

Cardiac MRI Segmentation with Conditional Random Fields

by

Janto F. Dreijer

Dissertation presented for the degree of Doctor of Engineering
in the Faculty of Engineering at
Stellenbosch University

Supervisors: Ben M. Herbst, Johan A. du Preez

December 2013

Pectora roborant cultus recti

Declaration

By submitting this thesis/dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: September 2013

Abstract

This dissertation considers automatic segmentation of the left cardiac ventricle in short axis magnetic resonance images. The presence of papillary muscles near the endocardium border makes simple threshold based segmentation difficult.

The endo- and epicardium are modelled as two series of radii which are inter-related using features describing shape and motion. Image features are derived from edge information from human annotated images. The features are combined within a Conditional Random Field (CRF) – a discriminatively trained probabilistic model. Loopy belief propagation is used to infer segmentations when an unsegmented video sequence is given. Powell’s method is applied to find CRF parameters by minimising the difference between ground truth annotations and the inferred contours. We also describe how the endocardium centre points are calculated from a single human-provided centre point in the first frame, through minimisation of frame alignment error.

We present and analyse the results of segmentation. The algorithm exhibits robustness against inclusion of the papillary muscles by integrating shape and motion information. Possible future improvements are identified.

Opsomming

Hierdie proefskrif bespreek die outomatiese segmentasie van die linkerhartkamer in kort-as snit magnetiese resonansie beelde. Die teenwoordigheid van die papillêre spiere naby die endokardium grens maak eenvoudige drumpel gebaseerde segmentering moeilik.

Die endo- en epikardium word gemodelleer as twee reekse van die radiusse wat beperk word deur eienskappe wat vorm en beweging beskryf. Beeld eienskappe word afgelei van die rand inligting van mens-geannoteerde beelde. Die funksies word gekombineer binne 'n CRF (Conditional Random Field) – 'n diskriminatief afgerigte waarskynlikheidsverdeling. "Loopy belief propagation" word gebruik om segmentasies af te lei wanneer 'n ongesegmenteerde video verskaf word. Powell se metode word toegepas om CRF parameters te vind deur die minimering van die verskil tussen mens geannoteerde segmentasies en die afgeleide kontoere. Ons beskryf ook hoe die endokardium se middelpunte bereken word vanaf 'n enkele mens-verskafte middelpunt in die eerste raam, deur die minimering van 'n raambelyningsfout.

Ons analiseer die resultate van segmentering. Die algoritme vertoon robuustheid teen die insluiting van die papillêre spiere deur die integrasie van inligting oor die vorm en die beweging. Moontlike toekomstige verbeterings word geïdentifiseer.

Acknowledgements

I would like to extend my gratitude towards my supervisors Professors Ben Herbst and Johan du Preez for their advice and guidance. I would like to thank my friends and family for their continuous support.

Contents

Declaration	i
Abstract	ii
Opsomming	iii
Acknowledgements	iv
Contents	v
List of Figures	viii
List of Tables	xii
Nomenclature	xiv
1 Introduction	1
1.1 Problem description	1
1.2 Background	2
1.2.1 Cardiac MRIs	2
1.2.2 MRI capture and annotation	4
1.3 Literature overview	5
1.4 Objectives of this study	7
1.5 Contributions	7
1.6 Overview of this dissertation	8

2	Probabilistic models	11
2.1	Probability distribution as a model	11
2.2	Generative and discriminative models	12
2.3	Conditional random fields	15
2.4	Efficient factorisation	16
3	Model description	18
3.1	Problem formulation	18
3.2	Centre point estimation	21
3.3	The CRF model	23
3.4	Feature functions	23
3.4.1	Feature function based on edge classifiers	24
3.4.2	Spatial and temporal feature functions	27
3.4.3	Inner-outer radius feature functions	30
3.5	Summary	32
4	Segmentation	33
4.1	Belief propagation	34
4.2	Loopy belief propagation	36
4.3	Message schedule	40
4.4	Implementation	43
4.5	Summary	44
5	Parameter estimation	45
5.1	Maximum likelihood estimation	45
5.2	The loss functions	46
5.2.1	Landmark distance	47
5.2.2	Dice error metric	48
5.3	Black box parameter estimation	49
5.3.1	Gradient approximation method	50
5.3.2	Gradient-free method	51
5.3.3	Convexity	52

5.4	Remarks	53
6	Results and discussion	55
6.1	York cardiac segmentation dataset	56
6.1.1	Qualitative segmentation analysis	57
6.1.2	Quantitative segmentation analysis	59
6.1.3	Sensitivity to initial centre point placement	62
6.2	Sunnybrook cardiac segmentation dataset	63
6.3	Summary	67
7	Concluding remarks	70
7.1	Model limitations	70
7.2	Future work	71
7.3	Summary	72
	Bibliography	74
A	Edge properties	81
B	User interaction	85
C	Additional 3D image	87

List of Figures

1.1	Simplified diagram of the left and right ventricles.	2
1.2	MRI short axis view of cardiac ventricles and surrounding structure.	2
1.3	Video sequence of MRI short axis view of cardiac ventricles. The left ventricle undergoes contraction (systole) during images $T = 0 \dots 8$ and relaxation (diastole) during $T = 9 \dots 20$. Only even numbered images $T = 0 \dots 18$ are shown for brevity.	3
1.4	MRI short axis view of ventricles with human annotated (inner and outer) contours shown in yellow. Surrounding tissue is omitted for illustrative purposes.	4
1.5	Presence of papillary muscles obscure the edge of the inner contour due to its similar intensity to the cardiac wall. Human annotated inner and outer contours are shown in yellow. Surrounding tissue is cropped for clarity. . . .	5
2.1	Example of two linearly separable classes with complex distributions.	13
2.2	Two example factor graphs.	17
3.1	A representation of coordinates on the inner and outer contours in Cartesian and polar coordinates.	19
3.2	Log-polar transform with human annotated (ground truth) inner and outer contours in yellow. The centre point is chosen near the middle of the endocardium.	20

3.3	A partial factor graph of the temporal and spatial relationships between radius variables, ρ , in a single contour and rows in the log-polar transformed image, $d_n(t)$. Factor labels and some variable labels are omitted for clarity.	24
3.4	Window extracted in log-polar space and corresponding circular sector in untransformed image. The height of the window in these figures are set to eight angular units for illustrative purposes.	25
3.5	Inner and outer feature function responses to the image in Figure 3.2. The ground truth contours are indicated with yellow lines.	28
3.6	Histogram of the ratio between the inner and outer radii, $r_n^{\text{in}}(t)/r_n^{\text{out}}(t)$, against time, generated from a training dataset.	30
3.7	Response of feature, f_1^{cross} : difference between inner and outer log radii, $ \rho_n^{\text{out}}(t) - \rho_n^{\text{in}}(t) $ against time.	31
4.1	Factor graph of a “chain” and propagated messages.	34
4.2	Messages propagated in a partial factor graph to construct $m_R(n, t)$. The inwards message from the outer contour, $m_I(n - 1, t)$ is not shown.	38
4.3	Example of a non-continuous inner contour.	41
5.1	Illustration of the landmark distance of a single point and the Dice similarity of two areas.	47
5.2	Objective function evaluations (green points) against iteration. The blue line indicates a lower bound, i.e. the best value encountered up to this iteration. The vertical bars indicate iterations of decreasing error.	50
5.3	Line scan of the objective function, illustrating local minima. Red areas represent negative gradients and blue, positive.	54
6.1	Selection of images and their automatically segmented contours (inner contour is blue and outer is green) inferred from a testing set. The blue dot in the middle of the endocardium is the estimated centre point.	57
6.2	A selection of images that are incorrectly segmented by the system.	58
6.3	Areas inside contours of human annotation against the areas inside automated segmentation of testing data.	59

6.4	Contour Dice errors over time. For illustrative purposes, a random real value between zero and one was added to each frame number. The geometric mean for each frame number is indicated by a black line.	60
6.5	Contour Dice errors over different slices. For illustrative purposes, a random real value between zero and one was added to each slice depth. The geometric means for the slice positions are indicated by the black line. . . .	60
6.6	Incorrect automatically segmented contours (inner contour is blue and outer is green) of Subject 8 due to a disappearing endocardium. The blue dot in the middle of the endocardium is the estimated centre point.	61
6.7	Sensitivity of segmentation with regard to incorrect centre point in first frame. The value d is the fractional distance between the ground truth centre point and the ground truth inner contour. See text for more detail.	62
6.8	Examples from the Sunnybrook dataset with thin cardiac walls, as indicated by the yellow arrows.	64
6.9	Selection of images and their automatically segmented contours (inner contour is blue and outer is green). The blue dot in the middle of the endocardium is the estimated centre point.	65
6.10	Bland-Altman plots of end-systolic volume, end-diastolic volume, ejection fraction, mass, end-systolic area and end-diastolic area. Each dot represents a video sequence in the validation set. The vertical axes are of <i>automatically determined</i> minus <i>ground truth</i> values. Mean difference and standard deviation lines are included.	68
6.11	Selection of images from validation patient images with hypertrophy and their low quality automatically segmented contours (inner contour is blue and outer is green). The blue dot in the middle of the endocardium is the estimated centre point.	69
A.1	Edge classifier coefficients for the inner and outer contours for different angles. Each of the eight direction-dependent classifiers is represented by a grey line. The average for each coefficient over the classifiers in all the directions is represented by the black line.	84

C.1 Three dimensional construction of an automatic segmentation at end-diastole.	
.....	87

List of Tables

6.1	Segmentation errors as reported by different authors. These results are, however, not strictly comparable since they are based on different datasets and the error criteria differ.	61
6.2	Average Dice similarity metric and Average Perpendicular Distance (APD) for our segmentation of the Sunnybrook validation set (before and after training) and results reported by the different MICCAI challenge entries.	66
6.3	Patient specific Average Perpendicular Distance (APD), and Dice similarity between annotations and ground truth of the Sunnybrook validation set. . .	67

Nomenclature

$c(t)$	Centre points for video frame at time t
$D(t)$	Log polar transform of a single image, $I(t)$, at time t
D	All log polar images in a video sequence, $D = \{D(t)\}_{t=0,\dots,T-1}$
$E(\rho \theta, D)$	Energy of a segmentation
$f_q(\rho_q, D)$	Feature function
$F_q(\rho_q, D)$	Factor function, i.e. weighted feature $\theta_q f_q(\rho_q, D)$
$I(t)$	Greyscale image in an MRI video sequence
$I(t, \mathbf{p})$	Greyscale pixel value at position $\mathbf{p} = x, y$ in image $I(t)$
n	Node index with $0 \leq n < N$
ρ	Discretised log radius value $0 \leq \rho < M$
M	Radius discretisation (chosen as 256)
ρ_n or $\rho_n(t)$	Log radius for node n i.e. radius with a corresponding angle $\phi_n = n \frac{2\pi}{N}$
N	Number of nodes in a contour (chosen as 128)
P	Probability distribution
\tilde{P}	Pseudolikelihood estimate of P
Q	Probability distribution maximised by loopy belief propagation
$\rho(t)$	Segmentation for a single frame of a video (i.e. $\rho_1(t), \dots, \rho_N(t)$)
ρ	Segmentation for video sequence (i.e. a collection of $\rho(t)$)
T	Number of images in a video sequence
t_{ES}	Frame at end systole (chosen as $t = 8$ if $T = 20$)
θ	Feature function weight with $\theta \in \mathbb{R}$ and $0 \leq \theta$
θ	Parameter vector for CRF
θ^*	Optimal parameter vector w.r.t. some criterion
$Z(\theta)$	Partition function

Chapter 1

Introduction

1.1 Problem description

Cardiovascular diseases are the leading cause of death worldwide [48]. The quality of patient diagnosis and prognosis depends on the accurate measurement of cardiac properties, which can be derived from annotations of images of the cardiac structure. Annotation by human specialists in modalities such as magnetic resonance imaging is a time intensive process. Additionally, when papillary muscles are close to the endocardium a strong edge is absent which can lead to inconsistent annotations. Accurate segmentation therefore needs integration of spatial and temporal information. Since only one frame is annotated at a time, it is difficult for a human to take temporal information into account during annotation.

Automation of the segmentation process therefore has numerous benefits, even if used only to aid and not fully replace a human annotator. A number of automated segmentation techniques exist, but the integration of spatial and temporal information remains challenging. These challenges are addressed in this study.

1.2 Background

1.2.1 Cardiac MRIs

The heart's primary function is that of receiving oxygen poor blood from the body, moving it through the lungs and then distributing the oxygenated blood to the rest of the body.

Figure 1.1 illustrates a short-axis view of the basic cardiac structure.

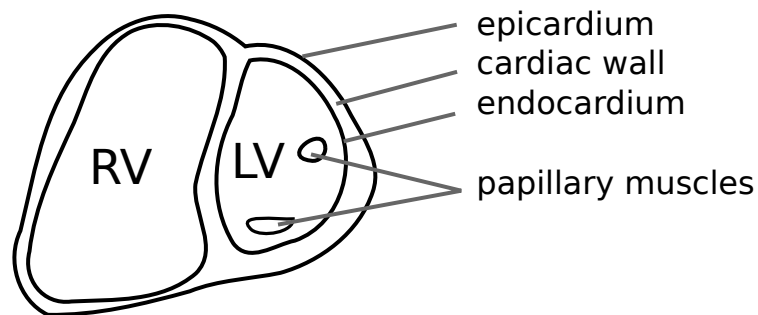


Figure 1.1: Simplified diagram of the left and right ventricles.

Stated simply, the right ventricle receives de-oxygenated blood from the body during ventricle relaxation (diastole) and pumps it to the lungs on contraction (systole). Simultaneously oxygenated blood is received from the lungs into the left ventricle during diastole and forced to the rest of the body on systole. The left ventricle therefore operates at a higher pressure than the right and consequently has more muscle mass.

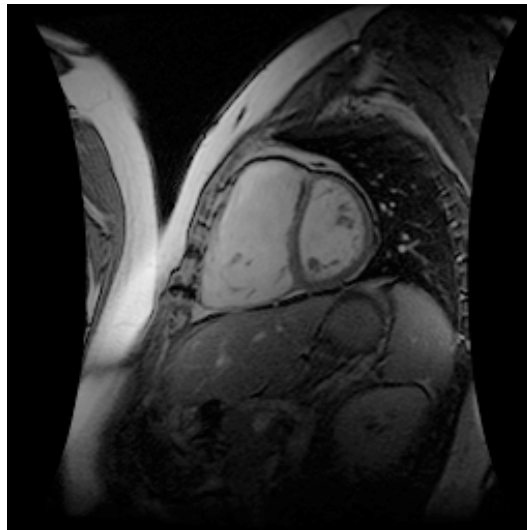


Figure 1.2: MRI short axis view of cardiac ventricles and surrounding structure.

Figure 1.2 shows a Magnetic Resonance Imaging (MRI) short-axis view of the cardiac ventricles and their surrounding structure. Magnetic Resonance Imaging is used in the visualisation of internal structure by producing images where high pixel intensity corresponds to a higher water content.

Cardiac MRI has a number of important advantages when compared to alternative imaging modalities such as X-ray computed tomography and ultrasound. It is non-invasive, does not emit ionising radiation, can be used with multiple imaging planes and has a wide topological field of view. Additionally, as MRIs respond to water content it can be used to produce images that have a high discriminative contrast between blood, the myocardium and surrounding soft tissues.

Magnetic Resonance Imaging can also produce a series of images in succession. Figure 1.3 contains frames from a video sequence of MRI short axis view of the cardiac ventricles, synchronised to a single cardiac cycle. The left ventricle undergoes contraction (systole) in the images $T = 0 \dots 8$ and relaxation (diastole) during $T = 9 \dots 20$.

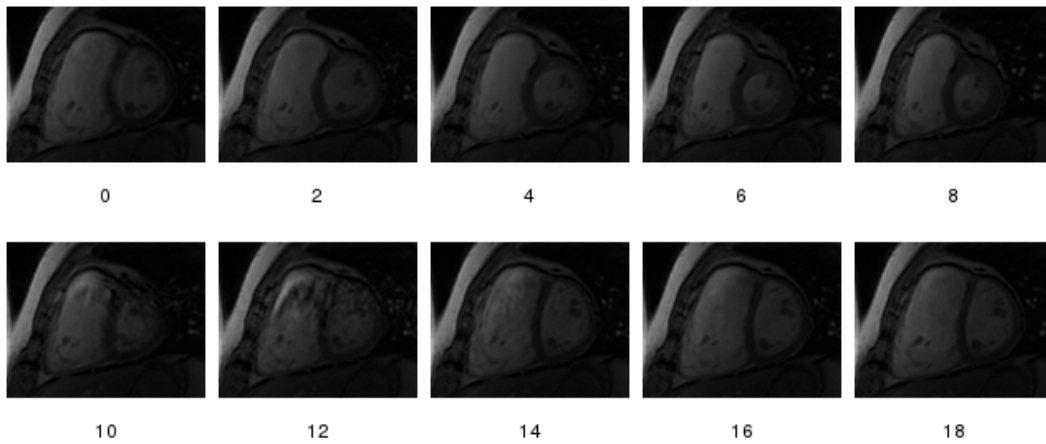


Figure 1.3: Video sequence of MRI short axis view of cardiac ventricles. The left ventricle undergoes contraction (systole) during images $T = 0 \dots 8$ and relaxation (diastole) during $T = 9 \dots 20$. Only even numbered images $T = 0 \dots 18$ are shown for brevity.

The direction of blood flow through the ventricles is controlled by heart valves. The opening and closing of the valves are driven by the pressure gradient across the valves. Papillary muscles (Figures 1.1 and 1.2) and chordae tendineae (“heart strings”), inside the ventricles, prevent the valves from inverting, which could cause backward flow of blood from the ventricles into the atria.

1.2.2 MRI capture and annotation

The cardiac MRI acquisition protocol often requires that the subject holds their breath while a video sequence of successive images is taken. Focus of the MRI instrumentation is then moved to a different short axis slice and the acquisition is repeated. If the subject breathes between different slices, then, due to different levels of inhalation and expiration, the cardiac structure can undergo significant displacement. For this reason we have not integrated information from different spatial slices into our model and focus only on the annotation of temporal sequences of images for a single slice, as shown in Figure 1.3.

Properties of the left ventricle, such as volume, ejection fraction [49] and wall thickness are important indicators for the diagnosis and prognosis of many heart-related problems. Motion and deformation descriptors also include ventricle boundary wall motion, endocardial motion, wall thickening and strain analysis. Many of these are conveniently extracted from Magnetic Resonant Images (MRIs). The calculation of these properties requires accurate annotation of the left ventricle to isolate it from its surrounding structure. Figure 1.4 contains an example of human annotated inner and outer contours of the left ventricle in a single image.

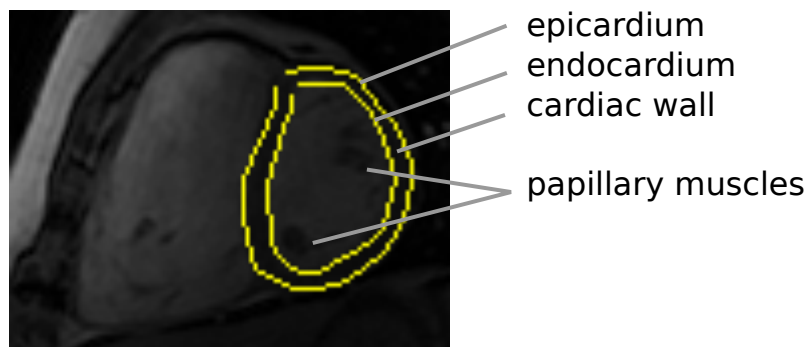


Figure 1.4: MRI short axis view of ventricles with human annotated (inner and outer) contours shown in yellow. Surrounding tissue is omitted for illustrative purposes.

Manual annotation is a tedious process and lacks consistency between human annotators [2, 21] and even between separate annotations by the same annotator. This problem primarily arises from the apparent ambiguity as to the extent to which the papillary muscles influence and, possibly, obscure the endocardium border. For research on the effects that discrepancies in annotations of the papillary muscles can have on the calculation of left

ventricle function and mass see e.g. [7,21,56]. When modelling the structural properties of the ventricle wall, it is often desirable to include these muscles inside the inner contour.

The examples in Figure 1.5 illustrate the presence of papillary muscles close to the endocardium border and a human annotator’s segmentation. From inspection of the human annotation, it is clear that the presence of the endocardium border is inferred from prior external information of the motion and shape of the ventricle, and not only from what is available in the image itself, such as strong intensity gradients. Such considerations are likely to lead to inconsistencies, in particular when there is little difference between the intensity of the endocardium and surrounding structure.

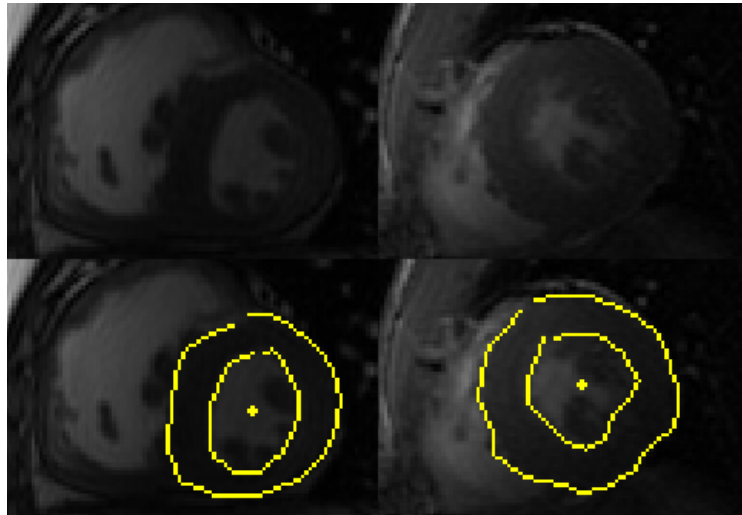


Figure 1.5: Presence of papillary muscles obscure the edge of the inner contour due to its similar intensity to the cardiac wall. Human annotated inner and outer contours are shown in yellow. Surrounding tissue is cropped for clarity.

1.3 Literature overview

A number of automated techniques have been developed for the segmentation of the left ventricle from its surrounding structure (see e.g. [16,44,46]). For a review of deformable models in medical image analysis see e.g. [36]. We will briefly discuss the most popular techniques.

Active Shape Models (ASMs, [10]) track the edges in a video sequence by modelling the contour shape using methods such as Principal Component Analysis. This is often

combined with a Kalman filter to incorporate temporal smoothing in an online tracking framework. Typically only past information is used and future images ignored, often with adverse consequences if the tracked object temporarily disappears from view or significantly reduces in size. Note also that this technique often relies on annotating the first frame.

Andreopoulos and Tsotsos [3] fit a 3D Active Appearance Model (AAM) and investigate a hierarchical “2D + time” ASM that integrates temporal constraints by using the third dimension for time instead of space.

Generative models such as Markov Random Fields (MRF) are popular in pixel labelling and de-noising problems [31]. Most image segmentation applications of MRFs also model the texture within a region and are constructed to favour spatially smooth regions. These “surface modelling” MRFs are often used to isolate homogeneous objects from their backgrounds. The left ventricle, however, contains papillary muscles (see Figure 1.4), rendering this approach less effective.

Lorenzo-Valdés et al. [32] implement a surface MRF for cardiac segmentation. From their reported examples it is clear that the papillary muscles are not included. As surface MRFs do not model the edge explicitly, they do not directly encode any shape information. There have been attempts to unify models of the edge and surface: specifically, Huang R. et al. [19] propose coupling surface MRFs with a hidden state representing a deformable contour.

Cordero-Grande et al. [11] construct an generative MRF model of the inner and outer contours. They use the Gibbs sampling algorithm to extract segmentations and parameters. Modelling the image likelihood, as is typical in MRFs can, however, lead to intractable models with complex dependencies between local features. This can lead to reduced performance if oversimplified [51]. Careful manual design of the probability distributions is therefore often necessary.

