UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Segmentation of the Left Ventricle in Cardiac Magnetic Resonance

Carlos Jorge Andrade Mariz Santiago

Supervisor: Doctor Jacinto Carlos Marques Peixoto Nascimento

Co-supervisor: Doctor Jorge dos Santos Salvador Marques

Thesis approved in public session to obtain the PhD Degree in

Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

2017

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Segmentation of the Left Ventricle in Cardiac Magnetic Resonance

Carlos Jorge Andrade Mariz Santiago

Supervisor: Doctor Jacinto Carlos Marques Peixoto Nascimento

Co-supervisor: Doctor Jorge dos Santos Salvador Marques

Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

Jury

Chairperson: Doctor José Alberto Rosado dos Santos Victor, Instituto Superior Técnico da Universidade de Lisboa

Members of the Committee:

Doctor Joaquim Armando Pires Jorge, Instituto Superior Técnico da Universidade de Lisboa

Doctor Jaime dos Santos Cardoso, Faculdade de Engenharia da Universidade do Porto

Doctor José Manuel Bioucas Dias, Instituto Superior Técnico da Universidade de Lisboa

Doctor João Pedro de Almeida Barreto, Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Doctor Alexandre José Malheiro Bernardino, Instituto Superior Técnico da Universidade de Lisboa

Doctor Jacinto Carlos Marques Peixoto do Nascimento, Instituto Superior Técnico da Universidade de Lisboa

Funding Institutions

Fundação para a Ciência e a Tecnologia

2017

Nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under thy observation in life.

Marcus Aurelius

Abstract

Cardiovascular diseases are the leading cause of death worldwide. While prevention has a significant impact in reducing incidence rates, early detection and management is required to reduce mortality. Cardiac magnetic resonance (CMR) is considered the gold standard to assess cardiac function and anatomy. Despite its high costs, it is an invaluable choice for diagnosing and monitoring several cardiac diseases, because it is a non-invasive, non-ionizing and versatile imaging modality. The evaluation of cardiac function from CMR data requires measuring the left ventricle (LV) volume. This is achieved by manually delineating the LV border in the axial slices of a CMR volume, a time-consuming task that has to be performed by cardiologists. This has fostered the research on the automation of this process.

The goal of this thesis is to develop a segmentation algorithm for the LV in 3D+t CMR data. The proposed approach is based on active shape models (ASM), and it is divided into three parts. The first part of the thesis focuses on extracting the LV border by analyzing each 2D magnetic resonance (MR) slice individually. In this approach, a single 2D shape model is assumed to be capable of representing the LV contour and segment all the slices of the CMR data. The main challenge is the estimation of the shape model parameters from images with misguiding anatomical structures, such as papillary muscles. These structures are responsible for the appearance of many outliers that decrease the accuracy of the segmentations. To deal with this issue, a new Bayesian formulation is proposed that is able to reduce the influence of outliers in the estimation the model parameters.

The second part of the thesis explores using a 3D shape model to segment the whole MR volume simultaneously, instead of dealing with each slice independently. Contrary to the previous approach, a 3D shape model captures the three-dimensional structure of the LV and imposes constraints on the segmentations of the slices in a volume. The main problems faced here are related to the variable number of slices in MR volumes and to the existence of misaligned slices. These issues make both the training and the test phase of

an ASM methodology more challenging. The proposed approach consists of preprocessing the shape examples in the training set to remove misalignments and to resample them to a predefined number of contours. In the test phase, the shape model is adjusted to fit the test volumes with a different number of slices.

The final part takes into consideration the temporal dimension of CMR data. In this case, the segmentation is performed on the whole sequence simultaneously, by taking into account the motion of the LV. Two different approaches are proposed to embed temporal information in the ASM framework: i) using vector fields to represent the LV dynamics during the systolic and the diastolic phases of the cardiac cycle; and ii) imposing the LV dynamics through regularization of the shape model parameters.

In each part, the experimental results show the advantages of the proposed methodologies. Overall, it is possible to conclude that using more information during the segmentation process leads to an improvement of the segmentation accuracy.

Keywords

Cardiac magnetic resonance, Left ventricle, Segmentation, Active shape model, Expectation-maximization

Resumo

Doenças cardiovasculares são a principal causa de morte no mundo. Embora a prevenção tenha um grande impacto na redução da taxa de incidência, a deteção precoce e o tratamento são vitais na redução da mortalidade. Ressonância magnética cardíaca (RMC) é considerada a modalidade de referência na avaliação anatómica e funcional do coração. É uma técnica de imagiologia que acarreta custos elevados, mas é particularmente atrativa para o diagnóstico e acompanhamento de doenças cardíacas por ser não invasiva e não ionizante. Para avaliar a função cardíaca em dados de RMC, é necessário medir o volume do ventrículo esquerdo (VE). Isto é conseguido delineando manualmente a fronteira do VE nos vários cortes axiais dos volumes de RMC, o que é uma tarefa morosa que tem de ser efetuada por cardiologistas. Por este motivo, a automatização desta tarefa tem sido alvo de muita investigação.

O objetivo desta tese é o desenvolvimento de um algoritmo de segmentação do VE em dados 3D+t de RMC. A abordagem proposta baseia-se na utilização de active shape models (ASM), e está dividida em três partes. A primeira parte foca-se na extração do contorno do VE através da análise individual de cada imagem 2D obtida na ressonância magnética (RM). Nesta abordagem, assume-se que um único modelo de forma é capaz de representar o contorno do VE e segmentar todas as imagens de RM. O principal desafio é a estimação dos parâmetros do modelo de forma em imagens com estruturas anatómicas capazes de enganar o modelo, nomeadamente os músculos papilares. Estas estruturas são responsáveis pelo deteção de muitos falsos positivos, que consequentemente levam à diminuição da precisão das segmentações. Para lidar com este problema, é proposta uma nova formulação Bayesiana para a estimação dos parâmetros que diminui a influência dos falsos positivos.

A segunda parte da tese explora a utilização de modelos de forma 3D para segmentar o volume de RM inteiro, ao invés de cada imagem separadamente. Contrariamente à abordagem anterior, o modelo de forma 3D é capaz de representar a estrutura tridimensional do VE e impõe constrangimentos às segmentações das várias imagens de um mesmo

volume. Os principais problemas encontrados estão relacionados com o número variável de imagens que cada volume de MR tem e com a existência de desalinhamentos entre imagens consecutivas. Estes problemas tornam mais difícil tanto a fase de treino como a fase de teste de uma metodologia baseada em ASM. A abordagem proposta consiste em pré-processar os dados de treino por forma a remover desalinhamentos e a reamostrar as formas de treino para que tenham o mesmo número de contornos. Na fase de teste, o modelo de forma aprendido é também ajustado aos volumes de teste, que poderão ter um número diferente de imagens.

A parte final da tese tem em consideração a dimensão temporal dos dados de RM. Neste caso, a segmentação é efetuada na sequência toda simultaneamente, tendo em conta a dinâmica do VE. Duas abordagens distintas foram usadas para incluir informação temporal na metodologia ASM: i) usar campos vetoriais para representar a dinâmica do VE nas fases de sístole e diástole do ciclo cardíaco; e ii) impor a dinâmica do VE através da regularização dos parâmetros do modelo de forma.

Em cada uma destas partes, os resultados experimentais demonstram as vantagens das abordagens utilizadas. Globalmente, também é possível concluir que a utilização de mais informação durante a segmentação é beneficial para a qualidade das segmentações.

Palavras Chave

Ressonância magnética cardíaca, Ventrículo Esquerdo, Segmentação, Active shape model, Expectation-maximization

Acknowledgments

The work developed in this thesis was supported by FCT grant SFRH/BD/87347/2012 and by the FCT projects UID/EEA/50009/2013, PTDC/EEA-CRO/103462/2008 and PTDC/EEIPRO/0426/2014, and would not be possible without the contribution of several people.

The first two I would like to mention are Prof. Jacinto Nascimento and Prof. Jorge Marques, without whom this thesis would not be possible. I started working with both of them during my masters thesis and it is thanks to them that I had the opportunity and pleasure to do research and continue my studies. Thank you for believing in me, for always being there for me, for all the hard work and invaluable contributions, and for your patience and perseverance.

Many of my friends also contributed, directly or indirectly, to this thesis. Namely, Rodolfo Abreu, my tough figure reviewer and my go-to man whenever I needed help in mundane issues that would otherwise consume hours of my life and in not so mundane issues that made this a better thesis overall. I also have to thank him and Joana Pinto for all the moments we've spent enjoying non-working hours, in lunches, dinners, game nights, etc. These years would have been much more boring without your company! I also want to thank Filipe Condessa, Afonso Dias and Pedro Osório for all the interesting discussions that kept challenging me and making me learn new things.

I would also like to thank all the past and present fellow researchers in rooms 6.14 and 6.17, for your friendship and for all the lunch and coffee times we shared, and to all my tango friends for all your kindness and for all those enjoyable moments we've spent together.

Thanks to my parents for all their support, unconditional love, and for making my life so much easier. And last but not least, a special thanks to Catarina Barata for so many reasons that it is impossible no name them all. For making this journey with me, for helping me so much along the way, for making me work hard and making me want to

work hard, for making my life great, for making me happy, for everything, really.

Contents

List of Figures

List of Tables

Abbreviations

AAM	Active appearance model
AC	Active contour
ASM	Active shape model
bSSFP	Balanced steady-state free precession
CMR	Cardiac magnetic resonance
CT	Computerized tomography
CVD	Cardiovascular disease
DP	Dynamic programming
ECG	Electrocardiogram
ED	End-diastolic
EM	Expectation-maximization
EM-RASM	Expectation-Maximization Robust Active Shape Model
ES	End-systolic
GT	Ground truth
GCSP	Global circular shortest path algorithm
ICP	Iterative closest point
IMDP	Iterative multigrid dynamic programming
KKT	Karush-Kuhn-Tucker
LV	Left ventricle
MAP	Maximum *a posteriori*
ML	Maximum likelihood
MR	Magnetic resonance
MRI	Magnetic resonance imaging
OLS	Ordinary Least Squares

PCA	Principal Component Analysis
PDAASM	Probabilistic Data Association Active Shape Model
PMD	Point Distribution Model
PR	Parameter regularization
RANSAC	Random sampling consensus
ROI	Region of interest
RPM	Robust point matching
RV	Right ventricle
SSM	Statistical shape model
TDM	Translational deformation modes
VF	Vector field

1

Introduction

Contents

1.1 Motivation

Cardiovascular diseases (CVDs) are the leading cause of death worldwide, according to the World Health Organization [6]. Although a lot of effort as been put into preventing CVDs by promoting healthier life styles, a large number of people still present behavioral risk factors (*e.g.*, unhealthy diet, sedentary lifestyle, smoking and alcohol abuse). In these situations, the most effective preventive measure is the early detection and management of the disease.

Cardiac magnetic resonance (CMR) is considered the gold standard imaging modality to evaluate cardiac anatomy and function [71]. It allows an accurate, quantitative and highly reproducible assessment of ventricle volumes, ejection fraction, muscle mass, among other medical features. This information is crucial for an objective diagnosis of cardiomyopathies (cardiac muscle diseases) and is often unobtainable with other imaging modalities.

The downside of CMR is the post processing that it requires. In order to extract the desired medical features, it is necessary to know, at the very least, the location of the left ventricle (LV) border, in some of the acquired images. The medical protocol relies in the manual segmentation of these image to achieve this. This means that cardiologists have to analyze the images and manually introduce points along the LV border, through a computer interface. This is a laborious task that consumes 10-15 minutes from the clinical hours of a physician for each patient.

There have been several attempts at developing automatic segmentation algorithm for CMR [138]. However, there are several challenges associated with this process, such as the complexity of CMR images and the high variability of the heart anatomy, which make automation a difficult problem. There is still a great interest in the medical and engineering community to solve this problem, which explains why there have been many LV segmentation challenges in recent years (*e.g.*, MICCAI 2009 [3], MICCAI 2011 [2,167], Kaggle 2016 [5]) and why it is still an active research topic. This is also what motivated the work developed in this thesis, in collaboration with Dr. Pedro Matos from Hospital CUF Infante Santo in Lisbon, Portugal.

1.2 Objectives

The goal of this thesis is the development of an LV segmentation algorithm that would help cardiologists to speed up the process of analyzing CMR data. Ultimately, this could be a tool to relieve them from the laborious task of segmenting the LV, and automatically provide the required medical features used to diagnose and evaluate cardiac function.

CMR data of a patient consists of a 3D+t image (a sequence of volumes), which

depicts the heart moving during one cardiac cycle. Each volume of a specific frame is a stack of 2D images that are located at different heights along the longitudinal axis of the heart. This means that there is a lot of information available that has to be processed.

This thesis was divided into three parts, which address three challenging problems of increasing complexity separately:

1. Segmentation of the LV as a 2D problem: the CMR data is analyzed as individual 2D images, in which the main challenge is identifying the location of the LV border; this problem is addressed in Chapter 3;

2. Segmentation of the LV as a 3D problem: besides the 2D problem, the three-dimensional structure of the LV is taken into account, taking the correlation of the images along the longitudinal axis into account; this problem is tackled in Chapter 4;

3. Segmentation of the LV using temporal information: combine all the previous information with prior knowledge of the LV motion; this is addressed in Chapter 5.

These problems are increasingly more difficult due to the amount of information that the algorithm has to process during the segmentation. However, the additional information also has the potential to improve the accuracy of the segmentations.

Regarding the first part, Chapter 3 takes into account the fact that the LV border has a circular shape in all the slices of the magnetic resonance (MR) volume. In fact, using this shape information has been one of the most popular approaches in CMR segmentation [138]. Still, the identification of the LV border remains challenging due to the lack of contrast between the blood pool (inside the LV) and the myocardium (cardiac muscle surrounding the LV), and due to misleading anatomical structures, namely, papillary muscles (see 2.5).

In Chapter 4, by incorporating the three-dimensional structure of the LV, the prior information about its shape is expected to be more consistent with reality. Furthermore, by analyzing the whole volume simultaneously, the algorithm makes the segmentations coherent along the third dimension.

Finally, the approach explored in Chapter 5 takes advantage of prior knowledge about how the LV varies throughout the cardiac cycle. This allows the algorithm to correlated the segmentations in different frames and to make assumptions about their expected location. This information is combined with the approaches from the previous parts to further enhance the capabilities of the segmentation method.

1.3 Contributions

The problems mentioned above were addressed using an active shape model, which is a deformable model-based approach that uses information about the expected shape of

the segmented region. The main contributions of this thesis are the following:

- A new active shape model algorithm, called Expectation-Maximization Robust Active Shape Model (EM-RASM), for the segmentation of 2D objects (Chapter 3);
- A training and testing framework to use 3D shape models in CMR data (Chapter 4);
- Extension of the EM-RASM algorithm to 3D objects (Chapter 4);
- Extension of the EM-RASM algorithm for 2D+t and 3D+t segmentation:
  - An approach to embed a dynamical model based on vector fields into EM-RASM (Chapter 5);
  - A framework that adds temporal constraints to EM-RASM through parameter regularization (Chapter 5).

1.4 List of Publications

Parts of the work developed during my PhD studies was published in journals, book chapters and conferences, listed below.

Journal Papers

- C. Santiago, J. C. Nascimento, and J. S. Marques, "Fast Segmentation of the Left Ventricle in Cardiac MRI Using Dynamic Programming," *Computer Methods and Programs in Biomedicine*, 2017 - review received (Q1)
- C. Santiago, J. C. Nascimento, and J. S. Marques, "A new ASM framework for left ventricle segmentation exploring slice variability in cardiac MRI volumes," *Neural Computing and Applications*, pp. 1–12, 2016 (selected at IbPRIA for journal publication)
- C. Santiago, J. C. Nascimento, and J. S. Marques, "2D Segmentation Using a Robust Active Shape Model With the EM Algorithm," *Image Processing, IEEE Transactions on*, vol. 24, no. 8, pp. 2592–2601, Aug 2015 (Q1)
- C. Santiago, J. C. Nascimento, and J. S. Marques, "Automatic 3-D Segmentation of Endocardial Border of the Left Ventricle From Ultrasound Images," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 1, pp. 339–348, Jan 2015 (Q1)

Book Chapters

- C. Santiago, J. C. Nascimento, and J. S. Marques, "A 3D Active Shape Model for Left Ventricle Segmentation in MRI," in *Biomedical Image Segmentation: Advances and Trends*. CRC Press, 2016, pp. 99–116

- J. S. Marques, J. C. Nascimento, and C. Santiago, "Robust Deformable Models for 2D and 3D Shape Estimation," in *Deformation Models: Tracking, Animation and Applications*, M. González Hidalgo, A. Mir Torres, and J. Varona Gómez, Eds. Dordrecht: Springer Netherlands, 2013, pp. 169–185

Conference Papers

- C. Santiago, J. C. Nascimento, and J. S. Marques, "Fast and accurate segmentation of the LV in MR volumes using a deformable model with dynamic programming," in *Image Processing (ICIP), 2017 IEEE International Conference on - submitted*. IEEE, 2017

- C. Santiago, J. C. Nascimento, and J. S. Marques, "A New Robust Active Shape Model Formulation For Cardiac MRI Segmentation," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4112–4115

- C. Santiago, J. C. Nascimento, and J. S. Marques, "Segmentation of the left ventricle in cardiac MRI using a probabilistic data association active shape model," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, Aug 2015, pp. 7304–7307

- C. Santiago, J. C. Nascimento, and J. S. Marques, "Robust 3D Active Shape Model for the Segmentation of the Left Ventricle in MRI," in *Pattern Recognition and Image Analysis - IbPRIA'15, **(Best Paper Award)***, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015, vol. 9117, pp. 283–290

- C. Santiago, J. C. Nascimento, and J. S. Marques, "A Robust Active Shape Model Using an Expectation-Maximization Framework," in *Image Processing (ICIP), 2014 21th IEEE International Conference on*. IEEE, 2014, pp. 6076–6080

- C. Santiago, J. C. Nascimento, and J. S. Marques, "Non-rigid Object Segmentation Using Robust Active Shape Models," in *Articulated Motion and Deformable Objects*. Springer, 2014, pp. 160–169

- C. Santiago, J. S. Marques, and J. C. Nascimento, "A robust deformable model for 3D segmentation of the left ventricle from ultrasound data," in *Mathematical Methodologies in Pattern Recognition and Machine Learning*. Springer, 2013, pp. 163–178

- C. Santiago, J. C. Nascimento, and J. S. Marques, "3D left ventricular segmentation in echocardiography using a probabilistic data association deformable model," in *Image Processing (ICIP), 2013 20th IEEE International Conference on.* IEEE, 2013, pp. 606–610

- C. Santiago, J. C. Nascimento, and J. S. Marques, "Performance evaluation of point matching algorithms for left ventricle motion analysis in MRI," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE.* IEEE, 2013, pp. 4398–4401

2

Cardiac Magnetic Resonance

Contents

Over the past twenty years, CMR has become a very popular image modality for the analysis of cardiac structure, function, and tissue characteristics. Advances in scanner capabilities and acquisition techniques have made CMR increasingly faster, with higher resolution, and irreplaceable in the analysis of certain cardiovascular diseases. This chapter is composed of two parts. The purpose of the first part is to give the reader a brief overview of the basic principals of CMR, some of the variants that are used in clinical practice and their applications, and the main advantages over other image modalities. The second part focuses on short axis cine CMR, which are the images that are being analyzed in this thesis. A description of the image characteristics is provided, along with the medical procedure followed by cardiologists in the analysis of these images. Details about the main challenges faced by automatic segmentation algorithms and popular approaches are also provided.

2.1 Fundamentals of Magnetic Resonance

Magnetic resonance imaging (MRI) is a complex imaging modality whose fundamentals are deeply rooted in the relationship between the electromagnetic and quantum physics fields [17, 27, 114, 142]. It is based on a magnetic property of the atoms, known as nuclear spin. Atoms with a non-zero spin, such as hydrogen, generate an electromagnetic field called magnetic moment. In a resting state, the magnetic moments are randomly oriented and cancel each other. However, under the influence of an external magnetic field, they align with the external source. Therefore, instead of canceling, they sum constructively, giving rise to the so-called net magnetization.

During an MRI acquisition, a strong magnetic field is used to align the magnetic moments of the target region. Then, an additional oscillatory field is used to temporarily change the orientation of the net magnetization. Once the exposure to this field ceases, the net magnetization gradually converges to the previous equilibrium, in a process called relaxation. By applying at least three time-varying magnetic fields along the three spatial dimensions, it is possible to specify the location of the excited image sample in the 3D space. The variation of the net magnetic during the relaxation is captured by the MR scanner, which is then translated to a gray scale image. Additional magnetic pulses are often applied in order to perturb the net magnetization and enhance specific features in the image.

The most important source of the MR signal is hydrogen, due to its intrinsic nuclear spin properties, and because it is available in abundance in the human body, particularly in water and fat tissue. However, the intensity of a pixel in an MR image depends both on the type of tissue that pixel corresponds to, and on the pulse sequence used during the acquisition.

The following section briefly describes some of the most common types of CMR

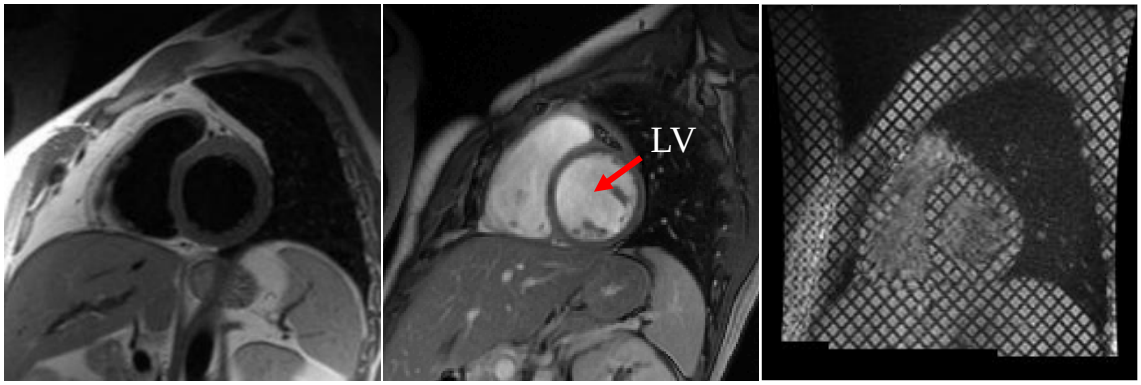


Figure 2.1: CMR imaging modalities: (a) dark blood imaging (extracted from [1]), (b) bright blood imaging, and (c) tagged CMR.

images that are acquired in the clinical setup and their purpose.

2.2 Types of CMR

The ability to generate different types of images is a feature of MRI that make it stand out over other imaging modalities. Different pulse sequences allow different types of contrast between tissues to be acquired. The following subsections explain some of the most commonly acquired CMR images and their uses [71].

2.2.1 Dark Blood Imaging

Dark blood imaging is an acquisition technique that is characterized by assigning high intensity values to tissues that are stationary or moving slowly. Since the blood is always flowing at a higher speed than any other tissue, in this type of images it appears as a dark region (low intensity), as shown in Fig. 2.1 (a).

This imaging modality may be acquired using either spin echo or inversion recovery techniques [77,121], in which the intensity of the image depends on the combined influence of two pulses. However, because the pulses are separated in time, and the blood tissue keeps flowing within the human body, this leads to the absence of signal inside the heart and blood vessels.

This modality is useful in many applications, including the analysis of the morphology of the heart and blood vessels, *e.g.*, for congenital heart disease [129] or the detection of myocardial masses [29].

2.2.2 Bright Blood Imaging

Bright blood imaging is typically used for applications that require a higher temporal resolution. Unlike the previous technique, the blood pool appears brighter than the surrounding muscle tissue. Although single images can also be used, as shown in Fig. 2.1

(b), the most common practice is to acquire a set of images that are then stacked to build a sequence, named cine imaging. This simulates a movie of the heart moving during the cardiac cycle.

This type of images are acquired using gradient echo pulse sequences, namely, the balanced steady-state free precession (bSSFP) technique [171], which are characterized by allowing faster acquisitions without significant loss in spacial resolution [162].

Bright blood imaging, and in particular cine imaging, is the most common type of CMR, used in applications in the assessment of myocardial function of both the left and right ventricles. This is useful in diagnosis and follow-up of diseases such as congestive heart failure or myocardial infarction [142].

2.2.3 Tagged CMR

Myocardial tagging has gained popularity over the past decade for its potential in the evaluation of regional myocardial function [74]. Conventional imaging techniques lack the necessary anatomical landmarks to assess local deformations of the cardiac muscle. Thus, it is only possible to analyze the motion of the inner and outer borders of the LV. With tagged images, however, the muscle tissue is marked with a magnetization pattern, *e.g.*, a grid, that appears as black stripes in the image, as shown in Fig. 2.1 (c). This temporarily creates markers (on the grid nodes) that follow the muscle tissue as it deforms, enabling the quantification of muscle strain and stress.

The two most widely used techniques for tagged CMR are harmonic phase [133] and displacement encoding with stimulated echoes [9]. Both of these approaches rely on marking tissue with a magnetization saturation in a specific pattern, which then appears darker than the neighboring unmarked tissue. They also include a post processing part that aims to track these markers in order to quantify its motion. The analysis is performed on the Fourier transform of the tagged image, from which motion estimates are extracted.

Tagged imaging has been used in the quantification of myocardial dysfunction, as well as enabling the identification of the local contractile capability, which is useful, *e.g.*, in assessing ischemic heart diseases [74].

2.3 Advantages of CMR

CMR is a popular imaging modality in clinical practice. It has several advantages over other modalities [27,71], such as computerized tomography (CT) or ultrasound. First it is non-invasive and non-ionizing, which facilitates the diagnosis and the monitoring of patients without the risks associated to radiation exposure (as in CT). Second, CMR can be acquired in any plane or acquisition view, which does not happen, *e.g.*, in ultrasound imaging, due to the fact that the ultrasonic beam is not able to bypass the chest bones. It can also be easily adjusted, as explained previously, to capture specific types of images, by

changing the pulse sequence of the acquisition, *e.g.*, to show the cardiovascular anatomy and structure, or to analyze myocardial viability and function. This makes it more versatile than other image modalities, and useful in a larger spectrum of applications. Finally, the quality of CMR images in terms of contrast, signal to noise ratio, and spatial resolution makes quantification of specific measurements, such as ventricle volumes and muscle mass, highly reproducible. These advantages make CMR the gold standard for the diagnosis of cardiac function and anatomy, with which other modalities are compared [21, 71].

2.4 Acquisition

This thesis is focused on a specific type of CMR data: bright blood images acquired using the bSSFP technique (recall Section 2.2.2). The goal of this technique is to obtain a sequence of MR volumes (*i.e.*, 3D+t data) comprising a stack of 2D images (slices) that are organized according to the phase of the cardiac cycle they belong to (frame) and to their position along the longitudinal axis (slice position). This section focuses on details about the acquisition procedure, which are useful to understand some the features of the corresponding images.

Although the bSSFP technique is faster at acquiring images than most conventional MRI pulse sequences, it is still not fast enough to capture an entire image associated with a specific phase of the cardiac cycle. Therefore, obtaining CMR data requires a complex acquisition procedure that takes place during several seconds. The standard approach is to divide each image into segments. These segments are small enough to be entirely acquired at a specific time instance (*e.g.*, corresponding to a time interval of 30 ms). Considering each heart beat corresponds to approximately 1 s, and that the cardiac cycle is divided into 20 phases (frames), then it is possible to acquire one segment of all the images in a specific slice position during a single heart beat ($20 \times 30 = 600$ ms < 1 s). If each image is made up of 20 segments, then the total time required to acquire the entire sequence on that slice position is approximately 20 heart beats [114]. Then, this process is repeated for the remaining slice positions.

In order to assign each image with a specific phase of the cardiac cycle, it is necessary to synchronize the acquisition with the electrocardiogram (ECG) of the patient. This is a unique feature of cine CMR that is accomplished using one of the possibilities, illustrated in Fig. 2.2:

Prospective Gating, also denoted by ECG triggering, relies on the detection of the R-wave (which signals the beginning of a new cardiac cycle) to trigger the acquisition of a predefined number of phases (typically 20). In order to avoid extending the acquisition of those phases beyond one cardiac cycle, data acquisition must stop before the next R-wave, which means it is not possible to acquire all the phases of the cardiac cycle.

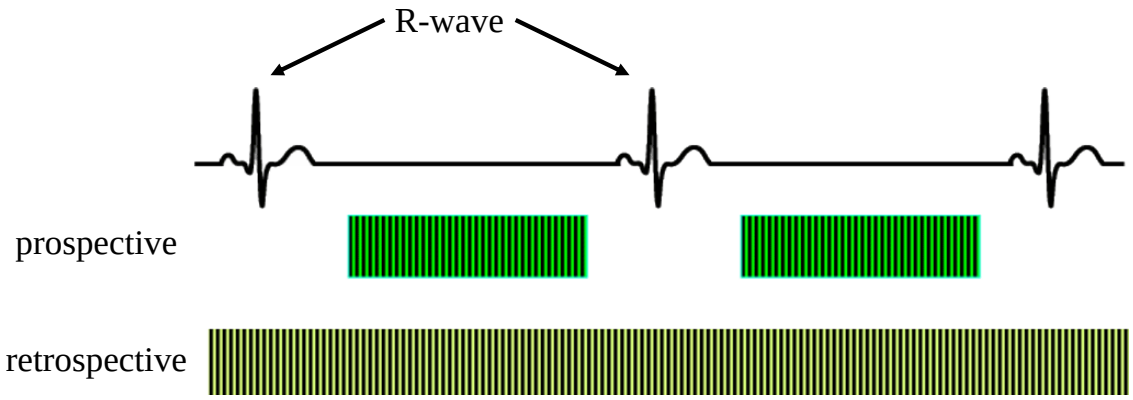


Figure 2.2: Schematic illustration of prospective and retrospective ECG gating (adapted from [4]).

Retrospective Gating, on the other hand, acquires data continually and assigns labels to each segment based on the phase of cardiac cycle it belongs to, relative to the R-wave. Then, the images are obtained by compiling all the segments of each phase, in a retrospective procedure. This allows the MR sequence to be comprised of equally spaced images throughout the whole cardiac cycle.

Older CMR techniques were severely hampered by breathing motion, which caused the heart to change its position during the acquisition. Since it is very difficult to cancel this effect by modeling the complex motion of the heart during the respiratory cycle, the most reliable approach is to acquire CMR during breath-hold [112]. In practice, this means that patients have to hold their breath several times during the acquisition to cover the whole heart. During each breath-hold, the MR scanner acquires the entire sequence of images for a specific slice position. In the example above (see Fig. 2.2), the 20 segments that compose each of 20 MR images in the sequence are acquired while the patient is breath-holding for approximately 20 seconds. Patients are asked to be consistent about the breath-hold position, in order to guarantee that the heart does not change position. Otherwise, motion artifacts may appear between consecutive slices, causing misalignments in the position of the heart in these slices.

2.5 Analysis of CMR

This section discusses the medical procedure that takes place after the acquisition of CMR. As mentioned previously, the bSSFP acquisition technique produces images where the blood inside the ventricle is bright (high intensity value) and the heart's muscle is darker (low intensity value). This difference in intensity values creates a contrast that helps locate the endocardium, *i.e.*, the border between the LV chamber and the myocardium around it (see Fig. 2.1 (b) for an example). The acquired images are combined to form a set of volumes covering one complete cardiac cycle. This includes the end-diastolic (ED)

phase, in which the heart muscles are relaxed and the heart is expanded, and the end-systolic (ES) phase, in which the muscles are contracted and volume is minimal. The difference between the ventricle volume in these two particular phases, denoted as stroke volume, is the amount of blood that is pumped out of the heart during each heart beat. This cardiac performance measurement is used to compute the ejection fraction - required to diagnose and assess cardiac function.

In the clinical practice, after acquiring the CMR, physicians have to personally analyze the data and perform the procedure shown by the diagram in Fig. 2.3[1]. They perform the following tasks:

(1) *Selection of a region of interest (ROI)*, which is region that depicts a close up of the LV in all the slices of the sequence;

(2) *Adjustment of the intensity* to increase the visual contrast between the inside and the outside of the LV;

(3) *Identification of the ED and ES phases*, which are the only ones required for the computation of the ejection fraction; and

(4) *Manual segmentation of the LV* in each slice of these two phases.

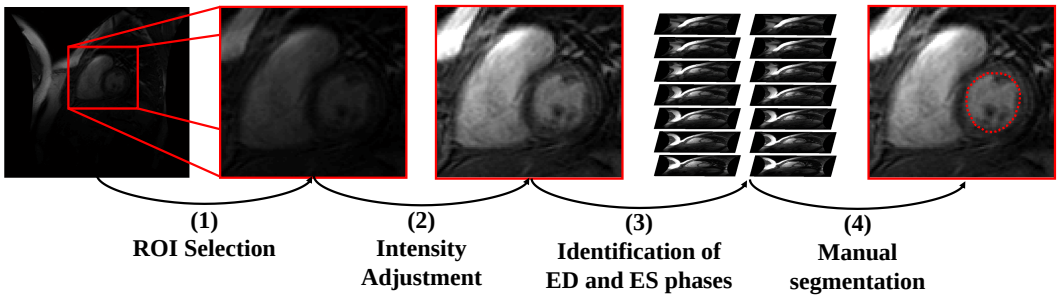


Figure 2.3: Diagram of the workflow for the analysis of a CMR sequence.

After all the segmentations have been done, the computer determines the ventricle volume by using the summation of discs technique [71]. This is based on Simpson's rule, which allows the computation of the volume of a complex structure by summing all the quantifiable subvolumes within it. In practice, this consists in approximating the volume of each slice of the LV by the product of the area of the segmentation (disc area) with the spacing between slices (disc height) and then summing all the slice volumes [26], as shown in Fig. 2.4.

The analysis described above is laborious and time consuming. In particular, the last task, (4) *Manual segmentation of the LV*, requires the physician to manually introduce 10-15 points in each of the 5-10 slices of the MR volume in the ED and ES phases, which

[1]This was the procedure used at Hospital CUF Infante Santo in Lisbon, Portugal, as described by Dr. Pedro Matos on July 2012.

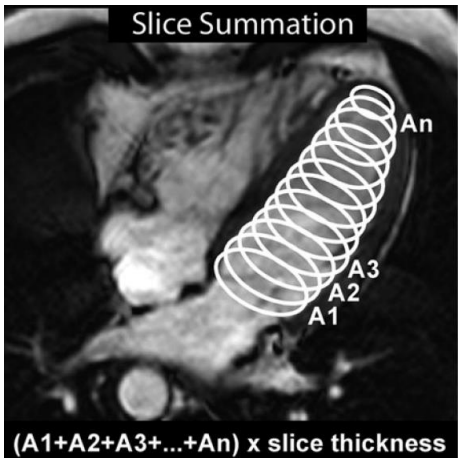


Figure 2.4: Summation of discs technique used to compute the ventricular volume [26]. The figure shows the LV in a longitudinal view and the corresponding slice segmentations that are used to compute the LV volume.

means introducing 100-300 points for a single patient. This is the reason why a lot of effort has been put into the automation of this process. However, obtaining a reliable automatic segmentation of the LV is not a simple task. In the next section, we discuss the problems associated to the automatic segmentation of the LV.

2.6 Automatic Segmentation of the LV

This thesis focuses on the automation of the last task of the analysis of CMR: *Manual segmentation of the LV*. The remaining tasks are additional steps that may be used to improve the performance of the automatic algorithm, but are not fundamental for its completion. The first task, ROI selection, is used by physicians to zoom in on the region they are interested in. In the automation process, this step could be used for initialization purposes, or by imposing that the method should only look for the LV border in a specific region of the slices, thus reducing the computational complexity of the method. The second task, adjustment of the intensity, only influences the way the slices are shown to the physicians, and not their actual content. While this is unnecessary for the automation process, it may be helpful to replace this step with a pre-processing step to improve image quality and facilitate the segmentation. Finally, regarding the third task, identification of the ED and ES phases, the ideal approach would be to select the phases where the LV volume is largest and smallest, respectively. However, this is difficult to do without knowing beforehand the LV segmentations in each phase - the physicians are able to do this by roughly comparing the size of the LV blood pool in each phase [26]. The automatic alternative would be to segment the entire sequence, compute the volume in each phase, and identify the ED and ES phases by comparison of the volumes. This means that this task is easily achievable once the segmentations are available, leaving the

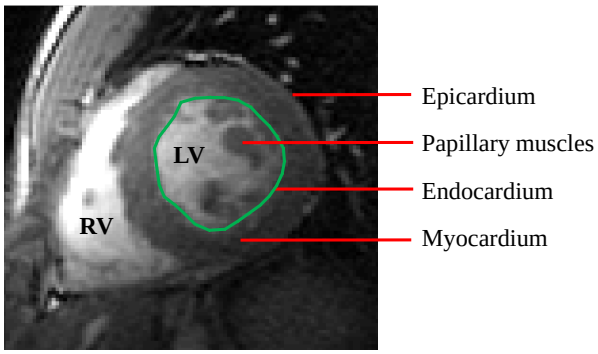


Figure 2.5: Example of a CMR image.

automation of the fourth task as the most important for the whole procedure.

2.6.1 Challenges

Developing an automatic LV segmentation method for CMR involves dealing with three main problems associated to this imaging modality: i) identifying the LV border in each MR slice; ii) slice misalignments caused by different levels of breath-hold; and iii) motion of the LV during the cardiac cycle.

The problem of determining the location of the LV border is the most critical. Fig. 2.5 shows an example of a slice annotated with some of the most important structures in the ROI. The LV blood pool is typically a bright circular structure surrounded by the myocardium, which has a darker intensity. The endocardium is the inner border of the LV that we want do segment (green line). Notice that the image intensity alone does not provide a trustworthy indication of where the endocardium is. The papillary muscles, in particular, are anatomical structures that often mislead automatic segmentation methods that do not impose shape constraints. In other cases where the myocardium is very thin, it is also difficult to distinguish the endocardium from the epicardium in the region opposite to the right ventricle (RV). It is also increasingly difficult to segment the slices near the apex and the base of the LV [167], because the contrast between the blood pool and the surrounding tissues is not so clear (see Fig. 2.6), and due to the proximity to the aortic and mitral valves, respectively. Even among cardiologists, the correct location of the LV border is ambiguous [167], as shown by the examples in Fig. 2.7.

Besides the above problem, two other issues make the automation a difficult task. First, the acquisition process may create translational artifacts in the position of the LV between consecutive slices. This is caused by different levels of inspiration for the breath holding during the acquisition (recall Section 2.4). The consequences of this motion artifact can be seen in the example shown in Fig. 2.8 (a), where the location of the LV in the third and fourth slice is significantly different. Because these misalignments are unpredictable, they are hard to model, and it is difficult to perceive the LV as a 3D shape. Second, the contraction of the heart muscle during the cardiac cycle makes the heart a moving

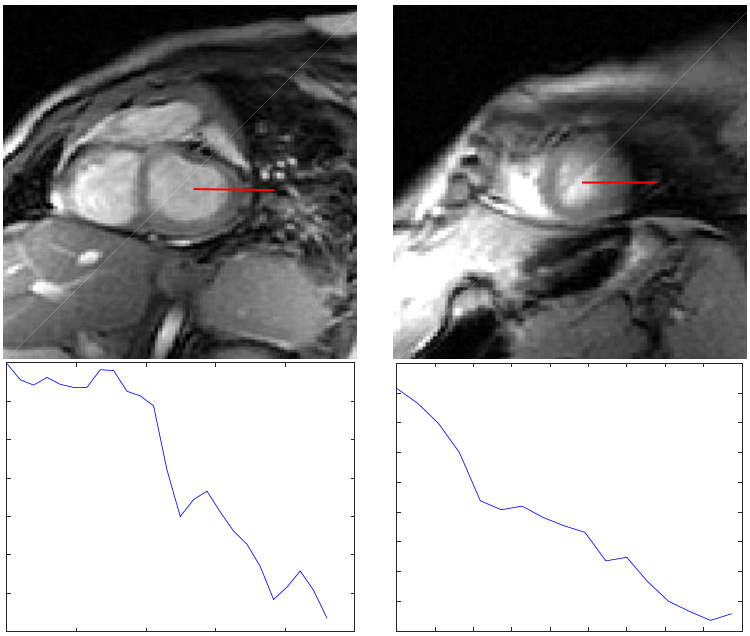


Figure 2.6: Difference in contrast between the blood pool and the myocardium on the basal slice (left) and on the apical slice (right). The images below show the intensity profile along the red line.

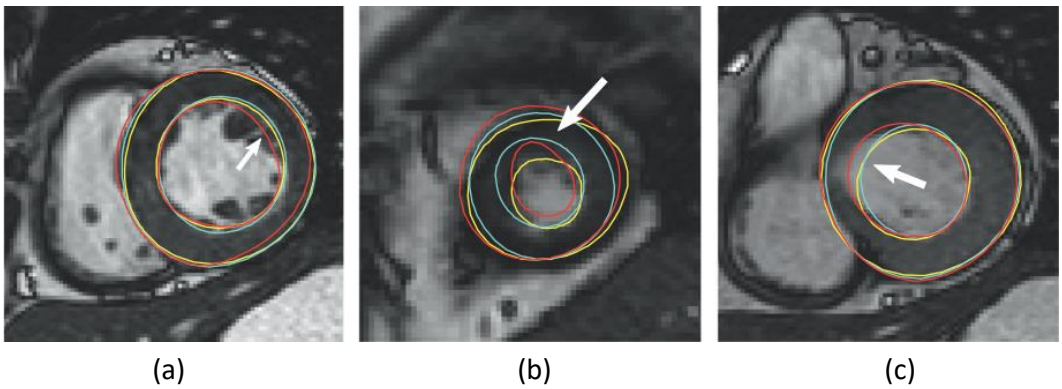


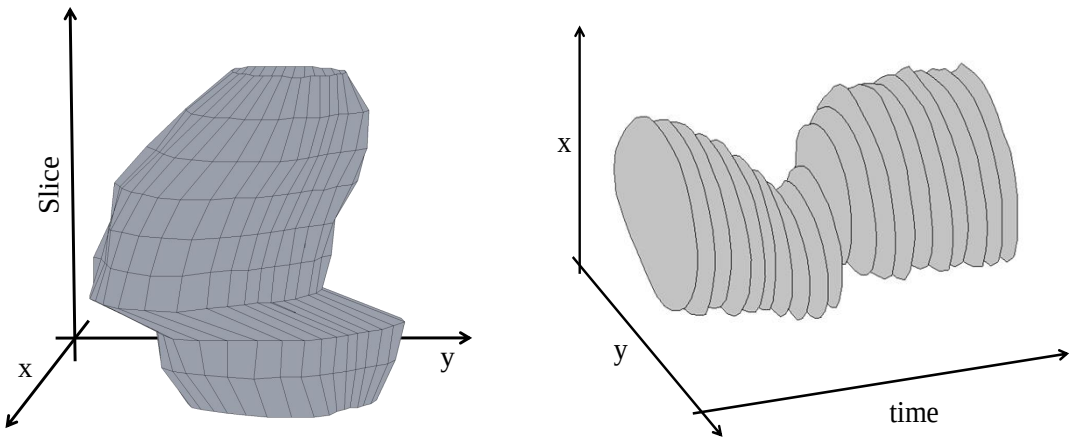
Figure 2.7: Examples of disagreement in the delineation of the endo and epicardium by experts in: (a) a mid slice, (b) an apical slice, and (c) a basal slice. Extracted from [167].

organ. Consequently, throughout the cardiac cycle, the location and size of the LV changes significantly. Fig. 2.8 (a) shows an example of the segmentation of the LV in one slice throughout time, in which this issue is very clear.

These three main difficulties are addressed throughout the course of this thesis in Chapters 3, 4 and 5, respectively.

2.6.2 State of the Art

Over the past decades, several methods for segmenting the LV in CMR have been proposed [138, 170]. These methods use a wide range of approaches, which may be cat-



(a) (b)

Figure 2.8: Challenging features of CMR: (a) slice misalignment, which cause translational artifacts in the position of the LV along the slices of an MR volume, and (b) cardiac motion, showing how the position of the LV in a specific slice significantly changes throughout time.

egorized in one of the following: image-driven [35, 41, 68, 69, 96, 103]; dynamic programming [89,137,175]; deformable models [15,59,61,87,93,101,107,160,166,185]; biomechanical models [24, 30, 165]; shape/appearance models [11, 94, 118, 119, 130, 131, 139, 141, 164, 177, 188, 190]; machine learning [14, 106, 109, 128]; and atlas/registration [16, 49, 98, 115].

Besides the type of approach used, it is also possible to categorize the proposed methods based on what type of data is being analyzed, in which case the alternatives are: 2D approaches, in which each MR slice is segmented independently; 2D+t approaches, which segment a sequence of MR slices using temporal information; 3D approaches, where the purpose is to extract the location of the LV surface in each MR volume; and 3D+t approach, which perform the segmentation using 3D and temporal information.

Table 2.1 shows a small collection of the works in the literature and where they are located in this categorization scheme. Note that blank entries do not necessarily mean that that type of approach does not exist. However, the distribution of works in the table gives an idea of what are the most popular approaches.

The following sections briefly explains how each type of approach addresses the segmentation of the LV in CMR and its challenges.

2.6.3 Image-Driven

Image-driven approaches comprehend a large spectrum of image processing techniques, namely, thresholding, edge detection, region growing, watershed, and/or morphological operations. They rely solely on image analysis and do not require any pre-training or learning procedure.

The goal of this type of approaches is to determine which pixels are located inside the

Table 2.1: Popular approaches of LV segmentation algorithms.

	2D	**2D+t**	**3D**	**3D+t**
Image-driven	[68, 69, 103]		[35, 41, 96]	
Dynamic programming	[89, 137]		[175]	
Deformable models	[61, 93, 185]	[15, 160, 166]	[59, 87, 101]	[107, 165] [24, 30]
Shape/appearance models	[119]	[139, 141, 164]	[118, 130] [94, 177]	[188, 190] [11, 131]
Machine learning	[14, 106, 128]			[109]
Atlas/registration			[16, 115]	[49, 98]

LV chamber, and which belong to other structures or to the background [35,41,69,96,103]. To achieve this, unsupervised classification algorithms are often used, *e.g.*, Ostu's method [135] or k-means, which transform the original intensity image into a binary (or labeled) image. After this step, the segmented region usually requires additional processing, based on morphological (*e.g.*, open/close) or convex hull operations, to obtain a reasonable shape. Another common pipeline is to perform an edge detection algorithm, and them extract the LV border through region growing or watershed techniques [35, 68].

The main advantage of these methods is their low computational complexity. However, they rely on strong intensity differences between the blood pool and the myocardium to determine the location of the LV, which, as mentioned in Section 2.6.1, is not always the case.

2.6.4 Dynamic programming

Dynamic programming (DP) approaches are closely related with image-driven approaches. The difference between them lies in the fact the DP is used to extract a curve corresponding to the LV border, rather than to labeling the pixels explicitly.

The typical approach is to convert the image into a cost matrix, in which low cost are assigned to pixels along the LV border. Different approaches may be used to compute the cost matrix, such as through thresholding [89,137] or edge detection. It is also common to convert the cost matrix to polar coordinates, where the LV border is assumed to be approximately a straight line.

DP has also been used to impose temporal constraints on the parameters of a shape model [175]. In this case, a higher dimensional DP problem is solved to determine the

optimal parameters along the time frames.

Although this class of approaches offers optimal solutions to the estimation of the LV contour(s), its performance greatly depends on the ability of the cost matrix to correctly assign low weights to the desired path.

2.6.5 Deformable Models

Deformable models have been popularized by two distinct approaches: active contours [85] and level-sets [108], both of which were able to increase the robustness of the segmentations while keeping the computational complexity low. The goal of this type of approaches is to model the border of the LV as a deformable structure (either a curve, a surface, or a volumetric object). This deformable structure is governed by a data term, which attracts it to specific image features associated to the location of the LV border, and a regularization term that imposes shape constraints. Although many of these approaches impose some shape constraints, these are typically smoothness-related or hand designed [11]. This means that the model may not be able to accurately represent the LV border in some cases, but it also means that it does not depend on having a representative training dataset.

This type of approaches is attractive because their formulation is easily adaptable to include different forms of prior shape or intensity information [185], as well as to impose smooth changes throughout the cardiac cycle [160]. Biomechanical models, for instance, are a subclass of deformable models, in which the constraints imposed are based on physics that try to mimic the mechanical properties of the heart [24,30,165]. The main advantage here is that a model of the LV dynamics may be used to ensure consistent segmentations across the temporal dimension. Despite these advantages, these approaches are sensitive to the initialization, because their accuracy typically depends on the initial proximity of the LV border, and to image conditions, since imaging artifacts can make the segmentation extremely difficult without more information [11,125].

2.6.6 Shape/Appearance Models

The drawbacks associated with deformable models contributed to the popularity of shape models, in which training sets of annotated data are used to learn the expected shape of the LV. A classical method developed using this approach is the Active Shape Model (ASM) [37]. The main idea behind ASM is to describe the shape of LV through a discretized contour or surface and the corresponding statistics, *i.e.*, the mean shape and the main modes of deformation [11, 66, 94, 130, 131, 177]. This allows the shape of the segmented region to be constrained such that it is always similar to those found in the training set.

ASMs have also been used to incorporate motion information about the LV [11,131,

188, 190]. In this case, the idea is to use statistical information about the motion of the LV throughout the cardiac cycle to constrain the segmentations of the MR sequence. This makes the segmentation consistent across different frames and helps guide the model in frames that are harder to segment.

Shape models were latter extended to appearance models, namely using Active Appearance Models (AAM) [38]. Contrary to shape models, which are typically edge-driven algorithms, AAMs also include statistical information about the expected appearance (intensity) of the LV. Therefore, AAMs provide a framework that increase the robustness of shape models to image conditions [11, 118, 119].

The disadvantage of this type of approach is that it relies on having a representative training set. Unseen shapes and appearances are deemed as highly unlikely, and thus are not accurately segmented with these methods. This may be prejudicial in the segmentation of CMR data from diseased patients that are not represented in the training set.

2.6.7 Machine Learning

The goal of machine learning approaches is to classify the pixels in the MR images either as inside or outside the LV border. Contrary to more simple image-driven approaches, they rely on a pre-training set in which a classifier is trained on previously labeled data.

Besides the lack of large dataset of labeled data, machine learning approaches suffer from the fact that the blood pool inside the LV is not very texture-rich and it does not have any particular intensity properties. Thus, from that point of view, is not easily distinguishable from other anatomical structures in the image. Furthermore, pixel classification alone cannot prevent unexpected shapes of the segmented regions, which is why some methods rely on a post-processing algorithm based on deformable/shape models [14, 128].

A common trait of these methods is that they depend on preprocessing steps, for intensity normalization and preliminary ROI selection [14, 106, 109, 128]. This is a useful strategy to prevent false positive classifications to arise from other neighboring structures.

2.6.8 Atlas/Registration

Atlas-based approaches rely on previously learned CMR atlases to extract the segmentation of the LV. Atlases are typically generated from one manually labeled volume or by combining information from multiple labeled volumes [16, 138]. Labeled volumes often include not only the location of the LV, but also of the remaining cardiac structures (*e.g.*, RV and myocardium).

New volumes are segmented by aligning them with the atlas(es), through a non-rigid registration process [49, 98]. The registration includes a global affine transformation, and local deformations, *e.g.*, using the free-form deformation model [163]. These transforma-

tions are estimated by maximizing the similarity between the two volumes, which can be achieved using different metrics, such as intensity likelihood [16] or normalized mutual information [98]. It has also been shown that performing patch-based registration may be advantageous to allow refining local deformations [16].

Although this type of approaches has the advantage of providing the segmentations of multiple cardiac structures simultaneously, it comes at the expense of an increase computational cost [170].

2.7 Conclusions

This chapter provides an overview of the problem addressed in this thesis and the main challenges that are faced. A brief description of the characteristics of CMR data, how it is acquired, and what is the medical procedure to analyze MR images and extract all the information required to assess the cardiac function and diagnose pathologies. This procedure involves the laborious task of manually delineating the border of the LV, whose automation is the main goal of this thesis. However, performing the automatic segmentation of the LV is not a trivial task. The main challenges include: i) identifying the LV border in each MR slice, which is not always simple due to the presence of other anatomical structure in its vicinity and to the lack of contrast in some of the images; ii) dealing with motion artifacts, caused by different levels of breath-hold during the acquisition process, which lead to the appearance of misaligned slices; and iii) taking into consideration that the LV considerably changes position throughout the cardiac cycle. These challenges are addressed throughout this thesis.

A review of the state of the art in LV segmentation algorithms is also presented. Several approaches have been used during the past decades, each with advantages and disadvantages. Interestingly, deformable models, shape/appearance models and atlas/registrations are the most popular approaches among those that use temporal information. This choice is related to the adaptable and extensible frameworks that these approaches use, which easily allow the inclusion of a prior information about the shape, appearance and dynamics of the LV.

3

2D Segmentation

Contents

The segmentation of the LV in CMR is popular topic in the literature [138]. It is a common approach to treat this problem in a 2D setting, in which the segmentation algorithm analyzes each slice of the MR sequence individually. This is advantageous because the amount of data being processed is much smaller and because it bypasses some of the issues related with the three-dimensional structure of the LV and its motion throughout the cardiac cycle. This chapter addresses the segmentation of the LV in the 2D setting.

3.1 Introduction

When analyzing CMR from a 2D perspective, it becomes clear that one of the most distinguishable features of the LV is its circular shape (see the examples shown in Fig. 3.1). This information about the shape of the LV is often used by segmentation algorithms to impose constraints on the shape of the segmented region and to guarantee reasonable estimates [138].

Statistical shape models (SSM) are one of the most well established approaches based on deformation models that use prior information about the shape of the segmentation [66]. The most popular SSM is the Active Shape Model (ASM) algorithm, proposed by Cootes et al. [37]. It consists of describing the contour of an object by a mean shape and its most significant modes of deformation, learned from training data. In this method, the contour of an object is characterized by the parameters of a (global) pose transformation and by deformation coefficients. This provides a simple and effective way of modeling the contour of the object, and preventing unpredictable or unexpected segmentation shapes.

Fitting the learned model to the boundary of an object in an image comprises two steps: 1) searching for observation points located on the object's boundary in the image, and 2) estimating the model parameters (pose and deformation parameters) based on those observations.

In the ASM method, the model parameters are estimated with the least squares method, under the assumption that all the observation points belong to the object boundary. However, in most real applications this is not true and many of the observations are often outliers. For instance, in CMR, the presence of other anatomical structures in the image, *e.g.*, the papillary muscles (recall Section 2.6), lead to the detection of these outliers. In these cases, the estimates obtained with least squares method correspond to poor segmentations [65, 143].

One approach that has been used to overcome this limitation of the standard ASM is to improve the boundary detection method [12,36,38,43,179,184]. By reducing the number of outliers detected, these works are able to improve the accuracy of the segmentations. Another approach is to use an estimation method that is able to deal with the presence of outliers [7, 143].

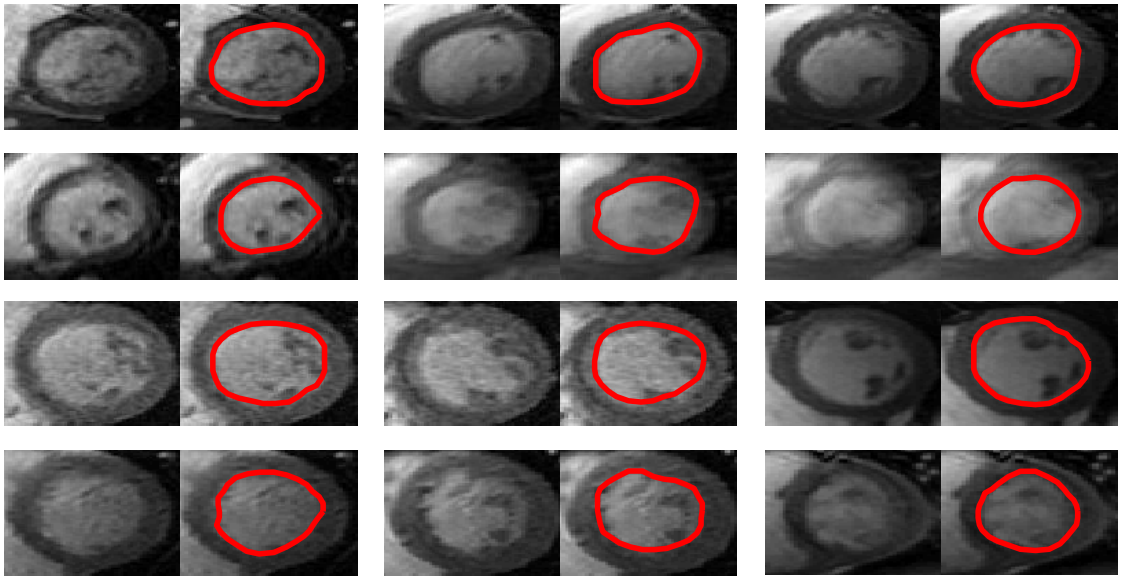


Figure 3.1: Examples of CMR images from different patients, slices and phases. Each pair shows a slice with and without the corresponding LV segmentation (red contour).

