configuration of the surface is shown in Fig. 6.5 (b). Note that simplex meshes are not able to represent edges due to its intrinsic structure. Nonetheless, the final surface is able to segment the cylinder accurately in 44 iterations.

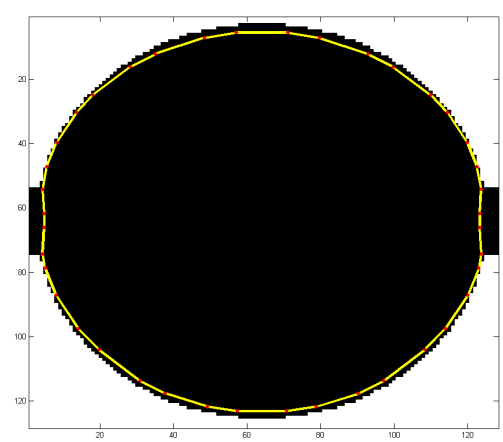
Sectioned Sphere

The second volume was developed to test the surface behavior in the absence of features, which is common in echocardiographic images. The volume contains a sphere that has been cut in two semi-spheres. As a result, no features are detected along the cut and, consequently, the only detected patches are the two semi-spheres with a gap in the middle, as can be seen in Fig. 6.6 (a).



(a)





(b) (c)

Figure 6.6: (a) Patch detection in a synthetic sectioned sphere. (b) and (c) Final configuration of the surface.

The final configuration of the surface in this particular volume is highly dependent on the values of α and β: the slower the displacement of the vertices towards the corresponding features (defined by β), the better the surface will behave at the vertices without features. Fig. 6.6 (b) shows the 3D model of the surface and Fig. 6.6 (c) shows a slice of the volume and the intersection with the surface. Even in the absence of data, it can be seen that the model tries to maintain a smooth surface.

Sphere in Clutter

The third volume contains a sphere whose boundary is surrounded by clutter generated by adding white Gaussian noise with zero mean in random 3D locations in the vicinity of the synthetic sphere (see Fig. 6.8 (c)). The objective of this test is to evaluate the robustness of the segmentation in a cluttered environment. This type of environment is also present in echocardiographic volumes, though here the sphere boundary is always detectable.

The model is initialized as a sphere with smaller radius than the radius of the synthetic sphere and they are concentric. Fig 6.8 (a) shows the detected patches consisting of a large patch that corresponds to the sphere, and a few other small patches (enhanced for a better visualization). The labeling algorithm does not group noisy features into large patches due to their spatial inconsistency. Fig 6.8 (b) shows a detail of Fig 6.8 (a) in which it is possible to see smaller detected patches (in green and purple) and where it can also be seen that the large blue patch is not a perfectly round sphere, which means that the clutter influences the location of the low-level features.

In the first iteration of the adaptation process, there was a total of over 2300 detected low-level features and the histogram in Fig. 6.7 shows that there was an average of 3 features associated with each vertex. Consequently, the labeling algorithm also labeled a large number of patches (approximately 1200) among which there was one single large patch with 700 features (one feature *per* vertex). However, note that, as mentioned in Section 5.2, patches considerably smaller than the largest detected one are disregarded by the model. Consequently, the model discarded all detected patches except for the large blue one and it only considered two possible interpretations: $I_1 = 0$ and $I_2 = 1$.

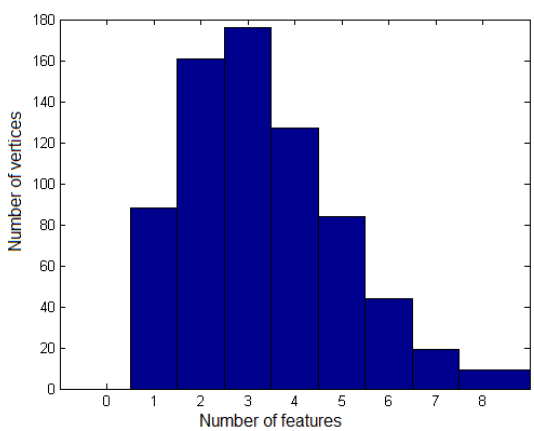


Figure 6.7: Histogram of the number of detected features *per* vertex.

Nonetheless, the association probabilities of the two possible interpretations were the following:

$$\alpha_1 = P(I_1) = 0.04$$
$$\alpha_2 = P(I_2) = 0.96$$

This means the model practically disregarded I_1. These results are intrinsically related to the behavior of middle-level features' assemblage: large patches have a strong influence on the adaptation of the model, but noisy features are usually grouped in smaller patches that are disregarded by the robust model estimation block. Fig 6.8 (c) shows that the final configuration is unaffected by the presence of clutter.