Also of interest is the approach by Jolly [23], in which the segmentation problem is set in log-polar space where the least cost path in a single frame (calculated by dynamic programming) is defined as the desired contour. A cost function, which is related to an initial labelling of blood pool pixels, is required to determine the correct contour. This bears a similarity to our approach: if we limit our model to a single frame (i.e. remove

temporal linkage) belief propagation reduces to an estimation of a least cost path similar to that produced by dynamic programming.

Heiberg [18] constructs filters for the detection of concordant (signed) and discordant (unsigned) edges for the inner and outer contours.

Previously, in Dreijer et al. [13] we investigated modelling the endocardium edge using a semi-conditional MRF with mostly heuristically chosen features. Although this approach showed promise, practical experiments indicated a strong attraction between the resulting contour and the epicardium. This is attributed partly to the epicardium's stronger edge with respect to surrounding tissue. In the present study, the outer contour is also explicitly included in the model, establishing a statistical relationship between the two contours. In addition we derive discriminative feature functions from human annotated images to improve performance. We also discuss this work in Dreijer et al. (2013) [14].

In summary, a number of methods are currently available, with the integration of temporal and spatial constraints and the presence of the papillary muscles still posing significant challenges [44].

1.4 Objectives of this study

Taking into account the challenges discussed above, the main goals of this dissertation are

1. developing a model for cardiac MRI segmentation which should
 - (a) integrate edge, shape and temporal information derived from an annotated dataset, and
 - (b) mitigate the effect of papillary muscles on the segmentations
2. developing a strategy to efficiently infer segmentations from this model
3. developing a strategy to tractably derive suitable estimates of the model parameters.

1.5 Contributions

To achieve these goals we make the following contributions:

1. present a novel formulation of the left ventricle MRI segmentation problem through
 - (a) a Conditional Random Field (CRF) model of the inner and outer contours
 - (b) local gradient features derived from a discriminatively trained edge classifier
 - (c) contextual feature functions derived from spatial and temporal properties of annotated segmentations
2. develop a process for the accurate segmentation of the left ventricle through
 - (a) offline tracking of the centre point
 - (b) a loopy belief propagation schedule for efficient inference of the inner and outer contours
3. suitable parameter estimation through
 - (a) minimisation of a segmentation error through the application of a gradient-free optimisation technique
 - (b) and thus avoiding the calculation of the intractable CRF partition function present in a maximum likelihood setting.

We have fully implemented a process that can be used to quickly segment video sequence and thereby replace the time intensive annotation work currently required from a human specialist. The technique is also able to mitigate the effects of papillary muscles close to the endocardium border by integrating information from temporally bordering frames. We show that the trained model also easily adapts to new datasets and outperforms many existing techniques.

1.6 Overview of this dissertation

The primary focus of this dissertation is the semi-automatic segmentation of the left ventricle into an inner and outer contour from a probabilistic model. An introductory background on the probabilistic approach is presented in Chapter 2.

In Chapter 3 we assume that the inner and outer contours form approximately circular shapes around a shared centre point in a single frame. This allows us to simplify the

problem by applying the log-polar transform of each image in the video sequence and then representing a contour at a specific time as a series of radii around the centre point. The non-linear nature of the log-polar space provides for a higher resolution for small contours since, after discretisation, a finer grid is used to represent the area near the centre point.

The centre point for each image is automatically derived from an operator provided centre point for the first image by searching for centre points that minimise a weighted inter-frame alignment error. This is done through dynamic programming.

Features based on discriminative edge properties, local shape information and local temporal behaviour are weighted and combined in a log-linear Conditional Random Field model. The edge properties are derived from a neural network (with two nodes in a hidden layer), trained on examples of inner and outer contour edges.

To infer segmentations when a new video sequence is presented, we employ loopy belief propagation (Chapter 4) to query this model. The order of message propagation and backtracking is chosen to mitigate the effects of its approximate nature.

Suitable CRF weights are estimated by considering the training problem as a minimisation of the difference between the resulting inferred segmentations and the ground truth (Chapter 5). This has the advantage of avoiding the intractable calculation of the partition function that would be needed in a maximum likelihood setting. Our approach explicitly focuses on minimising a specific error metric, instead of trying to model the “true” distribution.

This does, however, lead to a non-convex objective function and causes difficulty in calculating its derivatives. We select optimal parameters through the application of a gradient-free optimisation method.

The derivation of the feature functions and their weights is done against the York cardiac MRI dataset (Chapter 6). We find that the features derived from example edges are able to exploit the difference in edge gradients on the left and right sides of the cardiac structure. Analysis of the resulting segmentations in a cross validation scenario indicates that, for the majority of video sequences, segmentations are in line with expected behaviour with regard to shape, position and motion. Of particular note is the inclusion of the papillary muscles inside the inner contour and robustness with regard to noisy images.

Quantitative analysis indicates that the majority of errors occur during end-systole, es-

pecially at lower slices of the heart, if the endocardium disappears from view. This is to be expected, as the current model assumes that both the inner and outer contours are present in all images.

We analyse the sensitivity of the quality of segmentations to the choice of the initial centre point. We determine that segmentations remain relatively good, as long as this point is within 20% of the distance between the actual centre point and endocardium border.

We also evaluate the generalisability of our technique by applying it to the Sunnybrook cardiac MRI dataset, which is from a different source. Only the CRF parameters are re-trained, while re-using the feature functions (e.g. edge-information and inner-outer ratio behaviour) as derived from the York dataset. We find good segmentation results that are competitive or superior to the results as reported by other authors on this dataset.

Current limitations in our model (Chapter 7) are primarily due to the formulation of the segmentation problem. If the endocardium completely disappears from view during systole, the segmentation is unlikely to be correct since the possible absence of an inner contour is not taken into account. Additionally, using the same centre point for the inner and outer contours is problematic as, while the outer contour remains relatively static, the endocardium can undergo significant translation.

Chapter 2

Probabilistic models

In this chapter we provide an overview and motivation for the probabilistic approach used in the rest of this dissertation. Discriminative models, specifically the Conditional Random Fields, are briefly discussed. For a practical overview of these probabilistic models, see e.g. [5,25,27,31,37].

2.1 Probability distribution as a model

Consider the supervised-learning classification task where, during a training phase, examples of *input* values $\mathbf{x}^{(i)}$ together with their *targets* $\mathbf{y}^{(i)}$ are specified. Specifically, in our application $\mathbf{x}^{(i)}$ is a video sequence of images and $\mathbf{y}^{(i)}$ their human annotations. During inference a new value $\mathbf{x}^{(j)}$ is observed and we want the system to automatically assign a suitable target, $\mathbf{y}^{*(j)}$ — the optimal choice between all possible targets. The development of techniques that generalise to the mapping of an input observation to an output value

$$\mathbf{y} = f(\mathbf{x}) \tag{2.1}$$

is central to all sciences that attempt to construct models for use in prediction and optimal decision making. Consequently a large number of heuristic techniques exist in the various scientific fields.

The unification of many algorithms, both old and new, under a probabilistic Graphi-

cal Model interpretation [5, 22, 27, 43] has led to the introduction of efficient algorithms to various applications from automatic speech recognition to image segmentation.

These unified techniques based on probabilistic models have gained popularity within the machine learning community over recent years [25]. Under a Bayesian interpretation, probabilities represent a logically consistent approach to reason in the presence of uncertainty [22]. More specifically, under this interpretation, probability distributions are models of relative certainty that, once their properties are derived from observations and prior knowledge, can be queried (*inference*) for optimal decisions on the unobserved [22, 43].

Various conditional and marginal probabilities can be constructed from both prior domain knowledge and training examples if both the input x and target y are observed, e.g. $P(x)$, $P(y)$, $P(x, y)$, $P(x|y)$, $P(y|x)$ ¹. These probability distributions can then be queried to find the most reasonable target when given only an input (observation), $x^{(j)}$, i.e.

$$y^{*(j)} = \arg \max_y P(y|x^{(j)}). \quad (2.2)$$

Probability distributions are used in this dissertation to model the relationship between the video sequences and their segmentations. These are then used to infer the most likely segmentation when only a video sequence is observed, i.e. the segmentation with the maximum probability conditioned on the images. This is further discussed in Chapter 4.

2.2 Generative and discriminative models

In a generative model the probability of the observations x conditioned on the latent variable y is explicitly modelled, i.e. $P(x|y)$. This allows one to generate instances of the observations x given an instance y .

In supervised learning, when the best configuration $y^{*(j)}$ given a new observation $x^{(j)}$ is sought, the configuration that maximises the posterior distribution is required, i.e. $y^{*(j)} = \arg \max_y P(y|x^{(j)})$. The primary strategy in applying these generative models to classification problems is to build a complete probabilistic model for each class, $P(x|y)$, and

¹We use the capital P to refer to both discrete and continuous probability distributions. It should be clear from the context which is used.

derive the posterior $P(y|x)$ through Bayes' theorem,

$$P(y|x) = \frac{P(x|y) P(y)}{P(x)}. \quad (2.3)$$

Much research has been conducted into suitable choices for the prior distribution $P(y)$. If only the maximising configuration is sought (as in (2.2)) and not its probability then $P(x)$, the marginal likelihood of the data, can be ignored as it does not influence the selection of the optimal configuration.

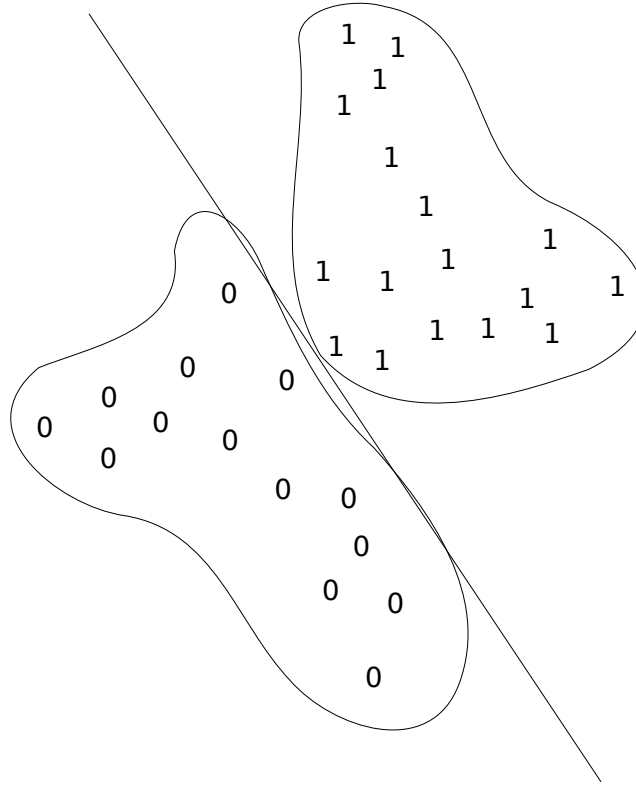


Figure 2.1: Example of two linearly separable classes with complex distributions.

As a simple example to demonstrate the generative approach, consider the simulated two-class problem in Figure 2.1. We wish to construct a classifier that assigns new observation to an appropriate class. In a generative approach we fit two distributions to the training data to represent the two classes, i.e. $P(x|y = 0)$ and $P(x|y = 1)$. For a new obser-

vation $\mathbf{x}^{(j)}$, the most likely class is then derived from Bayes' theorem,

$$y^{*(j)} = \arg \max_{y \in \{0,1\}} P(y|\mathbf{x}^{(j)}) \quad (2.4)$$

$$= \arg \max_{y \in \{0,1\}} \frac{P(\mathbf{x}^{(j)}|y) P(y)}{P(\mathbf{x}^{(j)})} \quad (2.5)$$

$$= \arg \max_{y \in \{0,1\}} P(\mathbf{x}^{(j)}|y) P(y). \quad (2.6)$$

The prior $P(y)$ is, for example, the fraction of examples from each class present in the training set. The more challenging aspect is the modelling of the class-specific distributions $P(\mathbf{x}|y=0)$ and $P(\mathbf{x}|y=1)$. Note from the illustration in Figure 2.1 that the data from the two classes are unlikely to be well represented by a simple Gaussian distributions. Many of the observations can be better represented by more flexible distributions, requiring more parameters.

The primary problem with this approach is that the available parameters (freedom in the model) is used to build two complete models of the classes. This implies that we also attempt to model data that are far from the separating boundaries which are unlikely to be confused during classification. If we are only interested in classification, then a more useful endeavour might be to put all the effort into attempting to model the separating boundary instead, i.e. use all available parameters to model $P(y|\mathbf{x})$ directly. This is known as the discriminative approach and has the advantage that modelling the potentially complex likelihood of \mathbf{x} is unnecessary.

There are various trade-offs between generative and discriminative models and their performances are often application dependent. See e.g. [27] for a discussion. Generative models, for example, are generally easier to adjust to changes in the individual classes and are easier to extend to unsupervised learning scenarios. Generative models, however, also generally assume independence between subsets of \mathbf{x} to improve tractability in higher dimensions. This simplification can be detrimental to classification accuracy if sufficiently complex dependencies exist between the observed variables [51].

We use a discriminative model in this dissertation, specifically the Conditional Random Field, which is discussed in the next section.

2.3 Conditional random fields

A Conditional Random Field [29, 53] models the conditional probability of a set of unobserved/latent variables \mathbf{y} given a set of observations \mathbf{x} , i.e. $P(\mathbf{y}|\mathbf{x})$. This probability is represented as a product of local potential functions,

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{q \in \mathcal{Q}} \psi_q(\mathbf{y}_q, \mathbf{x}), \quad (2.7)$$

where the normalisation, $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{q \in \mathcal{Q}} \psi_q(\mathbf{y}_q, \mathbf{x})$, sums over all possible configurations of \mathbf{y} . Notice that the CRF is discriminative in nature.

The potential functions $\psi_q(\mathbf{y}_q, \mathbf{x})$ encode relationships between subsets, or cliques $q \in \mathcal{Q}$, of the latent and observed variables. As such, it not only models the affinity between each latent variable and the observed variables, but also the contextual relationships between the latent variables. This is useful where consistency between assignments to the latent variables is important, i.e. where they are not statistically independent.

The selection of these potential functions is part of the design and training processes. Generally, the potential functions are non-negative and large when dependent variables are “compatible” (their values are likely to occur together) and small when incompatible. The un-normalised product of the potentials is thus large when evaluated on instances from the modelled distribution and small otherwise.

The partition function $Z(\mathbf{x})$ normalises the product of potentials so that the probabilities sum to one. The summation over all possible configurations of the latent variables, \mathbf{y} , is combinatorial in the number of latent variables and thus not computationally tractable in general. This has important implications for training and is discussed further in Chapter 5.1. When inferring an optimal configuration \mathbf{y}^* for the latent variables (i.e. one that maximises (2.7)) the partition function can, however, be ignored as discussed in Chapter 4.

The CRF is different from the Markov random field (MRF) which generatively models the probability distribution of the observed variables, given different configuration of the latent variables $P(\mathbf{x}|\mathbf{y})$ or their joint distribution $P(\mathbf{x}, \mathbf{y})$. The main advantage of a CRF is its discriminative nature, i.e. it does not require a detailed model of the observed information. Instead, computational resources are allocated to modelling the behaviour of the

latent variables, as explained above.

Indeed, note in (2.7) that while each potential function is defined over a subset of the latent variables, they are also defined over the joint of all the observed variables. This is practical in a CRF since modelling of the complex dependencies between the observations is not needed, which allows the use of discriminative models of the potentials. In MRFs these complex dependencies are modelled with generative models, which, for computational reasons, are often simplified, for example, by assuming independence between subsets of the observed variables.

Popular applications of CRFs include natural language processing and computer vision to segment and label images. Their popularity is partly due to their ability to manage the complex relationships between the observed variables that are present in these domains.

2.4 Efficient factorisation

The presence of many variables can make operations on probabilistic models computationally expensive. During inference, e.g., in (2.2), maximisation is done over all possible configurations of \mathbf{y} , which is intractable in general. A generalised distributive law [1], however, applies to various mathematical operations, e.g. the summation of products, the maximisation of products and the minimisation of sums. It is therefore often possible to rework these expressions into a manageable form through factorisation. Belief propagation, a specific inference algorithm, is further discussed in Chapter 4 and is based on this idea.

Factor graphs [28] are useful in the design and illustration of these algorithms, which are efficient and exact for simple graphs and, upon further extension, approximate in the presence of complex interdependencies between variables where many loops are present.

Factor graphs, e.g. Figure 2.2, are bipartite graphs with circular nodes representing variables and square nodes representing functions that relate the connected variables. Observed variables are often shaded and latent variables unshaded. Factor graphs illustrate the factorisation of a function into sub-functions that are dependent on subsets of the variables.

In the case of the factorisation in (2.7), the potentials ψ_q are the sub-functions that locally relate the variables. The un-normalised joint probability of a configuration, i.e. (2.7)

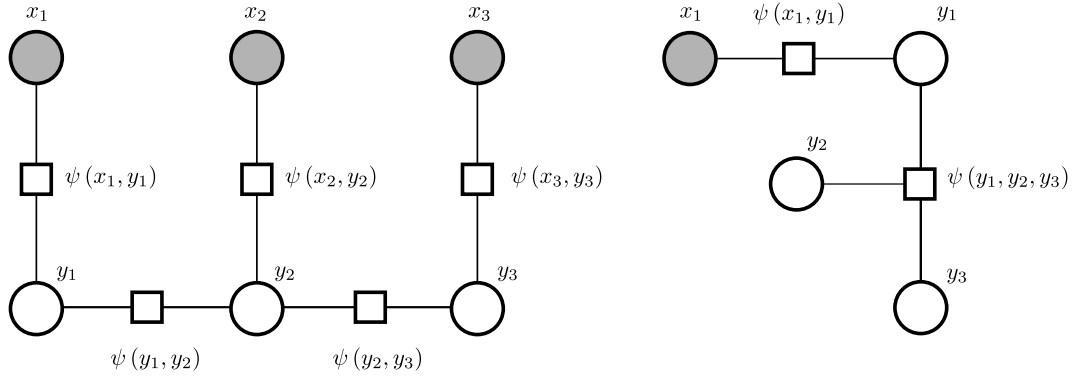


Figure 2.2: Two example factor graphs.

without $Z(\mathbf{x})$, is thus simply the product of the potential functions. The illustrations in Figure 2.2 represent the example factorisations $\psi(x_1, y_1) \psi(x_2, y_2) \psi(x_3, y_3) \psi(y_1, y_2) \psi(y_2, y_3)$ and $\psi(x_1, y_1) \psi(y_1, y_2, y_3)$.

The details as applied to the problem of this dissertation, are discussed in Chapter 4.

Chapter 3

Model description

In this chapter we discuss the segmentation of a video sequence of cardiac images as a collection of radii around a series of centre points. We also describe a semi-automated technique to determine centre points for all frames in the sequence from knowledge of the centre point in the first frame. Features describing local edge information, spatial and temporal behaviour and the ratio between the inner and outer contours, are derived from a training dataset. These features are weighted by parameters and combined in a Conditional Random Field.

3.1 Problem formulation

We refer to the left ventricle endocardium as the inner contour and the left ventricle epicardium together with the right ventricle's endocardium bordering the septum as the outer contour, as is indicated in Figure 1.4.

Our primary interest is in automatically segmenting a video sequence of T greyscale images $I(0), \dots, I(T-1)$ that is synchronised to a single cardiac cycle so that the first image $I(0)$ is before systole (contraction) and the last image $I(T-1)$ after diastole (relaxation). End-systole (maximum contraction) thus occurs in the middle of the sequence at approximately $I(T/2)$. We refer to the greyscale value of a pixel within a single image at time t and at the image-coordinate \mathbf{p} , as $I(t, \mathbf{p})$.

Assume an approximately circular shape for both the inner and outer contours of the

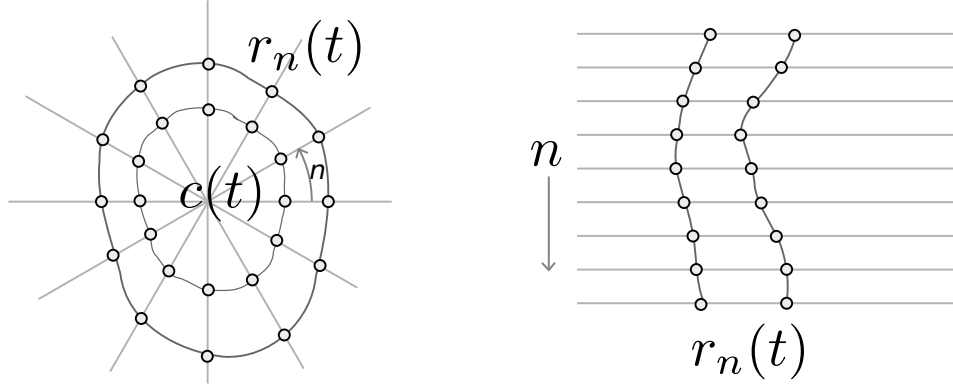


Figure 3.1: A representation of coordinates on the inner and outer contours in Cartesian and polar coordinates.

left ventricle. A closed contour in a frame at time t can then be represented by a series of N positive real values, $r_0(t) \dots r_{N-1}(t)$, at uniformly spaced angles, $\phi_n = n \frac{2\pi}{N}$, with respect to a centre point $c(t)$. We use a single shared centre point for the inner and outer contours in a frame. Figure 3.1 illustrates coordinates on the inner and outer contours using a small number of radii. In our implementation we use $N = 128$ angular directions.

For the implementation of the optimisation described in Chapter 4 it is convenient to work in the discretised log-space of the radii,

$$\rho_n(t) = \lfloor M \cdot r_{\text{init}} \cdot \log r_n(t) \rfloor. \quad (3.1)$$

Here $\lfloor x \rfloor$ is the floor function of x and $r_{\text{init}} = 50$ is experimentally chosen such that, for most segmentations in the training set, $\rho_n(t) \approx M/2$ at end-diastole. The log-radius is discretised as $\rho_n \in \{0, \dots, M-1\}$ where M is the resolution of the segmentation. In this study $M = 256$ has proven to be adequate. Due to its non-linear nature, one advantage of the log-space is that it provides a better spatial resolution at smaller radii, which is useful as the cardiac contours occupy a relatively small area of the images. Discretisation is necessary for the optimisation strategy as is discussed in Chapter 4.

Figure 3.2 illustrates the log-polar transformed image $D(t)$ of a single image $I(t)$. We denote the image values in the n th radial direction (i.e. the n th row in the log-polar image) as the vector $d_n(t)$. The greyscale value of a pixel in the log-polar space is then referred to as $d_n(t, \rho)$.

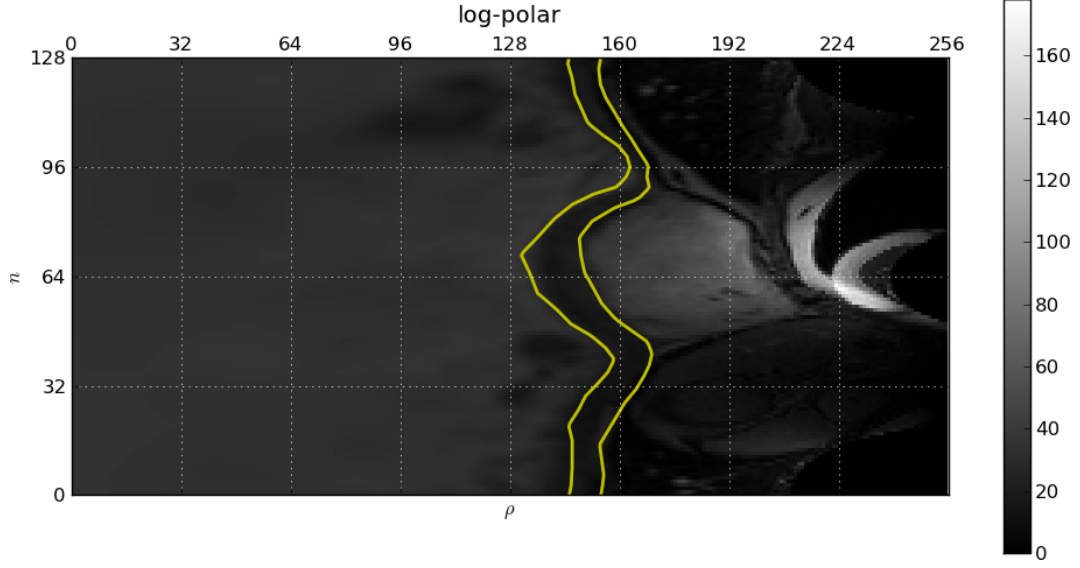


Figure 3.2: Log-polar transform with human annotated (ground truth) inner and outer contours in yellow. The centre point is chosen near the middle of the endocardium.