This chapter focuses on the second type of approach. The proposed approach is based on the assumption that a single 2D shape model is able to describe the LV contour of any image, regardless of which slice and frame it corresponds to. Then, a novel Bayesian framework is proposed for the estimation of the shape model parameters that allows for a reliable and robust segmentation in the presence of outliers. This new algorithm assumes that some of the observation points are outliers and takes this into account when estimating the parameters. The proposed approach was evaluated on synthetic images and in the segmentation of the LV in CMR. Furthermore, in order to show that the algorithm is more general than this particular problem, two other application were tested: the segmentation of the LV in ultrasound and the segmentation of the lip in frontal face images.

The remainder of this chapter is organized as follows. Section 3.2 provides a background about the methods discussed in the following sections, and presents an overview of the state of the art. Section 3.3 formalizes the problem, and the proposed framework is described in Sections 3.4-3.5. The experimental setup and statistical results are shown in Section 3.6-3.7, as well as a comparison with related works. Finally, Section 3.8 concludes the chapter.

3.2 Background

In the mid 80's, Active Contours (AC) (*e.g.*, Snakes) were very popular in computer vision problems. They consist of a curve that iteratively deforms until it fits a specific object in an image. These deformable curves follow a minimization scheme that includes two terms: an image-dependent term that attracts the curve towards the boundary of

the object of interest; and a model-dependent term that preserves the smoothness of the curve [113]. This approach leads to very good segmentation results and remains very appealing, mainly due to its flexibility and adaptability to different image segmentation problems. However, AC are extremely sensitive to nearby structures that do not belong to the object of interest. Consequently, they easily assume undesired shapes in order to fit all the surrounding structures. In this sense, a lot of effort has been made to distinguish between the true object boundaries and background structures [110] with successful results. Furthermore, in cases where parts of the object boundary are absent (gaps), AC are free to assume any smooth shape that fills those gaps, which can lead to unexpected shapes.

Alternatively, several works were proposed using more rigid models. For instance, Jain et al. [76] proposed using deformable templates to detect an object of interest in different images. Deformable templates are represented by an image with the expected object shape that is allowed to undergo similarity transformations (*i.e.*, translation, rotation and scaling), as well as local deformations. These deformations are obtained by applying a displacement field to the template image. When searching for the object of interest in a new image, Jain et al. use a Bayesian formulation to infer the best transformations that the template must undergo to fit the object. Since the object is expected to have a shape similar to the template, large deformations are penalized by receiving a smaller probability. This type of approach proved to be effective in the detection of rigid objects, but it is prone to fail in the medical images case due to the large variability of biological shapes [66].

3.2.1 Active Shape Model

In the 90's, Cootes et al. [37] proposed the use of ASMs to segment object with large variability. ASMs add specificity to the shape model by limiting the allowed type and extent of deformations, thus overcoming the problems of previous approaches.

The standard ASM methodology can be divided into three core components: (i) the shape representation; (ii) learning shape statistics; and (iii) using the learned shape model to segment new images. Each of these components will be addressed separately in the following subsections.

3.2.1.1 Shape Representation

ASMs rely on models that are described by a set of points. These points can either be located along the object boundary and on any internal or external features (*e.g.*, the center of curvature); points from adjacent objects can also be used to facilitate the fitting process (*e.g.*, using part of the right ventricle in a shape model of the LV in echocardiography [37]). The model points should be located in specific and application-dependent features of the image, such as anatomical landmarks in medical images. If persistent landmarks do not

exist in some applications, it is also possible to use semi-landmarks, which consist of points that do not need to match specific image features but are somehow consistent across different examples.

Formally, the goal of ASMs is to approximate the boundary of an object by a set of N points,

$$\widetilde{\boldsymbol{x}} = \begin{bmatrix} \widetilde{\boldsymbol{x}}^1 \\ \vdots \\ \widetilde{\boldsymbol{x}}^N \end{bmatrix}. \tag{3.1}$$

where $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{2N\times 1}$, and $\widetilde{\boldsymbol{x}}^i \in \mathbb{R}^2$ are the 2D coordinates of i-th point (see examples in Fig. 3.2). The location of these points is determined by applying a global (pose) transformation $\boldsymbol{T_\theta}$, with parameters $\boldsymbol{\theta} = \{\boldsymbol{a}, \boldsymbol{t}\}$, such that

$$\begin{aligned} \widetilde{\boldsymbol{x}}^i &= \boldsymbol{T_\theta}(\boldsymbol{x}^i) \\ &= \begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix} \begin{bmatrix} x_1^i \\ x_2^i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \\ &= \boldsymbol{A}\boldsymbol{x}^i + \boldsymbol{t}, \end{aligned} \tag{3.2}$$

where $\boldsymbol{x}^i$ is the position of the i-th model point in the shape space, which is defined by the average shape $\overline{\boldsymbol{x}} = (\overline{\boldsymbol{x}}^{1\,\top}, ..., \overline{\boldsymbol{x}}^{N\,\top})^\top \in \mathbb{R}^{2N\times 1}$, and a linear combination of the L main modes of deformation $\boldsymbol{D} \in \mathbb{R}^{2N\times L}$,

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{b}, \tag{3.3}$$

where each column of $\boldsymbol{D}$ corresponds to a specific deformation mode, and $\boldsymbol{b} \in \mathbb{R}^{L\times 1}$ are the deformation coefficients that determine the contribution of each deformation mode. Accordingly, the i-th model point in the shape space is given by

$$\boldsymbol{x}^i = \overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\boldsymbol{b}, \tag{3.4}$$

where $\boldsymbol{D}^i \in \mathbb{R}^{2\times L}$ is a matrix containing the lines of $\boldsymbol{D}$ associated to the i-th model point. Combining (3.2) with (3.4), the position of the i-th shape model point may be rewritten as

$$\widetilde{\boldsymbol{x}}^i = \boldsymbol{A}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\boldsymbol{b}) + \boldsymbol{t}. \tag{3.5}$$

The shape model parameters that define the contour of the object are: 1) $\boldsymbol{a} = (a_1, a_2)^\top$, which determines the scaling and rotation of the contour through matrix $\boldsymbol{A}$; 2) $\boldsymbol{t}$, which determines the translation of the contour; and 3) $\boldsymbol{b}$ which defines the contribution of each deformation mode.

3.2.1.2 Learning Shape Statistics

The goal here is to learn the shape statistics of the object. Since shape is a property of an object that does not change under similarity transformations (translation, rotation

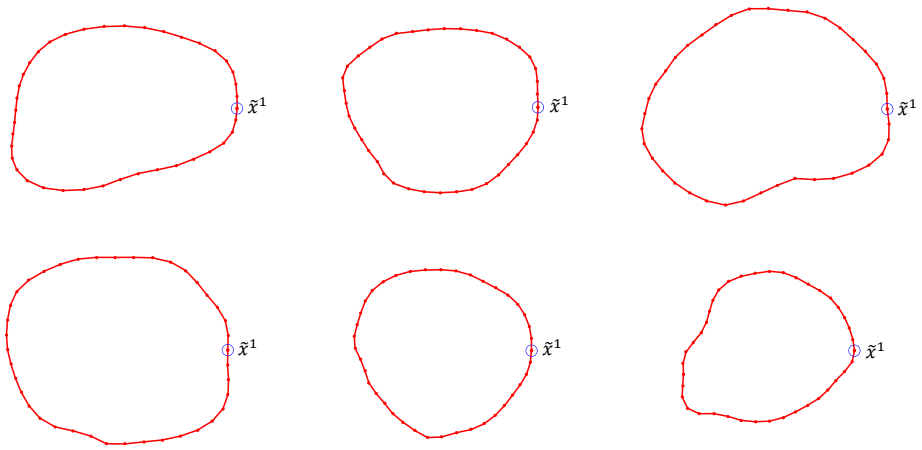


Figure 3.2: Shape examples used to learn the 2D shape model of the LV. The red dots mark the position of the contour points and the blue circle is the first model point.

and scaling) [66], a prerequisite for learning a statistical shape model is to remove these type of transformations between examples in the training set through alignment. In some applications, such as CMR analysis, one may also want to capture the variation in scale (changes in size of the LV during the cardiac cycle) as a shape feature, in which case the alignment transformation would have to disregard scaling. The learning procedure of ASMs can be divided in two steps:

1. Aligning the shape examples in the training set; and
2. Computing the mean shape and main modes of deformation.

Shape Alignment The alignment process aims to guarantee that corresponding points in each example are as close to each other as possible. Let $\widetilde{\boldsymbol{x}}_{\text{ref}}$ be the reference shape in a training set to which all the others are aligned (which could be either a randomly selected example of the average of all examples). The alignment of a specific example, $\widetilde{\boldsymbol{x}}$, and the reference shape, $\widetilde{\boldsymbol{x}}_{\text{ref}}$, is achieved by determining the global transformation parameters, $\boldsymbol{a}$ and $\boldsymbol{t}$, that minimize the sum of squared errors between the two shapes,

$$\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{t}} = \arg\min_{\boldsymbol{a},\boldsymbol{t}} (\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}_{\text{ref}})^\top (\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}_{\text{ref}}). \tag{3.6}$$

The solution to this optimization problem is the following [37]

$$\begin{bmatrix} \widehat{\boldsymbol{a}} \\ \widehat{\boldsymbol{t}} \end{bmatrix} = \left(\sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{X}^{i\top} \boldsymbol{X}^i & \boldsymbol{X}^{i\top} \\ \boldsymbol{X}^i & \boldsymbol{I} \end{bmatrix} \right)^{-1} \left(\sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{X}^{i\top} \widetilde{\boldsymbol{x}}^i_{\text{ref}} \\ \widetilde{\boldsymbol{x}}^i_{\text{ref}} \end{bmatrix} \right), \tag{3.7}$$

where

$$\boldsymbol{X}^i = \begin{bmatrix} x^i_1 & -x^i_2 \\ x^i_2 & x^i_1 \end{bmatrix}, \qquad \begin{bmatrix} x^i_1 \\ x^i_2 \end{bmatrix} = \widetilde{\boldsymbol{x}}^i. \tag{3.8}$$

Typically, to avoid biasing the shape model with the choice of reference shape, after transforming all the examples in the training set, the reference shape is updated to be the (new) mean shape, $\widetilde{\boldsymbol{x}}_{\text{ref}} = \overline{\boldsymbol{x}}$, and the alignment process is repeated until it converges.

Mean Shape and Deformation Modes Once the shapes in the training set are aligned, it is possible to analyze the statistics of shape variability. Cootes et al. introduced the notion of Point Model Distribution as a way to model the variation in the position of each landmark, noting that these variations are not uncorrelated from each other. Instead of analyzing each point independently, they consider each shape as a single point in a $2N$ dimensional space (the vector $\widetilde{\boldsymbol{x}}$). They determine the so-called "allowable shape domain" as the region inside which training shape points exist. Within this region, it is possible to create new shapes that will be similar to the ones in the training set.

Assuming that these points have an ellipsoidal distribution in the $2N$-D space, they determine the principal axes of the ellipsoid, which are related to the orthogonal modes of shape variation. This is achieved by performing Principal Component Analysis (PCA), which works as follows. Let $\widetilde{\boldsymbol{x}}_j$ be the j-th (aligned) example in a training set of M examples. First, the mean vector and a $2N \times 2N$ covariance matrix are computed

$$\overline{\boldsymbol{x}} = \frac{1}{M} \sum_{j=1}^{M} \widetilde{\boldsymbol{x}}_j \tag{3.9}$$

$$\mathbf{S} = \frac{1}{M} \sum_{j=1}^{M} (\widetilde{\boldsymbol{x}}_j - \overline{\boldsymbol{x}})(\widetilde{\boldsymbol{x}}_j - \overline{\boldsymbol{x}})^T. \tag{3.10}$$

Then, eigenvectors of $\mathbf{S}$ and the corresponding eigenvalues are determined

$$\mathbf{S}\boldsymbol{d}_l = \lambda_l \boldsymbol{d}_l, \quad l = 1, \ldots, 2N \tag{3.11}$$

where $\boldsymbol{d}_l$ and λ_l correspond to the l-th eigenvector and eigenvalue, respectively. Each eigenvector, $\boldsymbol{d}_l \in \mathbb{R}^{2N}$, can be viewed as a displacement vector related to a specific mode of deformation, and λ_l is the corresponding variance on the training set. The eigenvectors associated to the largest eigenvalues correspond to the most significant modes of variation. Since most of the variation can be explained by just a subset of the eigenvectors, the ones associated to smaller eigenvalues can be discarded to reduce the dimensionality of the representation. This is typically done by using only the first L eigenvectors such that these account for a large proportion of the total variation

$$L : \sum_{l=1}^{L} \lambda_l > r \sum_{k=1}^{2N} \lambda_k. \tag{3.12}$$

Typical values for the proportion parameter are $r \in [0.9, 0.98]$.

Using the learned mean shape and the main modes of variation it is possible to analyze new images, using the procedure described in the next section.

3.2.1.3 Fitting Algorithm

The final component of ASMs is to analyze unseen images, by identifying the location of the object of interest and determining the shape model parameters that make the model fit the object. Since it is unfeasible to do this in a global and single step, the most common approach is to iteratively adapt the model towards the desired position.

The ASM approach assumes that the object boundary is identified by an intensity transition in the image (an edge). Given an initial guess of the model position, edge points are extracted from search lines orthogonal to the contour at each model point. This leads to a set of observations, denoted by $\boldsymbol{y} = \left(\boldsymbol{y}^{1\top}, \ldots, \boldsymbol{y}^{N\top}\right)^\top$, where $\boldsymbol{y}^i$ corresponds to the desired position of the i-th model point.

Given the set of observations, the position of the shape model is updated by computing

$$\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{t}}, \widehat{\boldsymbol{b}} = \arg\min_{\boldsymbol{a},\boldsymbol{t},\boldsymbol{b}} (\widetilde{\boldsymbol{x}} - \boldsymbol{y})^\top (\widetilde{\boldsymbol{x}} - \boldsymbol{y}). \tag{3.13}$$

Rewriting this equation using (3.5) leads to

$$\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{t}}, \widehat{\boldsymbol{b}} = \arg\min_{\boldsymbol{a},\boldsymbol{t},\boldsymbol{b}} \sum_{i=1}^{N} \left\| \boldsymbol{A}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i \boldsymbol{b}) + \boldsymbol{t} - \boldsymbol{y}^i \right\|^2. \tag{3.14}$$

This is a non-convex optimization problem that is solved in [37] by alternating between: (i) the estimation of the transformation parameters, $\boldsymbol{a}$ and $\boldsymbol{t}$, assuming $\boldsymbol{b}$ fixed; and (ii) the estimation of the deformation coefficients, $\boldsymbol{b}$, assuming $\boldsymbol{a}$ and $\boldsymbol{t}$ fixed. This process is described next.

Update of the Transformaton Parameters For each iteration t of the fitting procedure, the shape model parameters $\boldsymbol{a}$ and $\boldsymbol{t}$ are updated by aligning the shapes $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{y}$ using a procedure similar to the alignment of the training examples described in Section 3.2.1.2. Let $\boldsymbol{a}_t, \boldsymbol{t}_t, \boldsymbol{b}_t$ be the model parameters at iteration t. Assuming $\boldsymbol{b} = \boldsymbol{b}_{(t-1)}$ fixed, the deformed model is given by $\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{b}_{(t-1)}$, and the transformation parameters are update by solving

$$\begin{aligned} \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{t}} &= \arg\min_{\boldsymbol{a},\boldsymbol{t}} \sum_{i=1}^{N} \left\| \boldsymbol{A}\boldsymbol{x}^i + \boldsymbol{t} - \boldsymbol{y}^i \right\|^2 \\ &= \arg\min_{\boldsymbol{a},\boldsymbol{t}} \sum_{i=1}^{N} \left\| \boldsymbol{X}^i \boldsymbol{a} + \boldsymbol{t} - \boldsymbol{y}^i \right\|^2, \end{aligned} \tag{3.15}$$

where $\boldsymbol{X}^i$ is given by (3.8). The solution of (3.15) is obtained by computing the derivative of the objective function and equating to zero, which leads to an equation similar to (3.7) (where $\widetilde{\boldsymbol{x}}_{\text{ref}}$ is now given by $\boldsymbol{y}$).

Update of the Deformation Coefficients The estimation of the deformation coefficients is achieved by assuming $\boldsymbol{a}$ and $\boldsymbol{t}$ to be known and fixed in (3.14), leading to

$$\widehat{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}} \sum_{i=1}^{N} \left\| \boldsymbol{A}_{(t)}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\boldsymbol{b}) + \boldsymbol{t}_{(t)} - \boldsymbol{y}^i \right\|^2. \tag{3.16}$$

As previously, this new optimization problem is linear and its solution is straightforward to compute

$$\widehat{\boldsymbol{b}}_{(t+1)} = \left(\sum_{i=1}^{N} {\boldsymbol{D}^i}^\top \boldsymbol{A}_{(t+1)}^\top \boldsymbol{A}_{(t+1)} \boldsymbol{D}^i \right)^{-1} \left(\sum_{i=1}^{N} {\boldsymbol{D}^i}^\top \boldsymbol{A}_{(t+1)}^\top \left(\boldsymbol{y}^i - \boldsymbol{A}_{(t+1)}\overline{\boldsymbol{x}}^i - \boldsymbol{t}_{(t+1)} \right) \right). \tag{3.17}$$

Since the deformation parameters obtained by this expression may correspond to an unexpected shape, an additional step is required. In order to constrain the solution given by (3.17), the deformation coefficients are shrinked, as needed, to values within the "expected". For this purpose, the Mahalanobis distance, d, is used as a way to measure what is acceptable. Formally, d has to be lower than a specific threshold, $d_{\max}$,

$$d^2 = \sum_{l=1}^{L} \frac{\widehat{b}_l^2}{\lambda_l} \leq d_{\max}^2. \tag{3.18}$$

where $\widehat{b}_l$ denotes the l-th component of $\widehat{\boldsymbol{b}}_{(t+1)}$, and λ_l is the eigenvalue associated to the l-th deformation mode. The threshold is chosen so that most of the shapes in the training set satisfy (3.18) (a typical value is $d_{\max} = 3$). If $\widehat{\boldsymbol{b}}_{(t+1)}$ does not satisfy (3.18), it is rescaled to the closest acceptable shape by

$$\widehat{\boldsymbol{b}}_{(t+1)} \leftarrow \widehat{\boldsymbol{b}}_{(t+1)} \frac{d_{\max}}{d} \quad \text{if} \quad d > d_{\max}. \tag{3.19}$$

Since in each of these steps the distance between the shape model and the observations is being minimized, this procedures eventually converges to a specific solution. However, there is no guarantee that this is the global optimal solution.

3.2.2 Related Work

The standard ASM, described above, was used in many works [20,37,39,63,117,179]. However, it estimates the shape model parameters under the assumption that all the observation points belong to the object boundary. In most applications, this is not true and, consequently, the segmentations obtained in the presence of noisy observations are often poor. This drawback of the ASM method caused subsequent works to propose alternative ways of estimating the shape model parameters [7, 36, 94, 143, 183].

Most works consider multiple candidates for the location of each model point. This makes the algorithm less dependent on the accuracy of the boundary detection method and

increases the probability that the true object boundary is among the detected observation points. Then, the estimation of the model parameters is accomplished by simultaneously selecting the subset of observation points that maximizes a specific objective function, which typically promotes: i) segmentations along points with specific image features, *e.g.*, edge points (located along an image edge), and ii) segmentations with shapes similar to those observed in the training set.

Different approaches have been used to accomplish this. Wang and Staib [183], for instance, use the gradient descent method to obtain the model parameters that maximize the posterior probability, which includes a prior about the parameters learned from a training set and the likelihood of each pixel location. Similarly, Cootes et al. [36] proposes an algorithm for maximizing the overall quality of fit between the shape model and the image, based on a random forest regressor that evaluates the quality of fit of the image pixels around each model point. However, these two methods do not explicitly account for the presence of outliers. Instead, they rely on the objective function (posterior probability/overall quality of fit) to guide the model toward the true object boundary.

Few works have proposed to deal with outliers explicitly. For instance, the approach used in [7] is based on the Robust Point Matching (RPM) algorithm [34]. The RPM algorithm pairs observation points with model points, and the points that are left unpaired are considered outliers and disregarded in the estimation of the parameters. In order to determine if points should be paired, they compute a matrix of correspondence degrees, in which each entry measures the likelihood of a correspondence between a pair of points. This matrix is then used in the estimation of the shape model parameters by minimizing the distance between all pairs of points, weighted by the corresponding confidence degree. They use a deterministic annealing term to control the fuzziness of the matrix of correspondences: in the beginning, they allow the matrix to be more fuzzy, and as the algorithm iterates, the force the matrix to become closer to a binary correspondence matrix. Rogers et al. [143], on the other hand, tests two types of approaches: 1) M-estimators, which weight the observation points based on some criteria, and 2) random sampling strategies (*e.g.*, the Random Sample Consensus (RANSAC) algorithm [51]) to determine the best subset of observation points to be used in the estimation of the shape model parameters. They conclude from their results that sampling strategies lead to better results than M-estimators, although both significantly outperform the standard approach. Lekadir et al. [94] proposes to determine which observation points are outliers using a local shape dissimilarity metric. This metric is based on the distance between pairs of model points. Observation points that do not respect the local shape of the object are treated as outliers. A disadvantage of this approach is that only one observation point is allowed for each model point.

In this chapter, a new method for estimating the model parameters is proposed using a Bayesian approach that assigns each observation a different weight based on the

probability of that observation belonging to the object boundary. The main difference between [143] and the proposed method is the following: in [143], the weights are obtained using an heuristic strategy (see equation (10) in [143]) designed by Huber [70]; here, these weights are automatically determined in a principled and well founded probabilistic way. Since outliers often receive low probabilities, their influence in the parameter estimation is reduced, making the method robust in the presence of outliers. A similar approach has been previously used in [123] for the estimation of a Snake model [85]. However, a drawback in [123] is that the Snake model can assume unexpected shapes, since is does not use prior information about the object shape. The following sections describe the problem and the proposed methodology.

3.3 Problem Formulation

The boundary of an object of interest is described using a shape model, $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{2N}$, where the position of the i-th model point is given by

$$\begin{aligned} \widetilde{\boldsymbol{x}}^i &= \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^i) \\ &= \boldsymbol{A}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\boldsymbol{b}) + \boldsymbol{t}. \end{aligned} \tag{3.20}$$

Given an image I, the segmentation of the object is obtained by estimating the model parameters, $\{\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}\}$.

Three formulations, illustrated in Fig. 3.3, were tested in this thesis. The first formulation is the closest to the original ASM. Given the image I, the idea is that each model point searches in its vicinity for observations (edge points) that are candidate positions for the object border. Many edge points may be assigned to the same model point, and the main assumption is that each observation point may either be a valid point (located on the object border) or an outlier. In the second formulation, the notion of observation is refined to edge segments composed of several edge points. In this case, each segment may either be valid or outlier, and the edge points in a specific segment are not considered independent observations. The last formulation explores the concept of data association [18] as a way to determine which observations are outliers or not. The idea of this approach is that the border of the object is given by an unknown subset (interpretation) of the detected edge segments. In other words, there is a combination of binary labels that represents the object border, in which the label one is assigned to valid segments and zero to outliers. In this case, there is a specific interpretation that represents the object border more accurately, while the other interpretations may be missing valid segments or including outliers. The tests showed that edge points achieved better results than the other two. Therefore, this chapter focuses on this particular approach. The formulations using edge segments and interpretations are described in Appendices A and B, respectively.

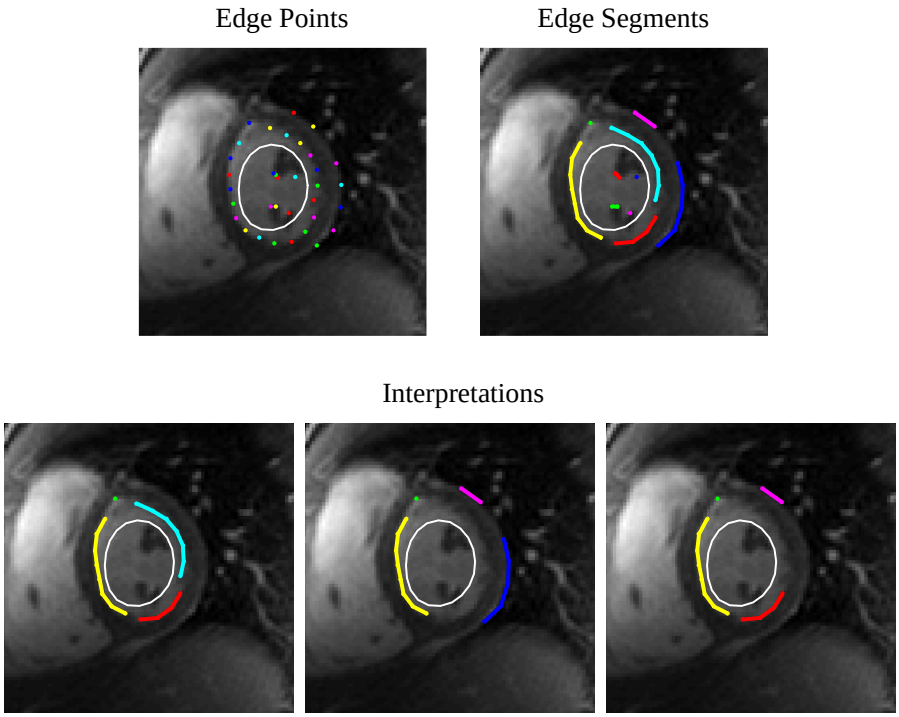


Figure 3.3: Different formulations from which the shape model parameters are estimated. **Edge points/segments**: each detected point/segment may either belong to the LV border or be an outlier. **Interpretations**: each interpretation is a subset of the detected edge segments, and of all the interpretations, only one represents the LV border accurately, while the remaining ones have outliers/are missing valid segments.

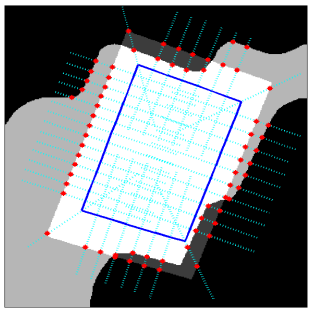


Figure 3.4: Example of the detection of observation points. The blue line corresponds to the model; the dashed cyan lines correspond to the search lines; and the red dots correspond to the detected observation points.

Assuming an initial guess of the model position is given, a set of observation points is extracted from the image by searching in the vicinity of each model point. This can be done using many approaches, *e.g.*, by searching along lines orthogonal to the contour at each model point, as shown in Fig. 3.4, or by searching within a region around each model point, as in [36]. Regardless of the method used to extract these observations, the result is that, for each model point, $\widetilde{\boldsymbol{x}}^i$, a set of observation points, $\boldsymbol{Y}^i = \{\boldsymbol{y}^{ij}, j = 1, \ldots, M^i\}$,

is obtained. In general, $M^i \neq M^j$ for $i \neq j$. An important remark is that many of the observations contained in the set $\boldsymbol{Y}^i$ do not belong to the object boundary and should be considered as outliers. Unfortunately, it is not possible to know, beforehand, which observation points are valid (truly belonging to the object boundary) or invalid (outliers). To tackle this issue, two sensor-observation models considered, and a binary label, $k^{ij} \in \{0, 1\}$, is assigned to each observation point, where $k^{ij} = 1$, if $\boldsymbol{y}^{ij}$ is considered valid, and $k^{ij} = 0$ otherwise. The two models occur with unknown probabilities, $p_0 = p(k^{ij}{=}0)$ and $p_1 = p(k^{ij}{=}1)$, with $p_0 + p_1 = 1$.

The model described so far comprises three sets of parameters: 1) global transformation parameters $(\boldsymbol{a}, \boldsymbol{t})$, where $\boldsymbol{a} = (a_1, a_2)$ and $\boldsymbol{t} = (t_1, t_2)$, 2) local deformation parameters $\boldsymbol{b}$, and 3) the label probabilities $\boldsymbol{p} = (p_0, p_1)$. In the following, the set of all the model parameters will be denoted by $\boldsymbol{\Theta} = (\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}, \boldsymbol{p})$.

Let the j-th observation point detected in the vicinity of the i-th model point be denoted as $\boldsymbol{y}^{ij} = \left(y_1^{ij}, y_2^{ij}\right)^\top$, and let k^{ij} be the corresponding model label assigned to $\boldsymbol{y}^{ij}$. Assuming that observation $\boldsymbol{y}^{ij}$ is valid ($k^{ij} = 1$) the following sensor model is considered

$$\boldsymbol{y}^{ij} = \widetilde{\boldsymbol{x}}^i + \mathbf{v}^i, \tag{3.21}$$

where $\mathbf{v}^i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}^i)$ is a zero mean white Gaussian noise with diagonal covariance matrix $\boldsymbol{\Sigma}^i$. This covariance matrix may either be estimated from training data, or defined by the user, allowing the uncertainty of the model to be tuned. In this chapter, the latter approach was chosen.

Combining (3.20) and (3.21) yields

$$\begin{aligned} \boldsymbol{y}^{ij} &= \boldsymbol{T}(\boldsymbol{x}^i) + \mathbf{v}^i \\ &= \boldsymbol{A}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\boldsymbol{b}) + \boldsymbol{t} + \mathbf{v}^i. \end{aligned} \tag{3.22}$$

It follows that the probability of $\boldsymbol{y}^{ij}$ being a valid observation is given by

$$p\left(\boldsymbol{y}^{ij}\middle|k^{ij}{=}1\right) = \mathcal{N}\left(\boldsymbol{y}^{ij}; \boldsymbol{A}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\boldsymbol{b}) + \boldsymbol{t}, \boldsymbol{\Sigma}^i\right), \tag{3.23}$$

where $\mathcal{N}(\cdot; \mu, \boldsymbol{\Sigma})$ denotes a normal distribution with mean μ and covariance matrix $\boldsymbol{\Sigma}$. On the other hand, it will be assumed that if $\boldsymbol{y}^{ij}$ has the label $k^{ij} = 0$ (*i.e.,* outlier), then it follows a uniform distribution within a region $V_{\widetilde{\boldsymbol{x}}^i}$ (validation gate) in the vicinity of $\widetilde{\boldsymbol{x}}^i$,

$$p\left(\boldsymbol{y}^{ij}\middle|k^{ij}{=}0\right) = \mathcal{U}(V_{\widetilde{\boldsymbol{x}}^i}). \tag{3.24}$$

The interpretation of these two models is that valid observations are more likely to appear near the shape model, whereas outliers are equally likely to appear anywhere within a specific region.

3.4 Expectation Maximization Framework

Given the probabilistic model described above, the goal is to estimate the model parameters (global transformation $(\boldsymbol{a}, \boldsymbol{t})$ and deformation coefficients $\boldsymbol{b}$) that fit the shape model to the observation points, $\boldsymbol{Y} = \{\boldsymbol{y}^{ij}\}$ extracted from the image. Ideally, the binary labels $\boldsymbol{K} = \{k^{ij}\}$ associated to each observation points $\boldsymbol{y}^{ij}$ would be known. However, this information is not available, *i.e.*, $\boldsymbol{K}$ is an unobserved (hidden) set of labels. The class-conditional generative model is thus obtained by marginalizing with respect to the missing label sequence, as follows

$$p(\boldsymbol{Y}|\boldsymbol{\Theta}) = \sum_{\boldsymbol{K}\in\{0,1\}^M} p\left(\boldsymbol{Y}, \boldsymbol{K}|\boldsymbol{\Theta}\right) \tag{3.25}$$

where $M = \sum_{i=1}^{N} M^i$ is the (often large) total number of detected observations.

The estimation of the model parameters by maximization of (3.25) would require taking into account all possible combinations of $\boldsymbol{K}$, which is infeasible. In fact, it is not possible to obtain a closed form expression nor to analytically optimize it. This difficulty is circumvented by searching for a sub-optimal solution using the Expectation-Maximization (EM) method [45]. The EM method iteratively updates the transformation and deformation parameters by maximizing an auxiliary function, $Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right)$ where $\widehat{\boldsymbol{\Theta}}_{(t)}$ is the most recent estimate of the unknown parameters. In this work, two approaches will be described: 1) maximizing the expectation of the likelihood function, which leads to the maximum likelihood (ML) estimate; and 2) maximizing the expectation of the joint probability function, which leads to the maximum *a posteriori* (MAP) estimate. These two approaches are detailed next.

3.4.1 Maximum Likelihood Estimation

In the ML framework, the complete log-likelihood of a set of observation points $\boldsymbol{Y}$ and labels $\boldsymbol{K}$ is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) &= \log p\left(\boldsymbol{Y}, \boldsymbol{K}|\boldsymbol{\Theta}\right) \\ &= \log\left(p\left(\boldsymbol{Y}|\boldsymbol{K}, \boldsymbol{\Theta}\right) p\left(\boldsymbol{K}\right)\right).\end{aligned} \tag{3.26}$$

Assuming conditional independence between observations, the previous equation can be factorized as follows

$$\begin{aligned}\mathcal{L}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) &= \log\left(\prod_{i=1}^{N}\prod_{j=1}^{M^i} p\left(\boldsymbol{y}^{ij}\middle|k^{ij}, \boldsymbol{\Theta}\right) p\left(k^{ij}\right)\right) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M^i} \log p\left(\boldsymbol{y}^{ij}\middle|k^{ij}, \boldsymbol{\Theta}\right) + \log p\left(k^{ij}\right),\end{aligned} \tag{3.27}$$

where $p\left(k^{ij}\right)$ is the probability of the label k^{ij} (*i.e.*, either p_1 or p_0), N is the number of model points, and M^i is the number of observation points associated with the i-th model point.

The EM algorithm comprises the following two steps [45]: 1) the *E-step*, in which the expectation of the log-likelihood is updated by computing the probability of each observation, based on a previous estimate of the parameters; and 2) the *M-step*, in which the parameters are updated by maximizing the expectation obtained in the E-step. These steps are detailed in the following subsections.

3.4.1.1 E-step

Let $\widehat{\boldsymbol{\Theta}}_{(t)} = (\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{t}}, \widehat{\boldsymbol{b}}, \widehat{\boldsymbol{p}})_{(t)}$ be the estimates of the model parameters at iteration t. Given the set of observations, $\boldsymbol{Y}$, and the most recent estimates, we can define the auxiliary function $Q_{\mathrm{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right)$ as the expected value of the log-likelihood function

$$
\begin{aligned}
Q_{\mathrm{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) &= \mathbb{E}_{\boldsymbol{K}}\left[\mathcal{L}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) \middle| \boldsymbol{Y}, \widehat{\boldsymbol{\Theta}}_{(t)}\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M^i} \mathbb{E}_{\boldsymbol{K}}\left[\log p(\boldsymbol{y}^{ij} \mid k^{ij}, \boldsymbol{\Theta}) + \log p(k^{ij}) \middle| \boldsymbol{Y}, \widehat{\boldsymbol{\Theta}}_{(t)}\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M^i}\sum_{l=0}^{1} p(k^{ij}{=}l|\boldsymbol{y}^{ij}, \widehat{\boldsymbol{\Theta}}_{(t)}) \left[\log p(\boldsymbol{y}^{ij} \mid k^{ij}{=}l, \boldsymbol{\Theta}) + \log p(k^{ij}{=}l)\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_0^{ij}\left(\log p\left(\boldsymbol{y}^{ij} \middle| k^{ij}{=}0, \boldsymbol{\Theta}\right) + \log p_0\right) + w_1^{ij}\left(\log p\left(\boldsymbol{y}^{ij} \middle| k^{ij}{=}1, \boldsymbol{\Theta}\right) + \log p_1\right),
\end{aligned}
\tag{3.28}
$$

where w_l^{ij}, $l \in \{0, 1\}$, denotes the confidence degree of observation $\boldsymbol{y}^{ij}$, given by

$$
\begin{aligned}
w_1^{ij} &= p\left(k^{ij}{=}1 \middle| \mathbf{y}^{ij}, \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\
&\propto \widehat{p}_{1_{(t)}}\; p\left(\boldsymbol{y}^{ij} \middle| k^{ij}{=}1, \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\
&\propto \widehat{p}_{1_{(t)}}\; \mathcal{N}\left(\boldsymbol{y}^{ij}; \widehat{\boldsymbol{A}}_{(t)}(\overline{\boldsymbol{x}}^i + \boldsymbol{D}^i\widehat{\boldsymbol{b}}_{(t)}) + \widehat{\boldsymbol{t}}_{(t)}, \boldsymbol{\Sigma}^i\right), \qquad (3.29) \\
w_0^{ij} &= p\left(k^{ij}{=}0 \middle| \mathbf{y}^{ij}, \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\
&\propto \widehat{p}_{0_{(t)}}\; \mathcal{U}\left(V_{\widetilde{\boldsymbol{x}}^i}\right), \qquad (3.30)
\end{aligned}
$$

such that $w_0^{ij} + w_1^{ij} = 1$. These weights correspond to the probability of the labels assigned to the observation $\boldsymbol{y}^{ij}$ being $k^{ij}{=}1$ and $k^{ij}{=}0$, respectively, given the current model estimate.

3.4.1.2 M-step

Given the most recent estimates of the unknown parameters at iteration t, $\widehat{\boldsymbol{\Theta}}_{(t)} = (\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{t}}, \widehat{\boldsymbol{b}}, \widehat{\boldsymbol{p}})_{(t)}$, the M-step aims to solve the following optimization problem

$$\widehat{\boldsymbol{\Theta}}_{(t+1)} = \arg\max_{\boldsymbol{\Theta}} Q_{\mathrm{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) . \tag{3.31}$$

This is done by taking the derivative of $Q_{\mathrm{ML}}(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)})$ with respect to each parameter and equating to zero. We simplify this step by sequentially updating the transformation and deformation parameters, as follows: 1) compute $\widehat{\boldsymbol{a}}_{(t+1)}$ and $\widehat{\boldsymbol{t}}_{(t+1)}$, assuming $\boldsymbol{b} = \widehat{\boldsymbol{b}}_{(t)}$ fixed; then 2) compute $\widehat{\boldsymbol{b}}_{(t+1)}$ assuming both $\boldsymbol{a} = \widehat{\boldsymbol{a}}_{(t+1)}$ and $\boldsymbol{t} = \widehat{\boldsymbol{t}}_{(t+1)}$ fixed; and finally, 3) update the probabilities of each model $\widehat{\boldsymbol{p}}_{(t+1)}$. These three steps can be solved using standard matrix calculus as will be described next.

Update of the Transformation Parameters Let $\boldsymbol{x}^i = (x_1^i, x_2^i)^\top = \overline{\boldsymbol{x}}^i + \boldsymbol{D}^i \widehat{\boldsymbol{b}}_{(t)}$, be the deformed shape according to the previous estimate of the deformation parameters, $\widehat{\boldsymbol{b}}_{(t)}$. The optimization of (3.31) with respect to the transformation parameters, $\boldsymbol{t} = (t_1, t_2)^\top$ and $\boldsymbol{a} = (a_1, a_2)^\top$, yields the following equations

$$\frac{\partial}{\partial \boldsymbol{a}} Q_{\mathrm{ML}}(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}) = 0 \iff \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} \boldsymbol{X}^{i\top} \boldsymbol{\Sigma}^{i-1} \left(-\boldsymbol{y}^{ij} + \boldsymbol{X}^i \boldsymbol{a}_{(t+1)} + \boldsymbol{t}_{(t+1)}\right) = 0, \tag{3.32}$$

$$\frac{\partial}{\partial \boldsymbol{t}} Q_{\mathrm{ML}}(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}) = 0 \iff \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} \boldsymbol{\Sigma}^{i-1} \left(-\boldsymbol{y}^{ij} + \boldsymbol{X}^i \boldsymbol{a}_{(t+1)} + \boldsymbol{t}_{(t+1)}\right) = 0, \tag{3.33}$$

where

$$\boldsymbol{X}^i = \begin{bmatrix} x_1^i & -x_2^i \\ x_2^i & x_1^i \end{bmatrix}.$$

Combining (3.32) and (3.33) leads to the following linear system of equations

$$\sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} \begin{bmatrix} \boldsymbol{X}^{i\top} \boldsymbol{\Sigma}^{i-1} \boldsymbol{X}^i & \boldsymbol{X}^{i\top} \boldsymbol{\Sigma}^{i-1} \\ \boldsymbol{\Sigma}^{i-1} \boldsymbol{X}^i & \boldsymbol{\Sigma}^{i-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{a}}_{(t+1)} \\ \widehat{\boldsymbol{t}}_{(t+1)} \end{bmatrix} = \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} \begin{bmatrix} \boldsymbol{X}^{i\top} \boldsymbol{\Sigma}^{i-1} \boldsymbol{y}^{ij} \\ \boldsymbol{\Sigma}^{i-1} \boldsymbol{y}^{ij} \end{bmatrix}. \tag{3.34}$$

The transformation parameters are updated by solving (3.34). This can be interpreted as a weighted least squares solution to the alignment of two sets of points (the observation points $\{\boldsymbol{y}^{ij}\}$ and the corresponding model points $\{\boldsymbol{x}^i\}$). This estimation of the transformation parameters differs from the standard ASM [37] in the following:

1. There is no restriction on the number of observation points detected in the vicinity of each model point, M^i, whereas in the standard ASM, $M^i = 1$; and
2. Each observation point has a specific weight, w_1^{ij} in the estimation, as opposed to a constant weight of 1 in the standard ASM.

The first difference means that more observation points can be detected, thus reducing the importance of the boundary detection method and increasing the chance of detecting the true object boundary. As for the latter, since outliers often receive lower values of w_1^{ij} than the valid observations, the update of the transformation parameters will be less influenced by the presence of outliers.

Update of the Deformation Coefficients Once the transformation parameters

$$\widehat{\boldsymbol{A}}_{(t+1)} = \begin{bmatrix} \widehat{a_1} & -\widehat{a_2} \\ \widehat{a_2} & \widehat{a_1} \end{bmatrix}_{(t+1)}, \quad \widehat{\boldsymbol{t}}_{(t+1)} = \begin{bmatrix} \widehat{t_1} \\ \widehat{t_2} \end{bmatrix}_{(t+1)},$$

have been updated using (3.34), $Q_{\mathrm{ML}}(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)})$ in (3.28) is maximized with respect to $\boldsymbol{b}$, leading to

$$\left(\sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} {\boldsymbol{D}^i}^\top \widehat{\boldsymbol{A}}_{(t+1)}^\top {\boldsymbol{\Sigma}^i}^{-1} \widehat{\boldsymbol{A}}_{(t+1)} \boldsymbol{D}^i \right) \widehat{\boldsymbol{b}}_{(t+1)} = \\ = \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} {\boldsymbol{D}^i}^\top \widehat{\boldsymbol{A}}_{(t+1)}^\top {\boldsymbol{\Sigma}^i}^{-1} \left(\boldsymbol{y}^{ij} - \widehat{\boldsymbol{A}}_{(t+1)} \overline{\boldsymbol{x}}^i - \widehat{\boldsymbol{t}}_{(t+1)} \right). \quad (3.35)$$

Note that, as in the standard ASM methodology, the deformation coefficients obtained by solving (3.35) may correspond to an unexpected shape. Therefore, the coefficients may need to be shrink to acceptable values, which is done using (3.19).

Update of the Model Probabilities Finally, the estimates of the probabilities of each model, $\widehat{p_0}_{(t)}, \widehat{p_1}_{(t)}$, are updated. Maximizing $Q_{\mathrm{ML}}(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)})$ with respect to p_0 and p_1 yields

$$\widehat{p}_{1_{(t+1)}} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} + w_0^{ij}} = \frac{1}{M} \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij}, \quad (3.36)$$

$$\widehat{p}_{0_{(t+1)}} = 1 - \widehat{p}_{1_{(t+1)}}, \quad (3.37)$$

where $M = \sum_{i=1}^{N} M^i$ is the total number of detected observation points.

3.4.2 Maximum Posterior Estimation

The estimation of the model parameters using the MAP method is done in a similar way. However, instead of using the complete log-likelihood, this approach uses the complete log-joint probability, which can be expressed as

$$\begin{aligned} \mathcal{P}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) &= \log p(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) \\ &= \log \Big(p(\boldsymbol{Y}, \boldsymbol{K} | \boldsymbol{\Theta})\, p(\boldsymbol{\Theta}) \Big) \\ &= \log p\left(\boldsymbol{Y} | \mathbf{K}, \boldsymbol{\Theta}\right) + \log p\left(\boldsymbol{K}\right) + \log p(\boldsymbol{a}, \boldsymbol{t}) + \log p(\boldsymbol{b}) + \log p(\boldsymbol{p}) \end{aligned} \quad (3.38)$$

The first two terms in (3.38) are the same as in (3.27); the remaining terms are the prior probabilities for the transformation parameters, $(\boldsymbol{a}, \boldsymbol{t})$, deformation parameters, $\boldsymbol{b}$, and sensor probabilities, $\boldsymbol{p}$. As previously, the EM algorithm is used to estimate the model parameters by maximizing the expectation of (3.38).

The next subsection describes the prior probabilities used in this chapter and the two steps of the EM algorithm for the MAP formulation.

3.4.2.1 Prior Probabilities of the Model Parameters

Transformation Parameters In this chapter non-informative priors will be used to define the prior term of the transformation parameters $p(\boldsymbol{a}, \boldsymbol{t})$. Although no prior information about the transformation parameters is used here, these priors can be useful in many applications, as will be explored in Chapter 5.

Non-informative priors, pioneered by Jeffreys [78–80], are a class of probability density functions capable of expressing ignorance about the model parameters. Rather than describing *a priori* beliefs, they should be viewed as a way of letting the data "dominate", while staying inside a Bayesian framework [75].

For a generic model parameter $\alpha \in \mathbb{R}^n$, the Jeffreys' prior is defined by

$$p(\alpha) \propto \sqrt{\det\left(\mathcal{I}(\alpha)\right)}, \tag{3.39}$$

where $\mathcal{I}(\alpha)$ is the Fisher information matrix introduced by Fisher [52]. Its best known use is in the Cramer-Rao (lower) bound (see, *e.g.*, [88, 161] or [180]). The coefficients of $\mathcal{I}(\alpha)$ are given by

$$\mathcal{I}(\alpha)_{ij} = -\mathbb{E}_{\boldsymbol{Y}}\left[\frac{\partial^2 \log p\left(\boldsymbol{Y}|\alpha\right)}{\partial\alpha_i \partial\alpha_j}\right], \tag{3.40}$$

where $p\left(\boldsymbol{Y}|\alpha\right)$ is the likelihood of a set of observations $\boldsymbol{Y}$, given the model parameter α.

Now consider the problem described in Section 3.3. The transformation parameters, $\boldsymbol{a} = (a_1, a_2)^\top$ and $\boldsymbol{t} = (t_1, t_2)^\top$, are assumed to be independent, thus the prior can be factorized as $p(\boldsymbol{a}, \boldsymbol{t}) = p(\boldsymbol{a})p(\boldsymbol{t})$. The log-likelihood of a set of observation points $\boldsymbol{Y}$ is given by[1]

$$\log p\left(\boldsymbol{Y}|\boldsymbol{a}, \boldsymbol{t}\right) = \sum_{i=1}^{N}\sum_{j=1}^{M^i} \log \mathcal{N}\left(\boldsymbol{y}^{ij}; \boldsymbol{A}\boldsymbol{x}^i + \boldsymbol{t}, \boldsymbol{\Sigma}^i\right).$$

After straightforward manipulations, the Fisher information matrix for each of the transformation parameters, $\boldsymbol{a}$ and $\boldsymbol{t}$, is given by

$$\mathcal{I}(\boldsymbol{a}) = -\sum_{i=1}^{N}\sum_{j=1}^{M^i} {\boldsymbol{x}^i}^\top {\boldsymbol{\Sigma}^i}^{-1} \boldsymbol{x}^i \boldsymbol{I}, \tag{3.41}$$

$$\mathcal{I}(\boldsymbol{t}) = -\sum_{i=1}^{N}\sum_{j=1}^{M^i} {\boldsymbol{\Sigma}^i}^{-1}, \tag{3.42}$$

[1] We are only considering the model $k = 1$ from equation (3.23), since the outlier model does not depend on the parameters $\boldsymbol{a}$ and $\boldsymbol{t}$.

where $\boldsymbol{I}$ is a 2×2 identity matrix. This leads to the following Jeffreys' priors

$$p(\boldsymbol{a}) \propto \sqrt{|\mathcal{I}(\boldsymbol{a})|} = \sqrt{\left| -\sum_{i=1}^{N} \sum_{j=1}^{M^i} \boldsymbol{x}^{i\top} \boldsymbol{\Sigma}^{i-1} \boldsymbol{x}^i \boldsymbol{I} \right|} = c_a \tag{3.43}$$

$$p(\boldsymbol{t}) \propto \sqrt{|\mathcal{I}(\boldsymbol{t})|} = \sqrt{\left| -\sum_{i=1}^{N} \sum_{j=1}^{M^i} \boldsymbol{\Sigma}^{i-1} \right|} = c_t, \tag{3.44}$$

where $c_{\boldsymbol{a}}$ and $c_{\boldsymbol{t}}$ are constants that do not depend on the parameters $\boldsymbol{a}$ and $\boldsymbol{t}$. This means $p(\boldsymbol{a})$ and $p(\boldsymbol{t})$ are both uniformly distributed in $\mathbb{R}^2$.

Deformation Coefficients The prior for the deformation parameters $\boldsymbol{b}$ is obtained from the PCA method (used to learn the shape model), which assumes they are normally distributed with zero mean

$$p(\boldsymbol{b}) = \mathcal{N}\left(\boldsymbol{b}; \boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{b}}\right) = c_{\boldsymbol{b}} \exp\left(-\frac{1}{2} \boldsymbol{b}^\top \boldsymbol{\Sigma}^{\boldsymbol{b}-1} \boldsymbol{b}\right), \tag{3.45}$$

where $c_{\boldsymbol{b}}$ is a normalization constant, and $\boldsymbol{\Sigma}^{\boldsymbol{b}}$ is a $L \times L$ diagonal matrix whose entries are the eigenvalues $\boldsymbol{\Sigma}^{\boldsymbol{b}}_{ll} = \lambda_l$, obtained by the PCA method. This prior assigns higher probability to shapes that are similar to the average shape in the training set, *i.e.*, that are not too deformed. Furthermore, a wider Gaussian distribution is assigned to deformation modes that show more variability.

Model probabilities For the model probabilities, since we do not know which model is more likely, the following prior is used [50]

$$p(p_0) = p(p_1) = \frac{1}{2} = c_{\boldsymbol{p}}, \tag{3.46}$$

which does not depend on the parameters and ensures that $p_0 + p_1 = 1$.

3.4.2.2 E-step

Replacing the priors (3.43)-(3.46) in (3.38) yields

$$\mathcal{P}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) = \log p\left(\boldsymbol{Y} | \boldsymbol{K}, \boldsymbol{\Theta}\right) + \log p(\boldsymbol{K}) + \log c_{\boldsymbol{a}} + \log c_{\boldsymbol{t}} + \log c_{\boldsymbol{p}} + \log c_{\boldsymbol{b}} - \frac{1}{2} \boldsymbol{b}^\top \boldsymbol{\Sigma}^{\boldsymbol{b}-1} \boldsymbol{b}, \tag{3.47}$$

Assuming independence between the observations, this equation can be rewritten as

$$\mathcal{P}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) = \sum_{i=1}^{N} \sum_{j=1}^{M^i} \log p\left(\boldsymbol{y}^{ij} \middle| k^{ij}, \boldsymbol{\Theta}\right) + \log p\left(k^{ij}\right) + C - \frac{1}{2} \boldsymbol{b}^\top \boldsymbol{\Sigma}^{\boldsymbol{b}-1} \boldsymbol{b}, \tag{3.48}$$

where $C = \log c_{\boldsymbol{a}} + \log c_{\boldsymbol{t}} + \log c_{\boldsymbol{p}} + \log c_{\boldsymbol{b}}$ is a constant. Thus, given an estimate of the model parameters, $\widehat{\boldsymbol{\Theta}}_{(t)}$, and a set of observation points, $\boldsymbol{Y}$, the expectation of (3.48) is

$$\begin{aligned} Q_{\text{MAP}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) &= \mathbb{E}_{\boldsymbol{K}}\left[\mathcal{P}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{\Theta}) \middle| \boldsymbol{Y}, \widehat{\boldsymbol{\Theta}}_{(t)}\right] \\ &= \mathbb{E}_{\boldsymbol{K}}\left[\sum_{i=1}^{N}\sum_{j=1}^{M^i} \log p\left(\boldsymbol{y}^{ij} \middle| k^{ij}, \boldsymbol{\Theta}\right) + \log p\left(k^{ij}\right)\right] + C - \frac{1}{2}\boldsymbol{b}^\top {\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1}\boldsymbol{b}, \end{aligned} \tag{3.49}$$

where the first term in the expectation is given in (3.28), *i.e.*,

$$Q_{\text{MAP}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) = Q_{\text{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) + C - \frac{1}{2}\boldsymbol{b}^\top {\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1}\boldsymbol{b}. \tag{3.50}$$

The constant C can be discarded since it does not change the outcome of the maximization step. Therefore, with these priors, the MAP formulation only differs from the ML in the last term, which will affect the estimation of the deformation coefficients.

3.4.2.3 M-step

The maximization of $Q_{\text{MAP}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right)$ in (3.50) with respect to the model parameters is similar to the ML estimation. As previously mentioned, because we use non-informative priors for the transformation parameters and model probabilities, the only difference to the ML estimation is in the case of the deformation parameters $\boldsymbol{b}$. The updates will be detailed in the following steps.

Update of the Transformation Parameters Since the prior terms in (3.50) do not depend on the transformation parameters, $(\boldsymbol{a}, \boldsymbol{t})$, these parameters are updated by solving (3.34), as in the previous formulation.

Update of the Deformation Coefficients Maximizing (3.50) with respect to the deformation coefficients, $\boldsymbol{b}$, yields an additional term, due to the Gaussian prior in (3.45). The update is achieved by solving

$$\begin{aligned} &\left[{\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1} + \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij} {\boldsymbol{D}^i}^\top \widehat{\boldsymbol{A}}_{(t+1)}^\top {\boldsymbol{\Sigma}^i}^{-1} \widehat{\boldsymbol{A}}_{(t+1)} \boldsymbol{D}^i\right] \widehat{\boldsymbol{b}}_{(t+1)} = \\ &\qquad \left[\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij} {\boldsymbol{D}^i}^\top {\widehat{\boldsymbol{A}}^\top}_{(t+1)} {\boldsymbol{\Sigma}^i}^{-1} \left(\boldsymbol{y}^{ij} - \widehat{\boldsymbol{A}}_{(t+1)}\overline{\boldsymbol{x}}^i - \widehat{\boldsymbol{t}}_{(t+1)}\right)\right], \end{aligned} \tag{3.51}$$

where the term ${\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1}$ "pulls" the coefficients of $\widehat{\boldsymbol{b}}_{(t+1)}$ closer to zero. This means that the algorithm will try to keep the deformation coefficients low, which guarantees that shape of the segmentation is similar to the shapes in the training set.

Update of the Model Probabilities The derivative of (3.50) with respect to the model probabilities, p_0 and p_1, yields the same update equation as the ML estimation case, given by (3.36)-(3.37).

3.4.3 Comparison With Karush-Kuhn-Tucker Conditions

Karush-Kuhn-Tucker (KKT) conditions can be used to add inequality constraints to an unconstrained optimization problem. This is a generalization of the Lagrange multipliers, which allow adding equality constraints to the solution [28].

Consider the update of the deformation parameters in the ML approach. The additional step in (3.19) aims at finding the estimate $\widehat{\boldsymbol{b}}$ that satisfies the condition in (3.18). This can be viewed as an inequality constraint on the solution of the optimization problem in (3.31). Formally, we wish to find

$$\begin{aligned} \widehat{\boldsymbol{b}}_{(t+1)} = \quad & \underset{\boldsymbol{b}}{\arg\max} \quad Q_{\mathrm{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\ & \text{subject to} \quad \boldsymbol{b}^{\top}{\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1}\boldsymbol{b} \leq d_{\max}^2. \end{aligned} \tag{3.52}$$

The KKT conditions allow us to obtain the solution to this problem by finding the value of $\boldsymbol{b}$ that satisfies the following equation

$$\nabla_{\boldsymbol{b}} Q_{\mathrm{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) - \mu \nabla_{\boldsymbol{b}} g(\boldsymbol{b}) = 0 \tag{3.53}$$

where $\mu \geq 0$ is a constant - a KKT multiplier, and $g(\boldsymbol{b}) = \boldsymbol{b}^{\top}{\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1}\boldsymbol{b} - d_{\max}^2$ is the inequality constraint in (3.18). This leads to the solution

$$\begin{aligned} &\left[\mu{\boldsymbol{\Sigma}^{\boldsymbol{b}}}^{-1} + \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}\mu{\boldsymbol{D}^i}^{\top}\widehat{\boldsymbol{A}}_{(t+1)}^{\top}{\boldsymbol{\Sigma}^i}^{-1}\widehat{\boldsymbol{A}}_{(t+1)}\boldsymbol{D}^i\right]\widehat{\boldsymbol{b}}_{(t+1)} = \\ &\qquad\left[\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}{\boldsymbol{D}^i}^{\top}\widehat{\boldsymbol{A}}_{(t+1)}^{\top}{\boldsymbol{\Sigma}^i}^{-1}\left(\boldsymbol{y}^{ij} - \widehat{\boldsymbol{A}}_{(t+1)}\overline{\boldsymbol{x}}^i - \widehat{\boldsymbol{t}}_{(t+1)}\right)\right]. \end{aligned} \tag{3.54}$$

This means that the MAP estimation of the deformation parameters in (3.51) is a particular case of (3.54), for $\mu = 1$.

