Analysis of 3D Echocardiography

Kiryl Chykeyuk

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2014

Abstract

Heart disease is the major cause of death in the developed world. Due to its fast, portable, low-cost and harmless way of imaging the heart, echocardiography has become the most frequent tool for diagnosis of cardiac function in clinical routine. However, visual assessment of heart function from echocardiography is challenging, highly operator-dependant and is subject to intra- and inter observer errors. Therefore, development of automated methods for echocardiography analysis is important towards accurate assessment of cardiac function. In this thesis we develop new ways to model echocardiography data using Bayesian machine learning methods and concern three problems: (i) wall motion analysis in 2D stress echocardiography, (ii) segmentation of the myocardium in 3D echocardiography, and (iii) standard views extraction from 3D echocardiography.

Firstly, we propose and compare four discriminative methods for feature extraction and wall motion classification of 2D stress echocardiography (images of the heart taken at rest and after exercise or pharmalogical stress). The four methods are based on (i) Support Vector Machines, (ii) Relevance Vector Machines, (iii) Lasso algorithm and Regularised Least Squares, (iv) Elastic Net regularisation and Regularised Least Squares. Although all the methods are shown to have superior performance to the state-of-the-art, one conclusion is that good segmentation of the myocardium in echocardiography is key for accurate assessment of cardiac wall motion.

We investigate the application of one of the most promising current machine learning techniques, called Decision Random Forests, to segment the myocardium from 3D echocardiograms. We demonstrate that more reliable and ultrasound specific descriptors are needed in order to achieve the best results.

Specifically, we introduce two sets of new features to improve the segmentation results: (i) *LoCo* and *GloCo* features with a local and a global shape constraint on coupled endo- and epicardial boundaries, and (ii) FA features, which use the Feature Asymmetry measure to highlight step-like edges in echocardiographic images. We also reinforce the traditional features such as Haar and Rectangular features by aligning 3D echocardiograms. For that we develop a new registration technique, which is based on aligning centre lines of the left ventricles. We show that with alignment performance is boosted by approximately 15%.

Finally, a novel approach to detect planes in 3D images using regression voting is proposed. To the best of our knowledge we are the first to use a one-step regression approach

for the task of plane detection in 3D images. We investigate the application to standard views extraction from 3D echocardiography to facilitate efficient clinical inspection of cardiac abnormalities and diseases. We further develop a new method, called the Class-Specific Regression Forest, where class label information is incorporating into the training phase to reinforce the learning from semantically relevant to the problem classes. During testing the votes from irrelevant classes are excluded from voting to maximise the confidence of output predictors. We demonstrate that the Class-Specific Regression Random Forest outperforms the classic Regression Random Forest and produces results comparable to the manual annotations.

Acknowledgements

I would like to thank the University of Oxford for being an incredibly motivating place.

I would like to thank my supervisor Prof. J. Alison Noble for being so loyal to my research.

And I would like to thank all my friends for those countless sleepless nights, which inspired my work.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions 3

1.3 Thesis Outline 6

2 Literature Review **7**

2.1 Echocardiography 7

2.1.1 Ultrasound Physics 7

2.1.2 2D Echocardiography 8

2.1.3 3D Echocardiography 9

2.1.4 Contrast Echocardiography 10

2.1.5 Stress Echocardiography 11

2.2 Echocardiography Image Analysis 12

2.2.1 Denoising of Echocardiographic Images 12

2.2.1.1 Adaptive filters. 12

2.2.1.2 Partial Differential Equations based approaches. 13

2.2.1.3 Multiscale Methods. 14

2.2.2 Features in Echocardiographic Images 16

2.2.3 Echocardiography Image Fusion 22

2.2.3.1 Echocardiography Segmentation 25

2.3 Support Vector Machines and Relevance Vector Machines 29

2.3.1 Support Vector Machine 29

2.3.2 Relevance Vector Machine 32

2.4 Random Decision Forest 33

2.4.1 Random Decision Forest in Medical Imaging 34

2.4.2 Structure of the Decision Forest 34

2.4.3 Datapoints and Features 36

2.4.4 Training the Random Decision Forest 37

2.4.4.1 Split Functions 39

2.4.4.2 Energy Models for Splitting 40

2.4.5 Testing the Random Decision Forest . . . 41
2.4.6 Key Model Parameters . . . 41

3 Feature Selection and Wall Motion Classification of 2D Stress-Echocardiography 43
3.1 Introduction . . . 43
3.2 Critique of Existing Classification Methodology . . . 45
3.3 Database . . . 47
3.4 Pre-processing . . . 48
3.5 Candidate Features . . . 49
3.5.1 Cavity Area . . . 49
3.5.2 Rate of Change of Cavity Area . . . 51
3.6 Feature Selection methods . . . 51
3.7 Feature Selection and Wall Motion Classification using SVM and RVM . . 53
3.8 Feature Selection and Wall Motion Classification using Lasso Regularization, Elastic Net Regularization and Regularized Least Squares . . . 58
3.8.1 Penalized Empirical Risk Minimization and Regularized Least Squares 58
3.8.2 Lasso Algorithm . . . 60
3.8.3 Elastic Net Regularisation . . . 61
3.9 Experiments . . . 62
3.10 Results . . . 64
3.11 Discussion . . . 66

4 Segmentation of 3D Echocardiograms using Random Forest. 69
4.1 Introduction . . . 69
4.2 Random Decision Forests for Classification . . . 70
4.2.1 Training the Classification Random Decision Forest . . . 70
4.2.1.1 Split Function . . . 71
4.2.1.2 Energy Model . . . 71
4.2.2 Leaf Node Prediction Model. Ensemble Prediction Model. Testing the Classification Random Decision Forest. . . . 73
4.3 Feature Sets . . . 74
4.4 Implementation . . . 75
4.5 Understanding the Classification Random Decision Forests . . . 77
4.6 Experiments . . . 79
4.6.1 Model Parameters . . . 80
4.7 Results . . . 83

5 Features for Accurate Segmentation of Left Ventricle in 3D Echocardiograms 86
5.1 Introduction . . . 86
5.2 Specifics of Segmentation of Ultrasound Images using Decision Random Forests . . . 87
5.3 Improving the Classic Features. Registration of 3D Echocardiograms. . . . 89
5.3.1 Modified Hough Transform for Circle Detection in Short-Axis Slices. 91
5.3.2 Long Axis Detection and Global Alignment of 3D Echocardiography 97
5.4 FA Features . . . 99
5.5 *LoCo* and *GloCo* Features. . . . 99
5.5.1 *LoCo* Features . . . 102
5.5.2 *GloCo* Features . . . 102
5.5.3 Landmark Detection. Centrally Oriented Feature Asymmetry Measure. . . . 104
5.5.4 Comparison of position and *LoCo* and *GloCo* features. . . . 107
5.6 Experiments . . . 109
5.6.1 Model Parameters . . . 109
5.6.2 Feature Parameters . . . 111
5.6.3 Validation Metrics . . . 112
5.7 Results . . . 112
5.8 Discussion . . . 113
5.8.1 The Improved Classic Features . . . 113
5.8.2 The FA Features . . . 114
5.8.3 *LoCo* and *GloCo* Features . . . 114

6 Standard Views Extraction from 3D Echocardiography using Regression Voting 125
6.1 Introduction . . . 125
6.2 Regression Random Forests for Plane Detection in 3D volumes . . . 127
6.2.1 Training . . . 127
6.2.2 Node Optimization . . . 128
6.2.3 Testing . . . 129
6.3 Class-Specific Regression Random Forests for Plane Detection in 3D volumes 131
6.3.1 Training . . . 132
6.3.2 Node Optimization . . . 132
6.3.3 Testing . . . 135
6.4 Features . . . 138
6.5 Experiments . . . 138
6.5.1 Model Parameters . . . 138

6.5.2 Validation Metrics 140
6.6 Results 141
6.7 Discussion 142

7 Conclusions and Future Work **144**
7.1 Summary 144
7.2 Future Work 146
7.2.1 Larger Datasets 146
7.2.2 GPU Implementation 146
7.2.3 Wall Motion Classification of 3D Echocardiography and MRI Cardiac Data 146
7.2.4 Oblique Classification Random Forests for Segmentation of 3D Echocardiography 147
7.2.5 Oblique Class-Specific Regression Random Forests with *LoCo* and *GloCo* features for the Standard Views Extraction 148

Bibliography **149**

List of Figures

2.1 The standard viewing planes used in 2D echocardiography. 8
2.2 Mechanical scanning technique for reconstructed 3D echocardiography . . 9
2.3 3D volume acquisition. 10
2.4 The effect of NL-mean filter applied to an ultrasound image. 16
2.5 Analysis of the Fourier components of features. 17
2.6 Feature Asymmetry measure. 22
2.7 Computing the discrete wavelet transform. 23
2.8 Fusion framework. 24
2.9 Echocardiography fusion. Example. 25
2.10 Linear Support Vector Machine. 30
2.11 Mapping the original data space into a space of high dimension. 30
2.12 Decision Tree. 35
2.13 Notations. 36
2.14 Split Node. 38
2.15 Split Function. 39

3.1 An example of visual side-by-side comparison of rest (left column) and stress (right column) images. 45
3.2 Patient dataset structure. 47
3.3 Pre-processing. Frames interpolation. 48
3.4 Cavity Area of 2C data. 49
3.5 Extracting candidate features. 50
3.6 Structure of the dataset. 53
3.7 Selection of Optimal Features using SVMs and RVMs. Stage 1. 55
3.8 Selection of Optimal Features using SVMs and RVMs. Stage 2. 56
3.9 Workflow for feature selection and wall motion classification using Lasso Regularization, Elastic Net Regularization and Regularized Least Squares. . 62
3.10 Feature selection using Lasso algorithm. 64
3.11 Feature selection using Elastic Net Regularization. 65

4.1 Decision Ransom Forest. Training. 71

4.2 Decision Ransom Forest. Testing. 74
4.3 Rectangular3D, Difference3D and Haar3D features. 76
4.4 Implementation outline of the Random Forest. 76
4.5 Pseudo Code for the training procedure of the Classification Decision Ransom Forest. 77
4.6 Pseudo Code for the training procedure of the Classification Decision Random Forest. 78
4.7 Understanding the Classification Decision Random Forest. 79
4.8 Dice similarity coefficients for the blood pool segmentation. 81
4.9 Dice similarity coefficients for the myocardium segmentation. 82
4.10 Dice similarity coefficients for the blood pool and myocardium segmentation. 82
4.11 Feature Acceptance. 83
4.12 Qualitative results of the 3D Echocardiography segmentation with the Classification Decision Random Forest. 84

5.1 Specifics of ultrasound data. 88
5.2 Algorithm flowchart for segmentation of 3D echocardiography using the classic features extracted from the registered echocardiographic volumes. . 90
5.3 Monogenic Orientation. 92
5.4 Modified Hough transform for circle detection. 93
5.5 Modified Hough transform for circle detection. 94
5.6 Image alignment. 98
5.7 FA features. 100
5.8 Algorithm flowchart for *LoCo* and *GloCo* features extraction. 101
5.9 *GloCo* features. 103
5.10 COFA images. 106
5.11 Landmark detection and *LoCo* and *GloCo* features. 108
5.12 Dice similarity coefficient for the blood pool segmentation using the classic forest. 115
5.13 Dice similarity coefficient for the myocardium segmentation using the classic forest. 116
5.14 Dice similarity coefficient for the blood pool and myocardium segmentation using the classic forest. 116
5.15 Effect of forest size on the Dice coefficient. The classic random forest. . . . 117
5.16 Dice similarity coefficient for the blood pool segmentation using the FA forest. 117
5.17 Dice similarity coefficient for the myocardium segmentation using the FA forest. 118

5.18 Dice similarity coefficient for the blood pool and myocardium segmentation using the FA forest. . . . 119
5.19 Effect of forest size on the Dice coefficient. The FA random forest. . . . 119
5.20 Dice similarity coefficient for the blood pool segmentation using the *LoCo* forest. . . . 120
5.21 Dice similarity coefficient for the myocardium segmentation using the *LoCo* forest. . . . 120
5.22 Dice similarity coefficient for the blood pool and myocardium segmentation using the *LoCo* forest. . . . 121
5.23 Effect of forest size on the Dice coefficient. The *LoCo* random forest. . . . 121
5.24 Feature acceptance for the classic random forest. . . . 122
5.25 Feature acceptance for the FA random forest. . . . 122
5.26 Feature acceptance for the *LoCo* random forest. . . . 123
5.27 Qualitative results of the 3D Echocardiography segmentation with the classic random forest. . . . 124

6.1 Standard views in 3D echocardiography. . . . 126
6.2 Training and node optimization. . . . 128
6.3 Mapping voxels-to-the-plane distance to origin-to-the-plane distance. . . . 130
6.4 Training. Class-Specific Regression Random Forest. . . . 133
6.5 Node optimization. Class-Specific Regression Forest. . . . 134
6.6 Leaf node statistics. . . . 136
6.7 Testing. Class-Specific Regression Random Forest. . . . 137
6.8 ”Distance” between the detected plane and the manually annotated plane. . . . 140
6.9 Visual comparison between manual 2C, 3C and 4C views and automatically detected 2C, 3C and 4C views. . . . 142

List of Tables

2.1 The 'four equations' of stress echocardiographic responses. 11

3.1 Global and local classification accuracy for 2C and 4C views using SVMs . 66
3.2 Global and local classification accuracy for 2C and 4C views using RVMs . 66
3.3 Global and local classification accuracy for 2C and 4C views using Lasso algorithm and Regularized Least Squares. 67
3.4 Global and local classification accuracy for 2C and 4C views using Elastic Net Regularization and Regularized Least Squares. 67

4.1 Mean $(\mu) \pm$ standard deviation (σ) of the Dice coefficient for the myocardium, the blood pool, and myocardium with the blood pool. 85

5.1 Mean $(\mu) \pm$ standard deviation (σ) of the Dice coefficient for the myocardium, the blood pool, and myocardium with the blood pool. 123

6.1 Distance in millimetre and angle in degrees between the planes extracted by the classic and the Class-Specific Regression Forests for the 2C, 3C and 4C view. 141

Chapter 1

Introduction

1.1 Motivation

An estimated 17 million people globally die from cardiovascular diseases every year, accounting for approximately 30% of deaths worldwide (1). This makes cardiac vascular diseases one of the major causes of death. Therefore, novel ways of monitoring and diagnosing cardiovascular diseases are highly relevant. Nowadays, many imaging modalities exist for this purpose: X-ray angiography, magnetic resonance imaging (MRI), cardiac computed tomography (CT) and CT angiography, positron emission tomography (PET) and single photon emission computed tomography (SPECT). But most widely used is ultrasound imaging or echocardiography (1). Echocardiography has some considerable advantages with respect to other modalities: it is relatively cheap and portable; and it does not exclude patients with prostheses, which are sensitive to magnetic fields. Echocardiography is also safe and has no harmful ionizing radiation as long as the guidelines for the use of diagnostic ultrasound are respected (2). Current systems are suitable for imaging in real-time, with resolutions comparable with MRI and CT. Echocardiography is a cardiac imaging modality to evaluate all cardiovascular diseases related to a structural, functional, or hemodynamic abnormality of the heart and great vessels, and it is an effective tool in the diagnosis of such cardiac diseases as myocardial ischemia, congenital diseases, and valvular diseases. Stress echocardiography (images of the heart taken at rest and after exercise or pharmalogical

stress) is a diagnostic technique for assessing left ventricle (LV) dysfunction and underlying coronary disease (3; 4). Due to developments in digital acquisition technology and improvements in image quality, stress echocardiography has evolved from a research tool to a well established routine technique for diagnosing the presence of myocardial ischemia (5; 6).

Recent advances in real-time 3D echocardiography (7; 8; 9; 10) show great potential in overcoming the major limitations of traditional 2D echocardiography (11; 12):

(i) Since 3D echo can image the left ventricle in 3D, diagnostic 2D cross-sections can be selected retrospectively for visual analysis. This can be done manually by following more consistent protocols, or the whole process can be automated (see chapter 6).

(ii) Since the heart is a 3D structure, the true 3D behaviour of the heart can be analyzed. 3D echo has potential for more accurate, realistic, and detailed volume and motion analysis because more image information is available.

Despite the aforementioned advantages of 3D echocardiography over 2D echocardiography, the wealth of data in 3D ultimately requires smart automated methods for quantification of left ventricular clinical parameters (13; 14). For example, to calculate the volume of the left ventricle, the endocardial border must be delineated in 3D. This is very difficult and labour intensive to do manually. The qualitative analysis is bound to lead to large interobserver and intraobserver errors. Therefore, it becomes apparent that development of automatic methods for heart function is highly desirable. Automated methods are invaluable for achieving 3D quantification as they limit the observer input and thus suppress variability in the assessment.

The main areas for automated analysis of 3D echocardiography include:

1. Wall motion and wall thickness analysis. By studying both the regional motion patterns of the left ventricle wall and the change in myocardial wall thickness over a cardiac cycle, myocardial tissue functionality can be assessed. Most often, the wall motion is assessed visually by assigning a qualitative score to each segment. This is highly subjective and therefore, it requires accurate automated and robust methods.

2. Segmentation of myocardium from 3D echocardiography. Accurate segmentation of heart muscle is important for both qualitative visual inspection of heart function and quantitative analysis such as ejection fraction estimation and myocardial wall thickness estimation.

3. Extraction of standard views from 3D echocardiography. In clinical routine, visual assessment is done by inspecting the standard 2-chamber, 3-chamber and 4-chamber views from 3D echo volumes. However, manual extraction of these views is both operator-dependant and time-consuming. Therefore, it calls for robust automated methods to reduce the human error.

In this thesis we aim to develop fully automatic techniques for the three aforementioned problems in order to facilitate accurate assessment of heart functionality from echocardiographic images and to improve clinical workflow.

1.2 Contributions

This thesis concerns three problems approached using machine learning methods:

(i) wall motion analysis in 2D stress echocardiography

(ii) segmentation of the myocardium in 3D echocardiography

(iii) standard views extraction from 3D echocardiography

We first present several methods for automatic classification of cardiac wall motion and show that accurate assessment of wall motion highly relies on segmentation of heart muscle. Therefore, we investigate a novel application of one of the most prominent machine learning techniques, called the Classification Decision Random Forest, to segment the myocardium in 3D echocardiograms. We further present several new methods to improve current state-of-the-art. We also develop a new one-step regression voting approach for the task of plane extraction from 3D volume. The method is investigated on the problem of extraction of the standard views from 3D echocardiography. We also present a novel Class-Specific Regression Forest for more accurate extraction of the standard views from 3D echocardiograms.

The main contributions of this thesis are summarised below:

1. We propose four new methods for feature selection and wall motion classification from 2D stress echocardiography. The methods are based on the following techniques:

(i) Support Vector Machines

(ii) Relevant Vector Machines

(iii) Lasso algorithm and Regularised Least Squares

(iv) Elastic Net regularisation and Regularised Least Squares

The first two methods are so called wrapper methods, when the proposed machine learning techniques are used to evaluate a set of features. We propose a smart way to search the space of all possible feature subsets, and extract the optimal set of features to maximise wall motion classification accuracy. The latter two methods are so called embedded methods, where the feature selection procedure is embedded into the training phase.

Parts of this work have been published at peer-reviewed conferences:

(i) K. Chykeyuk, D. A. Clifton, and J. A. Noble, "Feature extraction and wall motion classi- fication of 2D stress echocardiography with support vector machines," in SPIE Medical Imaging, pp. 79630H-79630H, International Society for Optics and Photonics, 2011. (15)

(ii) K. Chykeyuk, D. A. Clifton, and J. A. Noble, "Feature extraction and wall motion classi- fication of 2D stress echocardiography with relevance vector machines," in Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on, pp. 677-680, IEEE, 2011. (16)

2. We propose three ways to improve the segmentation results with the Classification Random Forests:

(i) we present a novel set of features, called *LoCo* and *GloCo* features, with local and global geometrical shape constraints on coupled endo- and epicardial boundaries

(ii) we employ a new set of features extracted from the feature asymmetry images

(iii) we reinforce the classic features by employing a novel registration method, which

focuses on aligning the left ventricle centre lines of all the hearts

Within this work the following contributions have also been made:

(i) a novel modified Hough transform for circles to detect the epicardial and endocardial centres in short-axis images. The new transform is based on the feature asymmetry and the local orientation in the image, and allows for accurate detection of the left ventricle centre line. The centre lines are further used to register 3D echocardiographic images.

(iii) a novel *COFA* measure for separation of epicardial and endocardial boundaries. This measure enables extraction of the proposed *LoCo* and *GloCo* features.

This work has been partially published at a peer-reviewed conference:

K. Chykeyuk, M. Yaqub, and J. A. Noble, "Novel context rich *LoCo* and *GloCo* features with local and global shape constraints for segmentation of 3D echocardiograms with random forests," in Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging, pp. 59-69, Springer, 2013. (17)

3. Within the work for standard views detection from 3D echocardiography we propose the following:

(i) we present a natural one-step approach to extract the standard views from 3D echocardiography using regression voting. To our knowledge, the problem of plane detection from 3D images is addressed by regression voting for the first time.

(ii) we also present a new method called a Class-Specific Regression Forest, where the class label information is incorporated into the training phase to further reinforce the outcome by collecting votes only from the semantically relevant parts of the image.

This work has been published at a peer-reviewed conference:

K. Chykeyuk, M. Yaqub, and J. A. Noble, "Class-specific regression random forest for accurate extraction of standard planes from 3D echocardiography," in Medical Computer Vision. Large Data in Medical Imaging, pp. 53-62, Springer, 2014. (18)

1.3 Thesis Outline

Chapter 2 describes the background of this thesis, reviews some of the literature on ultrasound image acquisition and techniques that have been successfully applied in the field of medical image analysis. The remainder of that chapter details three machine learning techniques, which have been extensively used in this thesis, namely Support Vector Machines, Relevant Vector Machines and Decision Random Forests.

Chapter 3 presents four methods for feature extraction and wall motion classification of 2D stress echocardiography, and shows that the outcome of the classification highly depends on accurate segmentation of myocardium.

Chapter 4 investigates the application of the Classification Decision Random Forests to segment the myocardium in 3D echocardiograms and describes the peculiarities of the algorithm when applied to ultrasound data.

Chapter 5 proposes two new sets of features - (i) *LoCo* and *GloCo* features, (ii) FA features. It also proposes to reinforce the conventionally used in image analysis features by employing a new registration technique to align the left ventricles in echocardiograms to the same coordinate system.

Chapter 6 presents a new method for extraction planes from 3D images using regression voting and applies the method to the problem of standard views extraction from 3D echocardiography. A new algorithm, called the Class-Specific Regression Forest, is further proposed to improve the regression outcome by twisting the learning phase and by gathering votes from semantically relevant parts of the image.

Chapter 7 concludes this thesis with remarks and possible directions for future work.

Chapter 2

Literature Review

This chapter gives a brief review on ultrasound imaging and techniques that have been successfully applied in the field of echocardiographic image analysis.

Section 2.1.1 describes in simple terms the physics of ultrasound and presents the current state of the art of ultrasound imaging. Section 2.1 reviews some of the available literature on image analysis, specifically denoising, segmentation, fusion and motion analysis. In the last two sections, 2.3 and 2.4, Support Vector Machines, Relevant Vector Machines and Decision Random Forests are described to set the background for the work in this thesis.

2.1 Echocardiography

2.1.1 Ultrasound Physics

Ultrasound signals are longitudinal waves that have frequency higher than human hearing (greater than 20kHz). While ultrasound used for clinical purposes is in the 1 MHz to 20 MHz range, echocardiography utilizes frequencies from 3 to 5 MHz. Higher frequencies can form sharper images, i.e. images with higher resolution. However, since tissues absorb high frequency energy more readily, such images typically have lower contrast. The speed of ultrasound is tissue dependant only. It depends on the attracting force between the molecules in the material, spacing of the molecules and their mass. The speed of ultrasound

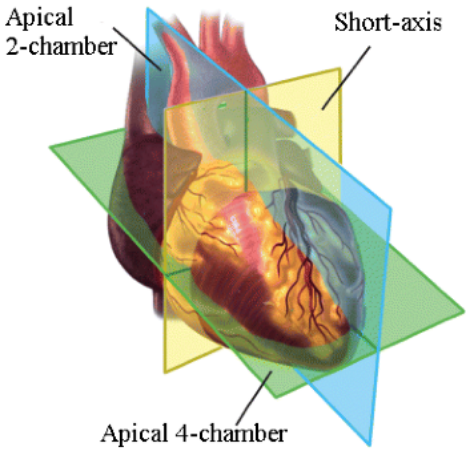


Figure 2.1: The standard viewing planes used in 2D echocardiography. From (19).

is higher in dense materials and lower in compressible ones. In air, ultrasound travels at approximately 330 m/s, in soft tissues at approximately 1540 m/s, and in bone 3400 m/s.

Ultrasound reflects at the boundaries between different tissues. The strength of the reflection is higher at the boundaries of tissues with large difference in acoustic impedance. Frequency of ultrasound remains unchanged as it passes through tissues.

Ultrasound waves are produced by a transducer which converts electrical signals to ultrasound waves in ”send” mode and receives the reflected waves in ”receive” mode. The received signals are turned back into electrical signals and used to form images.

2.1.2 2D Echocardiography

As mentioned in section 2.1.1, ultrasound waves are emitted and a transducer records their reflections from tissue interface. Early transducers used a single ultrasound beam when the beam was emitted and its reflection from the heart tissues was recorded. The movement of cardiac components and their thickness could be seen from these 1D+time images. Such images are called A-mode or M-mode ultrasound images (A and M stand for amplitude and motion correspondently).

Further improvements in ultrasound transducer technology allowed acquiring 2D+time images. These transducers used a linear array of elements and emitted an electronically

(a) Linear movements (b) Tilt-like movements (c) Rotation movements

Figure 2.2: Mechanical scanning technique for reconstructed 3D echocardiography. From (20).

steered ultrasound wave in an organized manner. The reflected signal was recorded to form real time 2D dynamic images. Such images are called B-mode ultrasound images, where B stands for brightness. 2D echocardiography allows visualizing the shape of the heart elements. It also allows for function measurements to be made more accurately.

The diagnostic views that are normally used in 2D echocardiography are the apical 2-chamber view (2C view), apical 4-chamber (4C view), and short axis (SAX), as shown in Fig. 2.1.

2.1.3 3D Echocardiography

2D echocardiography has an inherent disadvantage to represent a 3D organ by means of a set of 2D images. It assumes that the 2C view, 4C view and SAX view planes are found precisely. 2D echocardiography is also bound to some geometrical assumption measuring volumetric characteristics of the heart using a single 2D echocardiogram. Therefore, there has been considerable research in generating 3D echocardiorgams.

Initially, 3D volumes were obtained using multiple 2D images acquired by freehand or using fixed mechanical movements of a transducer. These 3D volumes are termed as *reconstructed 3D echocardiography.* In freehand scanning, freehand movements are tracked using a magnetic, optical or acoustic sensor (21). Mechanical movement based scanning uses fixed linear, tilt or rotational movement of the probe, Fig. 2.2. A 3D volume of the

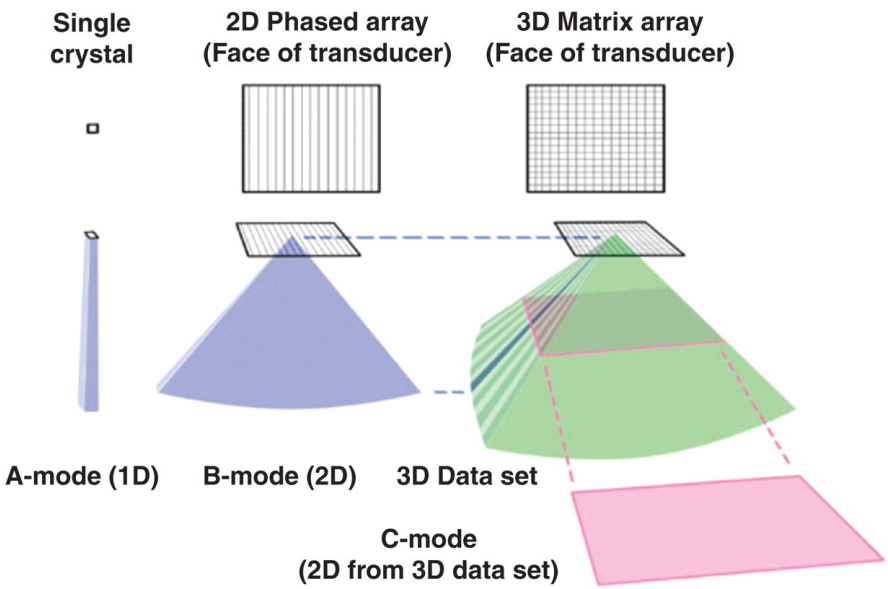


Figure 2.3: 3D volume acquisition. From (19).

heart is then reconstructed offline from these multiple 2D images. However, due to computational cost, offline post processing and artefacts introduced during breathing and other motions, the use of reconstructed 3D echocardiography was found to be limited (22).

Further advances in ultrasound transducer technology allowed acquiring of a full 3D volume. These volumes are termed as *volumetric 3D echocardiography* (19; 20). The transducer consists of a 2D grid of elements (matrix-array transducer) that are used to capture a 3D volume, Fig. 2.3.

No offline reconstruction is needed in this case, and therefore the acquisition is often referred to as real-time 3D echocardiography.

2.1.4 Contrast Echocardiography

Microbubbles are employed in contrast echocardiography in order to enhance borders between myocardium and the blood pool. The microbubbles are injected into the blood circulation system intravenously and the contrast-enhanced images are acquired within a short period of time whilst the microbubbles are still in the system. The microbubbles create a large difference in acoustic impedance between the myocardium and the blood pool making the

reflection of the ultrasound wave stronger. Contrast echocardiography is used when the quality of non-contrast images is deemed qualitatively too low to make an accurate assessment. This is due to undesirable use of the microbubbles to avoid any side effects that the contrast agents might cause.

2.1.5 Stress Echocardiography

In stress echocardiography, images are recorded when the heart is under physical or pharmacological stress to detect early heart disease. The stress is usually induced by physical exercise or pharmacological drug such as dobutamine. Physical stress can be obtained by using a bicycle or treadmill. However, some patients are not sufficiently fit to perform physical exercises making the physical stress less used compared to pharmacological stress. In the pharmacological case, dobutamine is injected intravenously to increase the need in oxygen and make the heart work harder.

Stress echocardiography aims to induce potentially hidden cardiac functional abnormalities. Cardiologists make visual assessment of regional walls in real time in order to indicate which of segments are working abnormally. The purpose of qualitative analysis is to describe the segments as being normal, hypokinetic, akinetic or dyskinetic, Table 2.1.

Case	Rest	+	Stress	=	Diagnosis
1	Normokinesis	+	Normokinesis, Hyperkinesis	=	Normal
2	Normokinesis	+	Normokinesis, Dyskinesis	=	Ischemia
3	Akinesis	+	Normokinesis, Normokinesis	=	Viable
4	Akinesis, Dyskinesis	+	Akinesis, Dyskinesis	=	Necrosis

Table 2.1: The 'four equations' of stress echocardiographic responses by Picano (23).

Due to high heart rate, stress echocardiographic images have lower temporal resolution compared to rest echocardiographic images and, therefore, are harder to analyze. It has become very important to introduce robust automatic methods for quantitative analysis of heart function and reduce the importance of subjective qualitative analysis. However, there is very little literature on this subject.

2.2 Echocardiography Image Analysis

In clinics, cardiologists make visual assessments of heart function from echocardiographic sequences. The qualitative analysis is bound to intra- and inter-observer errors and therefore is not 100% reliable. To reduce operator subjectivity in assessment of heart function, reliable and robust automatic methods for echocardiographic image analysis are needed.

Although considerable research has been devoted to echocardiographic image processing and analysis, this field requires more robust and quick methods for more accurate cardiac function evaluation. In this section, methods for echocardiographic data denoising, fusion and segmentation are reviewed and the most prominent ones of relevance to this thesis are discussed in greater detail.

2.2.1 Denoising of Echocardiographic Images

Although echocardiography is the most widely used test for cardiac function assessment, it has some inherent limitations related to the quality of ultrasound images. The images are affected by speckle. Speckle is a well-studied effect in acoustics and produced by interference of sound waves reflected from tissue components smaller than the wavelength of the sound. Nonetheless, whereas in some situations speckle is useful in image interpretation, often speckle is considered as noise and therefore, needs to be suppressed. Denoising of ultrasound images is challenging since speckle is tissue-dependent and speckle artefacts cannot be easily modelled. Therefore, ultrasound imaging filtering is not a trivial task and requires more sophisticated filters for speckle reduction.

The ultrasound speckle denoising filters can be divided into adaptive filters, partial differential equations-based approaches and multiscale methods (24).

2.2.1.1 Adaptive filters.

Adaptive filters (25) are the simplest technique to restore ultrasound images because they are easy to implement and control. Adaptive filters adjust their transfer function according to an optimization algorithm driven by an error signal. The first adaptive filter was proposed

by Lee et al. (26). The basic assumption is that the mean and variance of a voxel intensity over all subjects is equal to the local mean and variance of all voxels in a patch centred at this voxel. The noise is assumed to be multiplicative, which has the characteristic that brighter areas are noisier. Mathematically, it can be presented by

$$z_{i,j} = x_{i,j} u_{i,j} \tag{2.1}$$

where $z_{i,j}$ is a degraded voxel, $x_{i,j}$ is the true intensity value and $u_{i,j}$ is the noise component. An optimal linear approximation of 2.1 is then used to produce a filtering algorithm similar to that for the additive noise case.

Subsequent improvements were proposed by Lopes et al., Karaman et al., Park et al. (27; 28) where the technique exploits local image statistics to determine specific areas to be processed further.

2.2.1.2 Partial Differential Equations based approaches.

An Anisotropic Diffusion filter was proposed by Perona-Malik (29). The Anisotropic Diffusion filter performs intra-region smoothing and preserves edges. It performs well for images corrupted by additive noise. The work by Perona-Malik proposed the following nonlinear PDE for filtering an image:

$$\begin{cases} \frac{\partial I}{\partial t} = div\left[c\left(|\bigtriangledown I|\right) \cdot \bigtriangledown I\right] \\ I(t=0) = I_0 \end{cases} \tag{2.2}$$

Applied to ultrasound images, anisotropic diffusion essentially enhances the speckle, instead of filtering it. The strength of partial differential equations (PDE) to reduce edge sensitive noise has been further adapted for speckle reduction in ultrasound images, resulting in the Speckle Reducing Anisotropic Diffusion filter (SRAD) (30; 31; 32). Krissian et al. (33) proposed the Oriented SRAD filter (OSRAD), which takes into account the local directional variance of the image intensity, i.e. the local image geometry. Unlike adaptive

speckle reduction filters, PDE-based approaches preserve edges while smoothing images.

2.2.1.3 Multiscale Methods.

Several wavelet thresholding methods have been proposed (34; 35; 36; 37; 38). All of these methods are based on the assumption that the logarithm compression of ultrasound images transforms the speckle into an additive Gaussian noise. Pizurica et al. (39) proposed a wavelet-based generalized likelihood formulation, where no prior on noise and signal statistics was imposed. Other approaches also performed wavelet thresholding adapted to the non-Gaussian statistics of the signal in the Bayesian framework (40; 41; 42; 43).

All methods for speckle reduction in ultrasound images described above are based on the locally adaptive recovery paradigm (44). Recently, Buades et al. (45) has proposed a new patch-based nonlocal recovery paradigm where the local comparison of voxels is replaced by nonlocal comparison of patches. The so-called Nonlocal Means filter (NL filter) does not make any assumption on the location of the voxels that are used to denoise the current voxel. The restored intensity of a voxel x_i is the weighted average of all the voxel intensities $u(x_i)$ in the image I:

$$NL(u)(x_i) = \sum_{x_j \in I} w(x_i, x_j) u(x_j) \tag{2.3}$$

where the weight $w(x_i, x_j)$ assigned to a voxel x_j does not depend either on the spatial or on the intensity distance between voxels x_i and x_j. Instead of comparing the intensity of the voxels, which may be corrupted by noise, an NL filter considers the patterns around the voxels. More precisely, the weight $w(x_i, x_j)$ between voxels x_i and x_j defines the similarity between two local patches N_i and N_j centred on voxels x_i and x_j, such that $w(x_i, x_j) \in [0, 1]$:

$$w(x_i, x_j) = \frac{1}{Z_i} \exp(-\frac{dp}{h^2}) \tag{2.4}$$

where Z_i is a normalization factor ensuring that $\sum_{x_j \in I} w(x_i, x_j) = 1$, h is a filtering parameter controlling the decay of the exponential function and dp is the distance computed

between two image patches N_i and N_j. This distance is simply the L_2-norm convolved with a Gaussian kernel.

A disadvantage of the NL filter is its computational cost. Coupe et al. (46) proposed a fast optimized implementation of the NL-mean filter for 3D MRI and later adapted the filter for speckle noise (24). In fast implementation, the image is divided into overlapping blocks B_i and the intensities of these blocks are restored as follows:

$$NL(u)(B_i) = \sum_{B_j \in \Delta} w(B_i, B_j) u(B_j) \tag{2.5}$$

where Δ is a search volume, centred at the central voxel of the block B_j, $w(B_i, B_j)$ is calculated as in 2.4:

$$w(B_i, B_j) = \frac{1}{Z_i} \exp(-\frac{dp}{h^2}) \tag{2.6}$$

where for ultrasound images, the difference between two patches B_i, B_j is the so called Pearson distance (24):

$$dp(u(B_i), u(B_j)) = \sum_{p=1}^{P} \frac{(u^{(p)}(B_i) - u^{(p)}(B_j))^2}{(u^{(p)})^{2\gamma}(B_j)} \tag{2.7}$$

with P being the width of a patch. This distance is derived assuming the following general speckle model:

$$u(x) = v(x) + v^{\gamma}(x)\eta(x) \tag{2.8}$$

where $v(x)$ is the original image, $u(x)$ is the observed image, $\eta(x) \sim N(0, \sigma^2)$ is a zero-mean Gaussian noise. The factor γ depends on the ultrasound device and additional processing related to image formation (24). An example of the NL-mean filter applied to an ultrasound image can be seen in Fig. 2.4.

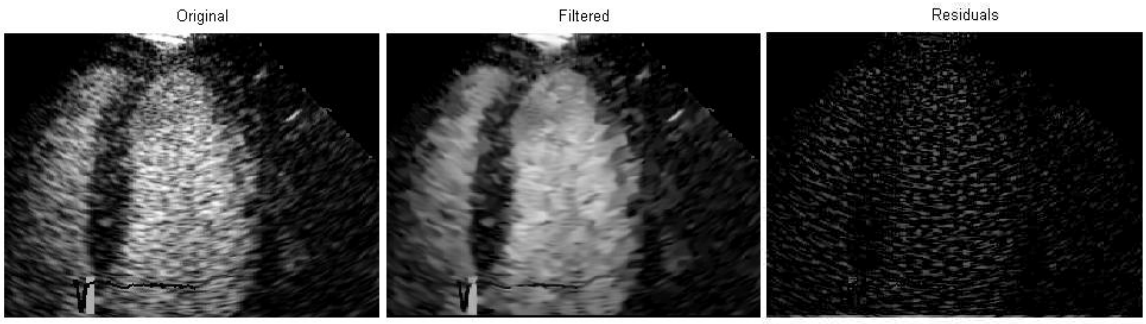


Figure 2.4: The effect of NL-mean filter proposed by Coupe in (24) and applied to an ultrasound image.

2.2.2 Features in Echocardiographic Images

The input data is usually too large and often redundant. We seek a reduced representation of data such that it contains only descriptors relevant to the considered problem. This representation is usually referred to as a features vector and transforming the input data into the set of features is called feature extraction. The feature extraction process extracts the relevant information from the input data and uses this reduced data to perform a desired task.

In echocardiography, low-level feature extraction is an important step before any analysis is made. For LV segmentation, the goal is to detect endocardial and epicardial boundaries, and it is traditionally assumed that they have step-like edge characteristics. Therefore, in echocardiographic image analysis, one usually seeks for low-level features that represent negative and positive step edges. Intensity based methods might not perform well due to the low-contrast nature of echocardiographic images, whereas local-phase based techniques and phase congruency (PC) methods have been shown to be intensity-invariant and less sensitive to speckle (47). It also has been shown that the local phase remains relatively stable across scales compared to intensity measures (48).

Morrone et al. (49), have shown that features such as edges, lines etc. all give rise to points where the Fourier components of the signal are maximally in phase, i.e. congruent, see Fig. 2.5.

A Phase Congruency measure was first introduced by Kovesi in (50) as a measure of

Figure 2.5: Analysis of the Fourier components of features. Left image: the step up corresponds to all Fourier components having zero phase. Right image: ridges up and down correspond to all Fourier components having phase $\pi/2$ or $3\pi/2$. From (49).

feature significance in images, a method of feature detection that is particularly robust towards changes in illumination and contrast. The phase congruency of the signal $f(x)$ is defined as:

$$PC(x) = max_a \frac{\sum_n A_n cos(\phi_n(x) - \bar{\phi}(x))}{\sum_n A_n} \tag{2.9}$$

where A_n represents the amplitude of the n_{th} Fourier component, and $\phi_n(x)$ represents the local phase of the Fourier component at position x.

To calculate *PC*, one needs to estimate the local phase, which is done using the Hilbert Transform and the Analytic Signal. The Hilbert transform is a transform that corresponds to rotation by $\frac{\pi}{2}$ in the Fourier domain and is defined as:

$$f_H(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(\tau)}{\tau - t} d\tau = \frac{1}{\pi} \int_{-\infty}^{\infty} f(\tau) \frac{1}{\tau - t} d\tau = \frac{1}{\pi} (f(t) * \frac{1}{t}) \tag{2.10}$$

Mathematically the local properties (amplitude and phase) of a signal are defined using the Analytic Signal:

$$f_A(t) = f(t) - j f_H(t) \tag{2.11}$$

where $f_H(t)$ is the Hilbert transform of $f(t)$. As in Fourier domain the Hilbert transform

$f_H(t)$ has the same magnitude as $f(t)$ but is rotated by $\frac{\pi}{2}$, this enables to define

$$\text{local energy as: } A(t) = \sqrt{f^2(t) + f_H^2(t)} \tag{2.12}$$

$$\text{and local phase as: } \tan\phi(t) = \frac{f(t)}{f_H(t)} \tag{2.13}$$

In practice, the analytic signal is computed using a quadrature pair of band-pass filters, i.e. filters that are $\frac{\pi}{2}$ phase shifted. A commonly used filter is the log-Gabor filter, which is defined in the frequency domain as:

$$G(w) = \exp(-\frac{(log(\frac{w}{w_o}))^2}{2(log(\frac{k}{w_o}))^2}) \tag{2.14}$$

where w is the spectral dimension, w_o is the central frequency which is related to the scale of the filter and k is related to the bandwidth of the band-pass filter. The analytic signal is then computed from the filter response of this filter:

$$even(x) = Re(F^I(F^F(f(x) * G(w)))) \tag{2.15}$$

$$odd(x) = Im(F^I(F^F(f(x) * G(w)))) \tag{2.16}$$

$$f_A(x) = even(x) + i\ odd(x) \tag{2.17}$$

where $F^F(..)$ and $F^I(..)$ are the forward and the inverse Fourier transfer functions.