The inner and outer segmentation contours in a single frame are thus fully represented as two vectors of log-radii,

$$\boldsymbol{\rho}^{\text{in}}(t) = \{\rho_0^{\text{in}}(t), \dots, \rho_{N-1}^{\text{in}}(t)\} \quad (3.2)$$

$$\boldsymbol{\rho}^{\text{out}}(t) = \{\rho_0^{\text{out}}(t), \dots, \rho_{N-1}^{\text{out}}(t)\}, \quad (3.3)$$

relative to the centre point $c(t)$. The segmentation of a video sequence of frames is represented by a sequence of these inner and outer radii, $\boldsymbol{\rho} = \{\boldsymbol{\rho}^{\text{in}}, \boldsymbol{\rho}^{\text{out}}\}$, around a sequence of centre points $\boldsymbol{c} = \{c(t)\}_{t=0, \dots, T-1}$, where

$$\boldsymbol{\rho}^{\text{in}} = \{\boldsymbol{\rho}^{\text{in}}(0), \dots, \boldsymbol{\rho}^{\text{in}}(T-1)\} \quad (3.4)$$

$$\boldsymbol{\rho}^{\text{out}} = \{\boldsymbol{\rho}^{\text{out}}(0), \dots, \boldsymbol{\rho}^{\text{out}}(T-1)\}. \quad (3.5)$$

The position of the centre point is allowed to vary between frames. This is necessary since the heart can change position during contraction and expansion.

Our segmentation process first determines a sequence of centre points, after which relative radial values are inferred. In the next section we turn our attention to finding suitable centre points.

3.2 Centre point estimation

Our model of the radial values requires that the centre points be positioned inside the left ventricle near the middle of the endocardium. This allows the segmentation process to restrict the spatial and temporal variability of the radii. In this section we describe a semi-automated procedure to determine suitable centre points for successive frames after a single centre point in the initial frame is provided by the user. This removes the need to manually specify the centre point for all frames in a video sequence.

A number of heuristic techniques (e.g. [44,46]) are available for estimating centre points. Most of these perform adequately when the papillary muscles are absent and there is high contrast between the blood pool and the cardiac wall. This is generally the case for the first few frames, but not at the end of systole (at approximately $T/2$) when the left ventricle's blood pool is at its smallest and sometimes disappears from view.

Online tracking techniques, such as those based on the Kalman filter [26], rely only on information from previous frames to correctly identify the tracked object's position in the current frame. This can become problematic when the object becomes very small or even disappears from view, degrading information used in tracking in later frames. If information from all frames are available for tracking, i.e. in an offline setting, then information from later frames can be used to mitigate these difficulties.

The procedure described below requires the user to provide the centre point $c(0)$ of the first frame, when the endocardium is clearly visible and the papillary muscles minimally obstruct the inner contour. Centre points for the remaining frames are then automatically derived through offline tracking. The procedure is therefore best described as being semi-automated. The robustness of this semi-automated procedure against variations in the selection of $c(0)$ is discussed in Section 6.1.3. In short, the system is not particularly sensitive to the choice of $c(0)$ if placed near the centre of the endocardium.

Each video sequence from our dataset contains a single heart beat (cardiac cycle). Due to its periodic nature, we therefore assume that the last frame has the same centre point as the first, i.e. $c(T-1) = c(0)$. This is not a severe restriction since the algorithm is robust against variations in $c(t)$, as demonstrated in Section 6.1.3.

To find centre points in the intermediate frames $c(1), \dots, c(T-2)$ we minimise a

weighted inter-frame alignment error,

$$\text{error}(\mathbf{c}) = \sum_{t=1}^{T-1} \sum_{\mathbf{p}} w^{\mathbf{c}(t)}(\mathbf{p}) \cdot \left(I^{\mathbf{c}(t)}(t, \mathbf{p}) - I^{\mathbf{c}(t-1)}(t-1, \mathbf{p}) \right)^2, \quad (3.6)$$

where $I^{\mathbf{c}(t)}(t)$ is the image $I(t)$ centred at $\mathbf{c}(t)$ such that $I^{\mathbf{c}(t)}(t, \mathbf{p}) = I(t, \mathbf{p} - \mathbf{c}(t))$ and zero when indexed out of bounds. Recall that \mathbf{p} is an (x,y)-coordinate pair and $I(t, \mathbf{p})$ is a pixel in the frame at time t before a log-polar transform is applied.

The weight $w^{\mathbf{c}(t)}(\mathbf{p}) = e^{(-\|\mathbf{c}(t) - \mathbf{p}\|^2 / \sigma^2)}$ locally enhances the error around the current frame's centre point. The width σ is experimentally chosen from a training dataset so that the inferred centre points closely resemble the mean of the ground truth inner contours. For computational efficiency we only calculate values within a 2σ radius of $\mathbf{c}(t)$ as the contribution becomes negligible further away.

The error in (3.6) is a non-linear function of the sequence of centre points but is efficiently solved using an optimisation strategy such as dynamic programming. In addition, in order to reduce computational effort we assume that the centre point translates less than three pixels between successive frames. Backtracking is initiated at the centre of the last frame $\mathbf{c}(T-1)$. In order to constrain $\mathbf{c}(0)$ to the value provided by the user, during dynamic programming, the cumulative cost function for the first frame is set to zero at $\mathbf{c}(0)$ and one otherwise.

Since the calculation of the centre points is performed before and separately from the calculation of the radial values, a faster heuristic method could be considered for the centre points. However, we prefer to pose the centre point estimation problem as one that can be solved with dynamic programming, and therefore within the same belief propagation framework as used to infer the radial values as described in Chapter 4. This allows both the centre point estimation and radial inference to be implemented using the same general belief propagation software infrastructure. If required, the centre point estimates can easily be refined after inference of the radial values. This, however, has not proven to be necessary and is not implemented in this dissertation — the centre points are fixed after applying the process described in this section.

3.3 The CRF model

In this chapter we represent a temporal sequence of log-polar images as $D = \{D(t)\}_{t=0,\dots,T-1}$ and, by assuming appropriately trained parameters, θ , we model the conditional probability $P(\rho|\theta, D)$ of a segmentation ρ . This is done through a log-linear CRF,

$$P(\rho|\theta, D) = \frac{1}{Z(\theta, D)} \exp(-E(\rho|\theta, D)). \quad (3.7)$$

The energy $E(\rho|\theta, D)$ is defined as the weighted sum of local feature functions f , defined over all cliques $q \in \mathcal{Q}$ in its graphical model,

$$E(\rho|\theta, D) = \sum_{q \in \mathcal{Q}} \theta_q f_q(\rho_q, D). \quad (3.8)$$

In this formulation, smaller energies indicate better segmentations, while bad segmentations are penalised with larger energies. The conditional notation of the energy function is not strictly necessary, but emphasises that its associated probability and composing feature functions are discriminative in nature.

The partition function

$$Z(\theta, D) = \sum_{\rho} \exp(-E(\rho|\theta, D)) \quad (3.9)$$

sums over all possible configurations of ρ , normalising the exponentiated energy into a probability. Its calculation is intractable in general, but can be avoided during inference, as discussed in Chapter 4.

There are no significant theoretical restrictions to the feature functions in a Random Field (RF). Restrictions and the choices of feature functions are design choices aimed at improving performance.

3.4 Feature functions

We restrict ourselves to positive feature functions f where small values are more desirable (i.e. smaller values indicate “better” segmentations). Positive parameters, θ , determine the

relative weights of the feature functions. The number of computations involved in calculating these feature functions grow exponentially with the number of dependent variables. In order to minimise the computational cost, we therefore restrict ourselves to features that couple at most two radial values in space or time.

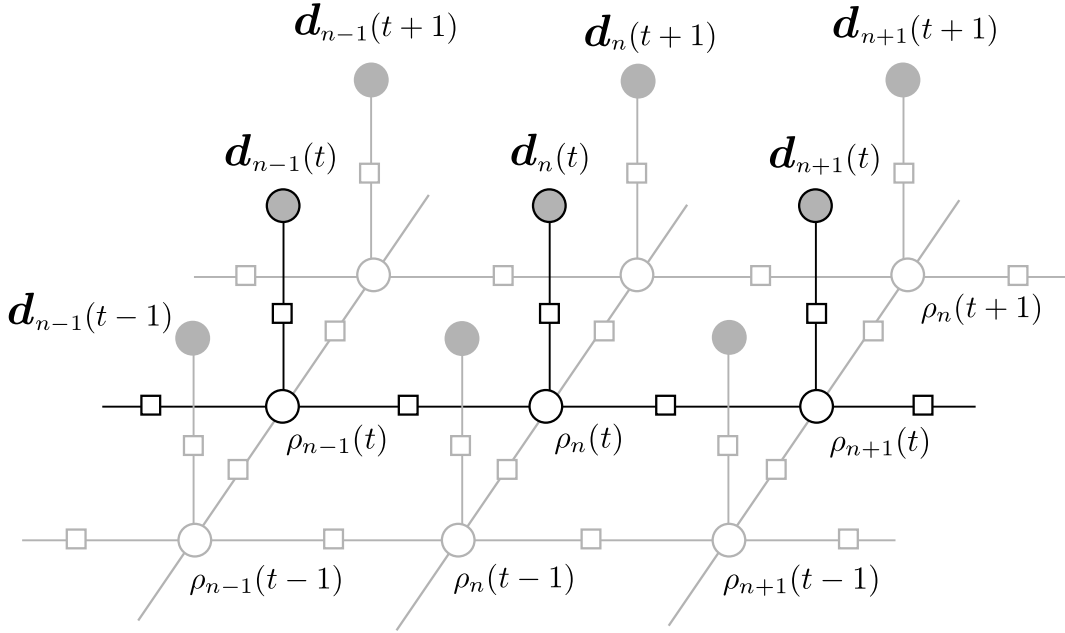


Figure 3.3: A partial factor graph of the temporal and spatial relationships between radius variables, ρ , in a single contour and rows in the log-polar transformed image, $d_n(t)$. Factor labels and some variable labels are omitted for clarity.

The partial factor graph in Figure 3.3 illustrates the temporal and spatial relationships between radius variables, ρ , in a single contour and rows in the log-polar transformed image, $d_n(t)$. The spatially circular nature (i.e. there is a feature (or factor) connecting $\rho_0(t)$ and $\rho_{N-1}(t)$) and the relationships between the inner and outer contours are omitted for clarity.

We derive the feature functions from discriminative properties of human annotated images as described below. Selection of the relative weights, θ , is discussed in Chapter 5.

3.4.1 Feature function based on edge classifiers

Consider a log-polar frame (such as in Figure 3.2 or Figure 3.4) and note that the inner and outer contours tend to occur at positions that contain an edge. Although there exist a number of edge detection algorithms, the selection of an appropriate technique is compli-

cated by different edge behaviour at different parts of the contours and the presence of the papillary muscles that can obscure the edge.

Since we are interested in specific types of edges, and annotated edges are available, it is possible to custom-design an edge detector for this specific application. We therefore train a simple artificial neural network, with two nodes in a hidden layer, to model the presence of the cardiac edges. A window extracted from the training set is considered to contain an edge if the centre of the window is no more than 2 log-radial units away from the ground truth contour, otherwise it is considered a “non-edge” training example.

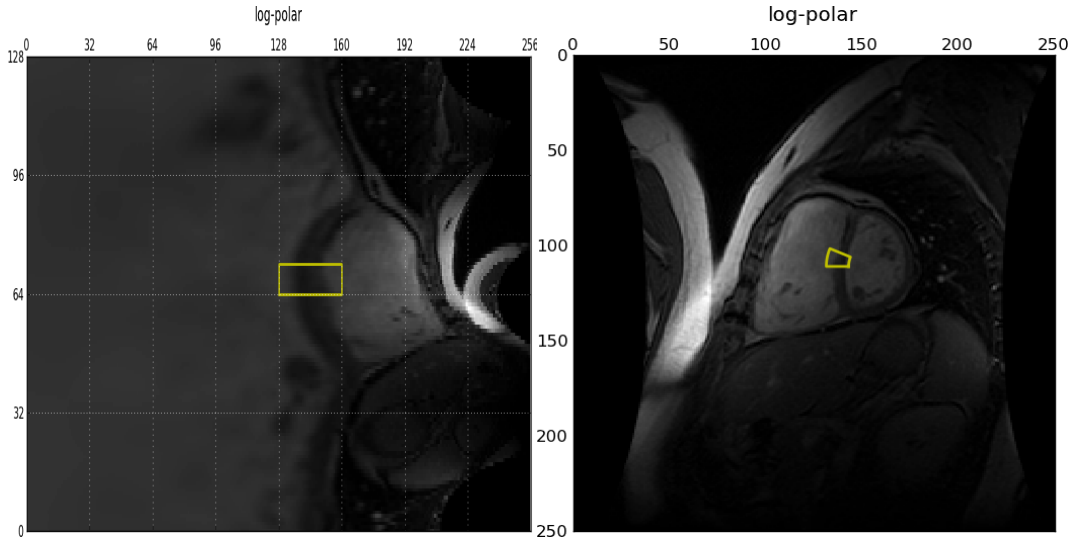


Figure 3.4: Window extracted in log-polar space and corresponding circular sector in untransformed image. The height of the window in these figures are set to eight angular units for illustrative purposes.

More specifically, for a log-polar frame at time t , consider a window (e.g. in Figure 3.4) of height 1 and width $w = \frac{M}{4} = 64$ centred at ρ and in the radial direction, n . This window is equivalent to a circular sector¹ in the original image before the log-polar transform is applied. We refer to the pixel values in this window as the vector $\mathbf{v}_\rho = \mathbf{d}_n(t, [\rho - \frac{w}{2}, \dots, \rho + \frac{w}{2}])$. A feature vector $\kappa(\mathbf{v})$ is derived from the window and is described below.

The feature vector $\kappa(\mathbf{v})$ consists of the concatenation of four expressions of the gradient in the radial direction, $\frac{\partial \mathbf{v}}{\partial \rho}$, to discriminate between edges and non-edges,

$$\kappa(\mathbf{v}) = \left(\frac{\partial \mathbf{v}}{\partial \rho}, \left| \frac{\partial \mathbf{v}}{\partial \rho} \right|, \text{sign} \left(\frac{\partial \mathbf{v}}{\partial \rho} \right), \left[\left| \frac{\partial \mathbf{v}}{\partial \rho} \right| > \epsilon \right] \right). \quad (3.10)$$

¹Strictly speaking, this window is equivalent to the intersection of a “track” and a sector.

The expression $\left[\left| \frac{\partial v}{\partial \rho} \right| > \epsilon \right]$ is a binary value indicating the presence of a gradient, i.e. equal to 1 if the edge magnitude is larger than ϵ and 0 otherwise.

From the short-axis view in Figures 1.4 and 3.4, it is apparent that the gradient sign on the left and right sides of the outer contour's edge differ, due to the intensity of the right ventricle's blood pool. We therefore train eight classifiers for different parts of the contour, i.e. instead of training a single classifier over all angular directions ($n = 0..N - 1$), groups of angles are treated separately ($n = 0, \dots, 15$; $n = 16, \dots, 31$ etc.) and thus leading to direction dependent classifiers. This allows the classifiers to exploit features that it might find relevant in that direction. This is an important advantage over standard multi-purpose edge detectors that, without customisation, would handle gradients in different parts of the images, similarly.

The process is repeated for the classifiers of the inner contour's edges since the endocardium edge on the left and right sides also behave differently, due to the presence of the papillary muscles mostly on one side (see Figure 1.5).

Heiberg [18] identifies the edges of the inner and outer contours as two classes: Concor-dant, where edge gradients have a similar sign, and Discordant, areas where the edge gradient signs differ. Accordingly, he explicitly includes the gradient sign for the inner contour and ignores the sign for the outer. Because our edge model is trained on annotated ground truth images, our classifier automatically differentiates between the signs when they are relevant to edge detection. The nature of the edges are further explored in Appendix A.

To fit within the framework of energy minimisation the response of the neural network, $g_0(\rho, n)$, to an image window centred at (ρ, n) , is transformed from an indicator/detection function into a cost function by subtracting its output value from one, i.e. $g_1(\rho, n) = (1 - g_0(\rho, n))$. The minimum cost in the radial direction is subtracted to avoid negative feature values, $g_2(\rho, n) = g_1(\rho, n) - \min_{\rho'} g_1(\rho', n)$, and is normalised by the sum, $g_3(\rho, n) = \frac{g_2(\rho, n)}{\sum_{\rho'} g_2(\rho', n)}$.

It should be noted that the typical neural network training techniques attempt to minimise a classification error. Recall, however, that we are not so much interested in the classification of edges than we are in using the output of the feature function to penalise non-contour positions, while at the same time penalising contour positions as little as possible. Ideally, we prefer a classifier that rejects very few possible edges, i.e. results in very

few false negatives.

To mitigate this unintended penalisation, the outputs of the constructed features are thus “smoothed” by applying a small minimum-filter (erosion) in the angular direction,

$$f(\rho_n) = \min_{n'=n-1, n, n+1} g_3(\rho, n'). \quad (3.11)$$

This will cause an area to have a high valued feature function response (i.e. be penalised) only if its neighbouring areas (in the angular direction) also have high feature responses.

We construct these networks for the inner and outer contour edges and thereby obtain the features $f^{\text{in}}(\rho_n^{\text{in}}(t), \mathbf{d}_n(t))$ and $f^{\text{out}}(\rho_n^{\text{out}}(t), \mathbf{d}_n(t))$.

Figures 3.5a and 3.5b, respectively, show the resulting inner and outer feature responses to the frame in Figure 3.2. The response is relatively low in the area of the ground truth, i.e. more desirable segmentations generally have smaller feature values.

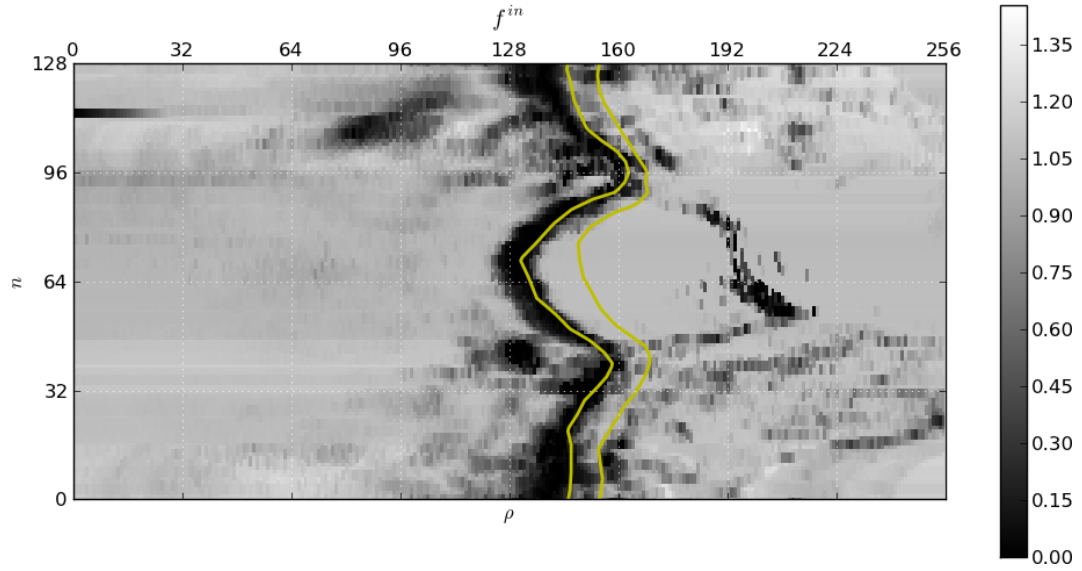
However, there are strong gradients at some non-contour positions and relatively weak gradients at some contour positions, especially when papillary muscles are close to the endocardium border. The resulting feature responses, on their own, are therefore unable to adequately provide a good indication for the position of the inner and outer contours. It should be emphasised that these response images have been obtained without taking into account any temporal behaviour or continuity requirements. We now investigate how to incorporate these properties into our model.

3.4.2 Spatial and temporal feature functions

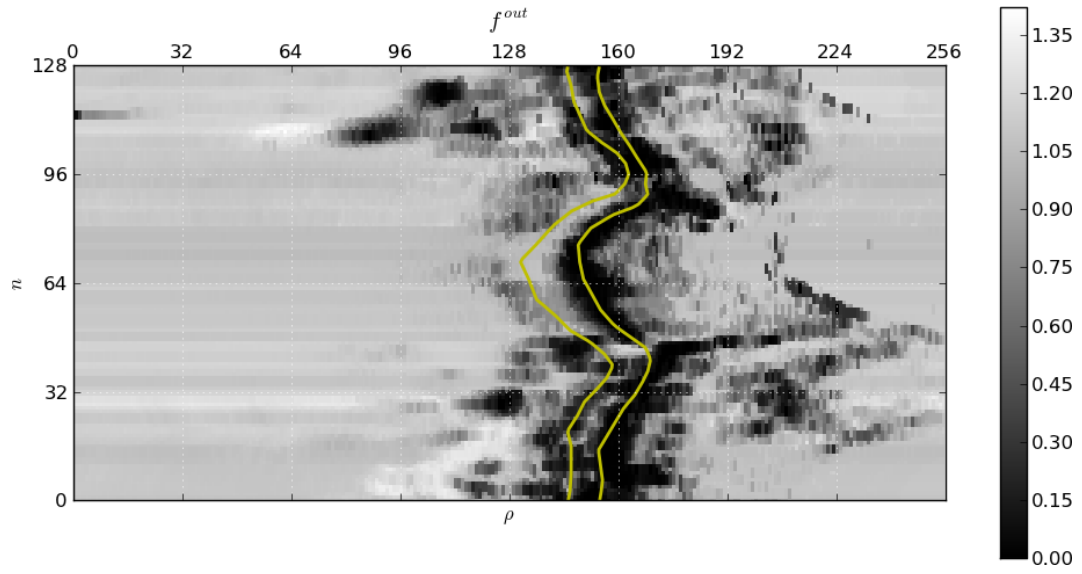
We now proceed to introduce spatial continuity, as well as temporal information directly into the model with the following feature functions,

$$f_r(\rho_n(t), \rho_{n-1}(t)) = \left(\frac{\rho_n(t) - \rho_{n-1}(t)}{M} \right)^2, \quad (3.12)$$

$$f_t(\rho_n(t), \rho_n(t-1)) = \left(\frac{\rho_n(t) - \rho_n(t-1)}{M} \right)^2, \quad (3.13)$$



(a) Inner edge feature function response



(b) Outer edge feature function response

Figure 3.5: Inner and outer feature function responses to the image in Figure 3.2. The ground truth contours are indicated with yellow lines.

were M is again the discretisation value, used here to scale the feature values to the same order of magnitude as the features described previously². Moreover, since direct inspection of training data indicates that few contours violate the properties $|\rho_n(t) - \rho_n(t-1)| \leq 24$ and $|\rho_n(t) - \rho_{n-1}(t)| \leq 8$, the search space can be significantly reduced by ignoring radial pair values that do not conform. This is discussed in more detail in Section 4.1.