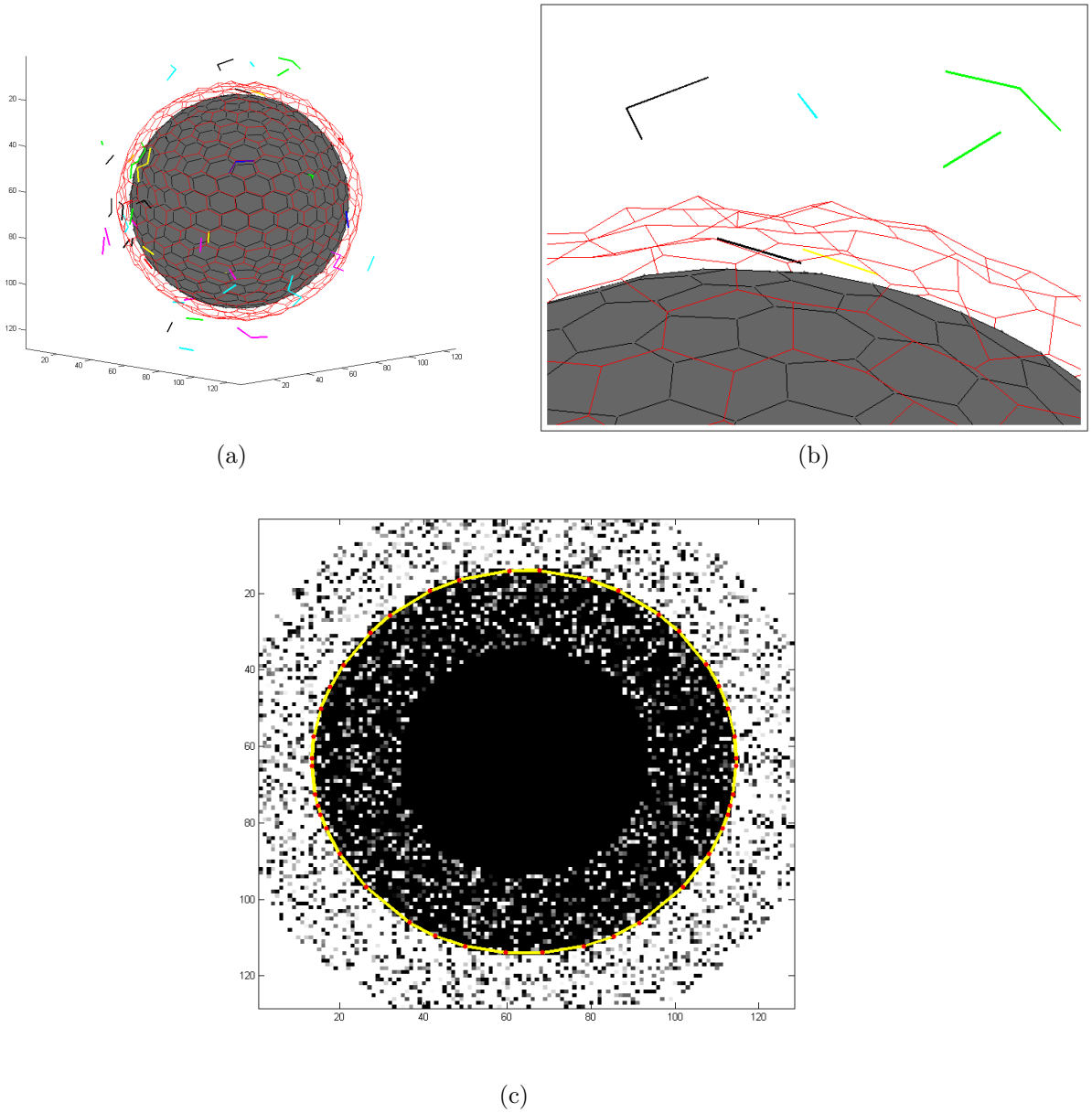


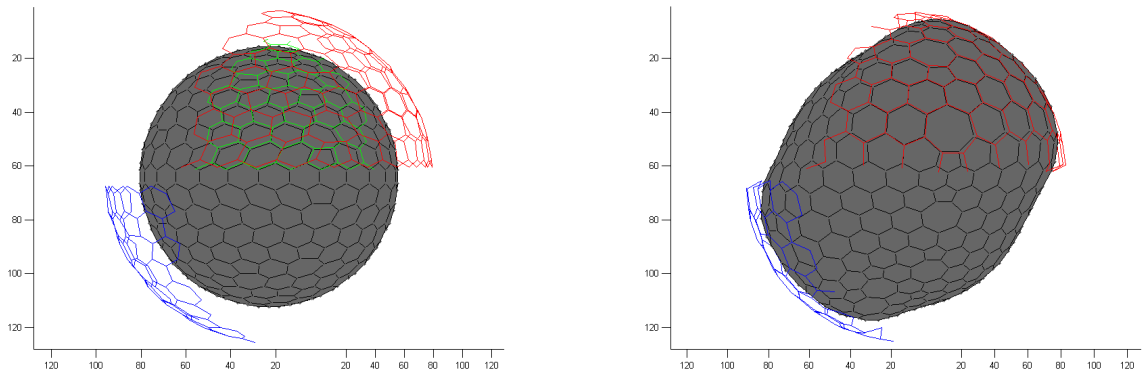
Figure 6.8: (a) Patch detection in a synthetic noisy sphere. (b) Close-up on a patch detail. (c) Slice view of the final configuration of the surface.

Sphere Portions

Unlike the previous one, this last synthetic volume consist of unconnected sphere portions. The volume contains three separate portions of a sphere, all of them concentric; one of the portions is a quarter of a sphere, whereas the other two are one eighth of a sphere; one of the smaller portions is closer to the initial surface model (similar radius), while the other two have the same radius and are farther from the surface. The purpose of this volume is to test the influence of the model estimation in the final configuration of the surface.

Fig. 6.9 (a) shows the detected patches. As expected, the model detects three large patches, each corresponding to a different portion of a sphere.

In this case, there 8 possible interpretations: assuming I_i^1 corresponds to the larger patch (red); I_i^2



(a) (b)

Figure 6.9: (a) Patch detection in a synthetic volume containing sphere portions. (b) Configuration of the surface after a few iterations.

corresponds to the closer patch (green); and I_i^3 is the remainder (blue), Table 6.1 shows all the interpretations.

	I^1	I^2	I^2
I_1	0	0	0
I_2	0	0	1
I_3	0	1	0
I_4	0	1	1
I_5	1	0	0
I_6	1	0	1
I_7	1	1	0
I_8	1	1	1

Table 6.1: Existing interpretations. Each I_i is an interpretation labeling I^j as valid or invalid.

In the first iteration, the association probabilities were as shown in Table 6.2. Note that I_7 and I_8 automatically receive an association probability of 0 because I^1 and I^2 overlap (see Section 5.2).

α_1	0.0296
α_2	0.0766
α_3	0.1192
α_4	0.3089
α_5	0.1297
α_6	0.3361
α_7	0
α_8	0

Table 6.2: Association probabilities of the interpretations in Table 6.1.

The highest association probability α was assigned to the interpretation $I_6 = \{1, 0, 1\}$, which corresponds to the interpretation that validates the two farthest patches. However, note that the interpretation $I_4 = \{0, 1, 1\}$ also has a high value, which means the both are very likely. Nonetheless, these association probabilities ultimately prefer I_6, leading to the adaptation towards the farther patches in just after a few iterations, as shown in Fig. 6.9 (b).

In a situation such as this, the model's behavior is dependent on the combination of the prior probability of the patches (dependent on their areas) and the likelihood of the interpretations (dependent on the average distance from the surface). Even though the green patch is closer to the surface, the model still prefers the interpretation with the combined larger area.

6.5 Segmentation of the LV in Echocardiography Data

The segmentation system was applied to four different echocardiographic volumes, courtesy of Dr. Jacinto Nascimento. Volumes were randomly selected and represent the heart in various stages of the heart beat.

As explained in Chapter 3.4, the segmentation procedure begins with the initialization of the model, that requires a rough manual segmentation of the LV. This initialization scheme will be shown next for the first volume, but will be omitted for all the other volumes as it does not contain any relevant information. Variations in the initialization procedure will be tested in Section 6.7. Fig. 6.10 shows the manual segmentation in each of the three reference planes of volume 1 and Fig. 6.11 shows the corresponding surface after adaptation to the space carved volume.

The following subsections show the results of the segmentation system obtained for each volume. The segmentation of each volume will be presented in the same way (to facilitate a qualitative comparison): we will show a middle-slice from each volume side-by-side with its correspondent segmentation (intersection of the surface with the slice) (Figures 6.12, 6.14, 6.16 and 6.18) and also the final configuration of the surface (Figures 6.13, 6.15, 6.17 and 6.19). Table 6.3 presents the number of iterations required for each segmentation and the corresponding elapsed time.

	Volume 1	Volume 2	Volume 3	Volume 4
Number of iterations	54	65	70	63
Elapsed time $/s$	10.7	12.9	14.8	13.2

Table 6.3: Number of iterations required and time spent in the adaptation procedure (after initialization) of each volume.

Volume 1

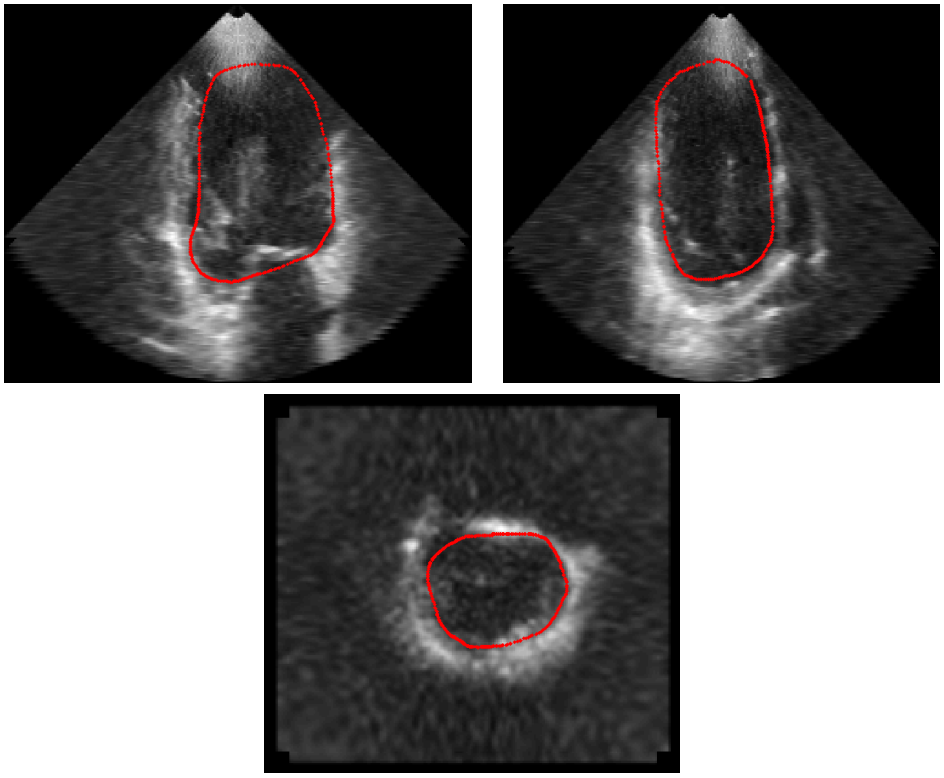


Figure 6.10: Manual segmentation in each reference plane used in the initialization of the model (volume 1).