PC in practice is calculated as follows (50):

1. For $n = q, ..., n^{max}$

$$even_n(t) = f(t) * M_n^e(t) \tag{2.18}$$

$$odd_n(t) = f(t) * M_n^o(t) \tag{2.19}$$

$$A_n(t) = \sqrt{even_n^2(t) + odd_n^2(t)} \tag{2.20}$$

2. Calculate:

$$F(t) = \sum_n even_n(t) \tag{2.21}$$

$$H(t) = \sum_n odd_n(t) \tag{2.22}$$

$$W(t) = \sqrt{F^2(t) + H^2(t)} \tag{2.23}$$

3. Calculate:

$$PC(t) = \frac{W(t)}{\varepsilon + \sum_n A_n(t)} \text{ typically } \varepsilon = 0.01 \tag{2.24}$$

where M_n^e and M_n^o are the n_{th} even and add log-Gabor functions, ε is incorporated to avoid division by 0.

The Hilbert transform is valid only for 1D signals and extension of the concept to 2D and 3D is not trivial. It was proposed to use a quadrature-pair of oriented band-pass filters for the local analysis of 2D and 3D signals (51). An oriented 3D log-Gabor filter can be

defined in the frequency domain as:

$$G(w,\phi,\theta) = \exp(-(\frac{(log(\frac{w}{w_o}))^2}{2(log(\frac{k}{w_o}))^2} + \frac{(\phi-\phi_o)^2}{2\sigma_\phi^2} + \frac{(\theta-\theta_o)^2}{2\sigma_\theta^2})) \tag{2.25}$$

where ϕ_o and θ_o are the filter angles in azimuth and zenith orientations, and σ_ϕ^2 and σ_θ^2 are the respective filter spreads.

As mentioned above, only features that have step-like edge characteristics need to be extracted for echocardiography analysis. Kovesi (50) adapted the measure in 2.24 to estimate local symmetry (valleys, ridges) and local asymmetry (steps, edges) in images and later Mulet-Parada et al. (47) adapted 2.24 to the so called Feature Asymmetry (FA) measure and showed that it was a good measure for detecting step-like edges in echocardiographic images:

$$FA_{3D}(x,y,z) = \sum_o \frac{\lfloor \|odd_o(x,y,z)\| - \|even_o(x,y,z)\| - T_o \rfloor}{\sqrt{(even_o(x,y,z))^2 + (odd_o(x,y,z))^2} + \varepsilon} \tag{2.26}$$

where o is the filter orientation, ε is a small constant to avoid division by zero, T_o is an orientation specific threshold that suppresses any response due to noise or symmetric points of the image.

Rajpoot et al. (52) modified the FA measure 2.26 using the monogenic signal (53). The monogenic signal is a high dimensional generalization of the analytic signal 2.11 and is based on Riesz transform instead of Hilbert transform (53). The monogenic signal is formed by combining the original image (for instance a 3D image) with its Riesz filtered components:

$$I_M(x,y,z) = [I(x,y,z), I*h_1(x,y,z), I*h_2(x,y,z), I*h_3(x,y,z)] \tag{2.27}$$

The Riesz filters are described in the frequency domain as:

$$(H_1(u,v,w),H_2(u,v,w),H_3(u,v,w)) = \left(i\frac{u}{\sqrt{u^2+v^2+w^2}},i\frac{v}{\sqrt{u^2+v^2+w^2}},i\frac{w}{\sqrt{u^2+v^2+w^2}}\right) \tag{2.28}$$

In practice, the image is first filtered using a band-pass filter. Selecting a bandpass filter is the only choice to be made in computing the monogenic signal aside from how to combine the results from different scales. Odd and even filter responses are calculated as follows:

$$even(x,y,z) = I_{M,1}(x,y,z) \tag{2.29}$$

$$odd(x,y,z) = \sqrt{I_{M,2}(x,y,z)^2 + I_{M,3}(x,y,z)^2 + I_{M,4}(x,y,z)^2} \tag{2.30}$$

where $I_{M,i}$ are the components of the monogenic signal 2.27.

Rajpoot et al. (52) combined the results in a similar fashion as in 2.24 and 2.26 but over multiple scales yielding the following FA measure:

$$FA_{3D}(x,y,z) = \sum_s \frac{\lfloor \|odd_s(x,y,z)\| - \|even_s(x,y,z)\| - T_s \rfloor}{\sqrt{(even_s(x,y,z))^2 + (odd_s(x,y,z))^2 + \varepsilon}} \tag{2.31}$$

where

$$T_s = \exp\left[mean\left(log\left(\sqrt{(even_s(x,y,z))^2 + (odd_s(x,y,z))^2}\right)\right)\right] \tag{2.32}$$

Different scales were obtained by varying the central frequency of the isotropic log-Gabor band-pass filter 2.14, where frequency $w = \sqrt{u^2+v^2+p^2}$ and u,v,p are the spectral dimensions.

An example of the FA measure over multiple scales as in 2.31 is presented in Fig. 2.6.

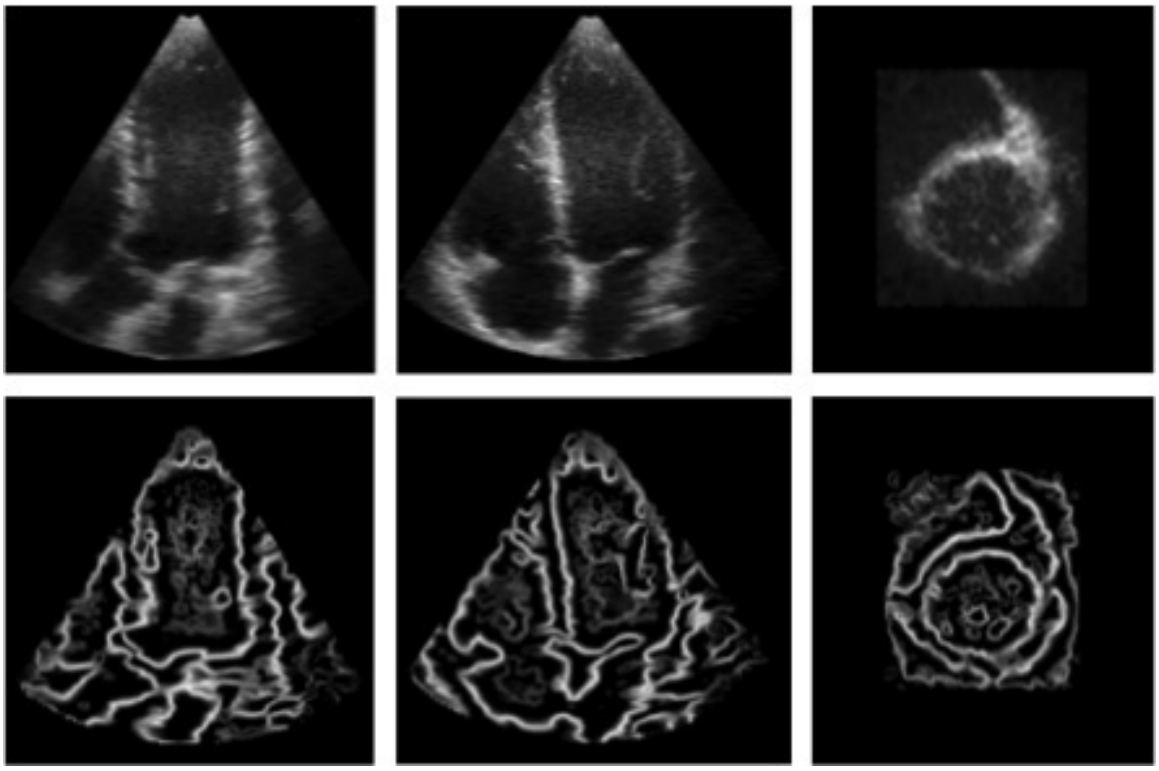


Figure 2.6: Feature Asymmetry measure computed as in equation 2.31.

2.2.3 Echocardiography Image Fusion

Since echocardiographic images may have missing anatomical boundaries and have a limited field-of-view (FOV), they are often not just challenging to analyse automatically but also hard to assess qualitatively. Building on work of Ye et al. (54) and then Grau et al. (55), Rajpoot et al. (56) proposed a multi-view 3D echocardiographic image fusion method to improve the quality of echocardiographic images, fill in missing anatomical information and extend the FOV. Multiple images from different transducer positions or angles are acquired, followed by a registration algorithm to bring these images into alignment. A wavelet-based fusion method is then used to combine the aligned images into a single fused image, preserving salient information from each image.

Wavelet analysis provides an excellent framework for simultaneous signal representation in the frequency and spatial domain and is often used for multi-resolution analysis. Wavelet analysis separates the signal into its low-frequency and high-frequency components and highlights features that are often not visible in the spatial domain. A Discrete

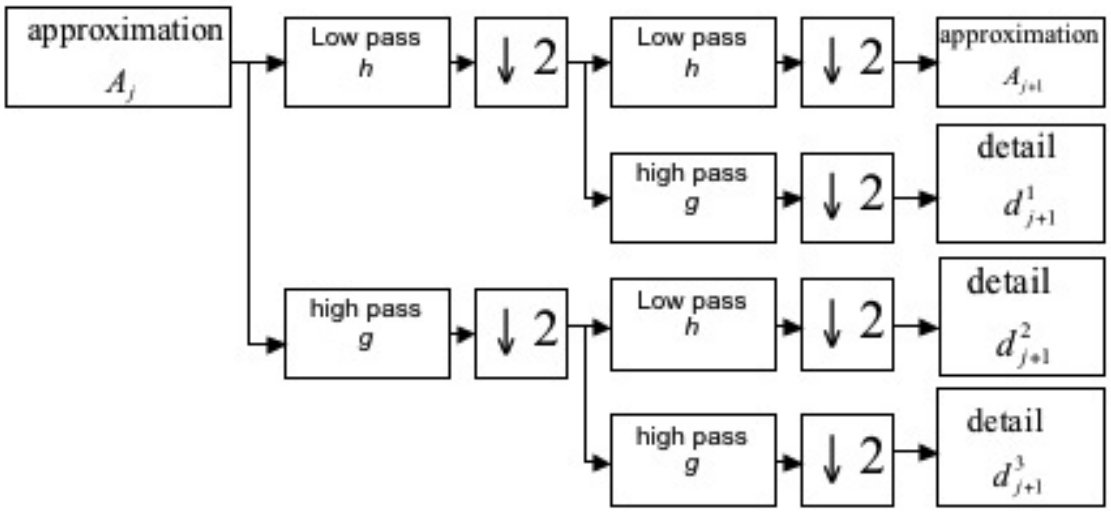


Figure 2.7: A discrete wavelet transform of a 2D signal. From (57).

Wavelet Transform (DWT), applied to a 3D volume, divides it into 8 subbands: 1 low-frequency and 7 high-frequency subbands. An example of wavelet 2D image decomposition is shown in Fig. 2.7.

The first step of the DWT results in a low-pass approximation of the image, which usually contains the salient features of the image, and high-pass details that contain high-frequency information and usually contain noise. Therefore, it is proposed to take the maximum of the low-frequency components of multiple single-view images and average the high-frequency components.

The wavelet-based fusion algorithm therefore consists of 4 steps (56):

1. Perform 1 step of wavelet transform on each single-view image. This results in decomposition of the image into low-frequency and high-frequency wavelet coefficients:

$$W^F(I_i) = [W_L^i, W_H^i] \tag{2.33}$$

where W_L is the low-frequency component of the 3D discrete wavelet transform, and W_H is a vector of 7 high-frequency components of the 3D wavelet transform.

2. Take the maximum of the low-frequency wavelet coefficients to fill in the missing

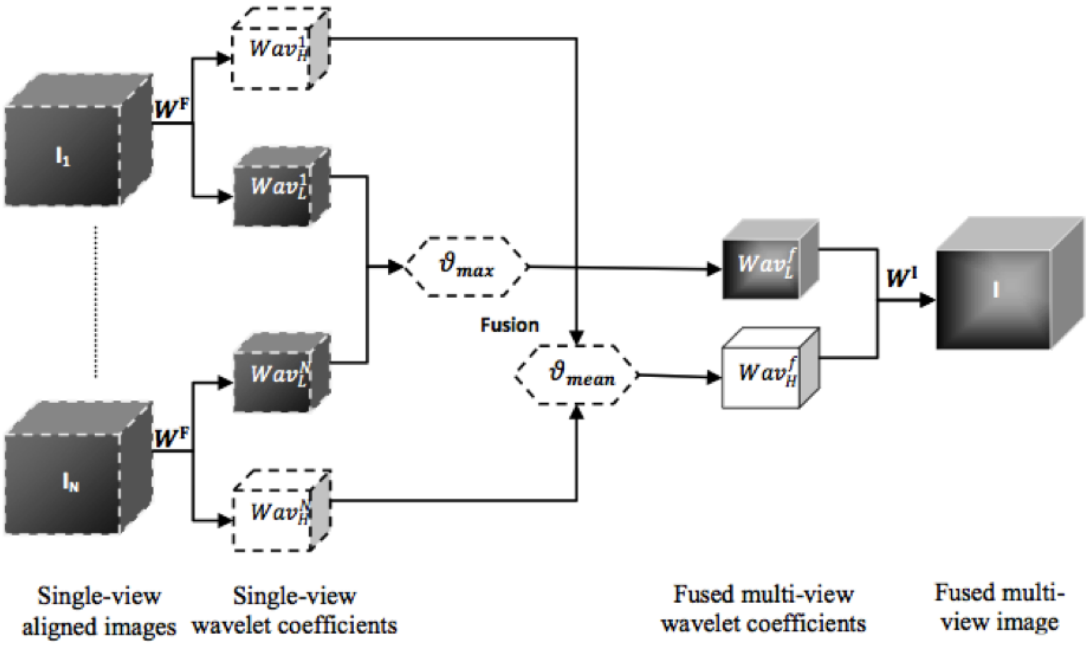


Figure 2.8: Wavelet based echocardiographic image fusion scheme as in (56). The vector W_H consists of 7 high-frequency subbands components. From (56).

anatomical information:

$$W_L^f = max(W_L^1, ..., W_L^N) \tag{2.34}$$

3. Take the mean of the high-frequency wavelet coefficients to suppress the noise:

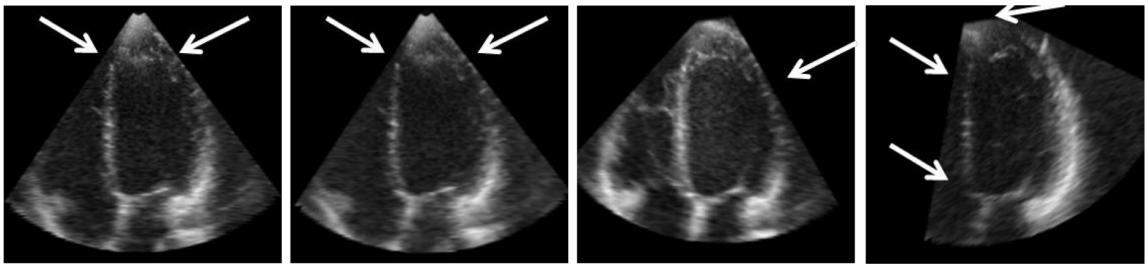
$$W_H^f = mean(W_H^1, ..., W_H^N) \tag{2.35}$$

4. Perform the inverse wavelet transform from the fused wavelet coefficients to reconstruct the fused image:

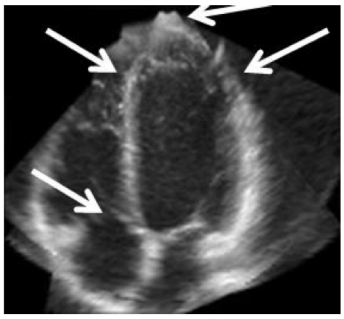
$$I^f = [W_L^f, W_H^f] \tag{2.36}$$

This 4-step process is graphically illustrated in Fig. 2.8.

In (56), it is assumed that the heart has rigid behaviour and therefore, rigid transformation is employed to align single-view images. However, due to the non-rigid motion of the heart, acquiring and aligning a large number of single-view images will lead to some



(a) Single-view images



(b) Fused image

Figure 2.9: Multiple images (a) from different transducer positions or angles are acquired followed by a registration algorithm to bring these images into alignment. A wavelet-based fusion method is then used to combine the aligned images into a single fused image (b), preserving salient information from each image. From (56).

errors in the final fused image. Non-rigid registration of single-view images for the task of echocardiographic image fusion remains the subject of future research.

2.2.3.1 Echocardiography Segmentation

Quantitative parameters derived from 3D echocardiography are essential for objective evaluation of heart function. To compute such parameters from 3D echocardiography, the structures of interest such as epicardial and endocardial borders have to be outlined. Manual delineation of heart wall borders is immensely labor intensive and is bound to intra- and inter-observer errors. Therefore, an automated analysis of heart function is highly desirable for an objective, reproducible quantification and a practical workflow.

Segmentation methods can be divided into 2 broad groups: image-driven methods that make use of information given in the image only, and population based methods that capture behaviour from empirical data and replicate it when applied to a new image. Image-driven methods often fail to segment an object of interest accurately. In other words, image-

driven methods do not work on images where there is not enough structural information about the object (e.g. missing anatomical boundaries) due to intensity drop out and attenuation. Population based methods, however, may be able to find an accurate decision since they learn such patterns from empirical data.

Image-driven methods. Image-driven methods can be divided into edge-driven and region driven methods.

In some of the edge-driven approaches (58; 59; 60; 61), a force that is normally defined by an intensity gradient-based detector, attracts a deformable model towards the object boundary. Some of these methods use energy-based optimization and are also known as balloons, snakes, and active contours. An energy function in this case consists of an external and an internal component:

$$E_{deformable_model} = E_{external} + E_{internal} \tag{2.37}$$

where the external component $E_{external}$ is defined by image forces acting on the deformable model and is supposed to be minimal when the deformable model is at the object boundary. The internal component $E_{internal}$ acts as a constraint on the geometrical form of the deformable model and is supposed to be minimal when the deformable model has the shape of the sought object.

Other edge-driven methods use level sets approach (62), where the curved surface is defined by a different deformation equation such that the shape of the borders is much less restricted and may consist of multiple disconnected surfaces. Rajpoot et al. (61) employed the level set approach for segmenting the LV on fused images using the Feature Asymmetry measure as an edge detector. Although the method utilized multi-view fused echocardiographic images with better image quality and the Feature Asymmetry based edge detector, which is superior to intensity gradient-based edge detectors, it was necessary to perform post-processing operations to extract an accurate delineation of the LV.

Region based methods (63; 64; 65; 66; 67) usually model intensities using some form of

a region similarity criterion. These approaches assume strong inter-region non-homogeneity and intra-region homogeneity. Starting from a seed in the object to be segmented, neighbouring voxels are sequentially added to the region if they belong to the object. Segmentation can also take a multi-scale approach, when the segmentation is propagated from coarse scale to the finer scale and refined using the region similarity criterion. The region (i.e. myocardium) is usually modelled by mean (65), Gaussian (66) or Rayleigh distributions (67), but none of these assumptions are robust for routine 3D echocardiographic images.

Population based methods. There are two types of population based methods for segmentation. One is referred to as classification and it classifies each voxel as belonging to one of the tissue classes, the another one is statistical modelling where statistical variations in actual patient data are modelled from a large dataset of images with expert-drawn borders.

Statistical modelling has become a popular concept for describing the typical variations in the shape, appearance, and composition of parts of the human body. In general, statistical modelling aims at deriving mathematical formulations which describe these typical variations. These models can be used to study variations in patient populations (e.g. differences between healthy and diseased populations) or for the purpose of segmentation.

For shape modelling, point distribution models are most often used (68). Point distribution models characterize shape variability based on Principal Component Analysis (PCA) (69; 70). Shape knowledge is derived from a training set of example shapes, e.g. contours in myocardium. The shapes themselves are expressed in coordinates of landmark points, which are placed at consistent locations in the images. After spatially aligning these shapes, PCA is applied, which generates a linear, mathematical model of a mean shape and a number of characteristic shape variations. Within certain statistical limits, shapes resembling those from the training set can be approximated using the mean shape and a linear combination of the shape variations. The average shape is placed on the image and the parameters of the shape model are then found iteratively: the local edge-like features drive the shape model to the actual borders. This is similar to the edge-driven approaches with the statist-

ical shape model being a geometrical constraint. A similar concept can be used to generate models of the typical look of an image, by applying Principal Component Analysis to image intensities. To remove size and shape dependence from these models, the images first need to be interpolated to a common coordinate frame. The resulting model is also called a texture model. The combination of shape and texture model is the so called Active Appearance Model (71). The model is then adapted to match the image iteratively: the difference between the model-synthesized image and real image determines the next best estimate of the appearance model.

Bosch et al. (72) proposed to incorporate temporal information of 3D+T echocardiographic data into appearance model by putting shape and texture information from multiple time points into the same model (known as Active Appearance Motion Model or AAMM). This way temporal information was modelled implicitly. In other work by Hansegard et al. (73) an Active Shape Model (ASM) (68) in association with a Kalman filter (74) was proposed for modelling the shape from training images. This model was applied for real-time LV segmentation. Leung et al. (75) developed a fully automatic Active Appearance Model for 3D echocardiography, where the model shape parameters were found using intensity gradient based features.

Although statistical modelling for segmentation has been quite popular in recent years, it has some challenges: (i) it is not clear how to select the training data such that most variations are captured (otherwise there will be an explicit constraint on the shapes beyond the learnt variations), and (ii) how to choose the right modelling parameters, for example the number landmark points and how they have to be distributed across the surface.

Classification techniques for segmentation of 3D echocardiography correspond to categorizing image voxels as belonging to LV cavity area, myocardium or background. Lempitsky et al. (76) were the first to propose a Random Forest technique (77) to classify each voxel as being myocardium or non-myocardium. Verhoek et al. (78) extended this method to 3D+T echocardiography through the use of optical flow motion cues.

2.3 Support Vector Machines and Relevance Vector Machines

The major focus of machine learning is to automatically learn the behaviour from empirical data and make intelligent decisions based on such knowledge. The main challenge of machine learning techniques is the fact that the variety of all possible behaviours is often too large to be covered by a set of observed examples. Therefore the learner must be able to produce useful outputs by generalizing from the training data. Here we give a background on two machine learning methods called Support Vector Machines and Relevance Vector Machines, and in chapter 3 we investigate how well these methods perform for the task of wall motion classification in stress echocardiography.

2.3.1 Support Vector Machine

The original SVM algorithm was proposed by Vapnik (79). The standard SVM takes a set of input data, and predicts, for each given input, to which of two possible classes the input belongs. The SVM is a non-probabilistic, non-linear classifier. Given a set of training examples, each marked as belonging to one of two classes, an SVM training algorithm builds a model that predicts whether a new example falls into one class or the other.

More formally, the SVM finds the hyperplane that has the largest distance (or margin) to the nearest training data points of each class, since the larger the margin the lower the classification error of the classifier, as shown in Fig. 2.10. Support vectors are defined to be the subset of the training vectors that lie on the margin. A linear classifier takes the form:

$$f(\mathbf{x}) = \sum_i a_i y_i (\mathbf{x}_i^T \mathbf{x}) + b \tag{2.38}$$

where a_i are coefficients determined as Lagrange multipliers in the optimisation procedure used to find the hyperplane, and take non-zero values only for the support vectors.

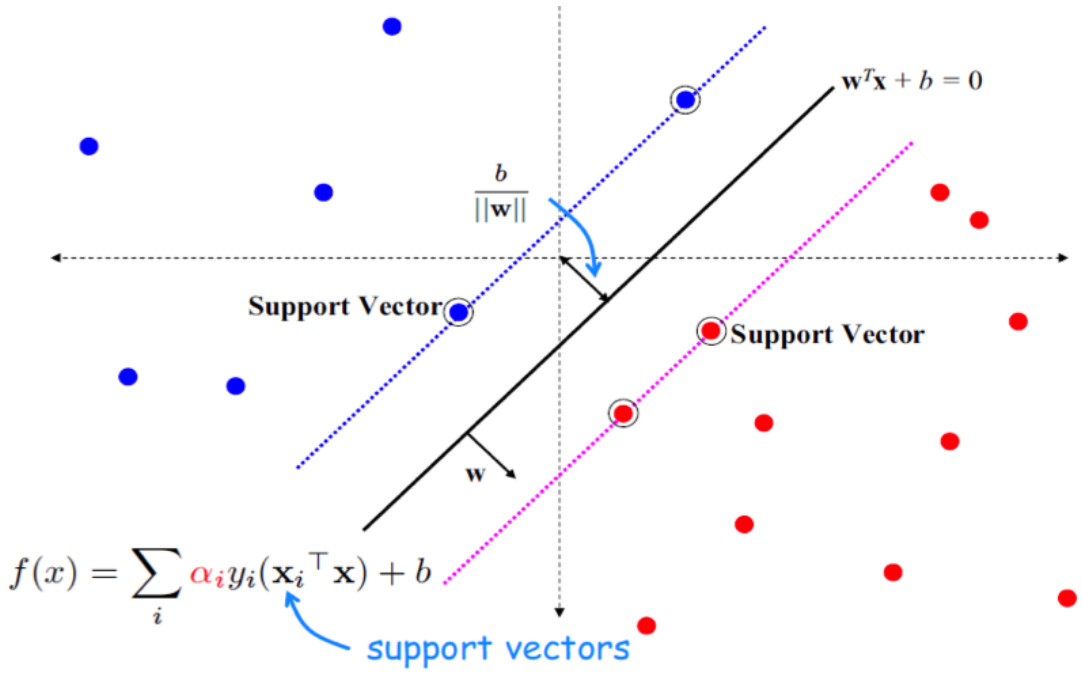


Figure 2.10: Linear Support Vector Machine.

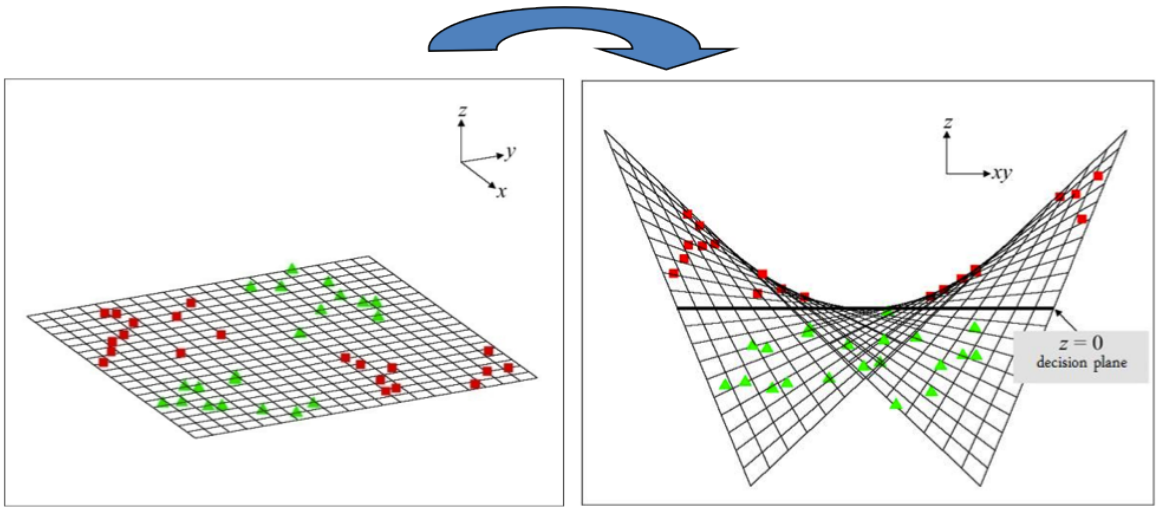


Figure 2.11: Mapping the original data space into a space of high dimension where the data is linearly separable.

Thus, it is evident that the support vectors are the only examples from the training data that explicitly used in the model (it is a sparse kernel technique).

The benefits of the SVM approach are that (i) the classifier concentrates automatically on examples that are difficult to classify, (ii) the calculation scales with the number of support vectors rather than the dimension of the data space, and (iii) the SVM can be shown to balance training error and model complexity, thereby avoiding over-fitting (80).

To allow for the possibility that the two classes cannot be entirely separated by a linear boundary, slack variables ε_i are introduced into the quadratic optimisation problem, thus allowing a minimal number of the training data to be misclassified.

Whereas the original problem may be stated in a finite dimensional space, it often happens that in that space the data are not linearly separable. For this reason it was proposed that the original finite dimensional space be mapped into a much higher dimensional space ϕ, presumably making the separation easier in that space, as illustrated in Fig. 2.11.

In this case, the final classifier in transformed feature space takes the form:

$$\textbf{Classifier: } f(\mathbf{x}) = \sum_i^N a_i y_i \mathbf{x}_i^T \mathbf{x} + b \longrightarrow f(\mathbf{x}) = \sum_i^N a_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b \tag{2.39}$$

$$\textbf{Learning: } min_{a,\varepsilon} \sum_{jk} a_j a_k y_j y_k \mathbf{x}_j^T \mathbf{x}_k + C \sum_i^N \varepsilon_i \longrightarrow \tag{2.40}$$

$$min_{a,\varepsilon} \sum_{jk} a_j a_k y_j y_k \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_k) + C \sum_i^N \varepsilon_i \tag{2.41}$$

subject to

$$y_i f(\mathbf{x}_i) >= 1 - \varepsilon_i \text{ for } i = 1...N \tag{2.42}$$

where C is a regularization parameter.

2.3.2 Relevance Vector Machine

A Relevance Vector Machine (RVM) is an important successor of the SVM, developed by Tipping (81). The RVM is a probabilistic extension of the SVM, which provides posterior probabilities of previously unseen test data belonging to each of the two classes. The RVM is a Bayesian sparse kernel technique for regression and classification that shares many of the characteristics of the SVM whilst avoiding its principal limitations: (i) the outputs of an SVM represent decisions rather than posterior probabilities, and (ii) the SVM uses a complexity parameter C that must be found using a hold-out method such as cross-validation. Additionally, the RVM typically leads to much sparser models resulting in correspondingly faster performance on test data whilst maintaining comparable generalization error (81; 80).

Much like the SVM, an RVM produces estimates of the posterior class distributions using a subset of the training data, termed relevance vectors. Typically, an RVM uses an order of magnitude fewer relevance vectors than the number of support vectors required by the equivalent SVM.

As with the SVM, an RVM assumes a kernel-based model:

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x_i}) \tag{2.43}$$

However, whereas the SVM is based on the maximum-margin principle, the RVM instead takes a Bayesian approach. The RVM assumes a Gaussian prior on the kernel weights w_i, which are assumed to have zero mean and variance a_i^{-1} (or precision a_i). The evidence of the observed data is maximized with respect to these hyperparameters and a significant proportion of posterior distributions of the weights w_i tend towards $\sim N(0,0)$, and hence weights from the model for which the corresponding precision $a_i \to \infty$ are removed. The kernel functions associated with these parameters therefore play no role in the predictions made by the model and so are effectively removed, resulting in a sparse design matrix ϕ, where $\phi_{i,j} = k(x_i, x_j)$. The remaining training examples (i.e. those for which a_i remains

finite) are relevance vectors. By this mechanism, overfitting is generally avoided. While the RVM and the SVM both base their decisions entirely on a subset of the training data, these subsets are usually quite different. Support vectors are always examples lying near the decision boundary, while relevance vectors may be spread throughout the distribution of each class.

2.4 Random Decision Forest

The Random Decision Forest is a machine learning framework that allows to tackle a number of different machine learning problems. The term Random Forest was introduced by Breiman (77) and has proven to show good performance in recent publications (82; 83; 76; 84; 85; 86; 87; 88; 89; 90; 91; 92; 93; 94; 95; 96). In the literature, the terms Random Forest, Randomised Trees, Random Decision Forest are used interchangeably. The Random Decision Forest has been shown to produce a higher accuracy than conventional decision trees (97) and comparable accuracy to the Support Vector Machines (SVM) (98; 99), boosting and other classification techniques (100).

The Random Decision Forests have become very popular within the machine learning and computer vision communities in part due to its recent success in the task of classification - the Microsoft Kinect for XBox 360 (101; 95). However, besides that, one of the main advantages of the Decision Random Forest is its ability to tackle most of machine learning problems in a unified framework. The Random Decision Forest can be tuned to tackle the following problems:

- Classification. Problems with discrete output are cast as a classification task. Examples of the classification problem include scene recognition (e.g., a wedding ceremony, indoors, countryside etc.) and pixel/voxel wise segmentation where a pixel/voxel belongs to one of the classes (e.g., myocardium, blood pool, background, for left ventricle segmentation).

- Regression. Problems with continuous output are regarded as regression task. Object detection and parameters estimation of a model are all examples of regression problem.

- Manifold learning. The problem of nonlinear dimensionality reduction - finding the manifold in which the data lies.

- Density estimation. The construction of an estimate of the underlying probability distribution, based on the observed data.

In this thesis we explore the insight of the random decision forest and employ it as a classifier to segment the LV from 3D echocardiograms (chapters 4 and 5) and as a regressor to automatically extract the standard views from 3D echocardiography data (chapter 6).

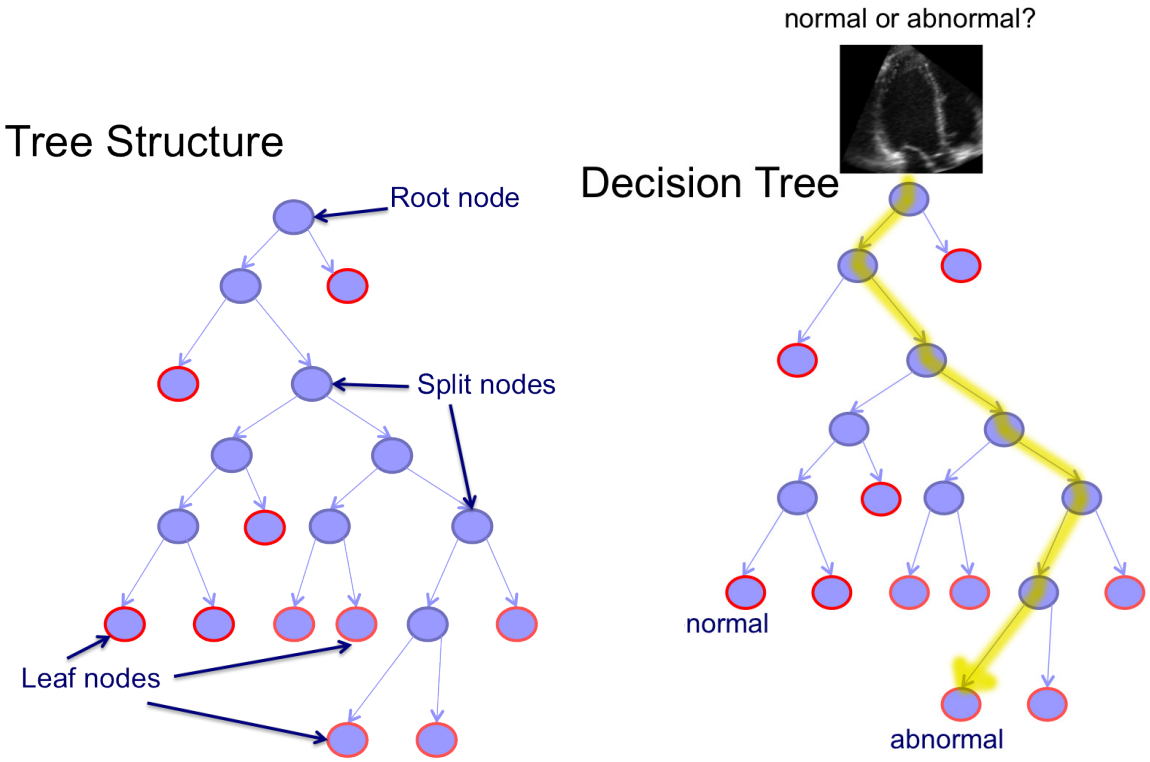
2.4.1 Random Decision Forest in Medical Imaging

The Random Decision Forest has been adopted for medical imaging problems fairly recently. One of the first works in medical image segmentation were (76) and (86), in medical image detection - (102). In recent years, there was an explosion of Random Forest based techniques in medical image analysis literature (84; 102; 87).

The work in (76) was the first to introduce the Random Decision Forest for the task of 3D Echocardiography segmentation. The authors applied the classic random forest algorithm (77) to segment myocardium in 3D images of the Left Ventricle (LV). Segmentation was formulated as a classification task - each voxel belonged to either myocardium or the background. The Regression Random Forest was first introduced in (102) to detect different organs in CT volumes by finding a 3D bounding box around each organ.

2.4.2 Structure of the Decision Forest

A Random Decision Forest is an ensemble of different in content, independent decision trees. Despite decision trees have been around for long time (103; 104), the technique has not been widely used due to inability of a single decision tree to generalise well from the training data to previously unseen data samples. It has been recently discovered that ensembles of different trees produce highly more accurate decisions on unseen data and thus, provide higher generalisation. This allowed the technique to be rediscovered.

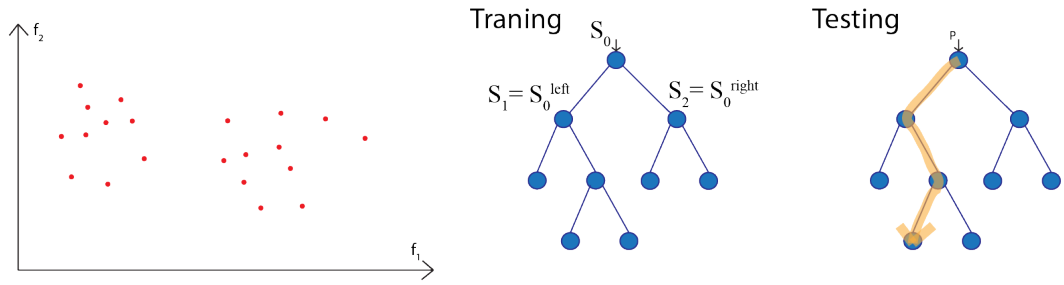


(a) Tree Structure (b) Decision Tree

Figure 2.12: Decision Tree. (a) A tree consists of nodes and edges organised in a hierarchical fashion. Split nodes are denoted by blue circles and leaf nodes are denoted by red circles. (b) A decision tree stores a split function at each split node to be applied to the incoming data, and a predictor at each leaf node.

The tree is a graph structure that consist of nodes and edges organised in a hierarchical manner as shown in Fig. 2.12a. All nodes have one incoming edge and are divided into split nodes and terminal nodes (leaf nodes). The split nodes have two outgoing edges (binary split) while the leaf nodes do not have any outgoing edges and terminate the tree. Each node represents a question about some known property of the object. Depending on the answer to the question at a node, the object is sent to either its left or right outgoing edge. Thus, traversing the tree, the object reaches one of the leaf nodes. The unknown characteristic is then estimated based on the label distribution stored at the leaf node, Fig. 2.12b.

The random decision forest gives a probabilistic estimation of the unknown characteristic of the object of interest. All the trees in the forest are independent and learnt separately. Each tree from the ensemble makes its own probabilistic decision on the output. The decisions from different trees are then unified to produce a single probabilistic decision about the unknown output. Below we describe the key elements of a tree functioning, the pro-



(a) Feature Space (b) Training (c) Testing

Figure 2.13: Notations. (a) Input data is a collection of points in the feature space. The feature space can be infinite (in this example it is 2D). (b) Training procedure involves sending all the training data to the root node, splitting it into two parts by minimising the objective function at the node. The parameters of the split function are saved at the root node. The procedure continues until the stopping criteria is reached. (c) A testing datapoing **p** traverses the tree until it reaches a leaf node. The statistics saved at the leaf node during training is used as a predictor for the datapoint.

cedure of constructing and testing the tree, and how the individual trees are combined into a single decision forest.

2.4.3 Datapoints and Features

A generic object, called a data point, is represented by a feature vector $\mathbf{p} = (x_1, x_2 ...)$ in the feature space, where x_1, x_2 etc. are the features, the descriptors of the datapoint, Fig. 2.13a. For example, in image processing applications, a datapoint can be a pixel in the image and its descriptors can be a pre-chosen filter bank response at the pixel location or just the sum of intensities in the chosen neighbourhood. The amount of features used and their type depends on the application and on the type of data. The dimensionality D of the feature space F can be very large and even infinite. This often makes it impossible to calculate all d features of a datapoint **p** in advance. In this case a subset of d' features $\mathbf{p}' = (x'_1, x'_2, ...x'_d)$ is extracted (either precomputed or on the fly), where d' is dimensionality of the subset. In practice, the dimensionality of the subspace is usually much smaller than the original one $d' << d$.

2.4.4 Training the Random Decision Forest

As discussed above, each split node represents a question to the data that resides at this node. The questions are defined by a split function. The split functions associated with the split nodes of each tree are the key to the forest functioning and are learnt during the training procedure from the exemplar data. In this thesis, the terms 'split function' and 'test function' are used interchangeably. A test function at a split node has a binary output:

$$\mathbf{h}(\mathbf{p}, \boldsymbol{\theta}) \in \{0, 1\}$$

where $\boldsymbol{\theta}$ denotes the parameters of the test function at the node j. The outcome of the test (0 or 1) determines whether the datapoint is sent to the left or to the right child node, Fig. 2.14b.

Thus, the training procedure consists of:

i) selecting the type of the objective function I (section 2.4.4.2)

ii) selecting the type of the split function $\mathbf{h}(\mathbf{p}, \boldsymbol{\theta})$ (section 2.4.4.1)

iii) learning the parameters of the split functions $\boldsymbol{\theta}$ (section 2.4.4.1) at each node.

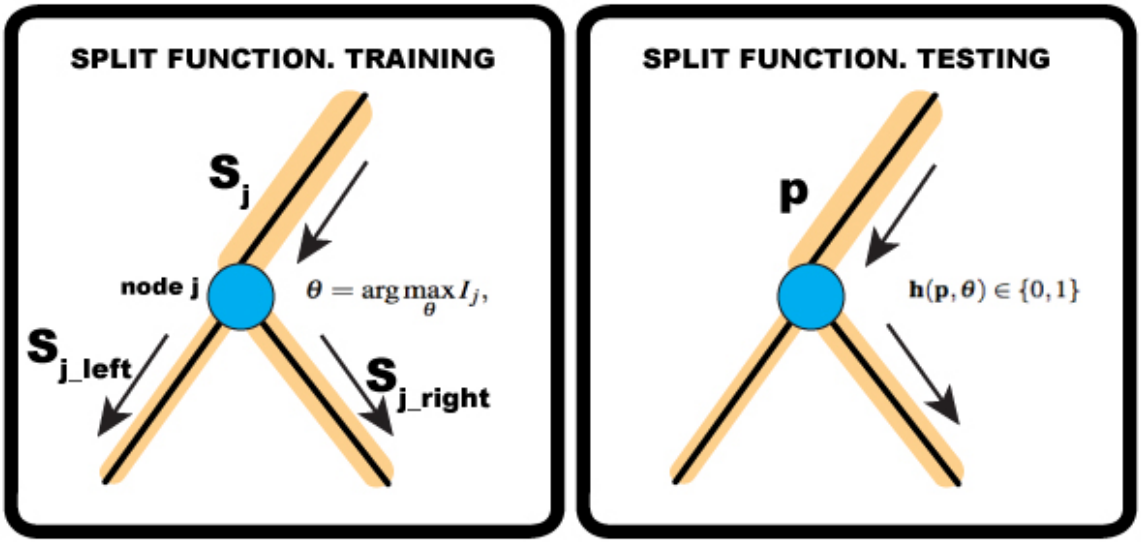
The learning of the split function (i.e. learning the function parameters $\boldsymbol{\theta}_i$) is done by minimising the chosen objective function I_j using the training data S_j that reside at the node j. The best parameters search is done in a greedy manner, testing different values of the set of parameters $\boldsymbol{\theta}$. The parameters that best split the training data S_j at a node j into its left S_{j_left} and right S_{j_right} child nodes are saved at the node j, Fig. 2.14a:

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} I_j, \tag{2.44}$$

where $I_j = I(S_j, \boldsymbol{\theta}_j)$

The type of the split function and the objective function here have a generic form and are different depending on the application.