Contour growth during systole and shrinkage during diastole is penalised by assuming that end-systole (maximum contraction) is reached at time t_{ES} and applying the feature

$$f'_t(\rho_n(t), \rho_n(t-1)) = \begin{cases} [\rho_n(t-1) < \rho_n(t)] & \text{if } t < t_{ES} \\ [\rho_n(t) < \rho_n(t-1)] & \text{otherwise.} \end{cases} \quad (3.14)$$

For simplicity, we have chosen a fixed $t_{ES} = 8$ from inspection of the training annotations. The correct selection of t_{ES} is sensitive to patient pathology. Enforcing a shrinkage followed by a growth in radius values without choosing a fixed t_{ES} can be done with a second order temporal model, but would add to the computational complexity. Alternatively, detecting when the optical flow in the images are suspended and reversed, might provide enough information to choose a better end-systole — a topic that might be fruitfully investigated in a future study.

Again, these temporal features are constructed separately for both the inner and outer contours.

Additionally, we assume that the “angular gradient” of the cardiac wall just inside the outer contour remains small over time and space through the feature functions,

$$f''_t(\rho_n^{\text{out}}(t), \rho_n^{\text{out}}(t-1)) = |\mathbf{d}_n(t, \rho_n^{\text{out}}(t) - \epsilon_\rho) - \mathbf{d}_n(t-1, \rho_n^{\text{out}}(t-1) - \epsilon_\rho)| \quad (3.15)$$

and

$$f''_r(\rho_n^{\text{out}}(t), \rho_{n-1}^{\text{out}}(t)) = |\mathbf{d}_n(t, \rho_n^{\text{out}}(t) - \epsilon_\rho) - \mathbf{d}_n(t, \rho_{n-1}^{\text{out}}(t) - \epsilon_\rho)| \quad (3.16)$$

where $\epsilon_\rho = 2$ is an experimentally chosen radial offset. Similar feature functions are constructed for the intensity just outside the inner contour.

²Strictly speaking, scaling is not necessary since unscaled values can be compensated for by the corresponding parameters. In practice, however, it is beneficial to scale the values beforehand e.g. if regularisation is required.

3.4.3 Inner-outer radius feature functions

Information on the cardiac structure can further be exploited by using the relationship between the inner and outer contours. Specifically, the ratio between the inner and outer radii, $r_n^{\text{in}}(t)/r_n^{\text{out}}(t)$, and thus the difference in log-space, $|\rho_n^{\text{out}}(t) - \rho_n^{\text{in}}(t)|$, is found to contain information on the temporal behaviour. This can be seen in Figure 3.6. This ratio is related to the wall thickness but is invariant to scaling, which can occur due to differences in patient physiology and MRI magnifications.

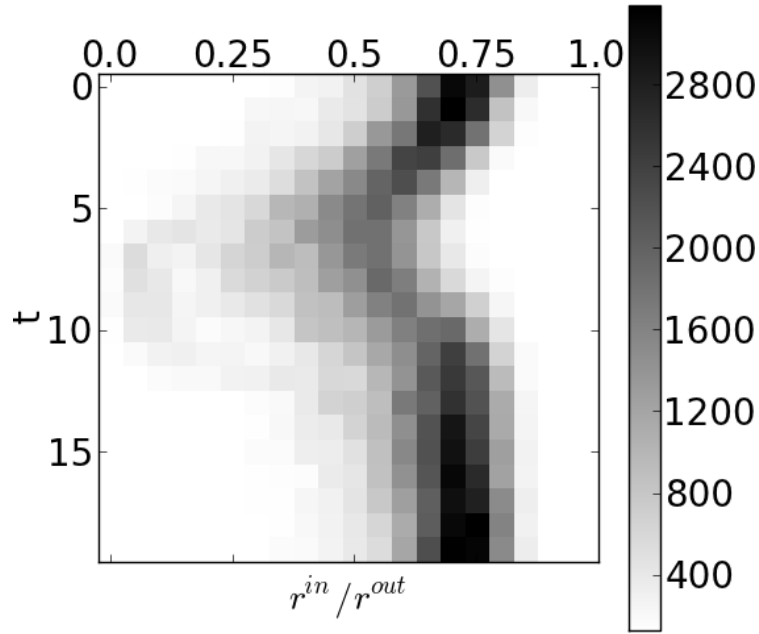


Figure 3.6: Histogram of the ratio between the inner and outer radii, $r_n^{\text{in}}(t)/r_n^{\text{out}}(t)$, against time, generated from a training dataset.

A probability distribution of the log-radial distance between the inner and outer contours, $P(\rho_n^{\text{out}}(t) - \rho_n^{\text{in}}(t))$, is derived from annotations in a training dataset and used to construct the feature function (see Figure 3.7),

$$f_1^{\text{cross}}(\rho_n^{\text{in}}(t), \rho_n^{\text{out}}(t)) = -\log P(\rho_n^{\text{out}}(t) - \rho_n^{\text{in}}(t)). \quad (3.17)$$

The relative homogeneity of the cardiac wall can also be exploited by minimising the

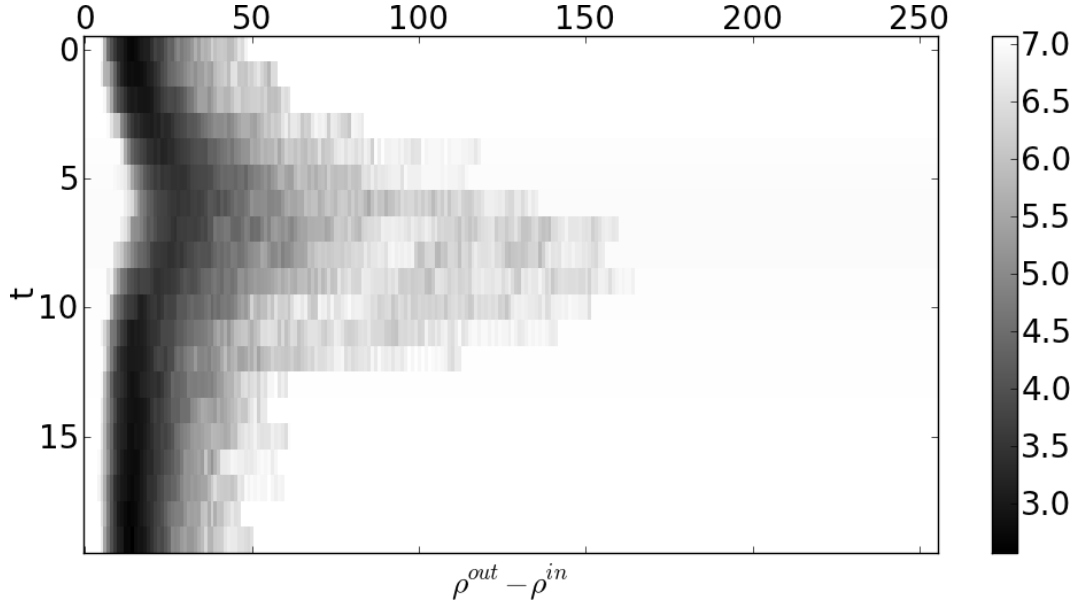


Figure 3.7: Response of feature, f_1^{cross} : difference between inner and outer log radii, $|\rho_n^{\text{out}}(t) - \rho_n^{\text{in}}(t)|$ against time.

variance in intensity of the area between the inner and outer contours with

$$f_2^{\text{cross}}(\rho_n^{\text{in}}(t), \rho_n^{\text{out}}(t), d_n(t)) = \frac{1}{W_n} \sum_{\rho=\rho_n^{\text{in}}(t)}^{\rho_n^{\text{out}}(t)} (d_n(t, \rho) - \mu_n)^2. \quad (3.18)$$

Here the mean wall colour μ_n is given by

$$\mu_n = \frac{1}{W_n} \sum_{\rho=\rho_n^{\text{in}}(t)}^{\rho_n^{\text{out}}(t)} d_n(t, \rho), \quad (3.19)$$

where the wall width is

$$W_n = \rho_n^{\text{out}}(t) - \rho_n^{\text{in}}(t). \quad (3.20)$$

From inspections of the training dataset, we enforce a minimum cardiac wall thickness of 10 log-radial units, i.e. $\rho_n^{\text{in}}(t) + 10 \leq \rho_n^{\text{out}}(t)$. This is used during belief propagation in Chapter 4 to enforce a realistic minimum cardiac wall thickness and to reduce the number of calculations required.

3.5 Summary

We derive discriminative features from a training dataset, describing local image properties of training annotations. We also extract spatio-temporal features from annotations. To model the conditional probability of segmentations these feature functions are weighted by parameters and combined in a CRF as in (3.8). The selection of these parameters is discussed in Chapter 5.

Of interest is the generative MRF model designed by Cordero-Grande et al. [11] which relies on modelling the generative likelihood of an image conditioned on a segmentation, i.e. $P(\text{Image}|\text{Segmentation})$. To construct a model of the images that is mathematically tractable, the modelled distribution of the images is simplified by considering reduced image properties, such as a homogeneous myocardium, and through strict assumptions of conditional independence and distribution types. This can be a challenging process since these simplifications could potentially remove image properties that are important to segmentation quality.

In contrast, our approach directly models the conditional probability of the segmentations given an image, $P(\text{Segmentation}|\text{Image})$ and is thus discriminative in nature. A model of the images is therefore unnecessary since we relate the variables representing the observed image and segmentations with the discriminative edge detector discussed above.

Chapter 4

Segmentation

We now turn our attention to inferring a segmentation when given a new video sequence and suitably trained¹ parameters, θ . The most probable segmentation under our probability model is given by

$$\rho^* = \arg \max_{\rho} P(\rho | \theta, D). \quad (4.1)$$

This is equivalent to searching for the segmentation with the smallest energy,

$$\rho^* = \arg \min_{\rho} E(\rho | \theta, D), \quad (4.2)$$

since the normalising partition function $Z(\theta, D)$ does not depend on the segmentation ρ .

As defined in (3.8), this energy is a weighted sum of the feature functions

$$E(\rho | \theta, D) = \sum_{q \in \mathcal{Q}} \theta_q f_q(\rho_q, D). \quad (4.3)$$

A brute-force approach to determine the minimising configuration would evaluate the energy of all possible configurations of ρ . However, with $M^{2NT} \approx 10^{12330}$ possibilities this approach is intractable. In this chapter we describe *belief propagation*, an exact and efficient approach to calculating the minimising configuration in simple graphs. Its approximate extension to general graphs, i.e. *loopy belief propagation*, is also discussed as it relates to our application.

¹Appropriate parameters are estimated in Chapter 5.

In this chapter, as the parameters θ are assumed to be known and paired with an associated feature, we simplify the notation by dropping reference to the parameters, i.e. we consider the energy,

$$E(\rho|\theta, D) = \sum_{q \in Q} F_q(\rho_q, D), \quad (4.4)$$

where $F_q(\rho_q, D) = \theta_q f_q(\rho_q, D)$ is referred to as a *factor*.

4.1 Belief propagation

Belief propagation [42] is an effective technique for inferring values for the latent variables in a graphical model. Messages representing cumulative belief are passed between variables and updated according to a specific order or schedule. Backtracking is then used to recover a solution. For a tutorial on general belief propagation, see e.g. [28]. We briefly describe belief propagation using a simplification of our model.

Consider the simple function minimisation,

$$\min_{\rho} E(\rho) = \min_{\rho_{N-1}} \dots \min_{\rho_0} \left(F(\rho_{N-1}, \rho_{N-2}) + \dots + F(\rho_1, \rho_0) \right), \quad (4.5)$$

where the dependencies of the N bivariate functions form a “chain” over N variables.

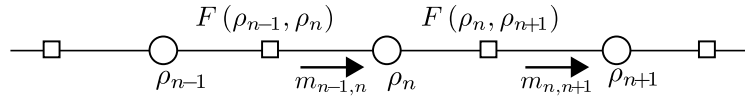


Figure 4.1: Factor graph of a “chain” and propagated messages.

This chain is illustrated by the factor graph in Figure 4.1. As mentioned in Chapter 2, factor graphs [28] are bipartite graphs that describe the dependencies of functions (represented by square nodes) on a shared set of variables (circular nodes).

When each variable ρ_n can take on M values, a direct minimisation over all configurations of $\rho_0 \dots \rho_{N-1}$ requires $O(M^N)$ operations and is therefore intractable for non-trivial applications. As this minimisation is fundamental to inferring optimal segmentation in our application, it is necessary to significantly reduce the number of calculations required.

The distributive law [1] is used to organise these calculations into a more manageable form. Specifically, by using the distributive nature of minimisation over summation, and

conditional independence of all the terms but the last to ρ_0 , we can factorise (4.5) to

$$\min_{\rho} E(\rho) = \min_{\rho_{N-1}} \dots \min_{\rho_1} \left(F(\rho_{N-1}, \rho_{N-2}) + \dots + F(\rho_2, \rho_1) + \min_{\rho_0} F(\rho_1, \rho_0) \right). \quad (4.6)$$

Repeating the process for the variables $\rho_1 \dots \rho_{N-2}$, the following recursive expression is obtained:

$$\min_{\rho} E(\rho) = \min_{\rho_{N-1}} \left(m_{N-2 \rightarrow N-1}(\rho_{N-1}) \right) \quad (4.7)$$

where the “message” to ρ_{n+1} is defined as

$$m_{n \rightarrow n+1}(\rho_{n+1}) = \min_{\rho_n} \left(F(\rho_n, \rho_{n+1}) + m_{n-1 \rightarrow n}(\rho_n) \right) \quad (4.8)$$

and terminates with the message to the first variable $m_{-1 \rightarrow 0}(\rho_0) = 0$.

Starting with the calculation of $m_{0,1}(\rho_1)$ and using (or *propagating*) its result to calculate $m_{1,2}(\rho_2)$, etc., the minimum over all values of ρ requires only $O(NM^2)$ operations.

To find the configuration $\rho_0^* \dots \rho_{N-1}^*$ that minimises the function, the minimising configurations of the direct “parents” are also stored during propagation, i.e.

$$\text{parent}_{n \rightarrow n+1}(\rho_{n+1}) = \arg \min_{\rho_n} \left(F(\rho_n, \rho_{n+1}) + m_{n-1 \rightarrow n}(\rho_n) \right). \quad (4.9)$$

After propagation these values are used to “backtrack” from the end of the chain to recover the complete minimising configuration through

$$\rho_{N-1}^* = \arg \min_{\rho_{N-1}} m_{N-2 \rightarrow N-1}(\rho_{N-1}), \quad (4.10)$$

and

$$\rho_n^* = \text{parent}_{n \rightarrow n+1}(\rho_{n+1}^*). \quad (4.11)$$

When applied to acyclic chains or tree structures, this belief propagation and backtracking is a form of dynamic programming, equivalent to the Viterbi algorithm [55], and results in a unique exact solution.

Since each message is a function of the M discrete values of ρ_n , we can represent each message as a vector of M “belief values” together with a vector containing corresponding

“minimising parent” identity values for each of the M values.

As mentioned above, the calculation of messages is quadratic in the variable cardinality, M . If it is known that certain combinations of ρ_n and ρ_{n+1} typically have very high function values, e.g. are unlikely to occur together in a good segmentation, then a technique similar to a *beam search* can be applied. For example, if the factor function sufficiently penalises large differences between successive variables, e.g. $F(\rho_n, \rho_{n+1}) \propto (\rho_n - \rho_{n+1})^2$, then in the minimisation in (4.8) only values for ρ_n where $|\rho_n - \rho_{n+1}| < \epsilon_M$ need to be considered. Here ϵ_M denotes the range which is application and message type dependent. This range of likely values can be selected based on inspection of the factor function and training examples. Since these unlikely combinations of radial values can be safely ignored in the calculation of a message the computational complexity can be reduced to $O(NM\epsilon_M)$ operations, which is significant if M is large.

4.2 Loopy belief propagation

The factors for the segmentation problem in this study do not form simple linear chains (see Figure 3.3), or a tree structure, as is required by belief propagation. Not only are the first and last radius variables in each contour “connected” (i.e. there is a feature function $f(\rho_{N-1}, \rho_0)$), there also exist many loops due to the presence of feature functions connecting variables over time as well as between the inner and outer contour variables.

It is, however, rather straightforward (see [27] for examples) to iteratively apply belief propagation of these same belief messages even in the presence of loops in a graph, resulting in *loopy belief propagation* algorithms [43].

We will now briefly state the loopy belief propagation algorithm as applied to general graphs. Let variable nodes be represented by v and factor nodes by u . Messages are sent from variable nodes to factor nodes, $m_{v \rightarrow u}$, and by factor nodes to variable nodes, $m_{u \rightarrow v}$.

On min-sum problems, the message sent by a variable node to a factor node is proportional to the sum of the incoming messages to that node,

$$m_{v \rightarrow u}(\rho_v) \propto \sum_{u' \in N(v) \setminus \{u\}} m_{u' \rightarrow v}(\rho_v), \quad (4.12)$$

where $N(v) \setminus \{u\}$ are the factor nodes connected to that variable, excluding the target factor.

The message that is sent from a factor node to a variable node is proportional to a minimisation over the sum of the incoming messages and the factor's function,

$$m_{u \rightarrow v}(\rho_v) \propto \min_{\rho'_u: \rho'_v = \rho_v} \left(F(\rho'_u) + \sum_{v^* \in N(u) \setminus \{v\}} m_{v^* \rightarrow u}(\rho'_{v^*}) \right), \quad (4.13)$$

where $N(u) \setminus \{v\}$ is the variable nodes connected to that factor, excluding the variable to which the message is sent.

The message described by (4.13) reduces to (4.8) when a variable has only a single parent factor. When applied to chain and tree structured graphs, this algorithm is equivalent to the dynamic programming discussed in the previous section. The proportionality is due to the omission of a normalising constant.

Our implementation stores only the messages emitted from factor nodes since messages sent from variable nodes can be discarded after being used to calculate the former. Messages that are passed from a factor to a variable node, $\rho_n(t)$, can be grouped into one of six types according to their "direction":

1. $m_R(n, t)$: Message sent "rightwards" from the factor connecting variables $\rho_{n-1}(t)$ and $\rho_n(t)$.
2. $m_L(n, t)$: Message sent "leftwards" from the factor connecting variables $\rho_{n+1}(t)$ and $\rho_n(t)$.
3. $m_U(n, t)$: Message sent "upwards" from the factor connecting variables $\rho_n(t-1)$ and $\rho_n(t)$.
4. $m_D(n, t)$: Message sent "downwards" from the factor connecting variables $\rho_n(t+1)$ and $\rho_n(t)$.
5. $m_I(n, t)$: Message sent "inwards" from the factor connecting variables $\rho_n^{\text{out}}(t)$ and $\rho_n^{\text{in}}(t)$.
6. $m_O(n, t)$: Message sent "outwards" from the factor connecting variables $\rho_n^{\text{in}}(t)$ and $\rho_n^{\text{out}}(t)$.

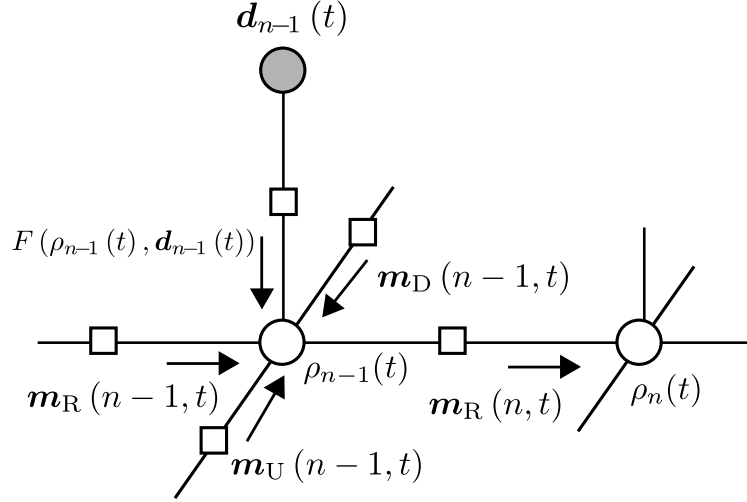


Figure 4.2: Messages propagated in a partial factor graph to construct $\mathbf{m}_R(n, t)$. The inwards message from the outer contour, $\mathbf{m}_I(n-1, t)$ is not shown.

As an example, consider $\mathbf{m}_R^{\text{in}}(n, t)$, the rightwards message received by the variable $\rho_n^{\text{in}}(t)$ from the feature connecting it to $\rho_{n-1}^{\text{in}}(t)$, as shown in Figure 4.2,

$$\begin{aligned}
 \mathbf{m}_R^{\text{in}}(n, t) = & \min_{\rho_{n-1}^{\text{in}}(t)} \left(F\left(\rho_{n-1}^{\text{in}}(t), \mathbf{d}_{n-1}(t)\right) \right. \\
 & + F_r\left(\rho_n^{\text{in}}(t), \rho_{n-1}^{\text{in}}(t)\right) + F_r''\left(\rho_n^{\text{in}}(t), \rho_{n-1}^{\text{in}}(t)\right) \\
 & + \mathbf{m}_R^{\text{in}}(n-1, t) + \mathbf{m}_I(n-1, t) \\
 & \left. + \mathbf{m}_U^{\text{in}}(n-1, t) + \mathbf{m}_D^{\text{in}}(n-1, t) \right) - Z_{n,t}. \quad (4.14)
 \end{aligned}$$

The factors $F_r = \theta_r f_r$ and $F_r'' = \theta_r'' f_r''$ in (4.14) are the local functions that restrict the variations of radii, as discussed in Section 3.4.2, and F is the function derived from the edge classifier.

The message that is sent from a factor connected to an observed variable, e.g. the observed image in the radial direction $\mathbf{d}_{n-1}(t)$, is a function of only the radius, $F(\rho_{n-1}^{\text{in}}(t), \mathbf{d}_{n-1}(t))$, i.e. the factor based on the edge classifier, since the image is known during segmentation.

The normalisation value $Z_{n,t}$ ensures that $\min_{\rho_n(t)} \mathbf{m}_R(n, t) = 0$. Due to the fact that features in an undirected graphical model, such as the CRF, are allowed to take on arbitrary values, propagated messages do not necessarily represent marginal probabilities, as is the case in directed graphs. If loopy belief propagation is applied without message normali-

sation, numerical overflow can occur after only a few iterations. We therefore normalise our messages by subtracting² the smallest value in each message before it is propagated. This ensures that the minimum value in each message is zero. For a “max-product” setting this is equivalent to normalising each message so that the largest value in the message is one. We have applied message normalisation and have not encountered any numerical overflow or underflow issues. For a discussion on the role of normalisation in belief propagation, see [35].

As mentioned in Section 3.4.2, to reduce computational cost, minimisation in (4.14) is only calculated over a sub range of $\rho_{n-1}(t)$ where $|\rho_n(t) - \rho_{n-1}(t)| \leq 8$, as larger radial differences are unlikely to occur.