3.4.4 Comparison Between the ML and the MAP Formulations

The difference between the ML and the MAP formulations lies in the estimation of the deformation parameters. The ML formulation has an additional step that shrinks the estimate of the deformation parameters, using (3.19), in order to guarantee that the corresponding segmentation has an acceptable shape and is not over-deformed [37]. This is required because the objective function $Q_{\mathrm{ML}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right)$ defined in (3.28), does not contain any (prior) information about the deformation coefficient. Without the additional step,

the algorithm could give significantly different solutions from those found in the training set. On the other hand, in the MAP formulation, the objective function $Q_{\text{MAP}}\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right)$, defined in (3.50), includes a constraint on the deformation coefficients due to the prior $p(\boldsymbol{b})$. This constraint is obtained in a natural and principled way and "pushes" the estimates to zero, thus inducing a shrinkage effect similar to the one in (3.19). Furthermore, in the ML formulation, the shrinkage step in (3.19) scales all the components of $\boldsymbol{b}$ uniformly, disregarding the importance of each mode of variation, whereas in the MAP estimation the shrinkage effect is weighted by the covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{b}}$ (recall (3.51)), which means it depends on the eigenvalue of each deformation mode.

3.5 Algorithm Overview

The algorithm described in the previous sections can be summarized as follows. Given an initial estimate of $\widehat{\boldsymbol{\Theta}}_{(t=0)}$, the shape model parameters are iteratively updated by:

1. *Searching for observation points* in the vicinity of the model;

E-Step

2. *Computing the observation probabilities* w_1^{ij}, w_0^{ij} using (3.29) and (3.30);

M-Step

3. *Updating the transformation parameters* $\boldsymbol{a}, \boldsymbol{t}$ using (3.34);
4. *Updating the deformation coefficients* $\boldsymbol{b}$ using (3.35) and (3.19) in the ML case, or (3.51) in the MAP case; and
5. *Updating the model probabilities* p_1, p_0 using (3.36)-(3.37).

This process is repeated until no significant changes are observed in the contour. This algorithm is denoted as Expectation-Maximization Robust Active Shape Model (EM-RASM).

3.6 Experimental Setup

The evaluation of EM-RASM is divided in two experiments: 1) in synthetic data, to illustrate the difference between the proposed method and the standard ASM; and 2) in the segmentation of the LV in CMR data. In both experiments, the quantitative evaluation of the results is based on the metrics described in Section 3.6.2.

The experiments using synthetic data compare the performance of EM-RASM against the standard ASM using two example images. These images will also be used to evaluate

the robustness of the method and its sensitivity to: 1) the presence of outliers; 2) the model initialization; 3) over/under training; and 4) the amount of deformation in the training set. Finally, we will also show the computational overhead of the proposed method compared to the standard ASM.

The proposed method is applied to the segmentation of the LV in CMR data from the dataset described in Section 3.6.1. Using the proposed approach, a single shape model is learned based on training shapes from different slices and different frames. This shape model is then used to segment the images in new MR sequences. The accuracy of the proposed method is compared to another state of the art method, proposed in [143], which combines an ASM with RANSAC [51] to determine which observation points are outliers.

Furthermore, in order to show that the proposed approach is more general than CMR segmentation, results are shown for two additional applications: the segmentation of the LV in ultrasound images, and the segmentation of the lip in face images.

Sections 3.6.4 and 3.6.3 details the model parameters used in the experiments, and describe the method used to detect the observation points.

3.6.1 CMR Dataset

The evaluation of the proposed methodologies is based on a publicly available dataset [11], acquired at the Department of Diagnostic Imaging of the Hospital for Sick Children in Toronto, Canada, using the acquisition technique bSFFP, namely the FIESTA scan protocol of a GE Genesis Signa MR scanner.

The dataset contains CMR data from 33 different patients, two of whom are healthy, two have an unknown diagnosis, and the remaining ones have a cardiomyopathy, such as tachycardia, ischemia, and ventricular hypertrophy. The patients are all children or teenagers, whose age ranges between 2 and 17 years old. For each patient, the CMR data is a sequence of 20 volumes that start and end in the diastolic phase, with the end-systolic phase around the 9-th volume. The number of slices in each volume varies from 8 to 15 slices. The LV is not present in all of these slices, but rather in just 4 to 10 of them. Spacing between the slices also depends on the patient, and ranges between 6 to 13 mm. Each slice is a 256×256 image with an average resolution of 1.4 ± 0.2 mm/pixel. The dataset also provides a manual segmentation of the LV border, that is used for evaluation purposes as ground truth (GT). These manual segmentations were processed so that the first point in all the contours was located on the junction of the left and right ventricles, closest to the posterior interventricular sulcus, and the remaining points were distributed in arc-length, with a total of 30 points.

3.6.2 Quantitative Evaluation Metrics

The segmentations obtained with the proposed methods are compared with the GT, and the accuracy is quantitatively determined using four metrics:

1. the Dice coefficient [47], d_{Dice};
2. the average minimum distance of each model point to the GT, d_{AV} [138];
3. the LV volume difference, d_{VD} [11]; and
4. the percentage of good segmentations [68],

which are described below.

Let S be a binary 2D image such that pixels inside the LV border have the value one and pixels outside have the value zero. Denoting S_1 as the binary image associated with the proposed segmentation and S_2 as the corresponding GT image. The Dice coefficient is computed as follows

$$d_{\text{Dice}}(S_1, S_2) = 2\frac{C(S_1 \wedge S_2)}{C(S_1) + C(S_2)}, \tag{3.55}$$

where $C(\cdot)$ counts the number of pixels within the region (number of ones) and $\wedge$ denotes a pixel-wise AND operator. A Dice coefficient of 1 means there is a perfect match between the segmentations and a value of 0 means the corresponding regions do not even overlap.

The average metric is defined as follows. Let $\mathcal{S}_1 = \left\{\boldsymbol{s}_1^1, \boldsymbol{s}_1^2, \ldots, \boldsymbol{s}_1^N\right\}$ be a collection of N 2D points that defines the proposed LV border in a specific image, and $\mathcal{S}_2 = \left\{\boldsymbol{s}_2^1, \boldsymbol{s}_2^2, \ldots, \boldsymbol{s}_2^M\right\}$ be the collection of M 2D points with the corresponding GT segmentation. Then, the average minimum distance metric is given by

$$d_{\text{AV}}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{N}\sum_{i=1}^{N} \min_j \|\boldsymbol{s}_1^i - \boldsymbol{s}_2^j\|. \tag{3.56}$$

Regarding the LV volume difference, d_{VD}, it is computed using the summation of discs technique, *i.e.*, by computing the area of the each segmentation, multiplying by the slice thickness, and then summing for all the slices in the volume.

Finally, the percentage of "good" segmentations has also been used in the literature [68, 69], because it provides a quantitative way to measure the number of wrong segmentations. These failures are not simply inaccurate segmentations, which could potentially be corrected through user interaction, but gross errors that should be discarded altogether. The most common criterion for deciding if a segmentation is "good" is if it has $d_{\text{AV}} \leq 5$ mm. Thus, the percentage of "good" segmentations is given by

$$\%\ \text{good} = \frac{\#\ \text{good segmentations}}{\#\ \text{segmentations}} \tag{3.57}$$

In the results shown in the next section, the average minimum distance metric is measured in millimeters (unless otherwise stated), and the volume difference in milliliters.

3.6.3 Detection of Observation Points

In this chapter, the observation points were obtained by searching for edge points along lines orthogonal to the contour at each model point, as originally proposed in [25]. The feature detection algorithm used was the match filter for edges (see [25] Section 5.2). This edge detector takes an image profile (1D signal of the image intensity along the search line) and convolves it with an edge mask. Edge points correspond to the maxima of the resulting filtered signal, which are detected by applying thresholding and non-maximum suppression. The threshold allows the sensitivity of the edge detector to be tuned, which may be different for each application. The length of the search line is also application-dependent since it is related to the uncertainty of the contour.

The ASM, as described in [37], performs a similar search method, but the observation points correspond to the strongest image edge along each search line. Furthermore, in this approach there is always a one-to-one mapping between model and observation points, while in the proposed approach there is a one-to-many mapping between a model point and the observations. Consequently, the total number of detected observation points is typically larger in the EM-RASM approach.

3.6.4 Parameters

The user defined parameters used in the proposed method were chosen based on several trials, by selecting the parameters that led to the best results. Some of these parameters vary depending on the application. For instance, the variance of the observation model for valid edge points, $\mathbf{\Sigma}^i$, $i = 1, \ldots, N$, was defined as a diagonal matrices, all with the same diagonal value, *i.e.*,

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}. \tag{3.58}$$

In the synthetic data tests, the variance was set to $\sigma = 10$, whereas in the LV segmentation in CMR it was defined as $\sigma = 5$. The same number of model points, $N = 40$, was used in both experiments. The training shapes were resampled in arc-length, such that the first point was located at the same position in all the examples. As for the number of deformation modes, in the synthetic data we used $L = 2$, and in the CMR data $L = 10$.

The length of the search line used to extract edge points from the image is related to the expected variance of the observation model. For the synthetic data, the length was set to 80 pixels in each direction (*i.e.*, 160 pixels in total), while in the LV segmentation it was set to 15 pixels (30 pixels in total). Regarding the thresholding operation, used exclude low gradient edge points, the threshold used in CMR data was $g_{\min} = 0.4 g_{\max}$, where $g_{\max}$ is the maximum gradient along each search line. In the synthetic data, different values for this parameter were used to control the amount of edge points detected, as will be explained below.

Regarding the initial guess of the shape model parameters, in all the tests performed, the model was initialized with the average shape $\overline{\boldsymbol{x}}$ (*i.e.*, $\boldsymbol{b}_{(0)} = \mathbf{0}$). The initial guess for the transformation parameters was obtained either by aligning the average shape $\overline{\boldsymbol{x}}$ with a contour obtained by human input using the standard least squares method (see (3.7)), or by providing an initial guess of the translation and scale of the model. The initial guess for the models probabilities was $p_0 = p_1 = 0.5$. No evidence was found suggesting that the initial values for these probabilities significantly changed the output of the algorithm.

3.7 Results

3.7.1 Synthetic Images

The performance of the EM-RASM method was evaluated in two synthetic images. These images consist of two different corrupted versions of a binary image of a rectangle, with intensity value 1 inside the rectangle and 0 outside. In the first example, the binary image was corrupted by white Gaussian noise with zero mean and variance $\sigma^2_{\text{noise}} = 0.5$ (see top left corner of Fig. 3.5). In the second example, the rectangle image was corrupted by black regions (see top right corner of Fig. 3.5).

The goal is to retrieve the correct location of the rectangle given an initial guess. However, in the corrupted versions of the image, additional observation points will be detected on edges that do not belong to the rectangle boundary, which means the set of observation points will be noisy. In these examples, the results using the ML and MAP estimation methods were very similar. For the sake of clarity in the presentation, we show only the results using the ML method.

The shape statistics was learned using synthetic data generated by adding random Gaussian perturbations to the true object boundary $\overline{\boldsymbol{x}}$. More specifically, each training example $\boldsymbol{x} \in \mathbb{R}^{2N\times 1}$ is a realization of

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{e} \ , \ \boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\text{train}}\mathbf{I}) \tag{3.59}$$

where $\sigma_{\text{train}} = 2$ is the standard deviation imposed on the synthetic model points. Since the Gaussian noise is isotropic and identical for all model points, the shape model is not expected to identify any significant modes of variation (which is why only two modes of deformation are used, *i.e.*, $L = 2$).

Fig. 3.5 shows the output of the proposed algorithm and the output of the standard ASM for different initial guesses. In the first example, the standard ASM was unable to cope with the outliers and, consequently, it did not fit the rectangle border accurately. The EM-RASM, on the other hand, was able to accurately segment the rectangle despite the detection of many outliers (note the large number of red dots in the images that do not lie along the rectangle edges). In the second synthetic example, the gradient associated to the edges of the black regions is stronger than the one associated to the edges of the

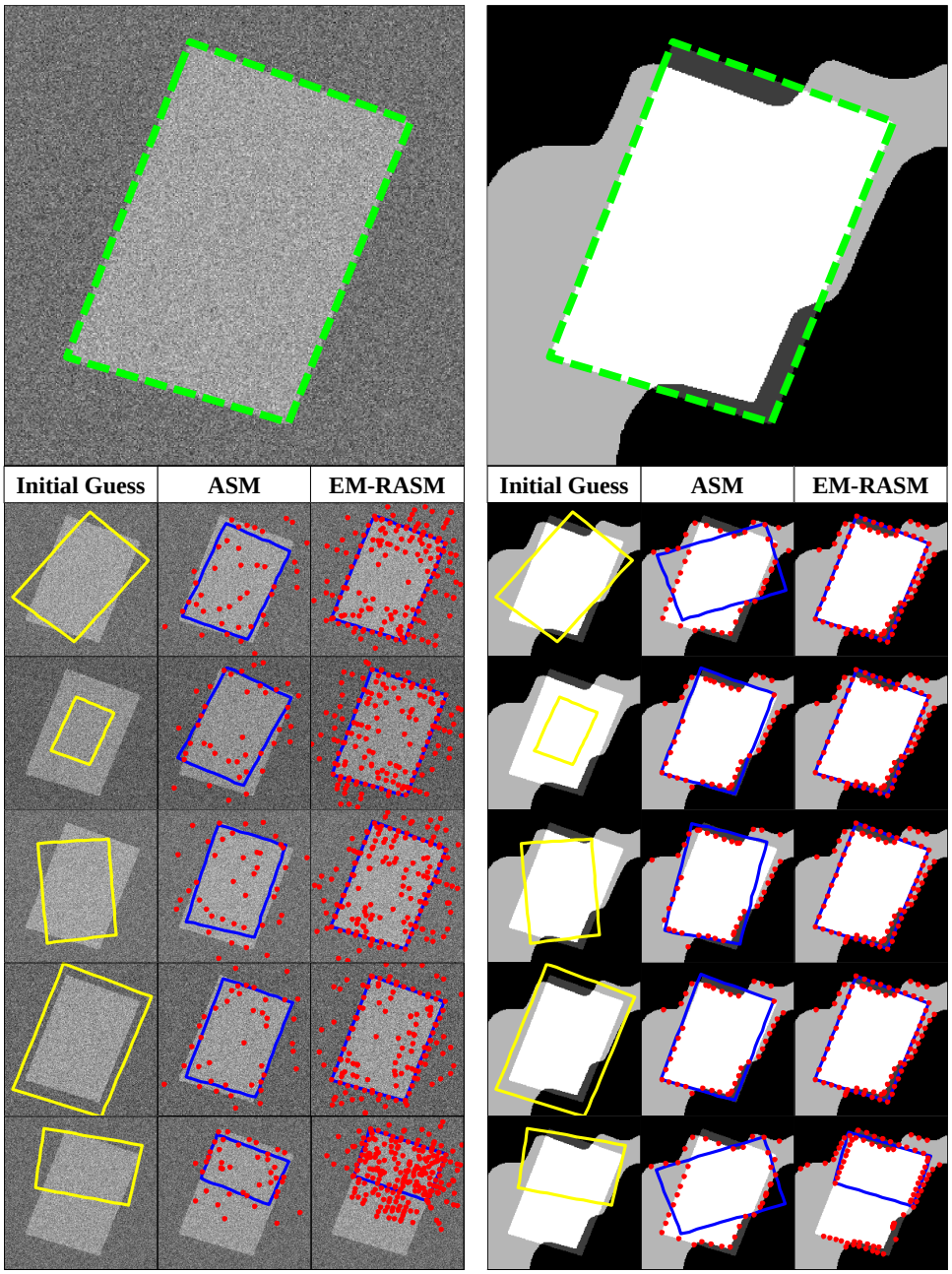


Figure 3.5: Segmentation of synthetic images of a rectangle: (top left) corrupted by white Gaussian noise with variance $\sigma^2_{\text{noise}} = 0.5$; and (top right) corrupted by black regions. The ground truth segmentation is shown in dashed green. Each row shows an example using a different initial guess (left column) and the final segmentation using the standard ASM (middle column) and the EM-RASM (right column). The red dots correspond to the detected observation points in the last iteration.

rectangle. Consequently, some of the detected observation points are outliers. The figure shows that the presence of these outliers causes the standard ASM to be unable to correctly segment the image, since it tries to fit all the observation points simultaneously (including outliers). The EM-RASM, on the other hand, is able to accurately segment the rectangle and disregard the outliers. The last row in the figure shows an example in which neither the standard ASM nor the EM-RASM were able to segment the rectangle. This was due to a poor initialization of the model parameters, as shown by the initial guess on the left column. In this case, due to an incorrect estimation of the orientation, the models end up fitting only the borders in the upper part of the rectangle.

3.7.1.1 Sensitivity to the presence of outliers

Further tests were performed using the first synthetic example (see Fig. 3.5 left), in order to provide statistical evidence that the proposed approach is robust in the presence of a large number of observation points. The shape model used in the previous section is used here again in several trials. In each trial, the test image was generated using different a value of $\sigma^2_{\text{noise}} \in [0, 1]$ (variance of the noise added to the binary image). Larger values of σ^2_{noise} lead to noisier images, which, in turn, lead to the detection of a larger number of outliers. In the final iteration of each trial, the number of outliers is measured by counting all the observation points located more than 2 pixels away from the true rectangle boundary. Depending on the percentage of outliers (number of outliers / number of observation points), we placed each trial in one of four categories: (a) $0 - 20\%$, (b) $20 - 40\%$, (c) $40 - 60\%$, and (d) $60 - 80\%$.

Fig. 3.6 shows the segmentation accuracy for each category, each with 100 segmentation trials, *i.e.*, a total of 400 trials are represented in the figure. The boxplots show that the standard ASM (on the left) and the proposed method (on the right) have a similar performance when few outliers are detected (up to 20%). However, for higher percentages of outliers, there is a clear increase in the error using the standard ASM, whereas the EM-RASM keeps the error value low, even for percentages of outliers of 60-80%. Note that the proposed method allows many observation points to be extracted for each model point. This means, on one hand, that there is a higher probability of detecting the valid observation points. On the other hand, it also means that the number of outliers (and thus the percentage of outliers) is often also higher.

3.7.1.2 Sensitivity to the model initialization

In order to demonstrate the sensitivity of the proposed method to the initialization, its performance was evaluated using the image of the second synthetic example (see Fig. 3.5 right) and with different initializations. Each transformation parameter was analyzed separately by varying its initial value while keeping the remaining parameters unchanged.

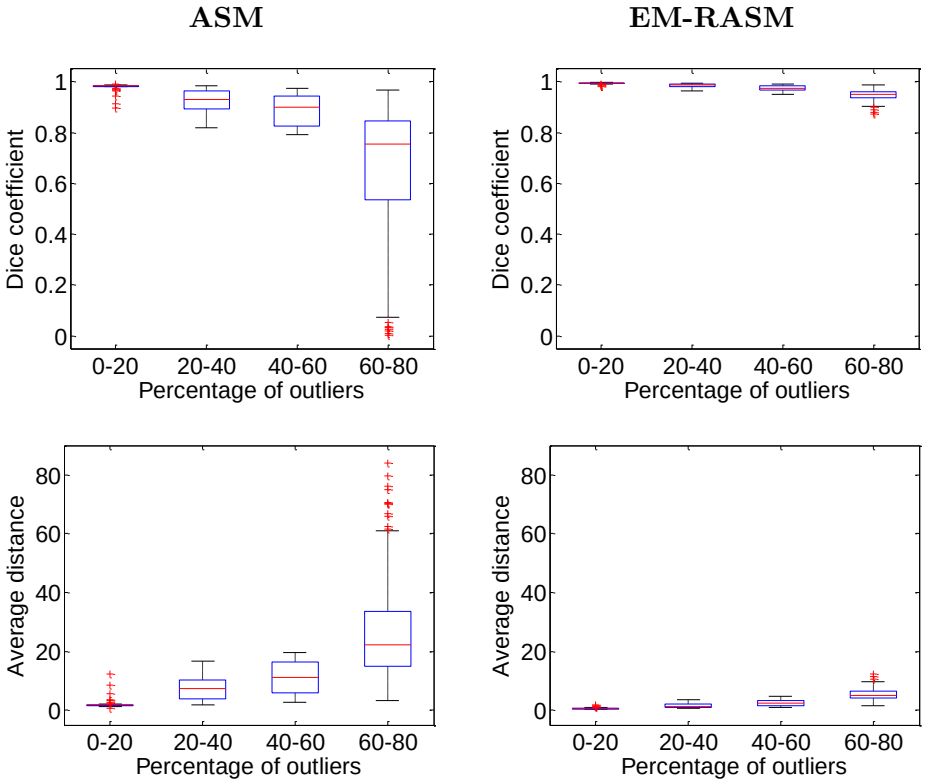


Figure 3.6: Comparison between the performance of ASM and EM-RASM for different percentages of outliers. The top row shows the accuracy results using the Dice coefficient and the bottom row using the average distance metric (measured in pixels).

For a clearer interpretation of the results, the transformation parameters a_1 and a_2 are replaced as *scale*, s, and *rotation*, ϕ, which are related to a_1 and a_2 by $a_1 = s\cos(\phi)$ and $a_2 = s\sin(\phi)$. The translation parameters, $\boldsymbol{t} = [t_1, t_2]^\top$, are shown separately as translation in the first and second dimension of the image, respectively.

Fig. 3.7 shows the performance of the algorithm obtained for each initialization test, using the ML approach (the results for the MAP approach were very similar). In this example, the interval of parameter values that achieves good performances is essentially bounded by the values that still initialized the whole model within the image domain. Outside these intervals, the performance of the algorithm decreases significantly. The rotation is the exception, which shows that for initial rotation angles of $|\phi| > 45°$, the algorithm is unable to accurately segment the rectangle. The reason is it tries to fit the model assuming a 90° rotation (see last row of Fig. 3.5 for an example).

3.7.1.3 Sensitivity to over/under training

Typically, learning algorithms require large training datasets to be flexible enough to perform well in the test dataset, but they may also overfit the training data. In this case, more training data means that the shape statistics are more accurate, which influences

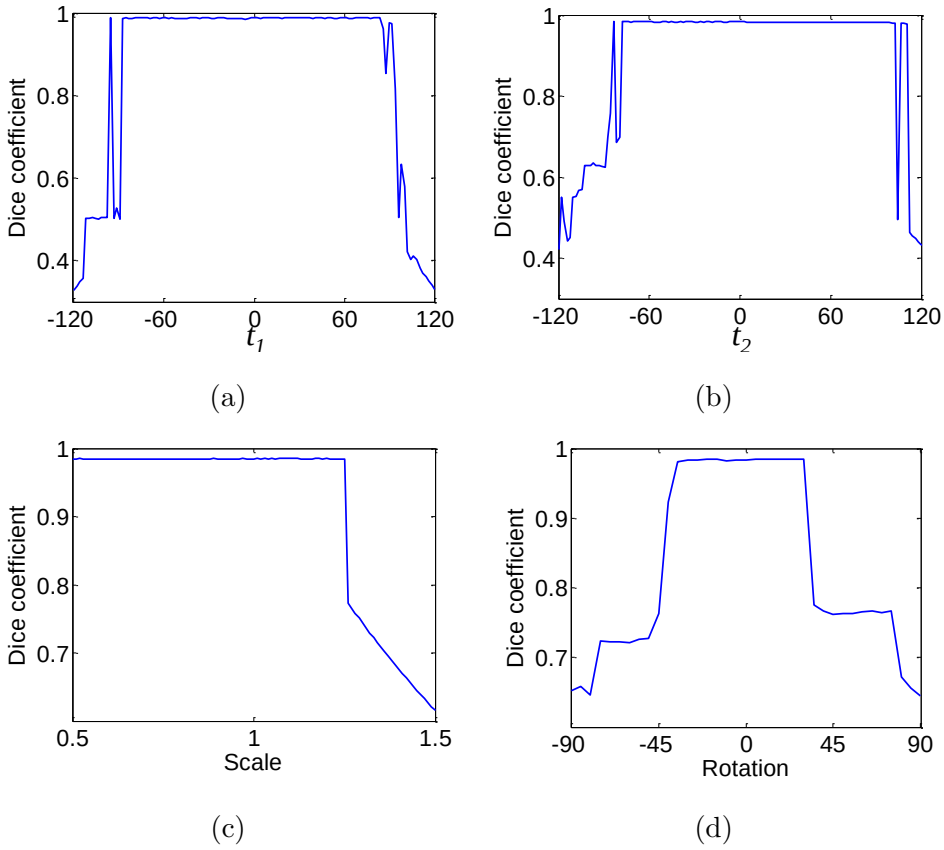


Figure 3.7: Performance of EM-RASM for different initializations. Each plot shows the Dice coefficient of the algorithm (ML) for different initial values of the transformation parameter: (a) t_1, (b) t_2, (c) scale, and (d) rotation.

the estimation of the deformation coefficients $\boldsymbol{b}$.

In order to evaluate the sensitivity of the proposed algorithm to over/under training, we performed several segmentation tests using an increasing number of training shapes for the segmentation of the image in the second synthetic example in Fig. 3.5. The training shapes were obtained using (3.59), which adds random Gaussian noise to the true object boundary. A total of 10^5 training shapes were generated: in the first test, only the first 10 training shapes were used; in the second test, the first 100 training shapes were used; and so on. In other words, the n-th test used 10^n training shape, including all the training shapes used in the $(n-1)$-th test.

Fig. 3.8 shows the superiority of the MAP over the ML. The MAP approach is able to outperform the ML when using a larger number of training shapes, suggesting that the prior information enriches the shape estimates as the training data increases. Nevertheless, the ML also exhibits good performances with a Dice coefficient of approximately $0.990 - 0.995$ and an average error of approximately $0.6 - 1.0$ pixels.

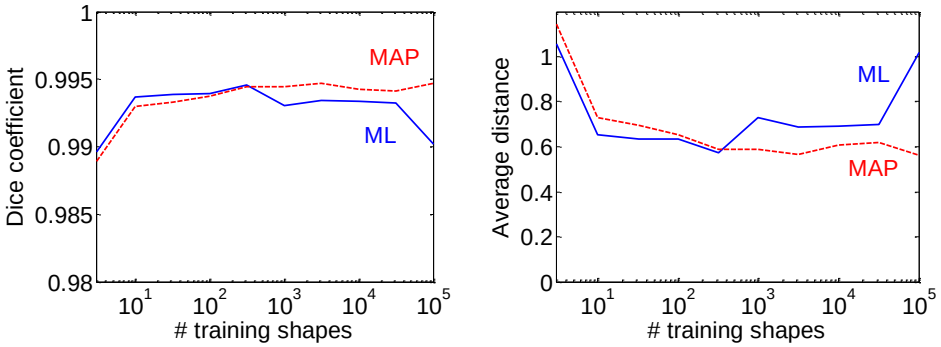


Figure 3.8: Performance of the ML and MAP formulations of EM-RASM for different sizes of the training set. The left plot shows the Dice coefficient and the right plot shows the average distance (in pixels). The blue lines shows the performance for the ML approach and the dashed red line the performance of the MAP approach.

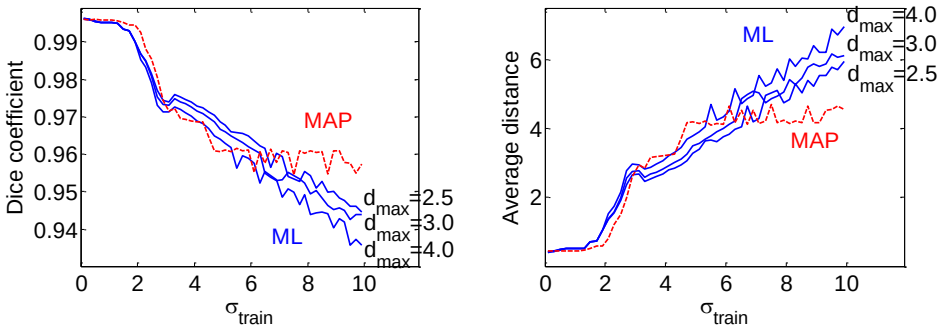


Figure 3.9: Performance of the ML and MAP formulations of EM-RASM for different values of σ_{train}. The left plot shows the Dice coefficient and the right plot shows the average distance (in pixels). The dashed red line depicts the performance of the MAP approach and the blue lines the ML approach for $d_{\max} = \{2.5, 3.0, 4.0\}$.

3.7.1.4 Sensitivity to amount of deformation

The sensitivity of the proposed algorithm to the amount of deformation in the training set was also tested. Several experiments were performed using $\sigma_{\text{train}} \in [0.1, 10]$ with increments of 0.1 to generate the training examples (see equation (3.59)). For each value of σ_{train}, we generated 1000 training shapes. The tests were performed using the image in the second synthetic example (see Fig. 3.5 on the left).

If the value of the diagonal observation model variance is set to $\sigma = \sigma_{\text{train}}$, then increasing the value of σ_{train} also makes the Gaussian distribution become wider. This means that the probabilities w_1^{ij} and w_0^{ij} become less distinct, and, thus, that the model is not able to clearly distinguish outliers from valid observation points, resulting in poorer segmentations.

The performance of the proposed methods is shown in Fig. 3.9. The results show

that the MAP approach performs better than the ML for larger values of σ_{train}. In order to show the influence of $d_{\max}$ in the ML approach (recall equation (3.19)), three different values were tested, $d_{\max} = \{2.5, 3.0, 4.0\}$. It is possible to see that larger values of $d_{\max}$ lead to poorer segmentations. However, in real problems, the object segmentation typically does not correspond to the average contour. Therefore, setting $d_{\max}$ too low can be prejudicial to the performance of the ML approach. Fig. 3.9 shows that the MAP formulation has a slightly lower performance when few training shapes are used, whereas for larger training sets, it outperforms the ML formulation.

3.7.1.5 Computational performance comparison

This section compares the computational overhead of the proposed algorithm against the standard ASM. The algorithm overview shown in Section 3.5 is similar to the standard ASM, with the addition of steps 2) *Computing the observation probabilities* and 5) *Updating the model probabilities*. However, the bottleneck remains the update of the model parameters (steps 3) and 4)). Regarding the update equations, the difference in computational complexity between the two methods lies in the number of detected observation models. The proposed algorithm allows multiple observation points for each model point, which has a cost in computational performance.

In order to evaluate the difference in computational performance, both algorithms (the proposed one and the standard ASM) were used on the first synthetic image (Fig. 3.5 on the left) using different values for $\sigma^2_{\text{noise}} \in [0, 1]$. This experiment was repeated until each algorithm ran a total of 1000 iterations. For each iteration, the running time and the number of detected observations were recorded.

Fig. 3.10 shows the computational performance as a function of the number of detected observations for each iteration. The blue crosses denote the results for the standard ASM and the red dots correspond to the results using the proposed algorithm. Note that the standard ASM always detects one observation for each model point. The results show that the computational performance is linearly proportional to the number of detected observations. When both algorithms detect the same number of observations points, their computational performance is very similar.

3.7.2 LV Segmentation in CMR

The proposed method was evaluated on the CMR dataset described in Section 3.6.1, which contains 33 sequences. For each test sequence, the shape statistics were learned using the remaining 32 sequences, *i.e.*, using a leave-one-sequence-out scheme.

The initialization of the transformation parameters for each test sequence was obtained by aligning the average contour $\overline{\boldsymbol{x}}$ with a circle in the image, whose center and radius was given by human input. This initialization was only performed for the first

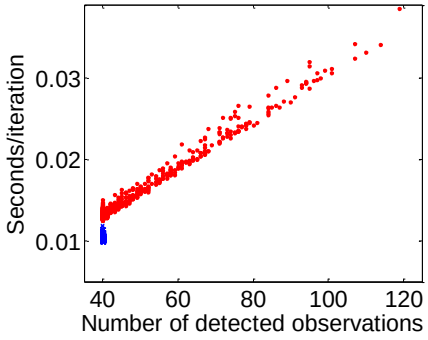


Figure 3.10: Comparison of the computation performance between ASM and EM-RASM. The blue crosses correspond to the standard ASM results and the red dots to the proposed method.

Table 3.1: Statistical results of the LV segmentation in CMR using EM-RASM and comparison with ASM and RANSAC: mean value (standard deviation).

	ASM	RANSAC	**EM-RASM**	
			ML	**MAP**
d_{Dice}	73.0 (9.5)	79.9 (7.3)	**84.2 (9.2)**	84.1 (9.5)
d_{AV}	4.7 (2.1)	3.3 (1.3)	**2.58 (1.68)**	2.63 (1.79)
d_{VD}	25.7 (38.8)	13.5 (17.8)	**11.8 (11.4)**	11.8 (12.0)
% Good	64.2	82.0	**89.2**	88.7

slice of each volume. The remaining slices were initialized by successively propagating the segmentation of one slice to the next.

Fig. 3.11 shows examples of the segmentations obtained with the MAP formulation of EM-RASM and the corresponding GT. Despite the irregularity of the LV border, both in terms of shape and image intensity, most of the times the algorithm is able to discern in which edges it should trust and which it should discard. Examples of images that were not so accurately segmented are shown in the last row of Fig. 3.11, where it is possible to see that the algorithm was misled by the papillary muscles or with the intensity transition associated with the outer border of the myocardium.

Table 3.1 shows statistical results of the segmentation accuracy. Comparing the accuracy of the proposed method with the accuracy of the standard ASM, it is possible to see that the proposed approach significantly improves the segmentation results. EM-RASM is also able to achieve better results than the RANSAC approach, even though the latter is also a significant improvement over the standard ASM. Comparing the performance of the ML and the MAP formulations, the table shows that ML has a slightly better performance. Fig. 3.9 suggests that having a larger training set may lead to a reversal of this results, *i.e.*, as more training shapes are used to learn the shape model, the performance of the MAP formulation should improve and eventually outperform the ML formulation.

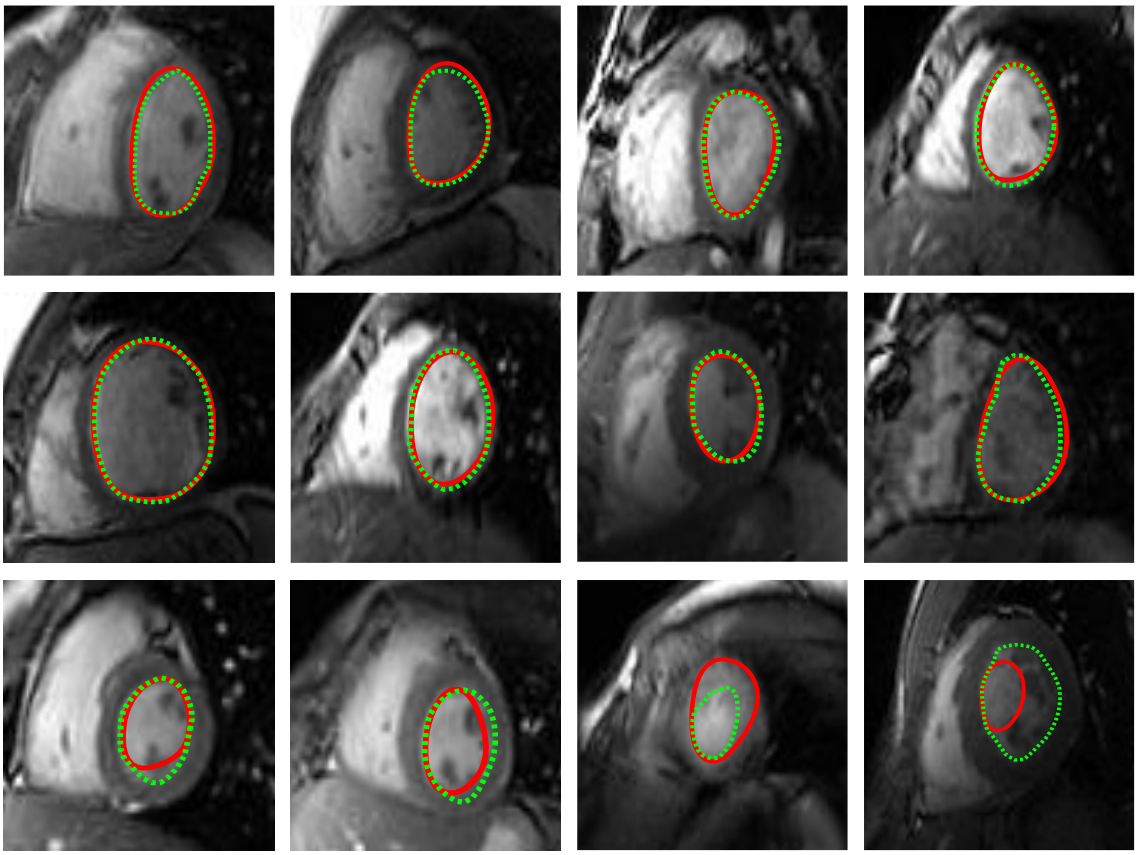


Figure 3.11: Segmentation of the LV in CMR images. Each plot shows one slice of a different MR sequence with the segmentation obtained using the MAP formulation of EM-RASM (red), and the corresponding GT (green).

3.7.3 Other applications

3.7.3.1 LV Segmentation in Ultrasound

The EM-RASM method was applied to the segmentation of the left ventricle in 2D ultrasound image sequences. The dataset is composed of five 2D sequences (five different patients), each with 16-20 frames, leading to a total of 87 images.

The shape model was trained using manual medical annotations of the left ventricle contours (ground truth). Each training example was obtained by resampling, in arc-length, the medical contours with a fixed number of points from the bottom left to the apex (top) and from the bottom right to the apex. We tested the proposed algorithm and the standard ASM using a leave-one-sequence-out cross validation strategy.

In each test sequence, the initial guess for the transformation parameters was obtained by aligning the average contour $\overline{\boldsymbol{x}}$ with a contour obtained by human input using the standard least squares method. A different human input contour was used in each test sequence, and the resulting initial guess was used in all the frames of the sequence (*i.e.*, we did not propagate the contours from one frame to the next).

Table 3.2: Dice coefficient of the LV segmentation in ultrasound images using EM-RASM and comparison with ASM: mean value (standard deviation).

	ASM	**EM-RASM**	
		ML	**MAP**
Seq 1	0.80 (0.05)	0.86 (0.05)	**0.87 (0.05)**
Seq 2	0.79 (0.04)	**0.87 (0.03)**	**0.87 (0.05)**
Seq 3	0.71 (0.05)	**0.87 (0.02)**	**0.87 (0.02)**
Seq 4	0.76 (0.05)	0.89 (0.04)	**0.90 (0.03)**
Seq 5	0.81 (0.04)	**0.89 (0.03)**	**0.89 (0.02)**
Average	0.78 (0.06)	**0.88 (0.04)**	**0.88 (0.03)**

Table 3.3: Average distance error of the LV segmentation in ultrasound images using EM-RASM and comparison with ASM: mean value (standard deviation).

	ASM	**EM-RASM**	
		ML	**MAP**
Seq 1	17.7 (4.2)	11.5 (3.6)	**11.1 (3.3)**
Seq 2	21.6 (4.5)	**12.1 (2.2)**	12.4 (2.2)
Seq 3	24.4 (4.0)	**9.5 (1.5)**	9.6 (1.3)
Seq 4	19.3 (3.6)	7.8 (2.7)	**7.5 (2.2)**
Seq 5	19.3 (3.6)	**10.2 (2.6)**	**10.5 (2.7)**
Average	20.4 (4.6)	**10.3 (3.0)**	**10.3 (2.9)**

Fig. 3.12 shows one frame of each sequence with the segmentation obtained with EM-RASM (using the ML approach) and with the standard ASM. In all the examples, a large number of the detected observations (red dots) are outliers. The figure shows that the EM-RASM performed better than the standard ASM and was able to fit the LV boundary, whereas the accuracy of the segmentation obtained using the standard ASM was hampered by the outliers.

Statistical results are presented in Tables 3.2-3.3 and in Fig. 3.13. The results show that the ML and the MAP approach perform similarly and that they lead to a significant improvement in accuracy over the standard ASM (approximately half the error using the average distance metric).

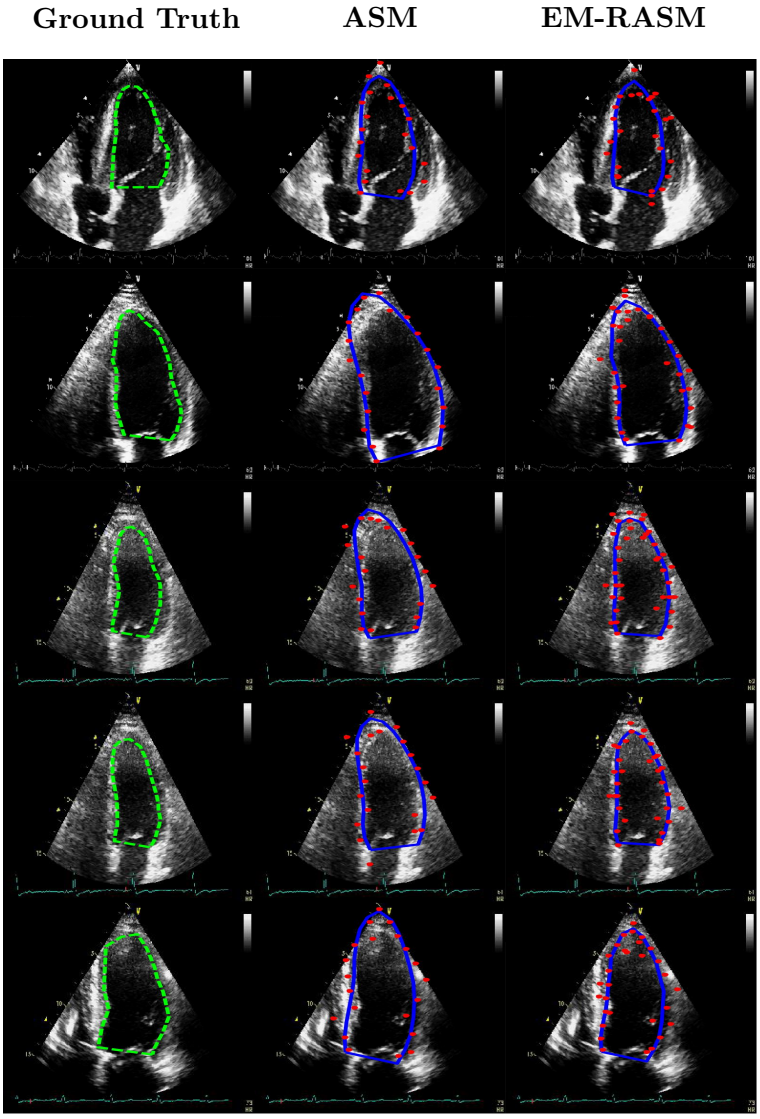


Figure 3.12: Segmentation of the LV in ultrasound images. Each row shows one frame from a different sequence. The green dashed line shows the ground truth (left column); the blue lines correspond to the estimated segmentation using the standard ASM (mid column) and the proposed algorithm (right column); and the red dots represent the detected observations in the last iteration.

3.7.3.2 Lip Segmentation

The last application in which the EM-RASM method was tested was the segmentation of the lip. The dataset consists of four sequences of face images. These sequences were acquired from the neutral expression samples of the Cohn-Kanade expression database [104], each with 10-20 frames, for a total of 58 images. The training shapes were obtained by resampling, in arc-length, the lip contours provided in the database so that the number of points in the lower lip and the number of points in the upper lip were the same in all the contours. As in the previous case, the shape model was trained using a leave-one-sequence-out cross validation strategy. The initial guess for the transformation

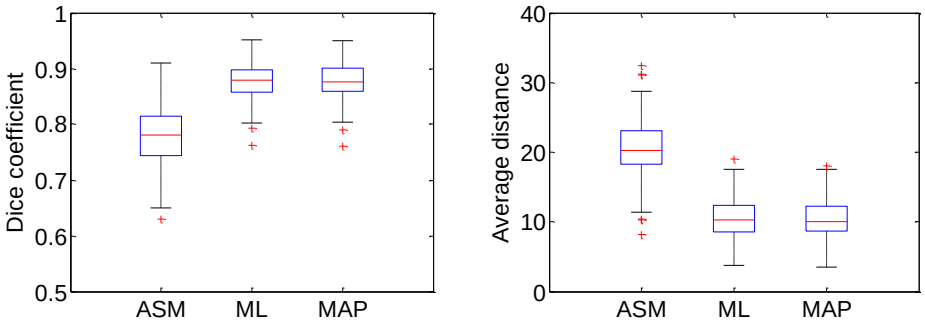


Figure 3.13: Boxplot of the results of the LV segmentation in ultrasound images (the average distance is measured in pixels).

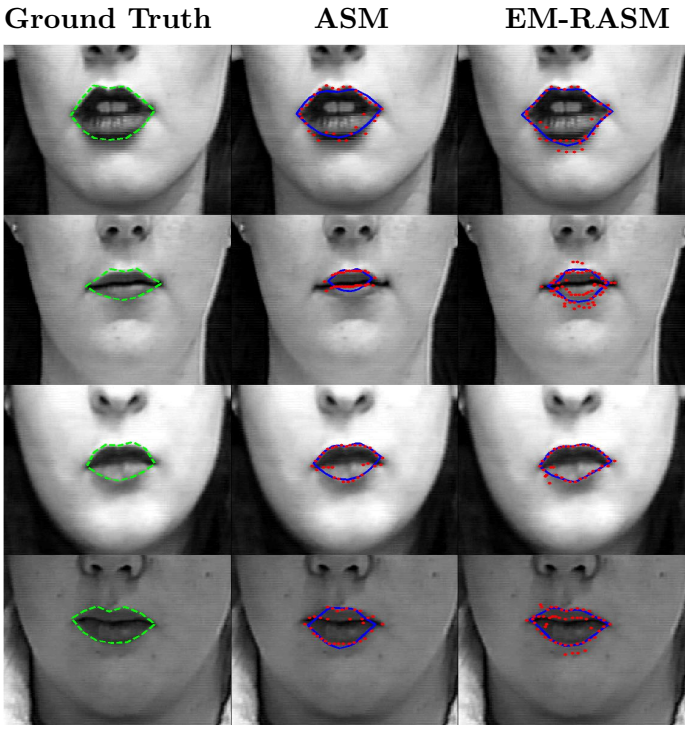


Figure 3.14: Segmentation of the lip in face images using EM-RASM and comparison with ASM. Each row shows one frame from a different sequence. The green dashed line shows the ground truth (left column); the blue lines correspond to the estimated segmentation using the standard ASM (mid column) and the proposed algorithm (right column); and the red dots represent the detected observations in the last iteration.

parameters for each test sequence was obtained by aligning the average contour $\overline{\boldsymbol{x}}$ with a contour obtained by human input. The same initial guess was used in all the frames of the test sequence. The performance of the proposed method was compared with the approach proposed in [143], which combines ASM with RANSAC.

Tables 3.4-3.5 and Fig. 3.15 show the statistical evaluation of EM-RASM, using the d_{Dice} and d_{AV} error metrics. The results compare the proposed approach with the standard

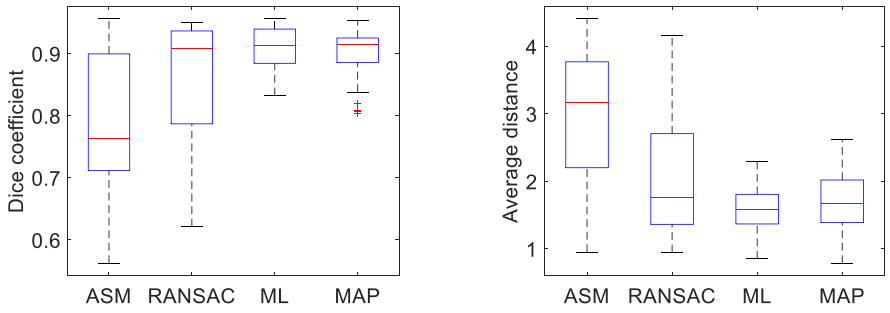


Figure 3.15: Boxplots of the results of lip segmentation in face images (the average distance is measured in pixels).

Table 3.4: Dice coefficient of the segmentation of the lip using the ML and MAP formulations of EM-RASM and comparison with ASM and RANSAC: mean value (standard deviation).

	ASM	RANSAC	**EM-RASM**	
			ML	**MAP**
Seq 1	0.91 (0.02)	**0.94 (0.01)**	**0.94 (0.01)**	0.92 (0.01)
Seq 2	0.67 (0.06)	0.74 (0.08)	**0.88 (0.03)**	0.87 (0.04)
Seq 3	0.73 (0.02)	0.91 (0.04)	**0.94 (0.01)**	0.93 (0.02)
Seq 4	0.85 (0.05)	**0.91 (0.01)**	**0.91 (0.01)**	**0.91 (0.01)**
Average	0.78 (0.11)	0.86 (0.10)	**0.91 (0.03)**	0.90 (0.04)

Table 3.5: Average distance error of the segmentation of the lip using the ML and MAP formulations of EM-RASM and comparison with ASM and RANSAC: mean value (standard deviation).

	ASM	RANSAC	**EM-RASM**	
			ML	**MAP**
Seq 1	2.0 (0.5)	**1.4 (0.1)**	1.5 (0.1)	1.9 (0.3)
Seq 2	3.6 (0.3)	3.2 (0.8)	**1.8 (0.3)**	**1.8 (0.5)**
Seq 3	4.0 (0.4)	1.6 (0.7)	**1.1 (0.3)**	1.2 (0.4)
Seq 4	2.3 (0.6)	**1.5 (0.2)**	1.7 (0.2)	**1.5 (0.2)**
Average	3.0 (0.9)	2.1 (1.0)	**1.6 (0.3)**	1.7 (0.4)

ASM and the RANSAC approach proposed in [143]. As in the previous segmentation problem, the proposed method outperforms the standard ASM. These results also show that the proposed method is more accurate than the RANSAC method, although the latter also performs significantly better than the standard ASM and has a similar performance to EM-RASM in some sequences. It is also possible to see that the ML and the MAP approaches have a similar performance.

Fig. 3.14 shows examples of the segmentations obtained with EM-RASM (using the ML approach) and with the standard ASM. It is possible to see that the ability to detect more than one observation point for each search line increases the probability of detecting the true object boundary, particularly if that boundary is not associated with a strong edge (*e.g.*, in Fig. 3.14 second and third rows). On the other hand, despite the consequent increase in the number of outliers, the proposed approach was able to fit the valid observations. This does not happen in the standard ASM, in which the obtained contour was misguided by the outliers. In some cases, if the number of outliers is small, both methods perform similarly, which is in agreement with the boxplots shown in Fig. 3.6.

3.8 Conclusions

In this chapter, a 2D ASM is combined with a robust estimation of the model pose and deformation using an outlier model, an algorithm denoted by EM-RASM. Contrary to the standard ASM, the proposed approach allows multiple observation points to be extracted from the image for each model point, and assumes that these points have some probability of being outliers. The estimation of the model parameters is achieved using the EM method that weights each observation point by the probability of that point belonging to the object boundary. Outliers typically have a lower probability of belonging to the object boundary, which means that their contribution to the estimation of the model parameters is reduced. This allows EM-RASM to accurately estimate the model parameters even in the presence of outliers. The results show EM-RASM significantly outperforms the standard ASM, both in synthetic and real images.

Regarding the segmentation of the LV in CMR, the robustness to outliers allows the shape model to disregard some of the anatomical structures that typically mislead LV segmentation algorithms, such as papillary muscles and the epicardium. Nonetheless, and even though the proposed method outperforms the standard ASM and the RANSAC approaches, the segmentation accuracy for CMR segmentation can still be improved by using additional information, namely, the three-dimensional structure of the LV and the temporal dynamics of the heart throughout the cardiac cycle. This is explored in the following chapters of this thesis.

A different approach to the 2D segmentation of the LV was also explored in Appendix C. It is inspired in [46, 111] and consists in analyzing the images in polar coordinates and using dynamic programming to determine the location of the LV border. It has the advantage of providing faster segmentations, but the EM-RASM approach has a better performance.

4

3D Segmentation

Contents

This chapter addresses the LV segmentation as a 3D problem. Since the LV is a three-dimensional structure, we explore how a 3D shape model can be used in the context of CMR segmentation. This has potential advantages over the previous 2D approach because it correlates the segmentations of the slices in a particular MR volume, leading to segmented regions that are more consistent with the 3D structure of the LV.

4.1 Introduction

Contrary to the 2D approach used in the previous chapter, a 3D shape model is able not only to capture the three-dimensional shape of the LV but also allows specific shape patterns to be identified for each slice of a volume. Unfortunately, applying a 3D shape model to the LV segmentation in CMR is not straightforward and some crucial considerations have to be taken into account.

First, the size of the LV varies from patient to patient based on the size of the heart. Consequently, the number of slices in the MR volume in which the LV is visible also varies [71]. This makes learning the shape model difficult, because the training examples are not equally sampled, which is required in order to use standard procedure for learning shape models [37].

Second, as mentioned in Chapter 2, different levels of expiration during the acquisition of CMR data can cause shifts in the position of the heart [11], leading to the appearance of misaligned slices (recall Fig. 2.8 (a)). These misalignments are unexpected and highly variable, which means that examples affected by this issue may introduce artifacts in the estimated shape of the LV. This makes learning the 3D shape model harder. Furthermore, it is also necessary to take this into account during the segmentation process. Since misalignments are not well represented by the shape statistics, the shape model will be unable to fit the LV border in such volumes.

These two issues affect both the training phase of an ASM methodology and the test phase, in which new volumes are segmented. For the training phase, these issues can be combined to formulated the following problem: *How can we learn the shape statistics of the LV from training data with a variable number of misaligned slices*? For the test phase, two problems have to be addressed: 1) *How can a 3D shape model be used to segment MR volumes with a variable number of slices*? and 2) *how will the shape model be able to cope with misaligned slices*? This chapter addresses these problems by proposing a framework to learn a 3D shape model from CMR data and to use it to segment new volumes.

The remainder of this chapter is organized as follows. Section 4.2 provides a background on how other methods have used 3D shape models for the segmentation of the LV in CMR, and how it differentiates from the proposed approach. Section 4.3 describes the proposed strategy to learn the 3D shape model from CMR data. Section 4.4 explains the proposed approach to segment the LV using the 3D shape model on new MR volumes to

obtain the LV segmentation. Sections 4.5 and 4.6 discuss the experimental setup and the results obtained. Finally, Section 4.7 concludes the chapter with final remarks about the proposed method.

4.2 Background

Many shape models are based on the Point Distribution Model (PDM) [37], in which the border of an object is defined using a set of specific landmarks (recall Section 3.2.1). When these landmarks are uniquely identified in all the shape examples, it is easy to analyze the distribution of corresponding landmarks to learn the mean shape and the main modes of deformation. However, in 3D shape models, establishing correspondences between landmarks of two objects may not be trivial [66]. In CMR, the number of slices in each volume varies, thus the number of contours that describe the 3D shape of the LV also varies (see Fig. 4.1). In this context, establishing landmark correspondences is not straightforward.

The Iterative Closest Point (ICP) algorithm [23] is a common approach to overcome this issue [31, 32, 54]. This algorithm aligns two sets of points (shapes) by iteratively performing the following steps: i) establishing landmark correspondences, based on the distance between them; and ii) estimating the pose transformation that aligns the sets of corresponding points. However, it assumes that the number of points in each example is the same. Variants of the ICP algorithm have been proposed to allow the algorithm to deal with surfaces with a different numbers of landmarks [33, 81, 140]. In these cases, they rely on the assumption that some of the landmarks may be discarded and the alignment is determined based only on a subset of landmarks that are common across the examples. In CMR data, the examples are defined by a variable number of contours at different positions along the LV axis, as shown in Fig. 4.1. Therefore, the ICP algorithm is only able to establish landmark correspondences in the basal and apical slices. That is why many works in the 3D LV segmentation problem use other approaches.

Mitchell et al. [118], for example, propose the use of a normalized cylindrical coordinate system to define the position of the landmarks. First, they resample each contour at predefined angle intervals around the LV axis, which guarantees that all the 2D contours have the same number of landmarks. It also guarantees that there is a correspondence between the landmarks in consecutive slices. Then, they interpolate the position of the landmarks along the LV axis using linear interpolation. This allows the contours in the shape examples to be resampled. A similar approach is also used by Andreopoulos et al. [11].