Besides the tree node optimisation, the shape of the tree, i.e. tree structure, has to be



(a) Split Function. Training Mode. (b) Split Function. Testing Mode.

Figure 2.14: Split Node. (a) A tree consists of nodes and edges organised in a hierarchical fashion. Split nodes are denoted by blue circles and leaf nodes are denoted by red circles. (b) A decision tree stores a split function at each split node to be applied to the incoming data, and a predictor at each leaf node.

optimised during the training procedure. The shape of the tree is defined by the stopping criteria for growing the branches of the tree. The training starts at the root of the tree, dividing the training data at the root node into its left and right branches by node optimisation procedure. The parameters of the split function are stored at the root node. Thus, two additional child nodes are constructed with different disjoint subsets from the training data. The same procedure is applied to the child nodes and the training procedure continues until the stopping criteria is reached. The following stopping criteria might be used:

a) a maximum number of levels D in the tree.

c) a minimum number of data points at a node m.

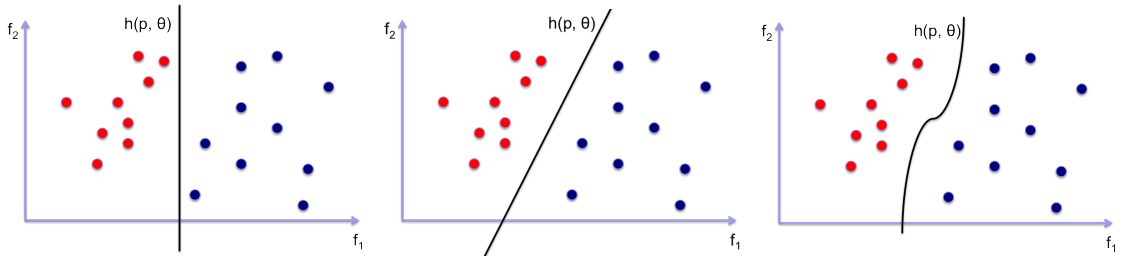
d) a minimum value of the maximum of the objective function I.

These parameters are normally estimated using cross-validation. After the training is done, the following learnt attributes are saved:

a) the structure of the tree.

b) the split functions associated with each node.

c) leaf node statistics.



(a) (b) (c)

Figure 2.15: Split function. (a) An axis-aligned linear split function. (b) An oblique linear split function. (c) Nonlinear split function. In this visualisation, each point $\mathbf{p} \in \mathfrak{R}^2$.

2.4.4.1 Split Functions

The geometric form of the hypersurface that splits the data in the feature space is defined by a splitting model. The splitting model plays a crucial role in the performance of the decision forest. Fig. 2.15 illustrates some examples of the classification problem with data splitting depending on the form of the splitting function. Generally, the splitting models can be divided into linear and nonlinear models.

Linear Splitting Models. The linear splitting can be defined as follows:

$$h(\mathbf{p}, \boldsymbol{\theta}_\mathbf{j}) = \lfloor \boldsymbol{\phi}_\mathbf{j}(\mathbf{p}) \cdot \boldsymbol{\psi} < \tau \rfloor \tag{2.45}$$

where $\lfloor . \rfloor$ denotes the indicator function, the parametrisation $\boldsymbol{\theta} = (\phi, \psi, \tau)$. The function $\phi_j(\mathbf{p})$ selects the type of features used at the node j, τ is a threshold on the split function $h(\mathbf{p}, \boldsymbol{\theta})$, and ψ is the geometric primitive used to separate the data, e.g an axis-aligned hyperplane, an oblique hyperplane etc. If the splitting model is linear, the separation is performed by a splitting point when one feature is used, by a line when two features are used, by a plane when three features are used and by a hyperplane when more features are used. For example, a linear support vector machine's splitting model is linear, i.e. hyperplane.

Nonlinear Splitting Models. The decision shape can also take a nonlinear form, for example a conical form (105). Another example is a nonlinear support vector machine which splits the data by a hypersurface modelled as a weighted sum of (semi)positive kernels placed on the chosen by the algorithm datapoints near the margin (called support vectors). In the majority of applications, the Decision Random Forest performs well using a single feature with linear separation (Fig. 2.15a). Nevertheless, depending on the problem, a combination of features might be preferred to gain more information about the data and make the splitting more accurately. Theoretically, a nonlinear separation model using all features that contain useful information about the unknown output provides the most accurate splitting. However, the computational cost of such algorithms makes the implementation impractical. Besides that, most of the problems we deal with have limited range of training data (especially in medical applications) and such algorithms tend to overfit the training data resulting in poor generalisation.

2.4.4.2 Energy Models for Splitting

Besides the splitting model, i.e. the form of the hypersurface for data splitting, one can think of the objective of the separation. This is expressed by the energy model. In fact, the choice of the energy function itself may influence the form of the splitting model. For example, the nonlinear support vector machine's objective is to maximise the margin between two classes (for binary classification problem). This automatically sets the form of the splitting hyperplane as a weighted sum of kernels placed on a set of datapoints near the margin (support vectors). In this case the objective function together with a choice of a kernel defines the model.

Development of energy models is an active research area. Most often used energy function in the Random Decision Forest is information gain, defined as follows:

$$I(Y|X) = H(Y) - H(Y|X); \tag{2.46}$$

where $H(.)$ denotes entropy, X and Y are random variables.

2.4.5 Testing the Random Decision Forest

In testing mode, the learnt model is utilised to make predictions about the unknown characteristic of an unseen object of interest. A datapoint traverses the tree until it reaches one of its leaf nodes taking a specific path. This path is the same for all datapoints that reach this leaf. In other words, all datapoints that reach a particular leaf node are asked the same series of questions associated with the nodes they visit along the path, and answer these questions identically. Thus, the datapoints resided at the same leaf are likely to be similar. This justifies employing the leaf node statistics to predict the unknown property of the previously unseen point.

The leaf node statistics is captured by the posterior distribution of the training examples at this node:

$$p(c|\mathbf{p})$$

where c is a label (continuous or discrete) and $\mathbf{p}$ is a test datapoint. For different needs, the whole distribution can be kept or the maximum aposteriori can be applied as $c^* = argmax_c p(c|\mathbf{p})$.

The leaf node statistics from different trees is then combined into a single predictor. Chapter 4 and 5 describe a possible way to combine the votes from different trees for the classification problem, while chapter 6 does that for the regression problem.

2.4.6 Key Model Parameters

The prediction abilities of the decision forest depend on the forest parameters. The key model parameters of the decision forest are:

1) A splitting model.

2) An objective function model.

3) The maximum depth of the tree D.

4) Forest size T.

5) The minimum number of data points at a split node m.

6) The minimum value of the maximum of the objective function I.

While the splitting model and the objective function are designed specifically for the application, the maximum depth of the tree D, the forest size T, the minimum number of datapoint at a split node m and the minimum value of the maximum of the objective function I are learnt using cross-validation technique. However, in practice, it is computationally impractical to span the whole parameter space, and parameters m and I are usually chosen empirically, while D is estimated using cross-validation.

Chapter 3

Feature Selection and Wall Motion Classification of 2D Stress-Echocardiography

3.1 Introduction

Stress echocardiography is a common clinical procedure for diagnosing heart disease, especially myocardial ischemia (5; 6). Stress echocardiography analysis is an often challenging area because of low image quality. Very little work has been reported on automated stress echocardiography in the literature in comparison with rest echocardiography. In clinics, the diagnosis of the motion of the heart wall depends mostly on visual assessment, which is highly subjective and operator-dependent (4). Thus, there is a need for automated methods for heart function assessment. Automated wall motion analysis consists of two main steps: (i) segmentation of heart wall borders and (ii) classification of heart function as being either normal or abnormal based on the performed segmentation. In this chapter we describe progress towards developing a technique to automatically analyze heart function from 3D stress echocardiography. We propose four different methods for automated features selection and classification of rest and stress 2D echocardiography that can straightforwardly be

extended to 3D stress echocardiography. Each of the algorithms achieve better classification performance compared to all methods reported to date. The four different methods are based on:

(i) Support Vector Machines (SVM) (79)

(ii) Relevance Vector Machines (RVM) (81)

(iii) Lasso Regularisation + Regularized Least Squares (RLS) (106)

(iv) Elastic Net Regularisation + Regularized Least Squares (RLS) (107)

In clinical practice, cardiologists make a visual side-by-side comparison of rest and stress images for more accurate clinical diagnosis, Fig. 3.1. This motivates the automatic analysis of heart wall motion based on combined rest and stress sequences. We propose a novel selection of features from these sequences, show that they can be used to combine rest and stress information, and thus improve classification accuracy compared with previous work. We performed global and local classification for 2-chamber (2C) and 4-chamber (4C) planes independently, noting that each describes anatomically different parts of the heart. Previous work in performing classification of the heart function considered rest and stress data independently, and only considered features at the two main frames, namely the end-of-diastolic and end-of-systolic frames (108; 109; 110). The work in (111) proposed performing wall motion classification based on the combined information from rest and stress, and utilized a Hidden Markov Model (HMM) as a classifier. The latter has demonstrated the best classification performance of all methods reported to date. In this work, we propose a new and alternative approach to wall motion classification of stress echocardiography, employing a Support Vector Machine (SVM) (section 2.3.1), a Relevance Vector Machine (RVM) (section 2.3.2), the Lasso regularisation algorithm (section 3.9) and the Elastic Net regularisation (section 3.8.3), and using combined information from rest and stress sequences as features upon which classification was performed. We show that the proposed methods overcome difficulties that occur when using the HMM-based approach.

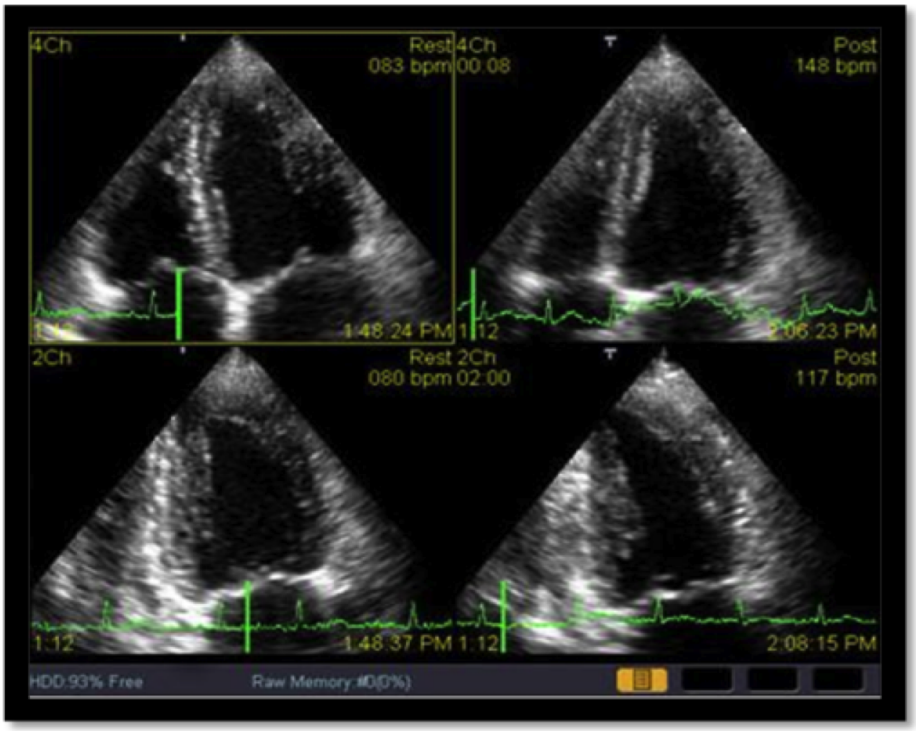


Figure 3.1: An example of visual side-by-side comparison of rest (left column) and stress (right column) images. Top row shows four-chamber (4C) view, bottom row - two-chamber (2C) view.

3.2 Critique of Existing Classification Methodology

The HMM is well suited for modeling dynamic processes and has been successfully applied to speech recognition, ECG segmentation, among many other applications. However, for the purpose of heart wall motion classification, although applied with some success in (112), use of the HMM requires two strong theoretical assumptions to be made which are not strictly valid for echocardiographic data.

Firstly, each state in an HMM represents a number of frames, and observations from each state are modelled by a separate univariate, unimodal Gaussian distribution. This assumes that all observations from one HMM state are considered to be identically distributed; i.e., the features from a number of frames represented by the HMM state over *all patients* are assumed to be identically distributed. This is a strong assumption, and it is likely that the features could be much better described by a different distribution *for each frame*, over all patients. To overcome the problem that the uniform Gaussian is used to fit into all observations from all frames represented by the HMM state over all patients, one

can think of using a Gaussian mixture instead of a single Gaussian. However, this would require the introduction of new parameters and an increase in training dataset size. On the other hand, one can think of introducing new HMM states so that every state represents only one frame with its own uniform Gaussian that represents the observations from one frame over all patients only. But this will introduce new obstacles. The number of frames (i.e. length of cardiac cycle) differs from patient to patient. Thus we do not know how many states the model should have. Interpolating the frames leads to the loss of the main advantage of HMM - capturing time-varying properties of the data. Also all 'hidden' states become visible as at each state the model can only be once before moving to the next one. However, introducing new states also introduces new parameters. It requires additional training datasets - otherwise the model will be overfitted, leading to poor generalisation.

The second theoretical problem with an HMM-based approach is that all observations from the same HMM state, and across all states, are assumed to be independent. The main reason for using information from all frames through the systolic phase of the cardiac cycle is to capture the dependence between different frames, and between rest and stress. However, in assuming that all observations are independent, the HMM does not capture the dependence between features from different frames. Thus, for example, it ignores the importance of instantaneous speed of the myocardial wall. To overcome this problem, a multivariate Gaussian distribution could be used to capture the dependency between different features at the same frame. Again, this would require more training data and still does not solve the problem of independency between features at different frames.

An SVM and RVM based approach overcome the disadvantages associated with making such strong assumptions about the data. A cardiac cycle is normalized in time to a number of frames and contours are interpolated in time at points of interest. This eliminates the need for a model that captures sequential properties of the data. The SVM and the RVM find a hyperplane in multidimensional feature space that best separates the normal and abnormal classes of patient data. An intrinsic property of the SVM and the RVM is that they model all possible correlations between each feature; i.e., while the marginal dis-

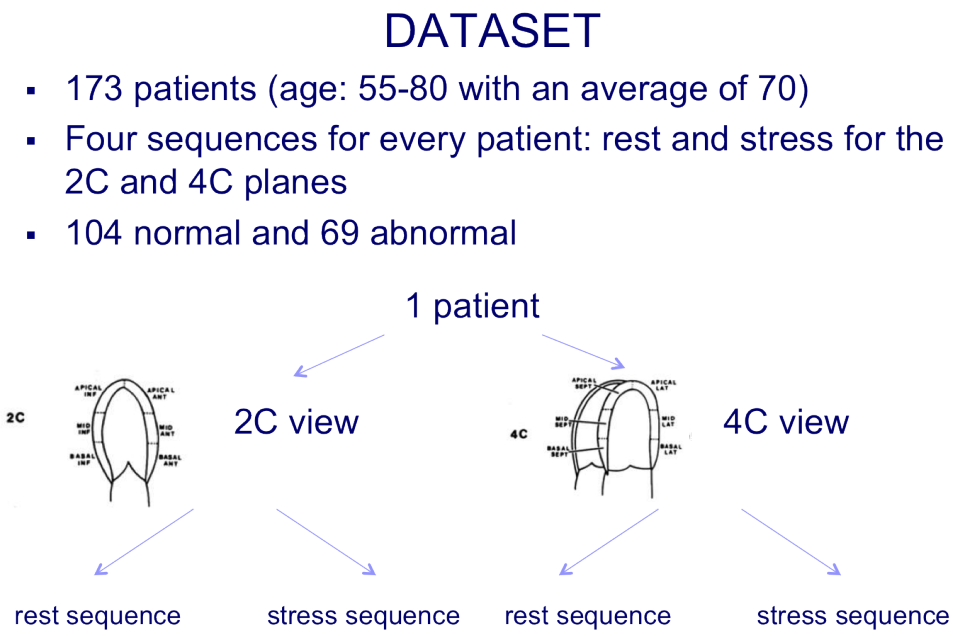


Figure 3.2: Patient dataset structure.

tributions of a pair of features $p(f_1)$ and $p(f_2)$ for normal and abnormal cases might not be sufficient to separate the two classes, estimating the conditional distributions $p(f_1|f_2)$ or $p(f_2|f_1)$, or the joint distribution over all features, could allow separation of the two classes in the high, potentially infinite, dimensionality of the RVM / SVM feature space, and hence overcome the limitations of the HMM (15; 16).

3.3 Database

The database used for this work was acquired by Mansor et al. (112), and contains contrast B-mode 2D echocardiography loops from 173 patients between ages $55-80$ with an average age of 70, of which 104 and 69 patients have been deemed by clinicians to be normal and abnormal, respectively. Four sequences were recorded for each patient, comprising rest and stress sequences for both the 2C and the 4C planes, Fig. 3.2. The 2D myocardial contours of all frames during systole were extracted using Quamus (Mirada Solutions Ltd., in conjunction with the University of Oxford, UK), a semi-automated segmentation software package. These contours were then corrected and validated by two experienced

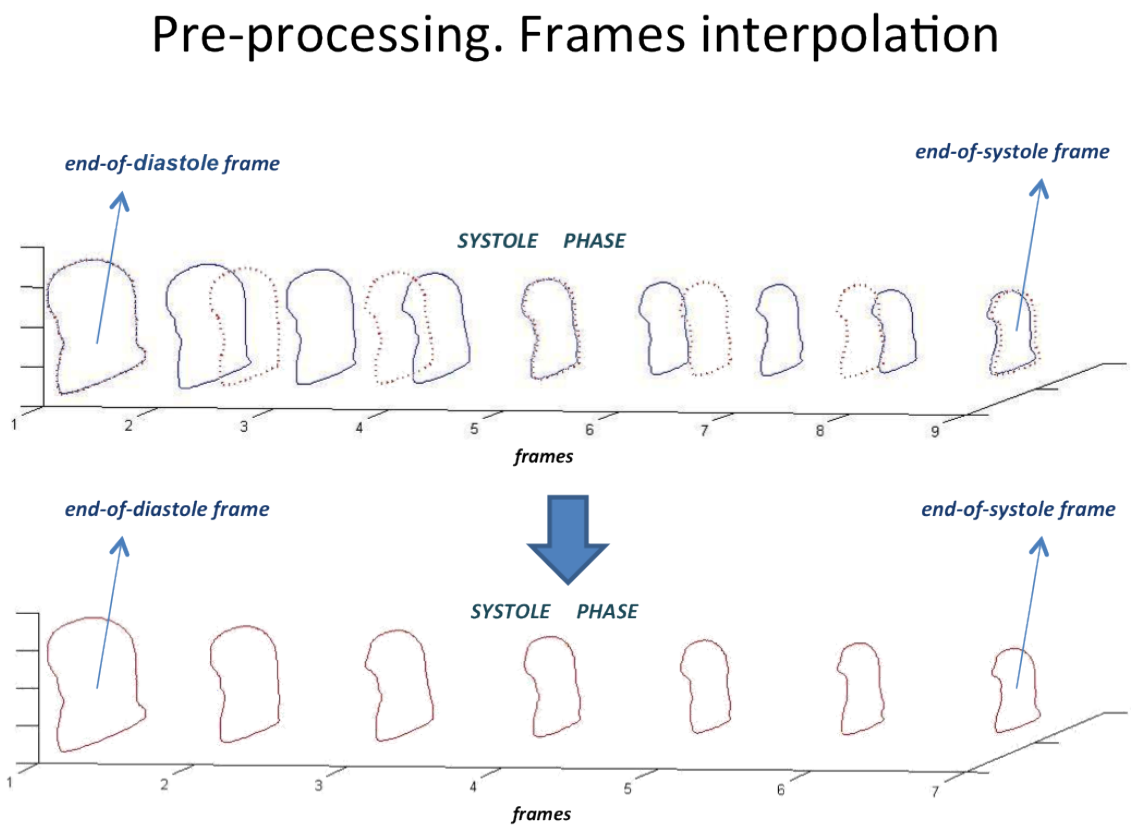


Figure 3.3: Frames interpolation. 7 frames are linearly interpolated from each sequence. The first frame corresponds to end-of-diastole, the last frame corresponds to end-of-systole, and the remaining five frames are interpolated evenly throughout the interval between them.

cardiologists.

For regional functional analysis, the left ventricle is divided into 17 segments but mostly only 16 segments are measurable (113). Clinical wall functional assessment evaluates systolic thickening and endocardial wall motion for each segment. Each of the 16 segments were labelled 1 - 4 (1 = normal, 2 = hypokinesis, 3 = akinetic, 4 = dyskinetic). In this study, we grouped the four labels into 2 classes: 1 is normal, 2-4 are abnormal.

3.4 Pre-processing

All four sequences acquired for each patient (2C and 4C for rest and stress) can contain different numbers of frames due to the difference in the heart rate between patients. Prior to feature extraction, we linearly interpolated seven frames for each sequence such that the

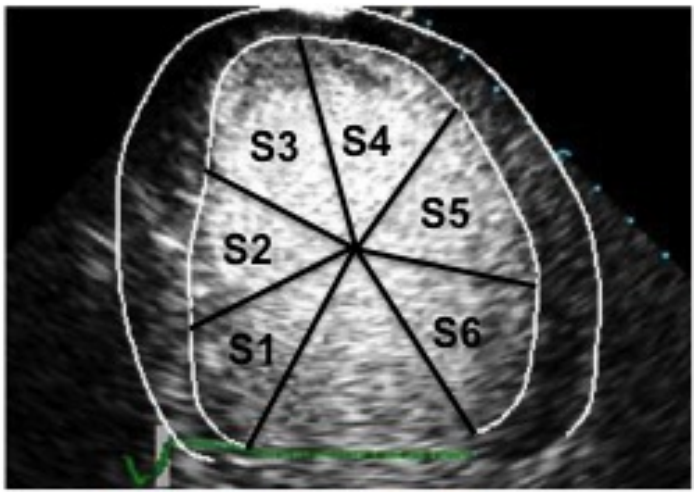


Figure 3.4: Cavity Area for all six segments of 2C data.

first frame corresponded to end-of-diastole, the last corresponded to end-of-systole, and the remaining five frames were uniformly distributed in between. End-of-diastole and end-of-systole frames were found as those corresponding to the maximum and minimum LV area respectively, Fig. 3.3.

3.5 Candidate Features

For each patient, we extracted a set of candidate features from rest and stress sequences and combine these features into a single feature vector, Fig. 3.5. We investigated which of the candidate features corresponded to highest classification accuracy, Fig 3.7.

3.5.1 Cavity Area

As in (112), from each frame, the area inside the delineated contour for global classification and the area inside the segment for local classification were extracted as features, as shown in Fig. 3.4. This area was then normalized with respect to the cavity area in the first frame to ensure that features across all patients were comparable, and independent of heart size. This necessarily resulted in the cavity area in the first frame having unit area, Fig. 3.5. From each sequence we obtained six cavity area features: $[x_1 \ldots x_6]$. Fig. 3.5 shows the cavity area

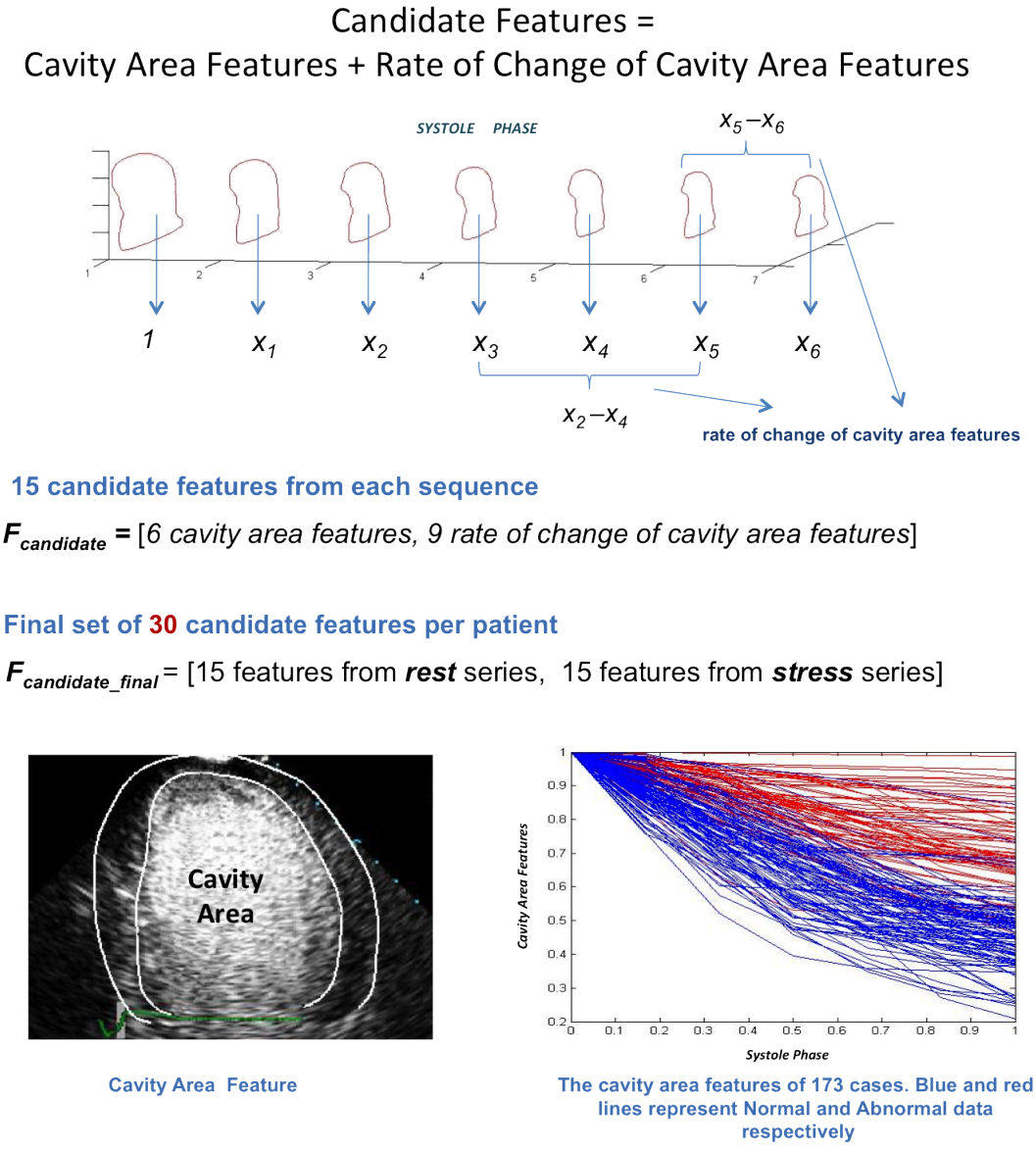


Figure 3.5: Extracting candidate features. The final pool of candidate features comprises 30 descriptors - 15 from rest series and 15 from stress series.

features from interpolated frames over all patients for both 'abnormal' and 'normal' cases. Due to the fact that the cavity area "accumulates" with time just as distance increases with time, during the early phase of heart contraction the cavity area feature did not provide enough information about the class label but served as a good descriptor towards end-systole. This suggested to include the rate of change of cavity area into the candidate features.

3.5.2 Rate of Change of Cavity Area

The derivative of the cavity area represents the instantaneous rate of change of the cavity area. We extracted the average rate of change of the cavity area between pairs of frames, providing another nine candidate features:

$$f_{RateOfChangeCavityArea} = [x_1 - x_2, x_2 - x_3, x_3 - x_4, x_4 - x_5, x_5 - x_6, \tag{3.1}$$

$$x_1 - x_3, x_2 - x_4, x_3 - x_5, x_4 - x_6] \tag{3.2}$$

Therefore, each sequence of frames provided $6 + 9 = 15$ candidate features. Extracting these features from both sequences gave the final set of 30 candidate features:

$$F_{30} = [\ 15 \text{ features from rest series},\ 15 \text{ features from stress series}] \tag{3.3}$$

Each feature was further normalised using a zero-mean and unit-variance transformation, such that each feature varied over approximately the same dynamic range.

3.6 Feature Selection methods

In many cases it is critical to select useful features from the pool of all candidate features. There are two reasons why this might be important:

1. In the case when one needs to understand the problem, it is worth and important to look at the best descriptors of the given problem to understand the roots of the problem.

2. A redundant representation of the object on which an intelligent decision is to be made might lead to a lower performance of the chosen algorithm.

There are three types of methods for feature selection:

1. **Filter methods.** This one relies on individual feature ranking, and therefore such methods are not reliable. Single feature classifiers are normally used, i.e. selecting features according to their individual predictive power. These methods detect linear dependencies between a feature and a target and have the following disadvantages:

a) Looking at features individually the filter method still provides redundant sets. Such methods also reject the features that are not descriptive on their own but jointly with other features.

b) No explicit selection is embedded in the method, thresholding is needed.

2. **Wrapper methods.** An algorithm of choice is used as a black box to evaluate the importance of a set of features for the given task. Here there are two choices to be made:

a) how to search the space of all possible feature subsets (greedy approaches such as forward-backward selection, mean strategy)

b) which 'black box' to use (decision trees, SVM, RLS)

However, the wrapper methods have the following disadvantages:

a) requires various re-training and parameter tuning.

b) not clear what strategy to choose to search the space of feature subspaces.

3. **Embedded methods.** The feature selection procedure is embedded into the training phase. The intuition is that if we keep all the features for training, we might overfit, at the same time a few features may lead to oversmoothing, thus we can embed the number of features in the regulariser of a empirical minimisation problem. We describe the embedded methods in more detail in section 3.8.

In this chapter we investigate the second and the third methods for feature selection problem. For the wrapper methods, we propose to use SVMs and RVMs as classifiers and describe the algorithm for a search of the space of all possible feature subsets using the mean strategy. For the embedded methods, we utilise two algorithms, namely the Lasso algorithm and Elastic Net Regularisation, to select the features, and further employ the Regularised Least Squares (RLS) algorithm on the selected features to estimate the validation error.

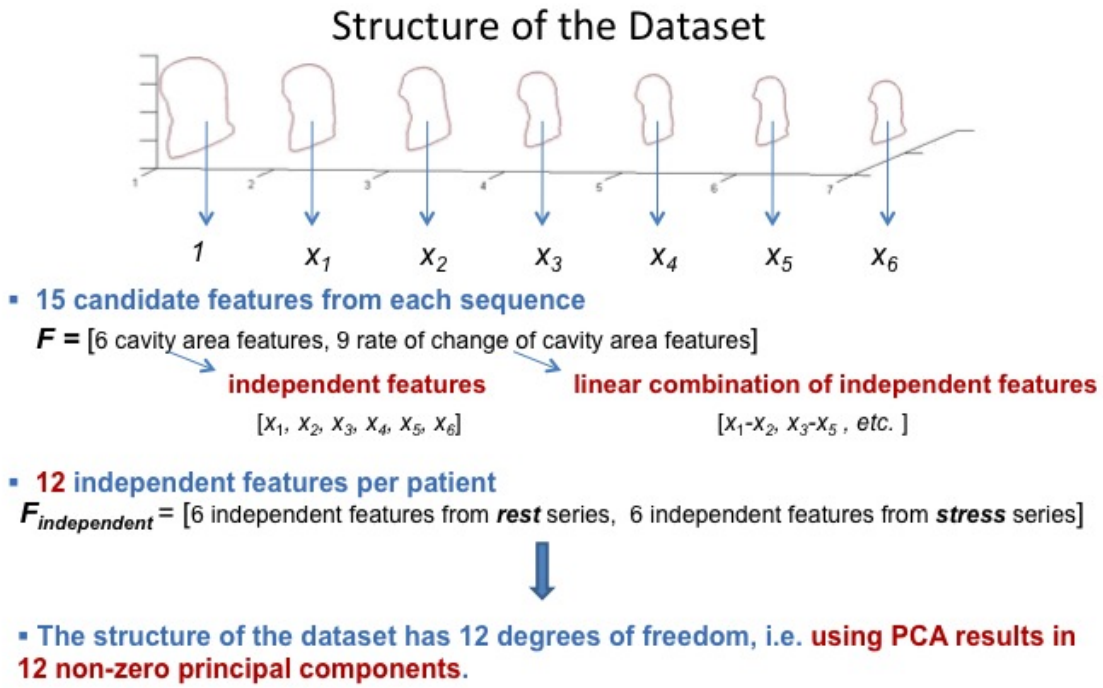


Figure 3.6: Structure of the dataset.

3.7 Feature Selection and Wall Motion Classification using SVM and RVM

After normalizing each feature using a zero-mean and unit-variance transformation, such that each varies over approximately the same dynamic range, we perform classification of heart function using different numbers of features randomly selected from the pool of features F_{30}, Fig. 3.7. For every selection, we perform 5-fold cross-validation; this is repeated 5 times in order to remove any bias introduced during the partitioning of the dataset in its training and validation sets. Thus, the following algorithm is performed:

for different number of features $N = 4$ **to** 30 **do**

  for number of experiments $n = 1$ **to** 100 **do**

    Randomly select N features from the 30 candidates in F_{30}

    Perform 5-fold cross validation 5 times, and calculate the mean error

  endfor

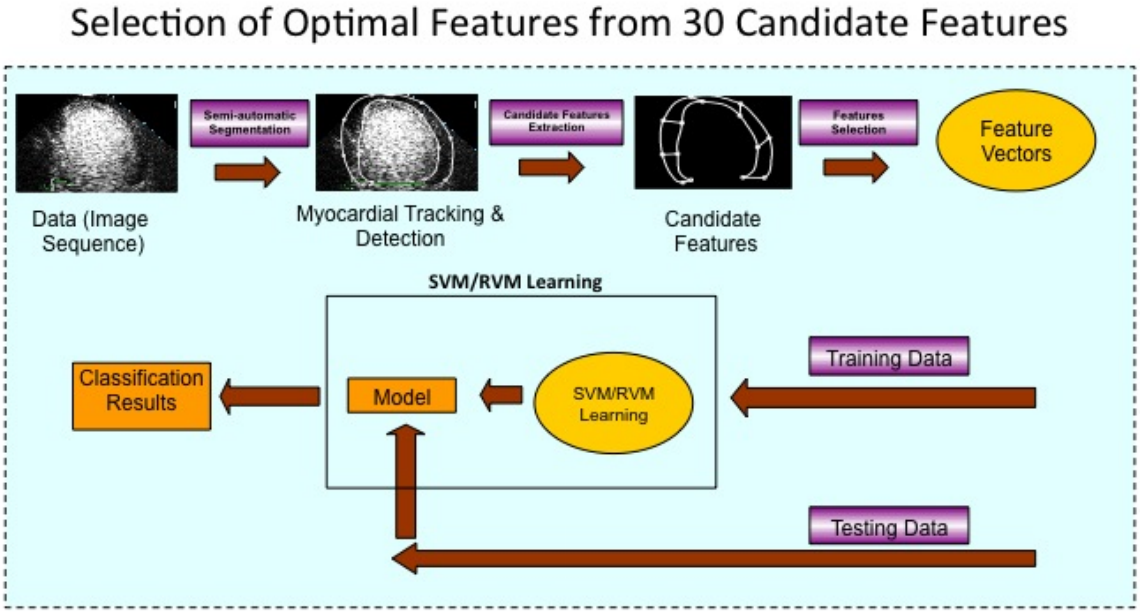
endfor

Fig. 3.7 shows the resulting distribution of classification errors obtained for each value of N and corresponds to the global classification case of 2C view data using combined rest and stress information. In this section we describe the technique on the example of 2C global wall motion classification using combined rest and stress information and SVM/RVM as a classifier. It may be seen from the figure that for $N = 4$ features, the standard deviation of the classification error is at its greatest, which generally decreases with increasing N. This large initial variance indicates that there are selections of $N = 4$ features that perform well, and which result in low classification error. As we increase the number of features, the mean error decreases significantly until $N = 7$ and continues to decrease gradually until $N = 19$, which indicates that including certain extra features after the initial four introduces useful classification information. The error reaches its minimum at $N = 19$, and then slightly increases as N increases to 30.

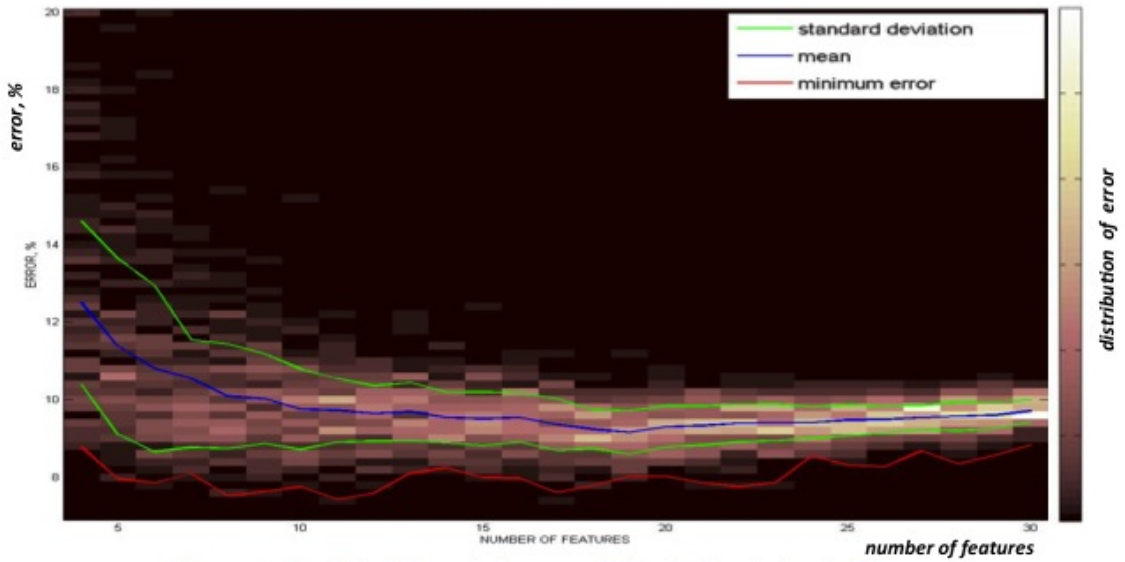
This increase in classification error for $N > 19$ is due to one of two reasons. Either hypothesis (i) there are only 19 features that contain useful information for classification, or hypothesis (ii) the curse of dimensionality results in an increase in the classification error (whereby the dimensionality of the feature space has increased such that there are insufficient data to accurately characterize the difference between classes). In order to determine if the phenomena are explained by hypothesis (ii), we investigated the performance of the SVM/RVM using the principal components of the $N = 19$ and $N = 30$ feature sets.

Using Principal Component Analysis (PCA) for $N = 19$ and $N = 30$ results in 12 non-zero principal components, Fig. 3.6. This shows that our data lie in a 12-dimensional subspace of the original 30-dimensional feature space F_{30}, which is to be expected given that only 12 features out of 30 are independent $(x_1, ..., x_6$ for both rest and stress sequences$)$, while the remainder are linear combinations of those 12 features. This means that the structure of the dataset has 12 degrees of freedom and that it can be fully described by 12 features, which are linear functions of the original 30 features.

In order to find the principal components for $N = 19$ features, we must select the set



for different number of features N = 4 **to** 30 **do**
repeat 100 **times**
Randomly select N features from the 30 candidates in $\boldsymbol{F}_{30}$
Perform 5-fold cross validation 5 times, and calculate the mean error
endfor
plot the distribution of classification error for all N



Mean, standard deviation, minimum, and distribution (colour) of the error.

Figure 3.7: Selection of Optimal Features using SVMs and RVMs. Stage 1.