For completeness we show the other propagated messages. Messages sent to the inner contour variables are

$$\begin{aligned} m_L^{\text{in}}(n, t) = & \min_{\rho_{n+1}(t)} \left(F(\rho_{n+1}^{\text{in}}(t), d_{n+1}(t)) \right. \\ & + F_r(\rho_{n+1}^{\text{in}}(t), \rho^{\text{in}}(t)) + F_r''(\rho_{n+1}^{\text{in}}(t), \rho^{\text{in}}(t)) \\ & + m_L^{\text{in}}(n+1, t) + m_I(n+1, t) \\ & \left. + m_U^{\text{in}}(n+1, t) + m_D^{\text{in}}(n+1, t) \right) - Z_{n,t}, \end{aligned} \quad (4.15)$$

$$\begin{aligned} m_U^{\text{in}}(n, t) = & \min_{\rho_n(t-1)} \left(F(\rho_n^{\text{in}}(t-1), d_n(t-1)) \right. \\ & + F_t(\rho_n^{\text{in}}(t), \rho_n^{\text{in}}(t-1)) + F_t'(\rho_n^{\text{in}}(t), \rho_n^{\text{in}}(t-1)) \\ & + F_t''(\rho_n^{\text{in}}(t), \rho_n^{\text{in}}(t-1), d_n(t)) \\ & + m_R^{\text{in}}(n, t-1) + m_I(n, t-1) \\ & \left. + m_U^{\text{in}}(n, t-1) + m_D^{\text{in}}(n, t-1) \right) - Z_{n,t} \end{aligned} \quad (4.16)$$

²We are working in the log domain.

and

$$\begin{aligned}
m_D^{\text{in}}(n, t) = & \min_{\rho_n(t+1)} \left(F(\rho_n^{\text{in}}(t+1), d_n(t+1)) \right. \\
& + F_t(\rho_n^{\text{in}}(t+1), \rho_n^{\text{in}}(t)) + F_t'(\rho_n^{\text{in}}(t+1), \rho_n^{\text{in}}(t)) \\
& + F_t''(\rho_n^{\text{in}}(t+1), \rho_n^{\text{in}}(t), d_n(t)) \\
& + m_R^{\text{in}}(n, t+1) + m_I(n, t+1) \\
& \left. + m_U^{\text{in}}(n, t+1) + m_D^{\text{in}}(n, t+1) \right) - Z_{n,t}. \tag{4.17}
\end{aligned}$$

The messages to the outer contour variables, $m_R^{\text{out}}(n, t)$, $m_L^{\text{out}}(n, t)$, $m_U^{\text{in}}(n, t)$, $m_D^{\text{in}}(n, t)$ are similar, except for replacing the inwards messages $m_I(\dots)$ with outwards messages $m_O(\dots)$. The outwards messages, sent from the inner contour variables to the outer, are defined as

$$\begin{aligned}
m_O(n, t) = & \min_{\rho_n(t-1)} \left(F(\rho_n^{\text{in}}(t), d_n(t)) \right. \\
& + F_1^{\text{cross}}(\rho_n^{\text{in}}(t), \rho_n^{\text{out}}(t)) \\
& + F_2^{\text{cross}}(\rho_n^{\text{in}}(t), \rho_n^{\text{out}}(t), d_n(t)) \\
& + m_R^{\text{in}}(n, t) + m_L^{\text{in}}(n, t) \\
& \left. + m_U^{\text{in}}(n, t) + m_D^{\text{in}}(n, t) \right) - Z_{n,t} \tag{4.18}
\end{aligned}$$

and similarly the inwards messages $m_I(n, t)$, with the “in” and “out” labels exchanged.

4.3 Message schedule

In acyclic chains and trees, the order that messages are propagated is clear, i.e. messages are calculated sequentially starting from the “leaf” nodes. However, in graphs containing loops some kind of propagation schedule is required.

In a parallel propagation schedule all messages are updated simultaneously after each iteration. We have found that, in a parallel propagation schedule, the effects from a feature function propagate relatively slowly through the graphical model. A similar observation is made by Goldberger and Kfir [17]. For this reason we choose a sequential propagation

schedule which, in our experience, allows for faster convergence.

We briefly discuss some issues related to the convergence of belief propagation and its effect on the choice of an appropriate propagation schedule and backtracking order.

When applied to graphs with a single loop, belief propagation converges to a stable fixed point (which is globally optimal) or oscillates periodically [58]. Convergence and global optimality for graphs with as many loops as ours is, however, not guaranteed [43].

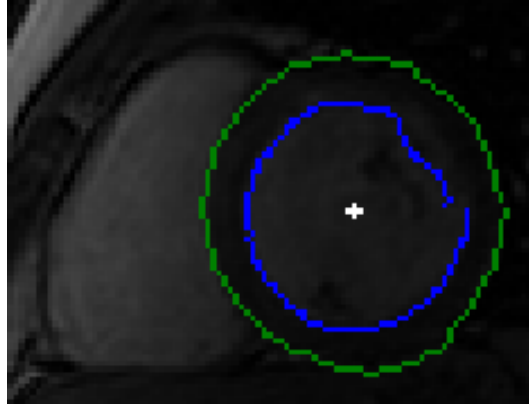


Figure 4.3: Example of a non-continuous inner contour.

Through experimentation we find that the inferred contours are sensitive to the order of message propagation and backtracking. Because a global optimum is not guaranteed we risk obtaining contours that deviate from desired behaviour, such as contours with sudden “jumps” (e.g. Figure 4.3) or inner-outer ratios that are inconsistent with the training data.

Before further discussing the optimality of belief propagation, let us consider the optimality of using an inference technique based on a basic Gibbs sampler to infer segmentation values, i.e. taking each radial variable in turn and selecting its optimal value conditioned on the rest. On convergence, no assignment to an individual radial variable thus can further improve upon this result. The neighbourhood within which this algorithm is optimal is therefore a single variable. To improve upon this local minimum, larger neighbourhoods need to be considered.

Weiss and Freeman [58] describe the Single Loops and Trees (SLT) neighbourhood within which the result of a loopy belief propagation algorithm is optimal. The SLT neighbourhood of a solution is defined as taking a subset of the variables that form disconnected combinations of trees and single loops, in the graphical model, and assigning them arbi-

trary values while keeping the other variables fixed. It is guaranteed that no assignment to these variables will yield better results, i.e. on convergence,

$$E(\boldsymbol{\rho}^* | \boldsymbol{\theta}, D) \leq E(\text{SLT}(\boldsymbol{\rho}^*) | \boldsymbol{\theta}, D). \quad (4.19)$$

The SLT neighbourhood is thus significantly larger than that of the Gibbs sampler and loopy belief propagation is thus much less likely to yield locally optimal configurations.

The nature of the SLT neighbourhood is utilised to inform us on the order of propagation and backtracking. Since we can only guarantee optimality within “tree” and “loop” subsets, we implicitly select the subsets, through propagation schedule and backtracking order, that encapsulate the “most important” factors and eliminate potentially problematic configurations.

We are primarily interested in deriving contours that are almost continuous in the log-polar space. We therefore select a propagation schedule and backtracking order that places more emphasis on contour continuity (i.e. the relationship between ρ_n and ρ_{n-1}) than the other radial relationships.

Specifically, for variables representing the inner contour, messages are first propagated in an angular direction ($n = 0, \dots, N-1$) and reversed ($n = N-1, \dots, 0$) before being propagated to the next temporal frame ($t = 0, \dots, T-1$) and back ($t = T-1, \dots, 0$). Similar steps are then repeated for the outer contour, taking into account messages passed from the inner contour.

The same reasoning is used in the selection of a backtracking order, i.e. we backtrack over all nodes in the inner and outer contours independently.

It is found that such a propagation schedule and backtracking order increases the probability that the inferred inner and outer contours in a single image, form continuous loops. We have not observed divergence or significant oscillation between configurations in our application.

4.4 Implementation

The majority of the software is implemented in the Python programming language with belief propagation implemented in C. Cordero-Grande et al. [11] reports using an MRF to segment a single 4D video sequence in approximately 56 minutes on a single 800MHz processor with 4MB cache.

For a similar number of images (12 slice positions with 20 frames each) on a single core of a 3400MHz processor with 8MB cache we can report a radial segmentation runtime of approximately 2 minutes. Since the slices are independent in our model, segmentation can easily be distributed in a multi-core environment. With approximately 12 slices per subject this can reduce the segmentation time by an order of magnitude for a single subject and potentially more if multiple subjects are to be analysed.

We have not included the time to estimate the centre point positions (approximately an additional minute for each 20 frame video sequence) as it can be replaced by a faster heuristic technique, such as the one briefly discussed in Section 3.2. Additionally, the time taken to derive the edge-classification features is negligible if a fast implementation is used. Such an implementation could take into account the equivalences between the “sliding window” classification and cross-correlation of the log-polar image with appropriate weights.

Our radial inference technique, which uses belief propagation, is thus approximately six times faster than [11] (after adjusting for our approximately four times faster processor), which can be attributed to their use of a Gibbs sampler during inference. Note also that the number of radial points we use to represent a contour and its radial discretisation are 128 and 256, respectively. This is significantly larger than the values of 7 and 31 used by Cordero-Grande et al., especially when comparing the size of the resulting configuration space.

After profiling the time complexity of our implementation, it was determined that a large part of belief propagation computational time is spent on calculating the messages that are passed between the inner and outer contours. This is to be expected as the ϵ_M is relatively large for these messages, i.e only a small number of “parent values” are skipped during calculation. This is due to the relatively wide range of wall widths possible at e.g. maximum contraction, as can be observed in Figure 3.7 at $t = 8$. It should be possible to

carefully select different ϵ_M for different frames in the cycle, however this is not done in our implementation.

4.5 Summary

To summarise, in this chapter we apply a loopy belief propagation algorithm with sequential message passing to derive a segmentation for a new video. We briefly discuss the messages and their normalisation together with the approximate nature of the algorithm. We remark on properties of the implementation and the speed of inference which is sufficient for our application.

Chapter 5

Parameter estimation

In this chapter we address the selection of CRF parameters, θ^* , which are used to weigh the effects of the feature functions relative to one another.

We discuss some of the disadvantages of a Maximum Likelihood formulation for parameter estimation and consider two black-box approaches to directly minimise the segmentation error: one in which the gradient is estimated numerically and another which is a gradient free technique.

We also discuss two loss functions: one used during training in this chapter and another used during evaluation in Chapter 6.

5.1 Maximum likelihood estimation

Before performing segmentation of an observed video sequence, as described in Chapter 4, suitable CRF model parameters, θ^* , are needed. A popular machine learning approach is to use parameters that “best explain the data”, i.e. maximise the likelihood of the training annotations. For our model this is expressed as

$$\theta^* = \arg \max_{\theta} \prod_i P(\rho^{(i)} | D^{(i)}, \theta), \quad (5.1)$$

where $D^{(i)}$ is a video sequence from a training dataset and $\rho^{(i)}$ is its human annotated segmentation. An important property of this maximum likelihood formulation (see e.g. [5])

is that it leads to an objective function that is convex with respect to its parameters (see e.g. [27]). This convexity is highly desirable as many optimisation techniques are guaranteed to converge to a unique global optimum. Often an iterative gradient-based method is used to search for this optimum.

Calculating the likelihood in (5.1) (and its derivatives) for a specific θ , requires evaluation of the partition function (and its moments). The derivative of the partition function $Z(\theta, D^{(i)})$ with respect to a specific parameter θ_q is given by

$$\frac{\partial Z(\theta, D^{(i)})}{\partial \theta_q} = - \sum_{\rho} \left(\exp \left(- \sum_{q'} \theta_{q'} f_{q'}(\rho_{q'}, D^{(i)}) \right) \cdot f_q(\rho_{q'}, D^{(i)}) \right). \quad (5.2)$$

Here the sum is again over all configurations of ρ which requires $O(M^{2NT})$ operations for general graphs. This is one of the significant challenges in applying CRFs in practice since the calculation quickly becomes intractable.

Attempts have been made by others [4, 52] to approximate the partition function. These result in approximate distributions such as the *pseudolikelihood*. Application of these techniques to our distribution, P , as described in Section 3.3, would therefore result in an approximate distribution \tilde{P} .

Apart from these computational difficulties, recall that loopy belief propagation yields an approximate minimising configuration to our distribution P . Equivalently, this can be interpreted as an exact minimising configuration to a related, hypothetical distribution, Q . The relationship between P and Q is discussed by [58] and in [27].

The effect of using parameters that maximise the likelihood of the training data under \tilde{P} , to infer values from the distribution Q is unclear. Since we are primarily interested in obtaining parameters that yield good segmentations under inference and less with estimating the “true” model distribution, we consider alternatives that avoid calculating the partition function.

5.2 The loss functions

Before continuing, we address the topic of the “loss function” or metrics used to compare inferred segmentations from our model to the human annotated ground truth.

A number of error metrics are described in the literature (see e.g. [3, 10, 38, 46]). Here we briefly discuss two that focus on the intersecting area of two contours and the distance between points on the contours.

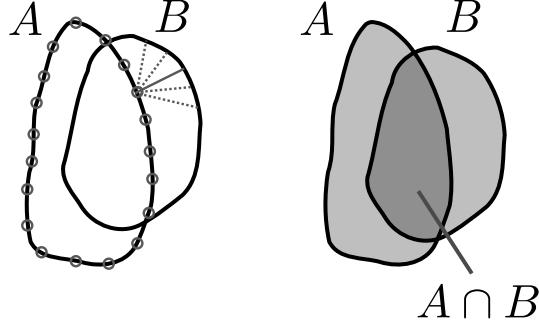


Figure 5.1: Illustration of the landmark distance of a single point and the Dice similarity of two areas.

5.2.1 Landmark distance

The *landmark distance* [3] is the average of the shortest distance between each point on the ground truth contour, A , and the inferred contour, B , i.e.

$$e_{\text{landmark}}(A, B) = \frac{1}{N} \sum_{n=0}^{N-1} \min_{n'=0..N'-1} \|a_n - b_{n'}\|, \quad (5.3)$$

where a_n and $b_{n'}$ are points on the two contours and N' is the number of points in contour B . To minimise the effect of discretisation, points on contour B are linearly interpolated, so that $N' = 1000$. See Figure 5.1 for an illustration.

Our experiments indicate that contours inferred by minimising this distance, tend to be jagged. This could be caused by the asymmetry of the landmark distance metric. More specifically, $e_{\text{landmark}}(A, B) \neq e_{\text{landmark}}(B, A)$, which allows the inferred contour to vary wildly while still maintaining small minimum distances, as long as there are short distances from each of the points on the ground truth. For this reason we construct a symmetric distance through the average of the “ground truth to inferred” distance and the “inferred to ground truth” distance, i.e.

$$e'_{\text{landmark}}(A, B) = \frac{e_{\text{landmark}}(A, B) + e_{\text{landmark}}(B, A)}{2}. \quad (5.4)$$

This symmetric landmark distance is used during the parameter estimation stage. In order to obtain a fair comparison of our results with that of Andreopoulos [3], we use the asymmetric landmark distance during evaluation in Section 6.1.

It is also common to convert these distances from pixels into millimetres using a conversion factor that is determined during the MRI acquisition process and provided with the dataset. These conversion factors are dependent on the level of magnification used and are, in part, influenced by the operator’s protocol and the size of the heart. The human heart differs significantly in size between subjects as it is dependent on properties such as age, sex, height and weight [40]. While a conversion to millimetres thus aids in a medically intuitive interpretation, it also results in some images being arbitrarily penalised.

In our opinion, measurements of segmentation accuracy should thus be measured in pixels or dimensionless quantities (e.g. a percentage) as is common in non-medical image segmentation literature.

We therefore use a single shared conversion factor for all images during training and evaluation to allow an intuitive medical interpretation. This shared conversion factor is equal to the mean of the conversion factors of all annotated frames in the dataset.

Note that the size of the landmark distance does not have an upper bound and can thus significantly influence results if outliers exist in the training or evaluation set.

5.2.2 Dice error metric

The *Dice coefficient* metric function [12], s_{dice} , is a measure of the amount of overlap between the area inside an inferred contour and the ground truth contour (see Figure 5.1):

$$s_{\text{dice}}(A, B) = \frac{2 \|A \cap B\|}{\|A\| + \|B\|}. \quad (5.5)$$

The *Dice error*, $e_{\text{dice}} = 1 - s_{\text{dice}}$, is therefore the amount of non-overlap of two contours. For ease of implementation, we estimate the areas inside the contours (and their intersection) by the number of pixels inside these areas, as drawn on the original untransformed images.

The average of the Dice errors of the inner and outer contours is used as the Dice error of a single image’s segmentation. The Dice error of a sequence is defined as the average of

the Dice errors of the individual image segmentations.

As the Dice error lies between zero and one, the effect of outlier segmentation errors is mitigated. For this reason we use the Dice error to analyse the general behaviour over time and at different slice positions.

The Dice error is, however, not very sensitive to minor variations in the contours being compared. This is a disadvantage if close structural similarity is sought, and the Dice error is therefore not used during parameter estimation.

5.3 Black box parameter estimation

Consider a video sequence, $D^{(i)}$, from a training dataset and its human annotated segmentation, $\rho^{(i)}$. Fundamentally, we are interested in obtaining the parameters θ^* that would lead to a segmentation, $\rho^{*(i)}$, of the sequence, that does not significantly differ from the annotated segmentation, $\rho^{(i)}$.

Moreover, we wish to find parameters, $\theta^* = \arg \min_{\theta} J(\theta)$, that minimises an objective function representing the average error over the entire training dataset

$$J(\theta) = \sum_i e'_{\text{landmark}}(\rho^{(i)}, \rho^{*(i)}). \quad (5.6)$$

Here the inferred segmentation of a video sequence (from belief propagation) is

$$\rho^{*(i)} = \arg \max_{\rho} Q(\rho | \theta, D^{(i)}). \quad (5.7)$$

The function e'_{landmark} is the symmetric landmark distance between the ground-truth and the inferred segmentations.

This black-box approach does not contain information on the partition function, due to the fact that (5.7) can be inferred without its calculation, as discussed in Chapter 4. However, without an explicit analytical expression for the gradient, optimisation methods need to either approximate the derivative numerically or somehow avoid its calculation. These are discussed in Section 5.3.1 and 5.3.2, respectively.

Although each evaluation of the objective function requires re-segmentation of all the video sequences in the training set, the calculation of their centre points (and edge fea-

tures) are independent of the CRF parameters and can be re-used to reduce the evaluation time. This is a noteworthy advantage of estimating the centre points separately from the segmentation.

After distributing the segmentation of our training set across our multi-core environment, each objective function evaluation takes approximately 30 seconds to complete. We are therefore interested in convergence in relatively few iterations.

We investigate two popular optimisation techniques: BFGS with a numerical gradient approximation and the gradient-free Powell’s method.

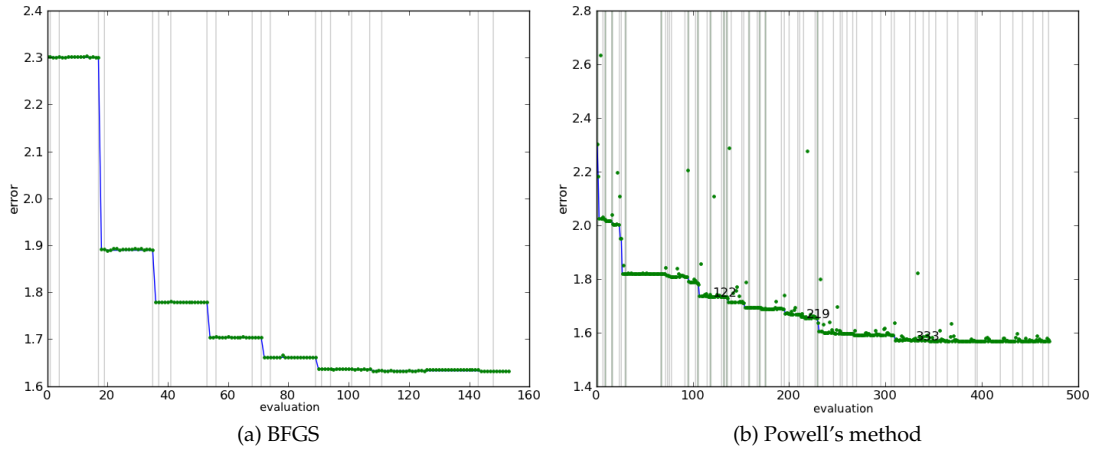


Figure 5.2: Objective function evaluations (green points) against iteration. The blue line indicates a lower bound, i.e. the best value encountered up to this iteration. The vertical bars indicate iterations of decreasing error.

5.3.1 Gradient approximation method

We investigate the application of the limited memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (see e.g. [39]) as implemented in SciPy’s optimisation library [24]. BFGS and related techniques generally perform well, even for the optimisation of non-smooth objective functions [30, 50].

To constrain the parameters, θ , as they are passed to the objective function, to strictly positive values, the parameters are transformed before function evaluation, i.e.

$$J'(w) = J(\theta = \exp(w)), \quad (5.8)$$

allowing the parameters w , as operated on by the BFGS optimiser, to take on both positive and negative values.

BFGS requires access to the gradient of the objective function, which we approximate numerically through a finite difference

$$\frac{\partial J'(w)}{\partial w_d} \approx \frac{J'(w + \Delta w_d) - J'(w)}{\|\Delta w_d\|}, \quad (5.9)$$

where $\|\Delta w_d\| = \log(1.01)$ is a small delta in a base direction. Our choice of delta thus evaluates each parameter direction with a change of 1% in the original parameter space.

We observe convergence after approximately 90 function evaluations (see Figure 5.2a).

5.3.2 Gradient-free method

There exist a number of techniques designed to minimise an objective function without explicitly calculating its gradient. For a review of these algorithms and implementations see [47]. Many of these gradient-free techniques are designed to work efficiently in relatively low dimensional spaces (i.e. less than 20). Fortunately, our objective function has only 16 parameters, making the application of these methods feasible.

We use Powell’s conjugate direction method [45] to search for suitable parameters without calculating a gradient. During each iteration, Powell’s method successively calculates bidirectional line searches along directions contained in a list. The list of directions is initialised as the orthogonal bases. After each iteration, the resultant direction of improvement replaces the most correlated direction in the list.

The goal behind Powell’s method is to update the directions in which line searches are done, such that they are aligned with the most promising direction, and thus avoid “zigzagging” towards an optimum. We use the SciPy implementation [24], based on Brent’s method [6] for line searches where a bracketing interval of plausible values for the minimum in a direction is gradually refined.

To constrain parameters to strictly positive values, the exponent of the parameters is used during function evaluation, similar to Section 5.3.1.

Powell’s method is known to converge for quadratic and strictly convex objective functions (see e.g. [60]), which our objective function is not. We have, however, observed good

behaviour for our application and briefly investigate the nature of the objective function in the next section. We observe convergence after approximately 300 function evaluations (see Figure 5.2b).

5.3.3 Convexity

Many optimisation techniques converge to local optima or stationary points. If the objective function is convex then this point will be a global optimum. This is not often the case and if multiple local optima are present, special measures are needed in order to ensure a global optimum. In many cases this is too expensive and one has to settle for a good, if not global, optimum.

An analysis of the convexity of our objective function is complicated by the absence of an analytical gradient. A cursory numerical inspection of the objective function (see Figure 5.3) provides us with some information on its behaviour.

The objective function is evaluated at different positions, $J(\theta')$, in each of the base dimensions, d , by multiplicatively scaling (to ensure parameters are positive) the obtained optimal parameters θ^* , i.e.

$$J(\theta') = J(\theta^* \cdot \exp(\lambda_d e_d)). \quad (5.10)$$

This “line scan” in each of the dimensions illustrates how the objective function changes as each parameter is varied. Noting that the sign of the objective function’s gradient changes more than once and is zero over multiple values, it is clear that the objective function is not convex.

This can be understood by considering the segmentation of a single video sequence after a very slight change in the parameters. If the change is small enough, it is likely that it will have no effect on the results of the inference process, due to the “arg max” in (5.7) and finite discrete values possible for ρ . This will result in plateaus in the objective function which can cause problems for techniques that rely on an equivalence between stationary points and a global minimum.

The segmentation of multiple video sequences influence these plateaus, but does not eliminate the problem, i.e., the objective function remains non-convex.