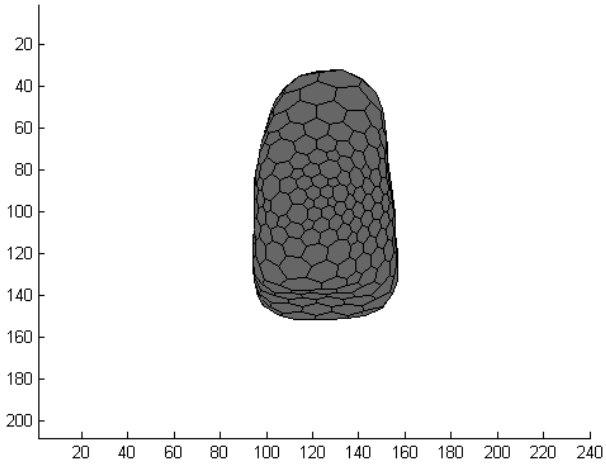


Figure 6.11: Initial simplex mesh (volume 1).

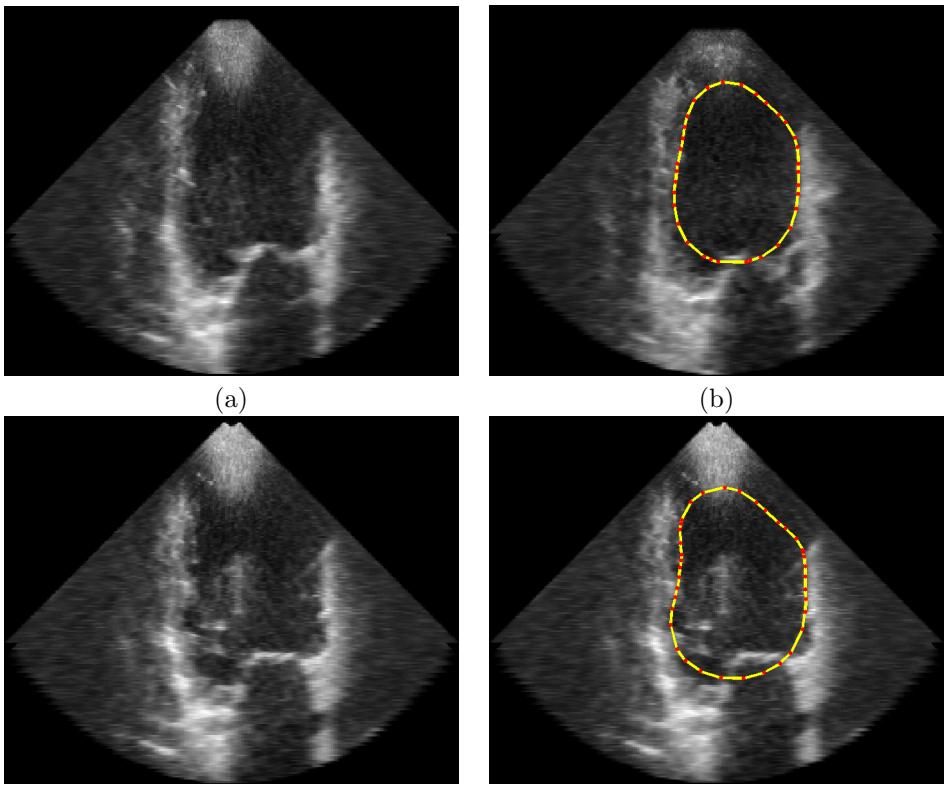


Figure 6.12: Slice from the echocardiographic volume 1 before ((a) and (c)) and after the segmentation ((b) and (d)).

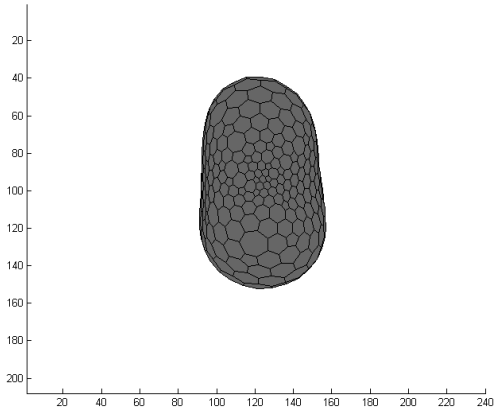


Figure 6.13: Final configuration of the surface (volume 1).

Volume 2

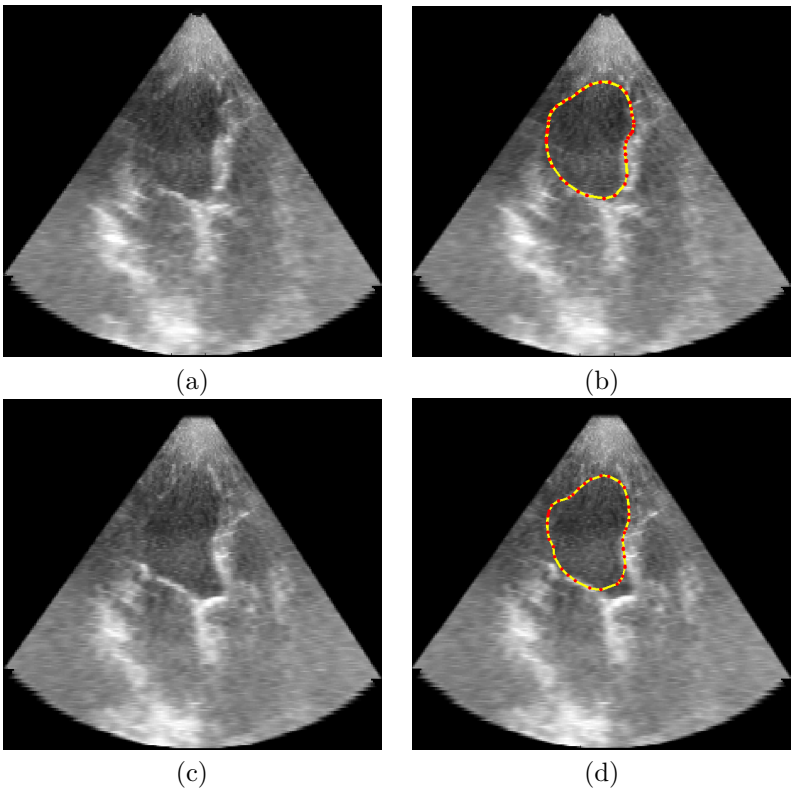


Figure 6.14: Slice from the echocardiographic volume 2 before ((a) and (c)) and after the segmentation ((b) and (d)).

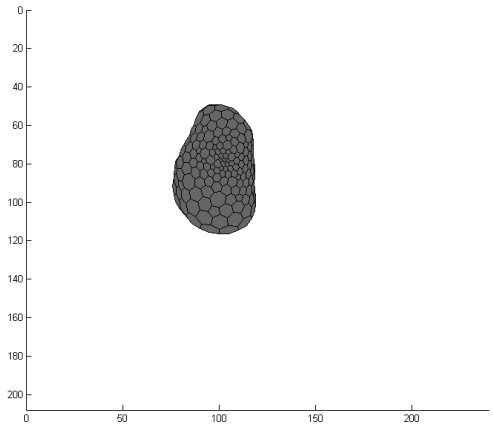


Figure 6.15: Final configuration of the surface (volume 2).