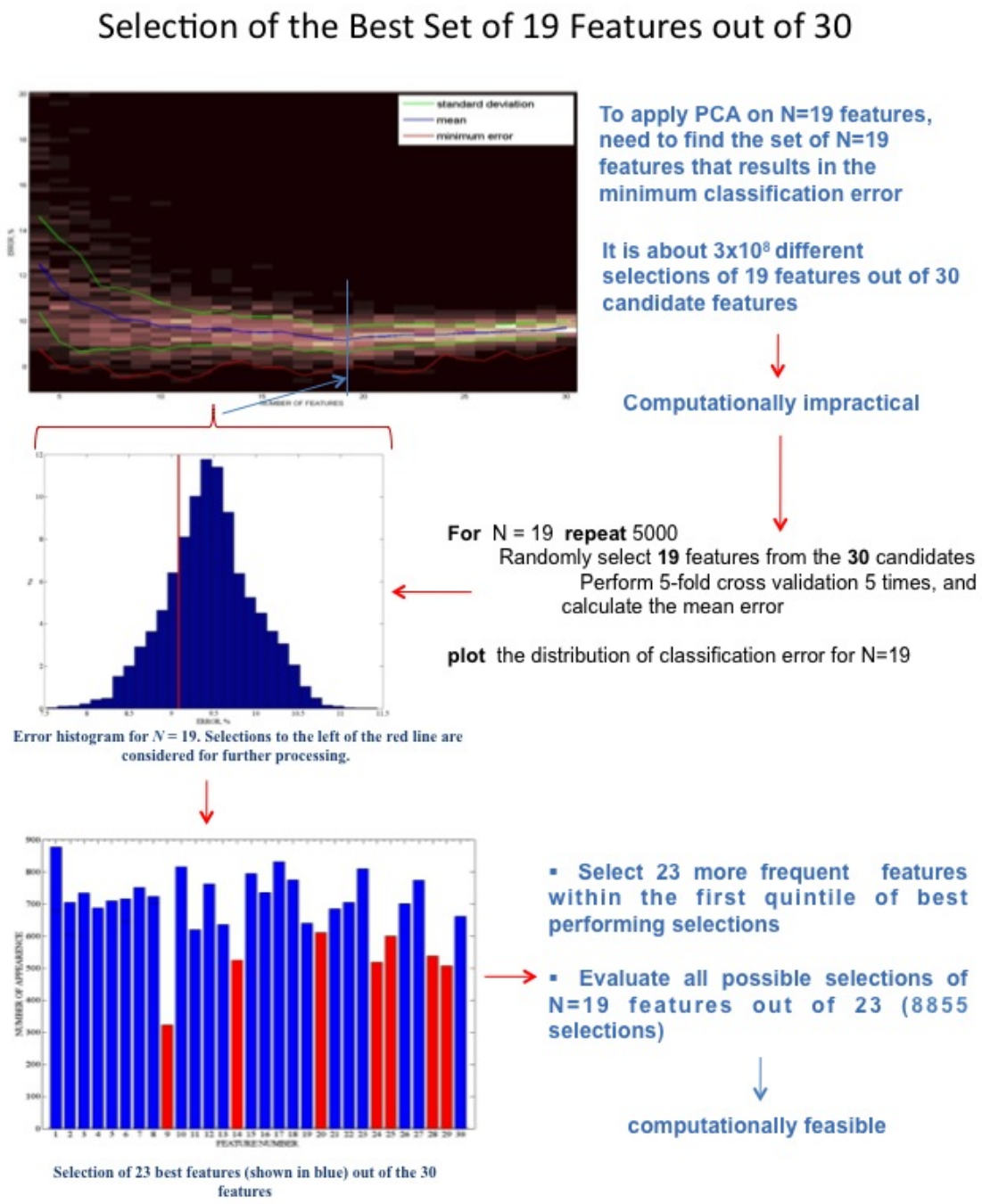


Figure 3.8: Selection of Optimal Features using SVMs and RVMs. Stage 2.

of 19 features F_{19} that results in minimum classification error. Noting that an exhaustive search over the space of all possible selections of 19 features from 30 is impractical (it is over 3^{10} different selections of 19 features out of 30), we performed $5,000$ iterations of a random selection of $N = 19$ features. A histogram of the classification errors over these $5,000$ iterations is shown in Fig. 3.8, which may be seen to be approximately normally distributed. To find the "optimal" selection F_{19}, we took the first quintile of all selections (i.e., $1,000$ selections from $5,000$ total selections) that gave lowest error. Fig. 3.8 shows the relative frequency of occurrence of each of the 30 candidate features within the $1,000$ best-performing sets of 19 features. The figure shows that seven of the 30 candidate features appear less frequently in the $1,000$ best-performing feature selections, while 23 appear more frequently, and are hence deemed to be more useful for classification. We evaluated all possible selections of $N = 19$ features from these remaining 23 features ($8,855$ selections), and therefore found the "optimal" set F_{19} as the one with the minimum classification error.

Having found F_{19}, we compared the performance of the RVM trained using the 12 non-zero principal components of this 19-dimensional feature space with an RVM trained using the 12 principal components of the original 30-dimensional feature space F_{30}. The former outperformed the latter. From this, we conclude that hypothesis (i) explains the increase in error, shown in Fig. 3.7, as the feature space is increased from 19 to 30 dimensions, and we therefore recommend that the "optimal" subset of features F_{19} is used in preference to the set of original candidate features F_{30}.

All figures shown here correspond to the global classification case of 2C view data using combined rest and stress information and the SVM as a classifier. The same procedure was applied to rest classification and stress classification separately, resulting in 21 SVMs and RVMs.

3.8 Feature Selection and Wall Motion Classification using Lasso Regularization, Elastic Net Regularization and Regularized Least Squares

In the previous section we proposed a wrapper feature selection method using the standard classifiers such as an SVM and an RVM to evaluate the importance of a set of features for the given task. Various sets are tested and the one with the best validation error is chosen as the best set of features. In this section, we explore the class of embedded feature selection methods, which embed the feature selection procedure into the training phase. We investigate whether this leads to a better classification accuracy. Two different embedded methods are investigated: the Lasso algorithm (106) and Elastic Net Regularisation (107). These methods are used for feature extraction, and for classification we prefer to use a more reliable classifier and choose a Regularized Least Squares to apply to the selected features. Below we describe our choice of the classifier followed by the two feature selection methods.

3.8.1 Penalized Empirical Risk Minimization and Regularized Least Squares

The Penalized Empirical Risk Minimization looks for a solution to minimize a two term functional:

$$ERR(f) + \lambda R(f) \tag{3.4}$$

where $ERR(f)$ is the empirical error and $R(f)$ is a regularizer. λ trades-off the two terms. The empirical error is denoted in terms of a loss function, i.e. in terms of the price

we pay when we predict $f(\mathbf{x})$ in place of y:

$$\min_f \frac{1}{n}\sum_{i=1}^{n} V(f(\mathbf{x}_i), y_i) + \lambda R(f) \tag{3.5}$$

where n is the number of datapoints.

Things to choose:

a) loss function $V(f(\mathbf{x}), y)$: different loss functions correspond to different algorithms

b) regularization term $R(f)$: encodes some notion of smoothness of the solution f

c) expression for f

In this work, we choose to use the square loss or L_2 loss $V(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$ and Tikhonov regularisation (114):

$$\min_f \frac{1}{n}\sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|^2 \tag{3.6}$$

From the Representer theorem, this can be solved by the following function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i k(\mathbf{x}, \mathbf{x}_i) \tag{3.7}$$

where k is a kernel.

In other words, for each subject (or datapoint, represented by a feature vector $\mathbf{x}$ in the feature space) we construct a kernel k. The final model f is a weighted sum of kernels k.

Combining 3.6 and 3.7, the following minimization problem is defined:

$$min_c \|Y - Kc\|^2 + \lambda c^T K c \tag{3.8}$$

The solution for the above is derived by setting the derivative with respect to c to zero:

$$c = (K + \lambda nI)^{-1} Y \tag{3.9}$$

3.8.2 Lasso Algorithm

Consider the general form of the Penalized Empirical Risk Minimization:

$$\min_f \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i)-y_i)^2+\lambda\|f\|^2 \tag{3.10}$$

To embed the feature selection into the training:

(i) define the function $f(x)$ as

$$f(\mathbf{x})=\sum_i \beta_i \mathbf{x}_i \tag{3.11}$$

Note that the summation is done over the features, not over the datapoints as in 3.7. This defines the space of functions to model the problem.

(ii) choose a different regularization R such that the majority of β_i go to zero, see below.

Generally, the following function should be minimized:

$$min_f \frac{1}{n}\sum_n (<\beta,\mathbf{x}_n>-y_n)^2+\lambda\|\beta\|_0 \tag{3.12}$$

where $\|..\|_0$ is the "zero"-norm, which denotes the number of non-zero β_i, i.e. it is a measure of how many features are important.

However, 3.12 is hard to solve since it is essentially an exhaustive search over all possible sets of features. A natural approximation of the "zero"-norm is given by the $L1$ penalty. Choosing the square residual as the loss function yields the Lasso Regularisation:

$$min_f \frac{1}{n}\sum_n (<\beta,\mathbf{x}_n>-y_n)^2+\lambda\|\beta\|_1 \tag{3.13}$$

where λ is a suitable regularization parameter, and the $L1$ norm is defined:

$$\|..\|_1=\sum_i |\beta_i| \tag{3.14}$$

The Lasso Regularization is solved either by (i) linear (quadratic) programming tech-

niques or (ii) iterative algorithms with convex analysis.

The Lasso algorithm has the following issues:

(i) the solution for 3.13 is not unique - 3.13 is convex but not strictly convex

(ii) if there are any correlated features, the algorithm chooses just one of them

3.8.3 Elastic Net Regularisation

To cope with the aforementioned problems of the Lasso Regularization, one can consider the following penalty term:

$$p_{elastic}(\beta) = \sum_i (|\beta_i| + \varepsilon\beta_i^2) \tag{3.15}$$

The coefficient vector β is estimated by the following Penalized Empirical Risk Minimization:

$$\min_f \frac{1}{n}\sum_n (<\beta, \mathbf{x}> - y_n)^2 + \lambda p_{elastic}(\beta) \tag{3.16}$$

where λ is the regularisation parameter found by cross-correlation, ε controls the amount of smoothness.

The main differences between the Lasso Regularization and the Elastic Net Regularization algorithms are:

a) The L1 term promotes sparsity whereas the L2 term smoothness

b) The Elastic Net Regularisation functional is strictly convex and the solution is unique

c) The whole group of correlated features is selected (Elastic Net) rather than just one feature in the group (Lasso).

Note that the function f is chosen to be a linear combination of the features (equation 3.11), i.e. it is a hyperplane in the feature space, and clearly cannot model the high non-linearity of the data. For this reason, in practice, a two-step procedure is usually adopted:

(i) the Lasso or the Elastic Net algorithm to select the relevant features

(ii) followed by the standard classification (regression) technique. In this work for this

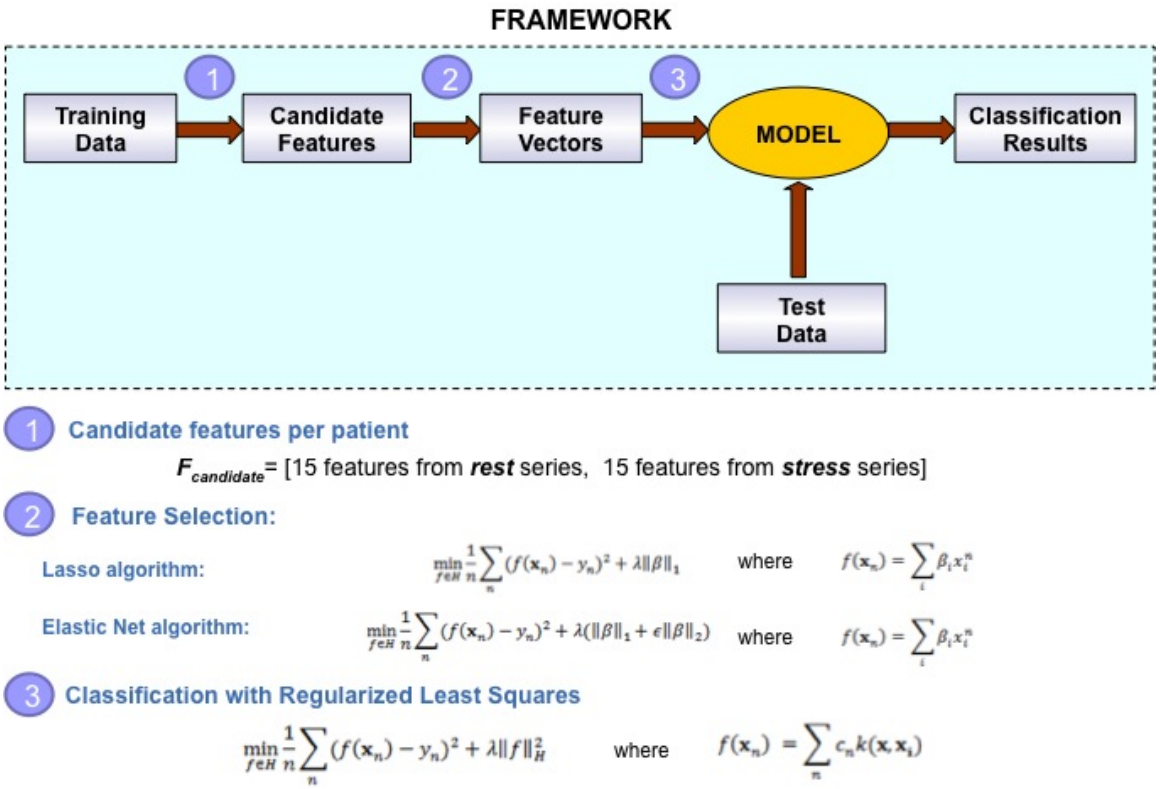


Figure 3.9: Workflow for feature selection and wall motion classification using Lasso Regularization, Elastic Net Regularization and Regularized Least Squares.

purpose Regularized Least Squares (described in 3.8.1) was deployed. This process is summarised in Fig. 3.9.

3.9 Experiments

Experiments were carried out independently for 2C and 4C planes as they describe anatomically different parts of the heart. For each of the planes we performed global and local classification of heart function as normal or abnormal from rest sequences, stress sequences and rest-and-stress sequences resulting in 42 SVMs, 42 RVMs, 42 RLS with Lasso and 42 RLSs with Elastic Net, the results of which (classification accuracy) are shown in Tables 3.1, 3.2, 3.3, 3.4. Each of the 42 SVMs, 42 RVMs, 42 RLS with Lasso and 42 RLSs with Elastic Net was the result of using the feature selection procedure described above. All figures shown in this chapter correspond to the global classification case of 2C view data

using combined rest and stress information.

In general the feature selection algorithms using SVMs and RVMs tended to choose more features compared with the Lasso algorithm and Elastic Net Regularisation. The range of selected features for the SVM classifier was from 17 to 23 features, with the average of 20 features; for the RVM classifier - 15 to 22 features, with the average of 18 features. The Lasso algorithm had a range of 3 to 7 selected features, with the average of 5 features, and Elastic Net Regularization had a range of 5-15 selected features, with the average of 7 selected features. Most of the selected features were those extracted towards end-systole.

Each of the four methods had two free parameters that had to be found by cross-validation: σ of the Gaussian kernel for each of the methods, and the regularisation parameter C for the SVM (eq. 2.40), a threshold value for the posterior probability in the RVM τ, which serves as the decision making parameter, and the coefficient λ for the Regularized Least Square, which controls a trade-off between the regularisation term and the empirical error in (eq. 3.6). To find these parameters we performed 5-fold cross-validation 5 times for each of the 168 experiments on the available 173 subjects. The data was randomly split into 5 non overlapping groups. One of the five groups was left out for validation and the remaining four groups for training, the procedure was repeated five times, selecting different folders for validation and the remaining folders for building the classifier. The optimal value of the parameter to be found corresponded to the minimum average validation error.

Aside from the aforementioned coefficients, there was one more parameter to be found during the feature selection procedure using the Lasso algorithm - λ_{Lasso}, and two more parameters during the feature selection procedure using Elastic Net Regularisation - $\lambda_{Elastic}$ and $\varepsilon_{Elastic}$. These parameters were found following the same cross-correlation procedure described above. The minimum average validation error corresponded to the optimal parameter value and determined the set of selected features. Examples of feature selection using the Lasso algorithm and Elastic Net Regularization for the global case of 2C view

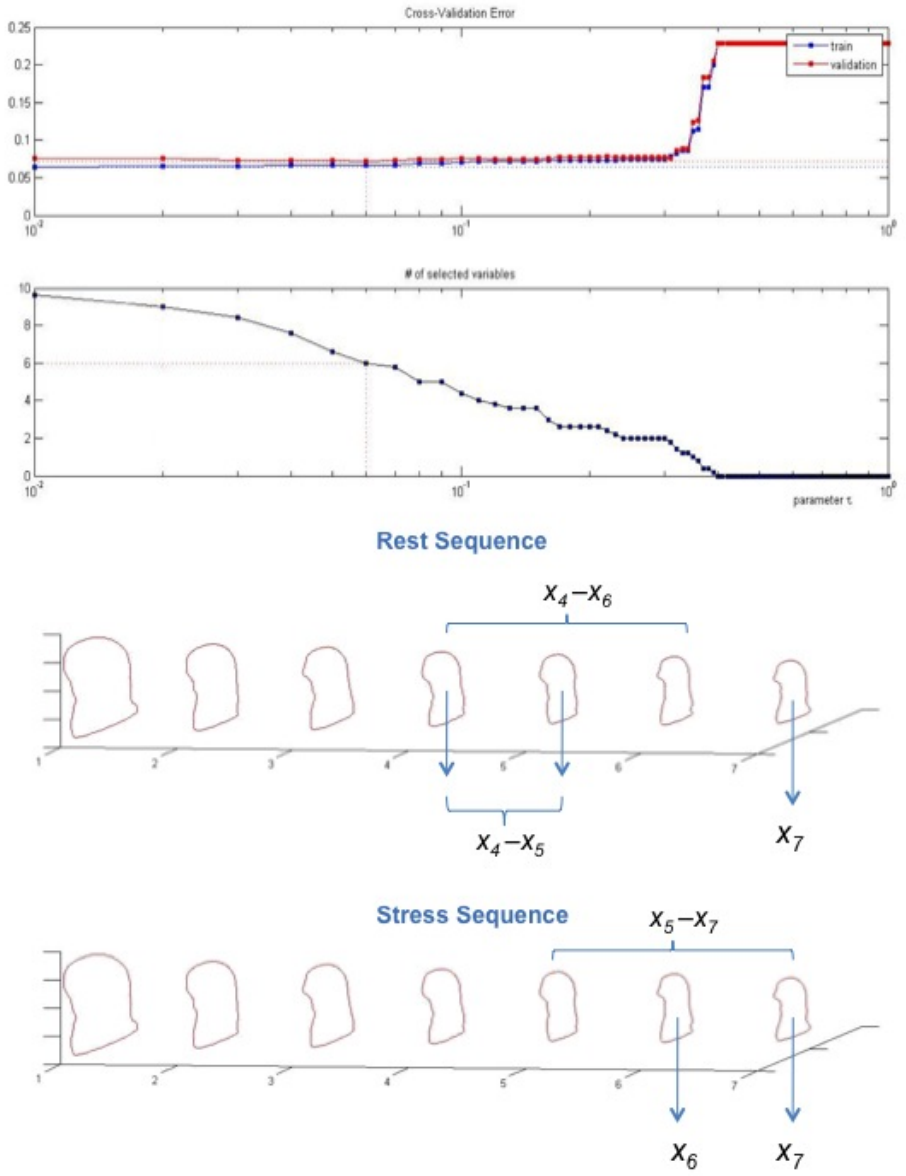


Figure 3.10: Feature selection using Lasso algorithm. Global case, 2C view data.

data are shown in Fig. 3.10 and Fig. 3.11. The selected features were further employed in the Regularized Least Square classifier, yielding results in the next section.

3.10 Results

The results of the 5-fold cross-validation for each of the feature selection and classification techniques - 42 SVMs, 42 RVMs, 42 RLS with Lasso and 42 RLSs with Elastic Net - are reported in Tables 3.1, 3.2, 3.3, 3.4. The four algorithms yield approximately the same

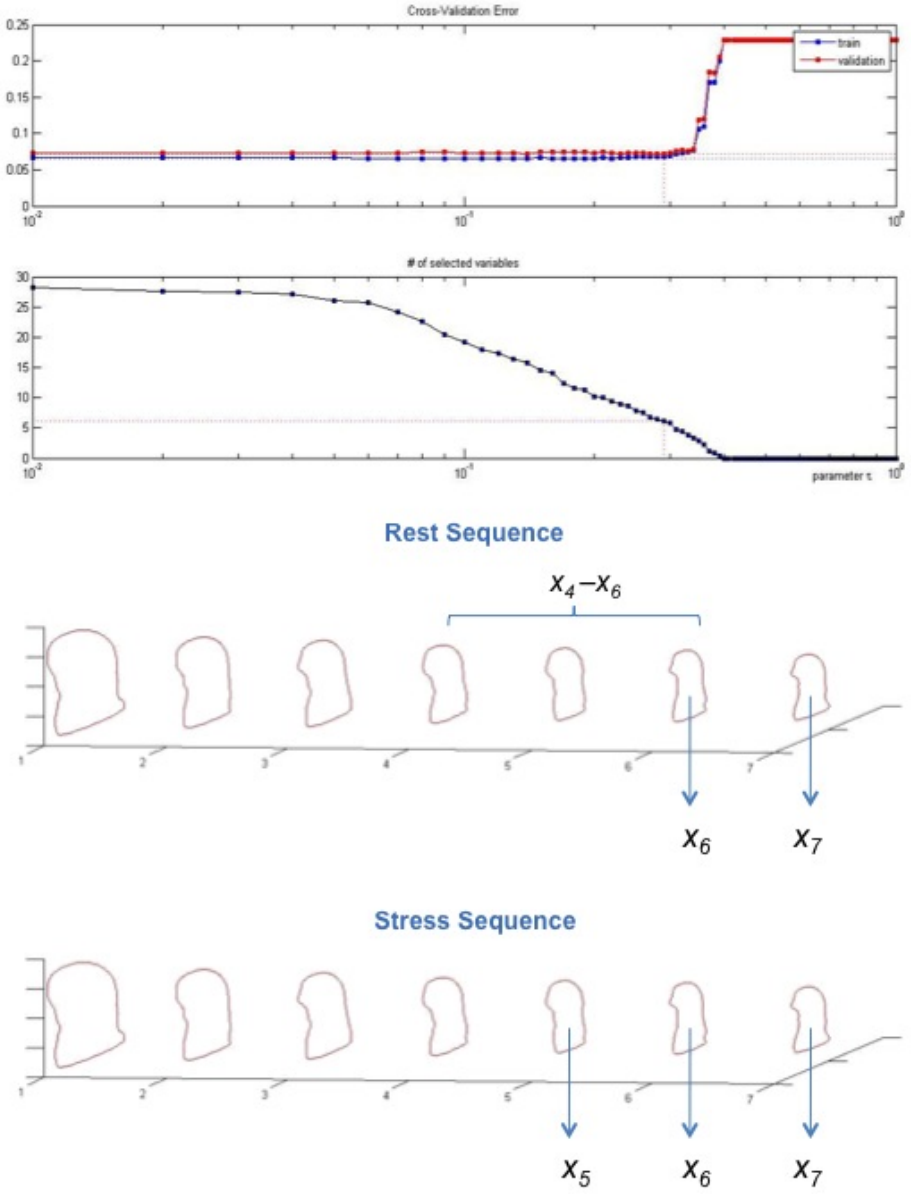


Figure 3.11: Feature selection using Elastic Net Regularization. Global case, 2C view data.

results, with SVMs and Elastic Net Regularisation giving a slightly better overall accuracy compared to RVMs and the Lasso algorithm.

	SVM					
	2C			4C		
SEGMENT	REST	STRESS	REST-STRESS	REST	STRESS	REST-STRESS
GLOBAL	87.18	91.34	92.47	82.38	87.43	88.07
S1	86.61	79.85	87.09	85.14	80.60	85.14
S2	76.89	82.87	84.49	71.67	82.38	83.15
S3	84.45	87.78	91.18	75.67	78.02	78.02
S4	85.56	88.15	89.12	85.14	80.60	88.74
S5	81.21	83.34	87.20	83.15	82.38	86.91
S6	73.12	79.44	84.14	69.70	75.34	77.65
MEAN	82.15	84.68	87.96	78.98	80.96	83.95

Table 3.1: Global and local classification accuracy for 2C and 4C views using SVMs.

	RVM					
	2C			4C		
SEGMENT	REST	STRESS	REST-STRESS	REST	STRESS	REST-STRESS
GLOBAL	88.56	91.34	93.02	79.54	84.74	86.61
S1	82.87	72.22	88.56	74.76	72.64	78.56
S2	81.21	84.57	88.56	71.54	74.49	83.44
S3	82.87	89.98	91.05	76.23	82.98	87.00
S4	79.42	92.10	88.56	76.23	85.55	87.80
S5	87.45	89.98	87.00	73.77	82.98	75.07
S6	76.32	84.57	83.44	70.36	72.64	70.6
MEAN	81.54	86.39	88.60	74.63	79.43	81.30

Table 3.2: Global and local classification accuracy for 2C and 4C views using RVMs.

3.11 Discussion

The technique described above determines the subset of features selected from the full set of candidate features that are useful for improving classification performance. For each data point in the feature space (where each such data point represents both rest and stress cardiac sequences from one patient), the RVM, the SVM and the RLS calculate a corresponding kernel weight w_i. For the SVM and the RVM, most of the weights w_i tend towards $\sim N(0,0)$ and so the kernel functions that correspond to these weights do not play a role in classification and are not used in computation of the objective function, resulting in a sparse model. This corresponds to the automatic selection of a subset of the available patients sequences as being useful for the task of classifying cardiac sequences as being normal or abnormal. The technique determines relevant patient-sequences, and subsequent

	Lasso Algorithm and Regularized Least Squares					
	2C			4C		
SEGMENT	REST	STRESS	REST-STRESS	REST	STRESS	REST-STRESS
GLOBAL	87.60	91.34	93.52	80.02	83.42	87.22
S1	85.70	77.34	89.06	81.86	78.57	86.50
S2	81.19	85.70	87.69	72.40	80.96	82.34
S3	84.76	90.08	91.35	74.13	80.81	82.12
S4	86.33	87.30	88.06	82.76	84.44	87.98
S5	86.55	90.96	87.06	79.38	81.78	80.17
S6	77.12	84.54	82.14	72.13	79.11	78.40
MEAN	84.18	86.75	88.41	77.53	81.30	83.53

Table 3.3: Global and local classification accuracy for 2C and 4C views using Lasso algorithm and Regularized Least Squares.

	Elastic Net Regularization and Regularized Least Squares					
	2C			4C		
SEGMENT	REST	STRESS	REST-STRESS	REST	STRESS	REST-STRESS
GLOBAL	89.79	93.42	94.52	78.12	88.55	89.21
S1	83.37	75.15	89.60	78.60	75.63	78.60
S2	86.10	88.82	87.96	76.38	79.90	81.04
S3	84.37	90.85	90.25	77.15	85.01	85.01
S4	80.02	91.54	89.49	80.34	83.76	83.03
S5	84.05	90.87	84.70	77.77	81.85	79.77
S6	77.24	83.45	83.45	76.06	78.14	78.76
MEAN	83.56	87.73	88.57	77.77	81.83	82.20

Table 3.4: Global and local classification accuracy for 2C and 4C views using Elastic Net Regularization and Regularized Least Squares.

classification of a new patient sequences is performed based on the information from the subset of useful training patients. Furthermore, the RVM calculates the posterior probabilities of a data point belonging to each of the classes, allowing us to define a threshold on the probability. This enables us to avoid making a classification (taking a dont know decision) in situations where the posterior probability of both classes is too similar to decide into which class the test data lie with any confidence. As opposed to the RVM, SVM's relevant patient-sequences are always examples lying on the boundary and are hard to classify.

As seen from Tables 3.1, 3.2, 3.3, 3.4, the local classification accuracy is lower than that of the global classifier for all cases because sometimes not all segments can be clearly seen in echocardiographic data, especially around the apex area. A further problem with regional wall motion analysis is that an infarct region is not necessarily confined to a single

segment of the LV segment model, and could be extended to adjacent segments. A good accuracy of classification was achieved despite the limitations stated above.

The results could be improved further by developing models for different types of abnormality and increasing the number of training data.

The general approach could further be extended to 3D echocardiography. We note that the methodology is not modality specific and could be equally applied to stress MRI.

Chapter 4

Segmentation of 3D Echocardiograms using Random Forest.

In this chapter we investigate the application of the Classification Random Forest to segment end-systolic 3D echocardiograms. Section 4.1 first describes the challenges of ultrasound image segmentation, the importance of accurate segmentation of echocardiographic volumes and how segmented 3D echocardiograms can be used to evaluate heart function. The method is further described in section 4.2, the validation and the results are presented in sections 4.6 and 4.7.

4.1 Introduction

Automated segmentation of 3D echocardiography provides quantitative information that is used in assessment of cardiac function and detection of cardiac diseases and abnormalities. Segmentation of 3D echocardiograms is a nontrivial task due to ultrasound artefacts such as speckle, signal drop-out, attenuation, shadowing, missing boundaries and similarity in appearance of different tissues. The success of the segmentation depends mainly on the classifier itself and on the set of features used by the classifier. While the Classification Random Decision Forest proved to be a good classifier for image segmentation (77), its performance greatly depends on the features used. This chapter explores the classic fea-

tures used in image segmentation such as Rectangular3D features, Difference3D features, Haar3D features (86; 84; 83) and position features (76).

4.2 Random Decision Forests for Classification

The problem of 3D echocardiogram segmentation with the Decision Random Forest is formulated here as a voxel classification problem - each voxel is treated as a datapoint and the classifier labels each datapoint as belonging to one of the classes. In Fig. 4.1 datapoints with different colours indicate different training labels. Each training datapoint is denoted as a pair $(\mathbf{p}, c)$, where $\mathbf{p}$ is a multi-dimensional vector of features $\mathbf{p} = (p_1, p_2, ..., p_n)$ and c is a class label, $c \in (c_1, c_2, ..., c_k)$. Testing datapoints which are not available during training are in a green colour, i.e. their class is not known and will be identified by the classifier. Training requires optimising the energy function described in section 4.2.1.2. During testing, the class label of each input datapoint $\mathbf{p}$ is inferred using the leaf node predictor $p(c|\mathbf{p})$.

For 3D echocardiography segmentation, three classes are used in this work to distinguish the blood pool, the myocardium and the background (everything but the LV).

4.2.1 Training the Classification Random Decision Forest

The general framework of the Random Decision Forest training procedure is described in section 2.4. The adaptation to the classification problem consists of defining the classification specific energy function I and the leaf prediction model $p(c|\mathbf{p})$. The training procedure starts at the root node with all the training data S_0, Fig. 4.1. The root node splits the training data S_0 into two parts S_1 and S_2 by minimising the classification specific energy function I, and further sends it to the child nodes. The parameters of the split function are stored at the root node. The procedure continues until the stopping criteria is met (section 2.4.4). The parameters of the prediction model $p(c|\mathbf{p})$ are stored at the leaf nodes.

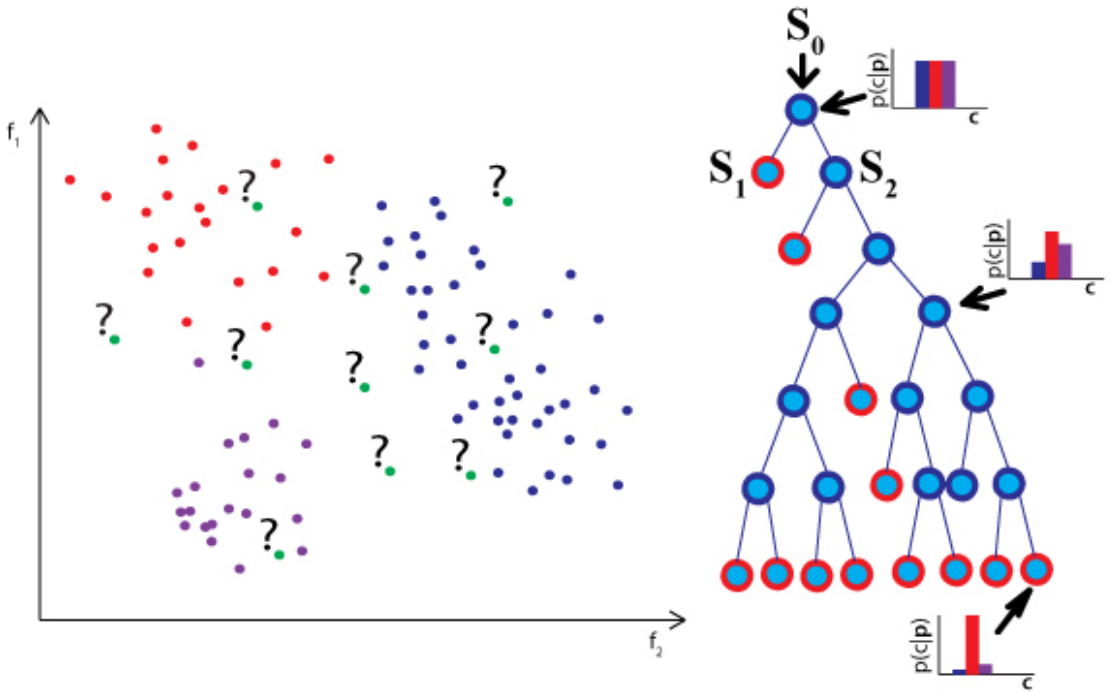


Figure 4.1: Decision Ransom Forest. Testing.

4.2.1.1 Split Function

A split function defines the hypersurface that splits the data in the feature space, section 2.4.4.1. The linear splitting is defined as follows:

$$\mathbf{h}(\mathbf{p}, \theta_j) = \lfloor \phi_j(\mathbf{p}) \cdot \psi < \tau \rfloor \tag{4.1}$$

In this work, an axis-aligned hyperplane is used as a split model. A hyperplane is axis-aligned when the line ψ is aligned with one of the axes in the feature space.

4.2.1.2 Energy Model

Development of energy models is an active research area. Here we discuss the basic building blocks of the common choice of the energy function in decision forests - entropy and information gain, and explain their meaning in regard to the decision forests.

Entropy. Entropy is a measure of the uncertainty about a random variable. It is the expected value of the information contained in a message delivered by a variable:

$$H(Y) = -\sum_{i=1}^{n} p(x_i) \log \frac{1}{p(x_i)} \tag{4.2}$$

For example a coin toss (a series of coin tosses) with a fair coin has the maximum entropy since there is no way to predict the outcome of the toss. A coin toss with identical sides coin has zero entropy since it is certain that the coin will always come up the same outcome. Real data collections are usually somewhere in between.

Information Gain. Information gain is a measure that quantifies the reduction in uncertainty about a random variable Y after a random variable X takes the value $X = x$:

$$I(Y|X = x) = H(Y) - H(Y|X = x) \tag{4.3}$$

where $H(Y|X = x)$ is called conditional entropy.

It is important to see that X is a random variable and x is a certain value of this variable and that $H(Y|X = x)$ and $H(Y|X)$ are fairly different entropies. However, the term $H(Y|X)$ is often used in the literature to denote $H(Y|X = x)$. In this thesis, $H(Y|X)$ is used to denote $H(Y|X = x)$ unless stated otherwise.

The information gain can be intuitively interpreted as follows: entropy $H(Y)$ is uncertainty about random variable Y, $H(Y|X)$ is uncertainty about random variable Y remaining after X is known, i.e. it is a measure of what X does not say about Y. The information gain then can be read as amount of uncertainty about Y minus amount of uncertainty about Y that is left after X is known. This is equivalent to the amount of uncertainty in Y that is removed by knowing X.

In the classification decision trees framework, the information gain can be read as the amount of uncertainty about the class c that is removed after the split of the data. Since the parameter vector $\boldsymbol{\theta}$ defines the split at the node, the variable Y in equation 4.3 is the class

label c, and X is the parameter vector θ. Maximising the information gain is equivalent to minimising uncertainty about the class label c. Thus, the information gain is defined:

$$I(c|\theta) = H(c) - H(c|\theta) = H(c) - \sum_{i=\{left,right\}} \frac{|S_i|}{|S|} H_{S_i}(c|\theta_i) \tag{4.4}$$

where S_i is the data at the left or right child node, $|.|$ denotes the number of datapoints.

In the case with the axis-aligned hyperplane split function, the split is defined by choosing a feature (done by the function ϕ) and a threshold τ. Thus, at a node j, for each chosen feature f, different values of threshold τ are tested. The best combination of a feature f and a threshold τ for this feature, i.e. the combination that maximises the energy function, is saved at the node j.

4.2.2 Leaf Node Prediction Model. Ensemble Prediction Model. Testing the Classification Random Decision Forest.

Growing the tree continues until one of the stopping criteria for splitting is reached. The stopping criteria are described in section 2.4.4. The class distribution histogram is saved at each leaf node yielding the leaf node prediction model $p(c|\mathbf{p})$. During testing, the datapoint $\mathbf{p}$ traverses the trees and falls into one of the leaf nodes at each tree. Thus, each tree t produces a posterior distribution $p_t(c|\mathbf{p})$, which is associated with the leaf node where the datapoint resides. These distributions are put together into the final forest posterior:

$$p(c|\mathbf{p}) = \frac{1}{T} \sum_{t=1}^{T} p_t(c|\mathbf{p}) \tag{4.5}$$

Fig. 4.2 illustrates the testing procedure with a three-tree forest.

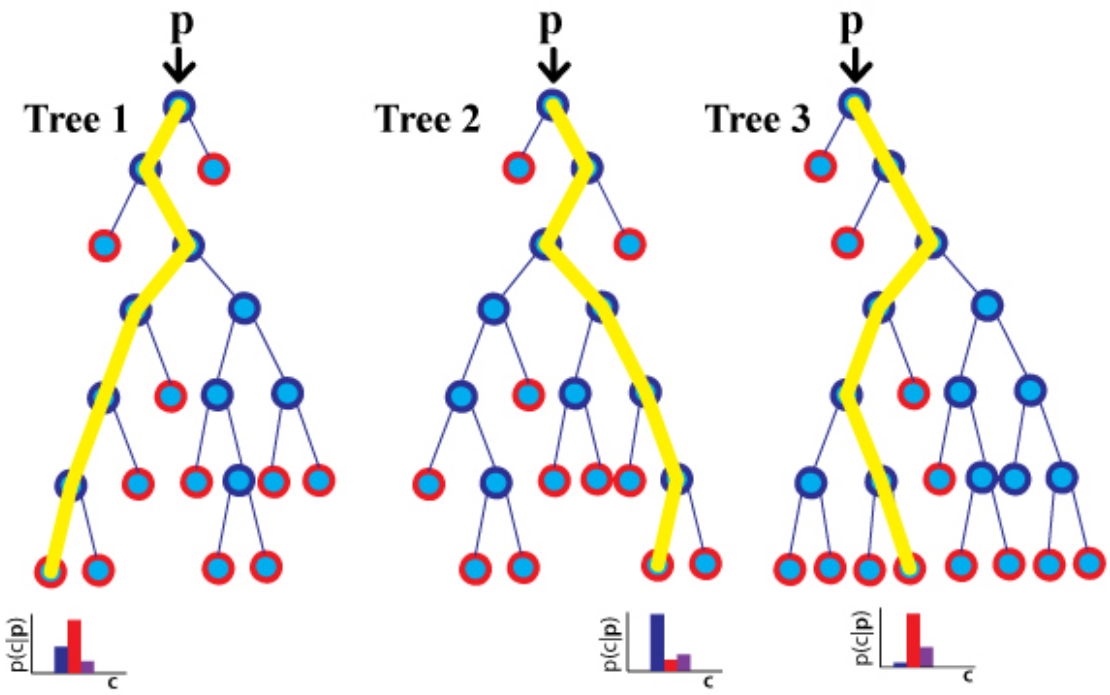


Figure 4.2: Decision Ransom Forest. Testing. The datapoint **p** traverses each tree in the forest collecting the posterior distributions from the leaf nodes where it ends up. These distributions are further aggregated to produce the final forest posterior.

4.3 Feature Sets

For 3D echocardiography segmentation with the Classification Decision Random Forest we adopted several classic low level features that showed promising results in different medical image segmentation and detection problems (86; 102). Cuboid3D, Haar3D, Difference3D (83; 115; 100; 116) and position features (76) were used.

Rectangular3D and Difference3D Features. The sum of intensities of a rectangular parallelepiped in the vicinity of the voxel constitutes the Rectanglar3D feature. This feature is defined by the following:

1. choose the offset (x_1, y_1, z_1) from voxel $\mathbf{p} = (x, y, z)$, where the centre of the rectangular parallelepiped lies.

2. choose dimensions of the parallelepiped (i, j, k).

In practice, the integral image is precomputed to speed up the calculation of the Rectanglar3D feature.

The Difference3D feature is the difference of two Rectangular3D features, Fig. 4.3. Since the Difference3D feature captures more information about the neighbourhood of the voxel **p**, it is usually appears to be stronger compared to the Rectangular3D feature. The acceptance of features during the training procedure of the forest is shown in Fig. 4.11.

Haar3D Features. The Haar3D feature is very similar to the Rectangular3D feature in that both use intensities in the rectangular parallelepiped in the neighbourhood of the voxel **p**, Fig. 4.3. The Haar3D feature halves the rectangular parallelepiped and takes the difference of the sums of intensities in each half. Thus, the Haar3D feature is defined as follows:

1. choose the offset (x_1, y_1, z_1) from voxel $\mathbf{p} = (\mathrm{x}, \mathrm{y}, \mathrm{z})$, where the centre of the rectangular parallelepiped lies.

2. choose dimensions of the parallelepiped (i, j, k).

3. choose how to split the parallelepiped in half and subtract one half from another.

The sum of intensities of individual halves is calculated using the integral image.

Position Features. Following the work in (76), we employ a position3D feature - one of the coordinates of the voxel - to capture the absolute position of the voxels in the image. The position features are high-level features and enforce some shape priors on the object of interest.

4.4 Implementation

The implementation of the Random Decision forest was done in C++. The organisation of the program and pseudocode are presented in Fig. 4.5, Fig. 4.4 and Fig. 4.6.

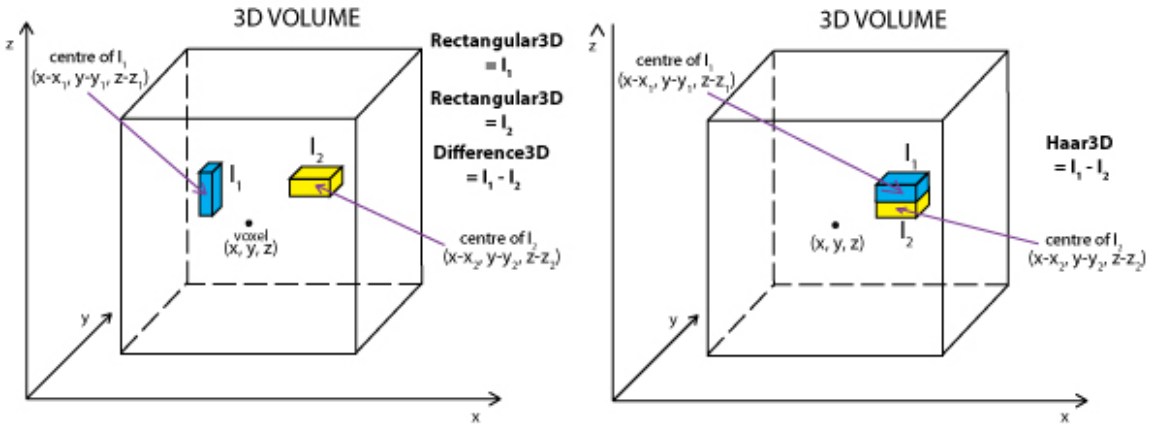


Figure 4.3: Rectangular3D, Difference3D and Haar3D features. The Rectangular3D feature for a voxel **p** is defined as the sum of intensities in the rectangular parallelogram in the vicinity of the voxel **p**. The Diffference3D feature is a difference of two independent Rectangular3D features. The Haar3D feature is the difference of the sum of intensities in two adjacent rectangular parallelograms in the neighbourhood of the voxel **p**.