5.4 Remarks

For our objective function, BFGS reaches its optimum in less function evaluations than Powell’s method. Powell’s method, however, reaches a better optimum if allowed to run for more iterations. The numerical estimate of the gradient, as used by BFGS, is problematic. It is possible to choose a delta, $\|\Delta w_d\|$, that is too small since the evaluated objective function does not change under very small variations in parameters. A sufficiently large delta causes BFGS to initially advance rapidly towards a good optimum, but then to not improve due to the rough estimate of the gradient.

Early experiments suggest that searching with BFGS for a few iterations, to quickly find an approximate good region, followed by Powell’s method, shortens the total number of function evaluations required to reach the minimum yielded by Powell’s method alone. Searching with Powell’s method followed by BFGS does not improve results.

Due to the relatively low dimensionality of the parameter search space (16 dimensions) and the informative nature of each annotated video sequence we have not encountered problems with data scarcity. It is, however, advisable that the training set contains a sufficient number of video sequences in which the papillary muscles obscure the endocardium border.

The 300 iterations of Powell’s method takes approximately 2 hours on our hardware (3400MHz processor with 8MB cache) if distributed across multiple processors.

For other approaches that focus on directly minimising a loss function and integration of inference and training see e.g. [57]. Additionally, see [15] for different discriminative training techniques of max-sum classifiers.

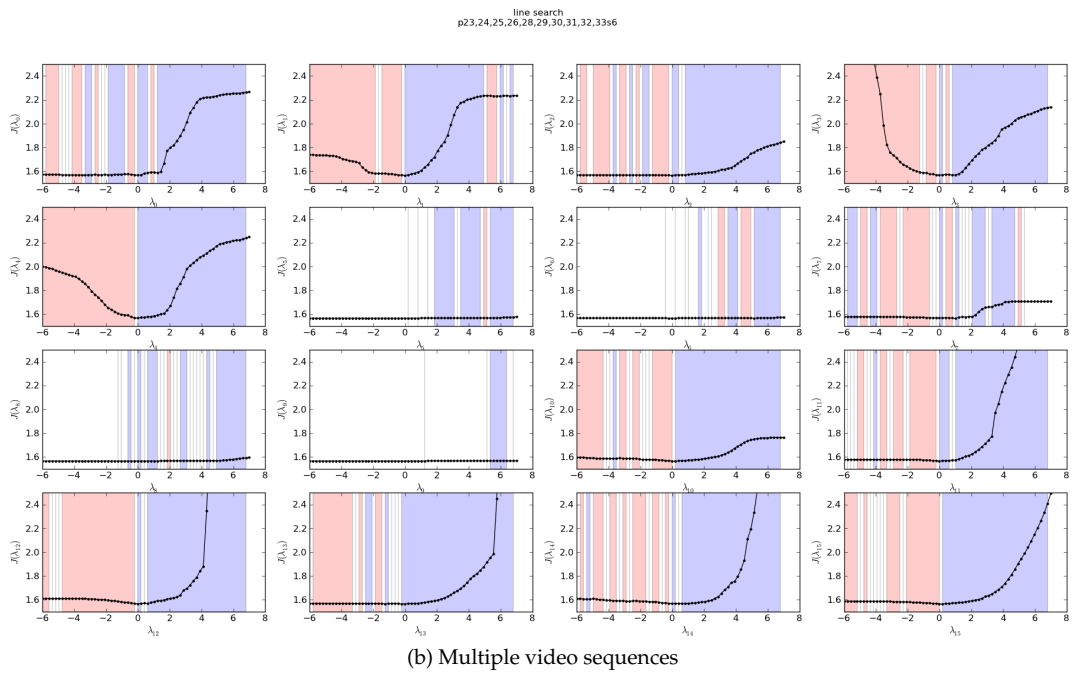
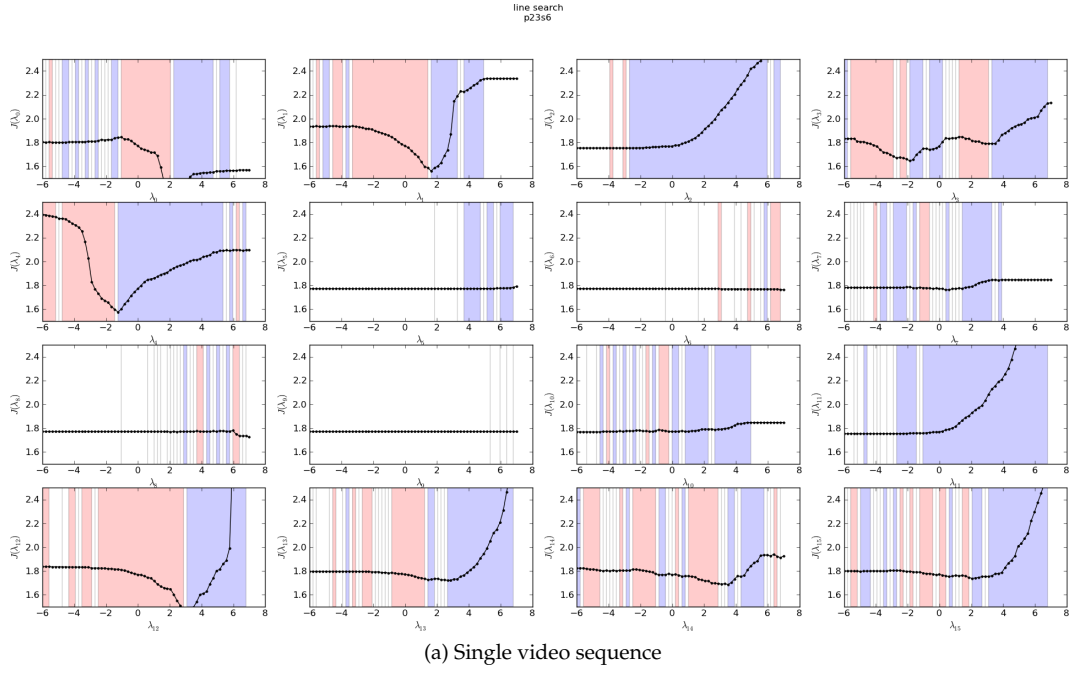


Figure 5.3: Line scan of the objective function, illustrating local minima. Red areas represent negative gradients and blue, positive.

Chapter 6

Results and discussion

In this chapter we analyse the segmentations produced by our process and compare them to those of a few existing techniques. Since it is difficult to illustrate video sequences in a manuscript like this, the reader is advised to view the video available at our website¹ and the 3D image included in Appendix C.

Some annotated cardiac MRI datasets are publicly available, e.g. [3, 38, 46], but have not yet been widely standardised; it is not unusual for different authors to evaluate their segmentation techniques on different datasets. This makes a comparison between different reported approaches difficult [44] since significant differences between datasets exist. Some datasets are, for example, based on patients with known cardiac problems and others on volunteer medical students that are generally healthy. Healthy hearts have less variance in shape and have more predictable temporal behaviour, which makes their segmentation easier.

A comparison of results reported in literature is further complicated by differing error criteria between techniques; variations on the Dice and landmark errors, as defined in Section 5.2, are often used.

We evaluate our model on two datasets. Our segmentation process is trained and analysed on the York dataset [3] with respect to segmentation behaviour and its sensitivity to placement of the initial centre point. This dataset contains ground truth annotations for all frames and therefore contains important temporal information.

¹<http://dip.sun.ac.za/~janto/>

For comparison with other authors our technique is evaluated on the Sunnybrook [46] MRI dataset. This dataset is not annotated for all frames, but has the advantage that more authors have used this set to report their results.

6.1 York cardiac segmentation dataset

We evaluate our model on the MRI York dataset [3] provided by the Department of Diagnostic Imaging of the Hospital for Sick Children in Toronto and annotated by Andreopoulos of York University. The dataset contains video sequences from 33 subjects, all under the age of 18, displaying a variety of heart abnormalities such as cardiomyopathy, aortic regurgitation, enlarged ventricles and ischemia. We split the dataset into three cross-validation sets: 11 subjects for training of the edge classifier, 11 subjects for CRF parameter estimation and 11 subjects for evaluation. Each set effectively contains approximately 100 video sequences and each video contains 20 frames at different z-axis slice positions. The inner and outer contours are manually annotated for all frames and are used as the ground truth in our experiments.

For efficiency we only use video sequences of the sixth z-axis slice of each patient to estimate the CRF parameters. These slices do not necessarily coincide with the mid ventricle for all patients. The number of slices where the ventricle is visible differs between patients and is likely dependent on patient pathology. A different choice of training data is thus likely to have a significant effect on the results of patients with, for example, hypertrophy. This, however, has not been further explored.

From a visual inspection of the ground truth it is clear that there are inconsistencies in the human annotations with regard to the inclusion of the papillary muscles in the inner contour. These inconsistencies reduce the discriminative ability of the edge classifier and influence the optimal CRF parameter values estimated during training. Also, because the human annotated contours are used for evaluation, inconsistencies of the human annotations need to be taken into account when interpreting any results based on this as ground truth. In short: inconsistent examples in the evaluation set will result in an upper limit to the accuracy achievable by any consistent system.

6.1.1 Qualitative segmentation analysis

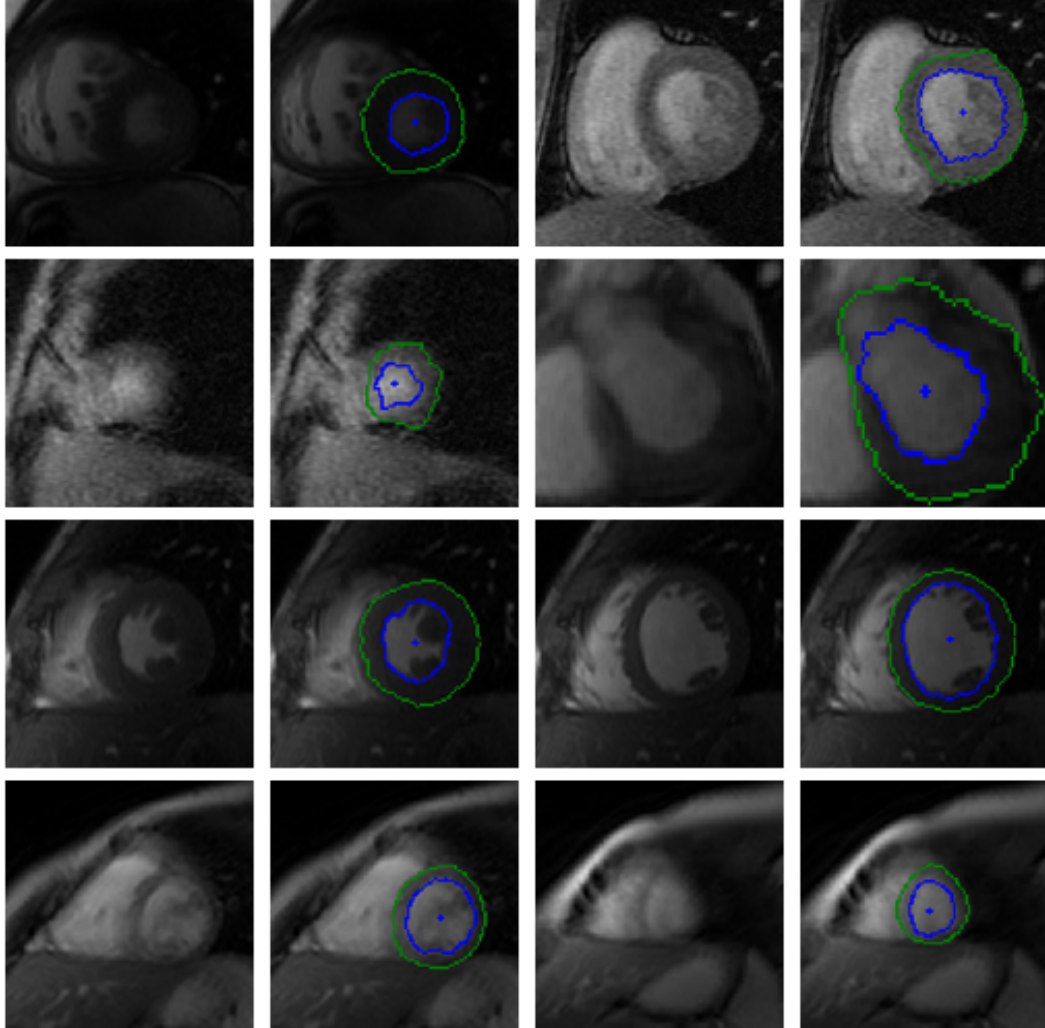


Figure 6.1: Selection of images and their automatically segmented contours (inner contour is blue and outer is green) inferred from a testing set. The blue dot in the middle of the endocardium is the estimated centre point.

Figure 6.1 illustrates our results on a selection of images from the testing dataset. A visual inspection indicates that, for the majority of video sequences, our automated annotations are in line with expected behaviour with regard to shape, position and motion. Of particular interest is the inclusion of the papillary muscles inside the inner contour. In many of these cases the segmentation process is able to use local shape and temporal behaviour to identify the inner contour even though the edge is weak or absent.

We also observe robustness to some noisy images. These examples suggest that in some

of these images where the automatic segmentation differs from the ground truth, the automatic segmentation is superior to the manual approach. This is attributed to the fact that the automated system is able to integrate temporal behaviour, something that is an arduous task for a human.

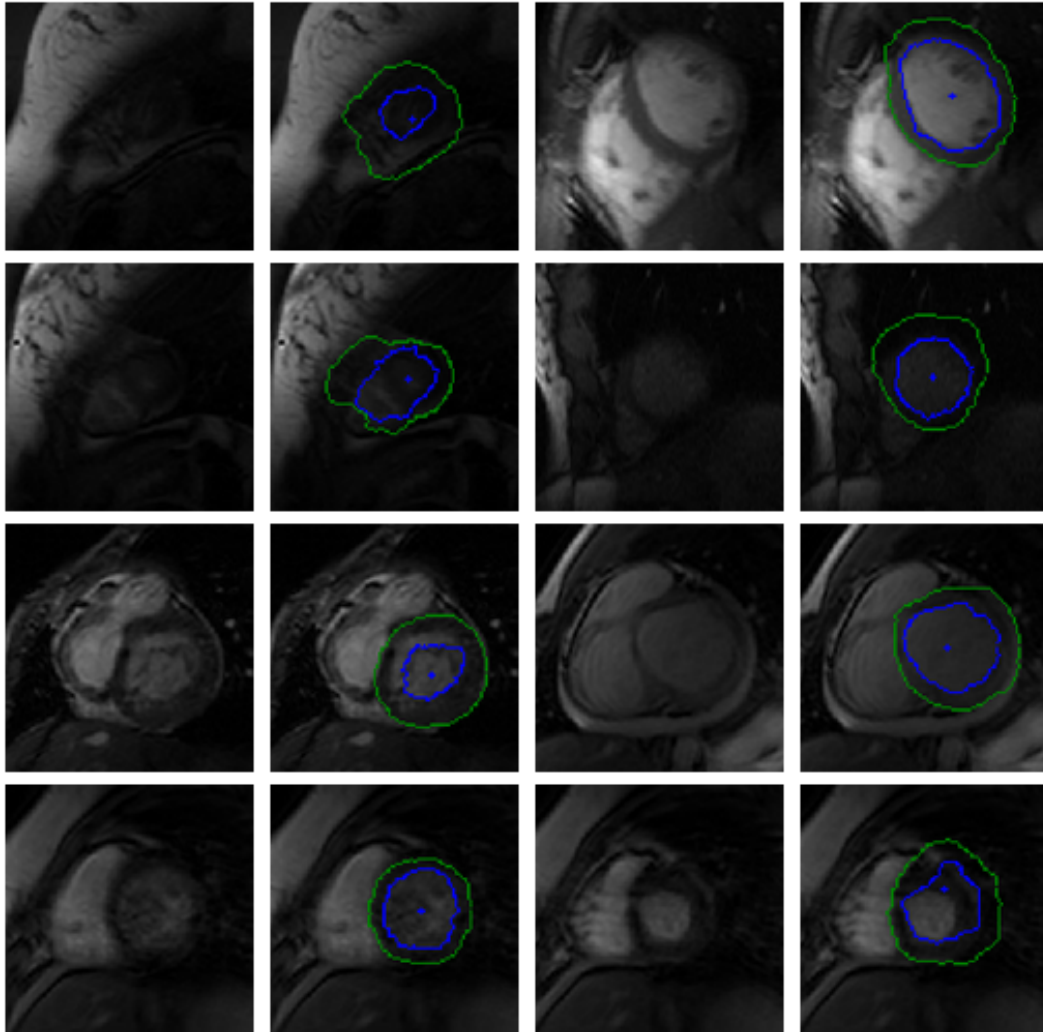


Figure 6.2: A selection of images that are incorrectly segmented by the system.

Figure 6.2 contains a selection of images that are incorrectly segmented. If the centre point is initialised outside the blood pool then the inner contour often tends to include the cardiac wall in addition to the blood pool. This causes the outer contour to only partially locate on the epicardium. A disappearing endocardium causes the inner contour to snap onto the epicardium and the outer contour to locate onto structures outside the heart.

Additionally, if the contrast between the blood pool and cardiac wall is very low, or there are strong (non-papillary) edges within the blood pool then the inner contour often snaps onto these incorrect strong edges. In our experience the outer contour remains largely unaffected in these situations.

6.1.2 Quantitative segmentation analysis

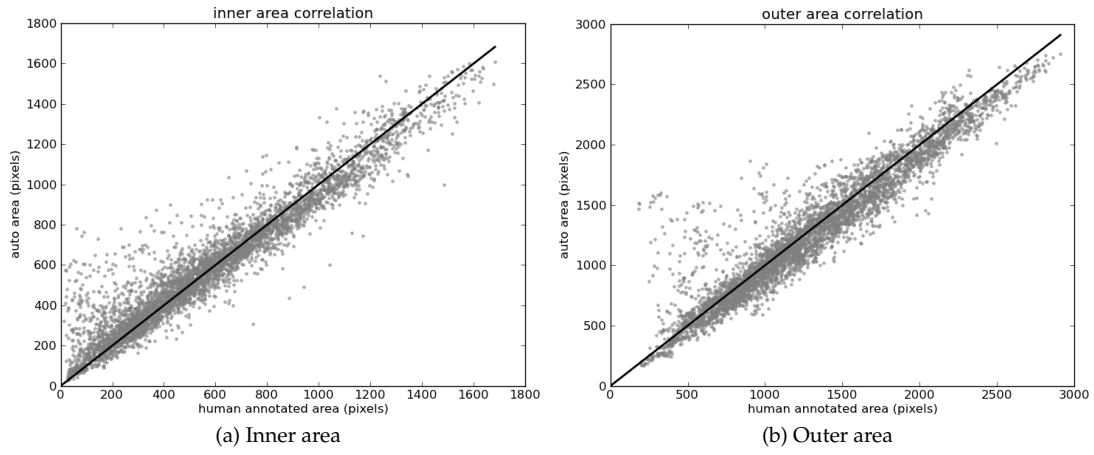


Figure 6.3: Areas inside contours of human annotation against the areas inside automated segmentation of testing data.

The areas inside the contours of the automatic annotations are plotted against the areas inside the human annotated ground truth in Figures 6.3a and 6.3b. For small inner contours our technique often yields segmentations larger than the ground truth. This can be attributed to the automated segmentations being more “inclusive” of the papillary muscles, which can significantly affect small contours. Our technique also provides slightly smaller outer contours. A comparison with the ground truth indicates that our segmentation is often temporally smoother. This is attributed to the human annotator segmenting one frame at a time, and thereby largely disregarding temporal behaviour.

The figures in 6.4 and 6.5 show frame Dice errors against time and slice position for the inner and outer contours. The vertical axes of these graphs are logarithmically scaled. The geometric means (arithmetic mean in the log-scale) in these figures are indicated by solid black lines. We observe that the majority of incorrect segmentations occur during end systole ($t \approx 8$) and spatially lower slices (depth > 9) where the endocardium is at its small-

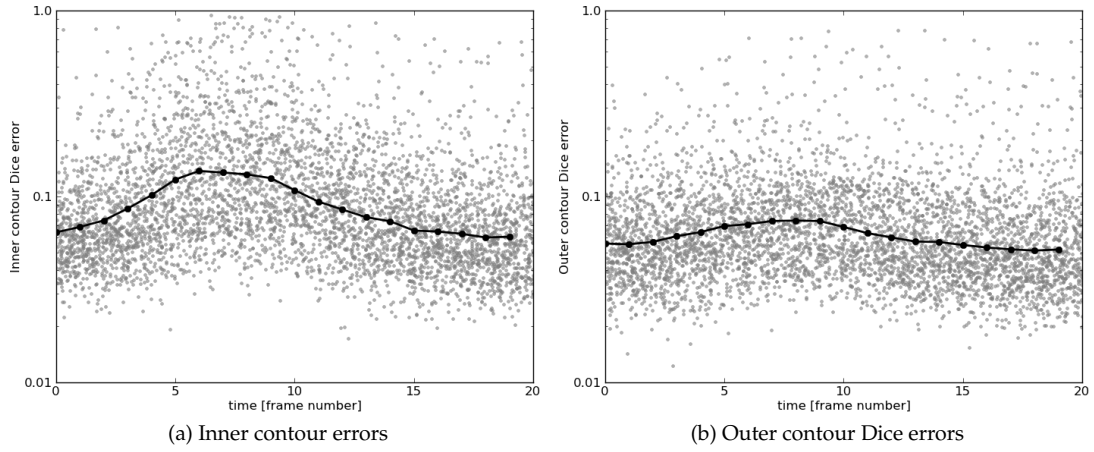


Figure 6.4: Contour Dice errors over time. For illustrative purposes, a random real value between zero and one was added to each frame number. The geometric mean for each frame number is indicated by a black line.

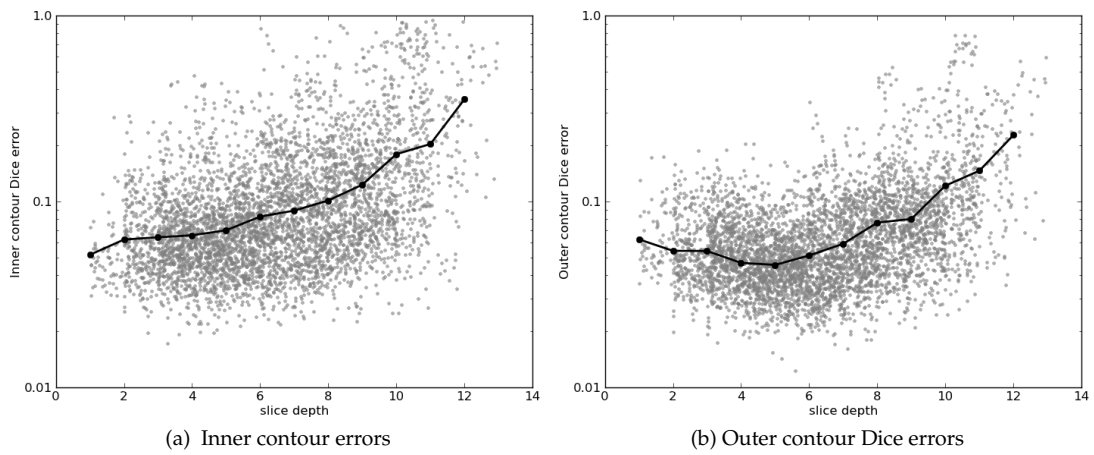


Figure 6.5: Contour Dice errors over different slices. For illustrative purposes, a random real value between zero and one was added to each slice depth. The geometric means for the slice positions are indicated by the black line.

est (sometimes completely disappearing from view). In these frames papillary muscles are most visible, obscuring not only the border, but a significant part of the endocardium.