Other works perform a preliminary step to establish the landmarks' position and correspondences based on volumetric registration [53, 177]. They use a volumetric mesh, in which voxels are labeled based on the type of structure they belong to. With this in-

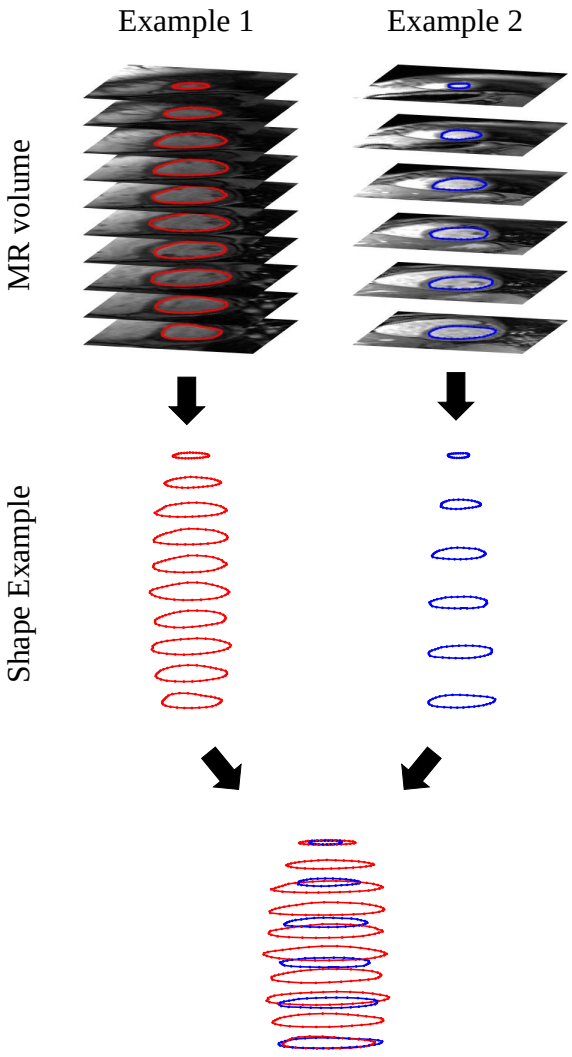


Figure 4.1: Example of two training shapes that are used to learn the shape model. Besides the contours in the basal and apical slices, it is not possible to establishing landmark correspondences between the two shape examples.

formation, the alignment of a specific volume in the training set with a reference volume is obtained using a non-rigid registration algorithm [144]. After aligning the two volumes, the landmarks in the reference volume are supposed to segment the training volume accurately, despite having a different discretization. Then, knowing the transformation used to align them, it is possible to extract the new landmarks of the training volume by applying the reverse transformation to the reference landmarks. This procedure is similar to the approach adopted in atlas-based segmentation algorithms [98, 99, 101], in which the segmentation of a volume is indirectly obtained by aligning the volume with a previously segmented atlas. Doing this for all the shape examples leads to a training set with shape examples that have the same landmarks as the reference volume.

Another alternative consists of using the level-set method [132] has also been used.

For instance, Tsai et al. [174] learn the shape statistics using the signed distance maps in the training set, instead of the landmark's positions. This approach has the advantage of not requiring landmark correspondences between shapes, as well as the advantage of being able to change the topology of the segmentation. The latter is not of particular interest for the problem of segmentation of the LV in CMR, since the LV does not change topology, but may be useful in other applications such as the segmentation of brain structures [173].

During the analysis of new volumes, the misalignment between slices is often disregarded [95, 102]. For 2D approaches, this does not constitute a severe problem, as each slice segmentation is independent, so there is no notion of misalignment. However, for 3D approaches, and in particular those using shape models, it may be difficult to fit the shape model to the LV border in all slices. In order to deal with this problem, Andreopoulos et al. [11], for instance, use preprocessing step to correct the slice position before applying the 3D model. The preprocessing step is based on an image registration procedure for translation only. A similar procedure is described in [102], although here the registration is also based on longitudinal CMR images. However, without knowing the location of the LV, performing a registration is, by itself, a complex task, and is the basis of many segmentation algorithms [98, 101, 115, 191, 192].

Another issue that arises in the analysis of new volumes is that the learned shape model and volumes may have a different number of slices, which means that some of the contours in the model will not correspond to the observed slices. Instead, the contours will be located in between slices. In these cases, finding the necessary image features to segment the volume may require interpolating voxel intensities [87, 118]. However, in CMR, the distance between consecutive slices is very large (compared to the distance between pixels within a slices) and the intensity transitions along the LV axis are not smooth. Combining this with the existence of misaligned slices makes interpolation a poor choice. Instead, the method proposed in [176, 177] interpolate the position of the shape model points to obtain new points located within the volume slices. These new points are used to extract the image features and to determine the location of the LV border. Then, this information is propagated to the original model points to update their position.

The method proposed in this chapter differs from the approaches mentioned above in several ways:

In the training stage:

1. Before the computation of the shape statistics, a translation-based pre-registration is performed to remove slice misalignments. This is necessary to remove artifacts induced by these misalignments.

2. A polynomial interpolation scheme is applied to the shape examples in the training set, in order to resample the contours along the LV axis, and to establish correspondences between them.

3. Two additional deformation modes are used to allow the shape model to shift the position of each contour within the slice plane, thus enabling the model to fit the LV border in misaligned slices.

In the test phase:

1. The learned shape model (*i.e.*, mean shape and main modes of deformation) is resampled taking into account the number of slices in the test volume, in order to avoid interpolating voxel intensities when searching for the LV border.

2. The estimation of the model parameters is achieved using an extension of the EM-RASM algorithm to 3D data.

A schematic representation of the proposed method is shown in Fig. 4.2. The following sections describe each step in detail.

4.3 Training Phase

This section describes a framework to learn a 3D shape model from CMR data. The training data consists of MR volumes with a variable number of slices and their corresponding annotations (expert segmentation of the LV). In each slice, the expert segmentation is defined by a contour with N points (landmarks), as in the previous chapter. In order to learn a 3D shape model, each training example is the collection of expert contours for a specific volume, which means that each example is described by a variable number of contours. Since the shape model is based on the PDM [37], the training examples are required to have the same number of landmarks. To meet this requirement, we propose to resample them along the LV axis so that all the training examples are represented by the same number of contours, S.

Let us consider a specific MR volume, say, v, in the training set. The volume slices are located at equally spaced positions, s_m, along the LV axis

$$s_m = \frac{m-1}{S^v - 1}, \tag{4.1}$$

where $m = 1, \ldots, S^v$, and $S^v \neq S$ is the total number of slices in the volume v. We assume that the first slice, $s_1 = 0$, corresponds to the basal slice, whereas the last slice, $s_{S^v} = 1$, corresponds to the apical slice. Therefore, the LV contour at the m-th slice is defined by a $2N^v \times 1$ vector

$$\widetilde{\boldsymbol{x}}_v(s_m) = \begin{bmatrix} \widetilde{\boldsymbol{x}}^1(s_m) \\ \widetilde{\boldsymbol{x}}^2(s_m) \\ \vdots \\ \widetilde{\boldsymbol{x}}^{N^v}(s_m) \end{bmatrix} \tag{4.2}$$

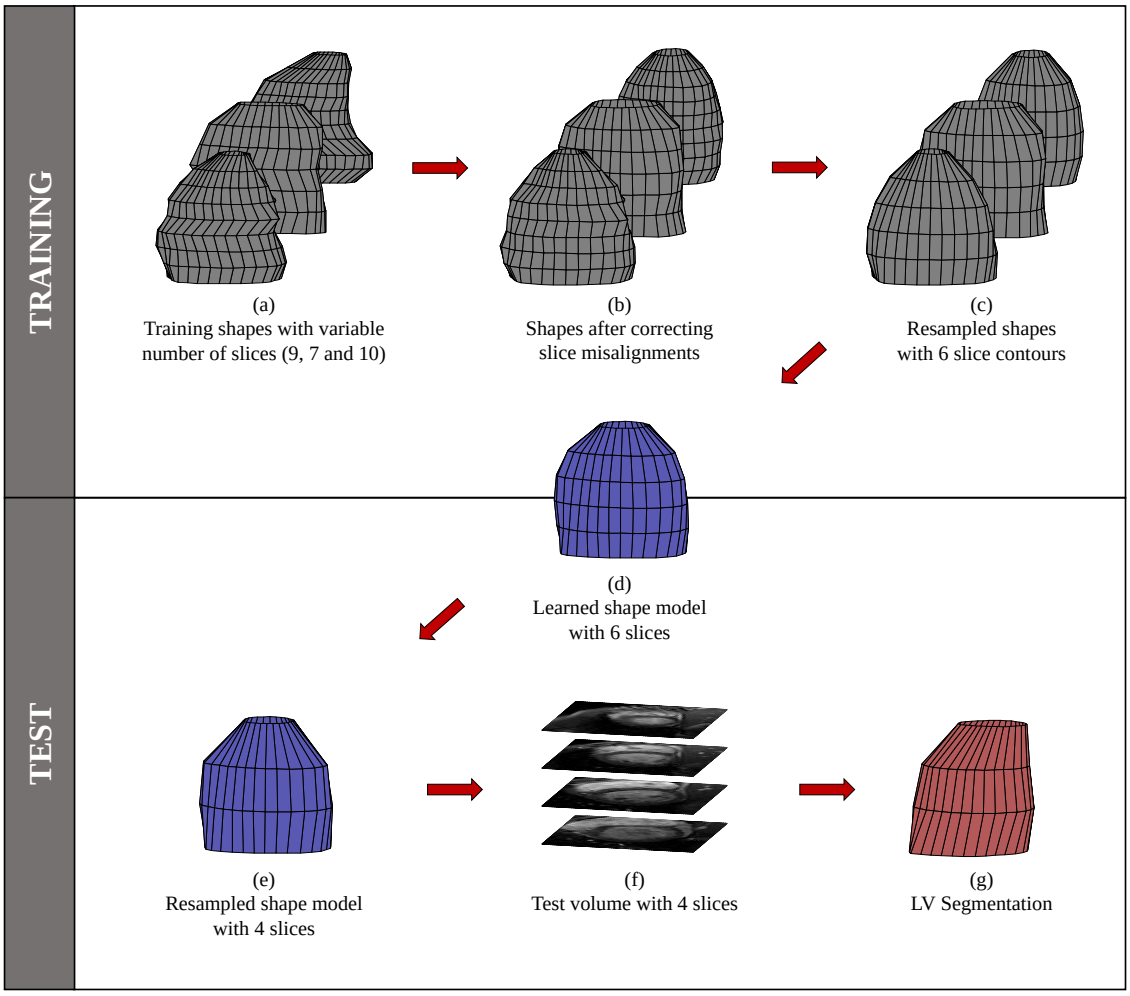


Figure 4.2: Illustration of the proposed methodology using a 3D shape model, comprising the training and test phases. From (a) the initial training shapes with a variable number of contours, it is necessary to perform (b) a correction of misaligned slices, before (c) resampling all the models to a pre-defined number of slices (in this case six slices); after this stage, (d) the shape model is obtained comprising both the mean shape and the main modes of the deformation; then, (e) the learned shape model is resampled, before applying it to segment the (f) a test volume that may have a different number of slices (in this case four slices), after which (g) the segmentation of the LV is obtained.

where $N^v \neq N$ is the number of points in each contour of volume v, and $\widetilde{\boldsymbol{x}}^i(s_m) = \left(x_1^i(s_m), x_2^i(s_m)\right)^\top \in \mathbb{R}^{2\times 1}$ is the position of the i-th contour point in the m-th slice.

The first step in the proposed framework corrects the contours' position, to remove misalignment errors (see Fig. 4.2 (b)), by performing a rigid pre-registration. This is achieved by simply aligning the centroids of all the slice contours in volume v.

The second step resamples all the training examples using a polynomial approximation scheme (see Fig. 4.2 (c)). This step is explained in the next section.

4.3.1 Resampling Shape Examples

This task aims to resample the shape examples in the training set so that they are all described by a pre-defined number of contours, S, regardless of the number of slices of the corresponding MR volume. This means that new contours, located at a generic position, s, along the LV axis, have to be determined. This corresponds to going from Fig. 4.2(b) to Fig. 4.2(c).

4.3.1.1 Approximation Model

In order to sample a new contour, denoted by $\widehat{\boldsymbol{x}}_v(s)$, at any position, $s \in [0,1]$, along the LV axis, we model it as a function of the position s, based on a combination of K polynomial basis functions, $\boldsymbol{\psi}(s) = \left[1, s, \ldots, s^{K-1}\right]^\top \in \mathbb{R}^{K\times 1}$,

$$\widehat{\boldsymbol{x}}_v(s) = \boldsymbol{C}_v \boldsymbol{\psi}(s), \tag{4.3}$$

where $\boldsymbol{C}_v \in \mathbb{R}^{2N\times K}$ is the coefficient matrix associated to volume v, defined by

$$\boldsymbol{C}_v = \begin{bmatrix} c^1_{1,1} & c^1_{1,2} & \cdots & c^1_{1,K} \\ c^1_{2,1} & c^1_{2,2} & \cdots & c^1_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ c^N_{1,1} & c^N_{1,2} & \cdots & c^N_{1,K} \\ c^N_{2,1} & c^N_{2,2} & \cdots & c^N_{2,K} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}^1_1 \\ \boldsymbol{c}^1_2 \\ \vdots \\ \boldsymbol{c}^N_1 \\ \boldsymbol{c}^N_2 \end{bmatrix}. \tag{4.4}$$

Notice that the coefficient matrix is specific for each MR volume, v, while the polynomial basis, $\boldsymbol{\psi}(s)$ depend only on the slice position, s. Using (4.3), any contour can be obtained for $s \in [0,1]$, as shown in Fig. 4.3, which means that shape examples can be resampled at a predefined number of slice positions.

In this work, the same model degree $K-1$ is used to resample all the shape examples in the training set, regardless of the number of slices of the corresponding volume, S^v. This means that in some volumes, where $S^v > K$, the blue lines shown in Fig. 4.3 will not go through the shape landmarks (shown in green), and only approximate the original contours. In other volumes, for which $S^v \leq K$, the blue lines go through the shape landmarks, and the resampling process is an interpolation of the contours.

4.3.1.2 Estimation of the Coefficient Matrix

The resampling process described above requires having previously computed the coefficient matrix, $\boldsymbol{C}_v$, of each volume in the training set. This section describes how these matrices are computed.

Let $\boldsymbol{X}_v \in \mathbb{R}^{2N\times S^v}$ be the concatenation of the S^v contours corresponding to the annotations of volume v that we wish to resample, defined by

$$\boldsymbol{X}_v = \left[\widetilde{\boldsymbol{x}}_v(s_1), \widetilde{\boldsymbol{x}}_v(s_2), \ldots, \widetilde{\boldsymbol{x}}_v(s_{S^v})\right], \tag{4.5}$$

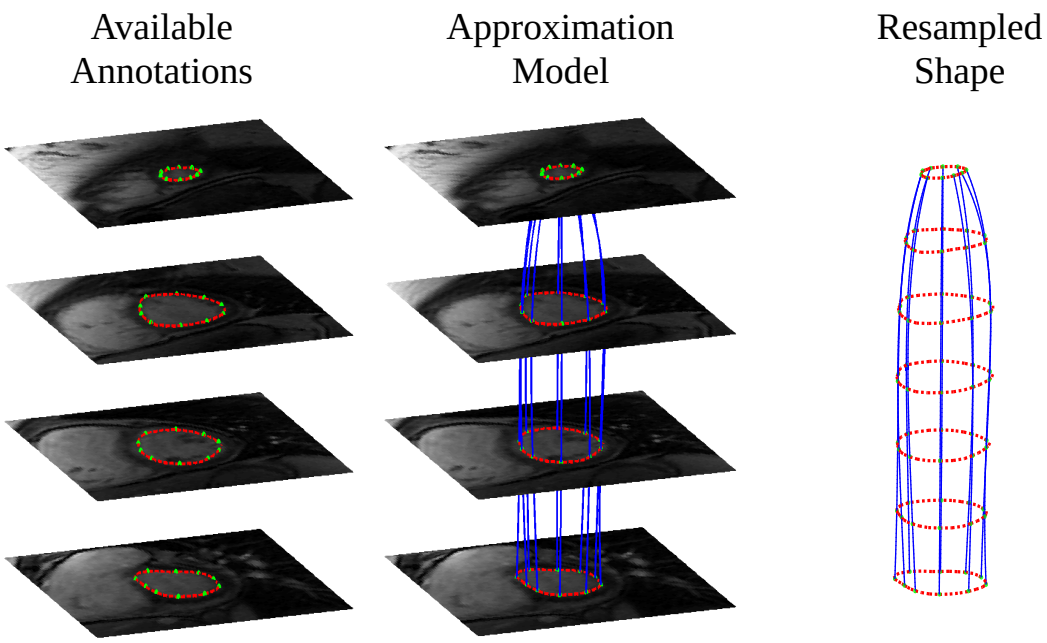


Figure 4.3: Illustration of the resampling process. The training example (left) consists of an MR volume with 4 slices and the corresponding contours (in red). The blue lines (mid and right) show the approximation model for a subset of the contour points (green dots), which allows any contour between the basal and apical slices to be obtained (right).

where each vector, $\widetilde{\boldsymbol{x}}_v(s_i)$, is a contour as defined in (4.2). Each pair of lines in $\boldsymbol{X}_v$, denoted herein by $\boldsymbol{X}_v^i \in \mathbb{R}^{2\times S^v}$, can be regarded as samples of the trajectory of the i-th contour point along the LV axis (see the green dots and the corresponding blue lines in Fig. 4.3). The trajectory can be written as

$$\boldsymbol{X}_v^i = \begin{bmatrix} x_1^i(s_1), \ldots, x_1^i(s_{S^v}) \\ x_2^i(s_1), \ldots, x_2^i(s_{S^v}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_{1v}^i \\ \boldsymbol{X}_{2v}^i \end{bmatrix}, \tag{4.6}$$

where each line, $\boldsymbol{X}_{jv}^i$, $j = 1, 2$, is associated with the coefficients, $\boldsymbol{c}_j^i$, of matrix $\boldsymbol{C}_v$ (recall (4.4)).

These trajectory samples are used to estimate $\boldsymbol{c}_j^i$ by solving the following ridge regression problem [67]

$$\boldsymbol{c}_j^i = \arg\min_{\boldsymbol{c}} \|{\boldsymbol{X}_{jv}^i}^\top - \boldsymbol{\Psi}\boldsymbol{c}^\top\|^2 + \gamma\|\boldsymbol{c}\|^2, \tag{4.7}$$

where $\boldsymbol{\Psi} = [\boldsymbol{\psi}(s_1), \ldots, \boldsymbol{\psi}(s_{S^v})]^\top \in \mathbb{R}^{S^v\times K}$ is the concatenation of the polynomial basis $\boldsymbol{\psi}(s_m), m = 1, \ldots, S^v$, and $\gamma \geq 0$ is a regularization constant. The solution of (4.7) is given by

$$\boldsymbol{c}_j^{i\top} = \left(\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \gamma\boldsymbol{I}\right)^{-1}\boldsymbol{\Psi}^\top\boldsymbol{X}_{jv}^{i\top}, \tag{4.8}$$

where $\boldsymbol{I}$ is the $K \times K$ identity matrix. This solution differs from the Ordinary Least Squares (OLS) solution due to the regularization term (assuming $\gamma > 0$), which guarantees

that the term $\left(\boldsymbol{\Psi}^\top\boldsymbol{\Psi}+\gamma\boldsymbol{I}\right)$ is invertible for any polynomial degree, $K \in \mathbb{N}$, used in the resampling process. The ridge regression was chosen, because we are using a fixed value for K, regardless of S^v, which would not be possible with the OLS solution (obtained by setting $\gamma = 0$) since it can only be computed for $K \leq S^v$.

The solution in (4.8) can be rewritten taking all the lines in $\boldsymbol{C}_v$ into account as follows

$$\boldsymbol{C}_v^\top = \left(\boldsymbol{\Psi}^\top\boldsymbol{\Psi}+\gamma\boldsymbol{I}\right)^{-1}\boldsymbol{\Psi}^\top\boldsymbol{X}_v^\top, \tag{4.9}$$

where $\boldsymbol{X}_v$ is given by (4.5). Once the coefficient matrix $\boldsymbol{C}_v$ has been computed, the approximation model defined in (4.3) can be used to obtain the contours at the desired positions along the LV axis. This is done for all the shape examples in the training set and guarantees that they all have the same number of landmarks.

4.3.2 Computation of the Shape Statistics

We propose to learn the shape statistics for each slice independently, using a learning procedure similar to the one described in Section 3.2.1, *i.e.*, computing the mean shape, $\overline{\boldsymbol{x}}(s_m)$, and the main modes of deformation, $\boldsymbol{D}(s_m)$, for each slice separately. This is not a typical 3D shape model [66], in which the surface of an object is described by a single vector and the shape statistics are computed for the whole surface. The proposed approach is similar to the strategy used in hierarchical shape models [11,100], which divide the whole shape into smaller patches and compute the shape statistics for each patch. This is advantageous because it reduces the dimensionality of the data, while still allowing for a 3D shape model to be used. In this case, each patch corresponds to the contour of a specific slice.

We propose to describe the 3D structure of the LV using a set of S 2D contours, such that each contour,

$$\widetilde{\boldsymbol{x}}(s_m) = \begin{bmatrix}\widetilde{\boldsymbol{x}}^1\\ \vdots \\ \widetilde{\boldsymbol{x}}^N\end{bmatrix}, \quad \widetilde{\boldsymbol{x}}^i = \begin{bmatrix}x_1^i\\ x_2^i\end{bmatrix}, \tag{4.10}$$

is given by

$$\widetilde{\boldsymbol{x}}(s_m) = \boldsymbol{T}_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}}(s_m) + \boldsymbol{D}(s_m)\boldsymbol{b}(s_m)\right), \tag{4.11}$$

where $\overline{\boldsymbol{x}}(s_m)$, $\boldsymbol{D}(s_m)$, and $\boldsymbol{b}(s_m)$ are the mean shape, deformation modes, and deformation coefficients, respectively, associated with the m-th slice, and $\boldsymbol{T}_{\boldsymbol{\theta}}(\cdot)$ is a similarity transformation. This transformation is assumed to scale, rotate, and translate the contours only in the axial plane, thus ensuring that the contours are always located on the volume slices (or on some parallel plane). It is also assumed that the transformation does not depend on the slice position, which means that the pose of the LV is fully described by a single

transformation applied to all the contours in the volume. The position of each contour point after applying the transformation is given by

$$\boldsymbol{T}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}^i(s_m)) = \boldsymbol{X}^i(s_m)\boldsymbol{a} + \boldsymbol{t}, \tag{4.12}$$

where

$$\boldsymbol{X}^i(s_m) = \begin{bmatrix} x_1^i(s_m) & -x_2^i(s_m) \\ x_2^i(s_m) & x_1^i(s_m) \end{bmatrix},$$

and $\boldsymbol{a} = [a_1, a_2]^\top$ defines the scale and rotation, and $\boldsymbol{t} = [t_1, t_2]^\top$ defines the translation on the axial plane.

The first step towards the computation of the shape statistics is to align all the shape examples in the training set by removing the pose of the LV. Consider a particular shape example defined by the resampled slice contours, $\widehat{\boldsymbol{x}}(s_m)$, with $s_m = \frac{m-1}{S-1}$ and $m = 1, \ldots, S$. This shape is aligned with a reference shape, $\boldsymbol{x}_{\text{ref}}$, by finding the parameters, $\boldsymbol{\theta}^\star = \{\boldsymbol{a}, \boldsymbol{t}\}$, of a global (pose) transformation, $\boldsymbol{T}_{\boldsymbol{\theta}}$, such that

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \sum_{m=1}^{S} \sum_{i=1}^{N} \left\| \boldsymbol{T}_{\boldsymbol{\theta}} \left(\widehat{\boldsymbol{x}}^i(s_m) \right) - \boldsymbol{x}_{\text{ref}}^i(s_m) \right\|^2, \tag{4.13}$$

where $\boldsymbol{T}_{\boldsymbol{\theta}} \left(\widehat{\boldsymbol{x}}^i(s_m) \right)$ is given by (4.12). This optimization problem is solved using a standard least squares approach, similar to the alignment algorithm described in Section 3.2.1.

After correcting the pose of all the shape examples in the training set, it is possible to compute the mean shape and the main modes of deformation. In Fig. 4.2, this corresponds to going from (c) to (d). The mean shape for each slice, denoted by $\overline{\boldsymbol{x}}(s_m)$, $m = 1, \ldots, S$, is computed as the average slice contour over all the models in the training set. The main modes of deformation and corresponding eigenvalues, denoted by

$$\boldsymbol{D}(s_m) = [\boldsymbol{d}_1(s_m), \ldots, \boldsymbol{d}_L(s_m)] \in \mathbb{R}^{2N \times L}, \tag{4.14}$$

$$\boldsymbol{\lambda}(s_m) = [\lambda_1(s_m), \ldots, \lambda_L(s_m)]^\top \in \mathbb{R}^{L \times 1}, \tag{4.15}$$

are obtained by performing PCA, where $\boldsymbol{d}_l(s_m) \in \mathbb{R}^{2N \times 1}$ and $\lambda_l(s_m) \in \mathbb{R}$ are the l-th main mode of deformation and corresponding eigenvalue, respectively, of the m-th slice, and $L \leq 2N$ is the number of main deformation modes that are used.

From this point on, the learned shape model is used to segment new MR volumes, in the test phase, which means estimating the model parameters, $\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}(s_1), \ldots, \boldsymbol{b}(s_S)$, that make the contours fit the LV border in each slice.

4.4 Test Phase

This section describes how the learned shape model (Fig. 4.2 (d)) is applied to a test volume (Fig. 4.2 (f)). The test volume may have a different number of slices, say

Figure 4.4: Example of an interpolated volume slice, at $s = \frac{s_4+s_5}{2}$, obtained by linear interpolation between two consecutive slices, s_4 and s_5.

S^t, which means that the position of the shape model contours may not coincide with the position of the volume slices. A simple approach would be to interpolate voxel intensities between the volume slices when searching for the LV border. However, this is not a good strategy due to the large distance between slices and to the existence of misaligned slices. Fig. 4.4 (middle) shows an example of an MR slice, located at $s = \frac{s_m+s_{m+1}}{2}$, obtained by linear interpolation of the two neighboring (misaligned) slices, s_m and s_{m+1}. The result is an image in which the LV borders are impossible to identify - besides being blurred, there appears to be two distinct borders, each created by one of the neighboring slices. Thus, in the proposed approach, instead of interpolating voxel intensities, the shape model is resampled with the same number of contours as the number of slices in the test volume, as described in the next section.

Besides this issue, it is also necessary to take into consideration the challenges posed by the existence of misaligned slices, and the accurately fitting the model to the LV border. The former is dealt with by adding two new deformation modes to each deformation matrix, $\boldsymbol{D}(s_m)$, that allow each contour to translate within the slice plane and fit misaligned slices. The latter problem is addressed by using an extension of the EM-RASM algorithm to 3D. These two topics are described in detail in Sections 4.4.2 and 4.4.3, respectively.

4.4.1 Resampling the Learned Shape Model

The mean shape of the learned shape model is resampled using the approach described in Section 4.3.1 to resample the shape examples. The contours of the mean shape at the new positions are obtained by: 1) computing the corresponding coefficient matrix, $\overline{\boldsymbol{C}}$, using (4.9); and 2) computing the new contours using (4.3).

Regarding the modes of deformation, resampling is not a trivial task. A different resampling strategy is proposed for the modes of deformation Suppose that a test slice is located at the position s between the contours of the learned shape model located at s_m and s_{m+1}, *i.e.*, $s \in [s_m, s_{m+1}]$. The goal is to estimate the modes of deformation of the shape model at the new location s. However, since the shape model is learned

independently for each slice, the correspondences between the modes of deformation of different slices also have to be determined.

Let $\boldsymbol{D}(s) = [\boldsymbol{d}_1(s), \ldots, \boldsymbol{d}_L(s)]$ be the deformation modes (to be estimated) of the shape model contour located at $s \in [s_m, s_{m+1}]$. Also, let $\alpha \in [0, 1]$ be the relative distance between s and s_m,

$$\alpha = \frac{s - s_m}{s_{m+1} - s_m}. \tag{4.16}$$

Without loss of generality, assume that s_m is the closest slice (*i.e.*, $\alpha \leq 0.5$). The l-th deformation mode and corresponding eigenvalue are obtained by

$$\boldsymbol{d}_l(s) = (1 - \alpha)\boldsymbol{d}_l(s_m) + \alpha \boldsymbol{d}_{F(l)}(s_{m+1}), \tag{4.17}$$

$$\lambda_l(s) = (1 - \alpha)\lambda_l(s_m) + \alpha \lambda_{F(l)}(s_{m+1}), \tag{4.18}$$

where

$$F(l) = \arg\min_n \|\boldsymbol{d}_l(s_m) - \boldsymbol{d}_n(s_{m+1})\| \tag{4.19}$$

defines the correspondence between the deformation modes in s_m and s_{m+1}. These equations may be interpreted as computing the new l-th deformation mode by linearly interpolating between the l-th deformation mode in s_m and the most similar deformation mode in s_{m+1}.

This interpolation process is repeated for all the deformation modes, $l = 1, \ldots, L$, and all the S^t new slice positions. Once this is done, the new shape model has the same number of contours as the number of slices in the test volume.

4.4.2 Deformation Modes for Fitting Misaligned Slices

The shape model described in the previous sections is defined by a global transformation, $\boldsymbol{\theta} = \{\boldsymbol{a}, \boldsymbol{t}\}$, that scales, rotates and translates all the slice contours equally, and by the deformation coefficients for each slice, $\boldsymbol{b}(s_m)$. However, at this point the learned shape model is not able to fit the LV border in all the slices if the test volume has misaligned slices, since it is not capable of translating each slice contour individually.

For the shape model to fit the LV border in a misaligned slice, two extra modes of deformation are added to each matrix of deformation that account for small translations within the slice plane. Formally, we define a new matrix

$$\boldsymbol{D}(s_m) = [\boldsymbol{d}_1(s_m), \ldots, \boldsymbol{d}_L(s_m), \boldsymbol{d}_{t_1}(s_m), \boldsymbol{d}_{t_2}(s_m)], \tag{4.20}$$

where $\boldsymbol{d}_{t_1}(s_m), \boldsymbol{d}_{t_2}(s_m) \in \mathbb{R}^{2N \times 1}$ are defined by

$$\boldsymbol{d}_{t_1}(s_m) = \frac{1}{\sqrt{N}} [1, 0, 1, 0, \ldots, 1, 0]^\top \tag{4.21}$$

$$\boldsymbol{d}_{t_2}(s_m) = \frac{1}{\sqrt{N}} [0, 1, 0, 1, \ldots, 0, 1]^\top. \tag{4.22}$$

The additional deformation modes, $\boldsymbol{d}_{t_1}(s_m)$ and $\boldsymbol{d}_{t_2}(s_m)$, apply a translation along the first and second dimension of the slice, respectively. The corresponding eigenvalues for these two modes of deformation, $\lambda_{t_1}(s_m), \lambda_{t_2}(s_m)$, determine the variance of the two translational deformation modes. This variance will influence the extent of translational deformation that is allowed, as will be explained in the following section. The deformation coefficients, $\boldsymbol{b}(s_m)$, also have two additional coefficients (a total of $L+2$ coefficients) that determine the extent of the translation in each direction. For example, if the last coefficient of $\boldsymbol{b}(s_m)$ is $b_{L+2}(s_m)$, then, recalling (4.11), the mean shape in the m-th slice, $\overline{\boldsymbol{x}}(s_m)$, is translated by $\frac{b_{L+2}(s_m)}{\sqrt{N}}$ pixels along the second dimension.

4.4.3 Extension of EM-RASM to 3D Data

The segmentation process described in this section revises the EM-RASM algorithm proposed in Chapter 3 and extends it to the 3D case. The difference between the method described here in comparison with the EM-RASM is that this new formulation uses a 3D shape model that aims to fit the LV border in all the slices simultaneously.

Formally, given a test volume, the segmentation of the LV, $\widetilde{\boldsymbol{x}}(s_m)$, $m = 1, \ldots, S^t$, is obtained by determining the global transformation parameters, $\boldsymbol{\theta} = \{\boldsymbol{a}, \boldsymbol{t}\}$, and deformation coefficients, $\boldsymbol{b}(s_m)$, such that $\widetilde{\boldsymbol{x}}(s_m)$ fits the LV border in all slices.

Given an initial estimate of the model parameters for a test volume, a set of edge points are extracted from each volume slice by searching along lines orthogonal to the contours, using the method described in Section 3.6.3. This typically leads to the detection of several edge points, of which only one may correspond to the LV border. Therefore, an outlier model is required to accurately estimate the model parameters.

Let $\boldsymbol{Y}^i(s_m) = \{\boldsymbol{y}^{ij}(s_m), j = 1, \ldots, M^i\}$ be the set of detected edge points, associated with the i-th model point $\widetilde{\boldsymbol{x}}^i(s_m)$, on the m-th slice. Since we do not know which edge points are valid or invalid (outliers), two observation models are considered. A binary label, $k^{ij}(s_m) = 1$, is assigned to $\boldsymbol{y}^{ij}(s_m)$ if it is considered valid; on the other hand, if $\boldsymbol{y}^{ij}(s_m)$ is invalid, then it will be assigned the label $k^{ij}(s_m) = 0$. These two labels occur with probabilities $P(k^{ij}(s_m) = 1) = p_1$ and $P(k^{ij}(s_m) = 0) = p_0$, respectively, with $p_0 + p_1 = 1$.

When an edge point $\boldsymbol{y}^{ij}(s_m)$ is valid, the following observation model is assumed to be active

$$\boldsymbol{y}^{ij}(s_m) = \widetilde{\boldsymbol{x}}^i(s_m) + \boldsymbol{v}^i(s_m), \tag{4.23}$$

where $\boldsymbol{v}^i(s_m) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^i(s_m))$ is a zero mean white Gaussian noise with diagonal covariance matrix $\boldsymbol{\Sigma}^i(s_m)$. It follows that the probability of the $\boldsymbol{y}^{ij}(s_m)$ is given by

$$p\left(\boldsymbol{y}^{ij}(s_m) \middle| k^{ij}(s_m)=1, \boldsymbol{\Theta}\right) = \mathcal{N}\left(\boldsymbol{y}^{ij}(s_m); \widetilde{\boldsymbol{x}}^{ij}(s_m), \boldsymbol{\Sigma}^{ij}(s_m)\right), \tag{4.24}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}(s_1), \ldots, \boldsymbol{b}(s_{S^t}), p_0, p_1\}$ is the set of all the model parameters. If $\boldsymbol{y}^{ij}(s_m)$ is considered an outlier (*i.e.*, $k^i(s_m) = 0$), then we assume that it follows a uniform

distribution

$$p\left(\boldsymbol{y}^{ij}(s_m)\middle|k^{ij}(s_m)=0,\boldsymbol{\Theta}\right)=\mathcal{U}(V_{\widetilde{\boldsymbol{x}}^i(s_m)}), \tag{4.25}$$

within a validation gate $V_{\widetilde{\boldsymbol{x}}^i(s_m)}$ in the vicinity of $\widetilde{\boldsymbol{x}}^i(s_m)$. This formulation is similar to the formulation described in Chapter 3, but now most of the variables depend on the slice position s_m, whereas previously each MR image was analyzed in the same way, regardless of the slice position.

Let $\boldsymbol{\mathcal{Y}}=\{\boldsymbol{Y}(s_1),\ldots,\boldsymbol{Y}(s_{S^t})\}$ be the set of the detected edge points in all the slices of the test volume, and $\boldsymbol{\mathcal{K}}=\{\boldsymbol{K}(s_1),\ldots,\boldsymbol{K}(s_{S^t})\}$ be the set of the corresponding labels, where $\boldsymbol{Y}(s_m)=\{\boldsymbol{y}^{ij}\}$ and $\boldsymbol{K}(s_m)=\{k^{ij}(s_m)\}$ are the set of edge points and corresponding labels in the m-th slice. The proposed algorithm aims to find the parameters, $\boldsymbol{\Theta}$, that maximize the log-joint probability distribution

$$\begin{aligned}\mathcal{P}(\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{K}},\boldsymbol{\Theta}) &= \log p(\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{K}},\boldsymbol{\Theta})\\ &= \log p\left(\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{K}}|\boldsymbol{\Theta}\right)+\log p(\boldsymbol{\Theta})\\ &= \log p\left(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{K}},\boldsymbol{\Theta}\right)+\log p(\boldsymbol{\mathcal{K}})+\log p(\boldsymbol{\Theta}).\end{aligned} \tag{4.26}$$

Assuming conditional independence between edge points, the equation above can be factorized, leading to

$$\mathcal{P}(\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{K}},\boldsymbol{\Theta})=\sum_{m=1}^{S^t}\sum_{i=1}^{N}\sum_{j=1}^{M^i}\log p\left(\boldsymbol{y}^{ij}(s_m)\middle|k^{ij}(s_m),\boldsymbol{\Theta}\right)+\log p(k^{ij}(s_m))+\log p(\boldsymbol{\Theta}). \tag{4.27}$$

The first term corresponds to the likelihood of the edge points, the second term is the probability of the observation model, k^{ij}, and the last term corresponds to the priors of the parameters $\boldsymbol{\Theta}$. The prior probability distribution for the deformation coefficients is assumed to be given by

$$p\Big(\boldsymbol{b}(s_1),\ldots,\boldsymbol{b}(s_{S^t})\Big)=\prod_{m=1}^{S^t}p\Big(\boldsymbol{b}(s_m)\Big), \tag{4.28}$$

where, as in the previous chapter,

$$p\Big(\boldsymbol{b}(s_m)\Big)=\mathcal{N}\left(\boldsymbol{b}(s_m);\boldsymbol{0},\boldsymbol{\Sigma}^{\boldsymbol{b}(s_m)}\right), \tag{4.29}$$

On the other hand, the prior probabilities for the other parameters are given by the Jeffreys' prior, *i.e.*, they do not depend on any of the model parameters, $\boldsymbol{\Theta}$, and hence do not influence the segmentation (recall Section 3.4.2.1).

Since it is infeasible to maximize (4.27) analytically, we use the MAP formulation of the EM algorithm as follows.

4.4.3.1 E-step

Let $\widehat{\boldsymbol{\Theta}}_{(t)}$ denote the current estimate of the model parameters ($\widehat{\boldsymbol{\Theta}}_{(0)}$ is the initial guess). At each iteration, a new set of edge points, $\boldsymbol{\mathcal{Y}}$, is extracted from the test volume. The EM algorithm is based on the auxiliary function

$$Q\left(\boldsymbol{\Theta};\widehat{\boldsymbol{\Theta}}_{(t)}\right)=\mathbb{E}_{\boldsymbol{\mathcal{K}}}\left[\mathcal{P}(\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{K}},\boldsymbol{\Theta})\middle|\boldsymbol{\mathcal{Y}},\widehat{\boldsymbol{\Theta}}_{(t)}\right], \tag{4.30}$$

where $\mathbb{E}_{\boldsymbol{\mathcal{K}}}[\cdot]$ denotes the expectation over the two observation models. This leads to

$$\begin{aligned}Q\left(\boldsymbol{\Theta};\widehat{\boldsymbol{\Theta}}_{(t)}\right)=\sum_{m=1}^{S^t}\sum_{i=1}^{N}\sum_{i=1}^{M^i} & w_0^{ij}(s_m)\left[\log p\big(\boldsymbol{y}^{ij}(s_m)\big|k^{ij}(s_m)=0,\boldsymbol{\Theta}\big)+\log p_0\right]\\ &+w_1^{ij}(s_m)\left[\log p\left(\boldsymbol{y}^{ij}(s_m)\middle|k^{ij}(s_m)=1,\boldsymbol{\Theta}\right)+\log p_1\right]\\ &+\log p(\boldsymbol{\Theta}),\end{aligned} \tag{4.31}$$

where w_0^{ij} and w_1^{ij} denote the confidence degree of each stroke, such that $w_0^{ij}(s_m)+w_1^{ij}(s_m)=1$ and

$$\begin{aligned}w_1^{ij}(s_m)&=p\left(k^{ij}(s_m)=1\middle|\boldsymbol{y}^{ij}(s_m),\widehat{\boldsymbol{\Theta}}_{(t)}\right)\\ &\propto p\left(\boldsymbol{y}^{ij}(s_m)\middle|k^{ij}(s_m)=1,\widehat{\boldsymbol{\Theta}}_{(t)}\right)p\left(k^{ij}(s_m)=1\middle|\widehat{\boldsymbol{\Theta}}_{(t)}\right)\\ &\propto \widehat{p}_{1(t)}\mathcal{N}\left(\boldsymbol{y}^{ij}(s_m);\widetilde{\boldsymbol{x}}^i(s_m),\boldsymbol{\Sigma}^i(s_m)\right)\end{aligned} \tag{4.32}$$

$$\begin{aligned}w_0^{ij}(s_m)&=p\left(k^{ij}(s_m)=0\middle|\boldsymbol{y}^{ij}(s_m),\widehat{\boldsymbol{\Theta}}_{(t)}\right)\\ &\propto \widehat{p}_{0(t)}\mathcal{U}\left(V_{\widetilde{\boldsymbol{x}}^i(s_m)}\right).\end{aligned} \tag{4.33}$$

These weights correspond to the probability of $k_1^{ij}(s_m)$ and $k_0^{ij}(s_m)$ being the correct label for $\boldsymbol{y}^{ij}(s_m)$, given the current model estimate.

Combining (4.31) with (4.24)-(4.25) and (4.28) yields

$$\begin{aligned}Q\left(\boldsymbol{\Theta};\widehat{\boldsymbol{\Theta}}_{(t)}\right)=c-\frac{1}{2}\sum_{m=1}^{S^t}\Bigg(&\boldsymbol{b}^\top(s_m)\boldsymbol{\Sigma}^{\boldsymbol{b}(s_m)}\boldsymbol{b}(s_m)-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M^i}w_0^{ij}(s_m)\log p_0+\\ &+w_1^{ij}(s_m)\Big(\boldsymbol{e}^{ij\top}(s_m)\boldsymbol{\Sigma}^{i-1}(s_m)\boldsymbol{e}^{ij}(s_m)+\log p_1\Big)\Bigg),\end{aligned} \tag{4.34}$$

where

$$\boldsymbol{e}^{ij}(s_m)=\boldsymbol{y}^{ij}(s_m)-\boldsymbol{A}\Big(\overline{\boldsymbol{x}}^i(s_m)+\boldsymbol{D}^i(s_m)\boldsymbol{b}(s_m)\Big)-\boldsymbol{t}, \tag{4.35}$$

and c is a constant that does not depend on the model parameters, $\boldsymbol{\Theta}$.

4.4.3.2 M-step

In this step, the model parameters are updated by maximizing the auxiliary function (4.34). We simplify this step by maximizing first with respect to the transformation parameters, $\boldsymbol{a},\boldsymbol{t}$, and only then for $\boldsymbol{b}(s_1),\ldots,\boldsymbol{b}(s_{S^v})$ and p_0 and p_1.

Update of the Transformation Parameters Regarding the transformation parameters, the update equation is obtained by taking the derivative of (4.34) with respect to $\boldsymbol{a}$ and $\boldsymbol{t}$, which leads to the following linear system,

$$\sum_{m=1}^{S^t}\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(s_m)\begin{bmatrix} \boldsymbol{X}^{i\top}(s_m)\boldsymbol{\Sigma}^{i-1}(s_m)\boldsymbol{X}^i(s_m) & \boldsymbol{X}^{i\top}(s_m)\boldsymbol{\Sigma}^{i-1}(s_m) \\ \boldsymbol{\Sigma}^{i-1}(s_m)\boldsymbol{X}^i(s_m) & \boldsymbol{\Sigma}^{i-1}(s_m)\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{a}}_{(t+1)} \\ \widehat{\boldsymbol{t}}_{(t+1)}\end{bmatrix} = \\ = \sum_{m=1}^{S^t}\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(s_m)\begin{bmatrix}\boldsymbol{X}^{i\top}(s_m)\boldsymbol{\Sigma}^{i-1}(s_m)\boldsymbol{y}^{ij}(s_m) \\ \boldsymbol{\Sigma}^{i-1}(s_m)\boldsymbol{y}^{ij}(s_m)\end{bmatrix} \quad (4.36)$$

where

$$\boldsymbol{X}^i(s_m) = \begin{bmatrix} x_1^i(s_m) & -x_2^i(s_m) \\ x_2^i(s_m) & x_1^i(s_m)\end{bmatrix},$$

$$\begin{bmatrix} x_1^i(s_m) \\ x_2^i(s_m)\end{bmatrix} = \boldsymbol{x}^i(s_m) = \overline{\boldsymbol{x}}^i(s_m) + \boldsymbol{D}^i(s_m)\widehat{\boldsymbol{b}}_{(t)}(s_m),$$

and $\boldsymbol{D}^i(s_m)$ are the lines of $\boldsymbol{D}(s_m)$ associated with the model point $\boldsymbol{x}^i(s_m)$. The solution of (4.36) corresponds to a weighted least squares solution to the problem of aligning the edge points extracted from the volume and the corresponding model points. The weights, $w_1^{ij}(s_m)$, mean that each edge point contributes to the estimation of the parameters based on their probability of being valid.

Update of the Deformation Coefficients The update of the deformation parameters, $\widehat{\boldsymbol{b}}_{(t+1)}(s_m)$, is obtained by taking the derivative of (4.34) with respect to $\boldsymbol{b}(s_m)$, for each slice, $m = 1, \ldots, S^t$, leading to

$$\left[\left(\boldsymbol{\Sigma}^{\boldsymbol{b}(s_m)}\right)^{-1} + \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(s_m)\boldsymbol{D}^{i\top}(s_m)\widehat{\boldsymbol{A}}_{(t+1)}^{\top}\boldsymbol{\Sigma}^{i-1}(s_m)\widehat{\boldsymbol{A}}_{(t+1)}\boldsymbol{D}^i(s_m)\right]\left[\widehat{\boldsymbol{b}}_{(t+1)}(s_m)\right] = \\ = \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(s_m)\left[\boldsymbol{D}^{i\top}(s_m)\widehat{\boldsymbol{A}}_{(t+1)}^{\top}\boldsymbol{\Sigma}^{i-1}(s_m)\left(\boldsymbol{y}^{ij}(s_m) - \widehat{\boldsymbol{A}}_{(t+1)}\overline{\boldsymbol{x}}^i(s_m) - \widehat{\boldsymbol{t}}_{(t+1)}\right)\right] \quad (4.37)$$

where

$$\widehat{\boldsymbol{A}}_{(t+1)} = \begin{bmatrix}\widehat{a}_{1(t+1)} & -\widehat{a}_{2(t+1)} \\ \widehat{a}_{2(t+1)} & \widehat{a}_{1(t+1)}\end{bmatrix}.$$

The term $\boldsymbol{\Sigma}^{\boldsymbol{b}(s_m)}$ comes from the prior distribution of the deformation coefficients, given by (4.28).

Update of the Model Probabilities Finally, the observation model probabilities, $\widehat{p}_{0(t)}$, $\widehat{p}_{1(t)}$, are updated by taking the derivative of (4.34) with respect to $\widehat{p}_{1(t)}$ (recall that

$\widehat{p}_{0(t)} = 1 - \widehat{p}_{1(t)}$). This leads to the following update equations

$$\widehat{p}_{1(t+1)} = \frac{\sum_{m=1}^{S^t} \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij}(s_m)}{\sum_{m=1}^{S^t} \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij}(s_m) + w_0^{ij}(s_m)} \tag{4.38}$$

$$\widehat{p}_{0(t+1)} = 1 - \widehat{p}_{0(t+1)}. \tag{4.39}$$

The difference between the 3D formulation of EM-RASM proposed in this chapter and the 2D formulation proposed in Chapter 3 comes from the fact that, in the 3D setting, the edge points from the whole MR volume will contribute to the estimation of the transformation parameters $\boldsymbol{a}$ and $\boldsymbol{t}$. This makes the estimation more robust to slices in which the LV border is not so easy to identify, because the contribution of the other slices in the volume can compensate for this issue.

4.5 Experimental Setup

This section describes the experimental setup used to evaluate the proposed method. The evaluation is based on the public database described in Section 3.6.1, which consists of 33 sequences of volumes. The annotations provided in the database (*i.e.*, the manual segmentations of the LV) were used in the training phase, to learn the shape statistics, and in the test phase, as GT for evaluation purposes. This was done using a leave-one-sequence-out cross validation, *i.e.*, for each test sequence, the shape model was learned using the shape examples from the other 32 sequences. The training shapes were resampled using the method described in Section 4.3, with the following parameters: number of points in each slice contour, $N = 40$; number of slices per volume, $S = 8$; number of basis functions, $K = 10$; and regularization constant, $\gamma = 10^{-4}$. The choice of these last two parameters was based on a number of tests comparing the performance of the resampling methodology for different values of K and γ (see Section 4.6.1). The shape model was learned using the approach described in Section 4.3.2, and the number of deformation coefficients used in each contour of the shape model was $L = 10$ (or $L = 12$, considering the two additional deformation modes for misalignments).

The initial guess of the model parameters is as follows. The transformation parameters $\widehat{a}_{1(0)}$ and $\widehat{\boldsymbol{t}}_{(0)}$ are initialized by human input of the rough location of the LV center and its radius in the basal slice. The remaining parameters are initialized by the default values: $\widehat{a}_{2(0)} = 0$, $\widehat{\boldsymbol{b}}_{(0)}(s_m) = \mathbf{0}$, $m = 1, \ldots, S^t$ and $\widehat{p}_{0(0)} = \widehat{p}_{1(0)} = 0.5$.

The quantitative evaluation of the results was based on the Dice coefficient, d_{Dice}, the average perpendicular distance, d_{AV}, the volume error, d_{VD}, and the percentage of good segmentations, all of which are described in Section 3.6.2.

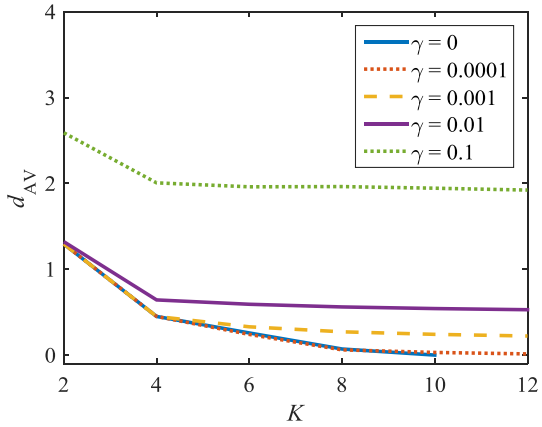


Figure 4.5: Average distance between the annotations and the approximation model for different values of K and γ.

4.6 Results

This section is divided in two subsections. The first shows the performance of the resampling strategy described in 4.3 on the training data, in order to determine the best parameters. The second shows the performance of the proposed method in the estimation of CMR data.

4.6.1 Accuracy of the Approximation Model

The performance of the resampling scheme described in Section 4.3 depends on two main parameters: i) the number of polynomial basis functions, K, and ii) the regularization constant, γ, used to estimate the coefficient matrix in (4.4). The first parameter is related to the degree of the polynomial used to model the trajectory of a contour point along the slices. There are two possible scenarios: i) K is smaller than the number of slices in the volume, S^v, in which case the polynomial may not able to accurately describe the trajectories of the contour points; and ii) $K \geq S^v$, in which case the accuracy of the approximation model is only limited by the regularization term. On the other hand, the regularization parameter, γ, is responsible for keeping the elements of the coefficient matrix small, which keeps the trajectory estimates smooth.

The evaluation was performed by comparing the original shape examples in the training set and their interpolated counterpart (with the same number of contours), using on the error metric d_{AV}. The idea is that the interpolated model should be as similar to the original shape example as possible. Fig. 4.5 shows how the d_{AV} error metric varies for different values of $K = \{2, 4, 6, 8, 10, 12\}$ and $\gamma = \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The figure shows that higher values of K lead to better results. It is possible to see that for each curve there is a value of K beyond which the accuracy improvement is not significant. As for the regularization constant, it is also possible to see that smaller values lead to better

results, as is expected. However, it is important to note that for $\gamma = 0$, the computation of the coefficient matrix has numerical instability for $K > S^v$, because $\mathbf{\Psi}^\top \mathbf{\Psi}$ becomes non-invertible[1] (recall equation (4.9) in Section 4.3.1.2). Since $S^v \leq 10$ for all the volumes in the dataset, it was not possible to compute any coefficient matrix for $K = 12$. Based on these results, the choice of parameter values to use in the following section was $K = 10$ and $\gamma = 10^{-4}$.

4.6.2 LV Segmentation

This section presents the results of the LV segmentation using the proposed approach. For comparison purposes, the proposed method was evaluated using two configurations: i) with and ii) without translational deformation modes (TDM) to compensate for misaligned slices. (recall Section 4.4.2). In this way, we are able to assess the benefits of introducing these additional modes of deformation.

Table 4.1 summarizes the results. Note that the approach that does not use TDM is not able to fit misaligned slices, and it is possible to see that this has a significant influence on the performance of the algorithm. The table shows that the addition of these deformation modes improves the segmentation accuracy by approximately 6%. The percentage of good segmentations is also representative of the amount of gross errors obtained by the approach without TDM, in which almost 25% of the segmentations have $d_{\text{AV}} > 5$ mm. Examples of the improvements achieved with the use of TDM are shown in Fig. 4.6. These examples clearly show that the correct segmentation is severely affected by slice misalignments. The segmentations obtained without using TDM are not capable of fitting the shifted slices, leading to poor results in which the model fits the LV border in some slices but fails in others. In the configuration that uses TDM, the model is able to follow the slice misalignments (except for one of the slices in example 1) and achieve accurate segmentation results.

Comparing the results of the 2D approach from the previous chapter with the 3D method proposed in this chapter, it is possible to see that the 3D modality achieves better results, particularly in the volume error measured by d_{VD}. Another interesting conclusion is that the variance of the results is also reduced using the 3D approach, meaning that the precision of the segmentations is higher. This is expected since the 3D shape model prevents large deviations from the mean three-dimensional shape of the LV, thus reducing gross errors. On the other hand, the small difference in the overall results may be a consequence of the 3D shape model being too restrictive and preventing the model contours from accurately fitting the LV border in all the slices. For further comparative results, Fig. 4.7 shows the average Dice coefficient for each volume in the dataset. Each image is a 33×20 matrix in which the color of each pixel is based on the average Dice coefficient

[1] In such cases, the d_{AV} was not computed and the volume was removed from the results.

Table 4.1: Statistical performance of the 3D EM-RASM with and without TDM and comparison with the 2D EM-RASM.

		$d_{\mathbf{Dice}}$ (%)	$d_{\mathbf{AV}}$ (mm)	$d_{\mathbf{VD}}$ (mL)	% Good
2D		84.2 (9.2)	2.6 (1.7)	11.8 (11.4)	**89.2**
3D	**without TDM**	78.3 (8.3)	3.8 (2.3)	10.6 (19.7)	75.3
	with TDM	**84.6 (6.8)**	**2.5 (1.2)**	**9.2 (9.4)**	89.1

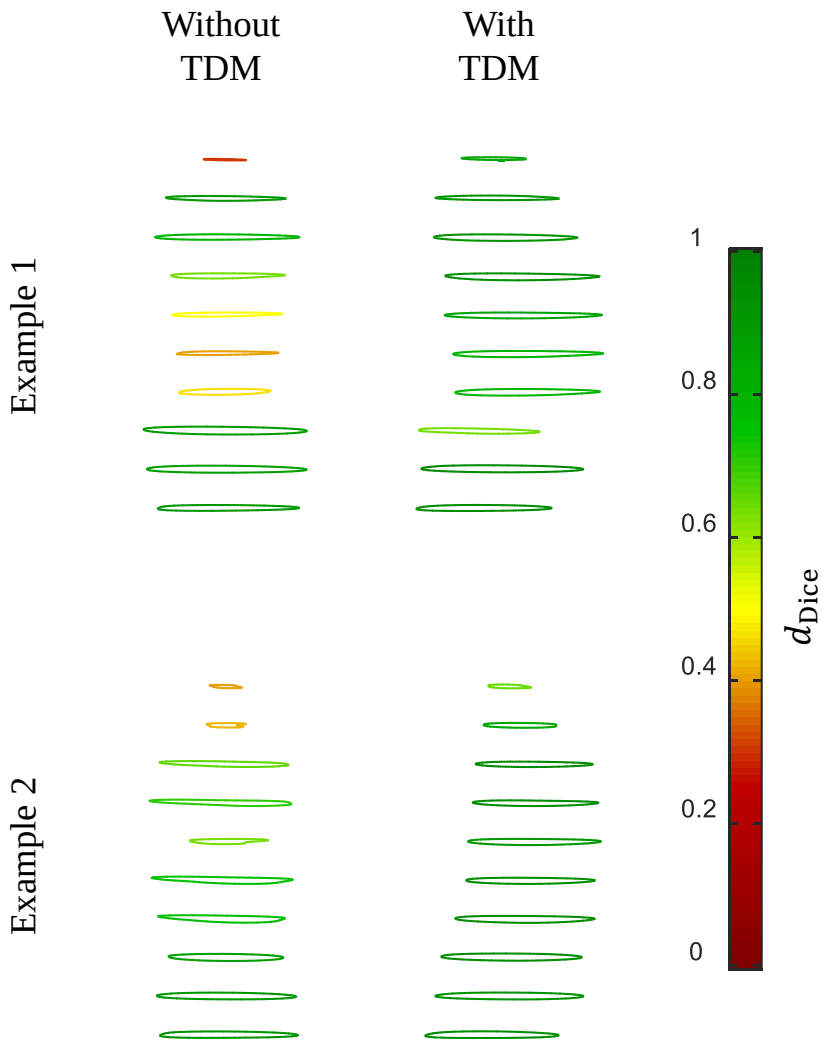


Figure 4.6: Two examples of the segmentations obtained with and without TDM. The colors are defined by the Dice coefficient based on colorbar on the right (greener is better).