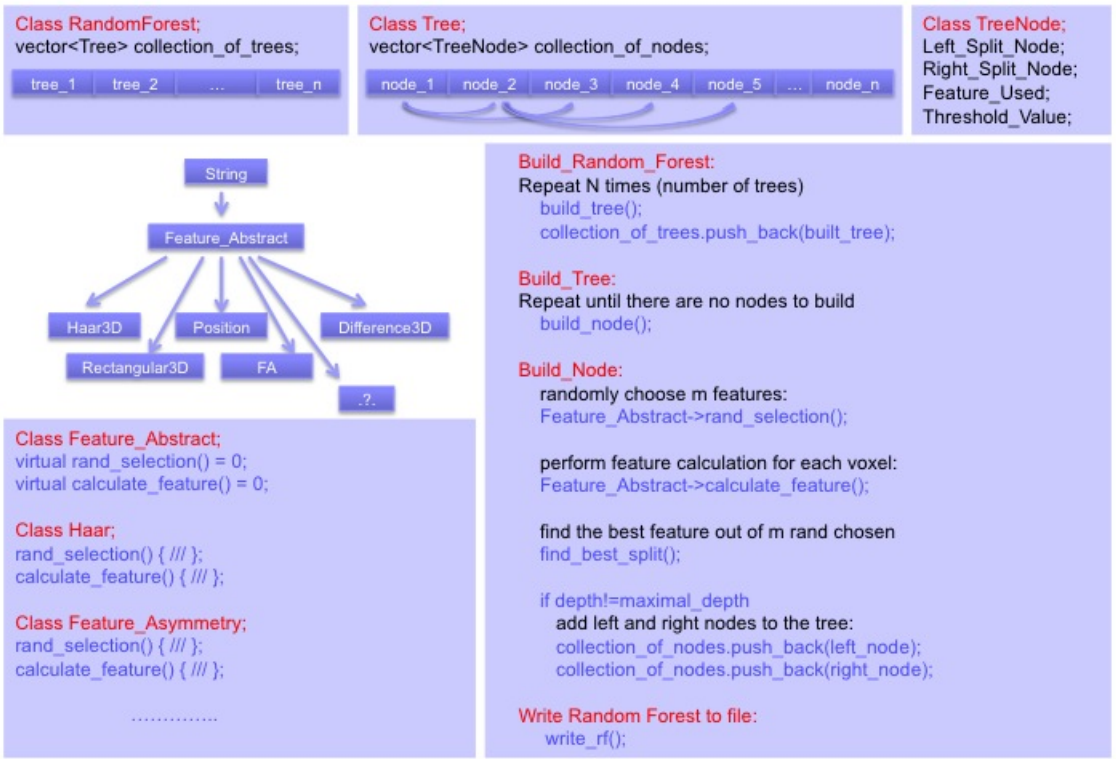


Figure 4.4: Implementation outline of the Random Forest.

```
Input:
-   Testing images H.
-   Segmented images.
_____________________________________________________________________
Extract testing datapoints S : (s1, s2, …, sk) from images H, where si
is a ith voxel in the image,

Function TestForest(trainedForest, S)
  For k=1 to K
    For t = 1 to T
      Distribution(k) =+ TestTree(sk, rootNode(t));
    End For
    Distribution(k) = Distribution(k)/T;
  End For
End TestForest
_____________________________________________________________________
Function TestTree(s, node)
  If node==leafNode
     Break & return leaf class distribution;
  Else If node.TestFunction(s) == Left
     TestTree(s, node.Leftnode);
  Else
     TestTree(s, node.Rightnode);
  End If
End TestTree
```

Figure 4.5: Pseudo Code for the testing procedure of the Classification Decision Ransom Forest.

4.5 Understanding the Classification Random Decision Forests

Regardless of the simple structure of the Random Decision Forest, the technique is mostly used as a black box with little understanding what is happening inside. The visualisation of the Random Decision Forest in Fig. 4.1 is useful to understand the structure of the forest and helps to implement the algorithm in a programming language. However, to appreciate the strength of the Random Decision Forest one may prefer to look at what the technique does in the data domain, Fig. 4.7. During training, each tree in the forest divides the feature space, i.e. the data space, into regions of rectangular (with possible infinite sides) regions such that each of the regions mainly contains datapoints belonging to one class. This is equivalent to minimising uncertainty about the class labels over all the data in all the regions. During testing, each of the trees of the random forest puts a new, previously unseen datapoint, into one of the regions by asking a series of questions about this datapoint. The new datapoint is considered to belong to the class distribution of this

```
Input:
- Training images V and the ground truth GT.
- Parameters:
    • T - number of trees in the forest
    • D - maximum depth of the tree
    • k - number of randomly chosen features at a node to test
    • m - minimum number of datapoints at a split node
Output:
T independent trees with learnt split function at each split node
(feature + threshold). Leaf nodes contain class label distribution.
```

Extract training examples $S : ((p_1, c_1), (p_2, c_2), \ldots (p_k, c_k))$ from images V and the ground truth GT, where p_i is a i^{th} voxel and c_i is its class label, $c_i \in \{c_1, c_2, \ldots c_r\}$

```
Function TrainForest(S, T, D, k, m)
  for t=1 to T
     TrainTree(S, D, k, m);
End TrainForest
```

```
Function TrainTree(S, D, k, m)
  If CurrentTreeDepth==D or CurrentNumberOfDatapoints<m
     Create a leaf node and safe the class label distribution;
  End If

  MaxInformationGain=0;
  For i=1 to k
     Randomly select a feature f_i from the pool of features;
     Find the threshold TH_i that maximizes the information gain I=I_i;
     If I_i>MaxInformationGain
          Set NodeParameters = (f_i, TH_i)
     End If
  End For

  Create a node with NodeParameters;

  Split the set S into S_left and S_right according to the split
  function with NodeParameters;

  TrainTree(S_left, D, k, m);
  TrainTree(S_right, D, k, m);

End TrainTree
```

Figure 4.6: Pseudo Code for the training procedure of the Classification Decision Random Forest.

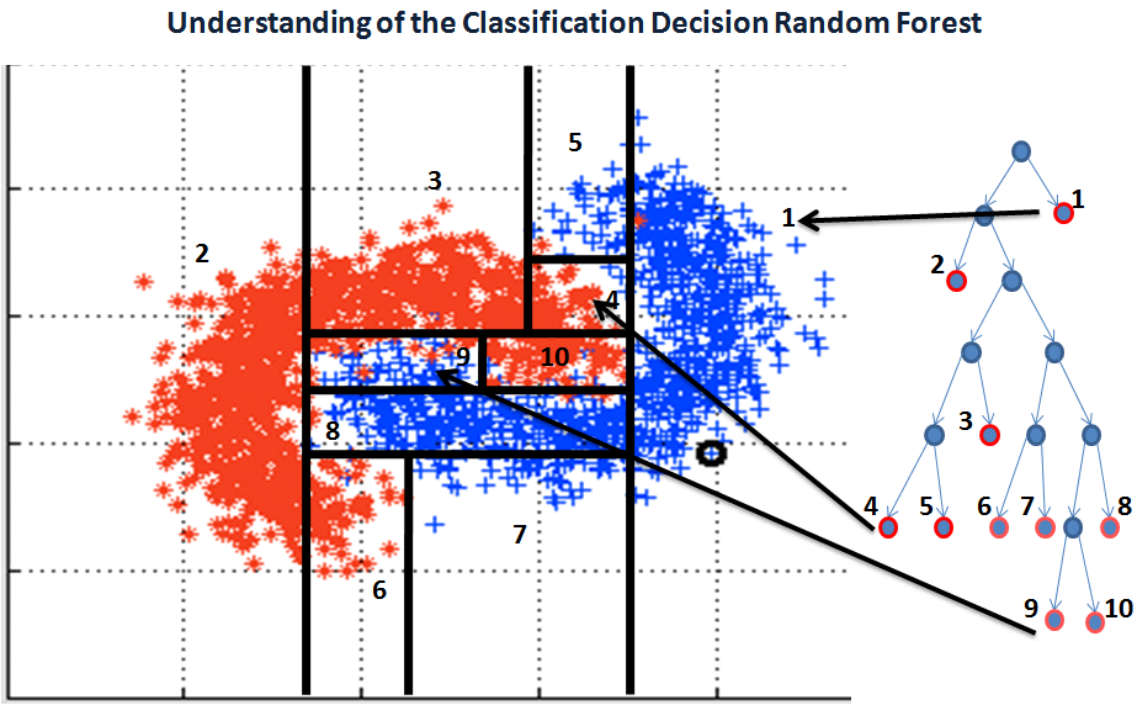


Figure 4.7: Understanding the Classification Decision Random Forest. Each tree in the forest divides the feature space into subregions (1-10 in this example) that predominantly contain datapoints of one class. Each subregion in the feature space is associated with a leaf node. Given a previously unseen datapoint, the tree places it into one of the subregions. The datapoint is then considered to belong to the class distribution of that subregion. Such distributions are collected from each tree and aggregated to produce a single posterior distribution.

region of the feature space. The distributions from different trees are collected and unified by equation 4.5 to produce an overall posterior probability distribution of the class label for the voxel.

4.6 Experiments

25 3D end-diastolic echocardiograms from healthy subjects were used in this study. A Philips iE33 ultrasound system was used to acquire the images. Volume dimensions are (224x208x208) voxels with an average of $0.88mm^3$ spatial resolution. The myocardium and the blood pool of all volumes were manually segmented by an expert.

The dataset was randomly divided into two sets - the training set $S_0 = 20$ images, and the validation set $S_v = 5$ images. The training set was used to train the trees and learn the parameters of the split function at each node, whereas the validation set was used to report

the results and estimate the optimal maximum depth of the trees D and the size of the forest T by applying the cross-validation technique.

Depending on the clinical application, one might be interested in accurate segmentation of either blood pool or the myocardium. For example, to measure the thickness of the myocardium, the segmentation of the myocardium must be as accurate as possible. If the task is to estimate the ejection fraction, the blood pool needs to be segmented as accurately as possible. For the general visualisation purposes and for visual assessment of the cardiac function, both the blood pool and the myocardium segmentation have to be optimum. Thus, three Decision Random Forest classifiers were trained: i) for Blood Pool segmentation ii) for Myocardium segmentation iii) for both the Blood Pool and the Myocardium segmentation. The results of the segmentation are reported in section 4.7.

The Classification Decision Random Forest was implemented in C++. Testing volumes were segmented in 20 seconds per volume using Intel Xeon 2.8GHz computer with 12 cores and 48GB RAM running Win7. With a parallel tree implementation, training required about 1 hour and only needed to be done once.

4.6.1 Model Parameters

The three Decision Random Forests were trained with 20 trees. During the training procedure, an axis-aligned hyperplane was used as the split function (that is equivalent to using a single feature at a node with a single threshold), and the information gain over the class labels was used as the objective function. At each node each of $m = 100$ randomly chosen features was independently used to optimise the split function. The feature f and the threshold τ that maximise the information gain were saved at the node. The minimum number of datapoints used to terminate the tree growth was $M = 50$, the maximum depth of the tree - 25.

For quantitative analysis of segmentation, we utilised the Dice similarity coefficient to

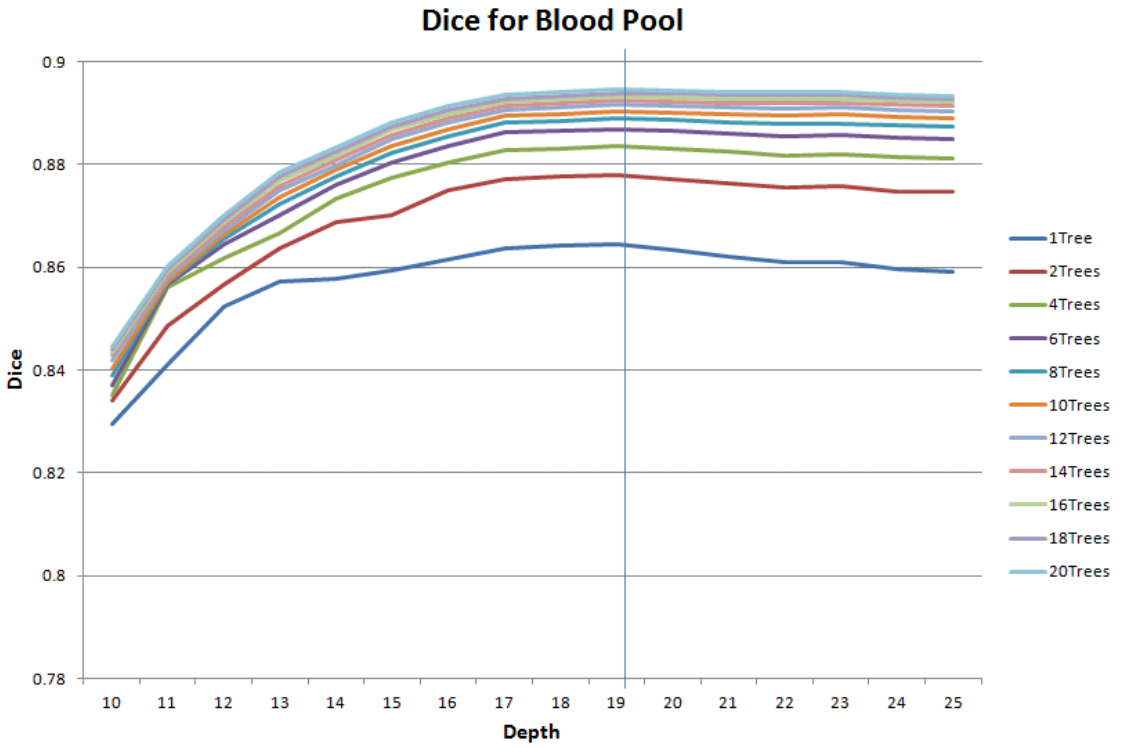


Figure 4.8: Dice similarity coefficients for the blood pool segmentation. The blue vertical line indicates the optimal depth.

report the accuracy of segmentation. The Dice coefficient is defined as:

$$Dice = \frac{2 \times |GT \cap Auto|}{|GT| + |Auto|} \tag{4.6}$$

where GT is the ground truth represented as manual delineation and $Auto$ is the automatic segmentation, $|.|$ is the cardinality of a set.

For each of the three forests - the Blood Pool forest, the Myocardium forest and the Blood Pool&Myocardium forest - we apply the cross-correlation technique to find the optimum parameters. Fig. 4.8, Fig. 4.9 and Fig. 4.10 show that for all three forests there is no much improvement in accuracy after the size of the forest $T = 14$. In this work, however, we use all 20 trees to report the segmentation results. For each of the forests, the optimum depth of the tree D was shown to be the same for the Dice similarity coefficient, and independent of the number of trees in the forest T, this is demonstrated in Fig. 4.8, Fig. 4.9

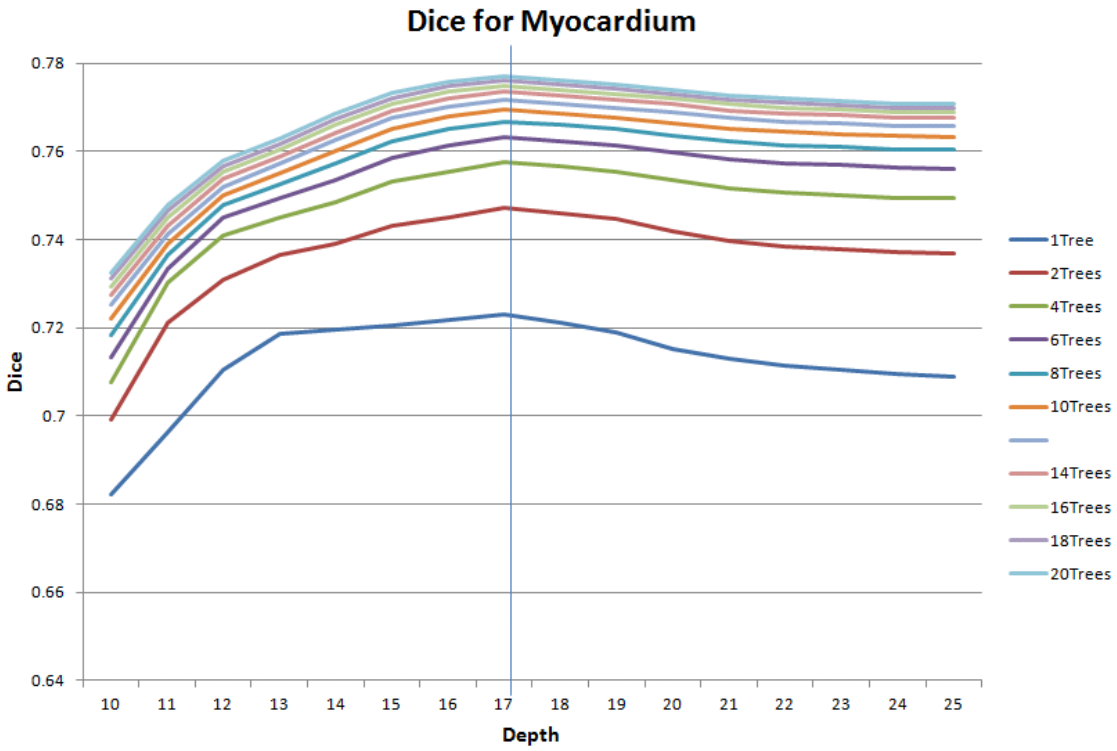


Figure 4.9: Dice similarity coefficients for the myocardium segmentation. The blue vertical line indicates the optimal depth.

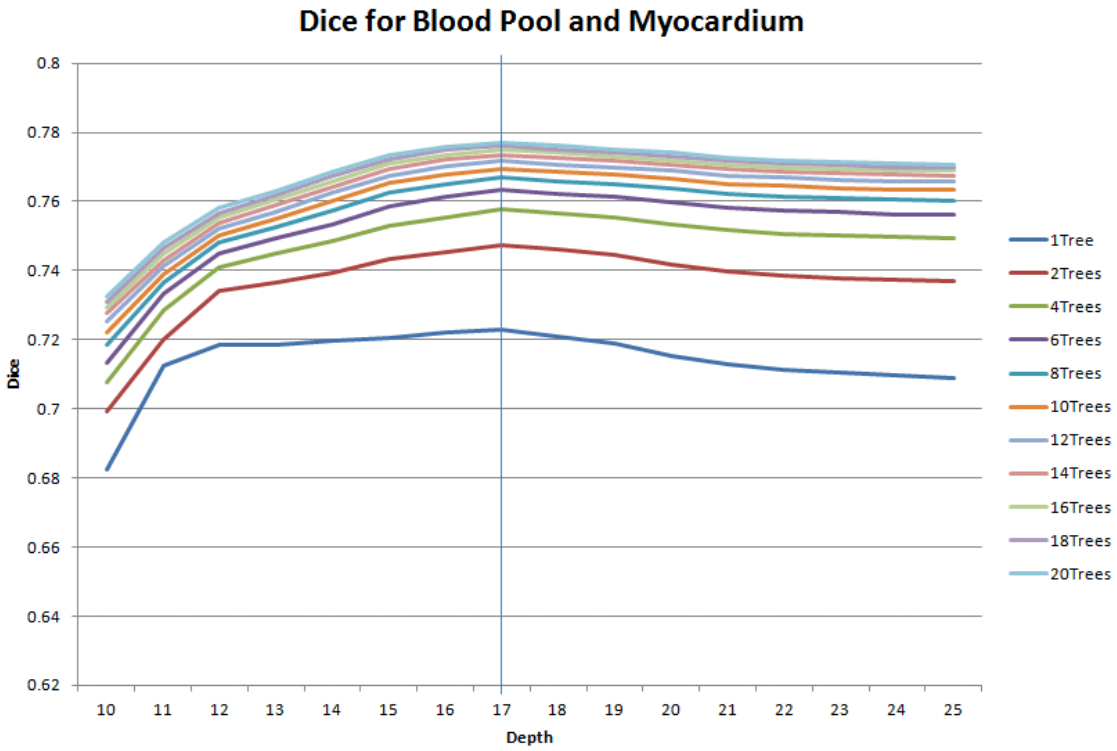


Figure 4.10: Dice similarity coefficients for the blood pool and myocardium segmentation. The blue vertical line indicates the optimal depth.

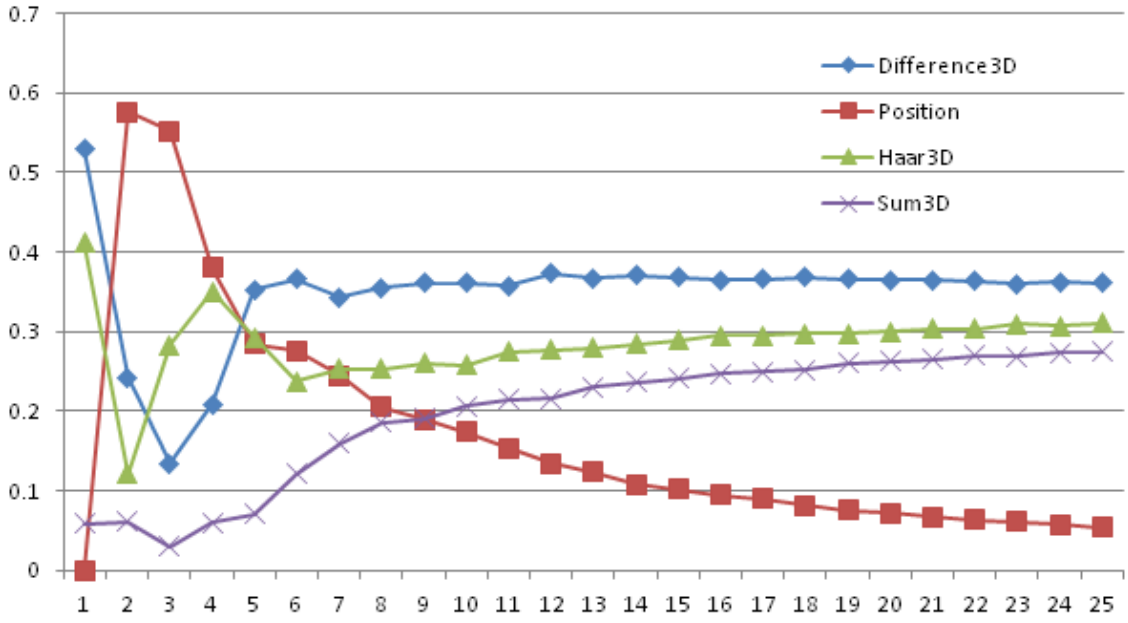


Figure 4.11: Feature Acceptance at different depth with the Classification Decision Random Forest.

and Fig. 4.10. The optimum maximum depth for the Blood Pool forest is D = 19, for the Myocardium forest - D=17, for the Blood Pool&Myocardium - D=17.

4.7 Results

The frequency of each type of feature selected during the training stage for different tree depths is shown in Fig. 4.11. Since the position feature looks only at the absolute position of a voxel, it is pre-dominantly selected by both the Decision Random Forest at depths 1-4 when the complexity of the data is the highest. At this stage, the appearance features poorly distinguish the classes well due to the similar appearance of different tissues or the different appearance of the same tissue.

The mean and standard deviation of the Dice similarity coefficient for (i) the myocardium, (ii) the blood pool are (iii) the blood pool and myocardium, are reported in Table 4.1. The qualitative results of the segmentation are presented in Fig. 4.12.

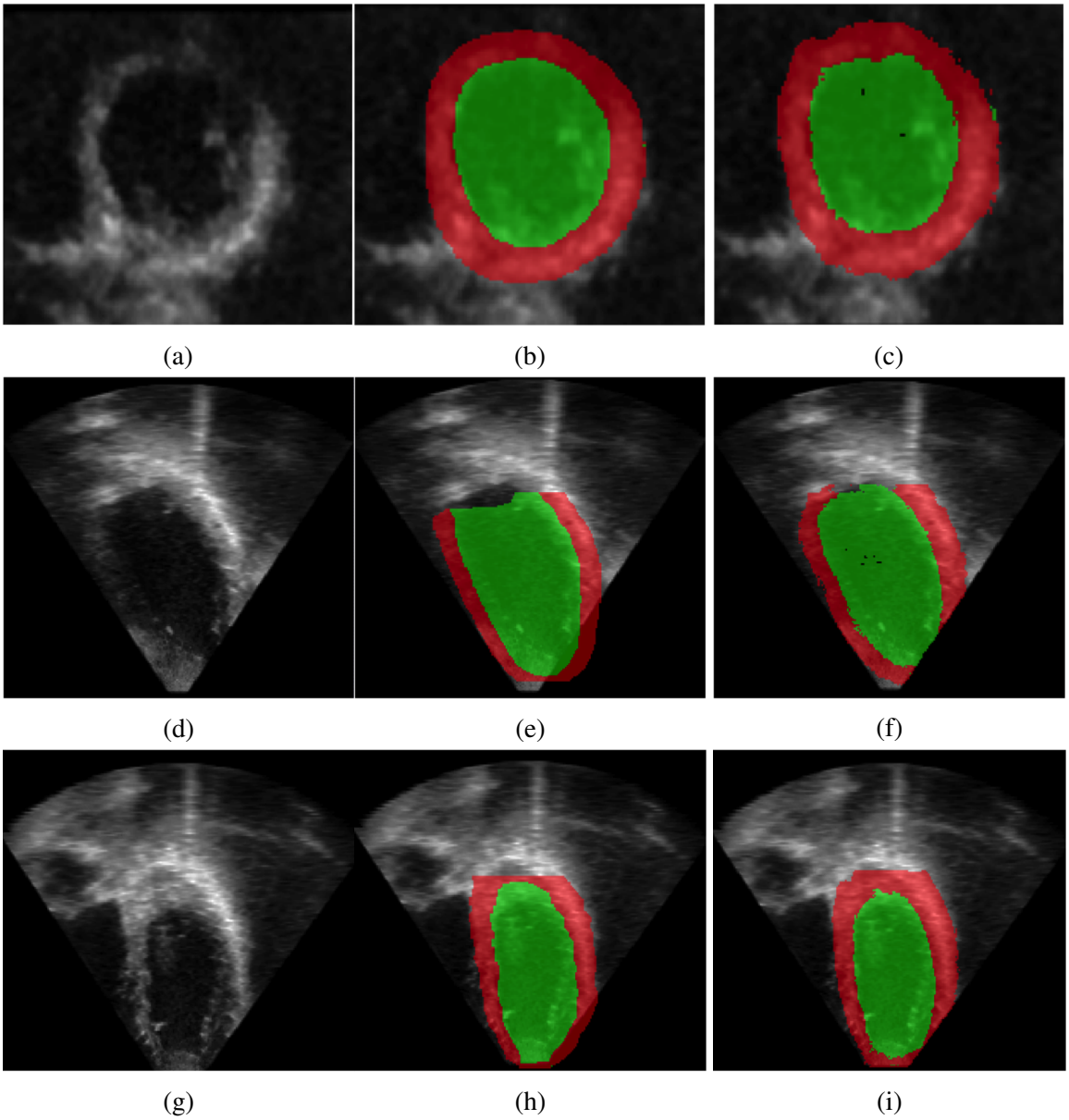


Figure 4.12: Qualitative results of the 3D Echocardiography segmentation with the Classification Decision Random Forest. (a), (d), (g) - original images, (b), (e), (h) - manual segmentation, (c), (f), (i) - automatic segmentation.

	Dice
Myocardium	0.78 ± 0.06
Blood Pool	0.89 ± 0.04
Blood Pool and Myocardium	0.78 ± 0.07

Table 4.1: Mean (μ) $\pm$ standard deviation (σ) of the Dice coefficient for (i) the myocardium, (ii) the blood pool, and (iii) myocardium with the blood pool.

Chapter 5

Features for Accurate Segmentation of Left Ventricle in 3D Echocardiograms

5.1 Introduction

Accurate automatic segmentation of the myocardium and the blood pool of the left ventricle (LV) provides quantitative data from 3D echocardiographic images that helps in the assessment of heart abnormalities and diseases. However, automatic segmentation of 3D echocardiography is challenging due to ultrasound artefacts such as shadowing, attenuation, signal drop-out, speckle, missing boundaries and similarity in appearance of different tissues, for instance of the left and right ventricles.

The accuracy of image-based classification techniques reported in the literature to segment and/or detect structures in medical images varies depending on, for example, the classification model, chosen feature set or the complexity of the structure of interest. Segmentation of the LV is challenging. Therefore, the classification model and features need to be chosen and developed carefully. The Decision Random Forests have proven to show superior performance in recent publications compared to other classifiers (98; 99), and was adopted in this work for 3D echocardiography segmentation. In the Decision Forest framework, segmentation is formalized as a voxel classification problem. Although the choice

of the classifier is important, the accuracy of the model is pre-dominantly determined by the features used within the classifier. In this work, we propose a novel image alignment method to make the widely used classic features correspond across different images, and also introduce three sets of new descriptors and report the accuracy results in section 5.7.

The contributions of this work are twofold. Firstly, as a pre-segmentation step, we propose a new method for left ventricular long axis detection in 3D echocardiography. The robust detection of the mid line is an important step in alignment of echocardiography volumes which is done prior to segmentation. Our method is based on a Feature Asymmetry measure (FA), Local Orientation (LO) and a proposed modified Hough transform. Secondly, we introduce a set of new context rich features that utilize the relevant position of a voxel with respect to a specific landmark or landmarks at the epicardial and/or endocardial boundaries of the left ventricle. The landmarks are detected using the new Centrally Oriented Feature Asymmetry measure (COFA) to highlight and separate the epicardial and endocardial boundaries in an image.

5.2 Specifics of Segmentation of Ultrasound Images using Decision Random Forests

Here, we discuss the weaknesses of the traditional decision forest for ultrasound segmentation and how it can be altered to improve segmentation results. Although such classic features as Haar3D, Difference3D etc. showed promising results in different medical image segmentation and detection problems (76; 86; 87), they are fragile on low quality data, and in many cases have similar values for different tissues in ultrasound images.

Fig. 5.1 demonstrates the challenging problem of tissue classification in 3D echocardiograms. It shows the distribution of voxels of two classes - myocardium (class 1) and background (class 2) taken from one of the nodes of a tree at depth 12 of the forest used in chapter 4. The distribution of voxels is plotted against the two features that resulted in the largest information gain when splitting the node. One can see that there is a big disam-

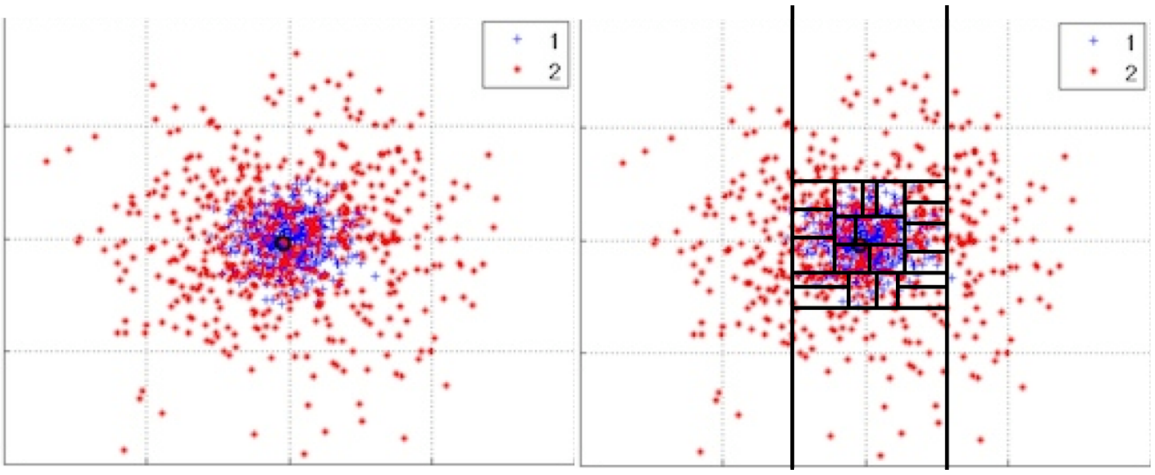


Figure 5.1: Specifics of ultrasound data. The image on the left shows the distribution of myocardium (class 1) and background (class 2) at one of the nodes at depth 12 of the random forest built in chapter 4. The classic features are poorly descriptive of the class label and the classifier has to perform a substantial number of splits of the feature space to get reasonable results. A possible further splitting of the feature space is shown on the right. Due to limited training data the splitting eventually terminates not reaching the desired confidence in the class labels at each subspace.

biguation about the class label. There is not much information about the class label gained at each split, and to get a reasonable confidence in the class label, the decision forest shall continue splitting the feature space reducing the uncertainty of the class label. However, due to limited amount of training data, the splitting terminates when the minimum number of data points is reached at a node. And the desired confidence in the class label is not achieved.

To address the aforementioned issue, one can:

1. Improve the descriptors of the class label, i.e. either (i) improve the currently widely used classic features or (ii) introduce a new set of features that would result in a better separation of classes in the feature space.

2. Make changes to the training procedure of the classifier. In particular, more elaborate non-linear functions, i.e. hypersurfaces, or linear function of two or more features (hyperplanes, or so called oblique functions), can be used to separate the highly non-linear ultrasound data.

The latter proposition very often requires a lot of computational power and is impractical in many cases. It has also been shown that the decision forests produce heavily non-linear decisions when many trees are combined (105), and introducing non-linear split

functions do not bring big changes to the ultimate solution.

The work in this chapter focuses on the first proposition and shows that the right descriptors play a vital role in the performance of the classifier.

We propose the following:

1) A new global registration method for global alignment of the echocardiographic volumes. Image alignment induces correspondence of features across different images, yielding a better segmentation accuracy.

2) A new set of features estimated from FA images (not intensity).

3) A new set of *LoCo* features.

4) A new set of *GloCo* features.

5.3 Improving the Classic Features. Registration of 3D Echocardiograms.

A little misplacement of the probe during freehand ultrasound image acquisition and varied distance from the probe to the heart would change the absolute position and absolute dimensions of the left ventricle in the volume. This, along with a large geometrical variation of the myocardium between subjects, causes non-correspondence of features across different images, i.e. when applied to identical, corresponding structures in different images, the same feature f would look at semantically different, non-corresponding regions to extract the descriptors of the structure.

By aligning the images, features extracted from different images correspond, which makes them better descriptors of the tissue class label. This should lead to a better accuracy of the classifier. For global registration of 3D echocardiography, we first propose a modified local phase based Hough transform for circle detection and apply it to the short-axis slices of the 3D echocardiographic images. The detected mid cavity points are then used to fit the long axis, which is further utilised to bring the 3D volumes to the same coordinate system, as described below.

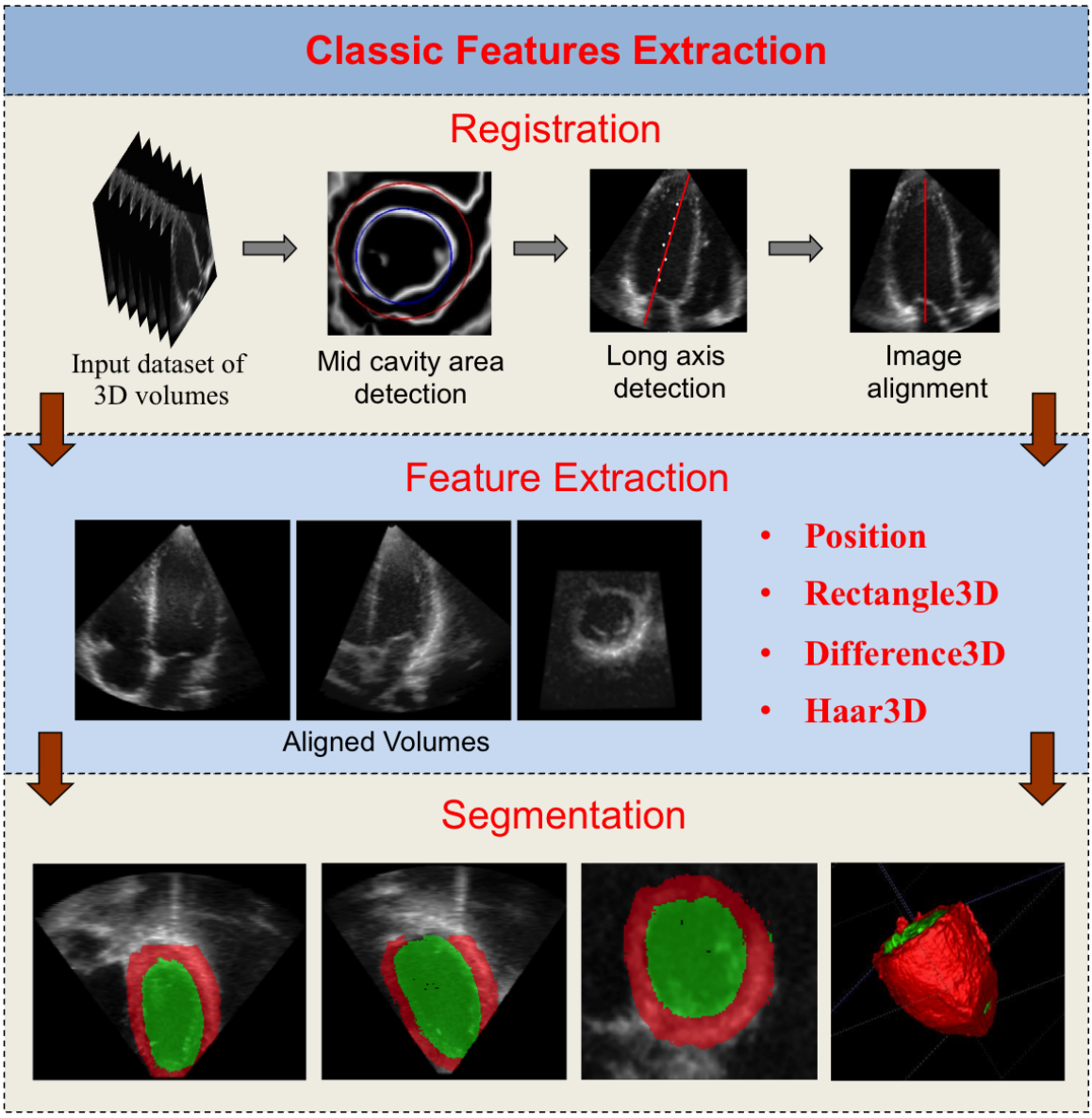


Figure 5.2: Algorithm flowchart for segmentation of 3D echocardiography using the classic features extracted from the registered echocardiographic volumes.

In this work we adopted the classic low level features, Rectangle3D, Haar3D, Difference3D and the position features, and applied them to the registered 3D echocardiographic images. The performance of these features is reported in section 5.7.

5.3.1 Modified Hough Transform for Circle Detection in Short-Axis Slices.

We propose a novel method for mid cavity detection from short-axis slices utilizing a new modified Hough transform for circles, which employs the 3D monogenic signal and 3D Feature Asymmetry images. The general Hough transform for circle detection in 2D images (117) has the following representation:

$$H_z(c_x, c_y, r) = \sum_{\theta} Image(x, y) \tag{5.1}$$

where

$$\begin{cases} x = c_x + r \cdot cos\theta \\ y = c_y + r \cdot sin\theta \end{cases}$$

To apply the Hough transform to an image, one needs to define an edge map, which in many cases might be the original image itself. However, due to attenuation in ultrasound images and its low-contrast nature, some edges become 'invisible' to the Hough transform when the original images are used as the edge maps. Therefore, more robust edge maps are needed for accurate circle detection in ultrasound images. In (52), the intensity-invariant FA measure, defined in section 5.7, has been shown to outperform the intensity based methods for detecting step-like edges in echocardiography. Therefore, we choose to use the FA images as edge maps for calculating the Hough transform for circles in echocardiographic images.

The FA measure is defined using the monogenic signal. The monogenic signal is a high dimensional generalization of the analytic signal and it is based on the Riesz transform,

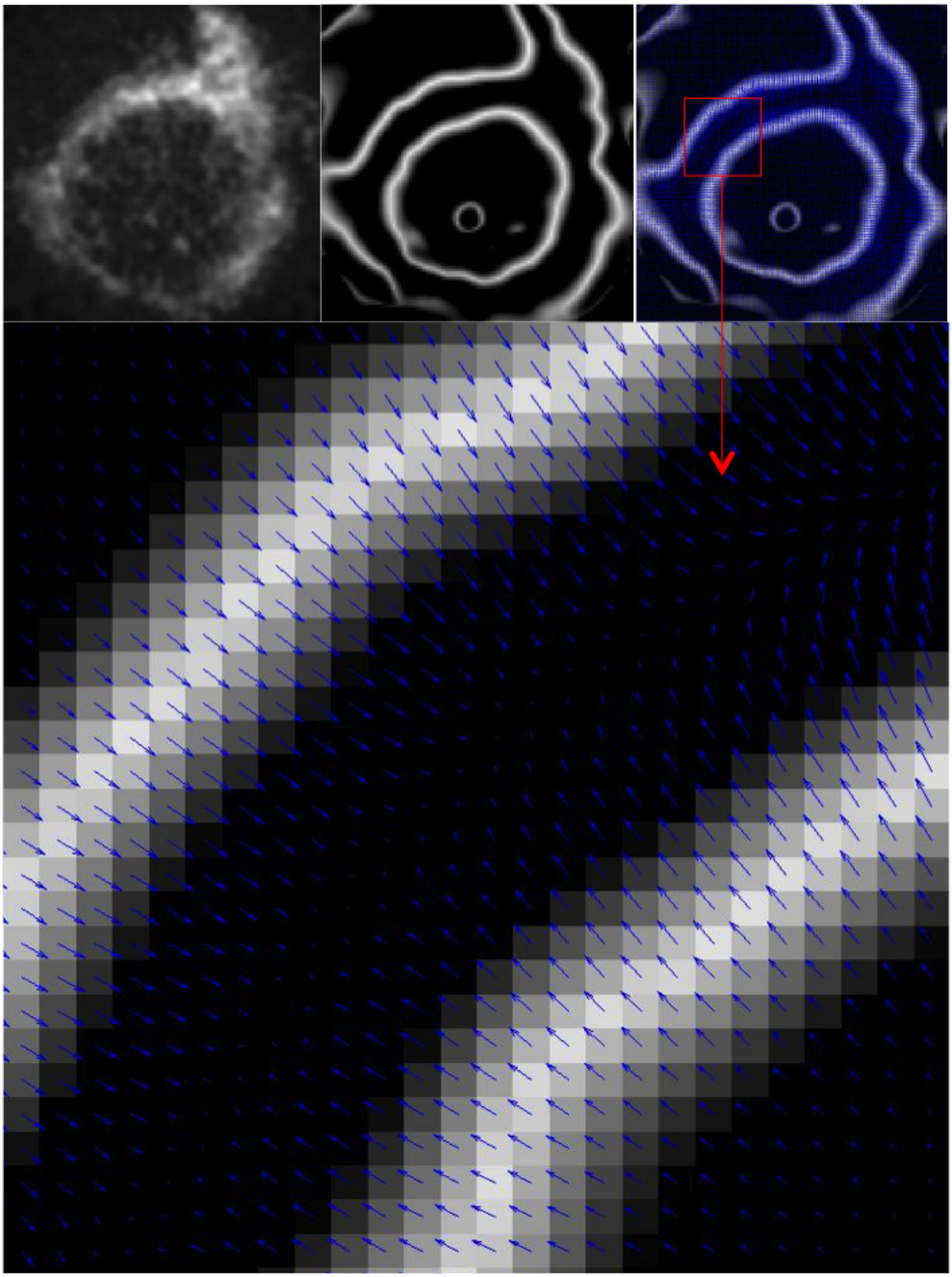
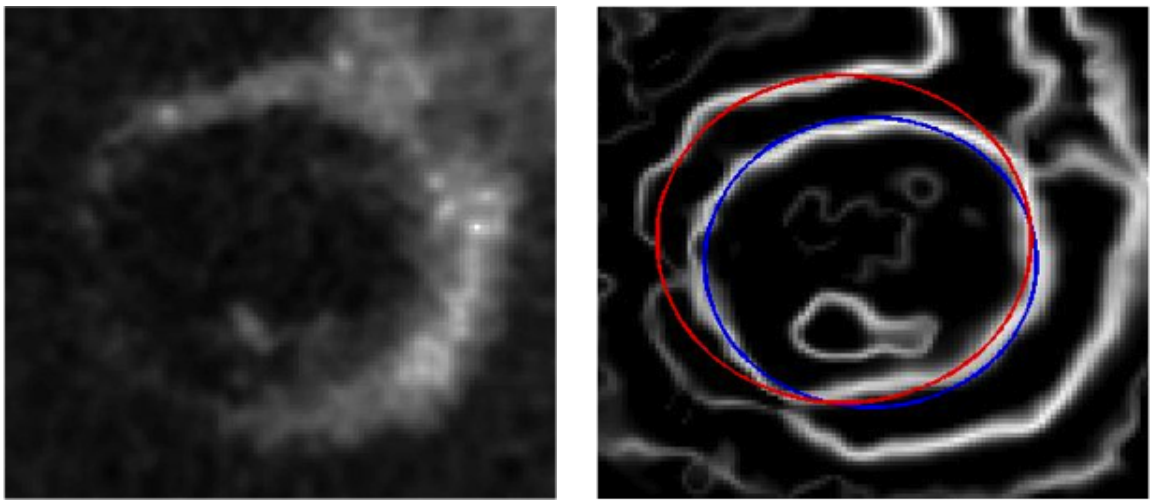


Figure 5.3: Monogenic Orientation. Top row: the original image, the FA image and the MO orientation on top of the FA edge map. Bottom image: the MO orientation points towards the centre at the epicardium and away from the centre at endocardium serving an indicator whether the boundary is epicardial or endocardial.