Authors	Technique	Inner contour error [mm]	Outer contour error [mm]
Our method	CRF	1.57	1.78
Our method (without subjects 8 and 27)	CRF	1.49	1.74
Andreopoulos and Tsotsos [3]	AAM	1.43	1.51
Üzümcü [54]	Landmark tracking	1.86	1.77
Jolly [23]	Shortest Path	2.44	2.05
Cordero-Grande et al. [11]	Edge MRF	1.37	1.22
Lorenzo-Valdés et al. [32]	Surface MRF	2.99	2.21

Table 6.1: Segmentation errors as reported by different authors. These results are, however, not strictly comparable since they are based on different datasets and the error criteria differ.

Table 6.1 contains segmentation errors of the inner and outer contours as reported by a selection of different authors. These results are from different datasets and there are also subtle, but important, differences in the definition of the error criteria. Refer to the individual papers for more detail. The results are therefore not strictly comparable. In this table we report our results as the landmark error [3], i.e. the average of the shortest distances between the points on the ground truth contour and the inferred contour. For a more comprehensive study of reported errors see [44].

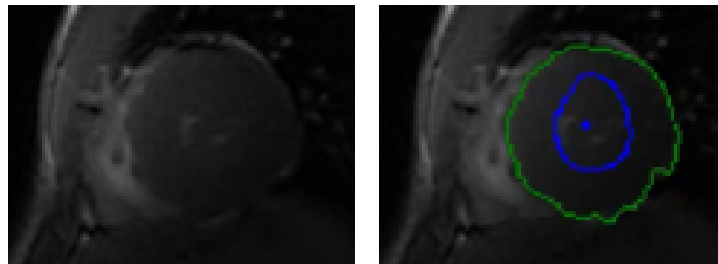


Figure 6.6: Incorrect automatically segmented contours (inner contour is blue and outer is green) of Subject 8 due to a disappearing endocardium. The blue dot in the middle of the endocardium is the estimated centre point.

On further inspection of the York dataset, the endocardium disappears from view in some video sequences of Subject 8 (refer to Figure 6.6) who is diagnosed with ventricular hypertrophy (enlarged cardiac wall thickness). It is also noted by the annotator of the

dataset [3] that this yields bad segmentations in their work. Images of Subject 27 also have relatively low contrast between the endocardium and the cardiac wall. We therefore regard Subjects 8 and 27 as outliers and remove them from the dataset. This significantly improves inner contour accuracy as indicated in Table 6.1.

6.1.3 Sensitivity to initial centre point placement

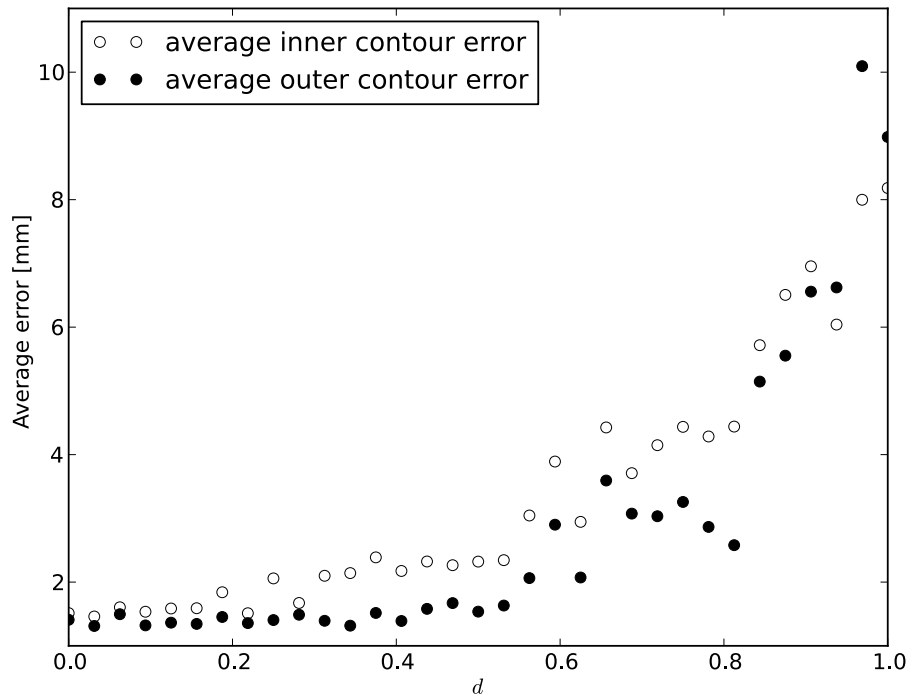


Figure 6.7: Sensitivity of segmentation with regard to incorrect centre point in first frame. The value d is the fractional distance between the ground truth centre point and the ground truth inner contour. See text for more detail.

Figure 6.7 illustrates the sensitivity of the segmentation to incorrect placement of the initial centre point, $c(0)$. For each video sequence, an initial centre point is generated at a fractional distance, d , between the ground truth centre and a randomly selected point on the ground truth inner contour. The CRF parameters are not retrained on these imperfectly placed centre points, which would possibly allow the system to weigh shape information less and thus improve results.

As can be seen from Figure 6.7, the segmentations of the inner and outer contours remain relatively stable if $c(0)$ is within the inner 20% of the endocardium. When placed at approximately 50% between the ground truth centre point and inner contour, the spatial continuity assumption of (3.12) is violated enough that the quality of the inferred contours begin to deteriorate significantly.

6.2 Sunnybrook cardiac segmentation dataset

Machine Learning algorithms run the risk of specialising on properties that do not generalise well to new datasets especially when the data is collected under conditions different from the training set. This can be a challenge when applied to real-world conditions, often requiring the user to normalise images, to remove variation, and risking degradation of image information that could be important for segmentation. It is therefore important to evaluate whether techniques are adaptable with minimal effort when data from a different source is given.

The Sunnybrook Cardiac MR Database [46] is provided by the Sunnybrook Health Sciences Centre and was used for the 2009 MICCAI Cardiac MR Left Ventricle Segmentation Challenge. The dataset contains 45 subjects, with an average age of 61, with diverse morphologies (heart failure with and without infarction, LV hypertrophy, and healthy subjects) and is manually segmented by a cardiologist. The inner contours are annotated only at end diastole and end systole, while the outer contours are annotated only at end systole. A ground truth segmentation of the intermediate frames is therefore not available, making extraction of temporal information for this dataset difficult.

Due to the sparsity of annotations in this dataset, all feature functions are re-used as derived from the York dataset. The model used to segment the Sunnybrook data therefore includes features derived from the trained edge classifiers, temporal behaviour and inner-outer relationships of the York dataset. Only the CRF parameters (i.e. the relative importance of the features) are retrained on a training subset of the Sunnybrook data. To further compensate for the relatively few examples, and thus avoid over-fitting, a very weakly weighted L1-norm parameter regularisation term ($\sum_d |\log \theta_d|$) is added to the objective function. L1 regularisation is often used to assign zero to a large number of param-

ters. However, since this term is applied to the log of the parameters, it effectively penalises the specialisation on features by the optimiser, tending to weigh them equally. A detailed analysis of the effects of parameter regularisation in our application is beyond the scope of this dissertation.

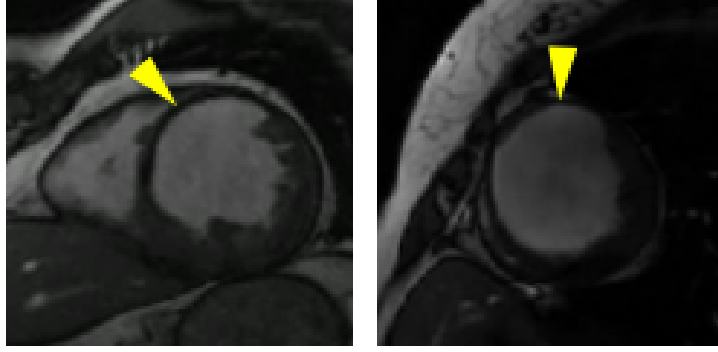


Figure 6.8: Examples from the Sunnybrook dataset with thin cardiac walls, as indicated by the yellow arrows.

A notable difference from the York dataset is the presence of images of patients with heart failure, where the cardiac wall is exceptionally thin, as indicated by the yellow arrows in Figure 6.8. To compensate we modified the belief propagation algorithm by reducing the minimum allowed difference (see Section 4.1) between the inner and outer variables (i.e. wall thickness) from $\epsilon_M = 10$ to $\epsilon_M = 2$. We also reduced the radial offset used in the features described by (3.15) and (3.16) from $\epsilon_\rho = 2$ pixels to $\epsilon_\rho = 1$, to adequately capture the wall colour when very thin.

This modification suggests that there are important model parameters that are dependent on the patient pathology. A practical segmentation tool could allow the operator the option to provide a prior diagnosis or more fine grained control over some settings.

During training of the edge classifier from the York dataset, it was assumed that the extracted radial window, v_ρ , contains an edge if the annotated edge is within two radial distances from the middle of the window. This assumption is problematic if used to train the edge classifier on the Sunnybrook dataset. A small wall thickness is common in this dataset and would cause both the inner and outer contours to fall within this radial distance. This would result in inconsistencies in the training examples and thus weaken the resulting edge detector. To train a classifier on this dataset it would thus be necessary to be

more strict with regard to the minimum radial distance.

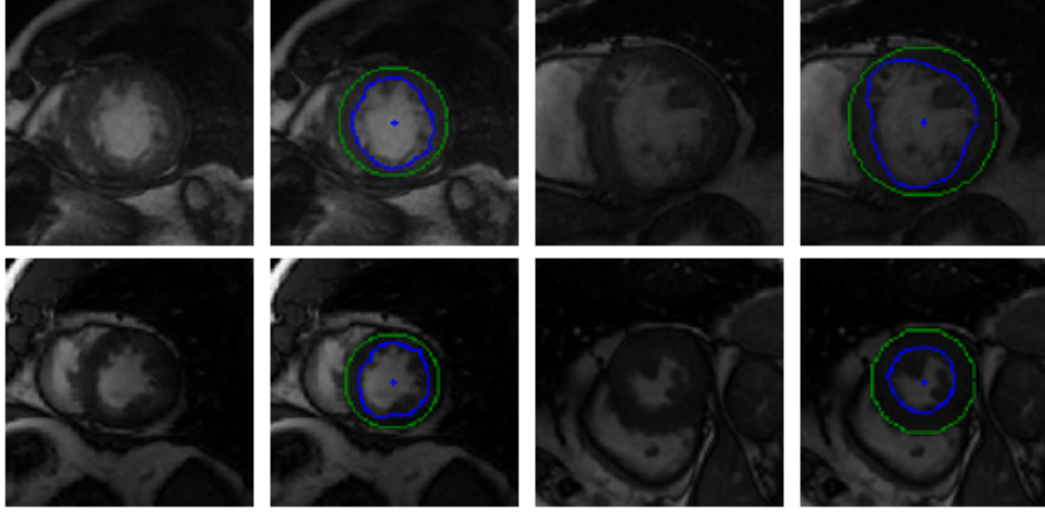


Figure 6.9: Selection of images and their automatically segmented contours (inner contour is blue and outer is green). The blue dot in the middle of the endocardium is the estimated centre point.

Figure 6.9 contains a selection of images from this dataset and their automatic segmentations. A qualitative examination of the results suggests that segmentations are generally of good quality, i.e., the papillary muscles are included even if they obscure the endocardium border.

Bland-Altman plots of end-diastole volume, end-systole volume, ejection fraction and mass are shown in Figure 6.10. The end-diastole volumes, as predicted by our technique and as annotated by the human specialist, agree with a small bias and variance (-3.35 ± 7.61 ml). End-systole volumes agree with a small bias, but a relatively large variance (1.75 ± 20.21 ml). This leads to a relatively small bias, but relatively large variance in the agreement of the calculated ejection fractions (-4.66 ± 10.73 %). The calculated left ventricle mass has a small bias and variance (-0.95 ± 11.58 %).

The end-diastole volume in Figure 6.10b and Bland-Altman plot of end-diastole inner contour area in Figure 6.10f also illustrate the algorithm’s tendency to yield contours smaller than the ground truth at end-diastole. The Bland-Altman plot of end-systolic volume in Figure 6.10a and end-systolic inner contour area in Figure 6.10e illustrate that small volumes are overestimated and large volumes underestimated.

Table 6.2 contains a summary of Dice similarities and APD as reported by various au-

Authors	Dice similarity		APD (mm)	
	inner	outer	inner	outer
Our method (trained on York)	0.87	0.92	2.70	2.23
Our method (after retraining)	0.91	0.93	1.84	1.95
Marak et al. [34]	0.86	0.93	2.6	3.0
Lu et al. [33]	0.89	0.94	2.07	1.91
Wijnhout et al. [59]	0.89	0.93	2.29	2.28
Casta et al. [8]	-	0.93	-	2.72
O'Brien et al. [41]	0.81	0.91	3.73	3.16
Constantinides et al. [9]	0.89	0.92	2.35	2.04
Huang S. et al. [20]	0.89	0.94	2.10	1.95
Jolly [23]	0.88	0.93	2.44	2.05

Table 6.2: Average Dice similarity metric and Average Perpendicular Distance (APD) for our segmentation of the Sunnybrook validation set (before and after training) and results reported by the different MICCAI challenge entries.

thors on the Sunnybrook dataset during the MICCAI challenge [44, 46] including our results, before and after parameter retraining. Prior to retraining (i.e. parameters as derived from the York dataset), results are comparable but slightly worse than the top performing challenge entries. After parameter re-estimation on the Sunnybrook training subset our results are superior to the entries on the validation set in terms of the inner contours' Dice metric. Our average Dice similarity of the outer contours is comparable to the best performing entries in the challenge.

Table 6.2 also contains the Average Perpendicular Distance (APD) of various authors. The APD is calculated as the average of the perpendicular distance from each point on the reference contour to the target contour. The APD is therefore similar to the non-symmetric landmark distance. Our average APD for the inner and outer contours are equal to or smaller than reported by any of the authors in the challenge.

Table 6.3 provides a more detailed report on the resulting segmentations of the patients in the validation set, as generated by the evaluation code provided with the dataset. The table also indicates the percentage of good contours for each subject, i.e. those with an APD smaller than 5mm.

Our segmentation process performs well on those subjects in the validation set with normal heart function (SC-N) and those with heart failure with (SC-HF-I) and without infarction (SC-HF-NI). Our process performs worst on the inner contour of patients in the validation set with hypertrophy (SC-HYP) as is illustrated in Figure 6.11. This effect is also

Patient	Good (%)		APD (mm)		Dice similarity	
	inner	outer	inner	outer	inner	outer
SC-HF-I-05	100	100	1.52	1.86	0.94	0.95
SC-HF-I-06	100	100	1.66	1.45	0.92	0.95
SC-HF-I-07	100	100	2.37	2.79	0.89	0.90
SC-HF-I-08	95	100	1.68	1.37	0.93	0.96
SC-HF-NI-07	100	100	2.21	2.20	0.91	0.93
SC-HF-NI-11	100	100	1.70	1.25	0.93	0.96
SC-HF-NI-31	100	100	2.06	1.58	0.91	0.95
SC-HF-NI-33	94	100	1.68	1.64	0.91	0.94
SC-HYP-06	92	100	1.67	2.05	0.90	0.92
SC-HYP-07	69	100	1.39	1.83	0.93	0.94
SC-HYP-08	68	100	2.27	2.33	0.90	0.93
SC-HYP-37	69	71	2.21	2.44	0.86	0.91
SC-N-05	93	100	1.67	2.45	0.89	0.89
SC-N-06	100	86	1.76	2.12	0.89	0.91
SC-N-07	100	100	1.79	1.95	0.88	0.90
mean	92	97	1.84	1.95	0.91	0.93
std deviation	12.4	8.0	0.30	0.44	0.02	0.02

Table 6.3: Patient specific Average Perpendicular Distance (APD), and Dice similarity between annotations and ground truth of the Sunnybrook validation set.

observed in other models by different authors [44].

6.3 Summary

Through analysis of results on the York and Sunnybrook datasets we conclude that our segmentation process compares well with, or outperforms many existing techniques in terms of accuracy on these datasets. We also conclude that our segmentation process generalises well to new datasets and can be easily retrained for increased accuracy.

Our segmentation process relies, however, on accurate placement of the initial centre point. Furthermore, segmentation for subjects with ventricular hypertrophy, where the endocardium can completely disappear from view during end-systole, is found to be difficult.

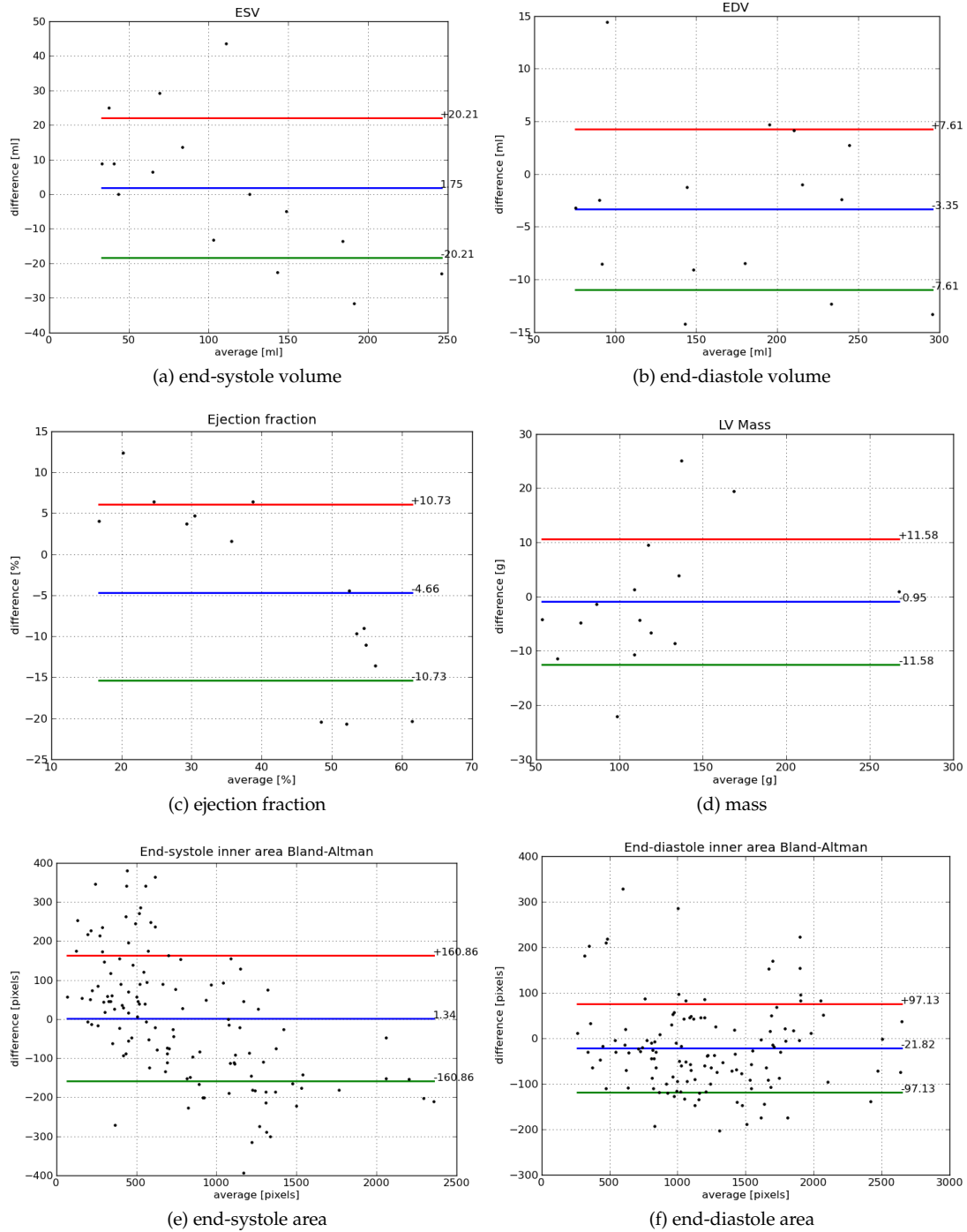


Figure 6.10: Bland-Altman plots of end-systolic volume, end-diastolic volume, ejection fraction, mass, end-systolic area and end-diastolic area. Each dot represents a video sequence in the validation set. The vertical axes are of *automatically determined* minus *ground truth* values. Mean difference and standard deviation lines are included.

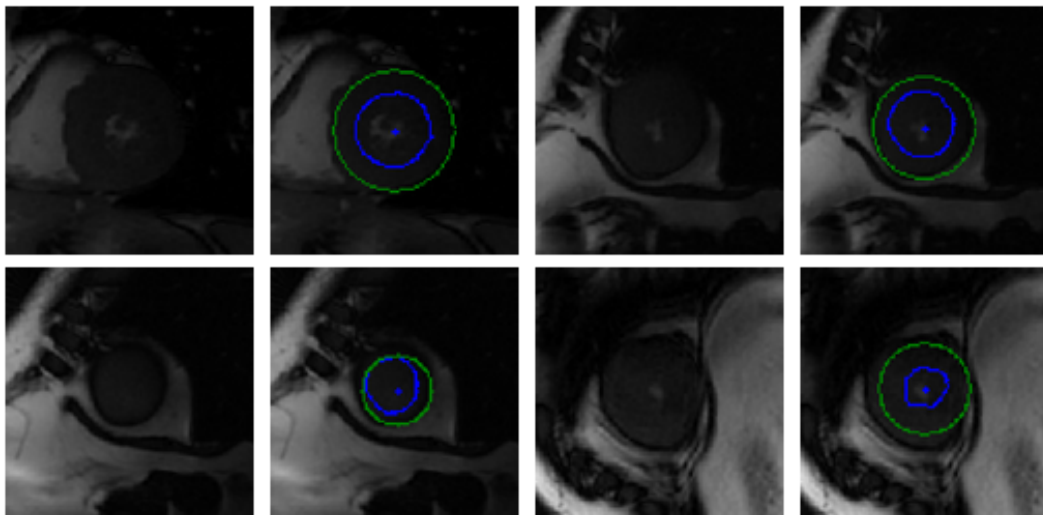


Figure 6.11: Selection of images from validation patient images with hypertrophy and their low quality automatically segmented contours (inner contour is blue and outer is green). The blue dot in the middle of the endocardium is the estimated centre point.

Chapter 7

Concluding remarks

The segmentation process, as outlined in this dissertation, provides an approach that mitigates the effects of papillary muscles and performs well on a variety of images. We achieve this by integrating features that combine edge, shape and temporal information derived from an annotated dataset into our model. We develop a strategy to tractably derive suitable estimates of the model parameters. A strategy to efficiently infer segmentations from this model is also presented. To our knowledge, this is the first time that either CRFs or belief propagation have been applied to this problem.

Our evaluation is, however, incomplete without discussing some of the limitations of the model and changes that can be made to improve segmentation accuracy.

7.1 Model limitations

Many of the current limitations in the model stem from the manner in which the segmentations are parametrised. Recall that the centre of the endocardium is used as a point of reference for both the inner and outer contour radii. The outer contour is relatively stationary during the cardiac cycle, while the inner contours can undergo significant translation. Using the translating centre point of the endocardium as a reference point for the outer contour, its radial values used to describe the contour can change significantly even though the contour remains stationary. This makes the feature function that describes the relationship of radial values of the outer contour over time, less effective. Using different centre points

for the inner and outer contours is possible, but complicates the calculation of features relating the behaviour of the inner and outer contours and cardiac wall.

The current model also assumes that the inner contour is visible in every image. The endocardium can, however, disappear from view in images in the lower slices of patients with ventricular hypertrophy (enlarged cardiac wall thickness). This causes the segmentation process to fit the inner contour to the nearest edges, which often corresponds to the outer contour, and forces the inferred outer contour to fit to non-cardiac structure edges. This can have a significant effect on the segmentation quality.