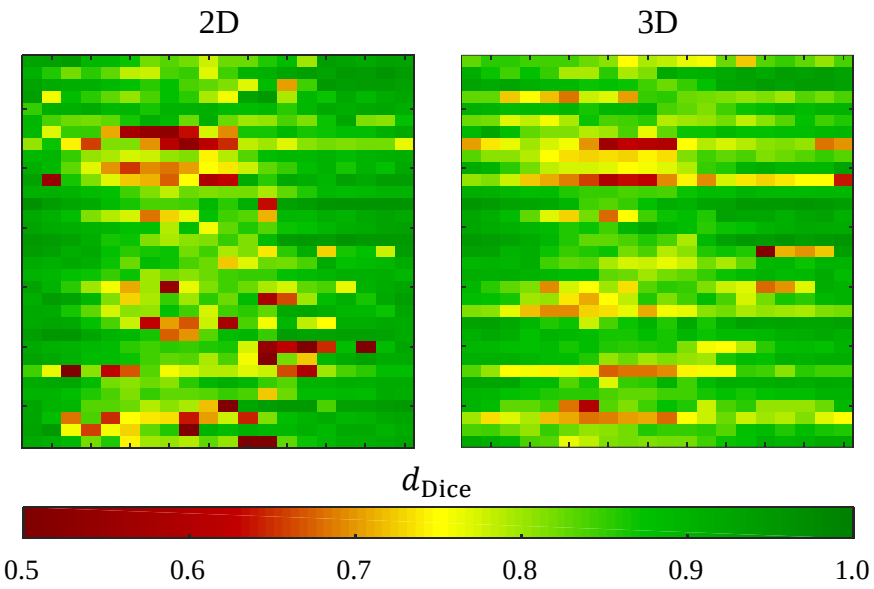


Figure 4.7: Quantitative evaluation of all the volumes using the Dice coefficient. The vertical axis corresponds to different patients and the horizontal axis to different phases of the cardiac cycle. The colors are defined by the colorbar in the bottom (greener is better).

of the volume segmentation for that particular patient and frame of the cardiac cycle. An analysis on the overall results leads to the conclusion that the 3D setting is able to prevent many of the gross errors of the 2D formulation (compare the amount of red pixels in both images). However, many pixels in the 3D image are yellow (*i.e.*, with an average Dice coefficient of approximately $0.7 < d_{\text{Dice}} < 0.8$), which did not happen much in the 2D formulation. This is in accordance with the analysis of the statistical results, which show that the 3D approach is better at avoiding gross errors, but is too restrictive to allow accurate segmentations of the remaining slices.

Fig. 4.8 shows other examples of the segmentations obtained using the proposed method. It is possible to see that, in almost all the cases, the obtained segmentation is very similar to the ground truth. The last row shows an example where the proposed segmentation was slightly hampered by the papillary muscles (second and third columns).

4.7 Conclusions

This chapter addresses the segmentation of the LV in CMR as a 3D segmentation problem. The proposed method is based on a 3D shape model that is able to deal with the two major difficulties associated with CMR: 1) the variable number of slices in a volume, and 2) misalignment between slices.

In the training phase, the proposed framework deals with the variability in the number of slices by resampling all the shape examples in the training set to a predefined number of contours. This overcomes the issue of learning a shape model from examples

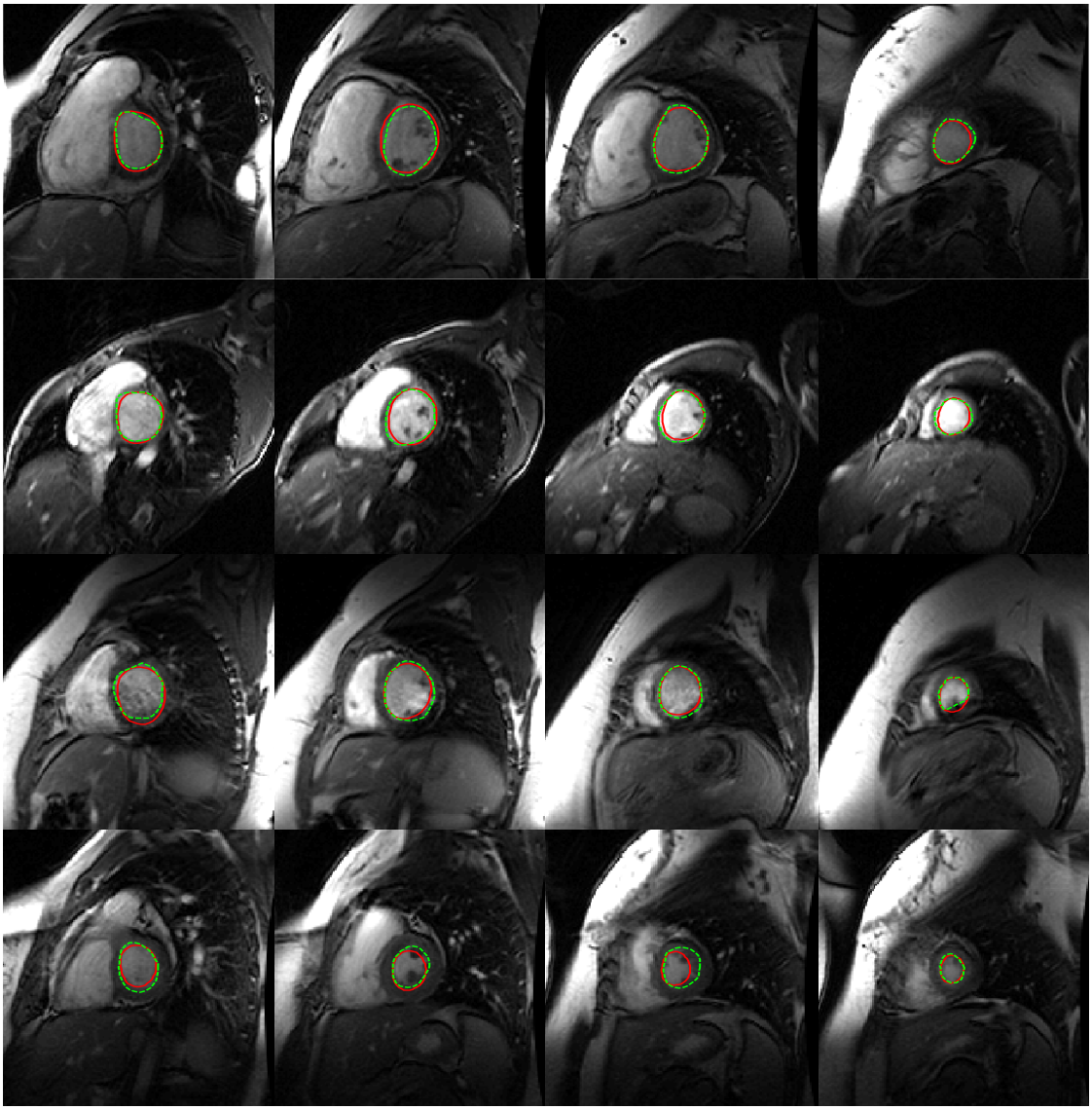


Figure 4.8: Examples of the LV segmentation using a 3D shape model. Each row shows four slices of a different volume depicting, the segmentation obtained using the proposed method (red), and the ground truth (green).

with a variable number of landmarks. For the segmentation of test volumes, the proposed algorithm avoids voxel intensity interpolation by resampling the mean shape and the main modes of deformation instead. This allows the 3D shape model to be used in any test volume, regardless of the number of slices it has.

Regarding the misaligned slices, the approach described in this chapter proposes to deal with this issue during the training phase by applying a pre-registration of the shape examples, in order to avoid shape artifacts caused by the misalignments. In the test phase, the 3D shape model is given the ability to fit misaligned slices through the introduction of two additional deformation modes for each contour, which enables the model to shift the contours within the slice plane. The results show that there is a significant advantage in using these new deformation modes. Without them, the model is unable to accurately fit

the LV border in many volumes, consequently leading to a large number of gross errors.

The statistical results shows that this proposed approach, which relies on an extension of EM-RASM to 3D data, is able to achieve good segmentation results, and is able to outperform the 2D setting described in the previous chapter. However, there are still examples in which the algorithm is not able to obtained accurate segmentations of the LV, particularly in the systolic phase of the sequence. This suggests that only by using temporal information will the segmentation algorithm take advantage of all the information available and achieve accurate segmentations. This is explored in the next chapter of this thesis.

5

Segmentation Using Temporal Information

Contents

The two previous chapters have addressed the segmentation of the LV using information extracted from: 1) individual MR slices (2D), and 2) the whole MR volume (3D), respectively. The next step considered in this thesis consists of integrating information about the LV temporal dynamics in the segmentation algorithm. This information has potential to improve the segmentation results by imposing constraints on how segmentations differ from one frame to another. This is particularly useful when the LV is not so clearly visible, or when the detected observations are not trust worthy, because it allows the model to disregard the observations extracted from the image and rely on prior knowledge about the LV dynamics. However, the integration of dynamical information raises two questions: 1) how to capture prior knowledge of the temporal variation of the LV; and 2) how to use this information in the segmentation algorithm. These two topics are the main focus of this chapter.

5.1 Introduction

A typical reasoning for a segmentation problem that is extended to the temporal dimension is that is becomes a tracking problem. In general terms, the goal of a tracking algorithm is to determine the position of a target object throughout time, *e.g.*, for surveillance purposes [19]. For most applications, the standard framework is a sequential alternation between receiving a new frame and updating the position of the target object in that frame. This can be viewed as an online segmentation scheme, in which the contour of the object is updated every time a new input is provided.

In medical applications, tracking algorithms have been popularly used in several applications, such as echocardiography [58, 122], microscopy [48, 116], as well as CMR [168, 190]. However, in CMR segmentation, there is no need for online algorithms, since the segmentation is only performed after the acquisition has been completed and the whole sequence is available.

Alternatively, it is possible to use temporal information as prior knowledge of the relationship between contours in different frames of an MR sequence. We know, for instance, that the LV is smaller in the ES phase (after heart contraction) than in the ED phase (after heart dilation), or that the segmentation of two consecutive frames should be similar. This information about the LV dynamics can be used to impose constraints on the segmentations.

The goal of this chapter is to embed temporal information into the algorithms developed in the previous chapters, and take advantage of both. Two possibilities are explored: i) using a vector field-based model [126] of the LV dynamics to impose the expected contraction-dilation relationship during the ES and ED phases; and ii) using regularization over the shape model parameters to impose constraints on their expected evolution throughout the cardiac cycle.

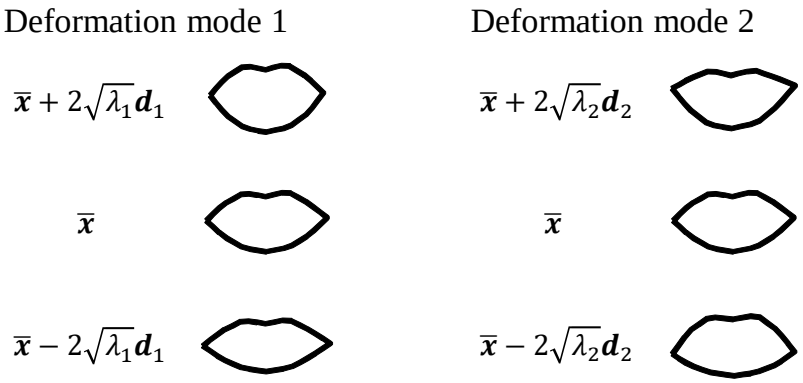


Figure 5.1: First two deformation modes of a 2D shape model of the lip contour.

The remainder of this chapter is organized as follows. The next section presents an overview of previous works and their benefits and disadvantages. Sections 5.3-5.4 describe the two proposed approaches. Section 5.5 describes the experimental setup and Section 5.6 shows and discusses the results. Finally, Section 5.7 concludes the chapter.

5.2 Background

Using temporal information within an ASM-based approach may be accomplished by either: i) embedding this information in the deformation modes of a high-dimensional shape model [11, 44, 64, 100]; or ii) combining the ASM with a specific dynamical model [10,24,30,172]. A brief description of these approaches and the state of the art is presented next.

5.2.1 High-dimensional Shape Model

As described in the previous chapters, statistical shape models have the ability to capture the main modes of variation of a particular shape. For instance, a 2D shape model was used in Chapter 3 to define a 2D contour with N points, $\boldsymbol{x} \in \mathbb{R}^{2N\times 1}$, based on a mean shape, $\overline{\boldsymbol{x}}$, and a linear combinations of the main modes of deformation $\boldsymbol{D} \in \mathbb{R}^{2N\times L}$,

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{b}, \tag{5.1}$$

where each column of $\boldsymbol{D}$, say $\boldsymbol{d}_l \in \mathbb{R}^{2N\times L}$, $l = 1, \ldots, L$, represents a specific mode of deformation, and $\boldsymbol{b}$ are the deformation coefficients. In this case, each mode of deformation applies a particular displacement to the position of each model point. As an illustrative example, Fig. 5.1 shows the influence that the first two main modes of deformation have in the shape model of the lips, obtained from the dataset described in Section 3.7.3.2. In this example, the first mode changes how wide the mouth is opened, and the second mode is related to the position of the corners of the lips.

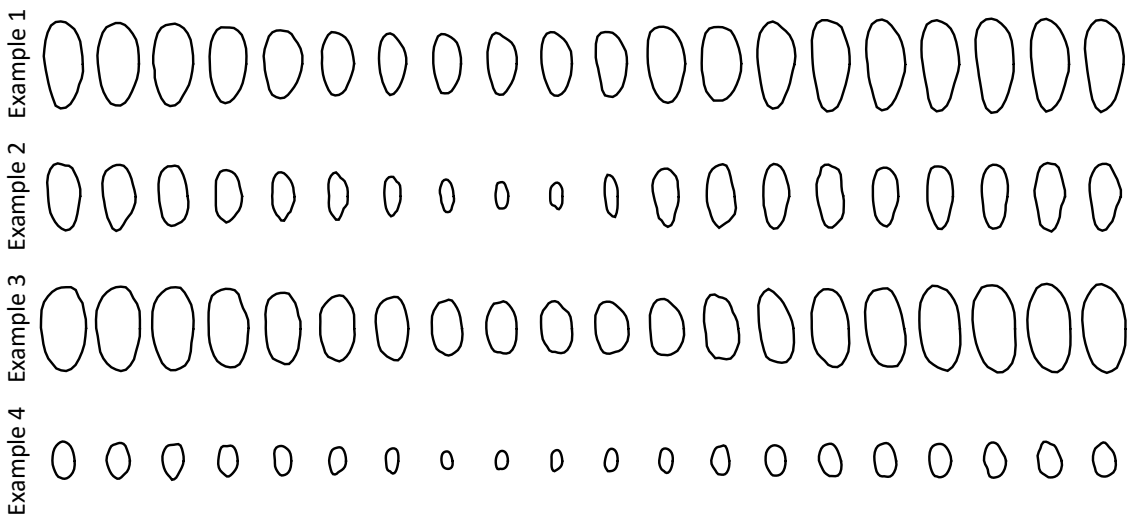


Figure 5.2: Examples from a training set for a 2D+t shape model in CMR. Each example consists of a set of $T = 20$ contours that define the LV border throughout the cardiac cycle (represented along the horizontal axis).

By analogy, in order to capture the variation of the LV along the cardiac cycle, one has to consider a higher-dimensional shape model, say $\boldsymbol{x}' \in \mathbb{R}^{2NT\times 1}$, which is the concatenation of T 2D contours (one contour for each frame). The training set for this 2D+t model would be composed of examples such as the ones shown in Fig. 5.2, *i.e.*, each example comprises T contours of the LV border. By performing PCA on a dataset with these examples, we obtain a shape model that captures the average contours throughout the cardiac cycle, as well as the most significant modes of variation. Fig. 5.3 illustrates the mean shape and the first main mode of variation. It is possible to see that this mode influences how much the LV contracts during the systolic phase.

High-dimensional shape models have been previously used for the segmentation of 2D+t sequences [11,44,64,100]. The downside of this approach is that they require a large amount of training data, a problem that is typically called the *curse of dimensionality* [11,131]. For instance, the public dataset [11] that was used in the previous chapters has 33 MR sequences, each with $T = 20$ and an average of $S = 6$ slices. Consequently, the training set would be composed of, at most, $(33-1)\times 6 = 192$ samples. However, in order to learn a statistical shape model with N points, a $2NT$-by-$2NT$ covariance matrix has to be estimated from the training data. Assuming $N = 40$, as in the previous chapters, this would lead to a 1600-by-1600 covariance matrix being estimated from a 1600-by-192 data matrix, which is not reasonable.

Another consideration that has to be taken into account is the lack of examples to capture the variability required to have a generalizable model. The inability of shape models to deal with test data that is not represented in the training set (*e.g.*, a cardiomyopathy that changes the contractile pattern of the LV) has always been a source of complaint in

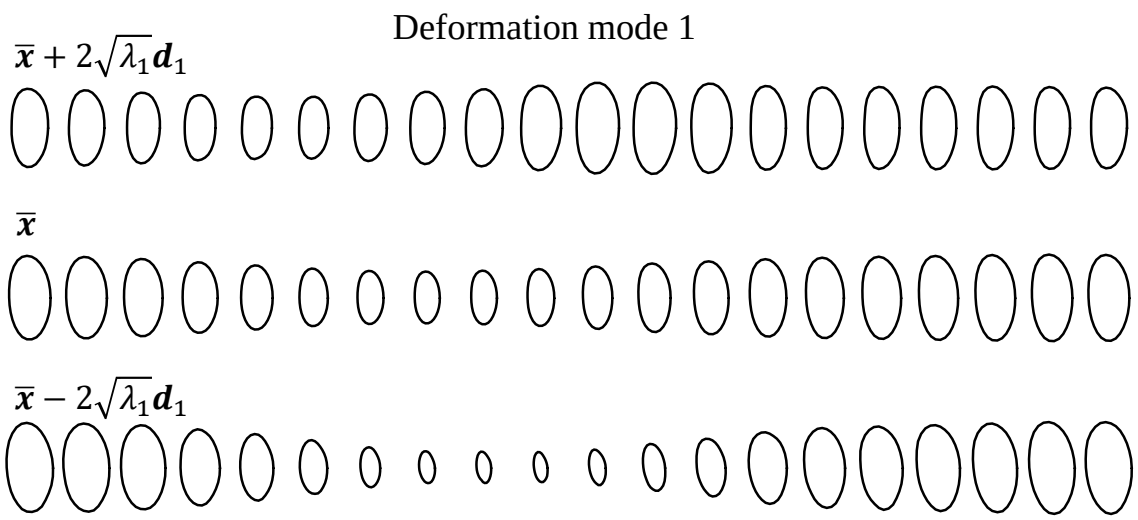


Figure 5.3: First main deformation mode for a 2D+t shape model of the LV in CMR.

the literature, particularly for medical images [160]. Although this is commonly overlooked for low-dimensional shape models, for high-dimensional models this becomes increasingly troublesome, which is why few works use high-dimensional shape models [11].

Most works resort to hierarchical ASMs [11,44,100] to overcome the *curse of dimensionality*. This variation of the traditional ASM divides the contour into several patches and learns a shape model for each patch. The main advantage is the significant reduction in the data dimensionality, thus allowing for temporal modes of variation to be learned. However, separating the contour model into patches leads to a loss of connectivity notion between patches, which may cause unexpected segmentations to be obtained. Furthermore, the model is also unable to capture the patterns of temporal variation that the patches might share.

5.2.2 Modeling the LV Dynamics

One of the main contributions to the understanding and modeling of the LV dynamics was the development of the tagged-MRI modality. As explained in Chapter 2, tagged-MRI allows a grid of specific myocardial points to be tracked throughout the cardiac cycle. With this information, it is possible to extract detailed information about tissue deformation, which is very difficult using other image modalities due to the lack of visual landmarks. Furthermore, over the past twenty years, there has been extensive research on electro-mechanical models of the heart [10, 24, 30, 72, 145, 165, 172, 181]. These models mimic the electrical stimulation of a complex network of myocardial fibers in the cardiac muscle and the resulting motion pattern, which helps understanding and diagnosing myocardial diseases. These developments, along with advances in the imaging technology, have fostered the appearance of many works in recent literature that tackle the problem of identifying pathological motion patterns through the analysis of muscle

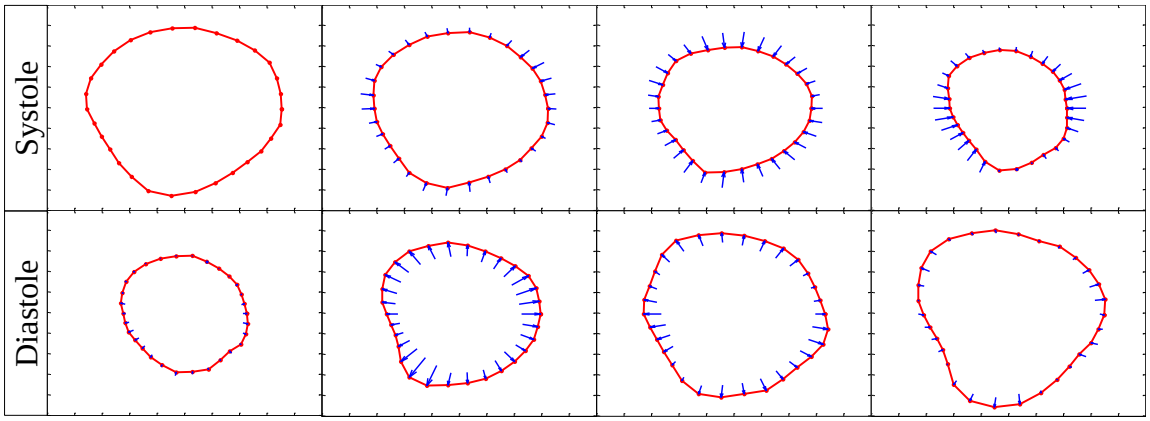


Figure 5.4: LV dynamics on eight frames along the cardiac cycle. Each plot shows the LV contour in red and the blue arrows depict the displacement from the previous to the current frame.

stress-strain [10, 40, 60, 134, 136, 186]. In the case of the LV segmentation, the focus is on the deformation pattern associated with the LV contour, shown in Fig. 5.4. Therefore, there is no need for such complex biological-inspired models.

Previous works have artificially modeled that LV dynamics using different approaches. For instance, Lynch et al. [107] model the contraction and dilation of the LV contour through the expected distance map of a level-set framework. The premise is that the distance of the LV contour to an arbitrary point in space (outside the LV chamber) increases and decreases as the heart contracts and dilates, respectively. This information is used to enforce this motion pattern on the segmentations of an MR sequence. Registration-based and optical flow-based methodologies have also been used to model temporal deformations [13, 56, 62, 83, 84, 91, 182]. Instead of using anatomical cues to extract the LV dynamics, these approaches rely on analyzing the motion of the image (or image features).

Regardless of how this prior knowledge is captured, it is necessary to embed this knowledge during the segmentation process. The classical tracking approach [42, 164, 168, 190] is to consider the dynamical model as a predictor of the LV position in one frame based on the segmentation of the previous frame. In other words, the LV dynamics provides an initialization (or bias) based on the previous frame. This allows replacing the human initialization in each frame (as was required in the previous chapters) with an automatic initialization provided by the dynamical model[1]. Fig. 5.5 shows a scheme based on this approach, in which the prediction model updates the position of the LV obtained in a frame f before the automatic segmentation of frame $f+1$ begins[2]. This process is repeated sequentially until the algorithm reaches the final frame of the MR sequence.

Although a more automated algorithm is desirable, it also has drawbacks. For instance, the prediction model is unlikely to provide better initializations than a human,

[1]Except for the first frame, which has no previous information of the LV position

[2]We adopt the notation f for frame instead of t to avoid confusing it with the iteration of the EM algorithm.

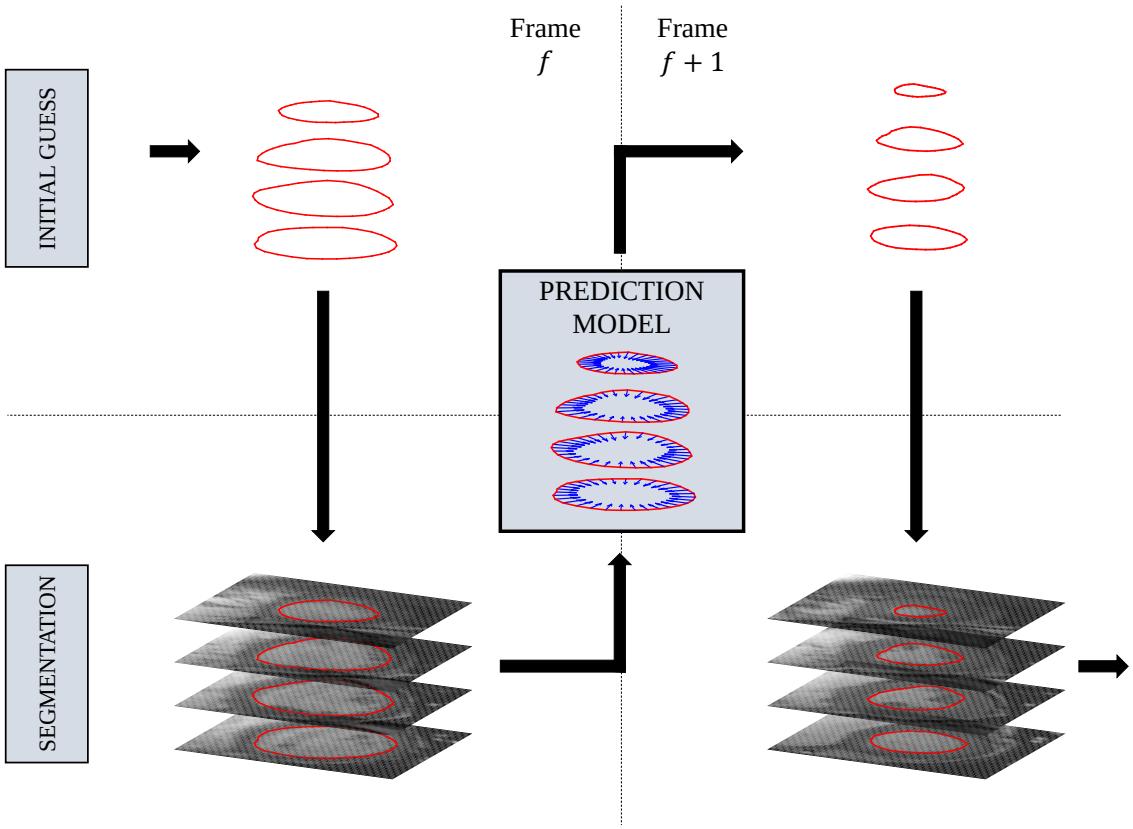


Figure 5.5: Schematic representation of an algorithm using the LV dynamics as a prediction model. The segmentation of a frame f is used obtain an initial guess for frame $f+1$ by applying the prediction model. This process repeats for each new frame.

i.e., the overall segmentation accuracy would be worse with this approach. Furthermore, it allows the accumulation of errors from one frame to the next when the output of the automatic segmentation is not accurate, leading to an increase in gross errors.

The drawbacks of tracking approaches are easily avoided by considering algorithms that segment all the frames concurrently. Given the fact that the whole MR sequence is available, it is possible to use the dynamical model to impose constraints on the whole sequence, instead of constraining each frame separately. The simplest approach relies on temporal smoothing, which forces the segmentations of consecutive frames to be similar between each other [120]. In this type of approach, the only prior information used is the assumption that the position of the LV contour changes smoothly over time. Periodicity has also been considered to improve the robustness of algorithms with weak priors [160]. More complex models, such as the ones presented above, can also be used to impose more detailed dynamical patterns on the segmentations.

In the particular case of this work, in which a shape model is used to segment, the LV dynamics are embedded using two different approaches. First, temporal smoothness will be imposed through the shape model parameters. Instead of imposing small variation of the contour points' position directly, the idea is to smooth the variation of the shape

model parameters throughout time. Second, the LV dynamics will be modeled through the expected displacement of the contour points explicitly (blue arrows in Fig. 5.4) using vector fields similar to those obtained with optical flow approaches. The following sections describe these two approaches in more detail.

5.3 Segmentation of MR Sequences Using Vector Fields

In this section, the goal is to learn a model of the LV dynamics, which corresponds to estimating the blue arrows depicted in Fig. 5.4. For this purpose, it is important to know the answer to the following questions: i) is it possible to describe the LV dynamics using a single model, or is it better to have specific models for specific frames? and ii) what is the influence of the slice position on the LV dynamics?

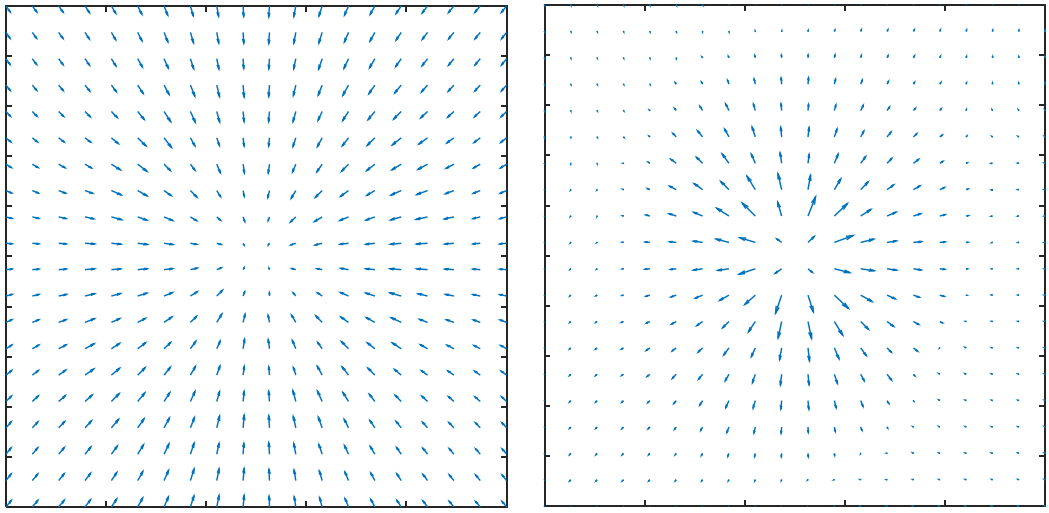
Regarding the first question, Fig. 5.4 clearly shows two distinct patterns: the LV shrinking due to the contraction of the cardiac muscle (systolic phase), and the LV dilating due to muscle relaxation (diastolic phase). Therefore, at least two different models have to be considered to describe the patterns of variation of the LV contour throughout the MR sequence. Furthermore, by looking closely at the length of the arrows, it is possible to see that it changes in each new frame. This suggests that having a specific models for each frame could be beneficial. However, in order to learn frame-specific models, the training data would have to be split into several groups (one for each frame), which would lead to an insufficient number of samples. Thus, the answer to the first question involves a trade-off between learning frame-specific dynamical models and having a larger training set.

Regarding the second question, there is also a trade-off between learning a specific dynamical model for each slice, which would also require splitting the training data into several groups, and learning more general dynamical models. But taking into account that MR volumes have a variable number of slices (recall Chapter 4), the former may not be the best approach.

A methodology that is able to deal with these issues is to use vector fields (VF)[3], which have been widely used to represent the behavior of dynamical systems [86]. They model the dynamics of an object as a function of its position.

We propose to use two 3D VFs [126] to describe the LV dynamics during the cardiac cycle. More specifically, the systolic and diastolic phases of the cardiac cycle are each modeled using a specific 3D VF. The vectors in the VF are located in a regular 3D grid, such that the displacement of each point in the LV contour in a specific slice is obtained through interpolation of the vectors in the grid. This approach has the additional advantage of avoiding the problems associated to the variable number of slices in MR volumes, since these fields do not require the contour points to match the grid nodes. An

[3]Denoted by displacement, velocity or motion fields, depending on the application.



(a) (b)

Figure 5.6: Vector field of the LV dynamics during: (a) systole and (b) diastole.

example of a middle slice of the VFs is shown in Fig. 5.6.

The following sections describe the VF parametrization, the estimation of these fields from training data, and how they are combined with the shape model formulation.

5.3.1 Vector Field Parametrization

In this section, the trajectory of each LV contour point throughout the cardiac cycle is modeled by two VFs. These VFs are assumed to be non-parametric representations of the velocities, which means it is necessary to discretize them within the data domain. We adopt the methodology described in [126], which consists of using a 3D grid with nodes at specific positions within the MR volume space, such that each node has an associated velocity vector. Since the LV corresponds to a very small fraction of the total image, it is beneficial to place the VF grid in a ROI concentric with the LV, as shown in Fig. 5.7. The velocity of a point in a generic position, which does not necessarily match the position of a node in the grid, is then obtained through tri-linear interpolation.

Formally, let $V : \mathbb{R}^3 \mapsto \mathbb{R}^3$ be a specific 3D VF, defined on a grid of N nodes. The vector $\boldsymbol{V} = \left[\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_N^\top\right]^\top \in \mathbb{R}^{3N}$ denotes the collection of the all the velocity vectors in the grid, where $\boldsymbol{v}_n \in \mathbb{R}^3$ is the vector associated with the n-th node. For a specific position $\widetilde{\boldsymbol{x}} \in \mathbb{R}^3$ within the image domain, the corresponding velocity vector, $V(\widetilde{\boldsymbol{x}}) \in \mathbb{R}^3$, is given by

$$\begin{aligned} V(\widetilde{\boldsymbol{x}}) &= \sum_{n=1}^{N} \phi_n(\widetilde{\boldsymbol{x}}) \boldsymbol{v}_n \\ &= \boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}) \boldsymbol{V}, \end{aligned} \tag{5.2}$$

where $\boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}) \in \mathbb{R}^{3 \times 3N}$ is a sparse matrix that determines the contribution of each node in

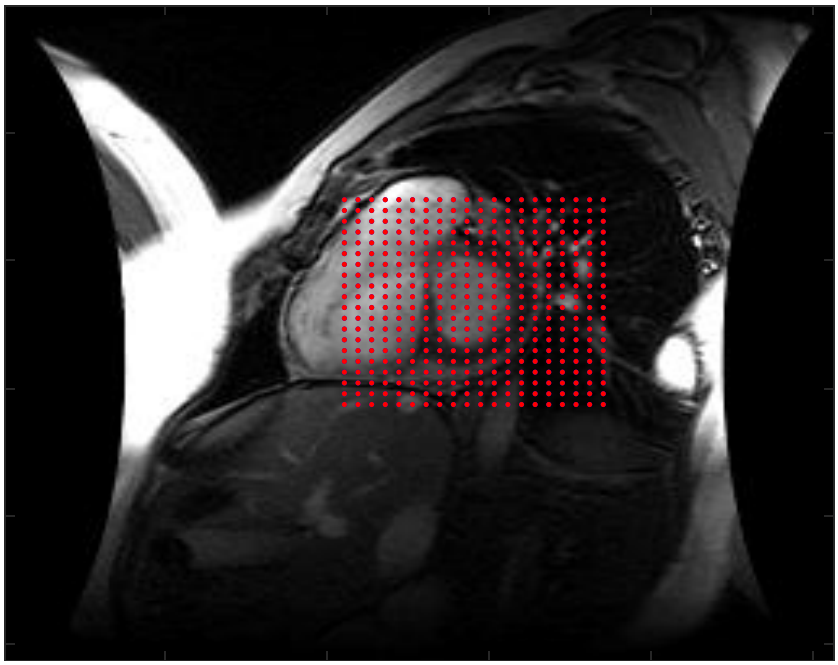


Figure 5.7: 2D view of the region of interested within which the VF grid (represented by the red dots) is located.

the grid to the computation of $V(\widetilde{\boldsymbol{x}})$, such that

$$\boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}) = \begin{bmatrix} \phi_1(\widetilde{\boldsymbol{x}}) & 0 & 0 & \dots & \boldsymbol{\phi}_N(\widetilde{\boldsymbol{x}}) & 0 & 0 \\ 0 & \phi_1(\widetilde{\boldsymbol{x}}) & 0 & \dots & 0 & \boldsymbol{\phi}_N(\widetilde{\boldsymbol{x}}) & 0 \\ 0 & 0 & \phi_1(\widetilde{\boldsymbol{x}}) & \dots & 0 & 0 & \boldsymbol{\phi}_N(\widetilde{\boldsymbol{x}}) \end{bmatrix}. \tag{5.3}$$

In the tri-linear interpolation scheme, only the eight closest grid nodes contribute, *i.e.*, at most only eight terms of $\{\phi_1(\widetilde{\boldsymbol{x}}), \dots, \phi_N(\widetilde{\boldsymbol{x}})\}$ are non-zero, and they satisfy the constraints

$$0 \leq \boldsymbol{\phi}_n(\widetilde{\boldsymbol{x}}) \leq 1 \tag{5.4}$$

$$\sum_{n=1}^{N} \boldsymbol{\phi}_n(\widetilde{\boldsymbol{x}}) = 1. \tag{5.5}$$

5.3.2 Estimation of the Vector Field

The two VFs used to describe the LV motion are estimated from a training data. This data consists of sequences of 20 contours along the cardiac cycle, for different slice positions (depending on the number of slices in each volume). Each sequence is divided in two phases: the systolic phase, in which the LV contracts, and the diastolic phase, in which the LV dilates. The frames of each phase are identifying by measuring the LV volume difference between consecutive frames: if the volume reduces, those frames belong to the systolic phase (approximately frames 1 to 8); if the volumes increases, then they belong to the diastolic phase (approximately frames 9 to 20). The goal of each VF is to model the trajectory of each contour point in the corresponding cardiac phase.

Consider one particular VF (the following procedure is valid for both VFs, but use different training trajectories). Let us assume we are given a set of K independent trajectories from a training set, such that the k-th trajectory is given by $\mathcal{X}^k = \{\widetilde{\boldsymbol{x}}^k(1), \ldots, \widetilde{\boldsymbol{x}}^k(T)\}$, where $\widetilde{\boldsymbol{x}}(f) \in \mathbb{R}^3$ denotes the position of a point at a specific frame $f = 1, \ldots, T$. Without loss of generality, the trajectories are assumed to be all sampled at the same frames and the same number of times. The VF describes the trajectory of a point through the following dynamical model [126]

$$\begin{aligned}\widetilde{\boldsymbol{x}}(f) &= \widetilde{\boldsymbol{x}}(f-1) + V\big(\widetilde{\boldsymbol{x}}(f-1)\big) + \boldsymbol{w} \\ &= \widetilde{\boldsymbol{x}}(f-1) + \boldsymbol{\Phi}\big(\widetilde{\boldsymbol{x}}(f-1)\big)\boldsymbol{V} + \boldsymbol{w},\end{aligned} \tag{5.6}$$

where $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma \boldsymbol{I})$ is white noise with Gaussian distribution.

The joint probability distribution of the trajectories in the training set, $\{\mathcal{X}^1, \ldots, \mathcal{X}^K\}$, and the VF, $\boldsymbol{V}$, is given by

$$\begin{aligned}p\Big(\mathcal{X}^1, \ldots, \mathcal{X}^K, \boldsymbol{V}\Big) &= p\Big(\mathcal{X}^1, \ldots, \mathcal{X}^K \Big| \boldsymbol{V}\Big) p\Big(\boldsymbol{V}\Big) \\ &= \prod_{k=1}^{K} p\Big(\mathcal{X}^k \Big| \boldsymbol{V}\Big) p\Big(\boldsymbol{V}\Big).\end{aligned} \tag{5.7}$$

The first term in the product is the likelihood of trajectory $\mathcal{X}^k$, and the second term corresponds to the prior probability of the coefficients of the VF. The model defined in (5.6) allows each of the likelihood terms to be factorized, since the position in a frame f only depends on the position in $f-1$, leading to

$$p\Big(\mathcal{X}^k \Big| \boldsymbol{V}\Big) = p\Big(\widetilde{\boldsymbol{x}}(1)\Big) \prod_{f=2}^{T} p\Big(\widetilde{\boldsymbol{x}}(f) \Big| \widetilde{\boldsymbol{x}}(f-1), \boldsymbol{V}\Big), \tag{5.8}$$

where $p\Big(\widetilde{\boldsymbol{x}}(1)\Big)$ is a constant, and

$$p\Big(\widetilde{\boldsymbol{x}}(f) \Big| \widetilde{\boldsymbol{x}}(f-1), \boldsymbol{V}\Big) = \mathcal{N}\Big(\widetilde{\boldsymbol{x}}(f) \Big| \widetilde{\boldsymbol{x}}(f-1) + \boldsymbol{\Phi}\big(\widetilde{\boldsymbol{x}}(f-1)\big)\boldsymbol{V}; \sigma^2 \boldsymbol{I}\Big). \tag{5.9}$$

Regarding the prior, $p(\boldsymbol{V})$, the formulation described in [126] uses a Gaussian prior that acts as a smoothness regularizer. This is accomplished by penalizing the difference between the velocity vectors of neighboring nodes. This leads to VFs in which all the vectors are pointing predominantly in the same direction. Here, the field is also expected to be smooth in the sense that its vectors should predominantly point either inwards or outwards, as shown in Fig. 5.6. In order to achieve the desired regularization effect, we adopt a prior term that penalizes large deviations in the radial and tangential components of the velocity vectors, with respect to a central point, $\boldsymbol{c}$. Formally, let $\boldsymbol{p}_n$ be the position of the n-th node, and $\boldsymbol{r}_n$ the corresponding normalized radial direction,

$$\boldsymbol{r}_n = \frac{\boldsymbol{p}_n - \boldsymbol{c}}{\|\boldsymbol{p}_n - \boldsymbol{c}\|}. \tag{5.10}$$

The prior acts on transformed velocity vectors, $\boldsymbol{v}'_n \in \mathbb{R}^2$, $n = 1, \ldots, N$, which corresponds to the radial and tangential components of $\boldsymbol{v}_n$, obtained by

$$\boldsymbol{v}'_n = \boldsymbol{R}_n \boldsymbol{v}_n, \tag{5.11}$$

where $\boldsymbol{R}_n \in \mathbb{R}^{2\times 3}$ is a transformation matrix that extracts the length of the radial and tangential components of the velocity vector, $\boldsymbol{v}_n$, in the n-th grid node, given by

$$\boldsymbol{R}_n = \left[\begin{array}{c|c} & \\ \boldsymbol{r}_n & \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{r}_n \\ & \end{array}\right]^\top . \tag{5.12}$$

This transformation can be performed for the whole VF by

$$\begin{aligned} \boldsymbol{V}' &= \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_2 & & \vdots \\ \vdots & & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{R}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_N \end{bmatrix} \\ &= \boldsymbol{R}\boldsymbol{V}. \end{aligned} \tag{5.13}$$

The goal of the regularization is to penalize the squared difference between all pair of nodes (m, n) encoded in the neighborhood set $\mathcal{N}$.

$$\sum_{(m,n)\in\mathcal{N}} \left\|\boldsymbol{v}'_m - \boldsymbol{v}'_n\right\|^2 = \sum_{(m,n)\in\mathcal{N}} (\boldsymbol{v}'_m - \boldsymbol{v}'_n)^\top (\boldsymbol{v}'_m - \boldsymbol{v}'_n). \tag{5.14}$$

Combining (5.14) with (5.11) leads to

$$\sum_{(m,n)\in\mathcal{N}} (\boldsymbol{R}_m \boldsymbol{v}_m - \boldsymbol{R}_n \boldsymbol{v}_n)^\top (\boldsymbol{R}_m \boldsymbol{v}_m - \boldsymbol{R}_n \boldsymbol{v}_n). \tag{5.15}$$

The neighborhood set $\mathcal{N}$ is based on a 6-connected neighborhood given by the 3D grid of the VF. The difference between a pair of neighboring nodes, $\boldsymbol{v}'_m - \boldsymbol{v}'_n$, may be compactly written in matrix notation for all the pairs in the grid as

$$\begin{bmatrix} \boldsymbol{v}'_1 - \boldsymbol{v}'_2 \\ \boldsymbol{v}'_2 - \boldsymbol{v}'_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & \ldots & & & 0 \\ 0 & 1 & 0 & -1 & 0 & \ldots & & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \ \ldots & 0 \\ & & & & \vdots & & & \end{bmatrix} \boldsymbol{V}' = \boldsymbol{\Delta}\boldsymbol{R}\boldsymbol{V}, \tag{5.16}$$

where each two lines of $\boldsymbol{\Delta}$ encode the difference between a specific pair of nodes. Thus, the sum in (5.14) may be rewritten as

$$(\boldsymbol{\Delta}\boldsymbol{R}\boldsymbol{V})^\top (\boldsymbol{\Delta}\boldsymbol{R}\boldsymbol{V}) = \boldsymbol{V}^\top \boldsymbol{R}^\top \boldsymbol{\Delta}^\top \boldsymbol{\Delta}\boldsymbol{R}\boldsymbol{V}. \tag{5.17}$$

This regularization effect is obtained using a Gaussian prior probability for $\boldsymbol{V}$, with zero mean and covariance $\boldsymbol{\Lambda}$, such that $\boldsymbol{\Lambda}^{-1} = \alpha \boldsymbol{R}^\top \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \boldsymbol{R}$, where α is a parameter that allows the strength of the prior to be tuned (larger values of α lead to more regularized fields). Thus, the prior term can be written as

$$\begin{aligned} p(\boldsymbol{V}) &\propto \exp\left\{-\frac{1}{2}\boldsymbol{V}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{V}\right\} \\ &\propto \exp\left\{-\frac{\alpha}{2}\boldsymbol{V}^\top \boldsymbol{R}^\top \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \boldsymbol{R} \boldsymbol{V}\right\}. \end{aligned} \tag{5.18}$$

Combining (5.7) with (5.8) and (5.18), the maximum posterior estimate of the VF is given by

$$\begin{aligned} \boldsymbol{V}^\star &= \arg\max_{\boldsymbol{V}} \log p\Big(\mathcal{X}^1, \ldots, \mathcal{X}^K, \boldsymbol{V}\Big) \\ &= \arg\min_{\boldsymbol{V}} \frac{1}{\sigma^2} \sum_{k=1}^{K} \sum_{f=2}^{T} \Big\| \widetilde{\boldsymbol{x}}^k(f) - \widetilde{\boldsymbol{x}}^k(f-1) - \boldsymbol{\Phi}\big(\widetilde{\boldsymbol{x}}^k(f-1)\big)\boldsymbol{V} \Big\|^2 + \alpha \boldsymbol{V}^\top \boldsymbol{R}^\top \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \boldsymbol{R} \boldsymbol{V}. \end{aligned} \tag{5.19}$$

The solution of (5.19) is obtained through the computation of the derivative of the objective function with respect to $\boldsymbol{V}$, which leads to the following linear equation

$$\begin{aligned} \left(\alpha \boldsymbol{V}^\top \boldsymbol{R}^\top \boldsymbol{\Delta}^\top \boldsymbol{\Delta} \boldsymbol{R} + \frac{1}{\sigma^2} \sum_{k=1}^{K} \sum_{f=2}^{T} \boldsymbol{\Phi}^\top\big(\widetilde{\boldsymbol{x}}^k(f-1)\big)\boldsymbol{\Phi}\big(\widetilde{\boldsymbol{x}}^k(f-1)\big)\right) \boldsymbol{V}^\star = \\ = \frac{1}{\sigma^2} \sum_{k=1}^{K} \sum_{f=2}^{T} \boldsymbol{\Phi}^\top\big(\widetilde{\boldsymbol{x}}^k(f-1)\big)\Big(\widetilde{\boldsymbol{x}}^k(f) - \widetilde{\boldsymbol{x}}^k(f-1)\Big). \end{aligned} \tag{5.20}$$

Note that since the trajectories are coplanar with the volume slices, the third component of the VFs is always going to be zero. Examples of the trajectories are shown in Fig. 5.8, and the corresponding VFs are learned using the procedure described above for a grid of $20 \times 20 \times 8$. Fig. 5.6 shows the 4th slice of the VF for the systolic and for the diastolic phases.

5.3.3 Segmentation of the LV with Vector Fields

The VFs learned using the procedure described in the previous section model the LV dynamics along the cardiac cycle. In order to use this information in the segmentation process, we take advantage of the EM-RASM formulation described in Chapters 3 and 4. Temporal information is added both to the 2D and to the 3D formulation of EM-RASM, using a similar rational. In this section, we only describe the VF approach for the 2D setting for the sake of readability, since the notation is simpler.

Recalling Chapter 3, the shape model parameters in iteration t, $\widehat{\boldsymbol{\Theta}}_{(t)}$, which include the transformation parameters, $\boldsymbol{a}$ and $\boldsymbol{t}$, and the deformation coefficients, $\boldsymbol{b}$, were updated by maximizing an auxiliary function,

$$\widehat{\boldsymbol{\Theta}}_{(t+1)} = \arg\max_{\boldsymbol{\Theta}} Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right), \tag{5.21}$$

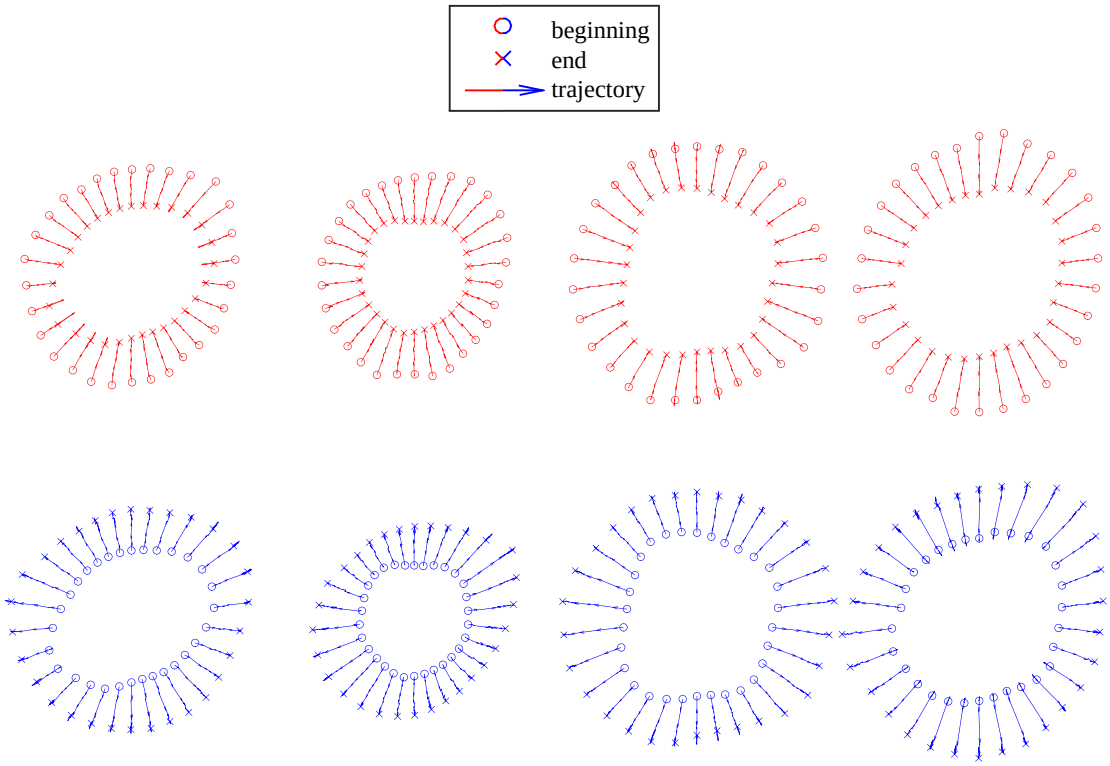


Figure 5.8: Examples of the trajectories of LV contour points. The top row shows the systolic phase and the bottom shows the diastolic phase.

This led to a weighted least squares minimization problem, in which the goal is to find the shape model parameters that minimize the sum of weighted distances between the model points, $\widetilde{\boldsymbol{x}}^i$, which depend on the shape model parameters $\widehat{\boldsymbol{\Theta}}_{(t)}$ through (3.20), and some observation points, $\boldsymbol{y}^{ij}$, extracted from the images. The weight of each observation point, w_1^{ij}, is given by the likelihood of that point belonging to the LV border (*i.e.*, having label $k^{ij} = 1$). Thus, the objective function was given by

$$\widehat{\boldsymbol{\Theta}}_{(t+1)} = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij} \left\| \widetilde{\boldsymbol{x}}^i - \boldsymbol{y}^{ij} \right\|_{\boldsymbol{\Sigma}^i}^2 + p(\boldsymbol{\Theta}), \tag{5.22}$$

where $\|\boldsymbol{v}\|_{\boldsymbol{\Sigma}}^2 = \boldsymbol{v}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{v}$, $\boldsymbol{\Sigma}^i$ is a diagonal covariance matrix associated with the i-th model point, and $p(\boldsymbol{\Theta})$ is the prior probability of the model parameters.

In this chapter, each variable explicitly depends on the frame, $f = 1, \ldots, T$, of the MR sequence to which they are associated. The optimization problem in (5.22) is rewritten as

$$\{\widehat{\boldsymbol{\Theta}}(1), \ldots, \widehat{\boldsymbol{\Theta}}(T)\}_{(t+1)} = \arg \min_{\boldsymbol{\Theta}(1),\ldots,\boldsymbol{\Theta}(T)} \sum_{f=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M^i} w_1^{ij}(f) \left\| \widetilde{\boldsymbol{x}}^i(f) - \boldsymbol{y}^{ij}(f) \right\|_{\boldsymbol{\Sigma}^i}^2 . \tag{5.23}$$

The goal is to penalize large deviations between the each shape model point, $\widetilde{\boldsymbol{x}}(f)$, $f = 2, \ldots, T$ and its expected position, given by

$$\boldsymbol{y}_V^i(f) = \widetilde{\boldsymbol{x}}^i(f-1) + V_f(\widetilde{\boldsymbol{x}}^i(f-1)), \tag{5.24}$$

where $V_f(\widetilde{\boldsymbol{x}}^i(f-1))$ is given by the active VF in frame f (either the systolic VF or the diastolic VF), using (5.2). This is achieved by adding a new term,

$$\sum_{f=2}^{T}\sum_{i=1}^{N}\left\|\widetilde{\boldsymbol{x}}^i(f)-\boldsymbol{y}_V^i(f)\right\|^2_{\boldsymbol{\Sigma}^i}, \tag{5.25}$$

to the objective function in (5.23). This leads to

$$\begin{aligned}\{\widehat{\boldsymbol{\Theta}}(1),\ldots,\widehat{\boldsymbol{\Theta}}(T)\}_{(t+1)} = \arg\min_{\boldsymbol{\Theta}(1),\ldots,\boldsymbol{\Theta}(T)} &\sum_{f=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(f)\left\|\widetilde{\boldsymbol{x}}^i(f)-\boldsymbol{y}^{ij}(f)\right\|^2_{\boldsymbol{\Sigma}^i} + \\ &+\lambda_V\sum_{f=2}^{T}\sum_{i=1}^{N}\left\|\widetilde{\boldsymbol{x}}^i(f)-\widetilde{\boldsymbol{x}}^i(f-1)-V_f(\widetilde{\boldsymbol{x}}^i(f-1))\right\|^2_{\boldsymbol{\Sigma}^i},\end{aligned} \tag{5.26}$$

where λ_V is a constant that determines the importance of the VF term.

Notice that $V_f(\widetilde{\boldsymbol{x}}^i(f-1))$ should also depend on the model parameters, $\boldsymbol{\Theta}$ (recall that $\widetilde{\boldsymbol{x}}^i(f-1)$ is given by (3.20)). However, this would make the optimization problem much more complex. Thus, we make the assumption that $\boldsymbol{y}_V^i(f-1)=\widetilde{\boldsymbol{x}}^i(f-1)-V_f(\widetilde{\boldsymbol{x}}^i(f-1))$ is fixed given the current shape model parameters. In practice, this can be seen as considering new observation points given by the expected LV motion, which are all weighted by λ_V. This will introduce a bias in the estimation of the shape model parameters that will make the segmentations combine the information from the image with temporal dynamics.

The solution of (5.26) is computed by alternating between the minimization with respect to the transformation parameters, $\boldsymbol{a}(f),\boldsymbol{t}(f)$, $f=1,\ldots,T$, and the minimization with respect to the deformation coefficients, $\boldsymbol{b}(f)$, $f=1,\ldots,T$, as in the previous formulation.