Volume 3

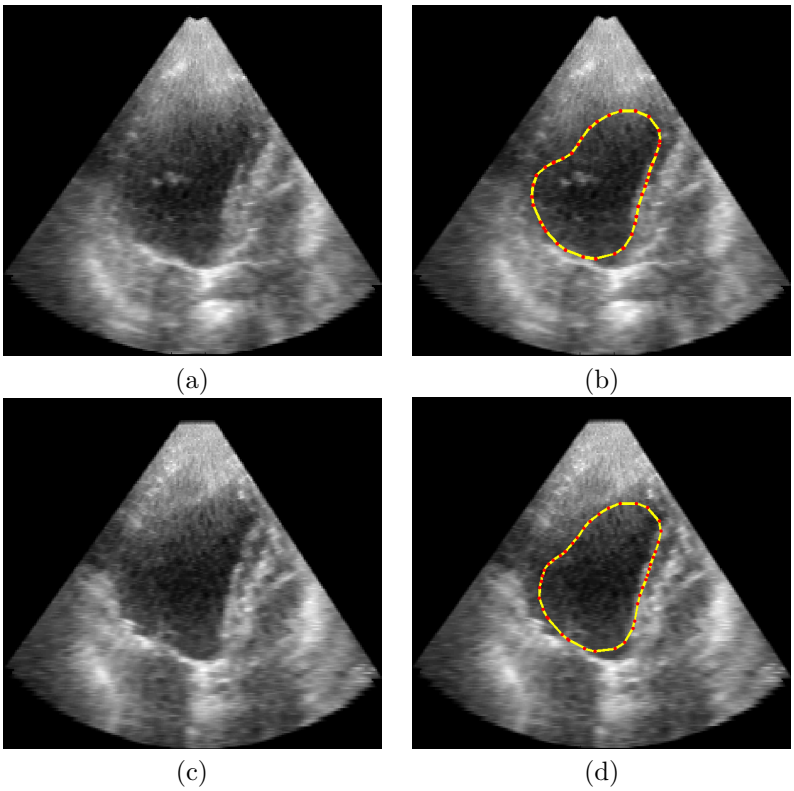


Figure 6.16: Slice from the echocardiographic volume 3 before ((a) and (c)) and after the segmentation ((b) and (d)).

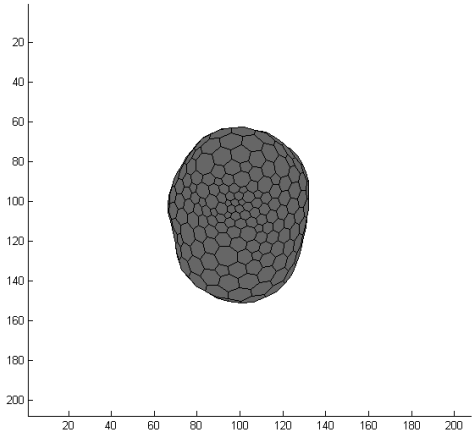


Figure 6.17: Final configuration of the surface (volume 3).

Volume 4

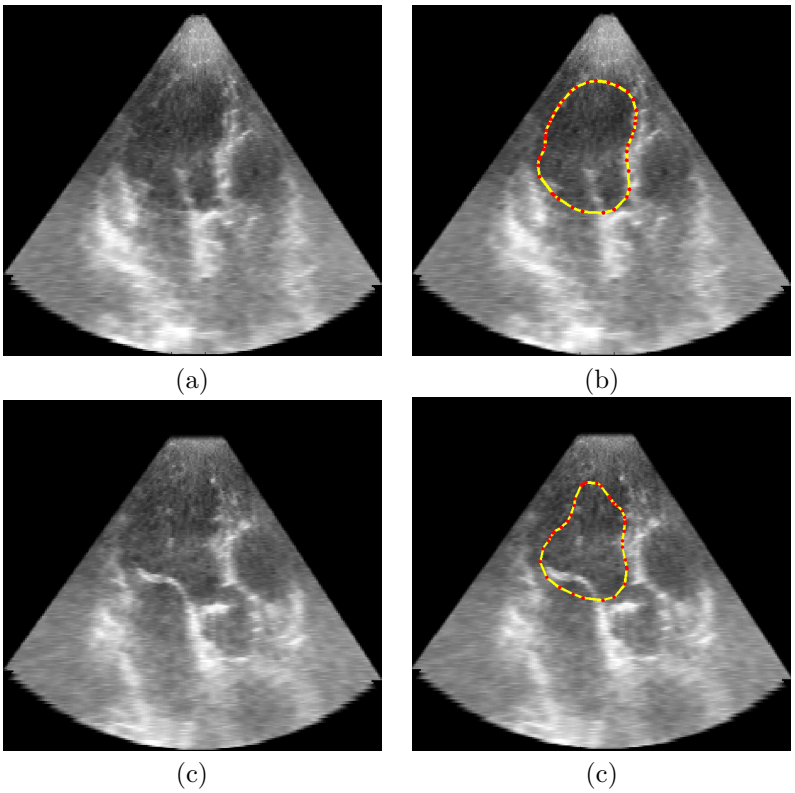


Figure 6.18: Slice from the echocardiographic volume 4 before ((a) and (c)) and after the segmentation ((b) and (d)).

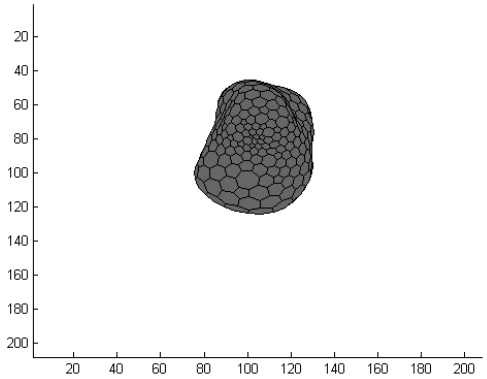


Figure 6.19: Final configuration of the surface (volume 4).

Quantitative Assessment of the Results

For a qualitative analysis of the results, Fig. 6.20 shows an example slice of each volume containing the proposed segmentation (in yellow) and the GT (in green). The images show that the estimated and GT contours are different.