(a) (b)

Figure 5.4: Modified Hough transform for circle detection. (a) the original image (b) Circle detection with original Hough transform (in red) when applied to the FA image, and with the modified Hough transform (in blue). The original Hough transform fails to detect the true endocardial circle.

which Fourier representation is:

$$(H_1(u,v,w),H_2(u,v,w),H_3(u,v,w)) = \tag{5.2}$$

$$(i\frac{u}{\sqrt{u^2+v^2+w^2}},i\frac{v}{\sqrt{u^2+v^2+w^2}},i\frac{w}{\sqrt{u^2+v^2+w^2}})$$

The monogenic signal is formed by combining the original band-pass image with its Riesz components:

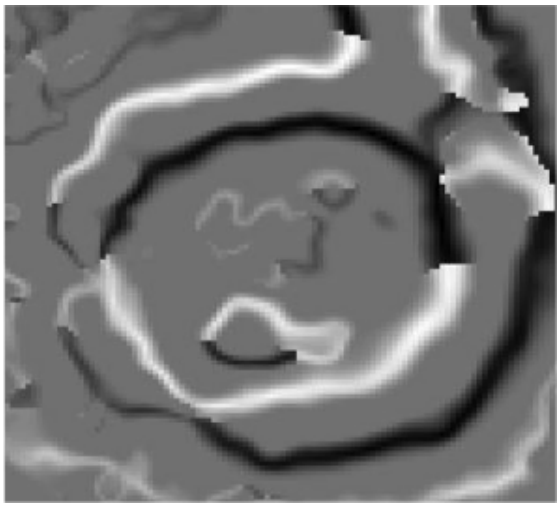
$$I_M(x,y,z) = [I(x,y,z)*g(x,y,z),I(x,y,z)*g(x,y,z)*h_x(x,y,z), \tag{5.3}$$

$$I(x,y,z)*g(x,y,z)*h_y(x,y,z),I(x,y,z)*g(x,y,z)*h_z(x,y,z)] \tag{5.4}$$

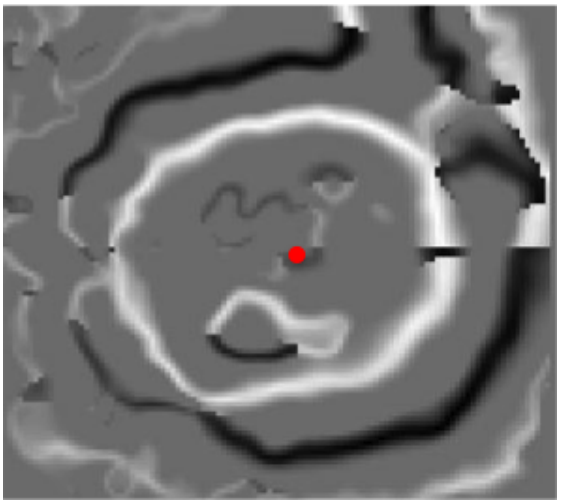
where $g(x,y,z)$ is the spatial domain representation of a bandpass filter and $h_i(x,y,z)$ are the Riesz filter components.

The odd and even filter responses are then defined as follows:

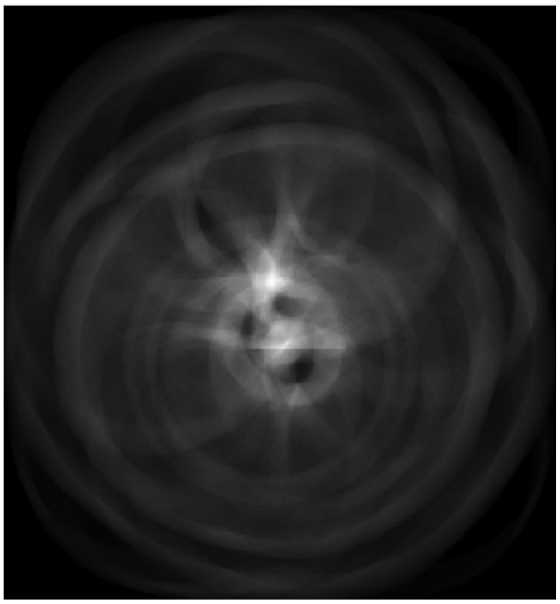
$$even(x,y,z) = I_{M,1}(x,y,z) \tag{5.5}$$



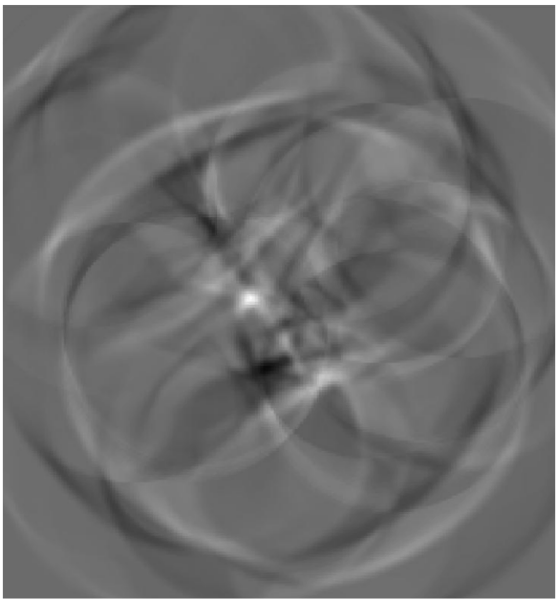
(a) Primary edge map provided to the proposed modified Hough transform.



(b) Edge map produced by the proposed Hough transform when detecting circles for the true mid cavity (in red).



(c) A slice of the original Hough accumulator.



(d) A slice of the proposed Hough accumulator.

Figure 5.5: Modified Hough transform for circle detection. (a) the second component of the MO orientation defined in 5.8 combined with the FA edge map - $sign(MO_y) \cdot FA$ (b) edge map when calculating the proposed modified Hough transform for the true centre of the circle (red dot), the epicardial and endocardial boundaries have opposite signs, (c) a slice of the original 3D Hough accumulator space when applied to the FA image. The two bright dots correspond to the location of the centres of the epicardial and endocardial circles. The locations of the centres are misplaced due to inability of the Hough transform to distinguish between the endocardial and epicardial boundaries. The radius is defined by the slice. (d) a slice of the modified 3D Hough accumulator. The bright white dot corresponds to the centre of the endocardial circle, the black dot corresponds to the epicardial circle.

$$odd(x,y,z) = \sqrt{I_{M,x}(x,y,z)^2 + I_{M,y}(x,y,z)^2 + I_{M,z}(x,y,z)^2}$$

In computing the monogenic signal, aside from choosing how to combine the results from different scales, the selection of a bandpass filter has to be made. In our case we used the log-Gabor filter though other filters could have been used.

The log-Gabor filter is defined in the frequency domain as:

$$G(w) = \exp[-(\frac{\log^2_(w/w_0)}{2\log^2(k/w_0)})] \tag{5.6}$$

The FA measure is defined as:

$$FA_{3D}(x,y,z) = \sum_s \frac{\lfloor ||odd_s(x,y,z)|| - ||even_s(x,y,z)|| - T_s \rfloor}{\sqrt{(even_s(x,y,z))^2 + (odd_s(x,y,z))^2 + \varepsilon}} \tag{5.7}$$

where ε is a small constant to avoid division by zero and T_s is a scale specific threshold that suppresses any response due to noise or symmetric points of the image:

$$T_s = \exp[mean(\log(\sqrt{(even_s(x,y,z))^2 + (odd_s(x,y,z))^2}))]$$

Examples of *FA* images of different scales are shown in Fig. 5.7.

However, the general Hough transform for circle detection in equation 5.1 confuses the epicardial and endocardial borders when applied to the FA images, and fails in the majority of cases, see the red circle in Fig. 5.4b. This happens due to the close proximity of the epicardial and endocardial borders in some cases, and more oval-like shape of the epicardial/endocardial contours in others. One can notice that the intensity gradient at the epicardial and endocardial boundaries in FA images points in the opposite directions - towards the centre at the epicardial border and away from the centre at the endocaridial border. Similarly, we can notice that the first two components of the Riesz filter (H_x, H_y) act in a analogous way as the derivatives in x and y directions. Therefore, we propose to use the following measure to distinguish between the epicardial and endocardial boundaries, in

this paper we refer to this measure as the Monogenic Orientation (MO):

$$MO = (I * g * H_x, I * g * H_y) \tag{5.8}$$

where * denotes convolution.

The orientation MO points in the opposite direction at the epicardial and endocardial boundaries, Fig. 5.3. This can serve as a good indicator of whether the boundary is epicardial or endocardial, i.e. whether there is an edge or not, the FA images are used, and the proposed MO measure indicates whether the edge is epicardial or endocardial.

To combine the FA edge maps and the orientation MO, the following definition of the modified Hough transform for circles can be used:

$$H_z(c_x, c_y, r) = \sum_s \sum_\theta sign(x - c_x) \cdot sign(I^s_{M,x}) \cdot FA^s_z(x, y) \tag{5.9}$$

where

$$\begin{cases} x = c_x + r \cdot cos\theta \\ y = c_y + r \cdot sin\theta \end{cases}$$

Although such formulation produces accurate results, we propose to add another summand to the equation and define the modified Hough transform as follows:

$$H_z(c_x, c_y, r) = \sum_s \sum_\theta sign(x - c_x) \cdot sign(I^s_{M,x}) \cdot FA^s_z(x, y) + \tag{5.10}$$

$$sign(y - c_y) \cdot sign(I^s_{M,y}) \cdot FA^s_z(x, y)$$

where

$$\begin{cases} x = c_x + r \cdot cos\theta \\ y = c_y + r \cdot sin\theta \end{cases}$$

c_x, c_y, r parameterize the circle, and $I^s_{M,x}$ and $I^s_{M,y}$ are the x and y monogenic signal components respectively, as defined in 5.3. See also Fig. 5.4 and 5.5.

In our implementation the summation is performed over multiple scales s of the band-pass filter. Notice that the Hough transform can be fairly accurately computed using only one of the summands in 5.10. However, we have found in our application that employing both terms leads to better accuracy and robustness. The endocardial and epicardial centre points correspond to the global maximum and the global minimum of the Hough transform. In our application we used the local-phase based Hough transform on each short-axis slice z, to give the endocardial and epicardial centre points, $(x^z_{c,endc}, y^z_{c,endc})$ and $(x^z_{c,epic}, y^z_{c,epic})$, and the endocardial and epicardial radii, r^z_{endc} and r^z_{epic}, as:

$$x^z_{c,epic}, y^z_{c,epic}, r^z_{epic} = argmax_{c_x,c_y,r} H_z(c_x, c_y, r) \tag{5.11}$$

$$x^z_{c,endc}, y^z_{c,endc}, r^z_{endc} = argmin_{c_x,c_y,r} H_z(c_x, c_y, r) \tag{5.12}$$

An example of circle detection using the modified Hough transform is illustrated in Fig. 5.4b.

5.3.2 Long Axis Detection and Global Alignment of 3D Echocardiography

Having detected the centre points, the long axis is then fitted as the first principal component of the detected endocardial centres. Each volume is then rotated to a common coordinate system so that the long axis is positioned vertically and centred in the aligned image. This corresponds to two translations and two rotations. The remaining five parameters (1 translation, 1 rotation and 3 scaling) are found using a standard rigid registration technique (in our work we used the FLIRT registration tool, http://www.fmrib.ox.ac.uk/fsl/flirt/), fixing the determined four parameters. The process of long axis detection and image alignment is repeated in an iterative manner until no further improvement is achieved.

Examples of image alignment is demonstrated in Fig. 5.6.

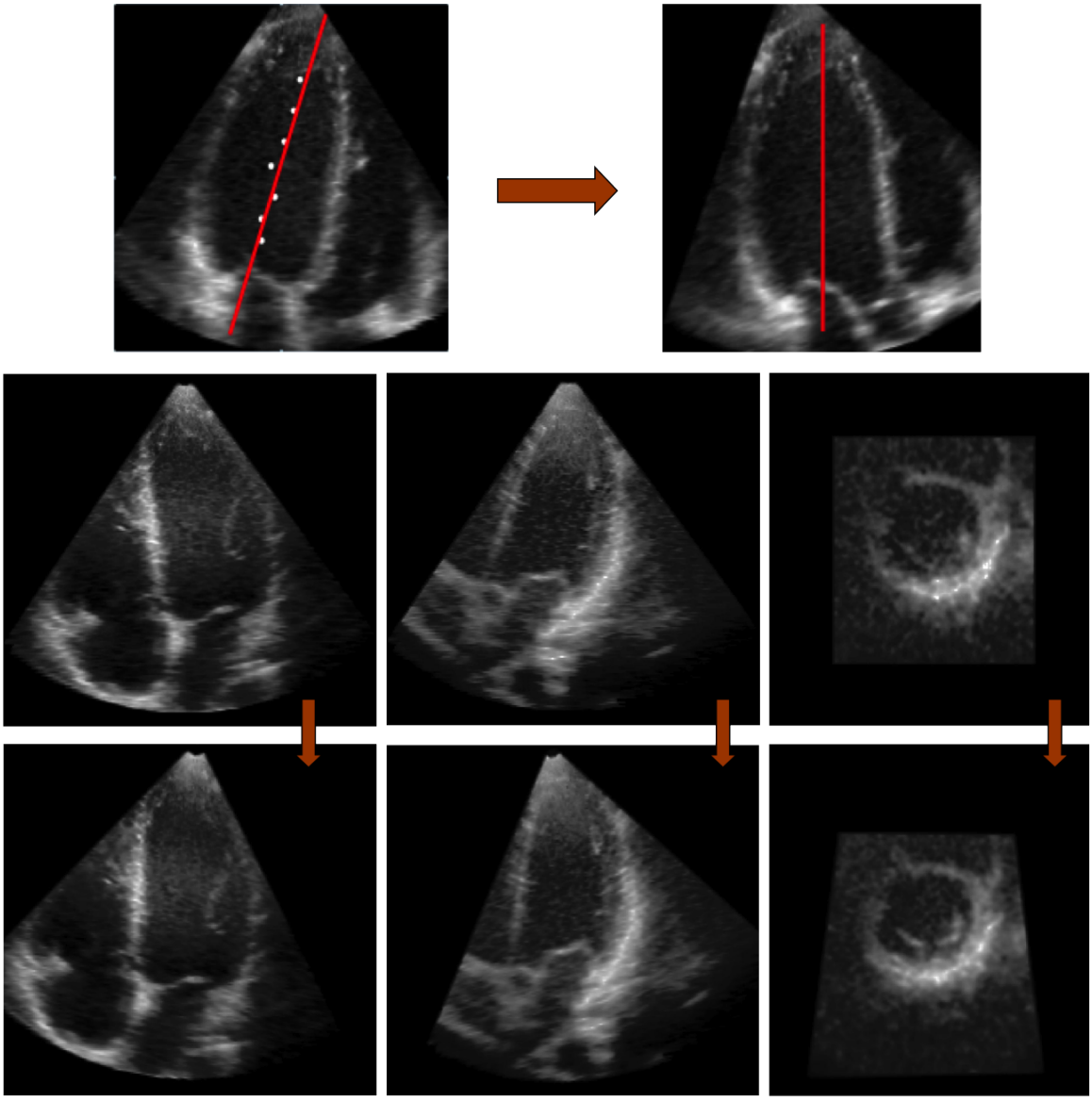


Figure 5.6: Image alignment. The long axis is fitted as the first principal component of the detected endocardial centres. Each volume is then rotated to a common coordinate system so that the long axis is positioned vertically and centred in the aligned image.

After all images are globally aligned, the standard classic features, such as Rectangular3D, Difference3D, Haar3D, DifferenceHaar3D and the position features, are extracted.

5.4 FA Features

The local appearance features are low level features used within local patches that characterise a limited range. High level features are appreciably important to represent semantic information and global shape of a structure. For LV segmentation, the goal is to detect endocardial and epicardial boundaries. It is usually assumed that they have step-like edge characteristics, and the FA measure proved to be a good measure for outlining edges in ultrasound images. Fig. 5.7 demonstrates edge enhancement in the FA images of different scales compared to the original image. Extracting the classic Rectangular3D, Difference3D, Haar3D and DifferenceHaar3D from the FA images forms the new set of FA features - FARectangular3D, FADifference3D, FAHaar3D and FADifferenceHaar3D. The FA features capture structural information in the neighbourhood of the voxel and helps decrease uncertainty about the voxel class label.

5.5 *LoCo* and *GloCo* Features.

The conventional appearance features do not explicitly exploit any information about the relative location of the voxel to various structures in the image. Here, we introduce a novel set of features, *LoCo* and *GloCo* features, which encode the relative geometrical position of the voxel to a number of landmarks on the epicardial or/and endocardial boundaries. We introduce a novel measure, called the Centrally Oriented Feature Asymmetry (COFA) measure, which separates and highlights the epicardial and endocardial boundaries, and assists in detection of the corresponding landmarks across different 3D echocardiographic images. This further enables extraction of the *LoCo* and *GloCo* features.

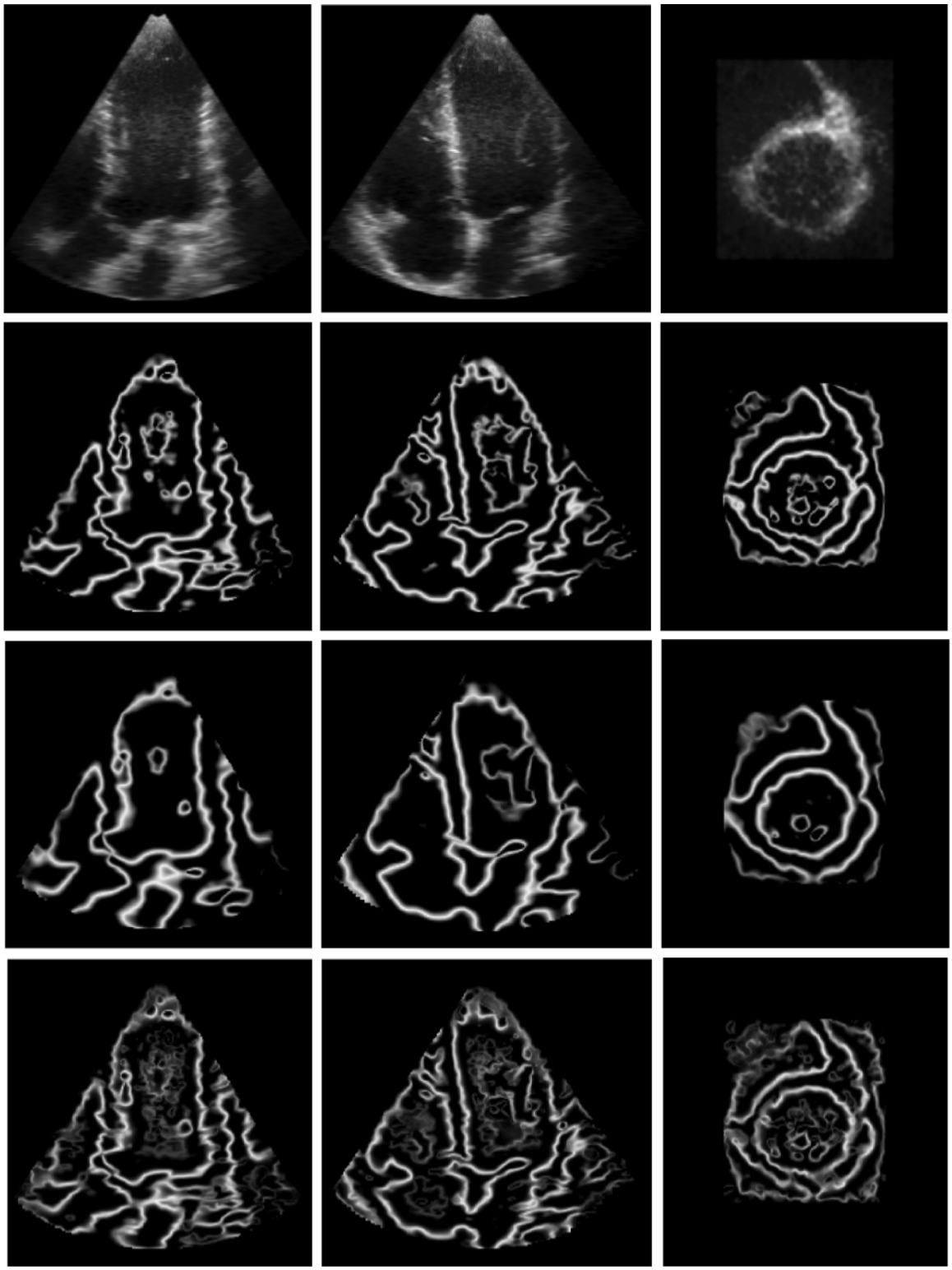


Figure 5.7: FA features. The original images (top raw), the FA images of different scales and the multiscale FA images (bottom row). The classic features such as Rectangular3D, Difference3D and Haar3D are extracted from the FA images forming a new set of FA features.

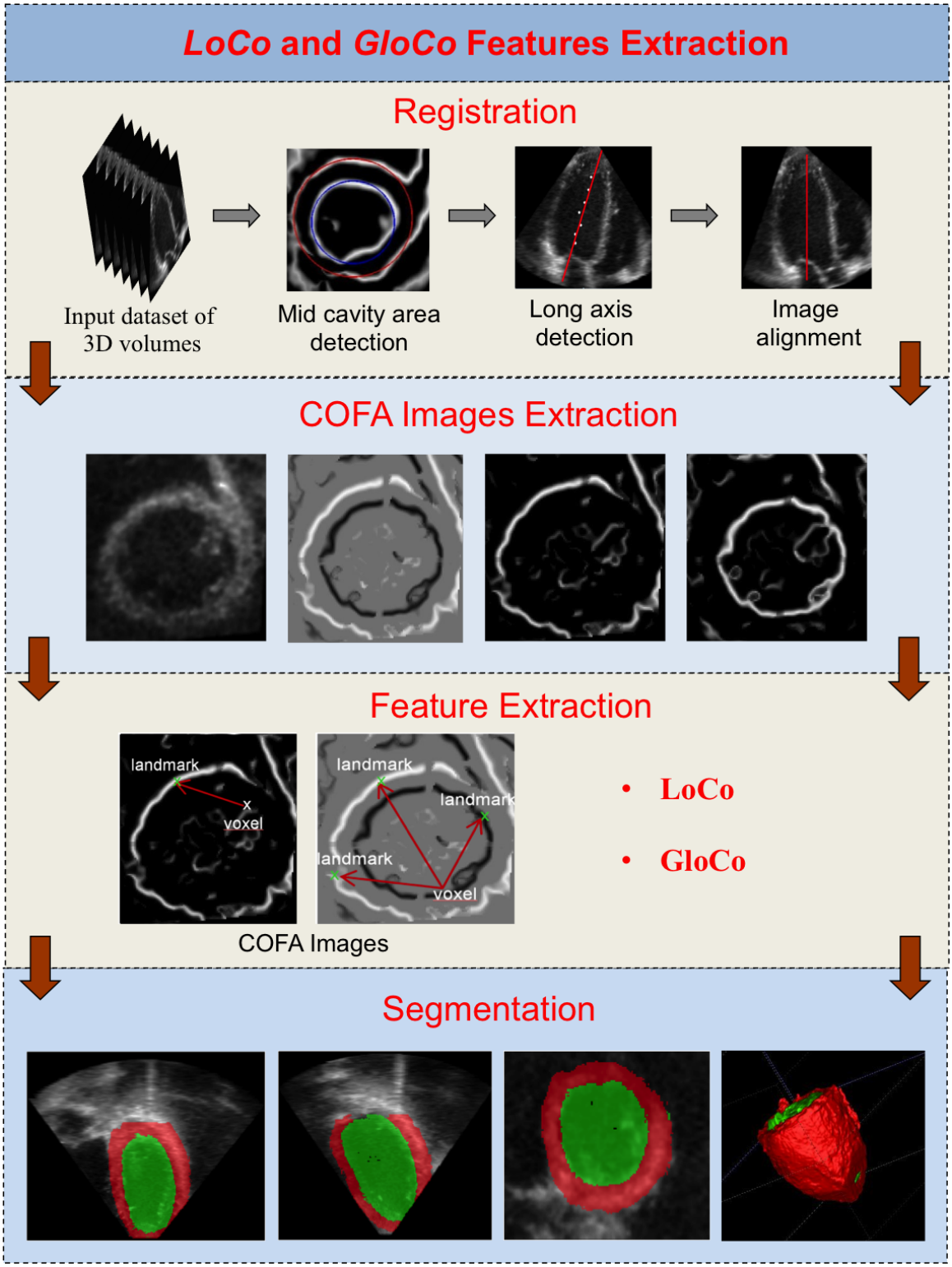


Figure 5.8: Algorithm flowchart for *LoCo* and *GloCo* features extraction.

5.5.1 *LoCo* Features

This type of feature is determined by the relative position of a considered voxel $\mathbf{v}^k = (v_x^k, v_y^k, v_z^k)$ to a pre-chosen landmark $\mathbf{p}^k = (p_x^k, p_y^k, p_z^k)$ on the endocardial or epicardial boundary in image k, Fig. 5.11c. The *LoCo* feature is a context rich feature that capture geometric variability of the myocardium and can be regarded as a shape constraint. The *LoCo* feature set is defined as:

$$f_{LoCo}^{x}(\mathbf{v}^k, \mathbf{p}^k) = v_x^k - p_x^k \tag{5.13}$$

$$f_{LoCo}^{y}(\mathbf{v}^k, \mathbf{p}^k) = v_y^k - p_y^k \tag{5.14}$$

$$f_{LoCo}^{z}(\mathbf{v}^k, \mathbf{p}^k) = v_z^k - p_z^k \tag{5.15}$$

$$f_{LoCo}^{xy}(\mathbf{v}^k, \mathbf{p}^k) = \sqrt{(v_x^k - p_x^k)^2 + (v_y^k - p_y^k)^2} \tag{5.16}$$

$$f_{LoCo}^{dist}(\mathbf{v}^k, \mathbf{p}^k) = |\mathbf{v}^k - \mathbf{p}^k| \tag{5.17}$$

The landmark detection procedure is described in detail below.

5.5.2 *GloCo* Features

Unlike the *LoCo* features, the *GloCo* features use the relative position of a voxel to a number of structures on the epicardial and endocardial boundary shapes. Barycentric coordinates are utilized to uniquely specify the location of the voxel with respect to the specified structures on the boundary shapes, Fig. 5.9. To calculate the Barycentric coordinates in 3D space, the 3D simplex, a tetrahedron, is constructed by randomly selecting four structure

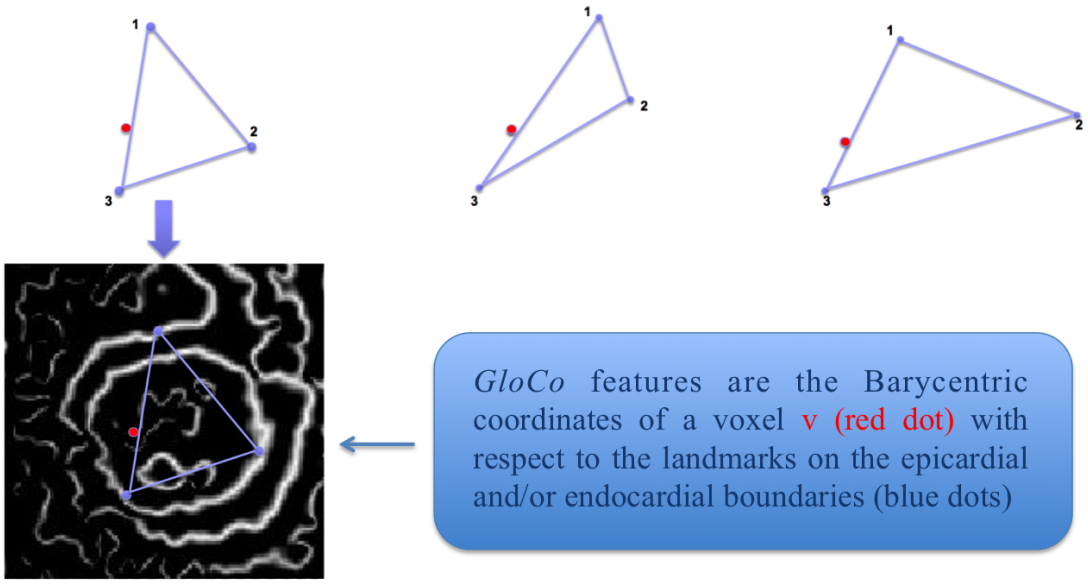


Figure 5.9: *GloCo* features. The three red dots in the figure above have the same Barycentric coordinates with respect to the landmarks 1, 2, 3. The Barycentric coordinates of the red dots are invariant to the variability of the triangular as long as the relative position of the red dot with respect to the landmarks remains the same. Similarly, the corresponding voxels in different images will have the same Barycentric coordinates with respect to a number of landmarks on epicardial and endocardial boundaries, i.e. they are invariant to the myocardium variability. This makes the *GloCo* features invariant to the local and global variability of the myocardium.

points on endocardial and epicardial boundaries. Mathematically, the scalars are the Barycentric coordinates of an arbitrary voxel $\mathbf{v} = (v_1, v_2, v_3)$ with respect to the four nonplanar structure points (landmarks) $\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}, \mathbf{p_4}$ if

$$\mathbf{v} = u_1\mathbf{p}_1 + u_2\mathbf{p_2} + u_3\mathbf{p_3} + u_4\mathbf{p_4} \tag{5.18}$$

subject to

$$\sum_{i=1}^{4} u_i = 1 \tag{5.19}$$

where $\mathbf{v}$ and $\mathbf{p_i}$ denote the Euclidean coordinates. The four landmarks on the epicardial or/and endocardial boundaries are randomly chosen as described next, and are ensured to be nonplanar. To ensure invariance to arbitrary pose of the epicardial and endocardial shape the coefficients are constrained to sum to one in 5.19.

The proposed *GloCo* features are the Barycentric coordinates and computed as follows:

$$f^1_{GloCo}(\mathbf{v^k};\mathbf{p^k_1};\mathbf{p^k_2};\mathbf{p^k_3};\mathbf{p^k_4}) = u_1 \tag{5.20}$$

$$f^2_{GloCo}(\mathbf{v^k};\mathbf{p^k_1};\mathbf{p^k_2};\mathbf{p^k_3};\mathbf{p^k_4}) = u_2 \tag{5.21}$$

$$f^3_{GloCo}(\mathbf{v^k};\mathbf{p^k_1};\mathbf{p^k_2};\mathbf{p^k_3};\mathbf{p^k_4}) = u_3 \tag{5.22}$$

$$f^4_{GloCo}(\mathbf{v^k};\mathbf{p^k_1};\mathbf{p^k_2};\mathbf{p^k_3};\mathbf{p^k_4}) = u_4 \tag{5.23}$$

where

$$(u_1,u_2,u_3)^T = T^{-1}(\mathbf{v}-\mathbf{p_4})$$

$$u_4 = 1-u_1-u_2-u_3$$

$$T = \begin{pmatrix} x_1-x_4 & x_2-x_4 & x_3-x_4 \\ y_1-y_4 & y_2-y_4 & y_3-y_4 \\ z_1-z_4 & z_2-z_4 & z_3-z_4 \end{pmatrix}$$

and x_i, y_i, z_i are the coordinates of landmark $\mathbf{p_i}$.

5.5.3 Landmark Detection. Centrally Oriented Feature Asymmetry Measure.

To extract the *LoCo* and *GloCo* features, we need to be able to distinguish between the endocardial and epicardial boundaries so that extraction of the corresponding landmarks is possible. To do that, we propose the Centrally Oriented Feature Asymmetry (COFA) measure defined in terms of the detected epicardial and endocardial centre points $(x^z_{c,i}, y^z_{c,i})$

as in 5.11 and 5.12, for each of the short-axis slices z, as follows:

$$COFA_z(x,y) = \sum_s \sum_{i=epic,endc} sign(x - x^z_{c,i}) \cdot sign(I^s_{M,x}) \cdot FA^s_z(x,y) + \tag{5.24}$$

$$sign(y - y^z_{c,i}) \cdot sign(I^s_{M,y}) \cdot FA^s_z(x,y)$$

See Fig. 5.10b. The epicardial and endocardial boundaries can be separated in the following way:

$$COFA_{z,epic}(x,y) = \sum_s \sum_{i=epic,endc} \lfloor sign(x - x^z_{c,i}) \cdot sign(I^s_{M,x}) \cdot FA^s_z(x,y) \rfloor + \tag{5.25}$$

$$\lfloor sign(y - y^z_{c,i}) \cdot sign(I^s_{M,y}) \cdot FA^s_z(x,y) \rfloor$$

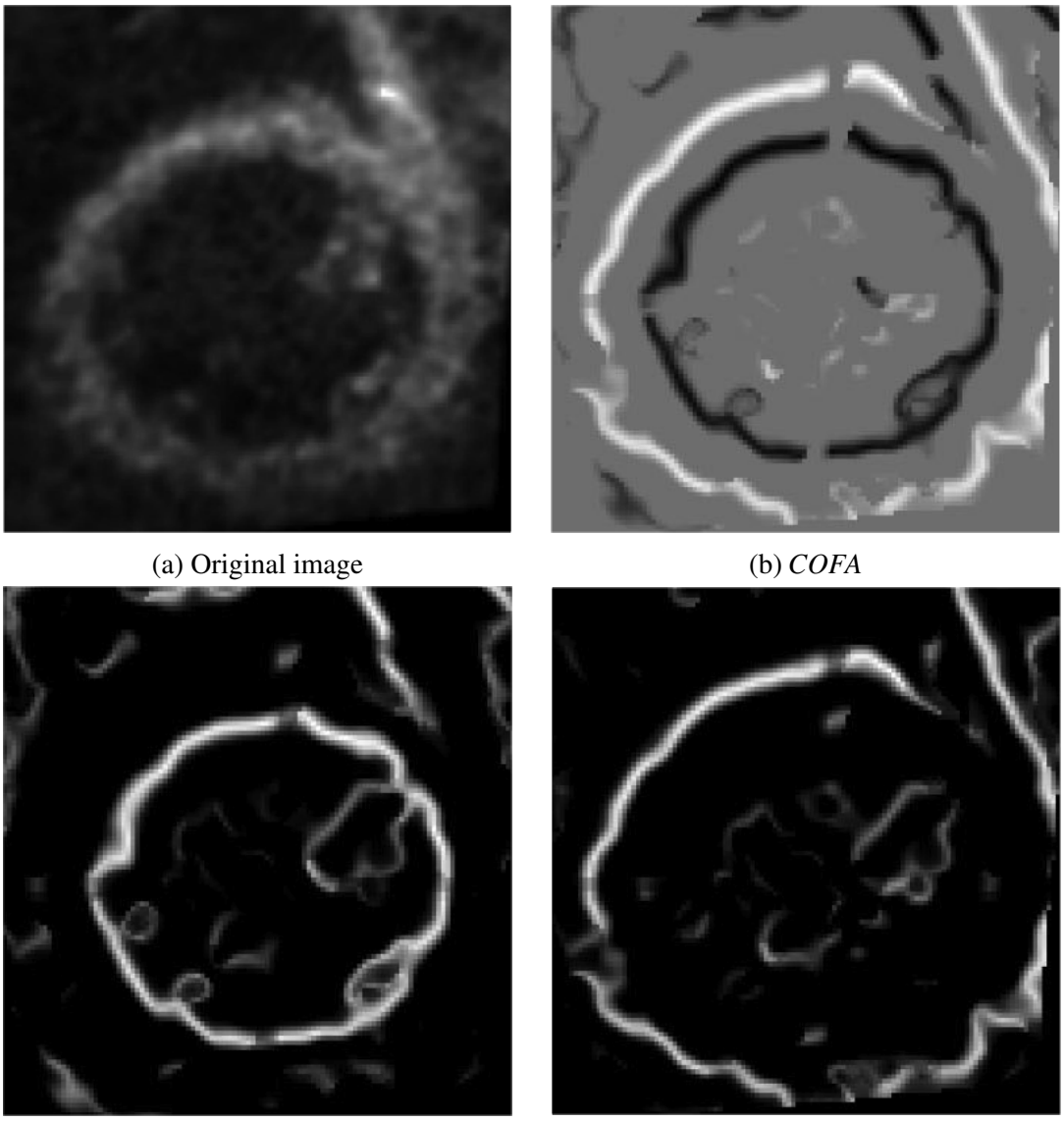
$$COFA_{z,endc}(x,y) = \sum_s \sum_{i=epic,endc} \lceil sign(x - x^z_{c,i}) \cdot sign(I^s_{M,x}) \cdot FA^s_z(x,y) \rceil + \tag{5.26}$$

$$\lceil sign(y - y^z_{c,i}) \cdot sign(I^s_{M,y}) \cdot FA^s_z(x,y) \rceil$$

where $\lfloor .. \rfloor$ and $\lceil .. \rceil$ are the operators that zero the negative and positive values correspondingly. See also Fig. 5.10c and 5.10d.

With all the images globally aligned, we can find the corresponding endocardial and epicardial points across different images. Two parameters, the short-axis slice z and the polar angle θ, define in which slice and in what direction from the centre of the cavity area the epicardial or endocardial landmark will be detected, Fig. 5.11. To find the epicardial and endocardial landmarks along the chosen direction, the COFA images are utilized. The maximum COFA detected feature along the direction defines the epicardial landmark, whereas the minimum defines the endocardial landmark, Fig. 5.11b.

Thus, a detected landmark is defined by the two parameters z and θ, and is detected in



(a) Original image (b) *COFA*

(c) *COFAendocardial* (d) *COFAepicardial*

Figure 5.10: *COFA* images. (a) The original image, (b) the *COFA* measure as in 5.24, (c) $COFA_{epicardium}$ as in 5.25, (d) $COFA_{endocardium}$ as in 5.26. The *COFA* images highlight and separate the endocardial and epicardial boundaries. This enables to further detect the corresponding landmarks on epicardial and endocardial boundaries across different images.

each image independently as follows:

$$\begin{cases} p_x^{k,q}, p_y^{k,q} = argmax_{x,y}|COFA_{z,q}(x,y)| \\ p_z^{k,q} = z \end{cases} \tag{5.27}$$

subject to

$$\begin{cases} x^q = x_{c,q}^z + r \cdot cos\theta \\ y^q = y_{c,q}^z + r \cdot sin\theta \end{cases}$$

$$\text{where } r_q^z - shift < r < r_q^z + shift$$

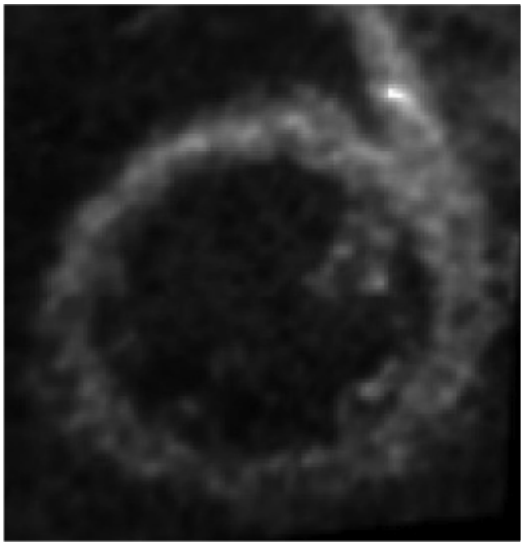
and shift defines a part of the directional line (defined by the directional angle θ), along which the landmark will be detected.

Here the endocardial/epicardial centres $(x_{c,q}^z, y_{c,q}^z)$ and radii r_q^z in the slice z are found by equation 5.11 and equation 5.12, q denotes the epicardial or endocardial boundary.

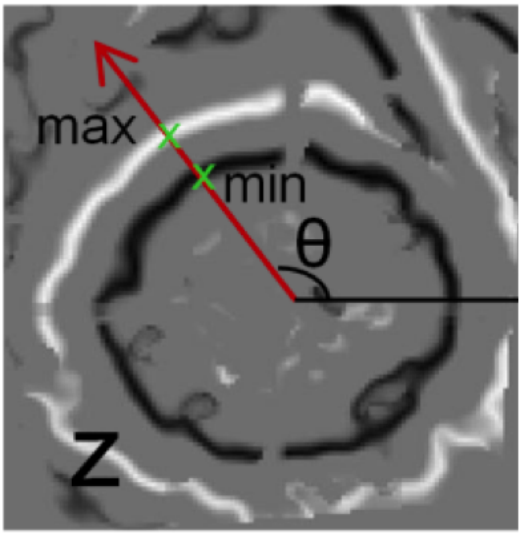
In the case of the *GloCo* features, the four landmarks $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$ are defined by randomly selecting a short-axis slice z and a directional angle θ for each of the points. The four landmarks are further detected separately in each image using equation 5.27.

5.5.4 Comparison of position and *LoCo* and *GloCo* features.

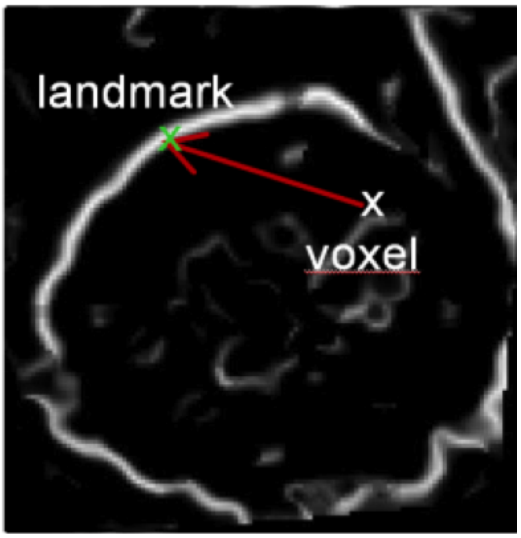
A position feature does not exploit appearance of the image and considers only the absolute position of the voxel in the image. Thus, it provides a weak constraint on geometric variability of the myocardium. On the contrary, the *GloCo* feature quantifies the relative position of the voxel with respect to four landmarks detected using intensity information from the COFA images. The detected landmarks are the tetrahedron vertices used in calculation of the Barycentric coordinates and have different spatial locations across the images. Thus, the tetrahedron captures the coupled endo- and epicardial shape variability enforcing a strong constraint on the geometry of the myocardium.