Update of the Transformation Parameters The updated transformation parameters in iteration $t+1$ of EM-RASM are obtained by computing the derivative of (5.26) with respect to $\boldsymbol{a}(f)$ and $\boldsymbol{t}(f)$, keeping $\boldsymbol{b}(f)$ constant. The solution for each frame f is independent, leading to the following linear equation

$$\Big(\boldsymbol{M}(f)+\boldsymbol{M}_V(f)\Big)\begin{bmatrix}\widehat{\boldsymbol{a}}(f)_{(t+1)}\\ \widehat{\boldsymbol{t}}(f)_{(t+1)}\end{bmatrix}=\boldsymbol{c}(f)+\boldsymbol{c}_V(f) \tag{5.27}$$

where $\boldsymbol{M}(f)$ and $\boldsymbol{c}(f)$ are, as in the previous formulation (see (3.34)), given by

$$\boldsymbol{M}(f)=\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(f)\begin{bmatrix}\boldsymbol{X}^{i\top}(f)\boldsymbol{\Sigma}^{i-1}\boldsymbol{X}^i(f) & \boldsymbol{X}^{i\top}(f)\boldsymbol{\Sigma}^{i-1}\\ \boldsymbol{\Sigma}^{i-1}\boldsymbol{X}^i(f) & \boldsymbol{\Sigma}^{i-1}\end{bmatrix}, \tag{5.28}$$

$$\boldsymbol{c}(f)=\sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(f)\begin{bmatrix}\boldsymbol{X}^{i\top}(f)\boldsymbol{\Sigma}^{i-1}\boldsymbol{y}^{ij}(f)\\ \boldsymbol{\Sigma}^{i-1}\boldsymbol{y}^{ij}(f)\end{bmatrix}, \tag{5.29}$$

and

$$\boldsymbol{M}_V(f) = \lambda_V \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{X}^{i\top}(f)\boldsymbol{\Sigma}^{i-1}\boldsymbol{X}^i(f) & \boldsymbol{X}^{i\top}(f)\boldsymbol{\Sigma}^{i-1} \\ \boldsymbol{\Sigma}^{i-1}\boldsymbol{X}^i(f) & \boldsymbol{\Sigma}^{i-1} \end{bmatrix}, \tag{5.30}$$

$$\boldsymbol{c}_V(f) = \lambda_V \sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{X}^{i\top}(f)\boldsymbol{\Sigma}^{i-1}\boldsymbol{y}_V^i(f) \\ \boldsymbol{\Sigma}^{i-1}\boldsymbol{y}_V^i(f) \end{bmatrix}. \tag{5.31}$$

Notice that this solution only differs from the one in (3.34) is the summation over the "new" observation points, $\boldsymbol{y}_V^i(f)$, weighted by λ_V.

Update of the Deformation Parameters The deformation parameters, $\boldsymbol{b}(f)$, $f = 1, \ldots, T$, are updated by computing the derivative of (5.26) with respect to each $\boldsymbol{b}(f)$. As in the case of the transformation parameters, the solution to this minimization problem is very similar to (3.35), except for the additional summation over the VF observation points, $\boldsymbol{y}_V^i(f)$,

$$\Big(\boldsymbol{N}(f) + \boldsymbol{N}_V(f)\Big)\widehat{\boldsymbol{b}}(f)_{(t+1)} = \boldsymbol{d}(f) + \boldsymbol{d}_V(f), \tag{5.32}$$

with

$$\boldsymbol{N}(f) = \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}(f)\boldsymbol{D}^{i\top}\widehat{\boldsymbol{A}}^{\top}(f)_{(t+1)}\boldsymbol{\Sigma}^{i-1}\widehat{\boldsymbol{A}}(f)_{(t+1)}\boldsymbol{D}^i, \tag{5.33}$$

$$\boldsymbol{N}_V(f) = \lambda_V \sum_{i=1}^{N} \boldsymbol{D}^{i\top}\widehat{\boldsymbol{A}}^{\top}(f)_{(t+1)}\boldsymbol{\Sigma}^{i-1}\widehat{\boldsymbol{A}}(f)_{(t+1)}\boldsymbol{D}^i, \tag{5.34}$$

$$\boldsymbol{d}(f) = \sum_{i=1}^{N}\sum_{j=1}^{M^i} w_1^{ij}\boldsymbol{D}^{i\top}\widehat{\boldsymbol{A}}^{\top}(f)_{(t+1)}\boldsymbol{\Sigma}^{i-1}\left(\boldsymbol{y}^{ij}(f) - \widehat{\boldsymbol{A}}(f)_{(t+1)}\overline{\boldsymbol{x}}^i - \widehat{\boldsymbol{t}}(f)_{(t+1)}\right), \tag{5.35}$$

$$\boldsymbol{d}_V(f) = \lambda_V \sum_{i=1}^{N} \boldsymbol{D}^{i\top}\widehat{\boldsymbol{A}}^{\top}(f)_{(t+1)}\boldsymbol{\Sigma}^{i-1}\left(\boldsymbol{y}_V^i(f) - \widehat{\boldsymbol{A}}(f)_{(t+1)}\overline{\boldsymbol{x}}^i - \widehat{\boldsymbol{t}}(f)_{(t+1)}\right). \tag{5.36}$$

The new update equations for the shape model parameters have embedded prior information about the LV motion, which is expressed in terms of the expected displacement between consecutive frames, provided by the VF.

5.4 Segmentation of MR Sequences Using Temporal Regularization

A different approach to embed temporal information in the LV segmentation is explored in this section. The goal is to use the expected variation shape model parameters throughout the cardiac cycle (shown in Fig. 5.9) to describe the displacements depicted in Fig. 5.4, which is an indirect way of capturing the LV dynamics contrary to the approach described in the previous section.

As in the case of the VFs, the approach proposed in this section can be combined with both the 2D and the 3D formulations of EM-RASM. To keep the notation simpler and improve readability, we only detail the combination with the 2D approach, but the same rational is used for the 3D setting.

Recal the equation of the 2D shape model (see (3.2)), in which the contour of LV is given by

$$\widetilde{\boldsymbol{x}} = T_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{b}), \tag{5.37}$$

where $\boldsymbol{b} \in \mathbb{R}^L$ are the deformation coefficients, and $T_{\boldsymbol{\theta}}(\cdot)$ is a similarity transformation that scales, rotates, and translates the contour based on parameters $\boldsymbol{\theta} = \{\boldsymbol{a} = [a_1, a_2]^\top, \boldsymbol{t} = [t_1, t_2]^\top\}$.

As in the previous section, let us explicitly associate the shape model parameters with a specific frame, $f = 1, \ldots, T$, by defining them as $\boldsymbol{a}(f)$, $\boldsymbol{t}(f)$ $\boldsymbol{b}(f)$. In this approach, we assume that the LV motion is entirely described by the variation of the shape model parameters $\boldsymbol{a}(f)$ and $\boldsymbol{t}(f)$ of the global transformation, *i.e.*, that there is no correlation between the variation of the deformation coefficients $\boldsymbol{b}(f)$ and the LV dynamics. This assumption is reasonable given the fact that the deformation coefficients, $\boldsymbol{b}(f)$, are only related with local variation of the contour shape, and not the overall motion of the LV.

The ground truth contours of a training set can be used to analyze the variation of the shape model parameters throughout time. By fitting the shape model to the contours of each frame in the training set, the corresponding shape model parameters can be extracted.

There are two possibilities to analyze the variation of the parameters: i) by comparing with the first frame; and ii) by comparing with the previous frame. These two possibilities are depicted in Fig. 5.9 (a) and (b), respectively. Fig. 5.9 shows the scale of the shape model, given by $\|\boldsymbol{a}(f)\|$, which approximately corresponds to $a_1(f)$, since $a_2(f)$ is almost zero in all cases [4]. This also means that the rotation, given by $\arctan\left(\frac{a_2}{a_1}\right)$ is approximately zero. Regarding the first possibility, the figure shows

$$\begin{matrix} \text{scale:} & a_1(f)/a_1(1) \\ \text{translation:} & \boldsymbol{t}(f) - \boldsymbol{t}(1) \end{matrix}, \; f = 1, \ldots, T, \tag{5.38}$$

while for the second possibility the figure shows

$$\begin{matrix} \text{scale:} & a_1(f)/a_1(f-1) \\ \text{translation:} & \boldsymbol{t}(f) - \boldsymbol{t}(f-1) \end{matrix}, \; f = 2, \ldots, T, \tag{5.39}$$

with $T = 20$.

The periodic behavior of contraction followed by dilation that is characteristic of the LV is very clear for the scale shown in Fig. 5.9 (top row). Regarding parameter $\boldsymbol{t}(f)$, the only noticeable pattern is related to $t_1(f)$, which shows that the LV tends to move left

[4] The highest value across all the training set is $|a_2| = 1.6 \times 10^{-4}$.

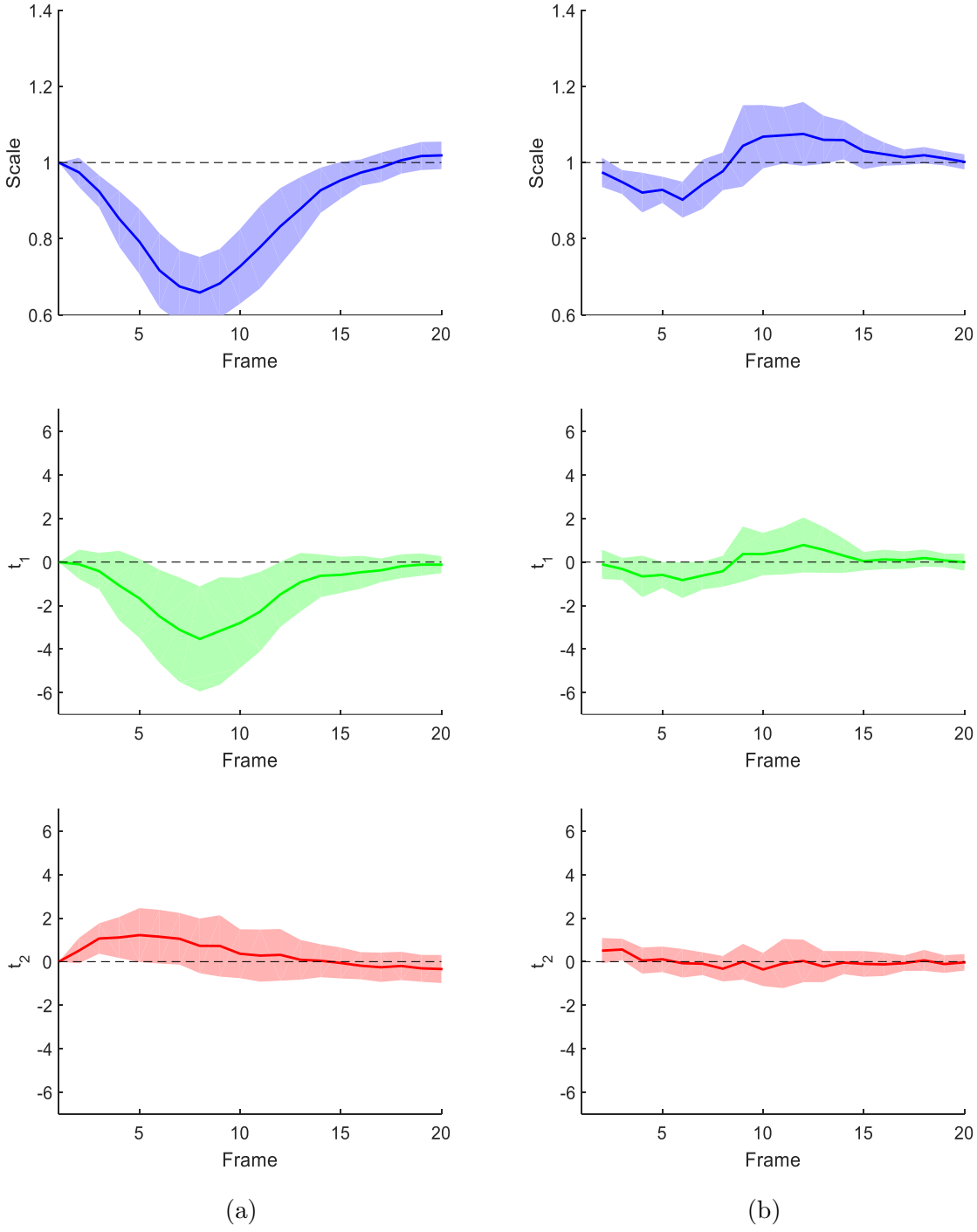


Figure 5.9: Variation of the global transformation parameters of the shape model throughout the cardiac cycle: (a) relative to the first frame; (b) relative to the previous frame. The top row shows the scale, and the other rows show the two coordinates of the translation. Each plot shows the mean value (line) and the standard deviation (region) observed in the training set.

while contracting and return when dilating. Other than that, the only strong evidence is the fact that the difference between consecutive frames is small.

The dependencies depicted in Fig. 5.9 allow the LV dynamics to be modeled through

constraints on the shape model parameters. The following section describes how the parameter variation given by the second possibility is taken into account during the segmentation process.

5.4.1 MAP Formulation With Informative Priors

In order to embed the parameter variation during the segmentation process, we take advantage of the EM-RASM algorithm proposed in Chapter 3. The idea of the MAP formulation of the algorithm, described in Section 3.4.2, is to maximize the joint probability distribution of the observations extracted from the image, $\boldsymbol{Y}$, their corresponding labels, $\boldsymbol{K}$, and the set of model parameters $\boldsymbol{\Theta}$ (see (3.38)). In this case, since the whole MR sequence is being segmented simultaneously, the time dependency of these variables has to be considered. Thus, for each frame $f = 1, \ldots, T$ in the sequence, the corresponding variables are denoted by $\boldsymbol{Y}(f)$, $\boldsymbol{K}(f)$ and $\boldsymbol{\Theta}(f)$, respectively.

Assuming that observations in different frames are statistically independent (and the same for their labels), then the log joint probability for this new formulation is defined as

$$\begin{aligned}\mathcal{P}\Big(\boldsymbol{Y}(1),\ldots,\boldsymbol{Y}(T),&\boldsymbol{K}(1),\ldots,\boldsymbol{K}(T),\boldsymbol{\Theta}(1),\ldots,\boldsymbol{\Theta}(T)\Big) = \\ &= \sum_{f=1}^{T} \log p\Big(\boldsymbol{Y}(f),\boldsymbol{K}(f)\Big|\boldsymbol{\Theta}(f)\Big) + \log p\Big(\boldsymbol{\Theta}(1),\ldots,\boldsymbol{\Theta}(T)\Big). \quad (5.40)\end{aligned}$$

This equation includes the sum over all the likelihood terms, $p\big(\boldsymbol{Y}(f),\boldsymbol{K}(f)\big|\boldsymbol{\Theta}(f)\big)$, which are used in the ML formulation, and prior term for the model parameters, $p\big(\boldsymbol{\Theta}(1),\ldots,\boldsymbol{\Theta}(T)\big)$, where $\boldsymbol{\Theta}(f) = \{\boldsymbol{a}(f),\boldsymbol{t}(f),\boldsymbol{b}(f),\boldsymbol{p}(f)\}$, where $\boldsymbol{p}(f)$ is the probability of the observation models (recall Section 3.4).

In Chapter 3, the model parameters were assumed to be independent between each other and across all the frames (although the latter part was assumed implicitly by segmenting each frame individually). Now, only the former holds, *i.e.*,

$$\begin{aligned}p\Big(\boldsymbol{\Theta}(1),\ldots,\boldsymbol{\Theta}(T)\Big) &= \\ = p\Big(\boldsymbol{a}(1),\ldots,\boldsymbol{a}(T)\Big)&p\Big(\boldsymbol{t}(1),\ldots,\boldsymbol{t}(T)\Big)p\Big(\boldsymbol{b}(1),\ldots,\boldsymbol{b}(T)\Big)p\Big(\boldsymbol{p}(1),\ldots,\boldsymbol{p}(T)\Big). \quad (5.41)\end{aligned}$$

Furthermore, $p(\boldsymbol{a}(f))$ and $p(\boldsymbol{t}(f))$ were assumed to be defined as the non-informative Jeffreys' priors. In this chapter, these prior terms will be used to embed the LV dynamics through the expected parameter variation presented above.

Parameter $\boldsymbol{a}(f)$ The scale and rotation parameters, $\boldsymbol{a}(f)$, $f = 1, \ldots, T$, are modeled by a multivariate Gaussian prior probability given by

$$p\big(\boldsymbol{a}(1)\ldots,\boldsymbol{a}(T)\big) = \mathcal{N}\left(\Big[\boldsymbol{a}^{\top}(1),\ldots,\boldsymbol{a}^{\top}(T)\Big]^{\top}\Big|\mathbf{0},\Lambda_{\boldsymbol{a}}\right), \quad (5.42)$$

where $\Lambda_{\boldsymbol{a}}$ is a covariance matrix that accounts for the dependencies between neighboring frames, such that

$$p\big(\boldsymbol{a}(1),\ldots,\boldsymbol{a}(T)\big) \propto \exp\left(-\frac{\lambda_{\boldsymbol{a}}}{2}\sum_{f=2}^{T}\|\boldsymbol{a}(f)-\boldsymbol{H}_f\boldsymbol{a}(f-1)\|^2\right), \tag{5.43}$$

where $\lambda_{\boldsymbol{a}}$ is a constant parameter that is used to tune the weight of the prior, and

$$\boldsymbol{H}_f = \begin{bmatrix} h_f & 0 \\ 0 & 1 \end{bmatrix} \tag{5.44}$$

is the update matrix, where h_f is a constant, learned from a training set, given by the average value of $\frac{a(f)}{a(f-1)}$ for each frame $f = 2,\ldots,T$. This coefficient is related to the curve shown for the second possibility discussed previously (see (5.39)), shown in the top row of Fig. 5.9 (b). In practice, the prior (5.43) expresses two ideas: i) that the expected value of $a_1(f)$ is given by $h_f a_1(f-1)$; and ii) that $a_2(f)$ is a static parameter, *i.e.*, it is not supposed to change significantly throughout the cardiac cycle. This prior intrinsically models the dynamical pattern that the LV exhibits during the cardiac cycle by penalizing large deviations from expected parameter variation.

Parameter $\boldsymbol{t}(f)$ Similarly to $\boldsymbol{a}(f)$, the translation parameters, $\boldsymbol{t}(f)$, $f = 1,\ldots,T$, are assumed to follow a prior probability defined by

$$\begin{aligned} p\big(\boldsymbol{t}(1),\ldots,\boldsymbol{t}(T)\big) &= \mathcal{N}\left(\Big[\boldsymbol{t}^\top(1),\ldots,\boldsymbol{t}^\top(T)\Big]^\top \Big| \boldsymbol{0},\Lambda_{\boldsymbol{t}}\right) \\ &\propto \exp\left(-\frac{\lambda_{\boldsymbol{t}}}{2}\sum_{f=2}^{T}\|\boldsymbol{t}(f)-\boldsymbol{t}(f-1)\|^2\right), \end{aligned} \tag{5.45}$$

where $\lambda_{\boldsymbol{t}}$ defines the weight of this prior. In this case, translation coefficient was assumed to be similar throughout the cardiac cycle, with no particular dynamical behavior. Although Fig. 5.9 (b) shows that the first coordinate of the translation coefficient, $t_1(f)$, tends to have a characteristic pattern throughout the cardiac cycle, modeling it as a static parameter does not mean it will lead to poor results. In fact, this prior will lead to a regularization term in the update equations, as will be shown next, that simply prevents large differences between the values of $\boldsymbol{t}(f)$ of consecutive frames.

Revisiting the MAP formulation of EM-RASM, the shape model parameters for a specific sequence are iteratively obtained by performing the following steps.

5.4.1.1 E-step

Let $\widehat{\boldsymbol{\Theta}}_{(t)} = \{\widehat{\boldsymbol{\Theta}}_{(t)}(1),\ldots,\widehat{\boldsymbol{\Theta}}_{(t)}(T)\} = \{\widehat{\boldsymbol{a}}(f),\widehat{\boldsymbol{t}}(f),\widehat{\boldsymbol{b}}(f),\widehat{\boldsymbol{p}}(f) : f = 1,\ldots,T\}_{(t)}$ be the estimates of the model parameters at iteration t for all the frames in the MR sequence.

Also, let $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{Y}(1), \ldots, \boldsymbol{Y}(T)\}$ and $\boldsymbol{\mathcal{K}} = \{\boldsymbol{K}(1), \ldots, \boldsymbol{K}(T)\}$ be a set of observations extracted from all the frames and their corresponding labels, respectively. Given the set of observations, $\boldsymbol{\mathcal{Y}}$, and the most recent estimates, an auxiliary function $Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right)$ is defined as the expected value of the log-joint distribution in (5.40),

$$\begin{aligned} Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) &= \mathbb{E}_{\boldsymbol{K}}\left[\mathcal{P}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\Theta}) \middle| \boldsymbol{\mathcal{Y}}, \widehat{\boldsymbol{\Theta}}_{(t)}\right] \\ &= \mathbb{E}_{\boldsymbol{K}}\left[\sum_{f=1}^{T} \log p\Big(\boldsymbol{Y}(f), \boldsymbol{K}(f) | \boldsymbol{\Theta}(f)\Big)\right] + \log p\Big(\boldsymbol{\Theta}(1), \ldots, \boldsymbol{\Theta}(T)\Big). \end{aligned} \tag{5.46}$$

The first term is the same that appears in the ML formulation. By expanding the expectation for the two possible values, $k = \{0, 1\}$, it leads to an equation similar to (3.28). The only differences are the additional sum over the T frames and the explicit dependency of each variable on the frame f.

Regarding the second term - the prior, it has been shown it can be expanded as in (5.41), leading to

$$\begin{aligned} \log p\big(\boldsymbol{\Theta}(1), \ldots, \boldsymbol{\Theta}(T)\big) = \log p\big(\boldsymbol{a}(1), \ldots, \boldsymbol{a}(T)\big) + \log p\big(\boldsymbol{t}(1), \ldots, \boldsymbol{t}(T)\big) + \\ + \log p\big(\boldsymbol{b}(1), \ldots, \boldsymbol{b}(T)\big) + \log p\big(\boldsymbol{p}(1), \ldots, \boldsymbol{p}(T)\big). \end{aligned} \tag{5.47}$$

Each term has been previously defined in (5.43), (5.45), (3.45), and (3.46), respectively, which are summarized here:

$$\log p\big(\boldsymbol{a}(1), \ldots, \boldsymbol{a}(T)\big) = -\frac{\lambda_{\boldsymbol{a}}}{2}\left(\sum_{f=2}^{T} \|\boldsymbol{a}(f) - \boldsymbol{H}_f \boldsymbol{a}(f-1)\|^2\right) + c_{\boldsymbol{a}}, \tag{5.48}$$

$$\log p\big(\boldsymbol{t}(1), \ldots, \boldsymbol{t}(T)\big) = -\frac{\lambda_{\boldsymbol{t}}}{2}\left(\sum_{f=2}^{T} \|\boldsymbol{t}(f) - \boldsymbol{t}(f-1)\|^2\right) + c_{\boldsymbol{t}}, \tag{5.49}$$

$$\log p(\boldsymbol{b}(1), \ldots, \boldsymbol{b}(T)) = \sum_{f=1}^{T} \log p(\boldsymbol{b}(f)) = -\frac{1}{2}\boldsymbol{b}^{\top}(f) \boldsymbol{\Sigma}^{\boldsymbol{b}^{-1}} \boldsymbol{b}(f) + c_{\boldsymbol{b}}, \tag{5.50}$$

$$\log p(\boldsymbol{p}(1), \ldots, \boldsymbol{p}(T)) = \sum_{f=1}^{T} \log p(\boldsymbol{p}(f)) = c_{\boldsymbol{p}}. \tag{5.51}$$

Note that this formulation implies that only the transformation parameters $\boldsymbol{a}(f)$ and $\boldsymbol{t}(f)$ are correlated between neighboring frames. Without the priors for these parameters, this would be the exact same formulation as in the previous chapters.

5.4.1.2 M-step

In the M-step of the EM algorithm, the idea is to find the model parameters $\boldsymbol{\Theta}$ that maximize the auxiliary function defined in (5.46). The difference with respect to the previous formulations resides in the new prior terms for the transformation parameters.

Update of the Transformation Parameters As previously, the maximization of (5.46) is first performed over the transformation parameters, $\boldsymbol{a}(f), \boldsymbol{t}(f)$, $f = 1, \ldots, T$. The derivatives of the term associated with the likelihood of the observations are the same as the ones shown in (3.32) and (3.33) (for each frame f). The derivatives of the prior terms (5.48) and (5.49) are the following

$$\frac{\delta}{\delta \boldsymbol{a}(f)}\Big(\log p\big(\boldsymbol{a}(1), \ldots, \boldsymbol{a}(T)\big)\Big) = \begin{cases} \lambda_{\boldsymbol{a}} \boldsymbol{H}_{f+1}^{\top}\Big(-\boldsymbol{H}_{f+1}\boldsymbol{a}(f) + \boldsymbol{a}(f+1)\Big), & \text{if } f = 1 \\ \lambda_{\boldsymbol{a}}\Big(-\boldsymbol{a}(f) + \boldsymbol{H}_f \boldsymbol{a}(f-1)\Big), & \text{if } f = T \\ \lambda_{\boldsymbol{a}}\Big(-\big(\boldsymbol{H}_{f+1}^{\top}\boldsymbol{H}_{f+1} + \boldsymbol{I}\big)\boldsymbol{a}(f) + \\ \quad + \boldsymbol{H}_f \boldsymbol{a}(f-1) + \boldsymbol{H}_{f+1}^{\top}\boldsymbol{a}(f+1)\Big) & \text{otherwise,} \end{cases} \tag{5.52}$$

$$\frac{\delta}{\delta \boldsymbol{t}(f)}\Big(\log p\big(\boldsymbol{t}(1), \ldots, \boldsymbol{t}(T)\big)\Big) = \begin{cases} \lambda_{\boldsymbol{t}}\Big(-\boldsymbol{t}(f) + \boldsymbol{t}(f+1)\Big), & \text{if } f = 1 \\ \lambda_{\boldsymbol{t}}\Big(-\boldsymbol{t}(f) + \boldsymbol{t}(f-1)\Big), & \text{if } f = T \\ \lambda_{\boldsymbol{t}}\Big(-2\boldsymbol{t}(f) + \boldsymbol{t}(f-1) + \boldsymbol{t}(f+1)\Big), & \text{otherwise,} \end{cases} \tag{5.53}$$

Note that for the extrema of the sequence, since the MR sequence represents one cardiac cycle, the parameter dependencies could also be considered cyclic, meaning that, for $f = 1$, there would be a term related to $f = T$ in place of the $f - 1$ term, and for $f = T$, there would be a term related to $f = 1$ in place of the $f + 1$ term.

The update equations for these parameters in a generic iteration t of the EM algorithm can be compactly written in matrix form as

$$\left(\begin{bmatrix} \boldsymbol{M}(1) & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}(2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{M}(T) \end{bmatrix} + \boldsymbol{R}\right) \begin{bmatrix} \boldsymbol{a}(1) \\ \boldsymbol{t}(1) \\ \vdots \\ \boldsymbol{a}(T) \\ \boldsymbol{t}(T) \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}(1) \\ \vdots \\ \boldsymbol{c}(T) \end{bmatrix} \tag{5.54}$$

where $\boldsymbol{M}(f)$ and $\boldsymbol{c}(f)$ correspond to (5.28) and (5.29), respectively, and $\boldsymbol{R} \in \mathbb{R}^{4T \times 4T}$ is the regularization matrix, related to the priors, composed of overlapping blocks along the diagonal. A generic block element, $\boldsymbol{R}(f) \in \mathbb{R}^{12 \times 12}$, corresponding to frame f (and overlapping with frames $f - 1$ and $f + 1$) is obtained from converting (5.52) and (5.53) to

matrix notation, leading to

$$\boldsymbol{R}(f) = \begin{bmatrix} \lambda_{\boldsymbol{a}}\left(\boldsymbol{H}_f^\top \boldsymbol{H}_f + \boldsymbol{I}\right) & \boldsymbol{0} & -\lambda_{\boldsymbol{a}}\boldsymbol{H}_f^\top & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & 2\lambda_{\boldsymbol{t}}\boldsymbol{I} & \boldsymbol{0} & -\lambda_{\boldsymbol{t}}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ -\lambda_{\boldsymbol{a}}\boldsymbol{H}_f & \boldsymbol{0} & \lambda_{\boldsymbol{a}}\left(\boldsymbol{H}_{f+1}^\top \boldsymbol{H}_{f+1} + \boldsymbol{I}\right) & \boldsymbol{0} & -\lambda_{\boldsymbol{a}}\boldsymbol{H}_{f+1}^\top & \boldsymbol{0} \\ \boldsymbol{0} & -\lambda_{\boldsymbol{t}}\boldsymbol{I} & \boldsymbol{0} & 2\lambda_{\boldsymbol{t}}\boldsymbol{I} & \boldsymbol{0} & -\lambda_{\boldsymbol{t}}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & -\lambda_{\boldsymbol{a}}\boldsymbol{H}_{f+1} & \boldsymbol{0} & \lambda_{\boldsymbol{a}}\left(\boldsymbol{H}_{f+2}^\top \boldsymbol{H}_{f+2} + \boldsymbol{I}\right) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\lambda_{\boldsymbol{t}}\boldsymbol{I} & \boldsymbol{0} & 2\lambda_{\boldsymbol{t}}\boldsymbol{I} \end{bmatrix}, \tag{5.55}$$

where $\boldsymbol{I}$ is a 2×2 identity matrix.

Update of the Deformation Parameters The update of the deformation parameters, $\boldsymbol{b}(f)$, $f = 1, \ldots, T$, remains the same as in the previous formulation (see equation (3.35)), for each frame f, since there is no new prior information being used that correlates these parameters in different frames.

Update of the Models Probabilities Regarding the probabilities of the valid and invalid models, $\boldsymbol{p}(f)$, $f = 1, \ldots, T$, the situation is similar to the deformation coefficients: the previous update equations, (3.36) and (3.37), hold for each frame f.

The only difference between this new formulation and the previous ones lies in the update equation for the transformation parameters, which now includes an additional regularization term, $\boldsymbol{R}$, that imposes the expected LV dynamics.

5.5 Experimental Setup

The evaluation of the two methods described in this chapter was performed on the dataset of 33 sequences described in Section 3.6.1, using the volumetric Dice coefficient (d_{Dice}), the average minimum distance (d_{AV}), the volume difference, d_{VD}, and the percentage of good segmentations, described in Section 3.6.2.

The manual segmentations provided in the database were used in the training and test phase using a leave-one-sequence-out cross-validation strategy. This means that, for each test sequence, the shape model, the vector fields and the prior parameter information were learned using the remaining 32 sequences. This process is then repeated for each test sequence.

The two proposed approaches were implement in both the 2D and 3D settings. In each setting, results are shown for the two approaches proposed in this chapter: VF and parameter regularization (PR). The results are compared with the previous formulations that do not use temporal information.

Table 5.1: Statistical performance of EM-RASM using: (1) no temporal information, (2) VFs, and (3) PR. The bold values indicate the best result on each column.

	Temporal information	d_{Dice} (%)	d_{AV} (mm)	d_{VD} (mL)	% Good
2D	**None**	84.2 (9.2)	2.6 (1.7)	11.8 (11.4)	89.2
	VF	84.5 (8.2)	2.5 (1.5)	12.3 (12.2)	89.7
	PR	**86.0 (6.9)**	**2.3 (1.0)**	10.2 (8.5)	**93.0**
3D	**None**	84.6 (6.8)	2.5 (1.2)	**9.2 (9.4)**	89.1
	VF	84.6 (7.0)	2.5 (1.3)	9.5 (9.5)	90.3
	PR	85.3 (7.0)	2.3 (1.2)	9.4 (8.6)	91.5

The shape model was learned using the approaches described in Chapters 3 and 4, using the same number of deformation modes as before, *i.e.*, $L = 10$ in the 2D formulation and $L = 10 + 2$ (with TDM) in the 3D formulation.

Several tests were performed for each setup in order to determine the best parameters, namely: the weight of the VF observations, $\lambda_V \in]0, 1]$, used in (5.26), and the weight of the parameter priors, $\lambda_{\boldsymbol{a}}, \lambda_{\boldsymbol{t}} \in]0, 1]$, used in (5.43) and (5.45), respectively. The parameters that led to the best results were the following: $\lambda_V = 0.25$, $\lambda_{\boldsymbol{a}} = \lambda_{\boldsymbol{t}} = 0.5$ for the 2D formulation; and $\lambda_V = 0.5$, $\lambda_{\boldsymbol{a}} = \lambda_{\boldsymbol{t}} = 0.2$ for the 3D formulation.

5.6 Results

This section shows examples of the segmentations obtained and presents statistical results of the accuracy of each combination of approaches.

Table 5.1 summarizes the quantitative results obtained using the proposed approaches. It is possible to see that the statistical results are similar across all formulations. However, the best overall results are obtained using PR, both in the 2D and the 3D setup.

It is unclear from the results in the table if there is any real benefit in using VFs as a way to introduce temporal information. But taking a closer look at the results for each individual volume, shown in Fig. 5.10, it is possible to see that there is indeed a

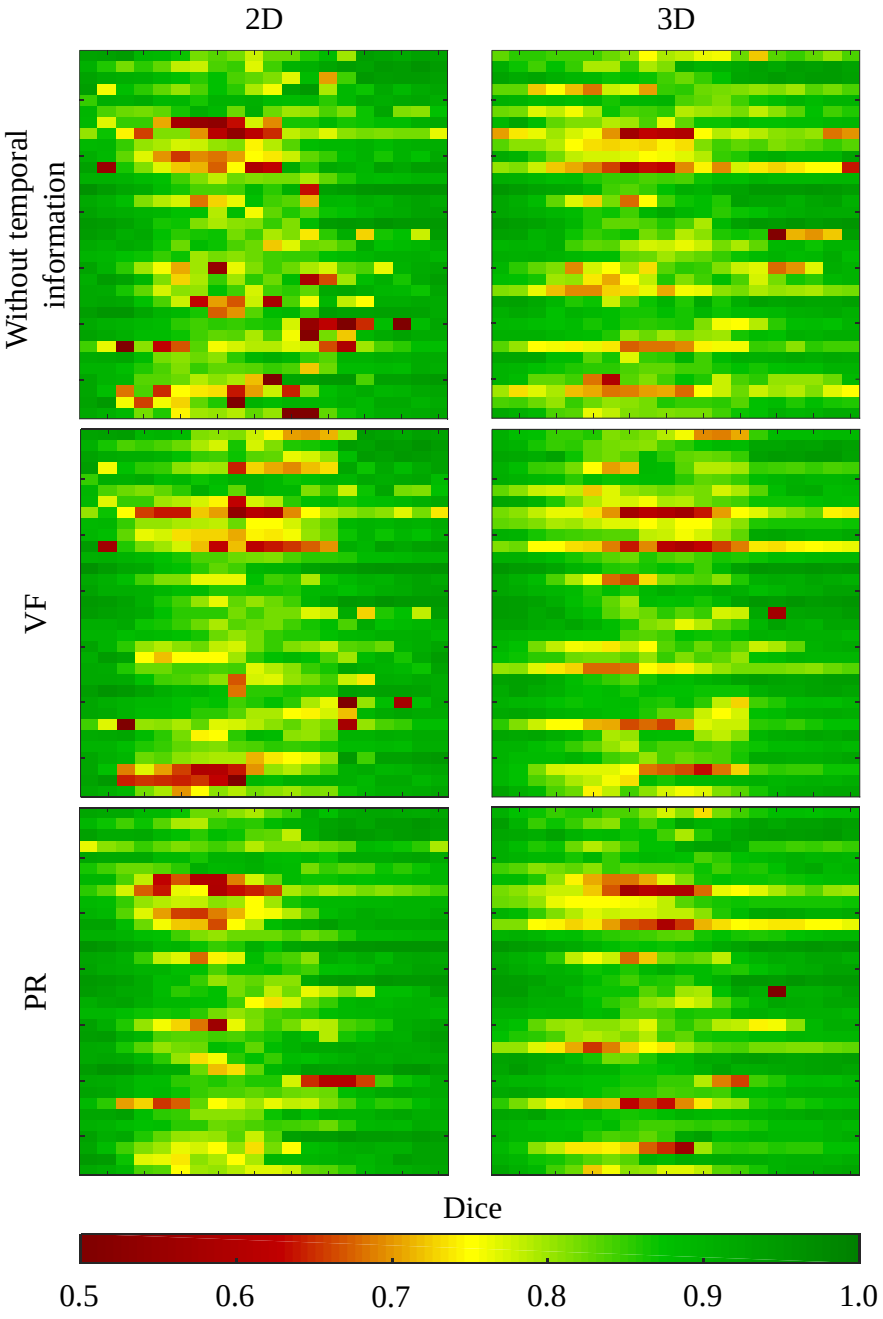


Figure 5.10: Quantitative evaluation of all the volumes using the Dice coefficient. The vertical axis corresponds to different patients and the horizontal axis to different phases of the cardiac cycle. The colors are defined by the colorbar in the bottom (greener is better).

decrease in poor segmentations using VF, compared to not using temporal information (compare the amount of red pixels between the first and the second row). Despite this difference, Table 5.1 shows that, on average, they have similar performances. This suggests that even though introducing VFs to impose temporal constraints on the segmentations helps to prevent some mistakes, these constraints also prevent the model from fitting the LV accurately in some other cases. Regarding the use of regularization, it is reasonable to conclude that there is a clear benefit in constraining the model parameters with prior

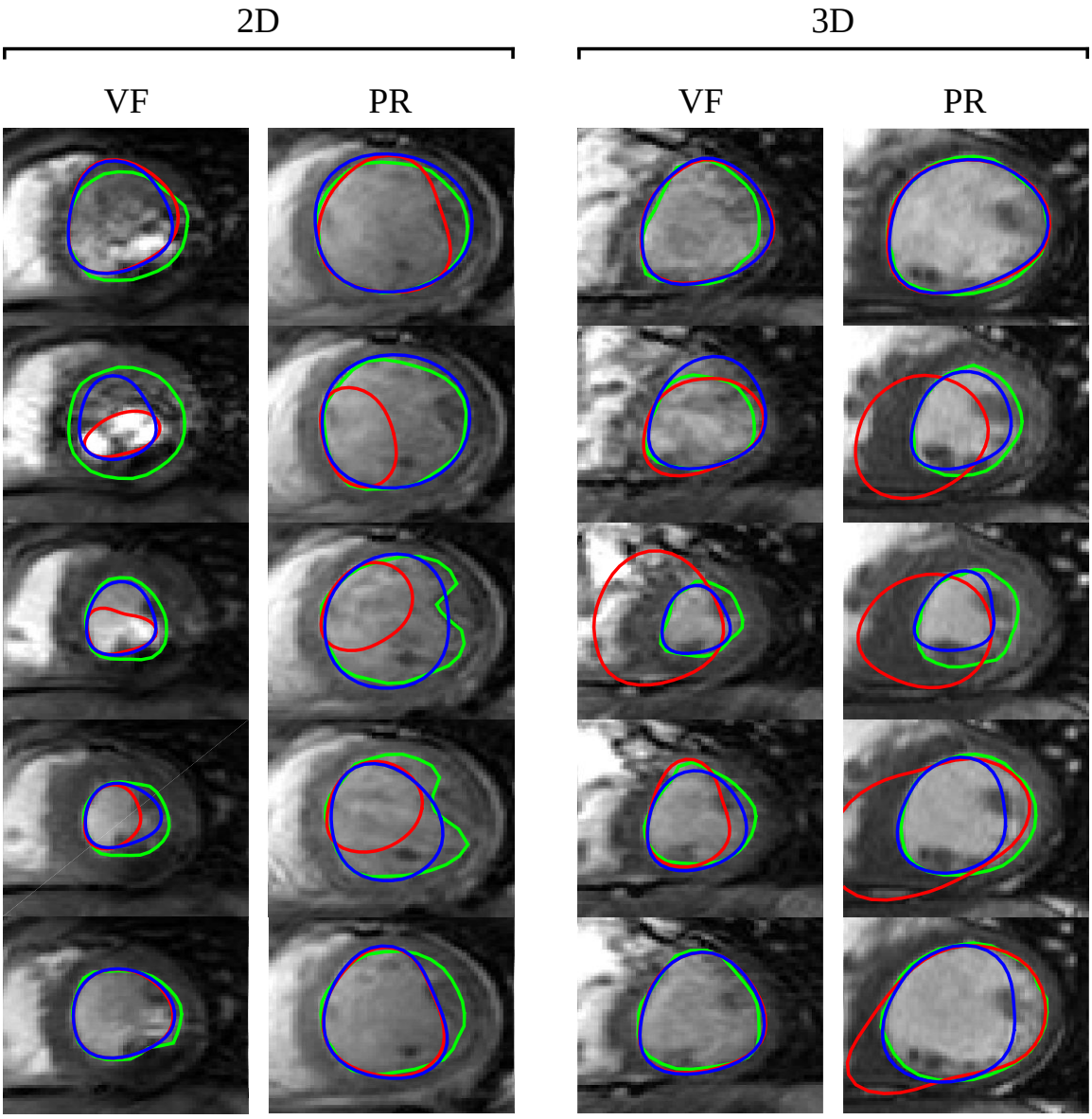


Figure 5.11: Examples of segmentations that were significantly improved by using temporal information. Each column shows a different example and each row corresponds to a different frame. The GT is shown in green, the segmentations obtained without temporal information in red and the ones obtained with temporal information in blue. The first two columns show examples for the 2D formulation and the last two columns for the 3D formulation. For each setup, the left example shows improvements using the VF approach and the right example using the PR approach.

temporal information. It allows avoiding many gross errors, as shown by the significant reduction of red pixels in Fig. 5.10, while still allowing the model to fit the LV border accurately. This is valid for both the 2D and the 3D formulation presented in this thesis. For a qualitative evaluation of the proposed methods, Fig. 5.11 shows examples in which the use of temporal information led to an improvement in the quality of the segmentations.

It is also interesting to note that some sequences are poorly segmented in all the

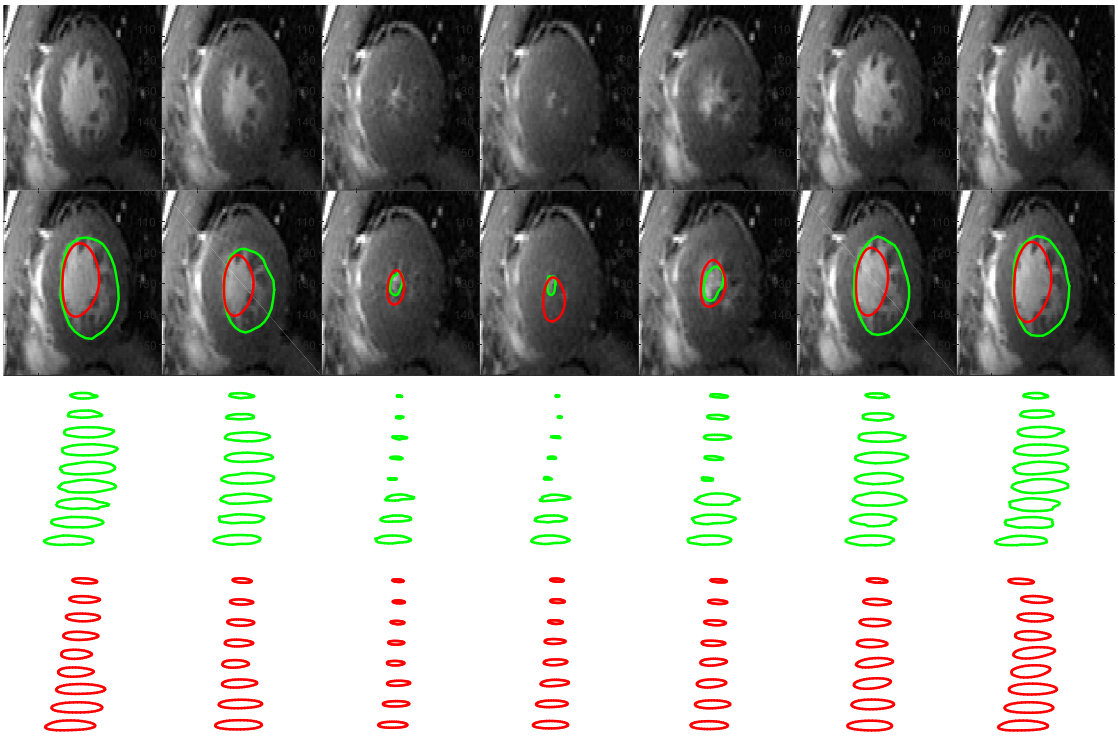


Figure 5.12: Example of a difficult MR sequence. The green contours correspond to the GT and the red contours correspond to the segmentations obtained with the 3D setup with PR. The first two rows show a particular slice of the sequence in frames 1,5,9,13 and 17 with and without segmentations, and the last two rows show the segmentations for the corresponding volumes.

approaches. An example of such a sequence is shown in Fig. 5.12. In this example, the LV border is barely visible, due to the presence of papillary muscles, particularly in the systolic phase (third, fourth and fifth image). The corresponding GT segmentations in these frames (third row) depict an unusual 3D LV shape, in which the area of the segmentations in the first three frames is significantly larger than in the remaining slices. This means that it is not possible to rely neither on the image to accurately identify the LV border, nor on the 3D shape of the LV for guidance. Temporal information, on the other hand, is also unable to compensate for this sort of behavior in the systolic phase. Thus, it is impossible for the proposed methodologies to deal with this example, and other alike.

5.6.1 Comparison With the State of the Art

Statistical results of the proposed segmentations are compared to other state-of-the-art methods in the boxplots shown in Fig. 5.13. This comparison is based on the reported results from the following works: [11, 15, 16, 41, 49, 59, 61, 69, 87, 98, 100, 103, 106, 107, 118, 130, 131, 139, 141, 160, 166, 177, 185, 185, 188]. It is important to note the following facts: i) most of these works use a different dataset, making a direct comparison possibly unfair;

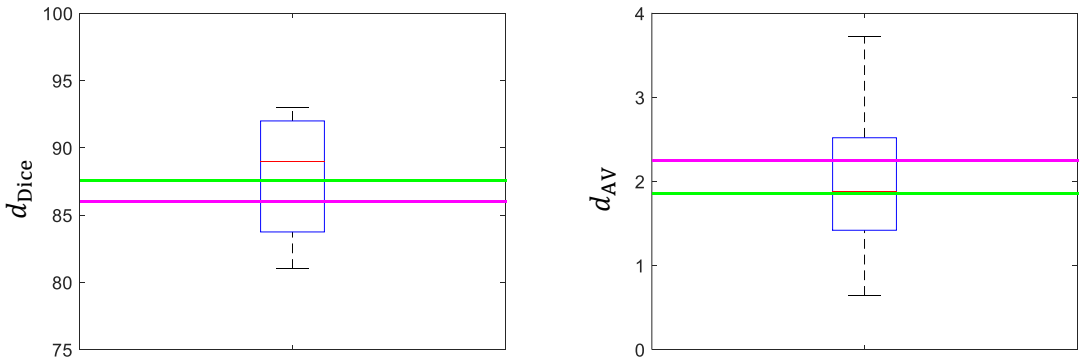


Figure 5.13: Comparison of the performance of EM-RASM with regularization against other state of the art approaches. The green line shows the average performance excluding the bad segmentations (*i.e.*, with $d_{\text{AV}} > 5$ mm) and the magenta line shows the average performance including all the segmentations.

ii) some of them only analyze specific slices (*e.g.*, only the mid slice [15, 141]), or specific frames (*e.g.*, only the ES and ED frames [59, 61, 87, 103]); iii) exclude gross errors from the performance statistics [118]; iv) exclude volumes with misalignments [87]; and v) use only a subset of the dataset used in this thesis [49, 139]. It should also be noted that not all the works use both the d_{Dice} and the d_{AV} metrics to quantitatively evaluate the segmentations. In this selections of works, 22 report the d_{AV} metric, while only 13 report the d_{Dice} metric.

Fig. 5.13 shows the results obtained using the proposed EM-RASM with PR (magenta line) and the results using the same algorithm excluding the segmentations with $d_{\text{AV}} > 5$ mm (green line), *i.e.*, the gross errors, which correspond to excluding 7% of the segmentations. Comparing the results with the state of the art shows that the performance of the proposed approach is approximately in the average values report in the literature.

Since these state of the art methods use different approaches to obtain the LV segmentation, evaluate their approaches on different (and often private) dataset, and that the reported values have a large variation, it is difficult to make any specific conclusions about which approaches are more successful and what each approach is lacking.

5.7 Conclusion

Temporal information has the potential to significantly improve the accuracy of segmentations, by exploiting the correlation between the position of the shape model in different phases of the cardiac cycle. This is particularly useful in cases where the LV border is not easily identifiable, because it allows the algorithm to disregard the information extracted from the image and rely on the expected heart motion.

This chapter considers two different approaches to include dynamical information for

the LV segmentation: 1) using two VFs to explicitly describe the motion of the LV border as displacements of each contour point; and 2) using regularization on the shape model parameters, in which the LV dynamics are described indirectly through changes in these parameters. Both of these approaches are able to capture the LV dynamics and provide a powerful means of including temporal information in the segmentation algorithm.

Regarding the VF approach, two 3D models were used: one for the systolic phase and one for the diastolic phase. These VFs were incorporated in the EM-RASM formulation by introducing an additional term in the optimization problem that consists in also minimizing the distance between the model points and the their expected position based on the the corresponding VF. This additional term acts as new observation points, which are extracted not from the image, but rather from the model position in the previous frame. In this way, the algorithms combines the information extracted from the image with a bias from the expected motion given by the VF. Results showed that this approach has the ability to improve some of the gross errors obtained with the approaches described in the previous chapters. However, the VF bias also constrains the segmentations in such a way that it prevents the shape model from accurately fitting the LV border in other cases. Overall, the average segmentation accuracy using VFs leads to a small reduction in the standard deviation of the statistical results (in the 2D setting) and also a small improvement in the percentage of "good" segmentations (in the 3D setting).

The second approach described in this chapter is based on the introduction of prior information about the expected shape model parameters, namely, in the parameters of the similarity transformation of the shape model (related to the pose of the LV). This approach is seamlessly coupled with the MAP formulation described in Chapter 3 that previously considered these priors to be non-informative (obtained using Jeffreys' priors). In this case, instead of using Jeffreys' priors, Gaussian priors were used, which induce a correlation between the parameters in consecutive frames. This correlation is learned from a training set to embed information about the expected LV motion. In this new formulation, temporal information is naturally introduced in the EM-RASM algorithm, and leads to new update equations that impose constraints on the similarity between the shape model parameters in consecutive frames. The results show that this approach improves the overall segmentation accuracy, and is able to avoid many of the mistakes obtained using the previous formulations.

The comparison with the state of the art showed that the results obtained with the proposed approach are within the average of what other methods have accomplished over the years. However, a more detailed comparison would require using a larger common dataset. This has been the effort of some research groups (*e.g.*, the Cardiac Altas Project [5] [167]), although it is not yet used as the baseline with which new methods are compared.

For future work, it would be interesting to explore the combination of the two

[5] `http://www.cardiacatlas.org/`

approaches discussed in this chapter, since they explore two different ways of embedding temporal information. On one hand, the VFs are able to express local deformations of the LV contours, since the displacements they induce affect not only the pose of the LV but also the deformation coefficients of the shape model. On the other hand, since the VFs depend on the position of the center of the LV, they are not able to prevent the segmentations from drifting way from the LV border. However, the PR approach may be able to deal with this issue by imposing that the translation parameter should be similar across all the frames, and to its initial guess.

6

Conclusions and Future Work

Contents

6.1 Conclusions

The problem of automatic segmentation of the LV in CMR data was addressed in this thesis. The work developed was divided into three subproblems: 1) the segmentation of the LV in a 2D approach, using an ASM-based methodology (Chapter 3); 2) the segmentation of the LV as a 3D structure, which explored the advantages of the 2D approach with information about the three-dimensional structure of the LV (Chapter 4); and 3) the segmentation of MR sequences (2D+t and 3D+t), in which the goal was to include information about the motion of the LV throughout the cardiac cycle (Chapter 5).

Each of these subproblems faced specific challenges. In Chapter 3, each MR slice was independently segmented. Therefore, it focused on identifying the location of the LV border, and preventing other anatomical structures, such as papillary muscles, from misguiding the segmentation algorithm. A 2D shape model was used for its ability to impose shape constraints in a simple and flexible framework that has been extensively used in the literature. The main assumption behind this approach was that a single 2D shape model would be able to describe the LV contour in any MR image, regardless of which slice or cardiac phase it corresponded to. Contributions were made to the field by proposing a new formulation, EM-RASM, for the estimation of the shape model parameters in images prone to outliers. The results showed that the proposed framework outperformed other related works and was able to achieve more accurate segmentations. Although the focus of this contribution was the segmentation of CMR images, the proposed framework is not limited to this particular application, and the results in other unrelated applications showed great potential.

In Chapter 4, the goal was to extend the EM-RASM framework to a 3D setting. The inclusion of the 3D information came at the expense of two additional issues: the variable number of slices in the MR volumes; and the existence of misaligned slices. These issues posed new challenges in both the training phase and the test phase of the shape model pipeline. First, it was necessary to pre-process the training data in order to guarantee that the shape examples were all characterized by the same number of landmarks and two remove misalignments before the computation of the shape statistics. Then, a novel resampling framework for the learned shape model was proposed, to allow the shape model to segment new MR volumes with a different number of slices, without resorting to the interpolation of voxel intensities. Finally, the inclusion of specific deformation modes was explored to deal with misaligned slices in the test volumes. The results showed that the 3D approach was able to improve the accuracy of the 2D method, and in particular to significantly decrease the variability of the results between slices of the same MR volume.

In Chapter 5, we explored two approaches to embed temporal information into the segmentation process. The first approach was based on describing the LV dynamics through VFs. In the proposed methodology, two VFs were used: one for the systolic

phase, which represented the contraction of the LV, and one for the diastolic phase, to represent the dilation of the LV. These VFs were added to the EM-RASM framework by penalizing large deviations from the expected motion pattern. In the second approach, PR was used to impose temporal constraints to the segmentations. This approach was easily introduced in the EM-RASM formulation through the prior terms associated with the shape model parameters. In this case, the priors acted as regularizers that forced the model parameters to follow the expected variation pattern along the cardiac cycle. These two approaches were tested in the 2D and 3D settings. Comparing the performance of all the methodologies proposed in this thesis, using PR was the approach the achieved better segmentation accuracies.

A comparison with the state of the art showed that the performance of EM-RASM with PR lies in the average of the results reported in the literature. However, a direct comparison may not be fair, as each work uses its own dataset and in specific conditions. Therefore, this comparison only gives an idea of where the proposed approach is place in the state of the art, but it is not possible to extract any conclusion about why and where other works perform better or worse.

6.2 Future Work

The main challenge in LV segmentations algorithms lie in accurately determining the location of the LV border. Even using sophisticated approaches such as EM-RASM, it is clear that the algorithm is still misguided by papillary muscles and other structures. In particular, methods that rely on edge detection are bound to struggle with this issue. Although using additional information from the 3D structure of the LV and its temporal dynamics helps reduce the number of segmentation failures, as shown in this thesis, it does not actually solve the problem, only alleviates it. Therefore, it is necessary to focus on new strategies to address this issue.

Chapter 3 focused on proposing a robust method to estimate the parameters of a shape model, from observations extracted from the images using a simple technique, and the remaining chapters are built on top of this approach. This leaves room for improvement by using robust methods to extract candidate locations of the LV border from the image. Active appearance models have been a popular trend in the last decade and several works have reported remarkable results [11, 138]. This indicates that it may be necessary to model the intensity of the LV as an alternative approach to identify the LV border. More recently, deep learning approaches have been outperforming many established algorithms in other application [92]. In CMR, some works have started to explore the potential of these methods in the segmentation of the LV [14, 105, 127, 189]. Advances in segmentation performance will greatly rely on these and other approaches to more accurately determine the location of the LV border in CMR images.

Finally, it would be advantageous to evaluate the work developed in this thesis using a larger, public dataset for a fair and informative comparison to be possible. Although the dataset used in this thesis is available for research purposes, there are not many works reporting their results on this data. Other dataset have also been made available, mainly associated with specific challenges, *e.g.*, MICCAI 2009 [3], MICCAI 2011 [2, 167] and Kaggle 2016 [5], and there is an ongoing effort by the Cardiac Altas Project[1] to establish a dataset for benchmarking. Fostering these initiatives will contribute to better evaluate and compare different approaches and their advantages and disadvantages.

[1] `http://www.cardiacatlas.org/`

Bibliography

[1] "Black Blood Double IR," http://mriquestions.com/dark-blooddouble-ir.html, accessed: 2017-02-16.

[2] "Cardiac Atlas Project: LV Segmentation Challenge," http://www.cardiacatlas.org/challenges/lv-segmentation-challenge/, accessed: 2017-02-23.

[3] "Cardiac MR Left Ventricle Segmentation Challenge," http://smial.sri.utoronto.ca/LV_Challenge/Home.html, accessed: 2017-02-23.

[4] "Gating vs Triggering," http://mriquestions.com/gating-v-triggering.html, accessed: 2017-02-16.

[5] "Second Annual Data Science Bowl: Transforming How We Diagnose Heart Disease," https://www.kaggle.com/c/second-annual-data-science-bowl, accessed: 2017-02-23.