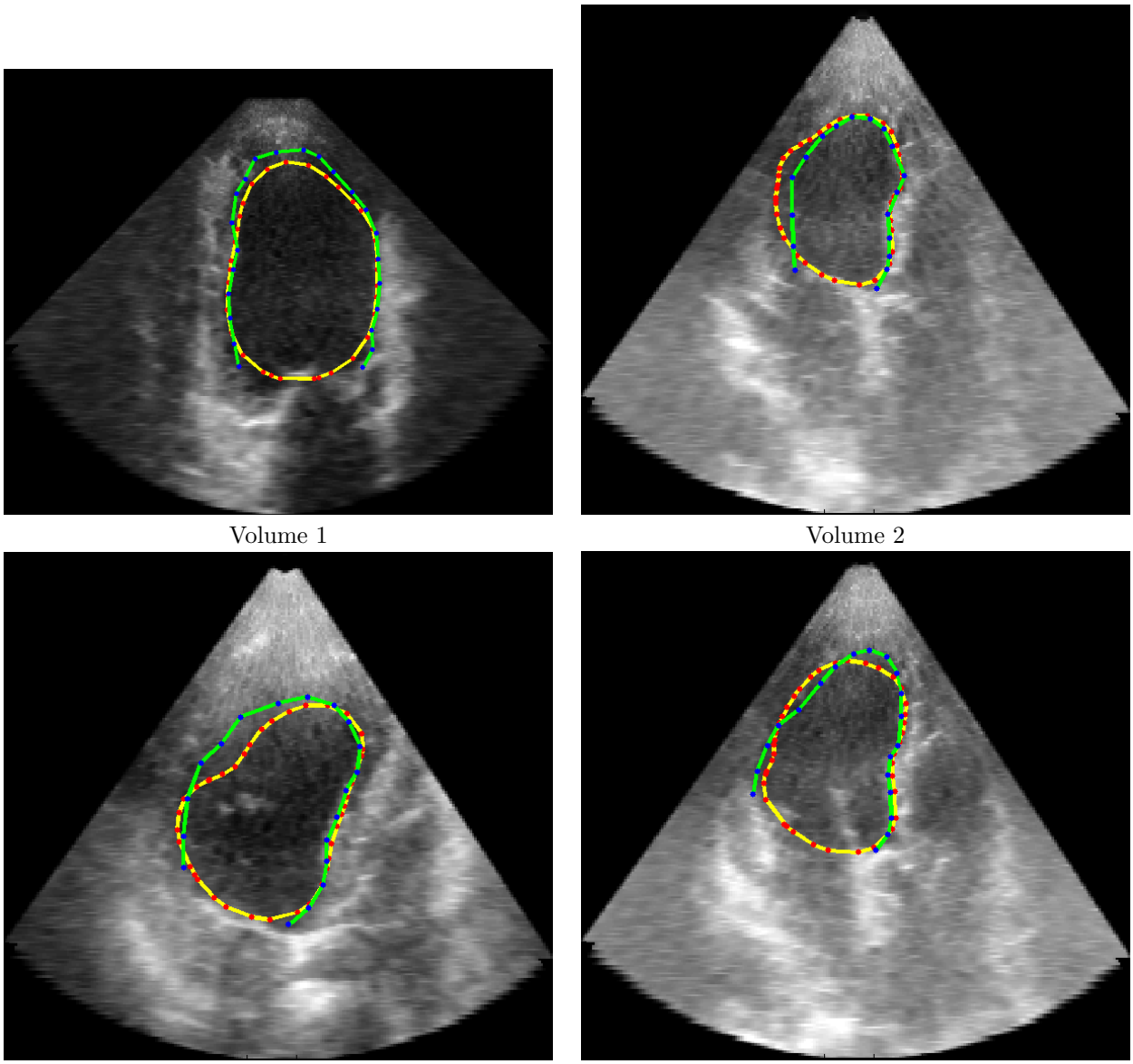


Figure 6.20: Proposed segmentation (yellow contour) and the corresponding GT (green contour).

Although the previous results show that the developed segmentation system performs reasonably well, quantification measurements are required to compare its performance with other similar methods. For each volume, the GT is a set of $4-8$ slices equally spaced and manually segmented. As explained in Section 6.2, to compare each segmentation slice and its corresponding GT the following metrics are applied :

1. Hammoude metric, d_{HMD};
2. Average metric, d_{AV};

3. Hausdorff metric, d_{HDF};

4. Mean Sum of Squared metric, d_{MSSD};

5. and Mean Absolute metric, d_{MAD}.

Note that these metrics can only be applied to 2D images. For an overall performance analysis, we compute each metric's average based on all slices with available GT. The results of this analysis are shown in Table 6.4.

	Volume 1	Volume 2	Volume 3	Volume 4
d_{HMD}	$.17 \pm 0.04$	0.25 ± 0.07	0.20 ± 0.04	0.26 ± 0.13
d_{AV}	4.1 ± 0.8	4.1 ± 1.0	4.1 ± 0.9	4.2 ± 1.1
d_{HDF}	9.9 ± 1.8	12.6 ± 4.5	10.3 ± 2.0	12.8 ± 6.3
d_{MSSD}	51 ± 26	83 ± 44	37 ± 17	65 ± 93
d_{MAD}	6.3 ± 1.7	7.7 ± 2.3	5.2 ± 1.4	6.0 ± 3.5

Table 6.4: Results of the evaluation methods for each volume.

Notice that almost all values have high standard deviations. This is a consequence of having slices in which the segmentation differs significantly from the GT and others where the contour almost matches, such as the slices in Fig. 6.20. Fig. 6.21 shows an example (from volume 2) in which the proposed segmentation does not match the expected.

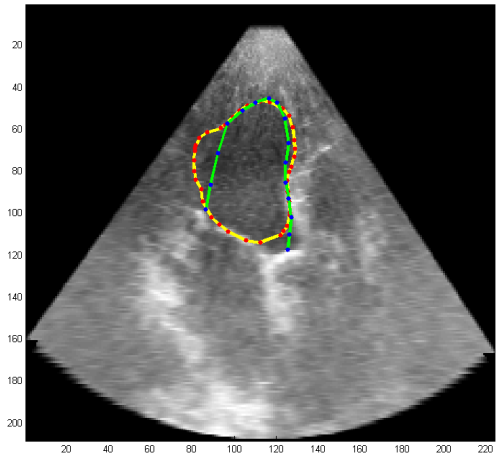


Figure 6.21: Proposed segmentation (yellow contour) and the corresponding GT (green contour).

6.6 Comparison with Classic Model Estimation

In order to demonstrate the robustness of the developed system, we will compare it with the classic model estimation technique, based on the Kalman filter described in Section 5.1.

The Kalman filter will be applied to volumes 1 to 4 and the results will be presented using one slice from each volume, in which all three segmentations will be shown: contour obtained using the developed robust model (yellow contour), using the classic technique (red contour) and the GT (green contour). The results are shown in Fig. 6.22.