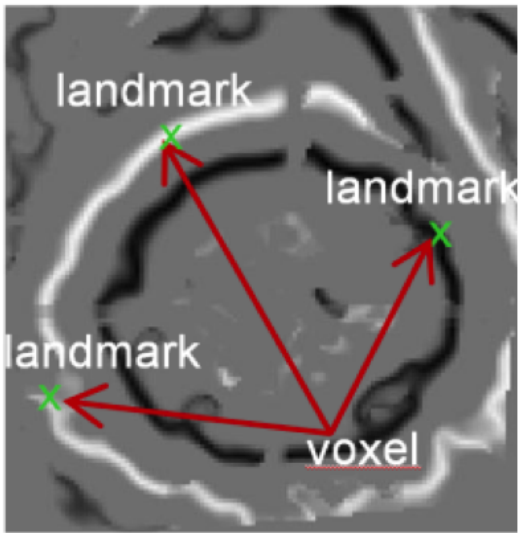
(a) Original image.



(b) Landmark detection.



(c) *LoCo* feature.



(d) *GloCo* feature.

Figure 5.11: Landmark detection and *LoCo* and *GloCo* features. Landmark detection procedure in (b): a slice z and a polar angle θ define in which slice and in what direction the landmark will be detected. Since the images are aligned, the slice z and the angle θ correspond across images. The maximum along direction θ defines the epicardial landmark, the minimum - the endocardial. (c) the *LoCo* features are the members of the distance vector from the voxel to the landmark, (d) the *GloCo* features are the Barycentric coordinates of the voxel with respect to four landmarks.

5.6 Experiments

In this study we have used the same dataset as in the previous chapter - 25 3D end-diastolic echocardiograms from healthy subjects acquired by the Philips iE33 ultrasound system. Volume dimensions were (224x208x208) voxels with an average of $0.88mm^3$ spatial resolution. The myocardium and the blood pool of all volumes were manually segmented by an expert.

To investigate the performance of the three types of features described in this chapter, we conducted the following experiments:

1) To unfold the effect of the proposed registration on the classic features and on the performance of the classifier, we ran an experiment with the decision forest built using the classic features extracted from the aligned 3D echocardiograms and compared the results with the results obtained using the decision forest built using the classic features extracted from the original unaligned images.

2) To investigate whether the new FA features contain any extra information about the class label, we trained the forest using both the classic and the new FA features. The feature selection rate and the validation accuracy of the classifier were further compared to the ones produced by the decision forest trained using only the classic features.

3) Similarly to the FA features, to understand the impact of the *LoCo* and *GloCo* features, we built the forest using the classic and the *LoCo* and *GloCo* features. The effect of the features is then evaluated comparing the results with the results from the classic forest.

To distinguish between the three aforementioned forest, we give them the corresponding names - the classic random forest, the FA random forest and the *LoCo* random forest.

5.6.1 Model Parameters

The stopping criteria for growing the trees for each of the forests were:

1) The maximum depth D. This parameter of the forest was found using cross-validation technique described below.

2) No information gain in splitting. This most often happens when all datapoints at a node belong to the same class, i.e. when there is no uncertainty about the class label.

3) The minimum number of datapoints at a node M. In our experiments we used $M = 50$.

Thus, there is only one parameter to be found - the depth of the trees D. For that we performed 5-fold cross-validation for each of the three experiments on the available 25 echocardiographic volumes. The data was randomly split into 5 non overlapping groups, 5 echocardiographic images in each group. One of the five groups was left out for validation and the remaining four groups were used for training, the procedure was repeated five times, selecting different folders for validation and the remaining folders for building the classifier. The average error for different values of D is reported in the results section 5.7, and the minimum average error is reported in Table 5.1. The optimal value of the parameter D corresponds to the minimum average validation error.

Apart from the stopping criteria parameters there are also two other parameters of the forest - the number of candidate features at a node m and the number of trees in the forest T. Adding extra trees to the forest may only improve the overall accuracy of the classifier, and therefore, we used $T = 20$ for all experiments and showed that increasing the number of trees would not improve the overall results. In some cases, less than $T = 20$ trees could be used, thus reducing the computation cost at testing. See Fig. 5.15, 5.19 and 5.23 for more details.

As for the number of features m tested at a node during training, we used the following numbers:

1) The classic random forest. We used $m = 200$ randomly chosen features at a node.

2) The FA random forest. For the comparison of the FA random forest to the classic random forest to be valid, we used the same number of randomly chosen classic features $m_{classic} = 200$ at a node, and $m_{FA} = 200$ FA features, thus checking whether feeding extra FA features brings any value.

3) The *LoCo* random forest. As with the FA forest, we kept the number of randomly

selected classic features unchanged $m_{classic} = 200$ and added extra $m_{LoCo} = 200$ *LoCo* and *GloCo* features at each node.

Depending on the clinical application, one might need to minimise uncertainty about the myocardium (for accurate estimation of the wall thickness for example), or about the blood pool (for accurate ejection fraction estimation), or about both the myocardium and the blood pool for visual inspection. Therefore, for each of the three forests - the classic forest, the FA forest and the *LoCo* forest - we find the parameter D for:

1) optimal blood pool segmentation

2) optimal myocardium segmentation

3) optimal myocardium and blood pool segmentation

The corresponding parameters D are reported in Fig. 5.12, 5.13, 5.14, 5.16, 5.17, 5.18, 5.20, 5.21 and 5.22.

Randomness was injected using a small proportion of features m at a node from the pool of available features.

5.6.2 Feature Parameters

The following parameters of the classic features, the FA features, and the *LoCo* and *GloCo* features were used:

1. The classic features. For the Rectangular3D feature, we used a neighbourhood area N of 20 voxels in each direction and the dimensions of the cuboid ranging from 2 to 20 voxels. For the Difference3D and Haar3D features, we used the same neighbourhood area and the same parameters for the rec. as for the Rectangular3D feature.

2. The FA features. Although the FA images contain the most useful information for segmentation, i.e. edges, its information is sparse. Therefore we include a wide range of the parameters for the classic features deployed on the FA images to capture more structural information near the voxel of interest. We used a neighbourhood area N of 25 voxels and dimensions of the cuboid varying from 10 to 30 voxels for all FA features (FARectangular3D, FADifference3D and FAHaar3D).

3. The *LoCo* and *GloCo* features. As described in section 5.5.3, the landmarks for *LoCo* and *GloCo* features are defined by choosing a short-axis slice z and a directional angle θ. After the images are aligned, the short-axis slices across different images correspond. For the *LoCo* and *GloCo* features, we used the short-axis slices ranging from 80 to 180 (the full range spans all the short-axis slices from 1 to 208). Although all slices could be injected into the classifier (the irrelevant slices would be rejected by the optimisation procedure), we chose to include only the meaningful slices where the epicardial and endocardial boundaries are present.

5.6.3 Validation Metrics

For quantitative analysis, the mean and the standard deviation of the Dice similarity coefficient for (i) the myocardium, (ii) the blood pool and (iii) the myocardium and the blood pool are reported in Table 5.1 for each of the three forests.

The Dice similarity coefficient is defined as:

$$Dice = \frac{2 \times |GT \cap Auto|}{|GT| + |Auto|} \tag{5.28}$$

where GT is the ground truth and $Auto$ is the automatic segmentation, $|.|$ is the cardinality of a set.

5.7 Results

The results of the 5-fold cross-validation for each of the three random forests - the classic random forest, the FA random forest and the *LoCo* random forest, and for each of the applications - optimum myocardium segmentation, optimum blood pool segmentation, and optimum both myocardium and blood pool segmentation, are reported in Fig. 5.12, 5.13, 5.14, 5.16, 5.17, 5.18, 5.20, 5.21, 5.22. The 5-fold cross validation was performed for different number of trees in the forest, and the optimum value of parameter D (depth of the tree) turned out to be the same for different number of trees in the forest T. The vertical

blue line in each image shows the optimum value of the parameter D.

Fig. 5.15, 5.19 and 5.23 demonstrate the effect of the size of the forest T for each of the three forests. For all the cases, adding trees to the forest until $T = 4$ dramatically improves the validation accuracy, which further gradually goes up until $T = 16$. There is not much improvement seen after the size of the forest $T = 16$. The results reported in Tables 5.1 were obtained using all $T = 20$ trees in the forest.

The frequency of acceptance of each type of feature at different depths is shown in Fig. 5.24, 5.25 and 5.26, and further discussed below in section 5.8.

For visual results of the segmentations refer to Fig. 5.27.

5.8 Discussion

For all three forests we see a substantial improvement in the validation accuracy compared to the results obtained using the standard classification random forest and the classic features employed on the original images, Table 5.1. For detailed analysis of the proposed features, we shall look into the frequency of selection of each type of feature during the training stage at different tree depths, Fig. 5.24, 5.25 and 5.26.

5.8.1 The Improved Classic Features

The position features are pre-dominantly selected by the decision forest at depths 1-4 when the complexity of the data is the highest, see Fig. 5.24. At this stage, the appearance features poorly distinguish the classes well due to the similar appearance of different tissues or the different appearance of the same tissue. The position feature looks only at the absolute position of a voxel, and hence acts as a prior. After the alignment procedure this prior becomes stronger, and the position features are chosen at the root of the tree more often. The registration also ensures that the appearance features correspond across different images improving the overall segmentation accuracy. Comparing the results of the classic random forest with the results of the random forest using the classic features on the original not

registered images (from the previous chapter) - there is an improvement in the accuracy of the segmentation by approximately 15%, Table 5.1.

5.8.2 The FA Features

As with the classic random forest, the FA forest pre-dominantly selects the position features at the root of the trees and splits the images spatially utilising no intensity information from the image. The FA features dominate over the classic features until depth 7 and use its structural information to split the data at each of the regions. The classic appearance features become prevailing after depth 7. Although the FA features are often selected by the optimisation procedure of the classifier over the classic features, the FA random forest produces similar results to those of the classic random forest (using the registered images). From figures above one can see that the depth of the threes of the FA forest is two levels lower than the one of the classic random forest, and thus, the FA forest requires less time to segment an image. However, there is an additional computational cost associated with computing the FA measure. Therefore, the classic random forest might be preferred over the FA random forest in real time situations where testing time is important.

5.8.3 *LoCo* and *GloCo* Features

Both the position and the *LoCo* and *GloCo* features divide the image domain spatially. In the case of the classic random forest, the position features tend to split the data spatially until the depth 4, and after depth 4 the appearance features prove to separate the classes better within each of the spatial subregions in the image. In the case of the random forest with the *LoCo* and *GloCo* features the behaviour is different. After depth 4, the *GloCo* features are shown to dominate, proving to be a good descriptor of the voxel class label. They continue to dominate until depth 11. This suggests that the *GloCo* features carry more detailed contextual information than the position or the *LoCo* features. Fig. 5.26 suggests that after depth 11 the separation is mainly done using the appearance information within each of the subregions in the image. Note that by splitting the image spatially, the *LoCo*

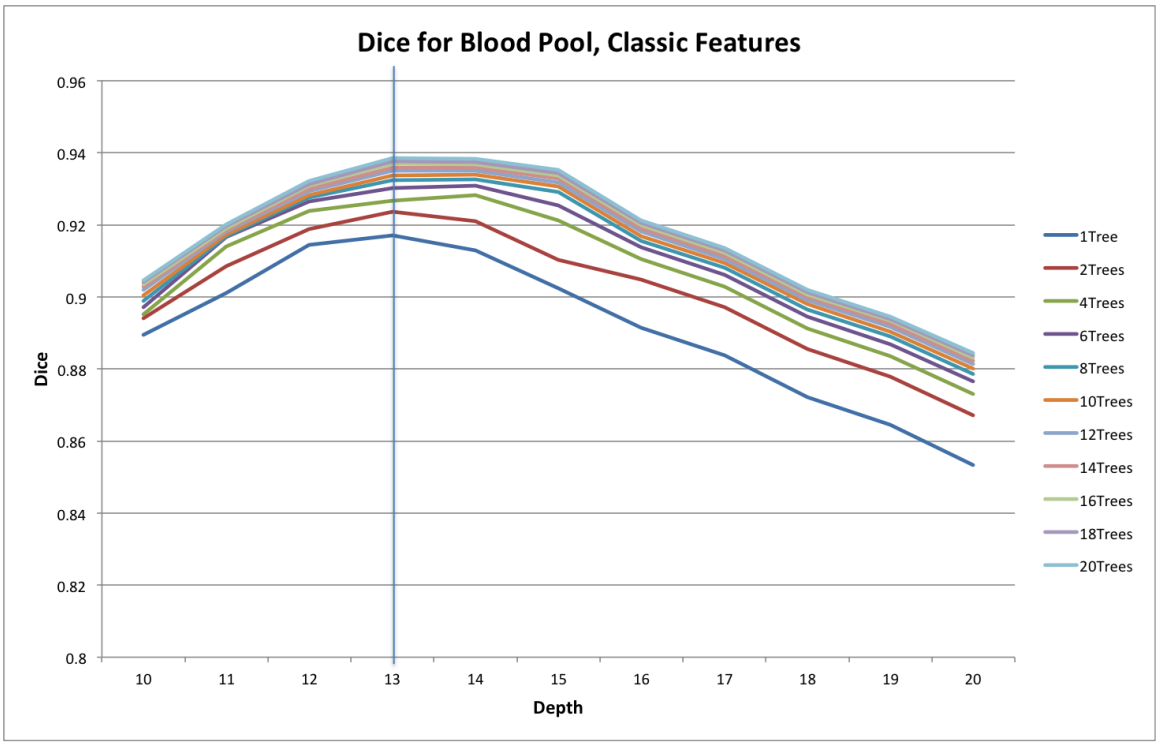


Figure 5.12: Dice similarity coefficient for the blood pool segmentation using the classic forest. The blue vertical line indicates the optimal depth.

and *GloCo* features also encode a constraint on local and global shape variability.

However, the *LoCo* random forest does not outperform the classic random forest on registered images or the FA random forest in the described settings. We caveat our findings with two comments related to the dataset we used. The current training set consists of 20 3D volumes which is relatively small to capture the geometrical complexity of coupled variability of the epicardial and endocardial shapes. Thus firstly one would hypothesise an improvement of the Dice measure for the myocardial segmentation with an increase of training dataset size. Secondly one cannot guarantee that the small training dataset used fairly captures the variability seen in the validation dataset. Having said this, the results are encouraging and convincingly demonstrate the usefulness of the *LoCo* and *GloCo* features for 3D echocardiography segmentation. The proposed features need to be tested on a larger training set to draw strong conclusions on the accuracy for myocardial segmentation. This

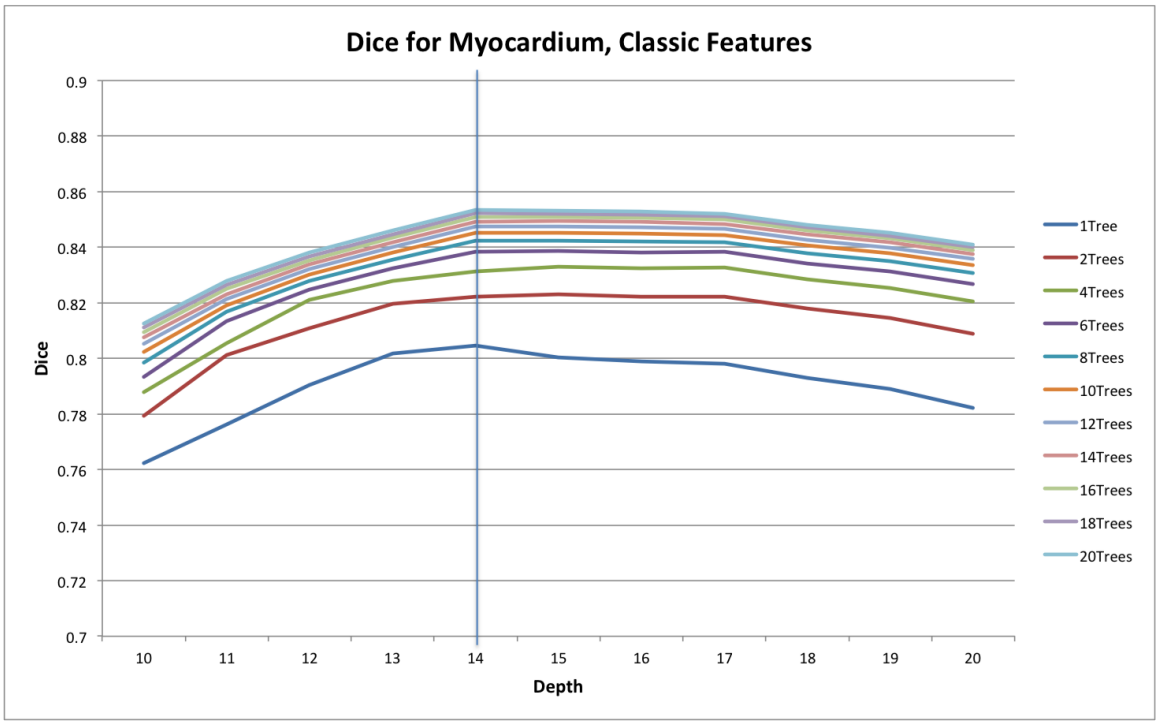


Figure 5.13: Dice similarity coefficient for the blood pool segmentation using the classic forest. The blue vertical line indicates the optimal depth.

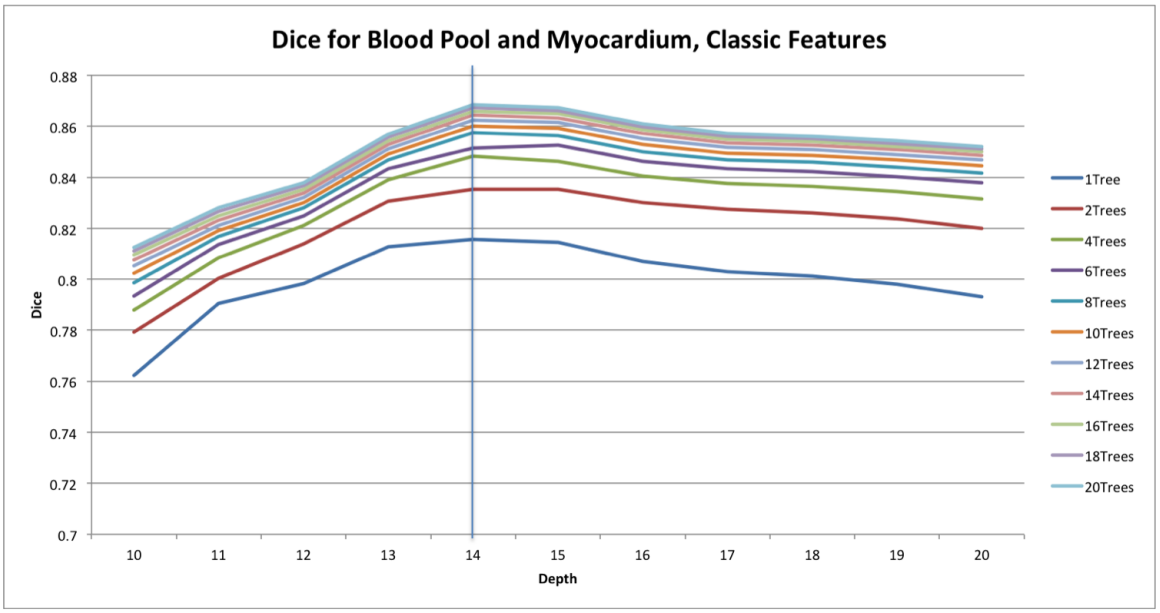


Figure 5.14: Dice similarity coefficient for the blood pool and myocardium segmentation using the classic forest. The blue vertical line indicates the optimal depth.

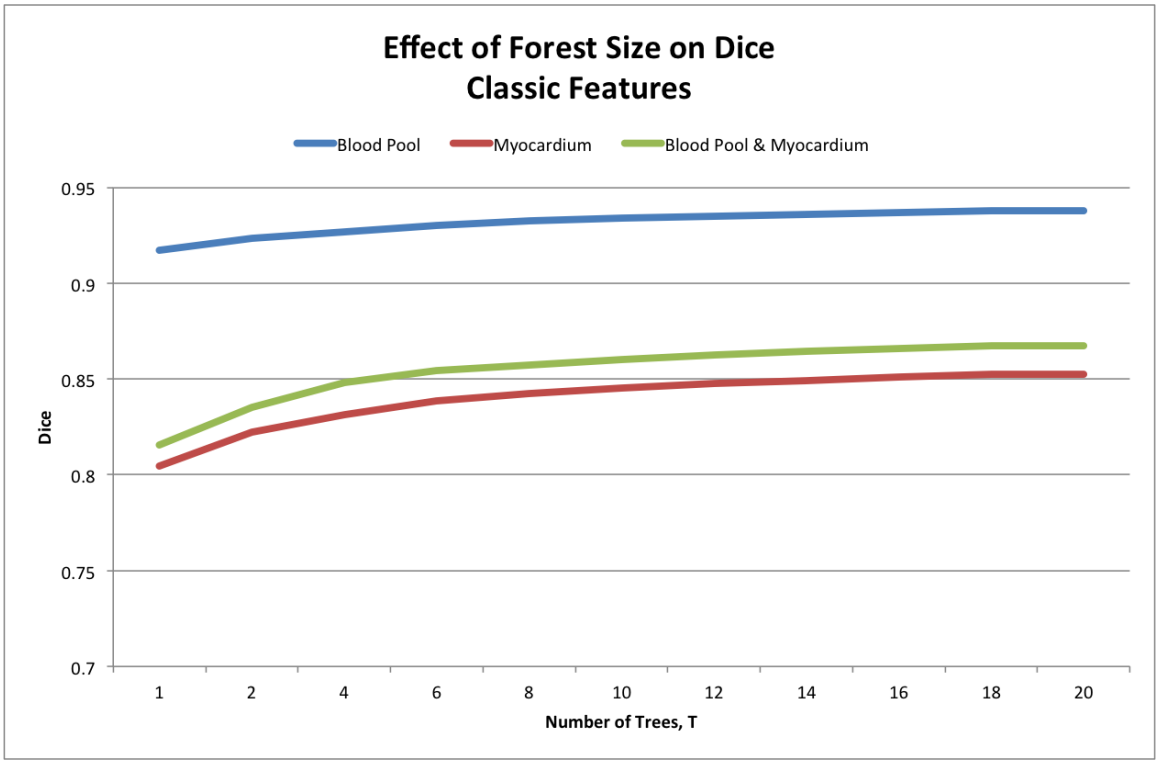


Figure 5.15: Effect of forest size on the Dice coefficient. The classic random forest.

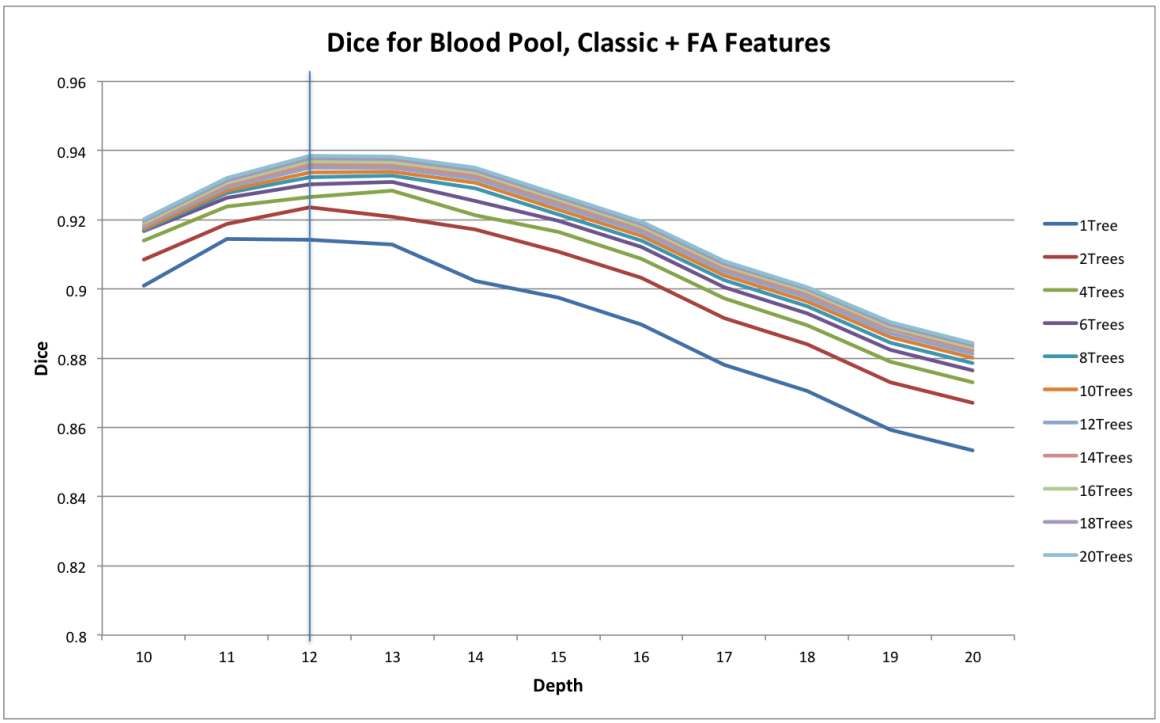


Figure 5.16: Dice similarity coefficient for the blood pool segmentation using the FA forest. The blue vertical line indicates the optimal depth.

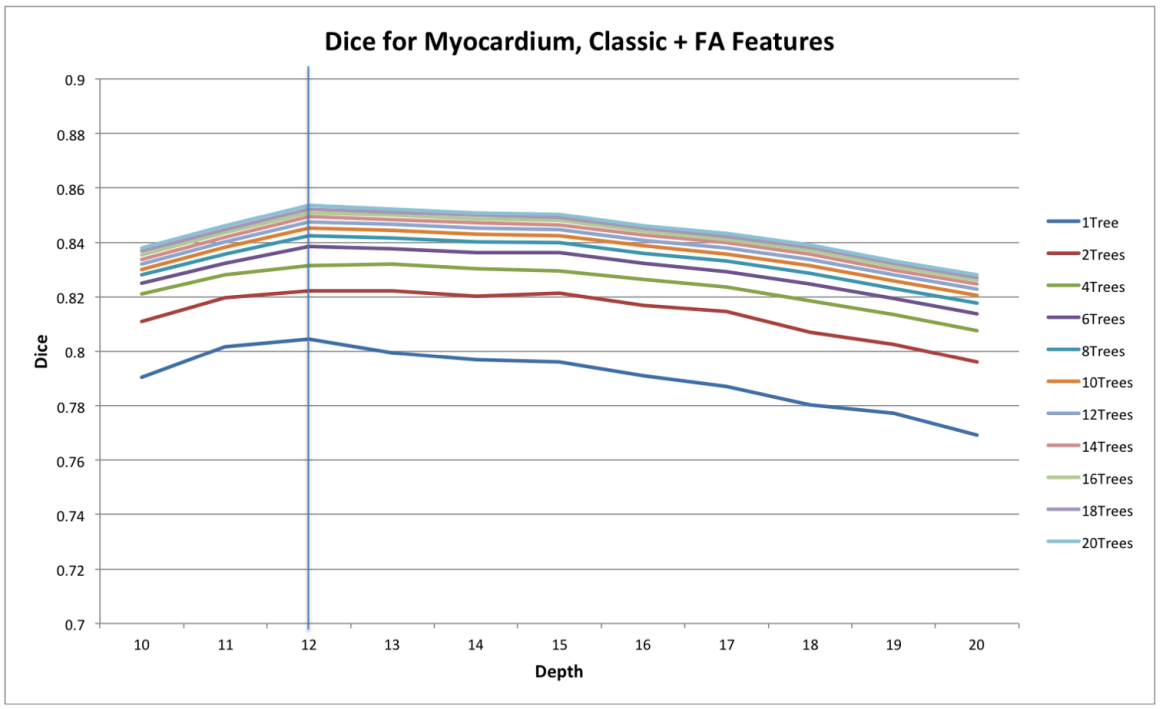


Figure 5.17: Dice similarity coefficient for the myocardium segmentation using the FA forest. The blue vertical line indicates the optimal depth.

will be the subject of future work.

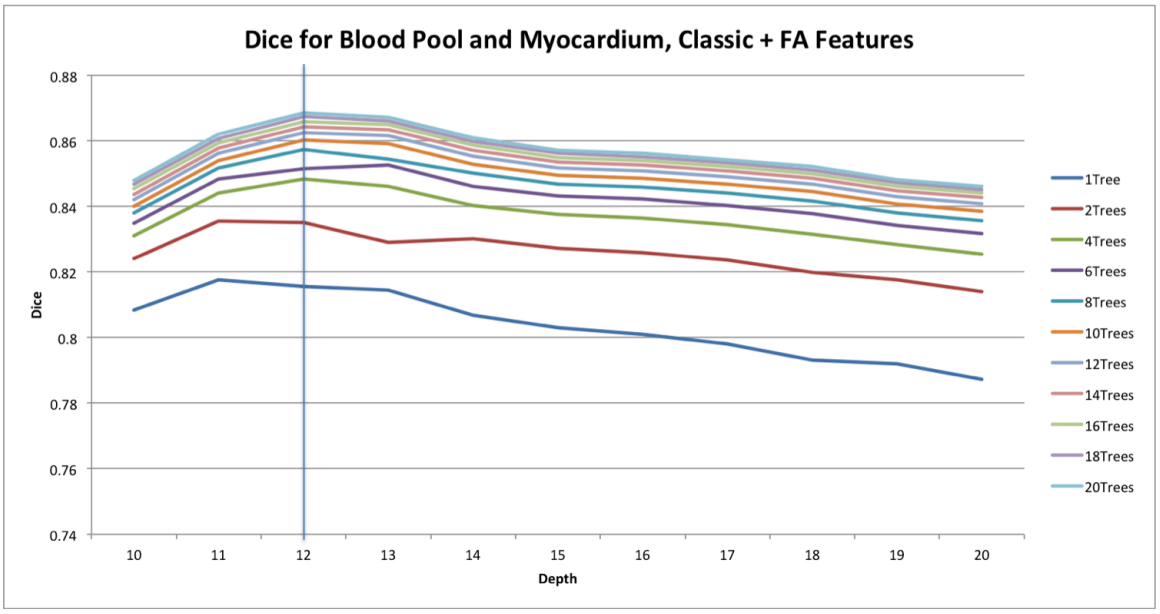


Figure 5.18: Dice similarity coefficient for the blood pool and myocardium segmentation using the FA forest. The blue vertical line indicates the optimal depth.

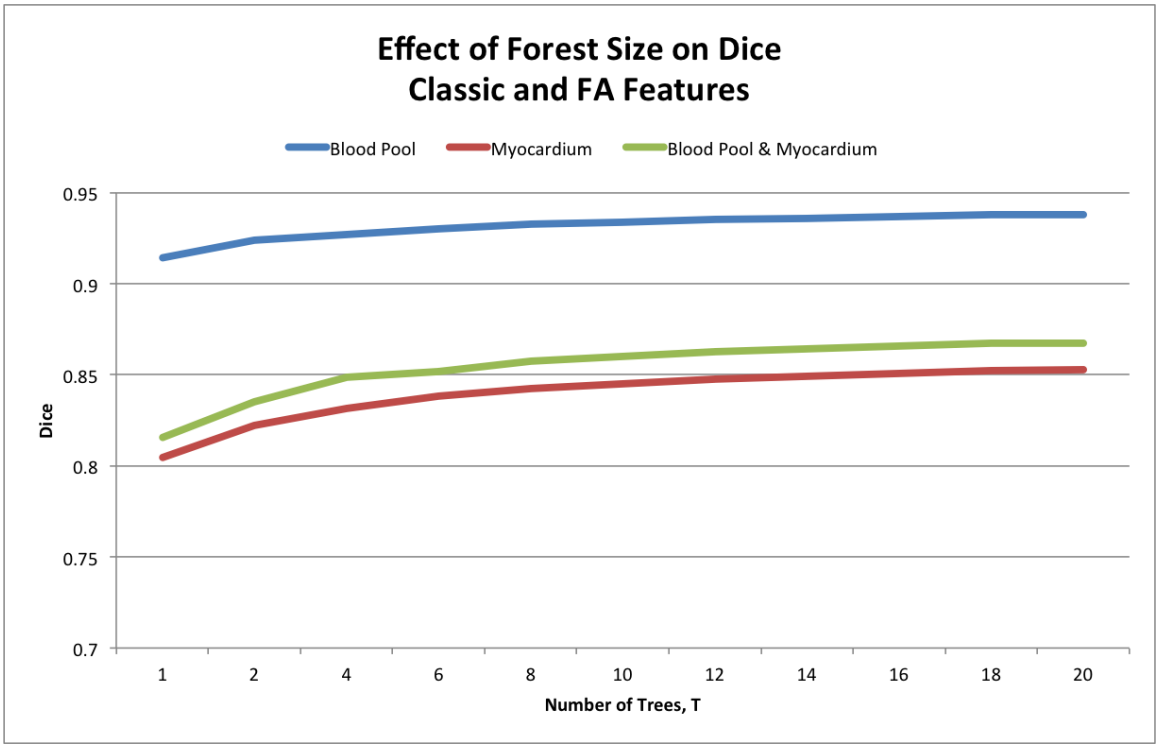


Figure 5.19: Effect of forest size on the Dice coefficient. The FA random forest.

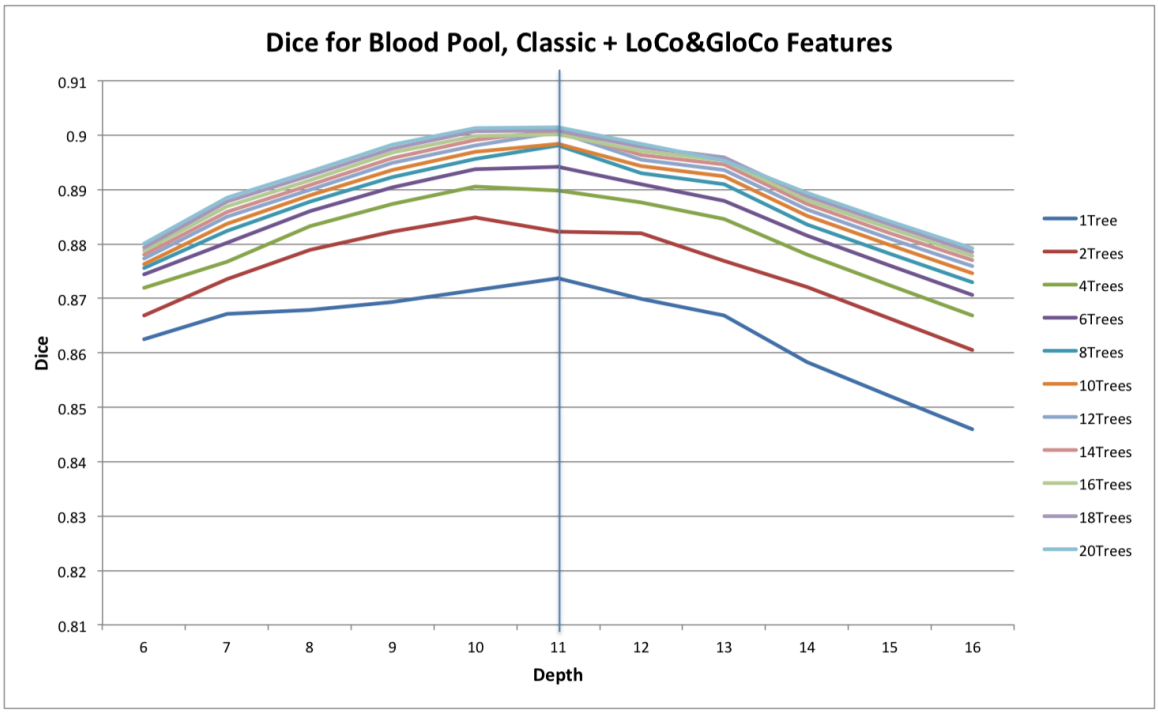


Figure 5.20: Dice similarity coefficient for the blood pool segmentation using the *LoCo* forest. The blue vertical line indicates the optimal depth.

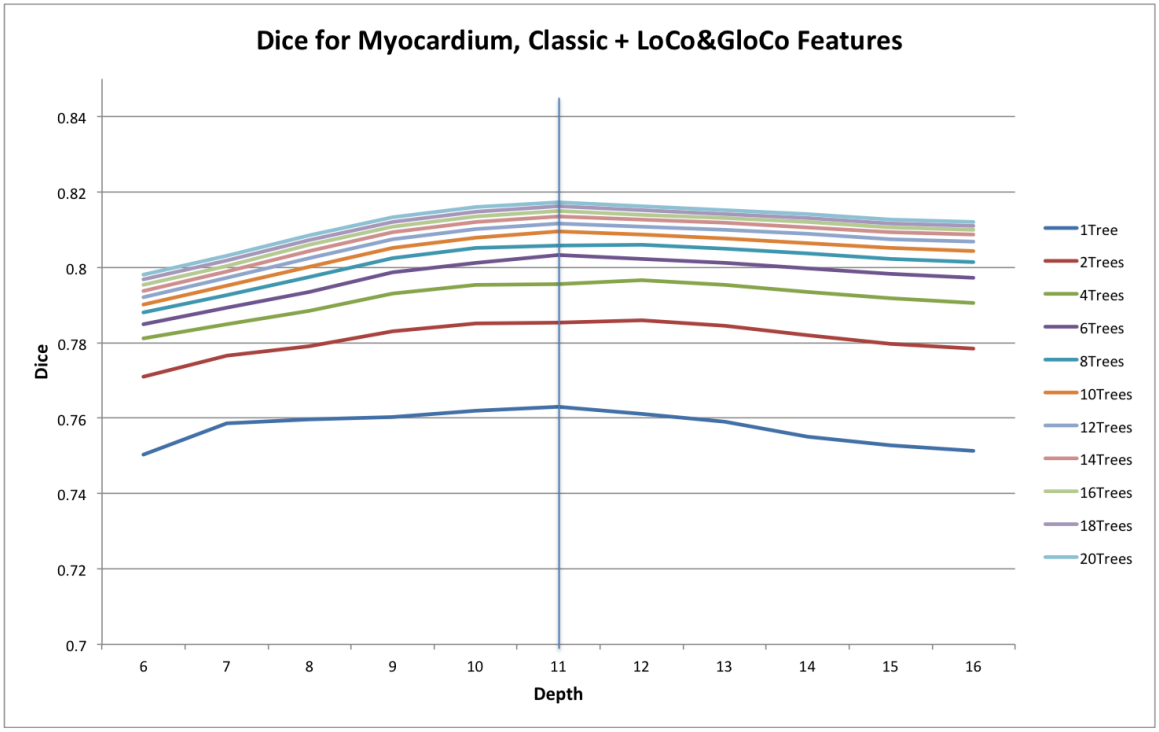


Figure 5.21: Dice similarity coefficient for the myocardium segmentation using the *LoCo* forest. The blue vertical line indicates the optimal depth.

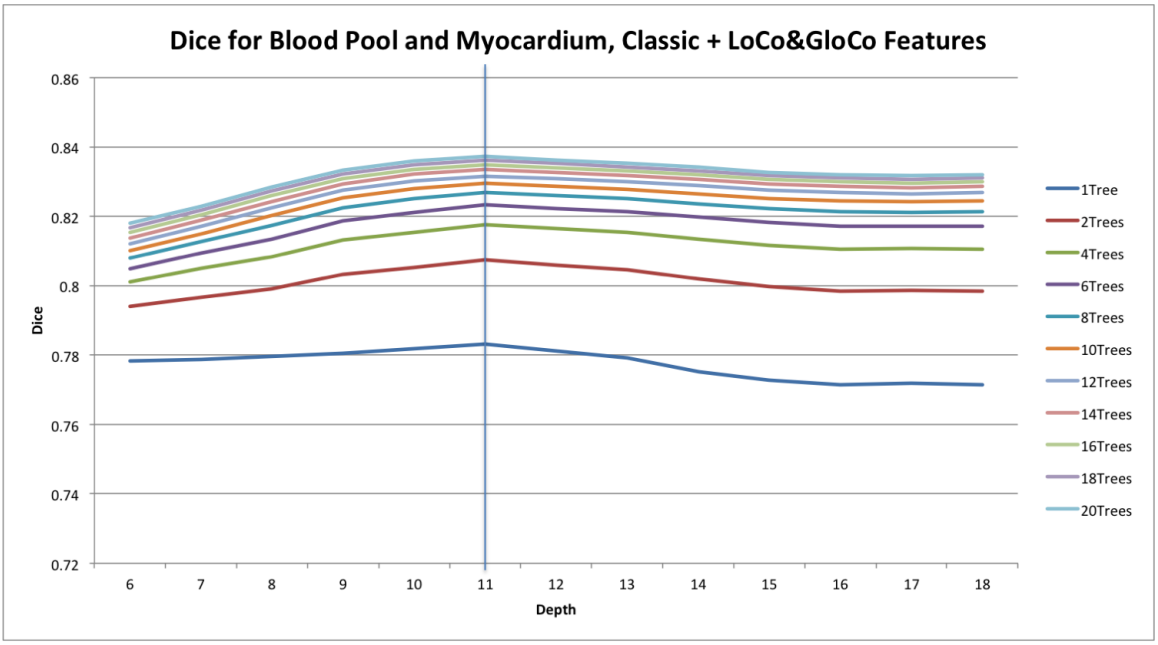


Figure 5.22: Dice similarity coefficient for the blood pool and myocardium segmentation using the *LoCo* forest. The blue vertical line indicates the optimal depth.