[6] "World Health Organization: Cardiovascular Diseases," http://www.who.int/mediacentre/factsheets/fs317/en/, accessed: 2015-06-23.

[7] J. Abi-Nahed, M.-P. Jolly, and G.-Z. Yang, "Robust active shape models: A robust, generic and simple automatic segmentation tool," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*. Springer, 2006, pp. 1–8.

[8] P. Abolmaesumi and M. R. Sirouspour, "An interacting multiple model probabilistic data association filter for cavity boundary extraction from ultrasound images," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 6, pp. 772–784, 2004.

[9] A. H. Aletras, S. Ding, R. S. Balaban, and H. Wen, "DENSE: displacement encoding with stimulated echoes in cardiac functional MRI," *Journal of Magnetic Resonance*, vol. 137, no. 1, pp. 247–252, 1999.

[10] A. A. Amini and J. L. Prince, *Measurement of cardiac deformations from MRI: physical and mathematical models*. Springer Science & Business Media, 2013, vol. 23.

[11] A. Andreopoulos and J. K. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI," *Medical Image Analysis*, vol. 12, no. 3, pp. 335–357, 2008.

[12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, 2011.

[13] O. Arif, G. Sundaramoorthi, B. W. Hong, and A. Yezzi, "Tracking Using Motion Estimation With Physically Motivated Inter-Region Constraints," *IEEE Transactions on Medical Imaging*, vol. 33, no. 9, pp. 1875–1889, Sept 2014.

[14] M. Avendi, A. Kheradvar, and H. Jafarkhani, "A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI," *Medical image analysis*, vol. 30, pp. 108–119, 2016.

[15] I. B. Ayed, S. Li, and I. Ross, "Embedding overlap priors in variational left ventricle tracking," *IEEE Transactions on Medical Imaging*, vol. 28, no. 12, pp. 1902–1913, 2009.

[16] W. Bai, W. Shi, D. P. O'Regan, T. Tong, H. Wang, S. Jamil-Copley, N. S. Peters, and D. Rueckert, "A probabilistic patch-based label fusion model for multi-atlas segmentation with registration refinement: application to cardiac MR images," *IEEE transactions on medical imaging*, vol. 32, no. 7, pp. 1302–1315, 2013.

[17] R. S. Balaban and D. C. Peters, "Basic Principles of Cardiovascular Magnetic Resonance," in *Cardiovascular Magnetic Resonance*, W. J. Manning and D. J. Pennell, Eds. Philadelphia: Churchill Livingstone, 2010, pp. 3 – 18.

[18] Y. Bar-Shalom, *Tracking and data association*. Academic Press Professional, Inc., 1988.

[19] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and data fusion*. YBS publishing, 2011.

[20] G. Behiels, D. Vandermeulen, F. Maes, P. Suetens, and P. Dewaele, "Active shape model-based segmentation of digital X-ray images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI'99*. Springer, 1999, pp. 128–137.

[21] N. G. Bellenger, N. J. Marcus, K. Rajappan, M. Yacoub, N. R. Banner, and D. J. Pennell, "Comparison of Techniques for the Measurement of Left Ventricular Function Following Cardiac Transplantation: Ventricular Function," *Journal of Cardiovascular Magnetic Resonance*, vol. 4, no. 2, pp. 255–263, 2002.

[22] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[23] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[24] F. Billet, M. Sermesant, H. Delingette, and N. Ayache, "Cardiac motion recovery and boundary conditions estimation by coupling an electromechanical model and cine-MRI data," in *Functional Imaging and Modeling of the Heart*. Springer, 2009, pp. 376–385.

[25] A. Blake and M. Isard, "Image processing techniques for feature location," in *Active Contours*. Springer, 1998, pp. 97–113.

[26] J. Bogaert, "Cardiac Function," in *Clinical Cardiac MRI*, ser. Medical Radiology, J. Bogaert, S. Dymarkowski, A. M. Taylor, and V. Muthurangu, Eds. Springer Berlin Heidelberg, 2012, pp. 109–165.

[27] J. Bogaert, S. Dymarkowski, A. Taylor, and V. Muthurangu, *Clinical Cardiac MRI*, ser. Medical Radiology / Diagnostic Imaging. Springer Berlin Heidelberg, 2012.

[28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[29] O. Buckley, R. Madan, R. Kwong, F. J. Rybicki, and A. Hunsaker, "Cardiac masses, part 1: imaging strategies and technical considerations," *American Journal of Roentgenology*, vol. 197, no. 5, pp. W837–W841, 2011.

[30] C. Casta, P. Clarysse, J. Schaerer, and J. Pousin, "Evaluation of the dynamic deformable elastic template model for the segmentation of the heart in MRI sequences," *MIDAS J-Card MR Left Ventricle Segmentation Challenge*, 2009.

[31] A. Caunce and C. J. Taylor, "Building 3D sulcal models using local geometry," *Medical Image Analysis*, vol. 5, no. 1, pp. 69–80, 2001.

[32] H. Chen and B. Bhanu, "Shape model-based 3D ear detection from side face range images," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*. IEEE, 2005, pp. 122–122.

[33] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299–309, 2005.

[34] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114–141, 2003.

[35] C. A. Cocosco, W. J. Niessen, T. Netsch, E.-j. Vonken, G. Lund, A. Stork, and M. A. Viergever, "Automatic image-driven segmentation of the ventricles in cardiac cine MRI," *Journal of Magnetic Resonance Imaging*, vol. 28, no. 2, pp. 366–374, 2008.

[36] T. F. Cootes, M. C. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 278–291.

[37] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[38] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[39] T. F. Cootes and C. J. Taylor, "A mixture model for representing shape variation," *Image and Vision Computing*, vol. 17, no. 8, pp. 567–573, 1999.

[40] L. Cordero-Grande, J. Royuela-del Val, S. Sanz-Estébanez, M. Martín-Fernández, and C. Alberola-López, "Multi-oriented windowed harmonic phase reconstruction for robust cardiac strain imaging," *Medical image analysis*, vol. 29, pp. 1–11, 2016.

[41] J. Cousty, L. Najman, M. Couprie, S. Clément-Guinaudeau, T. Goissen, and J. Garot, "Segmentation of 4D cardiac MRI: Automated method based on spatio-temporal watershed cuts," *Image and Vision Computing*, vol. 28, no. 8, pp. 1229–1243, 2010.

[42] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1262–1273, 2006.

[43] D. Cristinacce and T. F. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008.

[44] C. Davatzikos, X. Tao, and D. Shen, "Hierarchical active shape models, using the wavelet transform," *IEEE transactions on medical imaging*, vol. 22, no. 3, pp. 414–423, 2003.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[46] J. Dias and J. Leitão, "Wall position and thickness estimation from sequences of echocardiographic images," *IEEE Transactions on Medical Imaging*, vol. 15, no. 1, pp. 25–38, 1996.

[47] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[48] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE transactions on medical imaging*, vol. 29, no. 3, pp. 852–867, 2010.

[49] J. Ehrhardt, T. Kepp, A. Schmidt-Richberg, and H. Handels, "Joint multi-object registration and segmentation of left and right cardiac ventricles in 4D cine MRI," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2014, pp. 90 340M–90 340M.

[50] M. A. T. Figueiredo, "Lecture Notes on Bayesian Estimation and Classification," http://www.lx.it.pt/~mtf/learning/Bayes_lecture_notes.pdf, 2004, accessed: 2017-03-07.

[51] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[52] R. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Trans. of the Royal Society of London (A)*, vol. 222, no. 594-604, pp. 309–368, 1922.

[53] A. F. Frangi, D. Rueckert, J. A. Schnabel, and W. J. Niessen, "Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1151–1166, 2002.

[54] D. Fritz, D. Rinck, R. Dillmann, and M. Scheuering, "Segmentation of the left and right cardiac ventricle using a combined bi-temporal statistical model," in *Medical Imaging*. International Society for Optics and Photonics, 2006, pp. 614 121–614 121.

[55] J. Fu, J. Chai, and S. T. Wong, "Wavelet-based enhancement for detection of left ventricular myocardial boundaries in magnetic resonance images," *Magnetic resonance imaging*, vol. 18, no. 9, pp. 1135–1141, 2000.

[56] B. Gao, W. Liu, L. Wang, Z. Liu, P. Croisille, P. Delachartre, and P. Clarysse, "Estimation of cardiac motion in cine-MRI sequences by correlation transform optical flow of monogenic features distance," *Physics in Medicine and Biology*, vol. 61, no. 24, p. 8640, 2016.

[57] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 294–302, 1995.

[58] H. Geyer, G. Caracciolo, H. Abe, S. Wilansky, S. Carerj, F. Gentile, H.-J. Nesser, B. Khandheria, J. Narula, and P. P. Sengupta, "Assessment of myocardial mechanics using speckle tracking echocardiography: fundamentals and clinical applications," *Journal of the American Society of Echocardiography*, vol. 23, no. 4, pp. 351–369, 2010.

[59] S. Gopal and D. Terzopoulos, "A Unified Statistical/Deterministic Deformable Model for LV Segmentation in Cardiac MRI," in *Statistical Atlases and Computational Models of the Heart. Imaging and Modelling Challenges*. Springer, 2014, pp. 180–187.

[60] J. Gorcsan and H. Tanaka, "Echocardiographic assessment of myocardial strain," *Journal of the American College of Cardiology*, vol. 58, no. 14, pp. 1401–1413, 2011.

[61] D. Grosgeorge, C. Petitjean, J. Caudron, J. Fares, and J.-N. Dacher, "Automatic cardiac ventricle segmentation in MR images: a validation study," *International journal of computer assisted radiology and surgery*, vol. 6, no. 5, pp. 573–581, 2011.

[62] C. Guetter, H. Xue, C. Chefd'Hotel, and J. Guehring, "Efficient symmetric and inverse-consistent deformable registration through interleaved optimization," in *2011 IEEE international symposium on biomedical imaging: from nano to macro*. IEEE, 2011, pp. 590–593.

[63] G. Hamarneh and T. Gustavsson, "Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images," in *Computers in Cardiology 2000*. IEEE, 2000, pp. 115–118.

[64] G. Hamarneh and T. Gustavsson, "Deformable spatio-temporal shape models: extending active shape models to 2D+ time," *Image and Vision Computing*, vol. 22, no. 6, pp. 461–470, 2004.

[65] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 114.

[66] T. Heimann and H. Meinzer, "Statistical shape models for 3D medical image segmentation: A review," *Medical image analysis*, vol. 13, no. 4, pp. 543–563, 2009.

[67] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[68] H. Hu, H. Liu, Z. Gao, and L. Huang, "Hybrid segmentation of left ventricle in cardiac MRI using gaussian-mixture model and region restricted dynamic programming," *Magnetic resonance imaging*, vol. 31, no. 4, pp. 575–584, 2013.

[69] S. Huang, J. Liu, L. C. Lee, S. K. Venkatesh, L. L. San Teo, C. Au, and W. L. Nowinski, "An image-based comprehensive approach for automatic segmentation of left ventricle from cardiac short axis cine MR images," *Journal of digital imaging*, vol. 24, no. 4, pp. 598–608, 2011.

[70] P. J. Huber, *Robust statistics*. Wiley, New York, 1981.

[71] W. G. Hundley, D. A. Bluemke, J. P. Finn, S. D. Flamm, M. A. Fogel, M. G. Friedrich, V. B. Ho, M. Jerosch-Herold, C. M. Kramer, W. J. Manning *et al.*, "ACCF/ACR/AHA/NASCI/SCMR 2010 expert consensus document on cardiovascular magnetic resonance: a report of the American College of Cardiology Foundation Task Force on Expert Consensus Documents," *Journal of the American College of Cardiology*, vol. 55, no. 23, pp. 2614–2662, 2010.

[72] P. Hunter, A. McCulloch, and H. Ter Keurs, "Modelling the mechanical properties of cardiac muscle," *Progress in biophysics and molecular biology*, vol. 69, no. 2, pp. 289–331, 1998.

[73] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *International Journal of Computer Vision*, vol. 5, no. 2, pp. 195–212, 1990.

[74] E.-S. H. Ibrahim, "Myocardial tagging by Cardiovascular Magnetic Resonance: evolution of techniques–pulse sequences, analysis algorithms, and applications," *Journal of Cardiovascular Magnetic Resonance*, vol. 13, no. 1, p. 36, 2011.

[75] T. Irony and N. Singpurwalla, "Noninformative priors do not exist: A dialogue with José M. Bernardo," *Journal of Statistical Planning and Inference*, vol. 65, pp. 159–189, 1997.

[76] A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 3, pp. 267–278, 1996.

[77] H. Jara, B. Yu, S. Caruthers, E. Melhem, and E. Yucel, "Voxel sensitivity function description of flow-induced signal loss in MR imaging: Implications for black-blood MR angiography with turbo spin-echo sequences," *Magnetic resonance in medicine*, vol. 41, no. 3, pp. 575–590, 1999.

[78] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proc. of the Journal of theof the Royal Society (A)*, vol. 186, pp. 453–461, 1946.

[79] H. Jeffreys, *Theory of Probability*. Oxford University Press, Oxford., 1961.

[80] H. Jeffreys, *The theory of probability*. Oxford University Press, 1998.

[81] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1633–1645, 2011.

[82] M. Jolly, "Fully automatic left ventricle segmentation in cardiac cine MR images using registration and minimum surfaces," *The MIDAS Journal-Cardiac MR Left Ventricle Segmentation Challenge*, vol. 4, 2009.

[83] M.-P. Jolly, C. Guetter, and J. Guehring, "Cardiac segmentation in MR cine data using inverse consistent deformable registration," in *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, 2010, pp. 484–487.

[84] M.-P. Jolly, C. Guetter, X. Lu, H. Xue, and J. Guehring, "Automatic segmentation of the myocardium in cine MR images using deformable registration," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2011, pp. 98–108.

[85] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[86] A. Katok and B. Hasselblatt, *Introduction to the modern theory of dynamical systems*. Cambridge university press, 1997, vol. 54.

[87] M. R. Kaus, J. v. Berg, J. Weese, W. Niessen, and V. Pekar, "Automated segmentation of the left ventricle in cardiac MRI," *Medical Image Analysis*, vol. 8, no. 3, pp. 245–254, 2004.

[88] S. Kay, *Fundamentals of Statistical Signal Processing*. Prentice Hall, Englewood Cliffs, N.J., 1993, vol. I.

[89] U. Kurkure, A. Pednekar, R. Muthupillai, S. D. Flamm, and I. A. Kakadiaris, "Localization and segmentation of left ventricle in cardiac cine-MR images," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 5, pp. 1360–1370, 2009.

[90] A. Lalande, L. Legrand, P. M. Walker, F. Guy, Y. Cottin, S. Roy, and F. Brunotte, "Automatic detection of left ventricular contours from cardiac cine magnetic resonance imaging using fuzzy logic," *Investigative radiology*, vol. 34, no. 3, pp. 211–217, 1999.

[91] M. M. Lamacie, P. Thavendiranathan, K. Hanneman, A. Greiser, M.-P. Jolly, R. Ward, and B. J. Wintersperger, "Quantification of global myocardial function by cine MRI deformable registration-based analysis: Comparison with MR feature tracking and speckle-tracking echocardiography," *European Radiology*, pp. 1–12, 2016.

[92] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[93] H.-Y. Lee, N. C. Codella, M. D. Cham, J. W. Weinsaft, and Y. Wang, "Automatic left ventricle segmentation using iterative thresholding and an active contour model with adaptation on short-axis cardiac MRI," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 4, pp. 905–913, 2010.

[94] K. Lekadir, R. Merrifield, and G.-Z. Yang, "Outlier detection and handling for robust 3-D active shape models search," *Medical Imaging, IEEE Transactions on*, vol. 26, no. 2, pp. 212–222, 2007.

[95] Y. Liew, R. McLaughlin, B. Chan, Y. A. Aziz, K. Chee, N. Ung, L. Tan, K. Lai, S. Ng, and E. Lim, "Motion corrected LV quantification based on 3D modelling for improved functional assessment in cardiac MRI," *Physics in medicine and biology*, vol. 60, no. 7, p. 2715, 2015.

[96] X. Lin, B. R. Cowan, and A. A. Young, "Automated detection of left ventricle in 4D MR images: experience from a large study," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*. Springer, 2006, pp. 728–735.

[97] N. Liu, S. Crozier, S. Wilson, F. Liu, B. Appleton, A. Trakic, Q. Wei, W. Strugnell, R. Slaughter, and R. Riley, "Right ventricle extraction by low level and model-based algorithm," *27th Annual IEEE Engineering in Medicine and Biological Society*, vol. 27, pp. 677–680, 2005.

[98] M. Lorenzo-Valdés, G. I. Sanchez-Ortiz, A. G. Elkington, R. H. Mohiaddin, and D. Rueckert, "Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm," *Medical Image Analysis*, vol. 8, no. 3, pp. 255–265, 2004.

[99] M. Lorenzo-Valdés, G. I. Sanchez-Ortiz, R. Mohiaddin, and D. Rueckert, "Atlas-based segmentation and tracking of 3D cardiac MR images using non-rigid registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2002, pp. 642–650.

[100] J. Lötjönen, K. Antila, E. Lamminmäki, J. Koikkalainen, M. Lilja, and T. Cootes, "Artificial enlargement of a training set for statistical shape models: Application to cardiac images," in *International Workshop on Functional Imaging and Modeling of the Heart*. Springer, 2005, pp. 92–101.

[101] J. Lötjönen, S. Kivistö, J. Koikkalainen, D. Smutek, and K. Lauerma, "Statistical shape model of atria, ventricles and epicardium from short-and long-axis MR images," *Medical image analysis*, vol. 8, no. 3, pp. 371–386, 2004.

[102] J. Lötjönen, M. Pollari, S. Kivistö, and K. Lauerma, "Correction of Motion Artifacts From Cardiac Cine Magnetic Resonance Images," *Academic radiology*, vol. 12, no. 10, pp. 1273–1284, 2005.

[103] Y. Lu, P. Radau, K. Connelly, A. Dick, and G. A. Wright, "Segmentation of left ventricle in cardiac cine MRI: an automatic image-driven method," in *Functional Imaging and Modeling of the Heart*. Springer, 2009, pp. 339–347.

[104] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 94–101.

[105] G. Luo, G. Sun, K. Wang, S. Dong, and H. Zhang, "A novel left ventricular volumes prediction method based on deep learning network in cardiac MRI," in *Computing in Cardiology Conference (CinC), 2016*. IEEE, 2016, pp. 89–92.

[106] Y. Luo, B. Yang, L. Xu, L. Hao, J. Liu, Y. Yao, and F. van de Vosse, "Segmentation of the Left Ventricle in Cardiac MRI Using an ELM Model," in *Proceedings of ELM-2015 Volume 1*. Springer, 2016, pp. 147–157.

[107] 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 Transactions on Medical Imaging*, vol. 27, no. 2, pp. 195–203, 2008.

[108] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 2, pp. 158–175, 1995.

[109] J. Margeta, E. Geremia, A. Criminisi, and N. Ayache, "Layered spatio-temporal forests for left ventricle segmentation from 4D cardiac MRI data," in *Statistical Atlases and Computational Models of the Heart. Imaging and Modelling Challenges*. Springer, 2012, pp. 109–119.

[110] J. S. Marques, J. C. Nascimento, and C. Santiago, "Robust Deformable Models for 2D and 3D Shape Estimation," in *Deformation Models: Tracking, Animation and Applications*, M. González Hidalgo, A. Mir Torres, and J. Varona Gómez, Eds. Dordrecht: Springer Netherlands, 2013, pp. 169–185.

[111] J. S. Marques and P. Pina, "Crater delineation by dynamic programming," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 7, pp. 1581–1585, 2015.

[112] M. McConnell, V. Khasgiwala, B. Savord, M. Chen, M. Chuang, R. Edelman, and W. Manning, "Comparison of respiratory suppression methods and navigator loca-

tions for MR coronary angiography." *AJR. American journal of roentgenology*, vol. 168, no. 5, pp. 1369–1375, 1997.

[113] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Medical image analysis*, vol. 1, no. 2, pp. 91–108, 1996.

[114] D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince, *MRI from Picture to Proton*. Cambridge university press, 2007.

[115] P. Medrano-Gracia, B. R. Cowan, D. A. Bluemke, J. P. Finn, J. A. Lima, A. Suinesiaputra, and A. A. Young, "Large Scale Left Ventricular Shape Atlas Using Automated Model Fitting to Contours," in *Functional Imaging and Modeling of the Heart*, ser. Lecture Notes in Computer Science, S. Ourselin, D. Rueckert, and N. Smith, Eds. Springer Berlin Heidelberg, 2013, vol. 7945, pp. 433–441.

[116] E. Meijering, O. Dzyubachyk, I. Smal *et al.*, "9 Methods for Cell and Particle Tracking," *Methods in enzymology*, vol. 504, no. 9, pp. 183–200, 2012.

[117] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 504–513.

[118] S. C. Mitchell, J. G. Bosch, B. P. Lelieveldt, R. J. van der Geest, J. H. Reiber, and M. Sonka, "3-D active appearance models: segmentation of cardiac MR and ultrasound images," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1167–1178, 2002.

[119] S. C. Mitchell, B. P. Lelieveldt, R. J. Van Der Geest, H. G. Bosch, J. Reiver, and M. Sonka, "Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac MR images," *IEEE Transactions on medical imaging*, vol. 20, no. 5, pp. 415–423, 2001.

[120] J. Montagnat and H. Delingette, "4D deformable models with temporal constraints: application to 4D cardiac image segmentation," *Medical Image Analysis*, vol. 9, no. 1, pp. 87–100, 2005.

[121] R. Mooiweer, A. Sbrizzi, H. El Aidi, A. L. Eikendal, A. Raaijmakers, F. Visser, C. A. van den Berg, T. Leiner, P. R. Luijten, and H. Hoogduin, "Fast 3D isotropic imaging of the aortic vessel wall by application of 2D spatially selective excitation and a new way of inversion recovery for black blood imaging," *Magnetic resonance in medicine*, vol. 75, no. 2, pp. 547–555, 2016.

[122] V. Mor-Avi, R. M. Lang, L. P. Badano, M. Belohlavek, N. M. Cardim, G. Derumeaux, M. Galderisi, T. Marwick, S. F. Nagueh, P. P. Sengupta *et al.*, "Current and evolving echocardiographic techniques for the quantitative evaluation

of cardiac mechanics: ASE/EAE consensus statement on methodology and indications: endorsed by the Japanese Society of Echocardiography," *Journal of the American Society of Echocardiography*, vol. 24, no. 3, pp. 277–313, 2011.

[123] J. C. Nascimento and J. S. Marques, "Adaptive snakes using the EM algorithm," *Image Processing, IEEE Transactions on*, vol. 14, no. 11, pp. 1678–1686, 2005.

[124] J. C. Nascimento and J. S. Marques, "Robust shape tracking in the presence of cluttered background," *IEEE transactions on Multimedia*, vol. 6, no. 6, pp. 852–861, 2004.

[125] J. C. Nascimento and J. S. Marques, "Robust shape tracking with multiple models in ultrasound images," *Image Processing, IEEE Transactions on*, vol. 17, no. 3, pp. 392–406, 2008.

[126] J. Nascimento, M. Figueiredo, and J. Marques, "Activity Recognition Using a Mixture of Vector Fields," *Image Processing, IEEE Transactions on*, vol. 22, no. 5, pp. 1712–1725, May 2013.

[127] T. A. Ngo and G. Carneiro, "Left ventricle segmentation from cardiac MRI combining level set methods with deep belief networks," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 695–699.

[128] T. A. Ngo, Z. Lu, and G. Carneiro, "Combining deep learning and level set for the automated segmentation of the left ventricle of the heart from cardiac cine magnetic resonance," *Medical image analysis*, vol. 35, pp. 159–171, 2017.

[129] H. N. Ntsinjana, M. L. Hughes, and A. M. Taylor, "The Role of Cardiovascular Magnetic Resonance in Pediatric Congenital Heart Disease," *Journal of Cardiovascular Magnetic Resonance*, vol. 13, no. 1, p. 51, 2011.

[130] S. O'Brien, O. Ghita, and P. F. Whelan, "Segmenting the left ventricle in 3D using a coupled ASM and a learned non-rigid spatial model," in *3D segmentation in the clinic: a grand challenge III [workshop], MICCAI 2009, the 12th international conference on medical image computing and computer assisted intervention*. Springer, 2009, pp. 20–24.

[131] S. P. O'Brien, O. Ghita, and P. F. Whelan, "A Novel Model-Based 3D Time Left Ventricular Segmentation Technique," *Medical Imaging, IEEE Transactions on*, vol. 30, no. 2, pp. 461–474, 2011.

[132] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[133] N. F. Osman, W. S. Kerwin, E. R. McVeigh, and J. L. Prince, "Cardiac motion tracking using CINE harmonic phase (HARP) magnetic resonance imaging," *Magnetic resonance in medicine: official journal of the Society of Magnetic Resonance in Medicine/Society of Magnetic Resonance in Medicine*, vol. 42, no. 6, p. 1048, 1999.

[134] N. F. Osman, E. R. McVeigh, and J. L. Prince, "Imaging heart motion using harmonic phase MRI," *IEEE transactions on medical imaging*, vol. 19, no. 3, pp. 186–202, 2000.

[135] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[136] X. Papademetris, A. J. Sinusas, D. P. Dione, and J. S. Duncan, "Estimation of 3D left ventricular deformation from echocardiography," *Medical image analysis*, vol. 5, no. 1, pp. 17–28, 2001.

[137] A. Pednekar, U. Kurkure, R. Muthupillai, S. Flamm, and I. A. Kakadiaris, "Automated left ventricular segmentation in cardiac MRI," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1425–1428, 2006.

[138] C. Petitjean and J. Dacher, "A review of segmentation methods in short axis cardiac MR images," *Medical image analysis*, vol. 15, no. 2, pp. 169–184, 2011.

[139] V.-T. Pham, T.-T. Tran, K.-K. Shyu, L.-Y. Lin, Y.-H. Wang, and M.-T. Lo, "Multiphase B-spline level set and incremental shape priors with applications to segmentation and tracking of left ventricle in cardiac MR images," *Machine vision and applications*, vol. 25, no. 8, pp. 1967–1987, 2014.

[140] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[141] X. Qin, Y. Tian, and P. Yan, "Feature competition and partial sparse shape modeling for cardiac image sequences segmentation," *Neurocomputing*, vol. 149, pp. 904–913, 2015.

[142] J. P. Ridgway, "Cardiovascular magnetic resonance physics for clinicians: part I," *Journal of cardiovascular magnetic resonance*, vol. 12, no. 1, p. 71, 2010.

[143] M. Rogers and J. Graham, "Robust active shape model search," in *Computer Vision–ECCV 2002*. Springer, 2006, pp. 517–530.

[144] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 8, pp. 712–721, 1999.

[145] F. B. Sachse, *Computational cardiology: modeling of anatomy, electrophysiology, and mechanics*. Springer Science & Business Media, 2004, vol. 2966.

[146] C. Santiago, J. C. Nascimento, and J. S. Marques, "Segmentation of the left ventricle in cardiac MRI using a probabilistic data association active shape model," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, Aug 2015, pp. 7304–7307.

[147] C. Santiago, J. S. Marques, and J. C. Nascimento, "A robust deformable model for 3D segmentation of the left ventricle from ultrasound data," in *Mathematical Methodologies in Pattern Recognition and Machine Learning*. Springer, 2013, pp. 163–178.

[148] C. Santiago, J. C. Nascimento, and J. S. Marques, "3D left ventricular segmentation in echocardiography using a probabilistic data association deformable model," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 606–610.

[149] C. Santiago, J. C. Nascimento, and J. S. Marques, "Performance evaluation of point matching algorithms for left ventricle motion analysis in MRI," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, 2013, pp. 4398–4401.

[150] C. Santiago, J. C. Nascimento, and J. S. Marques, "A Robust Active Shape Model Using an Expectation-Maximization Framework," in *Image Processing (ICIP), 2014 21th IEEE International Conference on*. IEEE, 2014, pp. 6076–6080.

[151] C. Santiago, J. C. Nascimento, and J. S. Marques, "Non-rigid Object Segmentation Using Robust Active Shape Models," in *Articulated Motion and Deformable Objects*. Springer, 2014, pp. 160–169.

[152] C. Santiago, J. C. Nascimento, and J. S. Marques, "2D Segmentation Using a Robust Active Shape Model With the EM Algorithm," *Image Processing, IEEE Transactions on*, vol. 24, no. 8, pp. 2592–2601, Aug 2015.

[153] C. Santiago, J. C. Nascimento, and J. S. Marques, "Automatic 3-D Segmentation of Endocardial Border of the Left Ventricle From Ultrasound Images," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 1, pp. 339–348, Jan 2015.

[154] C. Santiago, J. C. Nascimento, and J. S. Marques, "Robust 3D Active Shape Model for the Segmentation of the Left Ventricle in MRI," in *Pattern Recognition and Image Analysis - IbPRIA'15,* ***(Best Paper Award)***, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015, vol. 9117, pp. 283–290.

[155] C. Santiago, J. C. Nascimento, and J. S. Marques, "A 3D Active Shape Model for Left Ventricle Segmentation in MRI," in *Biomedical Image Segmentation: Advances and Trends*. CRC Press, 2016, pp. 99–116.

[156] C. Santiago, J. C. Nascimento, and J. S. Marques, "A new ASM framework for left ventricle segmentation exploring slice variability in cardiac MRI volumes," *Neural Computing and Applications*, pp. 1–12, 2016.

[157] C. Santiago, J. C. Nascimento, and J. S. Marques, "A New Robust Active Shape Model Formulation For Cardiac MRI Segmentation," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4112–4115.

[158] C. Santiago, J. C. Nascimento, and J. S. Marques, "Fast and accurate segmentation of the LV in MR volumes using a deformable model with dynamic programming," in *Image Processing (ICIP), 2017 IEEE International Conference on - submitted*. IEEE, 2017.

[159] C. Santiago, J. C. Nascimento, and J. S. Marques, "Fast Segmentation of the Left Ventricle in Cardiac MRI Using Dynamic Programming," *Computer Methods and Programs in Biomedicine*, 2017.

[160] J. Schaerer, C. Casta, J. Pousin, and P. Clarysse, "A dynamic elastic model for segmentation and tracking of the heart in MR image sequences," *Medical Image Analysis*, vol. 14, no. 6, pp. 738–749, 2010.

[161] L. Scharf, *Statistical Signal Processing*. Addison Wesley Publishing Company, Reading, Massachusetts, 1991.

[162] K. Scheffler and S. Lehnhardt, "Principles and applications of balanced SSFP techniques," *European radiology*, vol. 13, no. 11, pp. 2409–2418, 2003.

[163] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," *ACM SIGGRAPH computer graphics*, vol. 20, no. 4, pp. 151–160, 1986.

[164] J. Sénégas, T. Netsch, C. A. Cocosco, G. Lund, and A. Stork, "Segmentation of medical images with a shape and motion model: A Bayesian perspective," in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*. Springer, 2004, pp. 157–168.

[165] M. Sermesant, H. Delingette, and N. Ayache, "An electromechanical model of the heart for image analysis and simulation," *IEEE transactions on medical imaging*, vol. 25, no. 5, pp. 612–625, 2006.

[166] H. Sliman, F. Khalifa, A. Elnakib, A. Soliman, G. Beache, A. Elmaghraby, and A. El-Baz, "A new segmentation-based tracking framework for extracting the left

ventricle cavity from cine cardiac MRI," in *Proceedings of the IEEE international conference on image processing (ICIP'13), Melbourne, Australia*, 2013, pp. 685–689.

[167] A. Suinesiaputra, B. R. Cowan, A. O. Al-Agamy, M. A. Elattar, N. Ayache, A. S. Fahmy, A. M. Khalifa, P. Medrano-Gracia, M.-P. Jolly, A. H. Kadish *et al.*, "A collaborative resource to build consensus for automated left ventricular segmentation of cardiac MR images," *Medical image analysis*, vol. 18, no. 1, pp. 50–62, 2014.

[168] W. Sun, M. Cetin, R. Chan, V. Reddy, G. Holmvang, V. Chandar, and A. Willsky, "Segmenting and tracking the left ventricle by learning the dynamics in cardiac images," in *Biennial International Conference on Information Processing in Medical Imaging*. Springer, 2005, pp. 553–565.

[169] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[170] V. Tavakoli and A. A. Amini, "A survey of shaped-based registration and segmentation techniques for cardiac images," *Computer Vision and Image Understanding*, vol. 117, no. 9, pp. 966–989, 2013.

[171] H. Thiele, E. Nagel, I. Paetsch, B. Schnackenburg, A. Bornstedt, M. Kouwenhoven, A. Wahl, G. Schuler, and E. Fleck, "Functional cardiac MR imaging with steady-state free precession (SSFP) significantly improves endocardial border delineation without contrast agents," *Journal of Magnetic Resonance Imaging*, vol. 14, no. 4, pp. 362–367, 2001.

[172] N. A. Trayanova, "Whole-heart modeling applications to cardiac electrophysiology and electromechanics," *Circulation research*, vol. 108, no. 1, pp. 113–128, 2011.

[173] A. Tsai, W. Wells, C. Tempany, E. Grimson, and A. Willsky, "Mutual information in coupled multi-shape model for medical image segmentation," *Medical Image Analysis*, vol. 8, no. 4, pp. 429–445, 2004.

[174] A. Tsai, A. Yezzi Jr, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 2, pp. 137–154, 2003.

[175] M. Üzümcü, R. J. van der Geest, C. Swingen, J. H. Reiber, and B. P. Lelieveldt, "Time continuous tracking and segmentation of cardiovascular magnetic resonance images using multidimensional dynamic programming," *Investigative radiology*, vol. 41, no. 1, pp. 52–62, 2006.

[176] H. C. Van Assen, M. G. Danilouchkine, M. S. Dirksen, J. Reiber, and B. P. Lelieveldt, "A 3-D active shape model driven by fuzzy inference: application to cardiac CT and

MR," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 5, pp. 595–605, 2008.

[177] H. C. Van Assen, M. G. Danilouchkine, A. F. Frangi, S. Ordás, J. J. Westenberg, J. H. Reiber, and B. P. Lelieveldt, "SPASM: a 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data," *Medical Image Analysis*, vol. 10, no. 2, pp. 286–303, 2006.

[178] R. Van der Geest, E. Jansen, V. Buller, and J. Reiber, "Automated detection of left ventricular epi-and endocardial contours in short-axis MR images," in *Computers in Cardiology 1994*. IEEE, 1994, pp. 33–36.

[179] B. Van Ginneken, A. F. Frangi, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 8, pp. 924–933, 2002.

[180] H. L. Van Trees, *Detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[181] E. Waks, J. L. Prince, and A. S. Douglas, "Cardiac motion simulator for tagged MRI," in *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, Jun 1996, pp. 182–191.

[182] L. Wang, A. Basarab, P. R. Girard, P. Croisille, P. Clarysse, and P. Delachartre, "Analytic signal phase-based myocardial motion estimation in tagged MRI sequences by a bilinear model and motion compensation," *Medical image analysis*, vol. 24, no. 1, pp. 149–162, 2015.

[183] Y. Wang and L. H. Staib, "Boundary finding with correspondence using statistical shape models," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 338–345.

[184] M. Wimmer, K. Stulp, S. Pietzsch, and B. Radig, "Learning local objective functions for robust face model fitting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 8, pp. 1357–1370, 2008.

[185] Y. Wu, Y. Wang, and Y. Jia, "Segmentation of the left ventricle in cardiac cine MRI using a shape-constrained snake model," *Computer Vision and Image Understanding*, vol. 117, no. 9, pp. 990–1003, 2013.

[186] P. Yan, N. Lin, A. J. Sinusas, and J. S. Duncan, "A boundary element-based approach to analysis of LV deformation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2005, pp. 778–785.

[187] J.-Y. Yeh, J. Fu, C. Wu, H. Lin, and J. Chai, "Myocardial border detection by branch-and-bound dynamic programming in magnetic resonance images," *Computer methods and programs in biomedicine*, vol. 79, no. 1, pp. 19–29, 2005.

[188] H. Zhang, A. Wahle, R. K. Johnson, T. D. Scholz, and M. Sonka, "4-D cardiac MR image analysis: left and right ventricular morphology and function," *IEEE transactions on medical imaging*, vol. 29, no. 2, pp. 350–364, 2010.

[189] T. Zhou, I. Icke, B. Dogdas, S. Parimal, S. Sampath, J. Forbes, A. Bagchi, C.-L. Chin, and A. Chen, "Automatic segmentation of left ventricle in cardiac cine MRI images based on deep learning," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2017, pp. 101 331W–101 331W.

[190] Y. Zhu, X. Papademetris, A. J. Sinusas, and J. S. Duncan, "Segmentation of the left ventricle from cardiac MR images using a subject-specific dynamical model," *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 669–687, 2010.

[191] X. Zhuang, D. Hawkes, W. Crum, R. Boubertakh, S. Uribe, D. Atkinson, P. Batchelor, T. Schaeffter, R. Razavi, and D. Hill, "Robust registration between cardiac MRI images and atlas for segmentation propagation," in *Medical Imaging*. International Society for Optics and Photonics, 2008, pp. 691 408–691 408.

[192] X. Zhuang, K. S. Rhode, R. S. Razavi, D. J. Hawkes, and S. Ourselin, "A registration-based propagation framework for automatic whole heart segmentation of cardiac MRI," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 9, pp. 1612–1625, 2010.

A

EM-RASM with Edge Segments

Contents

Alternatively to relying on edge points to estimate the model parameters, a different approach was also explored, which relied on edge segments. This new observation model is described in this appendix, along with the corresponding derivations of the EM-RASM formulation for edge segments.

A.1 Problem Formulation

We wish the find the shape model parameters that best describe the LV border of a particular CMR image. Previously, the estimation of the model parameters was based on a set of edge points extracted from the image (see Fig. A.1 left). Now, instead of look at individual edge points, the algorithm will rely on edge segments to determine the location of the LV border (see Fig. A.1 middle).

Given an MR image, edge segments may be extracted using different approaches [12, 124]. Here, they are obtained by linking the detected edge points using the mutual favorite pairing method [73], which is the approach used in [124]. This method consists in successively pairing edge points from neighboring model points if they are mutually the closest pairs available. To avoid unrelated edge points from being paired, a threshold is set on the maximum distance allowed between two neighboring edge points in the same segment. An example of this pairing algorithm is shown in Fig. A.1 (from the left image to the one in the middle).

Let $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^P\}$ denote the set of the P segments detected on the image, such that the i-th segment, $\boldsymbol{Y}^i \in \mathbb{R}^{2M^i \times 1}$, is given by M^i edge points

$$\boldsymbol{Y}^i = \left[\boldsymbol{y}^{i1\top}, \ldots, \boldsymbol{y}^{iM^i\top} \right]^\top, \tag{A.1}$$

where $\boldsymbol{y}^{ij} \in \mathbb{R}^{2\times 1}$ is the position of the j-th edge point in the i-th segment, $\boldsymbol{Y}^i$ (see Fig. A.1 (middle and right) for an illustration). The idea is to estimate the parameters of a shape model,

$$\widetilde{\boldsymbol{x}} = \boldsymbol{T}_{\boldsymbol{\theta}} \left(\overline{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{b} \right), \tag{A.2}$$

which include the global transformation parameters, $\boldsymbol{\theta} = \{\boldsymbol{a}, \boldsymbol{t}\}$, and deformation coefficients, $\boldsymbol{b}$, such that the model is as close as possible to the edge segments that belong to the LV border.

Unfortunately, we do not know which segments are valid (the ones that belong to the LV border) or invalid (outliers). Therefore, two observation models are considered. These models are associated to a binary label, k^i, that has the value $k^i = 1$ if $\boldsymbol{Y}^i$ is valid, and the value $k^i = 0$ if $\boldsymbol{Y}^i$ is an outlier. These two labels occur with probabilities $P(k^i = 1) = p_1$ and $P(k^i = 0) = p_0$, respectively, such that $p_0 + p_1 = 1$.

We assume that edge points in valid segments are generated by the following observation model

$$\boldsymbol{y}^{ij} = \widetilde{\boldsymbol{x}}^{ij} + \boldsymbol{v}^{ij}, \tag{A.3}$$

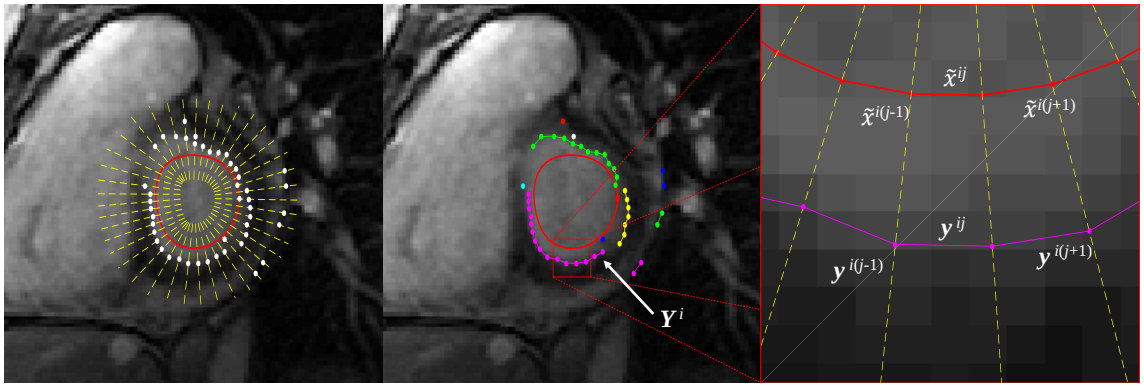


Figure A.1: Detection of edge segments in a CMR image: (left) detection of edge points along lines orthogonal to the contour model; (middle) edge points linked to form edge strokes; (right) zoom on a particular stroke.

where $\widetilde{\boldsymbol{x}}^{ij}$ is the position of the corresponding shape model point (*i.e.*, the contour point to which $\boldsymbol{y}^{ij}$ is associated), and $\boldsymbol{v}^{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{ij})$ is a zero mean white Gaussian noise with diagonal covariance matrix $\boldsymbol{\Sigma}^{ij}$. This means that the probability of a specific segment, $\boldsymbol{Y}^i$, is given by

$$\begin{aligned} p\left(\boldsymbol{Y}^i \middle| k^i{=}1, \boldsymbol{\Theta}\right) &= \prod_{j=1}^{M^i} p\left(\boldsymbol{y}^{ij} \middle| k^i{=}1, \boldsymbol{\Theta}\right) \\ &= \prod_{j=1}^{M^i} \mathcal{N}\left(\boldsymbol{y}^{ij}; \widetilde{\boldsymbol{x}}^{ij}, \boldsymbol{\Sigma}^{ij}\right), \end{aligned} \tag{A.4}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}, p_0, p_1\}$ is the set of all the model parameters. On the other hand, if $\boldsymbol{Y}^i$ is an outlier (*i.e.*, $k^i = 0$), then we assume that it follows a uniform distribution

$$\begin{aligned} p\left(\boldsymbol{Y}^i \middle| k^i{=}0, \boldsymbol{\Theta}\right) &= \prod_{j=1}^{M^i} p\left(\boldsymbol{y}^{ij} \middle| k^i{=}0, \boldsymbol{\Theta}\right), \\ &= \prod_{j=1}^{M^i} \mathcal{U}(V_{\widetilde{\boldsymbol{x}}^{ij}}), \end{aligned} \tag{A.5}$$

within a region $V_{\widetilde{\boldsymbol{x}}^{ij}}$ in the vicinity of $\widetilde{\boldsymbol{x}}^{ij}$.

Denoting the set of labels associated to $\boldsymbol{\mathcal{Y}}$ as $\boldsymbol{\mathcal{K}} = \{k^1, \ldots, k^P\}$ be the set of the corresponding labels, then the goal of the proposed algorithm is to find the model parameters, $\boldsymbol{\Theta}$, that maximize the following log-joint probability distribution

$$\begin{aligned} \mathcal{P}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\Theta}) &= \log p(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\Theta}) \\ &= \log p\left(\boldsymbol{\mathcal{Y}} | \boldsymbol{\mathcal{K}}, \boldsymbol{\Theta}\right) + \log p(\boldsymbol{\mathcal{K}}) + \log p(\boldsymbol{\Theta}). \end{aligned} \tag{A.6}$$

Assuming conditional independence between segments, the above equation can be factorized as follows

$$\mathcal{P}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\Theta}) = \sum_{i=1}^{P} \log p\left(\boldsymbol{Y}^i \middle| k^i, \boldsymbol{\Theta}\right) + \log p(k^i) + \log p(\boldsymbol{\Theta}). \tag{A.7}$$

The first term corresponds to the likelihood of the edge segments, the second term is the probability of the observation model, k^i, and the last term corresponds to the priors of the parameters $\boldsymbol{\Theta}$.

Regarding the prior probability of the model parameters, we make the same assumptions used in Chapters 3 and 4: the prior probability distribution for the deformation coefficients is assumed to be given by $p(\boldsymbol{b}) = \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}^{\boldsymbol{b}}\right)$, and the prior probabilities for the other parameters are given by the Jeffreys' prior, *i.e.*, they do not depend any of the model parameters, $\boldsymbol{\Theta}$.

A.2 Expectation-Maximization

As previously, the EM algorithm is used to solve this optimization problem, where in each iteration, t, the parameter estimates, $\widehat{\boldsymbol{\Theta}}_{(t)}$, are updated by maximizing an auxiliary function, $Q\left(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}_{(t)}\right)$, in a two step procedure described next.

A.2.1 E-step

Let $\widehat{\boldsymbol{\Theta}}_{(t)}$ denote the current estimate of the model parameters (where $\widehat{\boldsymbol{\Theta}}_{(0)}$ is the initial guess). The auxiliary function of the EM algorithm is given by

$$Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) = \mathbb{E}_{\boldsymbol{\mathcal{K}}}\left[\mathcal{P}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\Theta}) \middle| \boldsymbol{\mathcal{Y}}, \widehat{\boldsymbol{\Theta}}_{(t)}\right], \tag{A.8}$$

where $\mathbb{E}_{\boldsymbol{\mathcal{K}}}[\cdot]$ denotes the expectation over the two observation models. This leads to

$$\begin{aligned} Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) = & \sum_{i=1}^{P} w_0^i \left[\log p\left(\boldsymbol{Y}^i \middle| k^i{=}0, \boldsymbol{\Theta}\right) + \log p_0\right] \\ & + w_1^i \left[\log p\left(\boldsymbol{Y}^i \middle| k^i{=}1, \boldsymbol{\Theta}\right) + \log p_1\right] \\ & + \log p(\boldsymbol{\Theta}), \end{aligned} \tag{A.9}$$

where w_0^i and w_1^i denote the confidence degree of each segment, such that $w_0^i + w_1^i = 1$ and

$$\begin{aligned} w_1^i &= p\left(k^i{=}1 \middle| \boldsymbol{Y}^i, \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\ &\propto p\left(\boldsymbol{Y}^i \middle| k^i{=}1, \widehat{\boldsymbol{\Theta}}_{(t)}\right) p\left(k^i{=}1 \middle| \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\ &\propto \widehat{p}_{1(t)} \prod_{j=1}^{M^i} \mathcal{N}\left(\boldsymbol{y}^{ij}; \widetilde{\boldsymbol{x}}^{ij}, \boldsymbol{\Sigma}^{ij}\right) \end{aligned} \tag{A.10}$$

$$\begin{aligned} w_0^i &= p\left(k^i{=}0 \middle| \boldsymbol{Y}^i, \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\ &\propto \widehat{p}_{0(t)} \prod_{j=1}^{M^i} \mathcal{U}\left(V_{\widetilde{\boldsymbol{x}}^{ij}}\right). \end{aligned} \tag{A.11}$$

These weights correspond to the probability of k_1^i and k_0^i being the correct label for segment $\boldsymbol{Y}^i$, given the current model parameters.

Combining (A.9) with (A.4)-(A.5) and the prior terms yields

$$Q\left(\boldsymbol{\Theta};\widehat{\boldsymbol{\Theta}}_{(t)}\right) = c - \frac{1}{2}\boldsymbol{b}^\top\boldsymbol{\Sigma}^{\boldsymbol{b}}\boldsymbol{b} - \frac{1}{2}\sum_{i=1}^{N} w_0^i \log p_0 + w_1^i\Big(\log p_1 + \sum_{j=1}^{M^i} \boldsymbol{e}^{ij\top}\boldsymbol{\Sigma}^{ij-1}\boldsymbol{e}^{ij}\Big), \quad \text{(A.12)}$$

where

$$\boldsymbol{e}^{ij} = \boldsymbol{y}^{ij} - \boldsymbol{A}\Big(\overline{\boldsymbol{x}}^{ij} + \boldsymbol{D}^{ij}\boldsymbol{b}\Big) - \boldsymbol{t}, \quad \text{(A.13)}$$

and c is a constant that does not depend on the model parameters, $\boldsymbol{\Theta}$.

A.2.2 M-step

The model parameters are updated by solving the following optimization problem

$$\widehat{\boldsymbol{\Theta}}_{(t+1)} = \arg\max_{\boldsymbol{\Theta}} Q\left(\boldsymbol{\Theta};\widehat{\boldsymbol{\Theta}}_{(t)}\right), \quad \text{(A.14)}$$

first with respect to the transformation parameters, $\boldsymbol{a}, \boldsymbol{t}$, then for $\boldsymbol{b}$, and finally for p_0 and p_1.

Update of the Transformation Parameters Taking the derivative of (A.12) with respect to $\boldsymbol{a}$ and $\boldsymbol{t}$ and equating to zero leads to the following linear system

$$\sum_{i=1}^{P} w_1^i \sum_{j=1}^{M^i} \begin{bmatrix} \boldsymbol{X}^{ij\top}\boldsymbol{\Sigma}^{ij-1}\boldsymbol{X}^{ij} & \boldsymbol{X}^{ij\top}\boldsymbol{\Sigma}^{ij-1} \\ \boldsymbol{\Sigma}^{ij-1}\boldsymbol{X}^{ij} & \boldsymbol{\Sigma}^{ij-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{a}}_{(t+1)} \\ \widehat{\boldsymbol{t}}_{(t+1)} \end{bmatrix} = \sum_{i=1}^{P} w_1^i \sum_{j=1}^{M^i} \begin{bmatrix} \boldsymbol{X}^{ij\top}\boldsymbol{\Sigma}^{ij-1}\boldsymbol{y}^{ij} \\ \boldsymbol{\Sigma}^{ij-1}\boldsymbol{y}^{ij} \end{bmatrix} \quad \text{(A.15)}$$

where

$$\boldsymbol{X}^{ij} = \begin{bmatrix} x_1^{ij} & -x_2^{ij} \\ x_2^{ij} & x_1^{ij} \end{bmatrix}, \qquad \begin{bmatrix} x_1^{ij} \\ x_2^{ij} \end{bmatrix} = \boldsymbol{x}^{ij} = \overline{\boldsymbol{x}}^{ij} + \boldsymbol{D}^{ij}\widehat{\boldsymbol{b}}_{(t)}$$

Update of the Deformation Coefficients The update of the deformation coefficients, $\widehat{\boldsymbol{b}}_{(t+1)}$, is obtained by taking the derivative of (A.12) with respect to $\boldsymbol{b}$,

$$\left[\boldsymbol{\Sigma}^{\boldsymbol{b}-1} + \sum_{i=1}^{P} w_1^i \sum_{j=1}^{M^i} \boldsymbol{D}^{ij\top}\widehat{\boldsymbol{A}}_{(t+1)}^\top\boldsymbol{\Sigma}^{ij-1}\widehat{\boldsymbol{A}}_{(t+1)}\boldsymbol{D}^{ij}\right]\widehat{\boldsymbol{b}}_{(t+1)} = \\ = \sum_{i=1}^{P} w_1^i \sum_{j=1}^{M^i} \left[\boldsymbol{D}^{ij\top}\widehat{\boldsymbol{A}}_{(t+1)}^\top\boldsymbol{\Sigma}^{ij-1}\left(\boldsymbol{y}^{ij} - \widehat{\boldsymbol{A}}_{(t+1)}\overline{\boldsymbol{x}}^{ij} - \widehat{\boldsymbol{t}}_{(t+1)}\right)\right], \quad \text{(A.16)}$$

where

$$\widehat{\boldsymbol{A}}_{(t+1)} = \begin{bmatrix} \widehat{a}_{1(t+1)} & -\widehat{a}_{2(t+1)} \\ \widehat{a}_{2(t+1)} & \widehat{a}_{1(t+1)} \end{bmatrix}$$

Table A.1: Statistical performance of 2D EM-RASM using edge points or edge segments as observations

EM-RASM with	Dice (%)	AV (mm)	VD (mL)	% Good
Edge Points	84.2 (9.2)	2.6 (1.7)	11.8 (11.4)	89.2
Edge Segments	83.1 (8.0)	2.8 (2.2)	16.7 (23.2)	88.9

Update of the Model Probabilities The observation model probabilities, $\widehat{p}_{0(t)}$, $\widehat{p}_{1(t)}$, are updated by taking the derivative of (A.12) with respect to $\widehat{p}_{1(t)}$ (recall that $\widehat{p}_{0(t)} = 1 - \widehat{p}_{1(t)}$), which leads to

$$\widehat{p}_{1(t+1)} = \frac{1}{P}\sum_{i=1}^{P} w_1^i \tag{A.17}$$

$$\widehat{p}_{0(t+1)} = 1 - \widehat{p}_{0(t+1)}. \tag{A.18}$$

The update equations in this formulation are similar to the ones obtained in the approach described in Chapter 3. However, in this case, the EM weights, w_1^i, are now associated to edge segments, instead of individual edge points. Consequently, all the points in a specific edge will equally contribute to the estimation of the parameters.

A.3 Results

The experimental setup used to evaluate the approach using edge segments is the same as the one described in Chapter 3 for edge points.

Fig. A.2 shows an example depicting the detected edge segments (colored dotted lines), the current model (red contour) and the corresponding EM weights. It is possible to see that the valid segments have a much higher probability than the outliers.

Table A.1 shows the performance of the proposed method in the database used in this thesis. It also shows the performance achieved by the 2D EM-RASM method based on points. The performance of the algorithm using edge segments is slightly worse than using edge points.

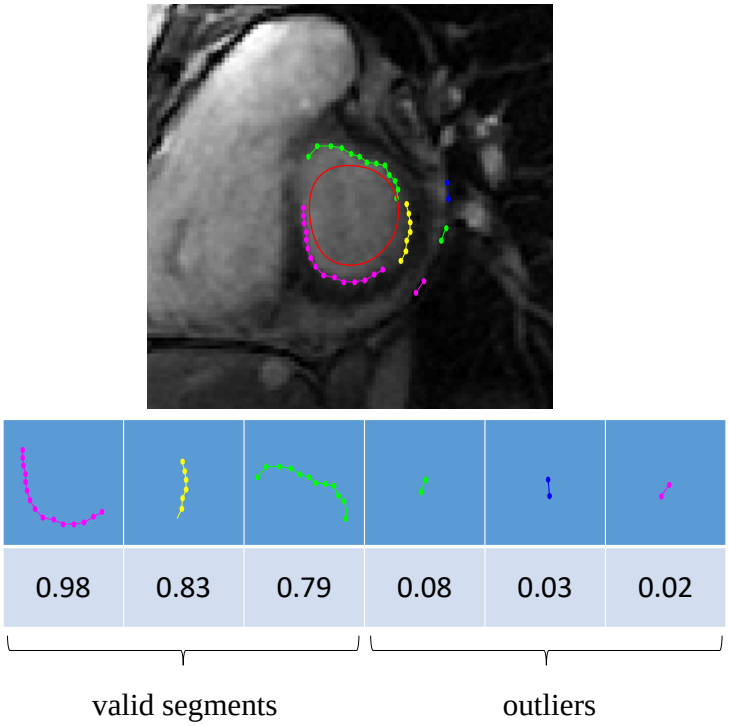


Figure A.2: Example of the detected edge segments and their probability. Each colored dotted line corresponds to an edge segment and the red contour is the current model.

B

EM-RASM With Data Association

Contents

This appendix explores using data association as an alternative approach to modeling the presence of outliers for the estimation of a shape model parameters. This concept was proposed in the context of radar points tracking [18], but has since been used in many other applications [8, 124, 125, 153]. The methodology proposed here is similar to the one proposed in [124], which uses data association to improve the accuracy of a Snake model. In this appendix, instead of a using a Snake model, data association is combined with the ASM formulation. Details of this approach are described next.

B.1 Problem Formulation

The idea of the proposed approach is to provide robust estimates of the shape model parameters, for the segmentation of 2D images (although the extension to 3D is straightforward). Assuming there is a shape model, previously learned from shape examples of the object of interest, defined by

$$\widetilde{\boldsymbol{x}} = \boldsymbol{T}_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{b}\right), \tag{B.1}$$

the goal is to estimate the parameters $\boldsymbol{\Theta} = \{\boldsymbol{a}, \boldsymbol{t}, \boldsymbol{b}\}$ such that $\widetilde{\boldsymbol{x}}$ fits the border of the object. Let $\boldsymbol{Y} = \{\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^P\}$ denote a set of edge segments extracted from the image. The i-th observation, $\boldsymbol{Y}^i \in \mathbb{R}^{2M^i \times 1}$, is given by M^i edge points

$$\boldsymbol{Y}^i = \left[\boldsymbol{y}^{i1^\top}, \ldots, \boldsymbol{y}^{iM^{i\top}}\right]^\top, \tag{B.2}$$

where $\boldsymbol{y}^{ij}(s) \in \mathbb{R}^2$ is the j-th edge point in that segment, associated to the model point $\widetilde{\boldsymbol{x}}^{ij}$ (recall Fig. A.1 (right) for a illustration of these observations).