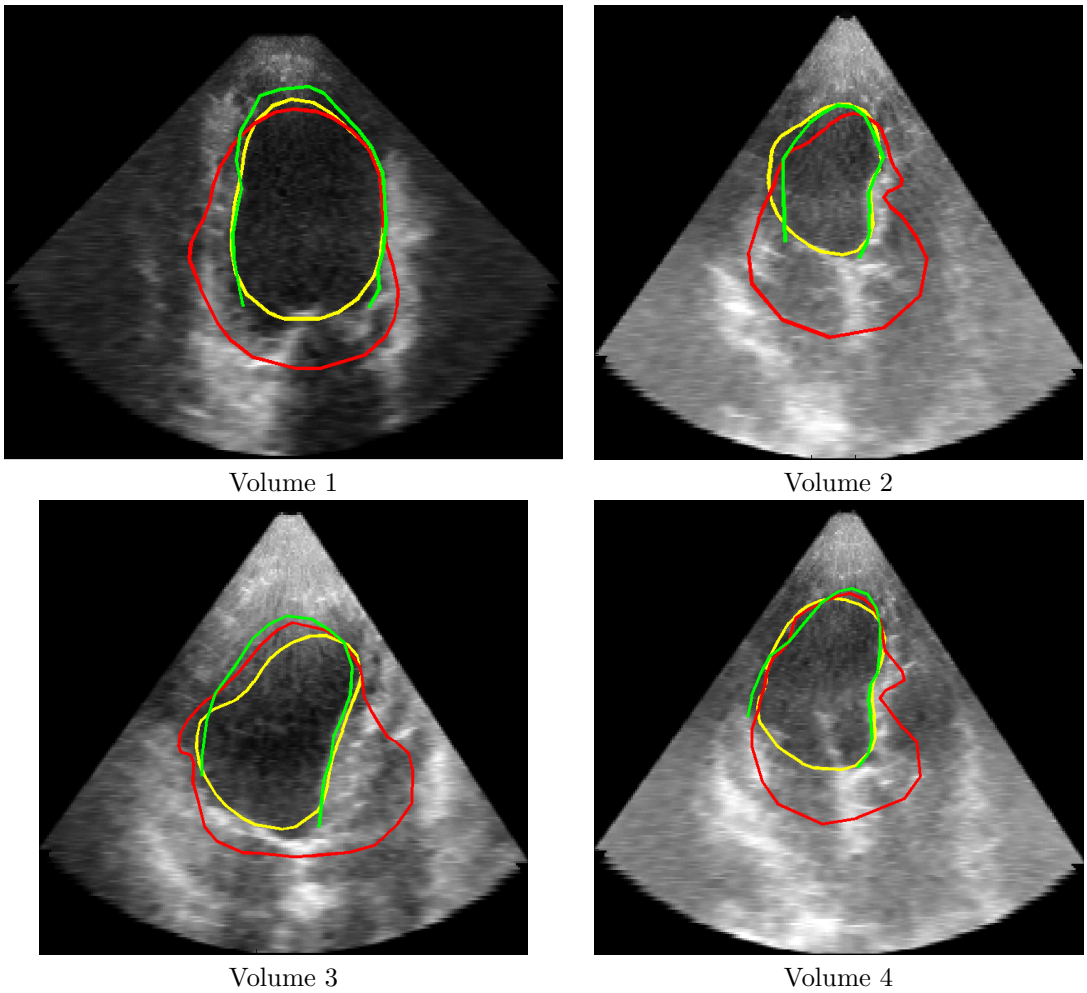


Figure 6.22: Segmentation using the developed model (yellow contour), the Kalman filter (red contour) and the corresponding GT (green contour).

As the figure shows, the classic model estimation is greatly influenced by all the detected features, particularly outside the LV chamber. The Kalman contour is pulled outside the LV boundary and is even attracted to the other structures in the image, such as the heart atria and right ventricle. This clearly demonstrates how ineffective traditional estimation technique are in echocardiographic data.

6.7 Parameter Variation

One of the objectives of using automatic segmentation systems is to improve the repeatability of the segmentations. Therefore, it is also important to verify the robustness of the model to the input parameters, such as the manual segmentation in the initialization procedure and the labeling threshold in middle-level features assembly.

Initialization Procedure

Robustness to initial conditions is of great importance. Since the initialization procedure requires (a rough) user input, it is interesting to analyze the its influence in the final segmentation. Ideally, the final configuration of the surface should be the same for similar initializations, otherwise, the proposed segmentation system

would not solve the repeatability problem of manual segmentation. To test this, we will present two examples in which different initializations are used to segment the same volume.

Using volume 1 as the dataset, Fig. 6.23 shows the two manual segmentations used in each segmentation. Notice that in example 1 we purposely overshoot the LV boundary, whereas in example 2 the surface is initialized well within the LV chamber.

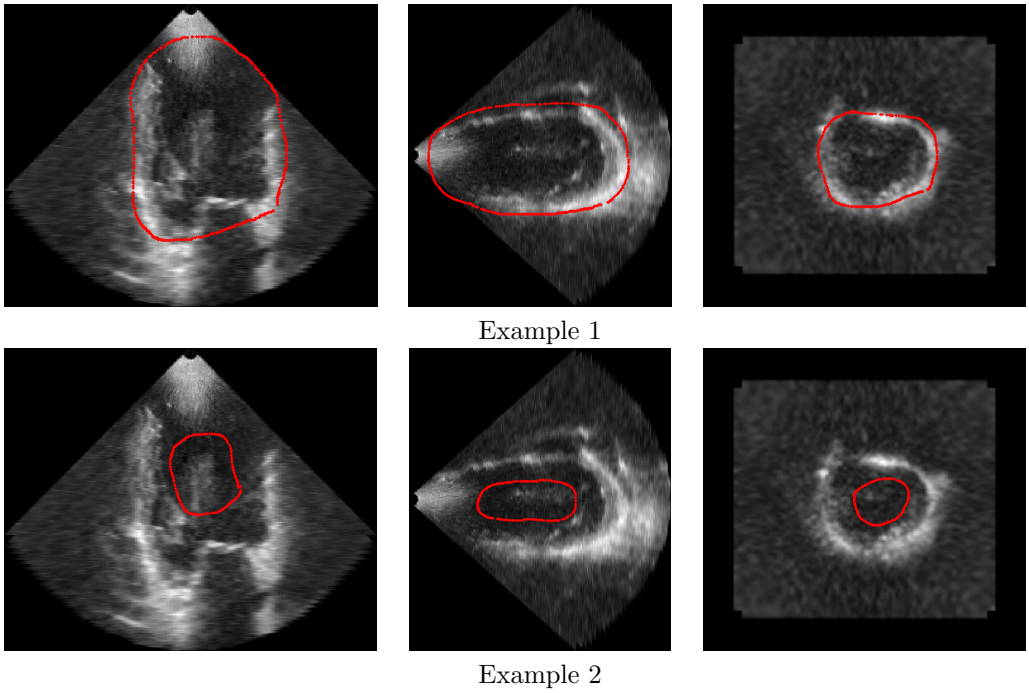


Figure 6.23: Manual segmentations used in the initialization of Example 1 and 2.

The resulting segmentations can be viewed in Fig. 6.24. The image shows that the final configuration of the surface is significantly influenced by the two initializations used. For a quantitative analysis, the Hammoude distance was computed, yielding the value: $\bar{d}_{HMD} = 0.24 \pm 0.02$.

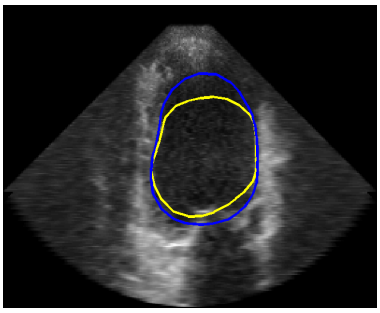


Figure 6.24: Slices from the final segmentation of volume 1 using different initializations.

The reason for the difference in the final segmentation is that the multiple observations are assigned different association probabilities depending on their distance to the model and that distance is related with the initialization of the model.

Initializing the model with five rough manual segmentations and comparing these with the GT resulted in the Hammoude distance values in Table 6.5. Results show that the Hammoude distance is similar in each run, with an average of $\bar{d}_{HMD} = .19 \pm 0.02$.

	Test 1	Test 2	Test 3	Test 4	Test 5
d_{HMD}	$0.19 \pm .03$	0.17 ± 0.01	0.21 ± 0.02	0.20 ± 0.02	0.16 ± 0.03

Table 6.5: Hammoude distance between the GT and five tests on volume 1.

Labeling Threshold

In this next example different labeling thresholds will be used in order to analyze the influence of middle-level features in the segmentation. The labeling threshold defines the maximum allowed distance between features in a patch. Variations in this threshold lead to different patch extractions, which may have significant influence on the final segmentation.

The labeling threshold is a distance measurement and, thus, depends on the image dimensions. However, the volumes used in this chapter have similar dimensions, which allows the definition of a global labeling threshold.