7.2 Future work

No image preprocessing is done to compensate for effects such as different intensity settings on the MRI equipment. Image equalisation could be used to compensate for these effects and would likely improve segmentation results.

Ideally, the time of end-systole in (3.14) should not be specified before inference; however, this might require a second order CRF or modelling t_{ES} as an additional unobserved variable. Alternatively, techniques based on detecting temporary suspension and reversal of optical flow in the images could be applied.

A second order system would also make the incorporation of contour smoothness information possible, as currently only contour continuity is taken into account. Alternatively, post-processing of the resultant contours by fitting them to splines, would improve contour smoothness.

Although there are no conceptual difficulties in using higher order systems, the increase in computational cost is, however, significant, both in terms of training and inference.

Future investigations might also concentrate on improving the centre point estimation via better features and improving runtime performance. As mentioned previously, formulating the centre point estimation as a model that can be solved with dynamic programming allows us to cast its optimisation as a belief propagation algorithm. This has the advantage that the centre point can be estimated as an additional latent variable in our CRF model, that needs to be inferred.

The development of a fast heuristic to estimate the centre points is also possible. Since

the blood pool disappears only once in each cardiac video sequence, we speculate that results from an online “greedy” tracking of the centre point from both the start and end of the sequences towards the middle frames, could be combined to produce a sufficiently accurate, yet fast, procedure.

Fusing the different spatial slices into a unified 3D and time model might also increase segmentation accuracy. Information at spatially higher slices, where the papillary muscles are less problematic, could then be used to improve the accuracy at lower slices. Linking the radial values between different slices would be relatively simple. This can be done with appropriate feature functions similar to those restricting radial continuity in a single contour. The main challenge is perhaps the alignment of the spatial slices to compensate for translation caused by different levels of inhalation and expiration between acquisition of the slices.

Modifying the results of the belief propagation process to address user interaction requirements, specifically responsiveness, is discussed in Appendix B.

7.3 Summary

In this dissertation we present a model for the automated segmentation of the inner and outer contours of the left ventricle. Features are derived from the discriminative information provided by a human annotated dataset. Robustness against the inclusion of the papillary muscles is obtained by integrating shape and motion information from all the frames in the video sequence.

To compensate for the approximate nature of inference and avoid calculating the partition function, a gradient-free approach is followed where the segmentation error is treated as a “black box”. This allows us to integrate the approximate inference process into the training stage.

It should be emphasised that the system is able to integrate temporal information available in the total video sequence and not only individual frames. Since this information is not readily available to humans, automated systems such as this might yield segmentations that are superior to that which are attainable by a human annotator that considers only one frame at a time, with only a rough consideration of previous frames.

By using our presented process, the segmentation of a cardiac MRI video sequence can thus be completed within a short time with minimal user interaction, i.e. only a single initial centre point is required for each cardiac cycle.

Bibliography

- [1] Srinivas M. Aji and Robert J. McEliece. The generalized distributive law. *Information Theory, IEEE Transactions on*, 46(2):325–343, March 2000.
- [2] Jeffrey L. Anderson, Aaron N. Weaver, Benjamin D. Horne, Heath U. Jones, Gerri K. Jelaco, Julie A. Cha, Hector E. Busto, Judy Hall, Kathy Walker, and Duane D. Blatter. Normal cardiac magnetic resonance measurements and interobserver discrepancies in volumes and mass using the papillary muscle inclusion method. *The Open General and Internal Medicine Journal*, 1:6–12, 2007.
- [3] Alexander Andreopoulos and John K. Tsotsos. Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI. *Medical Image Analysis*, 12(3):335–357, 2008.
- [4] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.
- [5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006.
- [6] Richard P. Brent. *Algorithms for Minimization without Derivatives*, chapter 4. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [7] M.D. Burkhard Sievers, Simon Kirchberg, Asli Bakan, M.D. Ulrich Franken, and M.D. Hans-Joachim Trappe. Impact of papillary muscles in ventricular volume and ejection fraction assessment by cardiovascular magnetic resonance. *Journal of Cardiovascular Magnetic Resonance*, 6(1):9–16, 2004.

- [8] 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. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009.
- [9] C. Constantinides, Y. Chenoune, N. Kachenoura, E. Roullot, E. Mousseaux, A. Herment, and F. Frouin. Semi-automated cardiac segmentation on cine magnetic resonance images using GVF-snake deformable models. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009.
- [10] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [11] Lucilio Cordero-Grande, Gonzalo Vegas-Sánchez-Ferrero, Pablo Casaseca de-la Higuera, J. Alberto San-Román-Calvar, Ana Revilla-Orodea, Marcos Martín-Fernández, and Carlos Alberola-López. Unsupervised 4D myocardium segmentation with a Markov random field based deformable model. *Medical Image Analysis*, 15(3):283–301, 2011.
- [12] L. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945.
- [13] Janto Dreijer, Johan du Preez, and Ben Herbst. Edge modelling MRFs for cardiac MRI segmentation. *Pattern Recognition Association of South Africa*, (21):81–86, 2010.
- [14] Janto Dreijer, Ben Herbst, and Johan du Preez. Left ventricular segmentation from MRI datasets with edge modelling conditional random fields. *BMC Medical Imaging*, 13(1):24, 2013.
- [15] Vojtech Franc and Bogdan Savchynskyy. Discriminative learning of max-sum classifiers. *Journal of Machine Learning Research*, 9:67–104, 2008.
- [16] A.F. Frangi, W.J. Niessen, and M.A. Viergever. Three-dimensional modeling for functional analysis of cardiac images, a review. *Medical Imaging, IEEE Transactions on*, 20(1):2–5, January 2001.
- [17] J. Goldberger and H. Kfir. Serial schedules for belief-propagation: Analysis of convergence time. *Information Theory, IEEE Transactions on*, 54(3):1316–1319, March 2008.

- [18] Einar Heiberg. *Automated Feature Detection in Multidimensional Images*. PhD thesis, Linköping University, December 2004.
- [19] Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas. A graphical model framework for coupling MRFs and deformable models. In *Proceedings of CVPR*, pages 739–746, 2004.
- [20] S. Huang, J. Liu, L. C. Lee, S. K. Venkatesh, L. L. S. Teo, C. Au, and W. L. Nowinski. Segmentation of the left ventricle from cine MR images using a comprehensive approach. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, August 2009.
- [21] Matthew Janik, Matthew D Cham, Michael I Ross, Yi Wang, Noel Codella, James K Min, Martin R Prince, Shant Manoushagian, Peter M Okin, Richard B Devereux, and Jonathan W Weinsaft. Effects of papillary muscles and trabeculae on left ventricular quantification: increased impact of methodological variability in patients with left ventricular hypertrophy. *Journal of Hypertension*, 26(8):1677–1685, August 2008.
- [22] Edwin Thompson Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.
- [23] M.P. 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*, 2009.
- [24] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.
- [25] Michael I. Jordan. Graphical models. *Statistical Science*, 19:140–155, 2004.
- [26] R.E. Kalman and Rudolph Emil. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [27] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [28] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, February 2001.

- [29] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.
- [30] A.S. Lewis and M.L. Overton. Nonsmooth optimization via BFGS. Technical report, Courant Institute of Mathematical Sciences, New York University, 2008. Submitted to SIAM J. Optimization.
- [31] Stan Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [32] Maria Lorenzo-Valdés, Gerardo I. Sanchez-Ortiz, Andrew G. Elkington, Raad H. Mohiaddin, and Daniel Rueckert. Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm. *Medical Image Analysis*, 8(3):255–265, 2004.
- [33] Y. Lu, P. Radau, K. Connelly, A. Dick, and G. Wright. Automatic image-driven segmentation of left ventricle in cardiac cine MRI. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009.
- [34] L. Marak, J. Cousty, L. Najman, and H. Talbot. 4D morphological segmentation and the miccai LV-segmentation grand challenge. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009.
- [35] Victorin Martin, Jean-Marc Lasgouttes, and Cyril Furtlehner. The role of normalization in the belief propagation algorithm. *ArXiv*, January 2011.
- [36] Tim Mcinerney and Demetri Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1:91–108, 1996.
- [37] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- [38] L. Najman, J. Cousty, M. Couprie, H. Talbot, S. Clément-Guinaudeau, T. Goissen, and J. Garot. An open, clinically-validated database of 3D+t cine-MR images of the left ventricle with associated manual and automated segmentation. <http://laurentnajman.org/heart>, June 2007.
- [39] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

- [40] Albert Oberman, Allen R. Myers, Thomas M. Karunas, and Frederick H. Epstein. Heart size of adults in a natural population - Tecumseh, Michigan. *Circulation*, 35(4):724–733, 1967.
- [41] S. O’Brien, O. Ghita, and P. Whelan. Segmenting the left ventricle in 3D using a coupled asm and a learned non-rigid spatial model. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009.
- [42] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA, 1982.
- [43] Judea Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.
- [44] Caroline Petitjean and Jean-Nicolas Dacher. A review of segmentation methods in short axis cardiac MR images. *Medical Image Analysis*, 15(2):169–184, 2011.
- [45] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964.
- [46] P. Radau, Y. Lu, K. Connelly, G. Paul, A. Dick, and G. Wright. Evaluation framework for algorithms segmenting short axis cardiac MRI. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009. Sunnybrook Hospital.
- [47] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. Accepted for publication in *Journal of Global Optimization*.
- [48] Mendis S., Puska P., and Norrving B., editors. *Global Atlas on Cardiovascular Disease Prevention and Control*. World Health Organization, Geneva, 2011.
- [49] Prediman K. Shah, Jamshid Maddahi, Howard M. Staniloff, A.Gray Ellrodt, Max Pichler, H.J.C. Swan, and Daniel S. Berman. Variable spectrum and prognostic implications of left and right ventricular ejection fractions in patients with and without clinical heart failure after acute myocardial infarction. *The American Journal of Cardiology*, 58(6):387–393, 1986.

- [50] Anders Skajaa. Limited memory BFGS for nonsmooth optimization. Master's thesis, Courant Institute of Mathematical Science, New York University, January 2010.
- [51] Charles Sutton and Andrew McCallum. *Introduction to Statistical Relational Learning, An Introduction to Conditional Random Fields for Relational Learning*, chapter 4. MIT Press, 2007.
- [52] Charles Sutton and Andrew McCallum. Piecewise pseudolikelihood for efficient CRF training. In *International Conference on Machine Learning*, pages 863–870. ACM Press, 2007.
- [53] Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields. *ArXiv*, November 2010.
- [54] Mehmet Üzümcü. *Constrained segmentation of cardiac MR image sequences*. PhD thesis, Leiden University, 2007.
- [55] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, April 1967.
- [56] J Vogel-Claussen, JP Finn, AS Gomes, GW Hundley, M Jerosch-Herold, G Pearson, S Sinha, JA Lima, and DA Bluemke. Left ventricular papillary muscle mass: relationship to left ventricular mass and volumes by magnetic resonance imaging. *Journal of Computer Assisted Tomography*, 30(3), May 2006.
- [57] Martin J. Wainwright and Max Chickering. Estimating the "wrong" graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research*, 2006.
- [58] Yair Weiss and William T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *Information Theory, IEEE Transactions on*, 47(2):723–735, 2001.
- [59] J. Wijnhout, D. Hendriksen, H. Van Assen, and R. Van der Geest. LV challenge LKEB contribution: Fully automated myocardial contour detection. *The MIDAS Journal - Cardiac MR Left Ventricle Segmentation Challenge*, July 2009.
- [60] WI Zangwill. Minimizing a function without calculating derivatives. *The Computer Journal*, 10(3):293–296, 1967.

Appendices

Appendix A

Edge properties

Analysis of the behaviour of the edges aids in an intuitive understanding of the model's behaviour. This is unfortunately difficult due to the complexity of multi-layer neural networks. To gain some insight into the nature of the edges we therefore briefly investigate a logistic regression classifier (see e.g. [5]) trained on the same data. One of the advantages of using a logistic regression classifier is that it allows us to analyse the resultant coefficients for an intuitive interpretation of its decision boundary. It should be emphasised that we are not attempting to derive hints towards the behaviour of the trained neural network, but are investigating properties of the inner and outer edge classes.

Similar to the neural network in Section 3.4.1 we train logistic regression classifiers based on the probability of the presence of the cardiac edge, e , given the image values in a window, v_ρ ,

$$P(e | \kappa(v_\rho), \rho) = \frac{1}{1 + \exp(-\beta \cdot \kappa(v_\rho))}. \quad (\text{A.1})$$

Recall that the input feature vector (the transformation of the image window values) is

$$\kappa(v) = \left(\frac{\partial v}{\partial \rho}, \left| \frac{\partial v}{\partial \rho} \right|, \text{sign} \left(\frac{\partial v}{\partial \rho} \right), \left[\left| \frac{\partial v}{\partial \rho} \right| > \epsilon \right] \right). \quad (\text{A.2})$$

Again, we train different edge classifiers for eight different angles,

$$P_{[8n/N]}^{\text{out}}(e | \kappa(v_{\rho_n}), \rho_n), \quad (\text{A.3})$$

and thus allow the classifiers to identify features that it might find relevant for that direction.

Figures A.1a and A.1b illustrate the coefficients for the inner and outer classifiers of $P(e | \kappa(v_{\rho_n}), \rho_n)$ for different angles. Specifically, the coefficients for each direction-dependent classifier are represented by a grey line. The average for each coefficient over the eight classifiers in all the directions is represented by a black line. The four coefficient groups (0 . . . 64, 64 . . . 128, 128 . . . 192 and 192 . . . 256) each corresponds to the four different gradient feature types in (A.2).

By observing a coefficient's magnitude and sign differences over the angular directions, we deduce the edge classifier's behaviour on different parts of the cardiac structure as they fall within a classifier's window.

It is important to note that the features are not whitened (scaled by their standard deviations) before being used to train the coefficients. This makes comparison between coefficients from different groups within a classifier deceptive as the scales of the types are inherently different. It is still, however, possible to compare the coefficients from the same group between different classifiers.

Consider the coefficients for the inner classifier in Figure A.1a. As expected, the coefficient magnitudes are the largest at positions corresponding to the centre of each window (at 32, 96, 160, 224), since gradient information near the position being considered for an edge is more important than further away.

Coefficients 0 . . . 64 weight the contribution of the signed gradient $\frac{\partial v}{\partial \rho}$. We can see that for the inner edge classifier all directions have strongly negative coefficients in the middle of the window and thus favour pixel transitions from high to low intensity. This is in agreement with the transition from the often lighter blood pool to the darker cardiac wall. This effect is also pronounced when only the sign of the gradient is considered (coefficients 128 . . . 192, $\text{sign}\left(\frac{\partial v}{\partial \rho}\right)$). This also corresponds to the heuristically derived features that are used in [13].

When considering the same coefficients (0 . . . 64 and 128 . . . 192) for the outer edge detector in Figure A.1b we similarly observe a strong response around the centre, but with a much larger variability across the different angular directions, with some coefficients positive and others negative. This is consistent with the outer contour's edge transitioning

from the cardiac wall to the darker area outside of the heart on the right side, and from the cardiac wall to the lighter right ventricle's blood pool on the left side.

The outer classifier also generally responds more strongly than the inner classifier to high values of the unsigned gradient $\left| \frac{\partial v}{\partial \rho} \right|$ (coefficients 64...128), i.e. edges with a large magnitude. We speculate that this is due to the presence of papillary muscles weakening the discriminative power of the unsigned gradient features for the inner edges.

The inner and outer classifiers both favour the presence of an edge above its absence (coefficients 192...256, $\left[\left| \frac{\partial v}{\partial \rho} \right| > \epsilon \right]$), however, the outer classifier's high coefficients are more concentrated around the centre of the window. This could be due to inconsistencies in the ground truth annotations with regard to the degree of inclusion of papillary muscles, as the presence of an edge could easily be missed by a human annotator if the intensity difference is small.

A smaller window size would allow for a finer classifier; however, with the possible loss of contextual information the usefulness of this reduction is questionable. Note for example the part of the sub-window to coefficients 32...64 of the inner classifier. These coefficients are relatively high and thus important for discriminating the inner edges. On inspection these sub-window areas often contain outer edges, which are informative for the presence of an inner edge at the window centre.

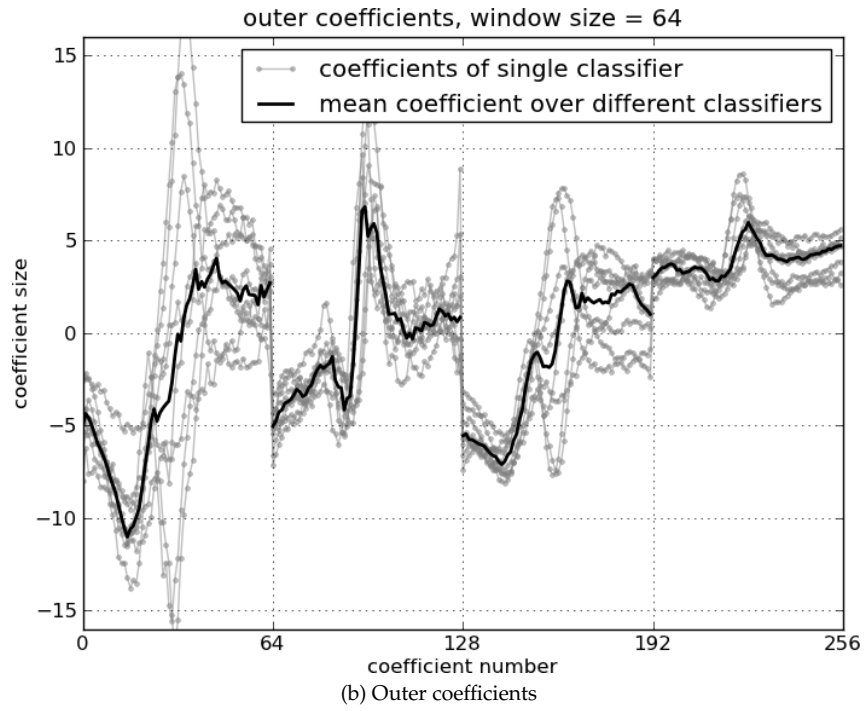
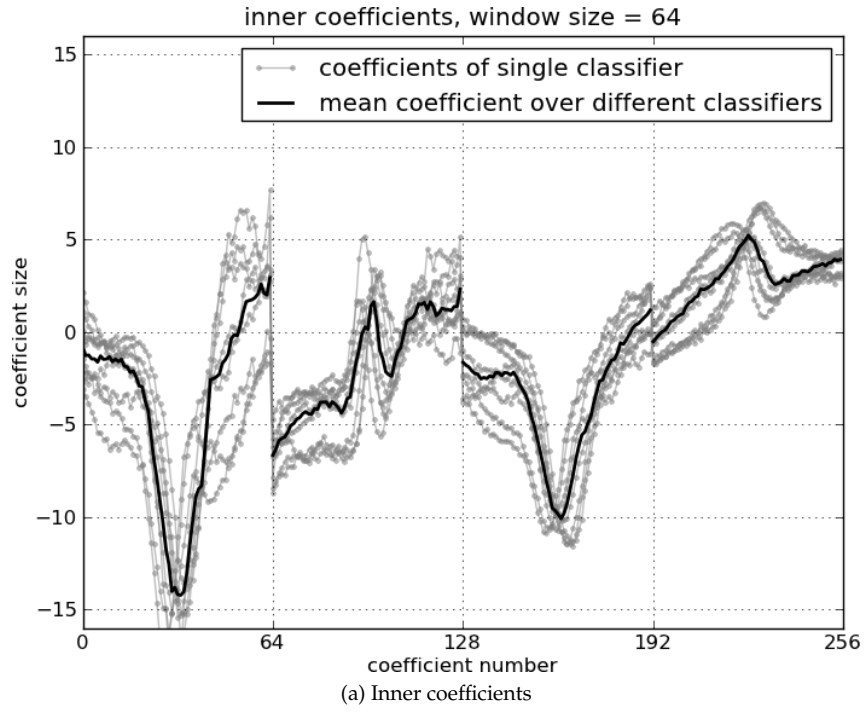


Figure A.1: Edge classifier coefficients for the inner and outer contours for different angles. Each of the eight direction-dependent classifiers is represented by a grey line. The average for each coefficient over the classifiers in all the directions is represented by the black line.

Appendix B

User interaction

To improve the usability as a software package it is desirable that the user can correct bad quality segmentations. We briefly describe a process that improves the contours in a video sequence after one or more contour points are provided by the user.

Assume that automatic segmentation has been completed and the user specifies a single point to which a contour must connect. The location of the point $\langle \hat{t}, \hat{x}, \hat{y} \rangle$ is first transformed into the log polar space as $\langle \hat{t}, \hat{n}, \hat{\rho} \rangle$.

To update the segmentation, belief propagation is repeated, but messages received by the variable $\rho_{\hat{n}}(\hat{t})$ are replaced by

$$m(\hat{t}, \hat{n}) = \begin{cases} 0 & \text{if } \rho = \hat{\rho} \\ K & \text{otherwise} \end{cases} \quad (\text{B.1})$$

where K is a large value. With a significantly large K , beliefs for incorrect assignments to this variable are thus penalised enough that they are unlikely to result during backtracking.

This process can be repeated for any number of points, however, propagation of messages through the graph takes a few seconds. This delay can significantly impact the user experience since typical user behaviour is to place a point, analyse the effects on segmentation and then to slightly modify the placed point. The time to belief propagation convergence can be reduced by initialising messages to their values before the point was added. We, however, wish to use a process that takes less than 100 milliseconds. This would al-

low points to be dragged with a mouse to better locations, while the rest of the contour is quickly updated.

To improve the user experience we recommend that message propagation be executed as a background process, while a temporary version of the contour being refined is shown to the user. To construct this temporary contour, after a single point is provided by the user, backtracking for this contour is repeated, but initiated from the user provided point. As backtracking is done first in the radial direction, this ensures that the rest of the contour is consistent with the radial factors. The backtracking is done on the messages before it is updated and is therefore fast enough to be used to smoothly update the contour while the user provided point is being dragged around.

Updating of the messages in the background is still necessary, since effect needs to be propagated to the other contours in the video sequence. There is therefore a delay between the addition of points by the user, however this effect is not as noticeable as an update while a point is being adjusted.

Appendix C

Additional 3D image

Figure C.1 shows a three dimensional reconstruction of an automatic segmentation at end-diastole. The different spatial slices were aligned to compensate for different breath hold levels by aligning the middle of the segmented outer contours. This example illustrates that a 3D animated model could be constructed from the automatically annotated images.

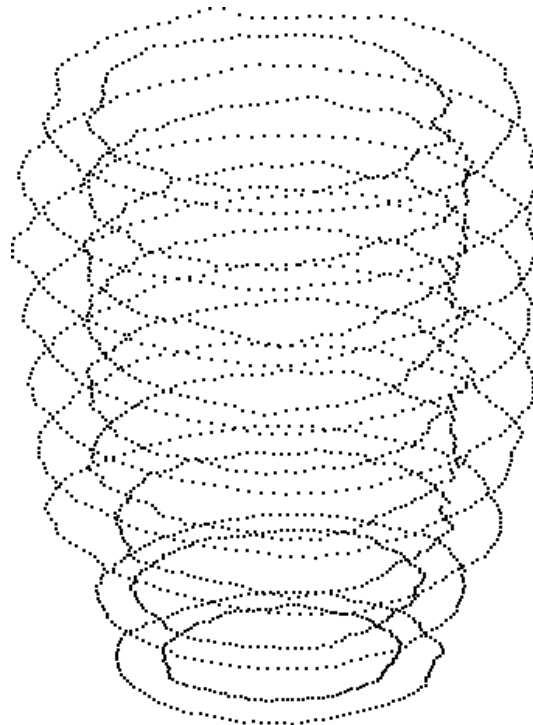


Figure C.1: Three dimensional construction of an automatic segmentation at end-diastole.