Of all the P detected segments, only a subset actually matches the border of the LV. This is represented by assigning a binary label to each segment, in which the label one is assigned to valid segments (*i.e.*, if they match the LV border) and zero to outliers. A data interpretation is defined as a specific combination of these P labels, which means that there is a total of 2^P possible data interpretations. However, not all the data interpretations are feasible. For instance, if two segments overlap with respect to the shape model (*e.g.*, one segment along the endocardium and one along the epicardium), then all the data interpretations that consider them valid segments have to be disregarded, because they cannot be simultaneously located at the LV border. This reduces the number of possible data interpretations.

Let $I^i = \{0, 1\}$ be the label associated to the i-th segment, such that $I^i(s) = 1$ means it is valid, and $I^i(s) = 0$ means it is an outlier. Similarly to Appendix A, edge segments are modeled using two different observations models: 1) a valid segment is assumed to

follow

$$p\left(\boldsymbol{Y}^i\middle|I^i{=}1,\boldsymbol{\Theta}\right)=\prod_{j=1}^{M^i}p\left(\boldsymbol{y}^{ij}\middle|I^i{=}1,\boldsymbol{\Theta}\right)$$
$$=\prod_{j=1}^{M^i}\mathcal{N}\left(\boldsymbol{y}^{ij};\widetilde{\boldsymbol{x}}^{ij},\boldsymbol{\Sigma}^{ij}\right), \tag{B.3}$$

where $\boldsymbol{\Theta}=\{\boldsymbol{a},\boldsymbol{t},\boldsymbol{b},p_0,p_1\}$ is the set of all the model parameters, and $\boldsymbol{\Sigma}^{ij}$ is the diagonal covariance matrix associated to $\widetilde{\boldsymbol{x}}^{ij}$; and 2) an outlier is assumed to follow

$$p\left(\boldsymbol{Y}^i\middle|I^i{=}0,\boldsymbol{\Theta}\right)=\prod_{j=1}^{M^i}p\left(\boldsymbol{y}^{ij}\middle|I^i{=}0,\boldsymbol{\Theta}\right),$$
$$=\prod_{j=1}^{M^i}\mathcal{U}(V_{\widetilde{\boldsymbol{x}}^{ij}}), \tag{B.4}$$

within a region $V_{\widetilde{\boldsymbol{x}}^{ij}}$ in the vicinity of $\widetilde{\boldsymbol{x}}^{ij}$.

The probability of a particular data interpretation, $I=\{I^1,\ldots,I^P\}$, is defined as the product of the probability of each segment given the corresponding labels, which can be expressed as

$$p\left(\boldsymbol{Y}\mid\boldsymbol{\Theta},I\right)=\prod_{i=1}^{P}\begin{bmatrix}I^i & 1-I^i\end{bmatrix}\begin{bmatrix}p\left(\boldsymbol{Y}^i\middle|I^i{=}1,\boldsymbol{\Theta}\right)\\ p\left(\boldsymbol{Y}^i\middle|I^i{=}0,\boldsymbol{\Theta}\right)\end{bmatrix}, \tag{B.5}$$

As previously, since the correct label of each segment, $I^1,\ldots,I^P$, is unknown, we use the EM algorithm to estimate the model parameters.

B.2 Expectation-Maximization

The EM algorithm allows an estimate of the model parameters, $\boldsymbol{\Theta}$, and the data interpretation, $I=\{I^1,\ldots,I^P\}$, to be determined simultaneously. This is achieved through the maximization of an auxiliary function

$$Q\left(\boldsymbol{\Theta};\widehat{\boldsymbol{\Theta}}_{(t)}\right)=\mathbb{E}_I\left[\mathcal{P}(\boldsymbol{Y},I,\boldsymbol{\Theta})\middle|\boldsymbol{Y},\widehat{\boldsymbol{\Theta}}_{(t)}\right], \tag{B.6}$$

where $\mathbb{E}_I[\cdot]$ denotes the expectation over all possible data interpretations, $\widehat{\boldsymbol{\Theta}}_{(t)}$ is the parameters estimate at iteration t, and $\mathcal{P}\left(\boldsymbol{Y},I,\boldsymbol{\Theta}\right)$ is the log-posterior probability, defined by

$$\mathcal{P}\left(\boldsymbol{Y},I,\boldsymbol{\Theta}\right)=\log p\left(\boldsymbol{Y},\boldsymbol{\Theta},I\right) \tag{B.7}$$
$$=\log p\left(\boldsymbol{Y}|\boldsymbol{\Theta},I\right)+\log p\left(I|\boldsymbol{\Theta}\right)+\log p\left(\boldsymbol{\Theta}\right). \tag{B.8}$$

The EM algorithm finds the model parameters by iterating between two steps [45]: 1) the *E-step*, in which the expectation in (B.6) is computed; and 2) the *M-step*, in which the model parameters are updated by maximizing the expectation obtained in the E-step. These steps are explained below.

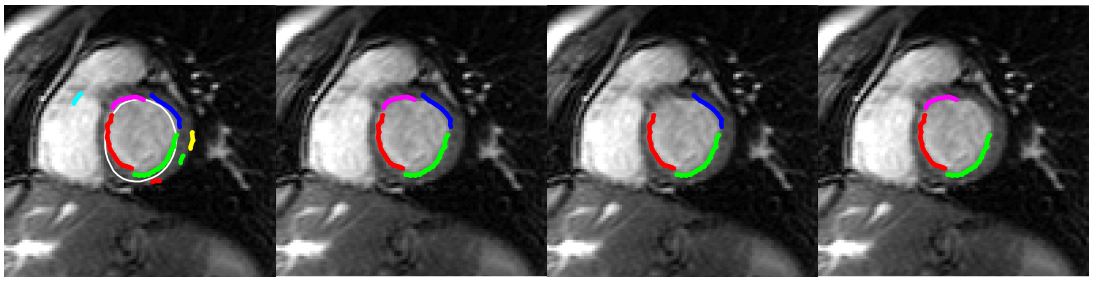


Figure B.1: Example of data interpretations on an MR image: (left) the detected segments, each with a different color (the white line corresponds to the current model estimate), and each of the following images show the most likely interpretations (*i.e.*, with higher probability, w_k).

B.2.1 E-step

Given the current estimate of the model parameters, $\widehat{\boldsymbol{\Theta}}_{(t)}$, at iteration t, the auxiliary function in (B.6) involves computing the expectation over all the possible data interpretations. Let $I_k = \left[I_k^1, \ldots, I_k^P\right]$ denote the k-th interpretation; then, equation (B.6) can be expanded as follows

$$Q\left(\boldsymbol{\Theta}; \widehat{\boldsymbol{\Theta}}_{(t)}\right) = \sum_k w_k \Big(\log p\left(\boldsymbol{Y}|\boldsymbol{\Theta}, I\right) + \log p\left(I|\boldsymbol{\Theta}\right) + \log p\left(\boldsymbol{\Theta}\right) \Big), \tag{B.9}$$

where w_k is the conditional probability of I_k, given by

$$\begin{aligned} w_k &= p\left(I_k \middle| \boldsymbol{Y}, \widehat{\boldsymbol{\Theta}}_{(t)}\right) \\ &\propto \prod_{i=1}^{P} p\left(\boldsymbol{Y}^i \middle| \widehat{\boldsymbol{\Theta}}_{(t)}, I_k^i\right) p\left(I_k \middle| \widehat{\boldsymbol{\Theta}}_{(t)}\right), \end{aligned} \tag{B.10}$$

such that $\sum_k w_k = 1$.

In this work, we assume that all the interpretations are equally probable, *i.e.*, $p\left(I_k \middle| \widehat{\boldsymbol{\Theta}}_{(t)}\right)$ is a constant, except for interpretations with overlapping segments, which are assigned a probability of zero a priori. Figure B.1 shows an example image, depicting the detected segments (left) and the three interpretations with higher probability.

B.2.2 M-step

In this step, the model parameters are updated by maximizing (B.9) with respect to the transformation parameters, $\boldsymbol{a}$ and $\boldsymbol{t}$, and then for $\boldsymbol{b}$.

Update of the Transformation Parameters Taking the derivative of (B.9) with respect to $\boldsymbol{a}$ and $\boldsymbol{t}$ and equating to zero leads to the following linear system

$$\sum_k w_k \sum_{i=1}^{P} \sum_{j=1}^{M^i} \begin{bmatrix} \boldsymbol{X}^{ij\top} \boldsymbol{\Sigma}^{ij\,-1} \boldsymbol{X}^{ij} & \boldsymbol{X}^{ij\top} \boldsymbol{\Sigma}^{ij\,-1} \\ \boldsymbol{\Sigma}^{ij\,-1} \boldsymbol{X}^{ij} & \boldsymbol{\Sigma}^{ij\,-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{a}}_{(t+1)} \\ \widehat{\boldsymbol{t}}_{(t+1)} \end{bmatrix} = \\ = \sum_k w_k \sum_{i=1}^{P} \sum_{j=1}^{M^i} \begin{bmatrix} \boldsymbol{X}^{ij\top} \boldsymbol{\Sigma}^{ij\,-1} \boldsymbol{y}^{ij} \\ \boldsymbol{\Sigma}^{ij\,-1} \boldsymbol{y}^{ij} \end{bmatrix} \quad \text{(B.11)}$$

where

$$\boldsymbol{X}^{ij} = \begin{bmatrix} x_1^{ij} & -x_2^{ij} \\ x_2^{ij} & x_1^{ij} \end{bmatrix}, \qquad \begin{bmatrix} x_1^{ij} \\ x_2^{ij} \end{bmatrix} = \boldsymbol{x}^{ij} = \overline{\boldsymbol{x}}^{ij} + \boldsymbol{D}^{ij} \widehat{\boldsymbol{b}}_{(t)}$$

Update of the Deformation Coefficients The update of the deformation coefficients, $\widehat{\boldsymbol{b}}_{(t+1)}$, is obtained by taking the derivative of (B.9) with respect to $\boldsymbol{b}$,

$$\left[\boldsymbol{\Sigma}^{\boldsymbol{b}\,-1} + \sum_k w_k \sum_{i=1}^{P} \sum_{j=1}^{M^i} \boldsymbol{D}^{ij\top} \widehat{\boldsymbol{A}}_{(t+1)}^{\top} \boldsymbol{\Sigma}^{ij\,-1} \widehat{\boldsymbol{A}}_{(t+1)} \boldsymbol{D}^{ij} \right] \widehat{\boldsymbol{b}}_{(t+1)} = \\ = \sum_k w_k \sum_{i=1}^{P} \sum_{j=1}^{M^i} \left[\boldsymbol{D}^{ij\top} \widehat{\boldsymbol{A}}_{(t+1)}^{\top} \boldsymbol{\Sigma}^{ij\,-1} \left(\boldsymbol{y}^{ij} - \widehat{\boldsymbol{A}}_{(t+1)} \overline{\boldsymbol{x}}^{ij} - \widehat{\boldsymbol{t}}_{(t+1)} \right) \right], \quad \text{(B.12)}$$

where

$$\widehat{\boldsymbol{A}}_{(t+1)} = \begin{bmatrix} \widehat{a}_{1(t+1)} & -\widehat{a}_{2(t+1)} \\ \widehat{a}_{2(t+1)} & \widehat{a}_{1(t+1)} \end{bmatrix}.$$

The method described above is denoted by Probabilistic Data Association Active Shape Model (PDAASM). It differs from the EM-RASM proposed in Chapter 3 in the following: instead of determining the probability of each individual observation belonging to the object border, it computes the probability of each combination of labels. This has two potential advantages over the EM-RASM proposed in this chapter. First, it allows some combinations to be discarded a priori, *e.g.*, a single model point cannot have multiple valid observations. Second, these combinations compete against each other to receive higher probabilities in the estimation of the EM weights, *i.e.*, if an observation associated to a specific model point receives a high probability, the remaining observations associated to that point will have to receiver low probability. This is different from the EM-RASM formulation, in which the competition happens between considering each observation point as valid or invalid, allowing for multiple observations along the same search line to receive high probability.

B.3 Results

The PDAASM algorithm was evaluated using the same experimental setup used in Chapter 3.

Table B.1: Statistical results of the segmentation accuracy using PDAASM and comparison with EM-RASM.

	Dice (%)	**AV (mm)**	**VD (mL)**	**% Good**
EM-RASM	84.2 (9.2)	2.6 (1.7)	11.8 (11.4)	89.2
PDAASM	83.7 (9.3)	2.5 (1.6)	10.9 (14.3)	89.2

Table B.1 shows statistical results of the segmentation accuracy, compared against the 2D EM-RASM approach. The segmentations obtained with PDAASM were similar to the ones obtained with EM-RASM. However, in some cases, the probabilities of the data interpretations, computed in (B.10), were almost binary, *i.e.*, either almost one or almost zero. In other words, the algorithm was too sure of which were the edge segments it could trust or not. Therefore, whenever the algorithm was wrong about which segments should be considered valid, the parameters estimations was essentially based on outliers, leading to poor segmentations.

C

LV Segmentation using Dynamic Programming

Contents

This chapter proposes a different approach for the segmentation of the LV in CMR, which does not rely on shape models. It is based on a simple and fast bottom-up strategy, inspired in [46], that contrasts with the work described throughout this thesis and in most of the literature.

C.1 Introduction

The proposed approach is based on two main assumptions about the LV border: 1) that it is approximately circular in each MR slice; and 2) that it is associated to edges in the image. The segmentation of each MR slice is performed in polar coordinates and involves the following steps. First, an edge map is built so that its valleys roughly correspond to location of the LV border. Second, a dynamic programming (DP) algorithm is applied to determine the optimal path along the edge map, which corresponds to the delineation of the LV contour.

DP is one of the most common choices for data-driven endocardial/epicardial border detection. This class of approaches is rooted in the work of Geiger et al. [57] and used in several works [46, 55, 90, 178, 187]. It searches for the optimal path (*i.e.*, the contour) by associating the image with a graph and defining a cost to each edge through a cost matrix. The design of the cost matrix itself is a challenging task and plays a core role in DP-based approaches, which motivated research on this subject. In [178], a threshold based approach is presented, in which the optimal threshold is found by computing the mean gray value of the maximal edge pixels. These pixels are found by generating orthogonal lines radiating from the epicardial center and collecting, for each line, the gray intensity of the pixel with highest edge value (*i.e.*, maximal edge) within the epicardial contour. A methodology termed iterative multigrid dynamic programming (IMDP) is introduced in [46]. Here, the contours of the LV in ultrasound image sequences are assumed to be one dimensional non-causal first order Markov random fields. DP is applied in a multigrid fashion, *i.e.*, first with a coarse resolution, followed by a refining stage that estimates the segmentation by searching in a smaller range, using a thinner resolution. To obtain the cost matrix, they model the intensity values of the tissues surrounding the ventricle border. In [97] a threshold-based operation is used, where the binary masks (*i.e.*, thresholded images) are jointly used with the global circular shortest path algorithm (GCSP). It is shown that an improved method is achieved by combining the advantages of these two techniques. Fuzzy logic is used in [90] and comprises two stages. The first stage accounts for the pixel gray values and presence of edges, while the second stage comprises the determination of cardiac contours based on fuzzy logic with DP. With these two ingredients, a degree to which each pixel belongs to the cardiac contour is computed, allowing the image to be represented by a membership degree matrix. The final step comprises a graph search on the cost matrix to determine the cardiac contour. In [55] histogram equalization followed

by wavelet transform are used to build the cost matrix. The branch-and-bound algorithm is used [187], where the main focus is to reduce the complexity of finding the optimal path that represents the endocardial border. In [41] the watershed framework is used and a notion of watershed in edge-weighted graphs is introduced for segmenting spatio-temporal images. A shortest path algorithm is also used in [82], which averages all the phases over one cardiac cycle, and contours in each image can be recovered using minimum surface segmentation.

In this work, we follow the algorithm proposed in [46] for the delineation of the LV border in ultrasound images. This approach relies on DP to extract the desired boundary by analyzing the image in polar coordinates. Combining this framework with the cost matrix based on edges proposed in [110] leads to significant improvements in terms of computational complexity, without compromising the accuracy of the segmentation. Finally, an additional step is included that is specific for CMR segmentation, which consists in automatically updating the information about the center and radius of the expected segmentation. This allows the algorithm to accurately segment the whole MR volume without the need for additional user input.

C.2 Proposed Methodology

The goal of the proposed method is to provide a fast and accurate segmentation of the LV in MR volumes. We address this problem by sequentially analyzing the slices (2D images) in a given volume. Starting with an initial guess of the location of the LV in the basal slice, the proposed algorithm aims to determine the location of the endocardium in that particular slice. This segmentation is then propagated to the next slice as an initial guess, and the algorithm is applied to this new slice. This procedure is repeated until all the slices in the volume have been segmented.

Fig. C.1 provides an overview of the proposed approach, illustrating the main steps that are performed for a given slice of the MR volume. The methodology herein proposed is based on the following two main stages: (*i*) conversion of the original image (MR slice) into an (inverted) edge map, whose valleys are considered as *coarse* candidate positions for the endocardium; (*ii*) computation of the optimal path along the edge map, which corresponds to a *fine* estimation of the LV border.

Concerning the first stage, we follow the approach proposed in [110]. Based on an initial guess of the LV center and radius, this algorithm converts the image to polar coordinates. Then, a gradient operator is applied along the radial dimension, in order to extract possible candidates for the location of the LV border. Finally, the resulting gradient image is transformed into an edge map that penalizes pixels with low gradient.

Typically, edge detection is not a reliable approach in CMR, as shown by the example illustrated in Fig. C.2. The left image shows the strongest edge point (highest

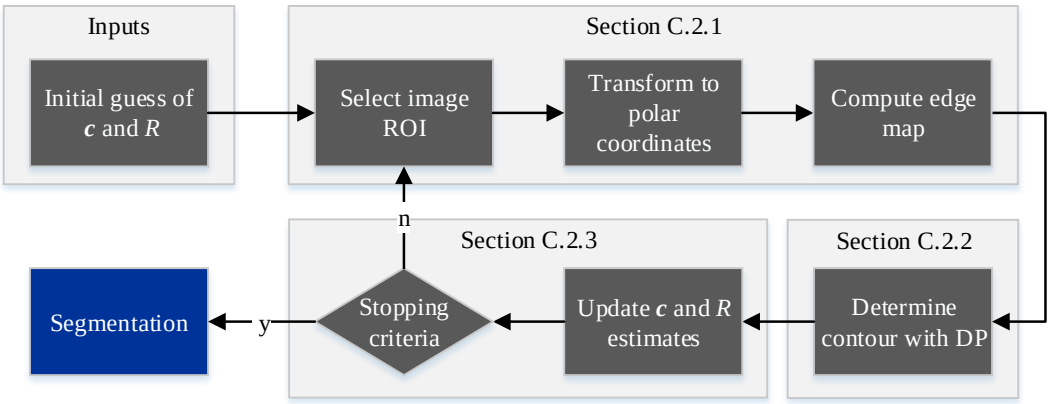


Figure C.1: Overview of the proposed DP methodology applied to each slice of the MR volume, where $\boldsymbol{c}$ and R denote the coordinates of the center and radius of the LV, respectively.

gradient) along the yellow lines, and it is possible to see that many are outliers (red dots). Furthermore, in the right image, in which all the edge points (local gradient maxima along the yellow lines) are shown, some of the edge points located on the LV border (green dots) are not detected. Therefore, additional information is required to avoid being misguided by these outliers. The goal of the second stage of the algorithm is precisely to extract a curve that satisfies specific shape constraints while still being able to follow the valleys of the edge map (*i.e.*, avoid going through pixels with low gradient). This trade-off between the shape constraints and the image information allows the algorithm to avoid erroneous edges related to papillary muscles and other misleading structures. The optimal curve is then converted to Cartesian coordinates for the segmentation of the LV to be obtained.

Once these two stages are complete, the LV center and radius can be recomputed from the segmentation. The updated parameters are used to determine if the initial guess was in agreement with the segmentation or if a new iteration is required. This iterative process allows the algorithm to recover from poor initial guesses.

The two stages of the proposed approach are described in Sections C.2.1 and C.2.2, respectively. The update of the LV center and radius in described in C.2.3.

C.2.1 Computation of the edge map

In this stage, the goal is to compute an edge map from the original MR image, such that its valleys follow the LV border. Since the morphology of the LV is roughly circular, this step is performed in polar coordinates, as in [46].

In order to obtain a representation of the MR image in polar coordinates, an initial guess of the LV's center, denoted by $\boldsymbol{c} = [c_x, c_y]^\top \in \mathbb{R}^2$, and its size, $R \in \mathbb{R}$, have to be provided. Then, the intensity of a particular pixel (r, θ) in the image in polar coordinates is obtained by computing

$$I_{\mathcal{P}}(r, \theta) = I(x, y), \tag{C.1}$$

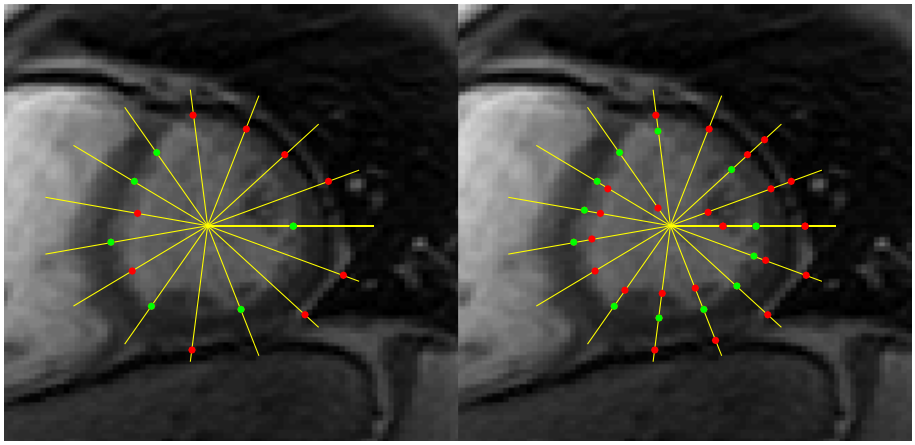


Figure C.2: Detection of edge points (bright to dark) along radial lines. On the left, only the strongest edge point in each line is shown, and on the right all the detected edge points are shown. Green dots belong to the LV border and red dots are outliers.

where the pixel position correspondence is given by

$$x = c_x + r\cos(\theta), \qquad y = c_y + r\sin(\theta). \tag{C.2}$$

This transformation does not guarantee that x, y take integer values, thus we use bilinear interpolation to obtain the value of $I_{\mathcal{P}}(r,\theta)$ (see [169] for details). The pairs (r,θ) for which $I_{\mathcal{P}}$ is defined belong to the domain $\mathcal{D}_r \times \mathcal{D}_\theta$,

$$\mathcal{D}_r = \big\{r_1,\ldots,r_M \in \mathbb{R} : r_i = r_{\min} + (i-1)\Delta r,\ i = 1,\ldots,M\big\} \tag{C.3}$$

$$\mathcal{D}_\theta = \big\{\theta_1,\ldots,\theta_N \in [0,2\pi] : \theta_j = (j-1)\Delta\theta,\ j = 1,\ldots,N\big\}, \tag{C.4}$$

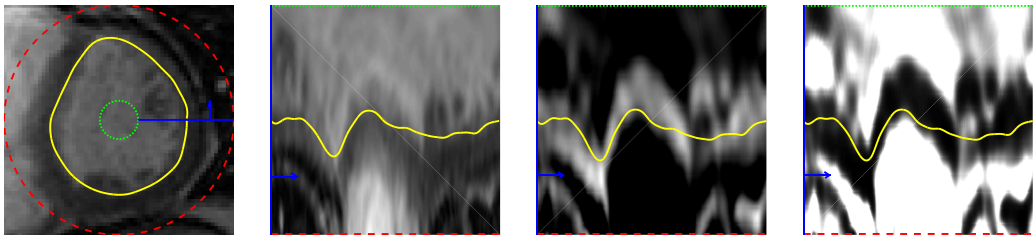
where $\Delta r = \frac{r_{\max}-r_{\min}}{M-1}$, and $\Delta\theta = \frac{2\pi}{N-1}$. The maximum and minimum radii, $r_{\max}$ and $r_{\min}$, define the width of the ring within which the LV border is expected to be found. Also note that $\theta_N = 2\pi = \theta_1$, *i.e.*, the pixels in the left and right borders of $I_{\mathcal{P}}$ correspond to the same positions in the original image.

Once $I_{\mathcal{P}}(r,\theta)$ is computed, a high-pass filter, H, is applied to obtain the radial gradient image $I_{\mathcal{G}}(r,\theta)$. We are only interested in computing the gradient along r, and in transitions from bright to dark, as those depicted in Fig. C.2. The impulse response of the high pass filter is given by

$$H(r) = \begin{cases} 1 & \text{if} \quad 0 < r \leq T \\ -1 & \text{if} \quad -T < r \leq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{C.5}$$

where T is a user defined parameter (in the results section, this parameter was set to $T = 6$). The radial gradient image is obtained by applying the convolution operator

$$I_{\mathcal{G}}(r,\theta) = I_{\mathcal{P}}(r,\theta) \star H(r). \tag{C.6}$$



(a) (b) (c) (d)

Figure C.3: Conversion of an MR image into an edge map in polar coordinates: (a) Original LV image, $I(x, y)$; (b) image in polar coordinates, $I_{\mathcal{P}}(r, \theta)$; (c) image gradient, $I_{\mathcal{G}}(r, \theta)$; and (d) edge map, $\mathrm{e}_{\mathrm{MAP}}(r, \theta)$. The yellow line corresponds to the LV segmentation. The green and red lines correspond to the minimum and maximum radius, respectively, and the blue line and arrow help illustrate the conversion to polar coordinates.

The gradient image $I_{\mathcal{G}}(r, \theta)$ takes values in $]-\infty, \infty[$, with: 1) large, positive values corresponding to edges such as the ones associated with the LV border; and 2) large, negative values corresponding to edges with the opposite gradient direction (from dark to bright); and 3) values close to zero indicating the absence of edges. We wish to transform $I_{\mathcal{G}}(r, \theta)$ into a cost map, such that the first type of edges have zero cost and every other possibility has a cost of approximately 1. To accomplish this, the sigmoid function proposed in [110] is adopted:

$$\mathrm{e}_{\mathrm{MAP}}(r, \theta) = \frac{1}{1 + \exp\big(\lambda(I_{\mathcal{G}}(r, \theta) - k)\big)}, \tag{C.7}$$

where $k > 0$ controls the inflection point, and $\lambda > 0$ controls the sharpness of the sigmoid. In this work, good values for these parameters were empirically determined to be $k = 20$ and $\lambda = 0.04$.

The edge map, $\mathrm{e}_{\mathrm{MAP}}(r, \theta) \in [0, 1]$ is now normalized, and its valleys correspond to rough candidate positions of the LV border. Figure C.3 illustrates the whole pre-processing stage, from the original MR image, $I(x, y)$, in Cartesian coordinates, depicted in Fig. C.3 (a), to the final edge map, $\mathrm{e}_{\mathrm{MAP}}(r, \theta)$, in polar coordinates, depicted in Fig. C.3 (d), whose valleys follow the path of the LV border, shown in yellow.

C.2.2 Contour estimation

The second stage of the algorithm aims to perform the delineation of the LV boundary, computed from the output of the previous stage. More specifically, the goal is to find a curve that follows the valleys of the edge map, $\mathrm{e}_{\mathrm{MAP}}$ (*e.g.*, the yellow line depicted in Fig. C.3 (d)).

In the original Cartesian coordinates, a popular approach to address this problem is the following: find a parametric curve, $\widehat{\mathbf{x}}(s) = (x(s), y(s))$, which is a function of the

curve parameter, $0 \leq s \leq 1$, such that

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathcal{E}(\mathbf{x}), \tag{C.8}$$

where $\mathcal{E}$ is a cost function analytically defined as

$$\mathcal{E}(\mathbf{x}) = \int_s E_{\text{int}}(\mathbf{x}(s)) + E_{\text{ext}}(\mathbf{x}(s))\, ds. \tag{C.9}$$

The first term is an internal energy (*i.e.*, the prior), typically imposing a smoothness constraint. The second term corresponds to the external potential energy function that is defined in terms of the image data, *e.g.*, the edge map e_{MAP} defined in (C.7). This approach is commonly used in deformable contours literature [25, 123].

Here, a similar problem is addressed but in discrete polar coordinates. Recalling that the edge map, e_{MAP}, can be viewed as a $M \times N$ matrix, we wish to determine the curve $\widehat{\mathbf{r}} = [r(1), \ldots, r(N)]^\top$ (*i.e.*, a sequence of radius values), such that $r(j) \in \mathcal{D}_r$ corresponds to the LV radius for angle θ_j (recall (C.3) and (C.4)). Similarly to (C.8), the curve $\widehat{\mathbf{r}}$ is obtained by computing

$$\begin{aligned} \widehat{\mathbf{r}} = \quad & \arg\min_{\mathbf{r}} \quad \mathcal{E}(\mathbf{r}) \\ & \text{s.t.} \qquad r(1) = r(N) \\ & \qquad\quad\; r(j) \in \mathcal{D}_r,\ j = 1, \ldots, N \end{aligned} \tag{C.10}$$

The constraint $r(1) = r(N)$ is required to guarantee that $\widehat{\mathbf{r}}$ is a closed curve in Cartesian coordinates. In this case, the cost function $\mathcal{E}(\mathbf{r})$ is defined as

$$\mathcal{E}(\mathbf{r}) = \sum_{j=1}^{N} E_{\text{int}}(r(j)) + E_{\text{ext}}(r(j)), \tag{C.11}$$

where the image-related term is given by

$$E_{\text{ext}}(r(j)) = \text{e}_{\text{MAP}}(r(j), \theta_j), \tag{C.12}$$

and the prior term,

$$E_{\text{int}}(r(j)) = d(r(j-1), r(j)) \tag{C.13}$$

$$= \begin{cases} 0 & \text{if} \quad |r(j) - r(j-1)| = 0 \\ \eta & \text{if} \quad |r(j) - r(j-1)| = \Delta r \cdot \\ \infty & \text{otherwise} \end{cases} \tag{C.14}$$

is used to impose a smoothness constraint on curve $\mathbf{r}$, by penalizing large variations in consecutive pairs $(r(j-1), r(j))$, with $E_{\text{int}}(r(1)) = 0$.

By replacing (C.12), (C.14) into (C.11), the global cost function can be rewritten as

$$\mathcal{E}(\mathbf{r}) = \text{e}_{\text{MAP}}(r(1), \theta_1) + \sum_{j=2}^{N} \text{e}_{\text{MAP}}(r(j), \theta_j) + d(r(j-1), r(j)). \tag{C.15}$$

Note that this cost function is a sum of local cost functions. Therefore, the optimal cost can be recursively computed through DP [22].

Let $\mathcal{E}_j(r_i)$ denote the optimal cost of reaching a specific position, (r_i, θ_j), in the image, starting in the first column (θ_1). This cost can be recursively computed using the optimal costs for reaching the positions in the previous column

$$\mathcal{E}_j(r_i) = \mathrm{e}_{\mathrm{MAP}}(r_i, \theta_j) + \min_{\rho \in \mathcal{D}_r} \Big[d(\rho, r_i) + \mathcal{E}_{j-1}(\rho) \Big]. \tag{C.16}$$

However, there is a global constraint, expressed in (C.10), which demands that $r(1) = r(N)$. In order to satisfy this constraint, the problem defined in (C.10) can be subdivided into M subproblems: for each subproblem, we choose a different initial value for $r(1) \in \mathcal{D}_r$, and impose that the optimal path starts in this position. This is achieved by changing the first column of the edge map to $\mathrm{e}_{\mathrm{MAP}}(r_i, \theta_1) = \infty, \forall r_i \neq r(1), i = 1, \ldots, M$ (all the paths that do not begin in $r(1)$ will have infinite cost). Then, the following two steps are performed:

1. *Forward step*: Compute the optimal costs of all the curves that start at θ_1 and end at θ_N, using (C.16), and, for each local minimization problem (second term in (C.16)), store the corresponding radii

$$\phi(r_i, \theta_j) = \arg\min_{\rho \in \mathcal{D}_r} \; d(\rho, r_i) + \mathcal{E}_{j-1}(\rho). \tag{C.17}$$

2. *Backward step*: Trace back the optimal path that ends at $r(N) = r(1)$, by using the radii stored in the previous step

$$r(N) = r(1) \tag{C.18}$$

$$r(\tau - 1) = \phi(r(\tau), \theta_\tau), \quad \tau = N, ..., 2 \tag{C.19}$$

The Algorithm 1 summarizes the process of applying DP to find a candidate LV contour starting at a specific position, $r(1)$.

This process is repeated for all possible starting point $r(1) \in \mathcal{D}_r$. Then, the path with the lowest global cost (computed using (C.15)) is selected as the proposed segmentation.

Notice that running Algorithm 1 (A1) for all starting positions may be costly, depending on the number of possibilities, M. Alternatively, Bioucas-Dias et al. [46] proposed to alleviate this by only running A1 two times (2-loop algorithm). They assume that the optimal path close to $j = N/2$ is not influenced by the initialization, $r(1)$. Thus, for whatever starting position they choose in the first run, say $r(1) = r_{\min}$ (blue curve in Fig. C.4), the optimal value for $r(N/2)$ will always be the same. Under this assumption, they simply reorder the two halves of the edge map, as illustrated by the colored rectangles in Fig. C.4 (center), and run A1 a second time, starting at position $r'(1) = r(N/2)$, where $\mathbf{r'} = [r'(1), \ldots, r'(N)]^\top$ denotes the optimal path on the new (reordered) edge map. To

Algorithm 1 Determining the best LV contour candidate starting at $r(1)$

for all $r_i \neq r(1)$ **do**
  $\mathcal{E}_1(r_i) = \infty$
end for
forward step
for $j = 1$ to N **do**
  for $i = 1$ to M **do**
    compute $\mathcal{E}_j(r_i)$ using (C.16)
    compute $\phi(r_i, \theta_j)$ using (C.17)
  end for
end for
backward step
select ending point $r(N) = r(1)$
for $j = N$ to 2 **do**
  $r(j-1) = \phi(r(j), \theta_j)$
end for

obtain the solution for the original edge map, the two halves of $\mathbf{r}'$ are rearranged back to the original order (red curve in Fig. C.4 (right)). This way, the complexity of the algorithm is reduced, allowing the for very fast segmentations to be obtained, independently of the choice of M.

Once the optimal (or sub-optimal) path has been computed, the LV segmentation in the original Cartesian coordinates, denoted by $\mathbf{x}$, is obtained by transforming the coordinates of each point in $\mathbf{r}$ as follows

$$\mathbf{x}(j) = \mathbf{c} + r(j)\big[\cos\theta_j, \sin\theta_j\big]^\top,\ j = 1, \ldots, N. \tag{C.20}$$

C.2.3 Automatic estimation of the c and r parameters

The algorithm described above provides a path $\boldsymbol{r} \in \mathbb{R}^N$ that defines the contour of the LV along the edge map, $\mathrm{e}_{\mathrm{MAP}}$. However, this path corresponds to a good estimate of the LV border under the premise that the LV is roughly a circle with a specific center, $\boldsymbol{c}$, and radius, $R \in \mathcal{D}_r$ (recall (C.2) and (C.3)). In most cases, these parameters are provided by using the segmentation obtained in the previous slice, which may be inaccurate. Consequently, the resulting LV segmentations may not be correct. This section describes how the estimates of R and $\boldsymbol{c}$ are refined based on the "optimal" path $\boldsymbol{r}$.

The initial premise that the LV is roughly a circle on the MR slice means that the path $\boldsymbol{r}$ is expected to be roughly a horizontal line along the edge map, $\mathrm{e}_{\mathrm{MAP}}$. Moreover,

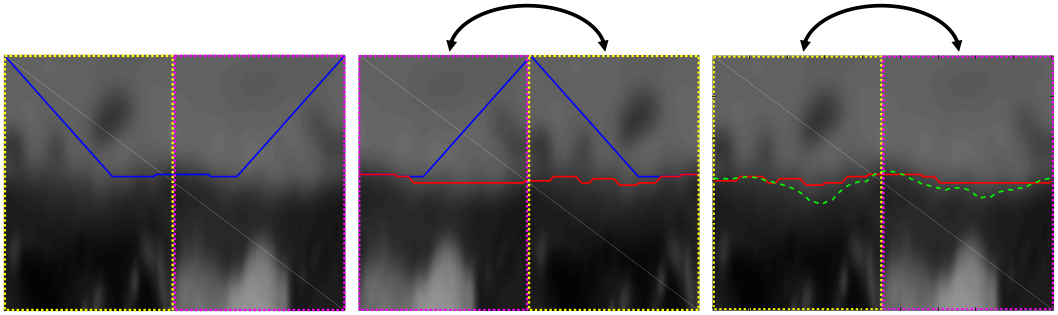


Figure C.4: 2-loop algorithm for LV segmentation using DP. The blue line depicts the solution of A1 with initial position $r(1) = r_{\min}$; the red line depicts the solution of A1 on the reordered edge map with initial position $r'(1) = r(N/2)$; the green dashed line represents the ground truth segmentation. The yellow and magenta rectangles and black arrows illustrate how the reordering of the edge map and the optimal path works.

if the initial estimate of the LV radius, r, is accurate, then the straight line should be located along the middle of the edge map, as shown in Fig. C.5 (a). An example of the segmentation obtained from inaccurate estimates of R and $\boldsymbol{c}$ is depicted in Fig. C.5 (b). On one hand, an inaccurate estimate of the center $\boldsymbol{c}$ leads to a sinusoidal curve along the edge map, instead of a straight line. On the other hand, if the expected radius is smaller (larger) than the actual LV radius, then the resulting curve is closer to the bottom (top) part of the edge map. In the extreme case, the segmentation may not be able to follow the actual LV contour, as shown in the figure, leading to a path that follows the border of the edge map. Thus, it is necessary to update the estimates of R and $\boldsymbol{c}$. This can be achieved through the analysis of the path, $\boldsymbol{r}$, as follows.

Let $\bar{r}$ be the average distance of the contour to the center estimate $\boldsymbol{c}$

$$\bar{r} = \frac{1}{N} \sum_{j=1}^{N} r(j), \tag{C.21}$$

where $r(j)$ is the j-th component of $\boldsymbol{r}$. The estimate of the expected LV radius is updated by

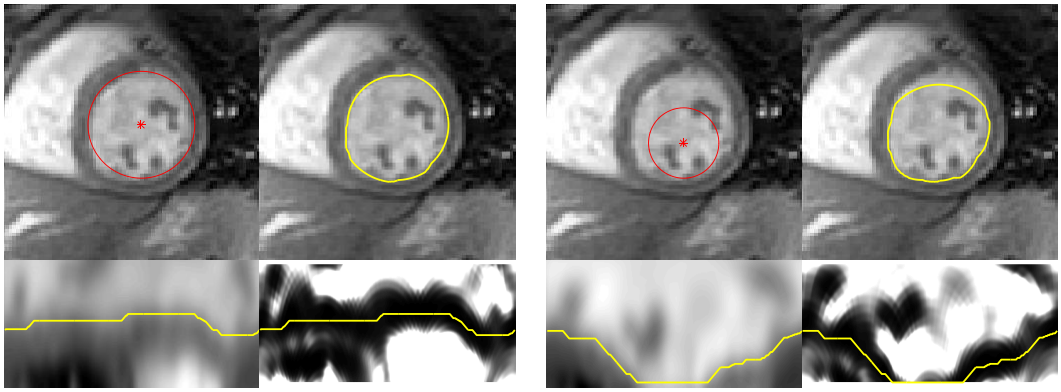
$$R \leftarrow \bar{r}. \tag{C.22}$$

Now, let $\boldsymbol{p}(j) \in \mathbb{R}^2$ be the position of the j-th contour point, associated to $r(j)$ and θ_j, in Cartesian coordinates

$$\boldsymbol{p}(j) = \boldsymbol{c} + r(j) \begin{bmatrix} \cos\theta_j \\ \sin\theta_j \end{bmatrix}, \tag{C.23}$$

and let $\overline{\mathbf{c}}$ be defined as the centroid of those points,

$$\overline{\mathbf{c}} = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{p}(j). \tag{C.24}$$



(a) (b)

Figure C.5: Examples of the segmentation obtained from (a) an accurate and (b) an inaccurate estimate of the LV radius, R, and center, $\boldsymbol{c}$. For each case, we show the original MR slice (top), the MR slice in polar coordinates (bottom left), and the edge map (bottom right). The red circles are the initial estimate of the center and radius and the yellow curves show the corresponding segmentations obtained using the proposed algorithm.

The updated estimate of the LV center is obtained by

$$\boldsymbol{c} \leftarrow \overline{\mathbf{c}}. \tag{C.25}$$

This update scheme allows the path, $\boldsymbol{r}$, to iteratively converge towards the LV border even when the initial estimate of its center and radius are inaccurate. Consider the example of the inaccurate initial estimate depicted in Fig. C.5 (b). Applying three iterations of the update scheme described above leads to the results shown in Fig. C.6. In this figure, it is possible to see that, in each iteration, the segmentation becomes more similar to a straight line in the polar coordinates space (bottom images) and gets closer to the correct LV segmentation in the original image (top images).

C.3 Results

The proposed method was evaluated on the 33 sequences of the public dataset of CMR data described in Section 3.6.1. The quantitative evaluation of the segmentation accuracy is based on the four metrics described in Section 3.6.2. Results are shown for the following configurations: 1) the algorithm A1 without the automatic update of $\boldsymbol{c}$ and R described in Section C.2.3, 2) A1 with automatic updates (A1+AU), and 3) the 2-loop algorithm with automatic updates (2L+AU). The computational performance of these configurations is compared to demonstrate the advantages of using the 2L algorithm.

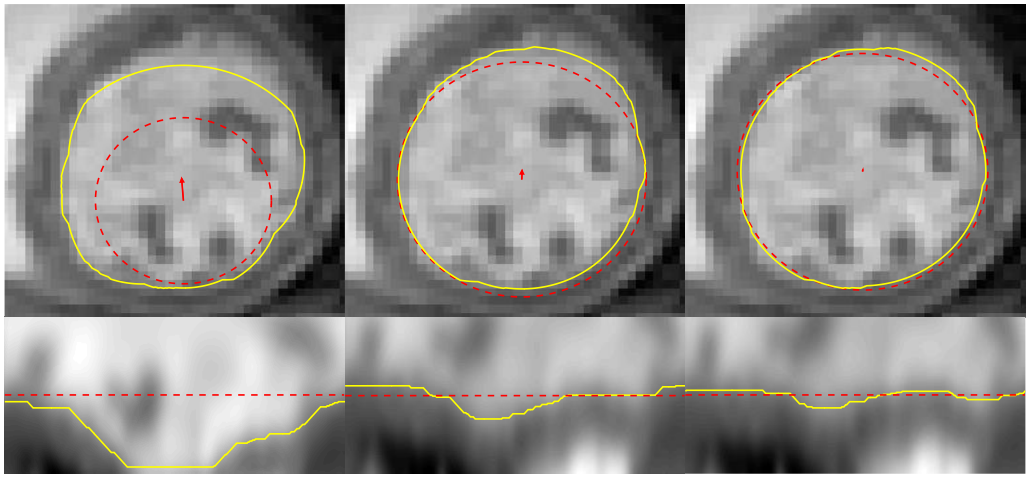


Figure C.6: Updated of the LV center and radius estimates, $\boldsymbol{c}$ and R, and corresponding segmentations. The first column shows the initial center and radius estimate and the following columns show new iterations of the update scheme. The red circles depict the center and radius estimates and the yellow curve shows the corresponding segmentation. The red arrows correspond to the center update.

C.3.1 Computational Performance

The computational complexity of the algorithms, A1 and 2L, depend on two main parameters: M and N, which determine the size of the edge map. These two parameters influence the performance of the proposed approaches both in terms of running, and quality of the segmentations.

Figure C.7 shows the Dice coefficient and time spent per 100 iterations, T, for different values of $M = \{20,\ 30,\ \ldots,\ 170,\ 180\}$, assuming $N = 361$ fixed, and $N = \{91, 181, 361, 541\}$, assuming $M = 120$ fixed. It is possible to see that, as expected, A1 depends linearly on the value of M, whereas the complexity of 2L does not depend on M. Consequently, the time required to run A1 may be significantly larger than that of 2L. However, in terms of segmentation quality, the difference between these two approaches is almost negligible. The value of the Dice coefficient reaches a plateau at around $M = 100$, with the maximum being achieved at $M = 120$. Larger values of M only decrease the speed of the overall algorithm, without improving its accuracy. Regarding the number of columns in the edge map, N, both algorithms have a similar behavior: the time per 100 iterations increases linearly with N, although at very different scales, while the Dice coefficient stabilizes for $N \geq 181$, with a maximum reached at $N = 361$. For the following section, the size of the edge map was set to $M = 120$ and $N = 361$.

C.3.2 Segmentation Accuracy

Table C.1 shows the statistical results of the three approaches discussed in this appendix. Regarding the accuracy of the segmentations, all the approaches have a similar

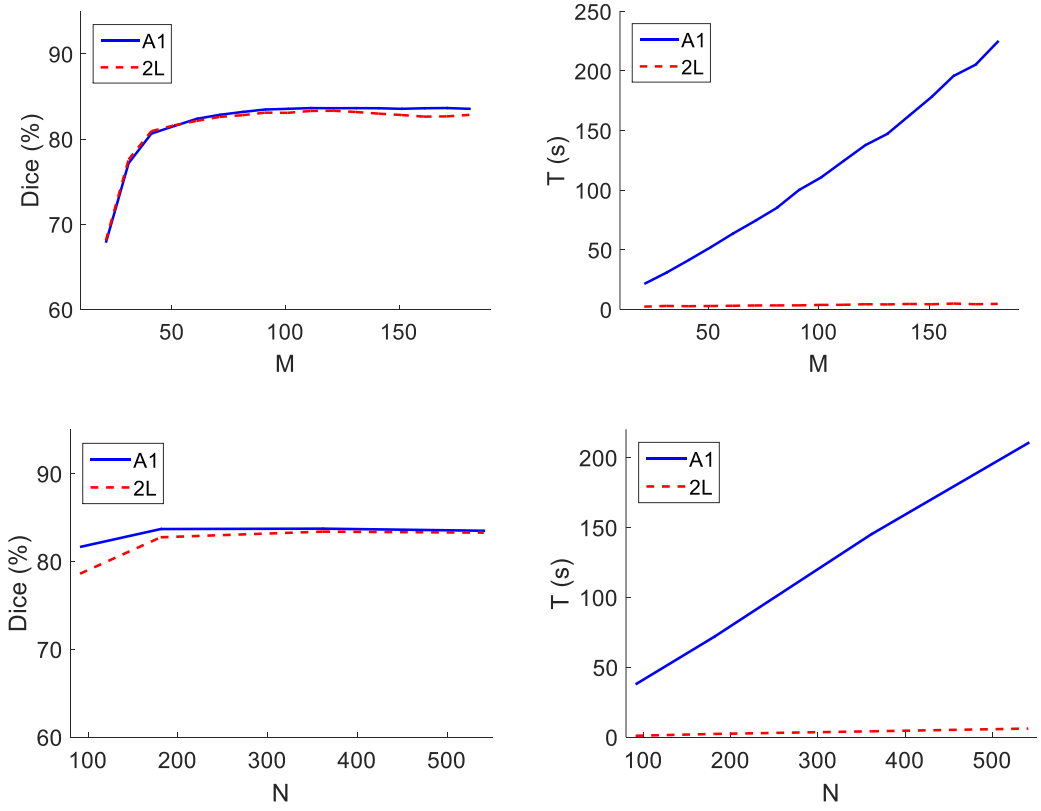


Figure C.7: Comparison of the computational performance of A1 and 2L: (left) the Dice coefficient and (right) the time spent per 100 iterations, T. In the top row, different values of M were tested, using $N = 361$ (fixed); in the bottom row, different values of N were tested, using $M = 120$ (fixed).

performance. Nonetheless, the performance of A1+AU is better than 2L+AU, which in turn is better than A1. Taking into account the time required to segment each volume, then the 2L+AU approach is considerably more desirable than the other two. Comparing these results with the EM-RASM formulation, it is also possible to see that 2L+AU is faster, although the accuracy of EM-RASM is superior.

The fact that the proposed approach segments the volume slices sequentially influences that importance of the automatic update of the LV center and radius. When one of the slices is incorrectly segmented, the error may propagate to the following slices. In these cases, updating the center and radius estimates allows the algorithm to recover from segmentation errors. The example shown in Fig. C.8 illustrates this advantage. The figure shows the segmentation of a particular volume using A1 (top row) and A1+AU (bottom row). In both cases, the initial guess of the center and radius in the basal slice (left) is the same. It is possible to see that the A1 algorithm is not able to accurately segment the LV. Even though the error is not significant, it escalates to meaningless segmentations in the following slices. By using the automatic update scheme, the A1+AU algorithm is able to recover from these poorer initial estimates.

Table C.1: Statistical performance of A1, A1+AU and 2L+AU, and comparison with the EM-RASM algorithm. Each entry shows the mean value and standard deviation.

	Dice (%)	**AV (mm)**	**VD (mL)**	**% Good**	**Time per volume (s)**
EM-RASM	**84.2 (9.2)**	2.6 (1.7)	11.8 (11.4)	89.2	2.6
A1	82.2 (9.1)	2.8 (1.4)	**11.5 (12.9)**	87.2	11.2
A1+AU	83.5 (9.1)	**2.6 (1.3)**	13.9 (17.4)	**89.7**	55.4
2L+AU	82.8 (11.2)	2.7 (1.7)	15.5 (21.0)	88.8	**1.5**

Fig. C.9 shows the average Dice coefficient obtained using 2L+AU for each of the 33×20 volumes in the dataset. It is possible to see that the proposed algorithm performs better during the diastolic phase (approximately frames 1-3 and 12-20) than in the systolic phase (6-10), which is expected since the edges along the LV border are clearer in these frames. Most segmentation failures (red pixels in Fig. C.9) were either caused by the presence of papillary muscles (see bottom right example in Fig. C.10), or because the segmentation was pulled towards the LV's outer border (see bottom left examples in Fig. C.10). These cases are hard to segment using an edge-based approach, since the strongest edge does not correspond to the desired border.

C.4 Conclusions

A fast methodology for the segmentation of the LV in CMR is presented in this appendix. This approach is built under the assumption that the LV segmentation in each slice has approximately a circular shape. Under this assumption, we propose to transform the original MR slice into an edge map in polar coordinates, whose valleys roughly follow the LV border. Then, the delineation of the LV contour is obtained using a DP approach. The results show that this approach is able to achieve good results, although its accuracy is inferior to the EM-RASM formulation used in this thesis. The proposed approach is also able to segment a whole volume in 1.5 seconds, *i.e.*, it provides fast and accurate segmentations that would significantly reduce the time spent by cardiologists in this laborious task.

The drawback of the proposed algorithm is the fact that it relies on edge detection

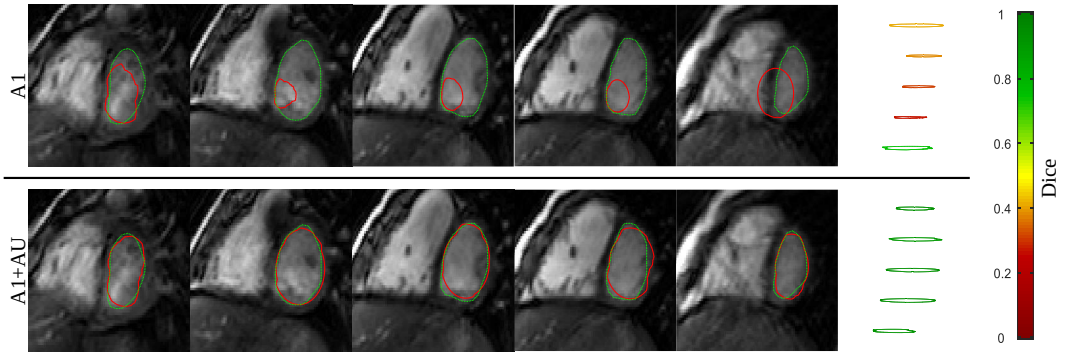


Figure C.8: Comparison between A1 (top row) and A1+AU (bottom row). Each column shows a slice of the volume, from the basal slice (left) to the apex (right); the red contour is obtained using the automatic algorithm and the green is the GT. The last column shows a 3D view of the volume segmentations and the corresponding color-coded Dice coefficient.

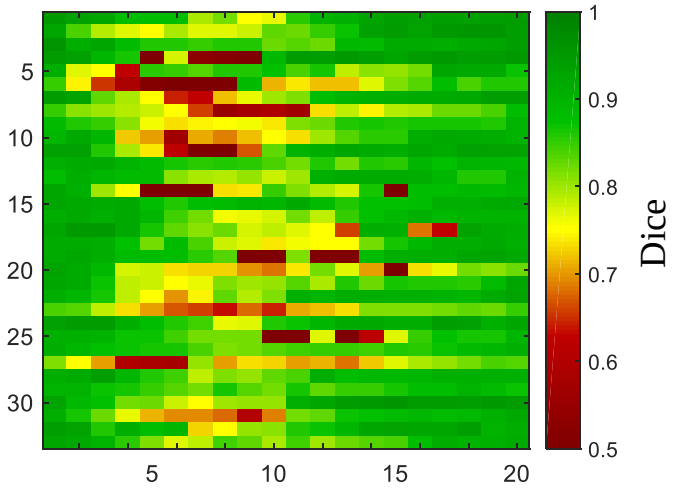


Figure C.9: Discriminated evaluation of the segmentation of each volume in the dataset (33 patients $\times$ 20 frames). The colormap indicates the Dice coefficient, in which green pixels correspond to good segmentation and red to poor segmentations.

to identify the position of the LV border. Although this is a common approach in the literature, the outer wall of the LV and the presence of papillary muscles may misguide these algorithms. Therefore, future work should focus on using a more robust approach to compute the edge map, instead of relying on edge detection only.

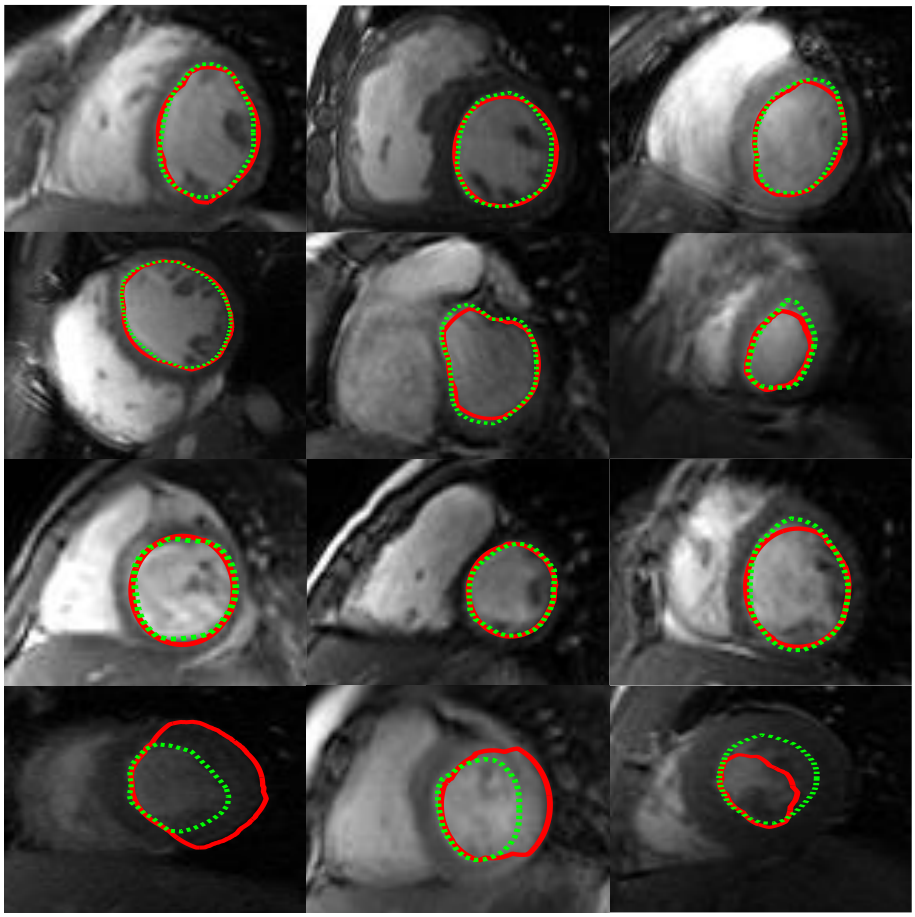


Figure C.10: Examples of segmentations obtained using 2L+AU (in red) and comparison with the GT (green). Each image shows one slice from a particular patient.