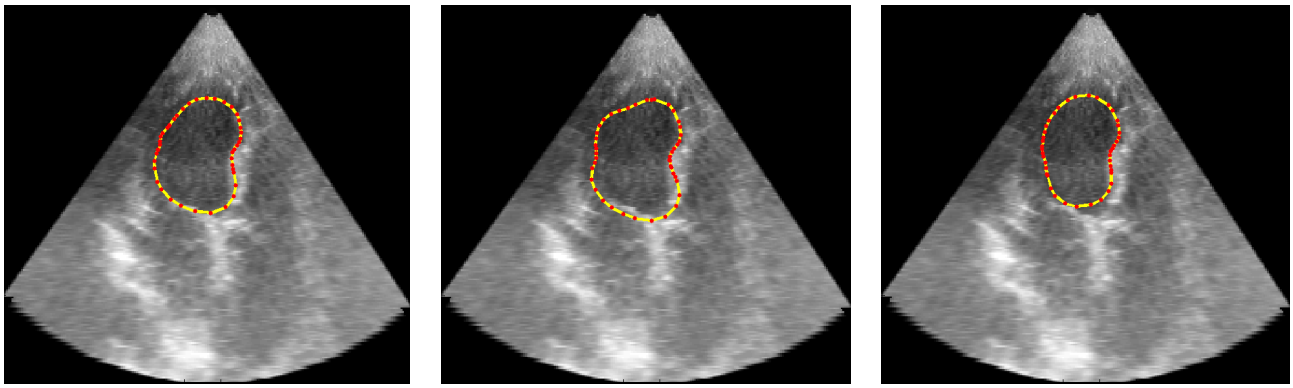
All previous results were obtained using a labeling threshold of $t_l = 8$ (about 4% of the image size). Three different threshold will be used next in the segmentation of volume 2: $t_l = 4$, $t_l = 16$ and $t_l = \infty$. Lower labeling thresholds were actually tried, but the number of extracted patches significantly increased, which led to unbearable computational costs. Fig. 6.25 shows a slice of the final segmentation and Table 6.6 shows the Hammoude distance to the GT. Results show that the Hammoude distance of the final configurations to the GT does not differ significantly. However, comparing the segmentations with each other yields the Hammoude distances in Table 6.7, which shows that there is a high variability between each segmentation.

	$t_l = 4$	$t_l = 16$	$t_l = \infty$
d_{HMD}	0.22 ± 0.12	0.23 ± 0.05	0.22 ± 0.06

Table 6.6: Hammoude distance between the GT and two segmentations using different labeling thresholds, t_l, on volume 2.

$\bar{d}_{HMD}$	$t_l = 4$	$t_l = 16$	$t_l = \infty$
$t_l = 4$	0	0.24 ± 0.03	0.25 ± 0.11
$t_l = 16$	-	0	0.20 ± 0.07

Table 6.7: Hammoude distance between segmentations using different labeling thresholds, t_l, on volume 2.



$t_l = 4$ $t_l = 16$ $t_l = \infty$

Figure 6.25: Slices from volume 2 containing the final segmentation using different labeling thresholds.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis proposes an automatic LV segmentation system for 3D echocardiographic data. The proposed algorithm uses a robust model estimation technique which is an extension of the S-PDAF algorithm to 3D datasets.

The developed segmentation system is composed of three main parts: the deformable surface, the feature extraction and the robust model estimation. Each part sought to achieve a fast and robust segmentation system able to correctly perform the segmentation of the LV in ultrasound images.

The surface model used is a simplex mesh, which is a simple and efficient meshed surface that deforms based on internal and external forces that tries both to maintain a smooth and coherent surface and to adjust to the LV boundary. This deformable surface is initialized using a preliminary step where user interaction is required, in order to guarantee that the model is initialized in the vicinity of the LV.

The feature extraction can be decomposed into two steps: the detection of features in the image and the grouping of these feature into middle-level patches. In the first step, a directional search is performed at each vertex of the surface, from which may result several features. Many of the detected features are outliers (do not belong to the LV boundary), which hampers the performance of the segmentation system. The second step is used to increase the robustness of the model by grouping features into patches. These patches have two main advantages: they significantly decrease the number of features and they help identify if the features are produced by the LV boundary or by the background.

The robust model estimation part is used to further increase the robustness of the segmentation system. Based on a probabilistic model that assigns a confidence degree to each possible combination of valid and invalid patches (interpretations), the model uses a weighted average of the detected features to determine the model estimation.

The results shown clearly demonstrate the advantage of the robust estimation scheme not only in the various echocardiographic volume segmented but also when compared to classic estimation techniques. The results also show that the developed system is slightly over-dependent on the initialization procedure, which does not help improving the repeatability of LV segmentations. Nonetheless, the proposed system shows very promising results.

7.2 Future Work

In future works, several aspects of the proposed system can be improved. To begin with, the initialization procedure can be replaced by an automatic initialization scheme, such as the one proposed in [64]. This would eliminated the need for user interference, making the system completely automated. On the other hand, allowing cardiologists to interactively correct the model whenever needed would also increase the potential of the segmentation system.

Secondly, the proposed system avoids using prior knowledge of the LV shape in the adaptation scheme to keep the model flexible to unexpected conditions. However, the proposed system can still be coupled with other techniques that use prior knowledge the help the adaptation process without strongly relying on these to achieve the final segmentation. For instance, using a deformable template to generate a coarse surface model of the LV and then using the proposed system for a better final fit to the LV boundary would possibly result in a faster algorithm and possibly achieving better final results.

Furthermore, coupling the proposed system with image filtering techniques to enhance the LV boundary could also improve the robustness of the model, but at the cost of lower performance times.

Naturally, the segmentation system could be coupled to automatic quantification algorithms. For instance, the final surface could be used to automatically compute valuable medical information of the LV, such as end-systolic and end-diastolic volumes and the corresponding ejection fraction. This would further increase the clinical value of the proposed method.

Bibliography

[1] Wikipedia. `http://en.wikipedia.org/wiki/Trilinear_interpolation`.

[2] F. Aurenhammer and R. Klein. *Handbook of Computational Geometry*. 2000.

[3] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Associaton*. New York: Academic, 1988.

[4] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.

[5] O. Bernard, B. Touil, A. Gelas, R. Prost, and D. Friboulet. A rbf-based multiphase level set method for segmentation in echocardiography using the statistics of the radiofrequency signal. In *ICIP*, 2007.

[6] A. Blake and M. Isard. *Active Contour*. Springer, 1998.

[7] T. Boettger, I. Wolf, T. Kunert, S. Mottl-Link, M. Hastenteufel, R. De Simone, and H.P. Meinzer. Semiautomatic 3d-segmentation of live-3d echocardiographic images. *Computers in Cardiology*, pages 73–76, 2004.

[8] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber. Automatic segmentation of echocardiographic sequences by active appearance motion models. *IEEE Trans. Med. Imag.*, 21(11):1374–1383, 2002.

[9] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu. Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree. *IEEE Trans. Med. Imag.*, 27(9):1342–1355, 2008.

[10] G. Carneiro and J. C. Nascimento. Multiple dynamic models for tracking the left ventricle of the heart from ultrasound data using particle filters and deep learning architectures. In *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2010.

[11] G. Carneiro, J. C. Nascimento, and A. Freitas. Robust left ventricule segmentation from ultrasound data using deep neural networks and efficient search methods. In *Int. Symp. Biomedical Imaging: from nano to macro (ISBI)*, 2010.

[12] V. Caselles, F. CattÃ©, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.

[13] K Chan and J. P. Veinot. *Anatomic Basis of Echocardiographic Diagnosis*. Oxford Univesity Press, 1st edition, 2011.

[14] T. Chen, J. Babb, P. Kellman, L. Axel, and D. Kim. Semiautomated segmentation of myocardial contours for fast strain analysis in cine displacement-encoded MRI. *IEEE Trans. Med. Imag.*, 27(8):1084–1094, 2008.

[15] L. D. Cohen. On active contour models and balloons. 61(1):38–59, 1991.

[16] T. Cootes, C. Beeston, G. Edwards, and C. Taylor. A unified framework for atlas matching using active appearance models. In *Information Processing in Medical Imaging*, pages 322–333, 1999.

[17] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Comput. Vis. Image Understand.*, 61(1):38–59, 1995.

[18] T. F. Cootes and C. J. Taylor. Active shape models - 'smart snakes'. *Proc. British Machine Vision Conf.*, pages 9–18, 1992.

[19] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti. Left ventricular volume estimation for real-time three-dimensional echocardiography. *IEEE Trans. Med. Imag.*, 21(9):1202–1208, 2002.

[20] S. Cotin, H. Delingette, and N. Ayache. General object reconstruction based on simplex mesh. *IEEE Trans. Visual. Comput. Graphics*, 5(1):62–73, 1999.

[21] E. Debreuve, M. Barlaud, G. Aubert, I. Laurette, and J. Darcourt. Space-time segmentation using level set active contours applied to myocardial gated SPECT. *IEEE Trans. Med. Imag.*, 20(7):643–659, 2001.

[22] H. Delingette. General object reconstruction based on simplex mesh. *International Journal of Computer Vision*, (32):111–142, 1999.

[23] A.F. Frangi, W.J. Niessen, and M.A. Viergever. Three-dimensional modeling for functional analysis of cardiac images, a review. *IEEE Trans. Med. Imag.*, 20(1):2–5, 2001.

[24] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta. Database-guided segmentation of anatomical structures with complex appearance. In *Conf. Computer Vision and Pattern Rec. (CVPR)*, 2005.

[25] A. Hammoude. *Computer-assisted Endocardial Border Identification from a Sequence of Two-dimensional Echocardiographic Images.* PhD, University Washington, 1988.

[26] A. Hammoude. Endocardial border identification in two-dimensional echocardiographic images: review of methods. *Computerized Medical Imaging and Graphics*, 22(3):181–193, 1998.

[27] X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(6):755–768, 2003.

[28] X. Hang, N.L. Greenberg, and J.D. Thomas. Left ventricle quantification in 3d echocardiography using a geometric deformable model. *Computers in Cardiology*, pages 649–652, 2005.

[29] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using hausdorff distance. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(9):850–863, 1993.

[30] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning. A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography. *IEEE Trans. Med. Imag.*, 21(3):226–238, 2002.

[31] R. Juang, E.R. McVeigh, B. Hoffmann, D. Yuh, and P. Burlina. Automatic segmentation of the left-ventricular cavity and atrium in 3d ultrasound using graph cuts and the radial symmetry transform. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 606–609, 2011.

[32] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 4(1):321–331, 1987.

[33] T. Kirubarajan and Y. Bar-Shalom. Probabilistic data association techniques for target tracking in clutter. *Proceedings of the IEEE*, 92(3):536–557, 2004.

[34] J.B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, (7):48–50, 1956.

[35] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, (38):199–218, 2000.

[36] N. Lin, W. Yu, and J. Duncan. Combinativemulti-scale level set framework for echocardiographic image segmentation. *Medical Image Analysis*, 7(4):529–537, 2003.

[37] M. Lynch, O. Ghita, and P. F. Whelan. Segmentation of the left ventricle of the heart in 3-D+t MRI data using an optimized nonrigid temporal model. *IEEE Trans. Med. Imag.*, 27(2):195–203, 2008.

[38] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(2):158–175, 1995.

[39] J. S. Marques and M. A. T. Figueiredo. Image super-segmentation: Segmentation with multiple labels from shuffled observations. *IEEE International Conference on Image Processing*, 2011.

[40] 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.

[41] T. McInerney and D. Terzopoulos. *Medical Image Analysis*, volume 1. Oxford Univesity Press, 1996.

[42] M. Mignotte, J. Meunier, and J. Tardif. Endocardial boundary estimation and tracking in echocardiographic images using deformable template and markov random fields. *Pattern Analysis and Applications*, 4(4):256–271, 2001.

[43] S. Mitchell, B. Lelieveldt, R. van der Geest, H. Bosch, J. Reiber, and M. Sonka. Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac MR images. *IEEE Trans. Med. Imag.*, 20(5):415–423, 2001.

[44] J. Montagnat and H. Delingette. Globally constrained deformable models for 3d object reconstruction. *Signal Processing*, 71(2):173–186, 1998.

[45] D. Musicki, B.F. La Scala, and R.J. Evans. Integrated track splitting filter–efficient multi-scan single target tracking in clutter. *IEEE Trans. Aerosp. Electron. Syst.*, 43(4):1409–1425, 2007.

[46] J. C. Nascimento and J. S. Marques. Robust shape tracking in the presence of cluttered background. *IEEE Trans. Multimedia*, 6(6), December 2004.

[47] J. C. Nascimento and J. S. Marques. Robust shape tracking with multiple models in ultrasound images. *IEEE Trans. Image Processing*, 17(3):392–406, 2008.

[48] M.M. Nillesen, R.G.P. Lopata, I.H. Gerrits, L. Kapusta, H.J. Huisman, J.M. Thijssen, and C.L. de Korte. 3d segmentation of the heart muscle in real-time 3d echocardiographic sequences using image statistics. *IEEE Ultrasonics Symposium, 2006*, pages 1987–1990, 2006.

[49] F. Orderud. *Real-time segmentation of 3D echocardiograms using a state estimation approach with deformable models*. PhD thesis, Norwegian University of Science and Technology, Department of Computer and Information Science, 2010.

[50] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169(2):463–502, 2001.

[51] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[52] N. Paragios. A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE Trans. Med. Imag.*, 22(6):773–776, 2003.

[53] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.

[54] K.N. Plataniotis, D. Androutsos, V. Sri, and A.N. Venetsanopoulos. An adaptive nearest-neighbor multichannel filter. *IEEE Trans. Circuits Syst. Video Technol.*, 6(6):699–703, 1996.

[55] A. Sarti, C. Corsi, E. Mazzini, and C. Lamberti. Maximum likelihood segmentation of ultrasound images with rayleigh distribution. *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, 52(6):947–960, 2005.

[56] B.C. Scowen, S.L. Smith, and M.A. Vannan. Quantitative 3d modelling of the left ventrical from ultrasound images. In *Euromicro Conference*, volume 2, pages 432–439, 2000.

[57] T. Shen, X. Huang, H. Li, E. Kim, S. Zhang, and J. Huang. A 3d laplacian-driven parametric deformable model. *Int'l Conf. Computer Vision (ICCV)*, 2011.

[58] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(11), November 1992.

[59] R. Szeliski, D. Tonnensen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Conf. on Computer Vision and Pattern Recognition (CVPR'93)*, pages 82 – 87, June 1993.

[60] G. Welch and G. Bishop. An introduction to the kalman filter, 1995.

[61] C. Xu. *Deformable Models With Application to Human Cerebral Cortex Reconstruction From Magnetic Resonance Images*. PhD, John Hopkins University, 2000.

[62] C. Xu, D.L. Pham, M.E. Rettmann, D.N. Yu, and J.L. Prince. Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Trans. Med. Imag.*, 18(6):467–480, 1999.

[63] C. Xu and J. L. Prince. Snakes, shapes and gradient vector flow. *IEEE Trans. Image Processing*, 7(3):359–369, 1998.

[64] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu. 3d ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers. In *2008 IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'08)*, pages 1–8, 2008.

[65] Jr. Yezzi, A., S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(2):199–209, 1997.

[66] H. Yu, M.S. Pattichis, and M.B. Goens. Robust segmentation and volumetric registration in a multi-view 3d freehand ultrasound reconstruction system. In *Fortieth Asilomar Conference on Signals, Systems and Computers, 2006 (ACSSC '06)*, pages 1058–6393, 2006.

[67] V. Zagrodsky, V. Walimbe, C. Castro-Pareja, J. Xin Qin, Jong-Min Song, and R. Shekhar. Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models. *IEEE Trans. Med. Imag.*, 24(9):1089–1099, 2005.

[68] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE Trans. Med. Imag.*, 27(11):1668–1681, 2008.

[69] X. S. Zhou, D. Comaniciu, and A. Gupta. An information fusion framework for robust shape tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(1):115–129, 2005.