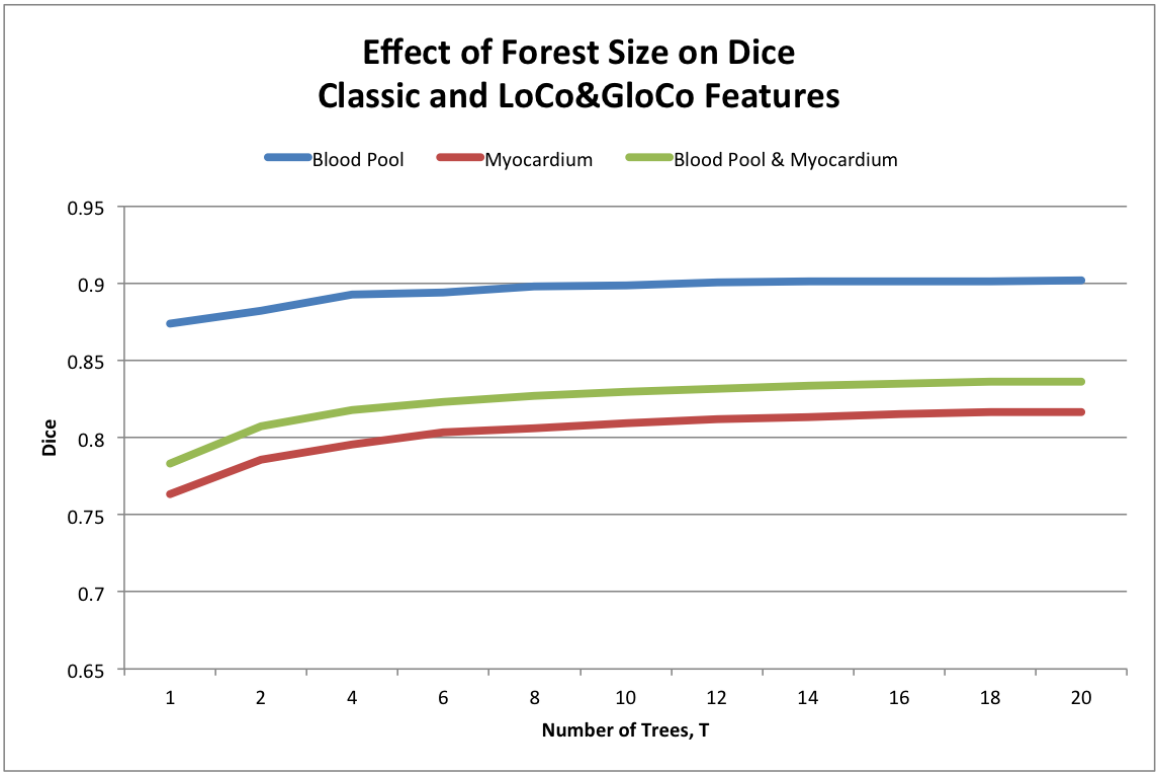


Figure 5.23: Effect of forest size on the Dice coefficient. The *LoCo* random forest.

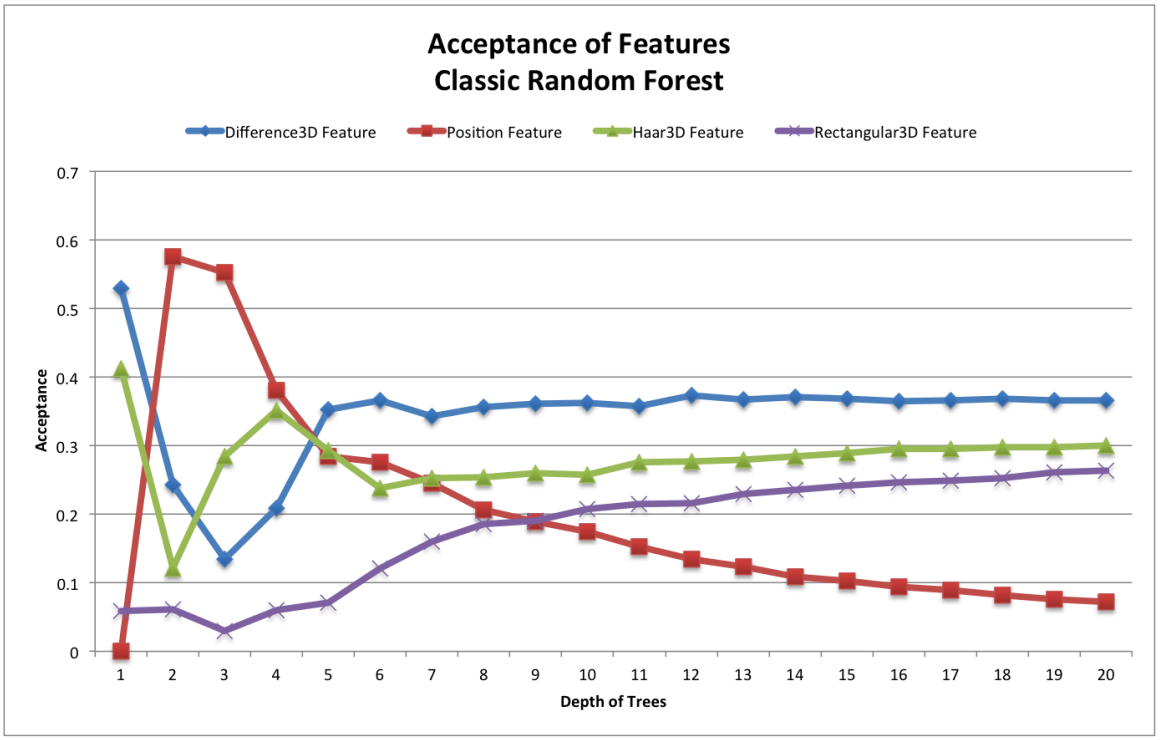


Figure 5.24: Feature acceptance for the classic random forest.

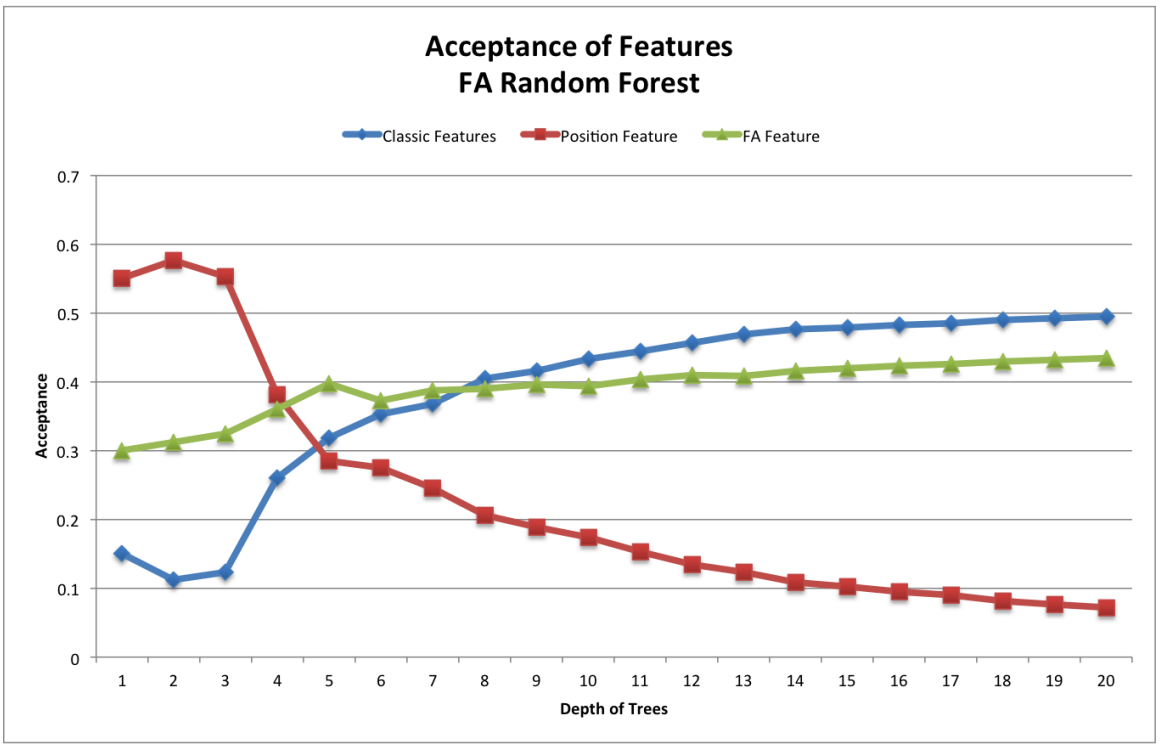


Figure 5.25: Feature acceptance for the FA random forest.

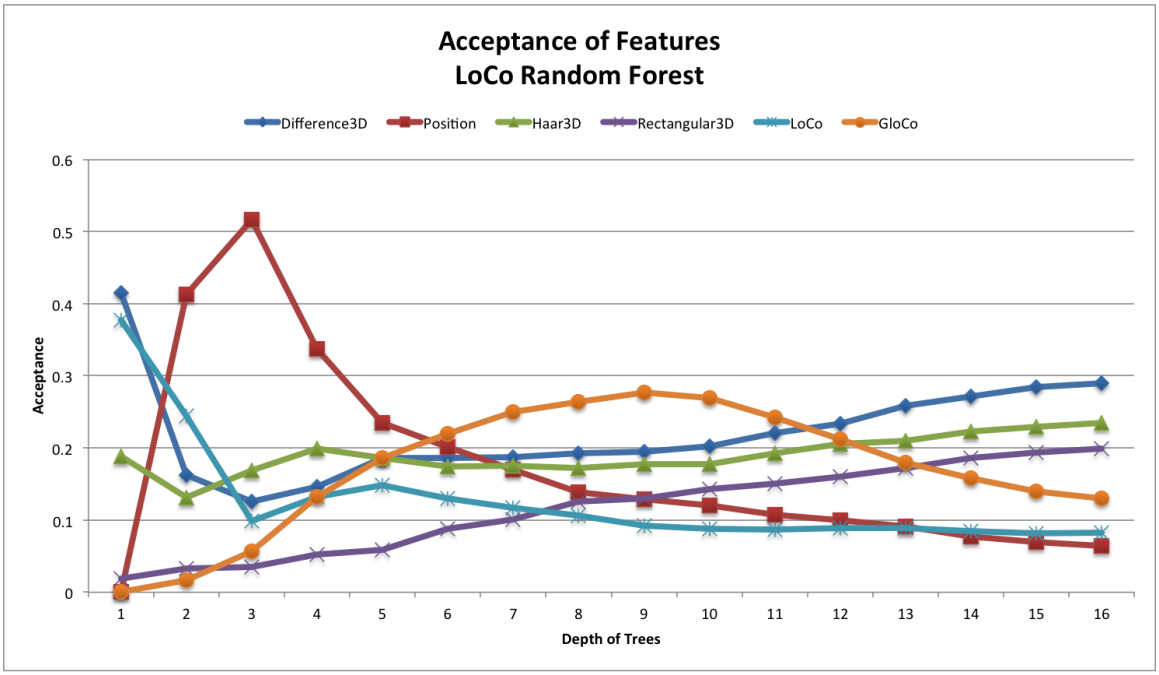


Figure 5.26: Feature acceptance for the *LoCo* random forest.

Random Forest with Classic Features on Original Images

	Dice
Myocardium	0.78 ± 0.06
Blood Pool	0.89 ± 0.04
Blood Pool and Myocardium	0.78 ± 0.07

Random Forest with Classic Features on Registered Images

	Dice
Myocardium	0.85 ± 0.05
Blood Pool	0.94 ± 0.02
Blood Pool and Myocardium	0.87 ± 0.05

FA Random Forest

	Dice
Myocardium	0.85 ± 0.06
Blood Pool	0.94 ± 0.03
Blood Pool and Myocardium	0.87 ± 0.05

***LoCo* Random Forest**

	Dice
Myocardium	0.78 ± 0.05
Blood Pool	0.89 ± 0.03
Blood Pool and Myocardium	0.84 ± 0.05

Table 5.1: Mean (μ) $\pm$ standard deviation (σ) of the Dice coefficient for (i) the myocardium, (ii) the blood pool, and (iii) myocardium with the blood pool using different random forests.

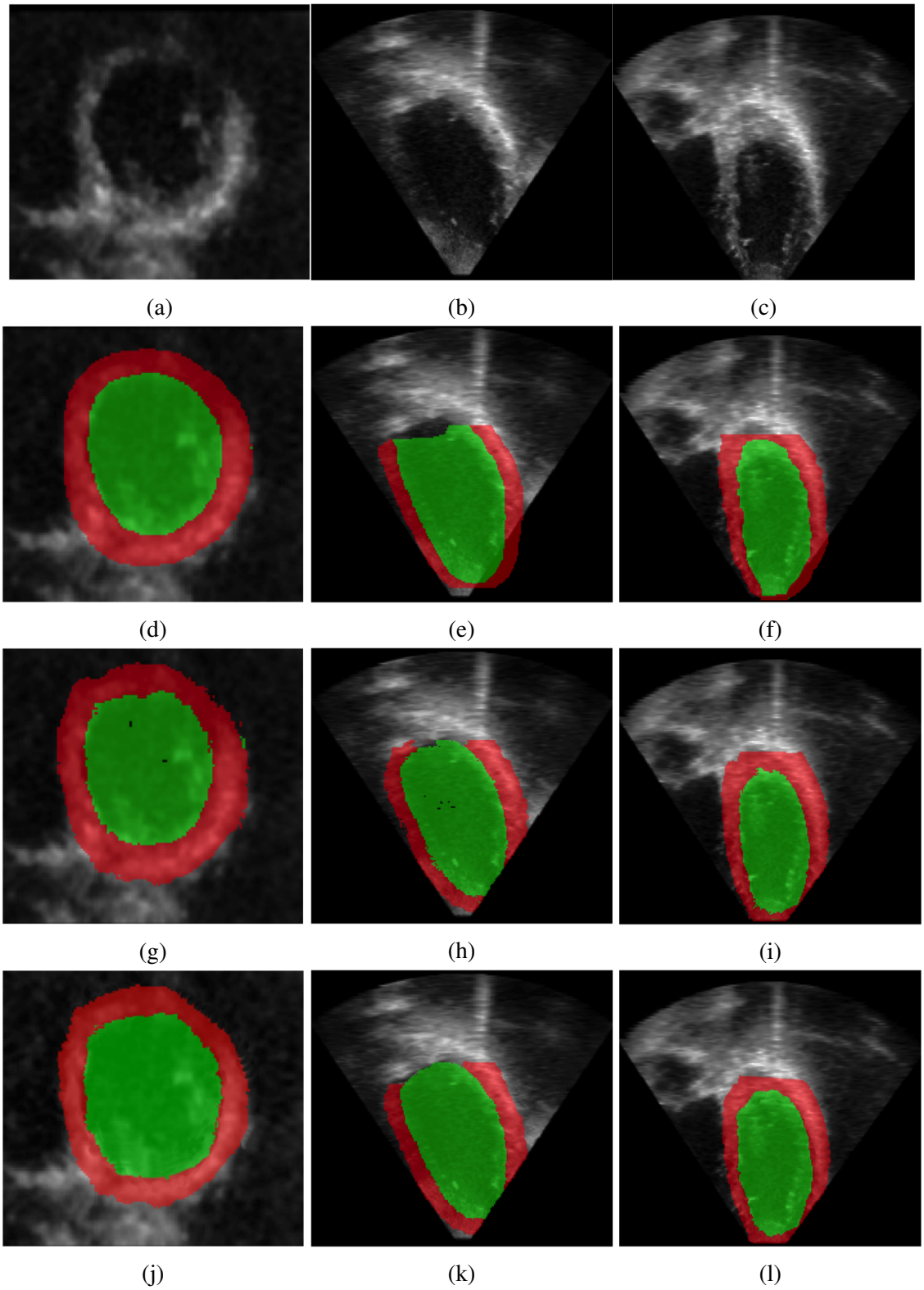


Figure 5.27: Qualitative results of the 3D Echocardiography segmentation with the classic random forest on the original images and the classic random forest on the registered images. (a), (b), (c) - original images, (d), (e), (f) - manual segmentation, (g), (h), (i) - automatic segmentation on the original images, (j), (k), (l) - automatic segmentation on the registered images.

Chapter 6

Standard Views Extraction from 3D Echocardiography using Regression Voting

6.1 Introduction

Automatic extraction of the standard views from 3D echocardiography is an important step in 3D echo clinical workflow. In clinical practice, cardiologists analyze the three standard views 2-chamber view (2C), 3-chamber view (3C) and 4-chamber view (4C), Fig. 6.1. This plane detection is done visually and is subjective, time-consuming and operator-dependent. Therefore, quick and reliable methods for automatic detection of the standard views in 3D echocardiography are pertinent towards the development of accurate assessment of heart function.

In this chapter we present a fully automatic, one-step method to extract the best 2C, 3C and 4C image planes from 3D ultrasound volumes of the left ventricle (LV) using regression voting. The traditional two-step detection approach, where a number of candidate planes are first extracted from the volume, followed by classification of the extracted planes, has two weaknesses: (i) the candidate selection space is extensive and the actual planes can

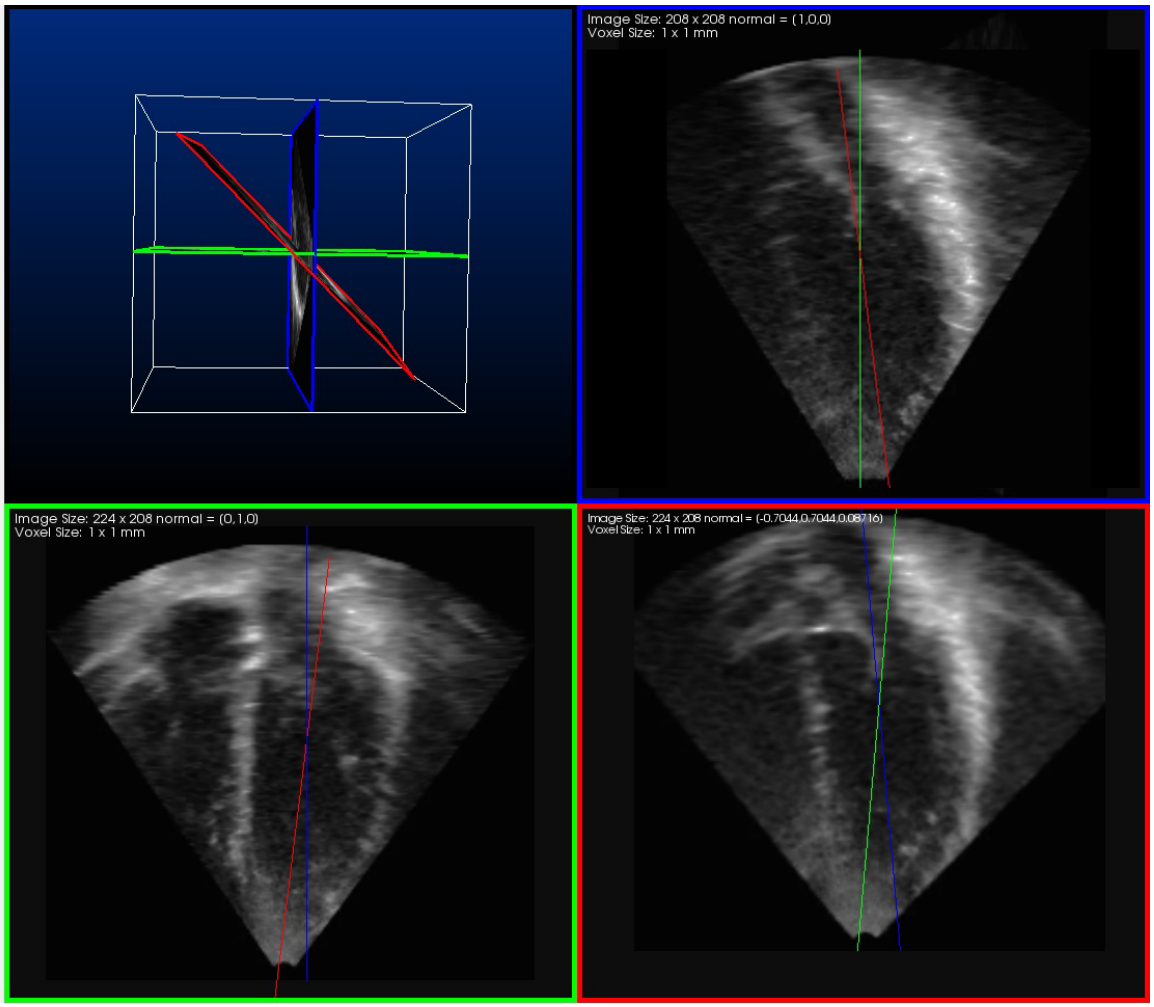


Figure 6.1: 2-chamber (bottom right), 3-chamber (top right) and 4-chamber standard (bottom left) views in 3D echocardiography.

be missed during the selection procedure, (ii) during classification, only information from the candidate slice is considered, ignoring any structural information in the rest of the volume. In this work, we use a natural parameterization of the plane, which considers the whole space of possible planes. During regression, every voxel of the volume votes on the parameters of the plane, and hence, the whole 3D volume participates in the decision.

We also present a novel regression algorithm, called a Class-Specific Regression Forest, which integrates class label information into the training of the regression forest in order to exclude from voting irrelevant to the problem classes and focus on the relevant ones during training. Two regression algorithms are proposed for the task of standard views extraction from 3D echocardiography:

(i) the Regression Random Forest

(ii) the Class-Specific Regression Random Forest.

Thus, the contributions of this work are twofold. The first contribution is in the use of a natural, one-step approach for plane extraction from a 3D volume using regression voting. The second contribution is the Class-Specific Regression Forest, where the class label information is incorporated into the training procedure to help maximize the confidence of the continuous outputs.

6.2 Regression Random Forests for Plane Detection in 3D volumes

The general random forest framework has been described in chapter 2. The adaptation to the regression task consists of defining the regression specific energy function I and the leaf prediction model $p(\mathbf{d}|l(\mathbf{v}))$. Unlike classification decision forests, regression forests estimate non-linear mapping from the input space to the space of continuous outputs. In plane detection, the regression forest learns the non-linear mapping directly from voxels to the parameters of the plane (the plane is parametrized by the distance to the plane from the origin $\mathbf{b} = (b_1, b_2, b_3)$). Regression forests found use in medical imaging fairly recently and first were applied in (87) for anatomy detection in CT scan of the whole body. Since then, the regression forests have been successfully applied in a few medical image analysis applications (96; 118; 119; 120).

6.2.1 Training

Let $\mathbf{v}$ be a voxel in a 3D volume, represented by a feature vector $\mathbf{f} = (f_1, f_2, .., f_n)$; each voxel is assigned a distance vector $\mathbf{d} = (\mathrm{d}x, \mathrm{d}y, \mathrm{d}z)$ to the plane Γ. The training dataset comprises a set of uniformly sampled voxels $\{\mathbf{V}_i = (\mathbf{f}_i, \mathbf{d})\}$. The regression forest consists of a set of trees; each tree is constructed using the training dataset $\mathbf{V}$. The training starts at the root node with all available training data, Fig. 6.2. The data is split to the left and the

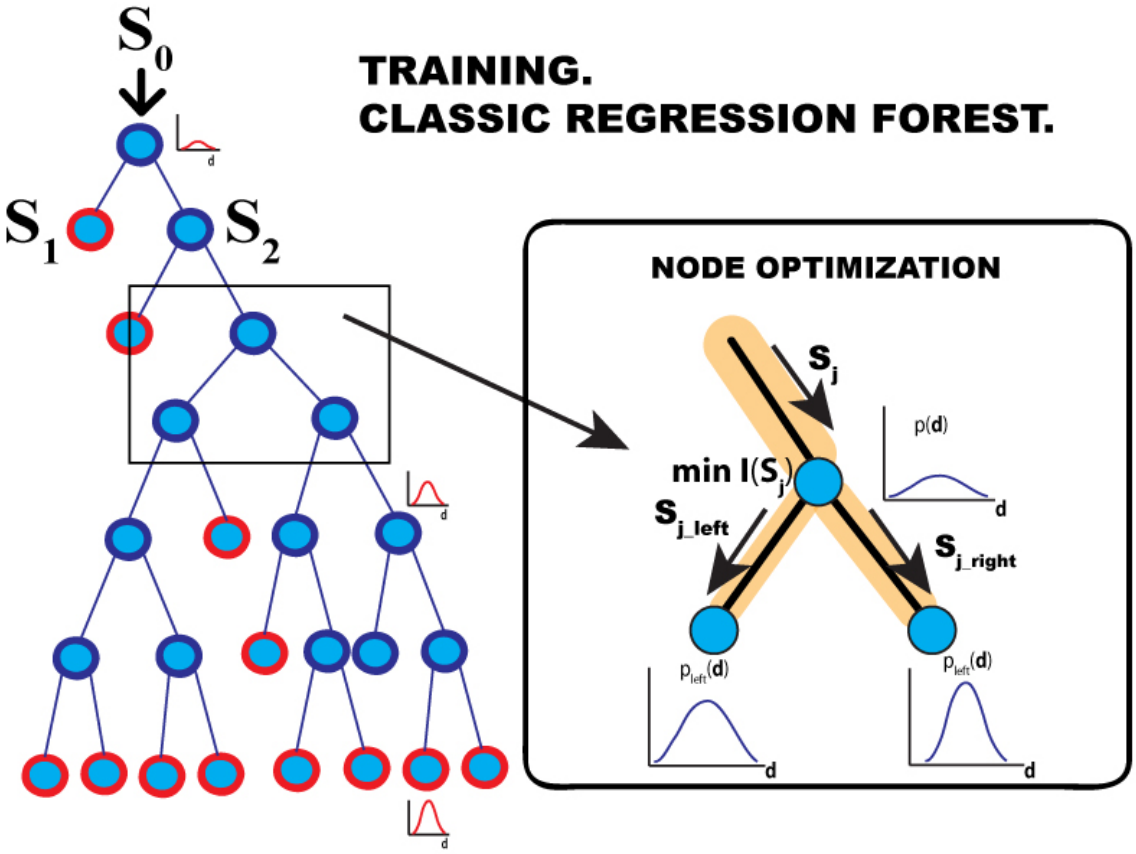


Figure 6.2: Training and node optimization for conventional regression forest. The training starts at the root node, where the data is split into the right and the left nodes so as to minimise the average uncertainty about the plane parameters **d** over the child nodes (eq. 6.4. The splitting continues until one of the stopping criteria is reached. The leaf nodes save the three-dimensional distance vectors **d**.

right child nodes according to the node optimization criteria described below. The splitting continues until the criteria to stop are reached. Each leaf node l stores the three-dimensional distance vectors **d** that arrived at this node. Randomness is injected via a small number of randomly selected features ($m = 100$) at each node.

6.2.2 Node Optimization

Each split node is assigned a binary test, which is defined as

$$h_i(\mathbf{v}) = \lfloor f_i(\mathbf{v}) < \tau \rfloor \tag{6.1}$$

where $f_i \in \{f_1, f_2, , f_n\}$ is the feature associated with the node, and $\lfloor .. \rfloor$ is the indicator

function. A set of m features is randomly selected from the pool of features $\mathbf{f} = (f_1, f_2, , f_n)$, $t = 10$ random thresholds τ are tested per feature to find the pair (f_i, τ) that maximizes the energy function 6.4 defined below.

The splitting minimizes the average uncertainty in the continuous parameter values over the child nodes:

$$\min(p_{left} I^{left} + p_{right} I^{right}) \tag{6.2}$$

where $I^{left,right}$ is the uncertainty of $\mathbf{d}$. These uncertainties are chosen to encode the sample variance of $\mathbf{d}$:

$$I(S^l) = \frac{1}{|S^l| - 1} \sum_{d \in S^l} \left(\mathbf{d} - \frac{1}{|S^l|} \sum_{d' \in S^l} \mathbf{d}' \right)^2 \tag{6.3}$$

Combining 6.2 and 6.3 yields the following objective function:

$$I_{\mathbf{d}}(S) = \sum_{l=\{left,right\}} p^l \left(\frac{1}{|S^l| - 1} \sum_{\mathbf{d} \in S^l} \left(\mathbf{d} - \frac{1}{|S^l|} \sum_{d' \in S^l} \mathbf{d}' \right)^2 \right) \tag{6.4}$$

6.2.3 Testing

Given a new unseen image I, all voxels of that image are passed through the trees of the forest. Each voxel traverses the tree until it reaches a leaf node l. The voxel then votes for the parameters of the plane $\mathbf{b} = (b_1, b_2, b_3)$ using the distance vectors $\mathbf{d}$, which are stored at the leaf. The mapping of the stored vectors $\mathbf{d}$ to the plane parametrization $\mathbf{b}$ can be expressed using Fig. 6.3:

$$\mathbf{b} = \mathbf{d} + \frac{\mathbf{d}}{|\mathbf{d}|} \cdot |\mathbf{v}|_{\mathbf{b}} = \mathbf{d} + \frac{\mathbf{d}}{|\mathbf{d}|} \cdot \frac{\mathbf{v} \cdot \mathbf{d}}{\mathbf{d}} = \tag{6.5}$$

$$\mathbf{d} \left(\frac{\mathbf{v} \cdot \mathbf{d}}{|\mathbf{d}|^2} + 1 \right) \tag{6.6}$$

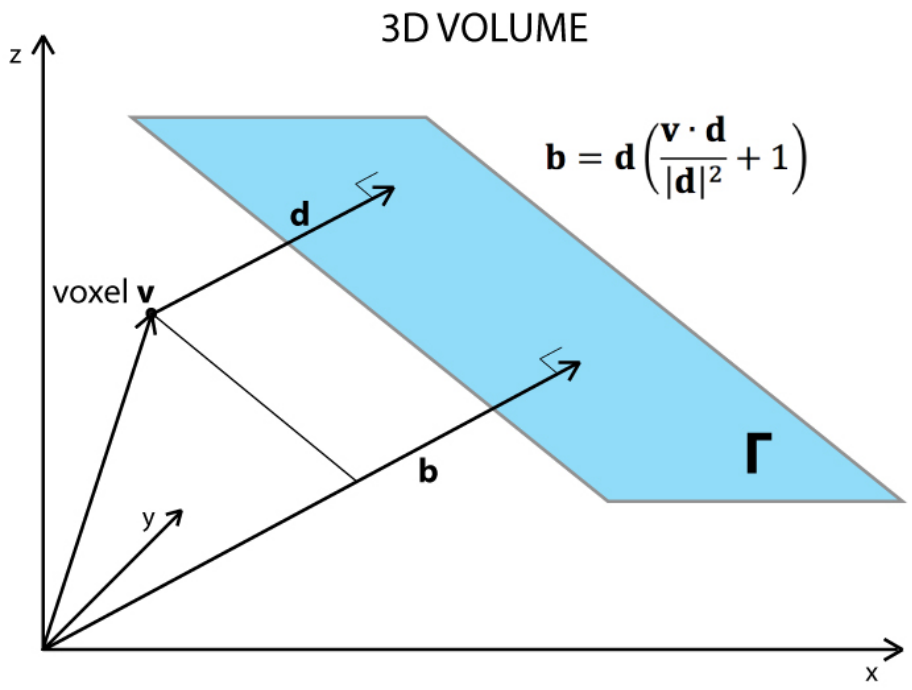


Figure 6.3: Mapping voxels-to-the-plane distance **d**) to origin-to-the-plane distance **b**.

where **v** is a voxel.

We are interested in the probability $p(\mathbf{b}|l(\mathbf{v}))$:

$$p\left(\mathbf{b}|l\left(\mathbf{v}\right)\right) = p\left(\mathbf{b} = \mathbf{d}_c\left(\frac{\mathbf{v}\cdot\mathbf{d}_c}{|\mathbf{d}_c|^2}+1\right)|l\left(\mathbf{v}\right)\right) \tag{6.7}$$

This distribution is estimated by a sum of Dirac measures δ_d as in (89):

$$p\left(\mathbf{b}|l\left(\mathbf{v}\right)\right) = \sum_{\mathbf{d}\in S^{l(\mathbf{v})}} \delta_d\left(\mathbf{d}\left(\frac{\mathbf{v}\cdot\mathbf{d}}{|\mathbf{d}|^2}+1\right)\right) \tag{6.8}$$

During testing the votes from voxels that have low confidence leaf predictors are excluded. We choose the sample variance of **d** as the uncertainty measure of the leaf predictors and set a threshold K on it to control the maximum uncertainty of the participating distributions $p(\mathbf{d})$. The parameters K is found by cross-validation along with the maximum depth of the trees D.

The votes from all voxels and all trees are collected to produce a single posterior prob-

ability distribution of the plane parameters $\mathbf{b}$ for image I:

$$p(\mathbf{b}|I) = \frac{1}{N_{voxels}} \sum_{\mathbf{v}} \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|S^l(\mathbf{v})|} \sum_{d \in S^l(\mathbf{v})} \delta_d \left(\mathbf{d} \left(\frac{\mathbf{v} \cdot \mathbf{d}}{|\mathbf{d}|^2} + 1 \right) \right) \tag{6.9}$$

where N_{voxels} is the number of voxels that participated in voting, T is the number of trees in the forest.

The maxima of the measure 6.9 can be searched by applying a Parzen estimator with a Gaussian kernel G. The parameters of the plane are then extracted as follows:

$$\mathbf{b}_{plane} = max_{\mathbf{b}} p(\mathbf{b}|I) * \mathbf{G} \tag{6.10}$$

The bandwidth of the Gaussian kernel is found by cross-validation.

6.3 Class-Specific Regression Random Forests for Plane Detection in 3D volumes

Here, we propose to integrate voxel label information into the training of the regression forest algorithm, which yields the Class-Specific Regression Forest. This extra information allows to exclude from voting semantically irrelevant to the problem tissues or regions in the image, and enrich the learning procedure focusing on the relevant parts of the image. Specifically, for standard views detection from 3D echocardiographic images of the LV, information beyond the left and right ventricles does not contain any useful knowledge about the problem. However, due to their high proportion in the image, the background voxels dominate in the learning process, resulting in the low confidence predictors, and, therefore, must be excluded from voting. In the Class-Specific Regression Forest, two objective functions are used to optimize for the class label and the class-conditional continuous parameters of the plane. This ensures that a single class label dominates at each leaf, and the confidence of its corresponding class-conditional parameter distribution is high. At the training stage only voxels from the myocardium and the blood pool parti-

cipate in the optimization of the plane parameters. During testing, each voxel uses only one high-confidence class-conditional distribution per leaf node for prediction of the plane. Predictors from other classes are excluded from voting.

This work takes a similar approach to that of Hough Forests in (89). During testing, the datapoints from activated leaf nodes vote directly in the Hough accumulator space. However, the Hough forests are trained for object classification, and the votes are allowed only from the instances of the same object, whereas in our work multiple classes, i.e. myocardium and blood pool, are allowed to vote for the parameters of the plane.

6.3.1 Training

The framework of the training procedure of the Class-Specific Regression Random Forest is similar to the one of the classic Regression Random Forest: the training starts at the root node and the data is split to the left and the right child nodes. The splitting continues until one of the criteria to stop is reached. However, with the Class-Specific Regression Forest each voxel $\mathbf{v}$ is assigned both the voxel class label c, and the distance vector $\mathbf{d} = (\mathrm{d}x, \mathrm{d}y, \mathrm{d}z)$ to the plane of interest Γ. The training dataset consists of uniformly sampled voxels $\{\mathbf{V}_i = (\mathbf{f}_i, \mathbf{d}_i, c_i)\}$. The node optimisation criteria accounts for class label information and is described below. Unlike the leaf nodes of the conventional regression forest, each leaf node l of the Class-Specific Regression Forest forest stores the discrete probability distribution of the class label $p(c|l)$, estimated as the proportion of voxels per class that reached the leaf, and three-dimensional class-conditional distributions $p(\mathbf{d}|c_i)$, Fig. 6.6.

6.3.2 Node Optimization

Just as with the classic Regression Random Forest, each split node is assigned the binary test $h_i(\mathbf{v}) = \lfloor f_i(\mathbf{v}) < \tau \rfloor$ that maximizes one of the objective functions defined below. The

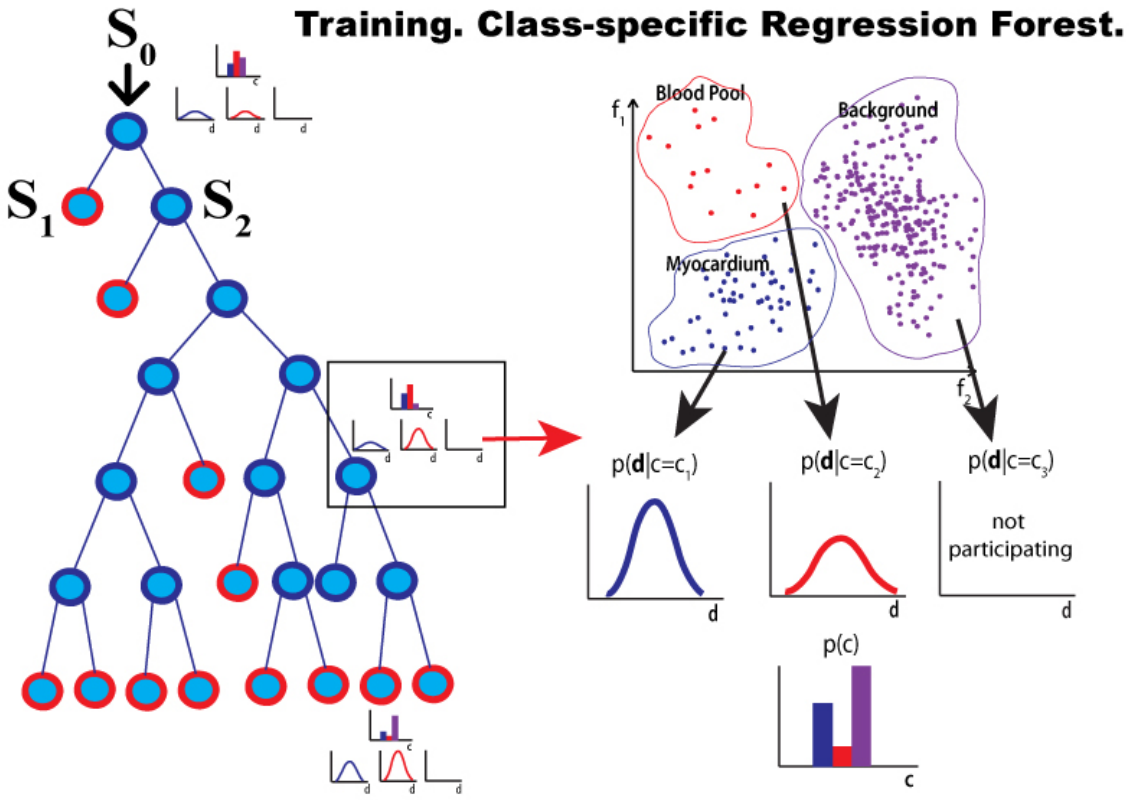


Figure 6.4: Training the Class-Specific Regression Random Forest. The training starts at the root node, the data is split into its right and left child nodes according to the node optimisation procedure described in section 6.3.2. The splitting continues until one of the stopping criteria is reached. The leaf nodes store both the class label distribution $p(c)$ and the class conditional distributions of the plane parameters $p(\mathbf{d}|c_i)$.

splitting minimizes the average uncertainty in the parameter values over the child nodes:

$$\min(p_{left}I^{left} + p_{right}I^{right})$$

Note, that p_{left} and p_{right} should model the probabilities that a voxel will be sent to the left or to the right child node during testing, and not training. For example, if p_{left} is very small and p_{right} is large, the node split function will put its emphasis on minimizing the I^{right} uncertainty at the right child node. This is done with the assumption that most of datapoints at testing will be sent to the right child node, making I^{right} more important. In practice, the probabilities p_{left} and p_{right} are estimated as a proportion of the training voxels sent to the left and to the right child nodes. However, in many cases when the training datapoints are sampled to have a uniformly distributed class label, these estimations are

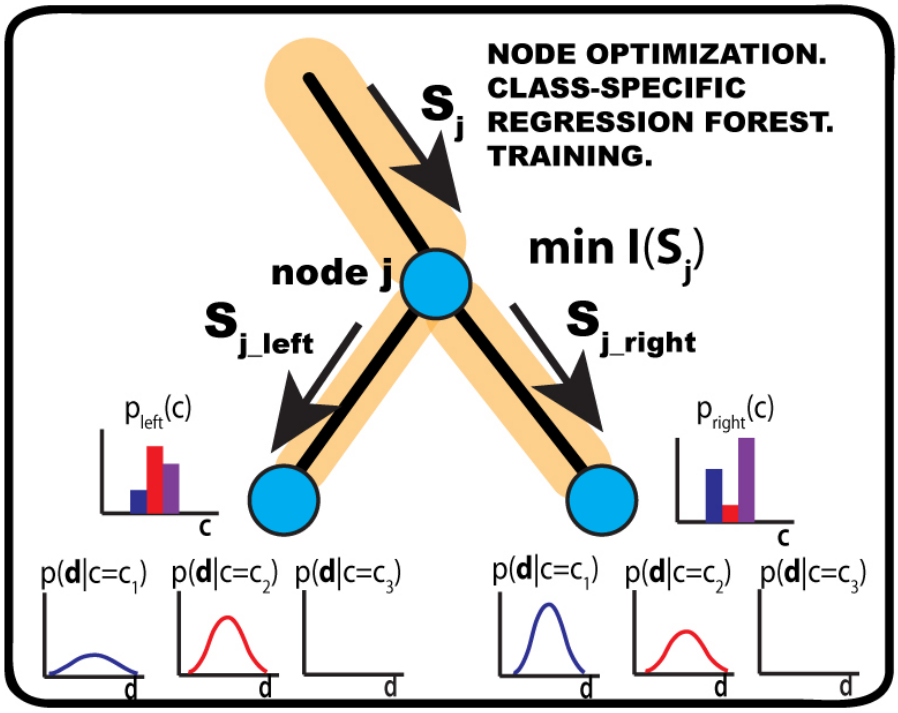


Figure 6.5: Node optimization for the Class-Specific Regression Forest. The data arrived at a split node j is partitioned so as to optimise for either the class label c (eq. 6.14) or the continuous parameters of the plane $\mathbf{d}$ (eq. 6.13). Conditional distributions $p(\mathbf{d}|c = background)$ are excluded from the optimisation process since they will not be used as predictors at the testing stage.

incorrect due to the fact that no such sampling is possible at the testing stage. Therefore, we sample the training dataset uniformly regardless to the class label c, and estimate $p_{left} = \frac{|S^{left}|}{|S|}$ and $p_{right} = \frac{|S^{right}|}{|S|}$, where $|S^{left}|$ and $|S^{right}|$ are the number of datapoints sent to the left and to the right child nodes, $|S|$ is the number of datapoints in the parent node.

The ideal splitting is the one that maximizes the confidence of the predictors in the leaf nodes. In this work, we are interested in the weighted class-conditional parameter distributions $p(\mathbf{d}|c_i)p(c)$ (more details in section 6.3.3), thus optimization should be both for $p(\mathbf{d}|c)$ and $p(c)$. We therefore define the following uncertainty measure for $p(\mathbf{d}|c_i)$ at the child nodes $l = \{left, right\}$:

$$I^l(S^l) = \sum_c I(S^l_c)p(c|S^l) \tag{6.11}$$

where $U(S^l_c)$ is the class-specific uncertainty of $\mathbf{d}_c$, chosen to encode the sample vari-

ance of $\mathbf{d}_c$:

$$I(S_c^l) = \frac{1}{|S_c^l| - 1} \sum_{d \in S_c^l} \left(\mathbf{d} - \frac{1}{|S_c^l|} \sum_{d' \in S_c^l} \mathbf{d}' \right)^2 \tag{6.12}$$

Combining 6.2, 6.11 and 6.12 yields the following objective function:

$$I_{\mathbf{d}}(S) = \sum_{l=\{left,right\}} p^l \left(\sum_c p(c|S^l) \cdot \frac{1}{|S_c^l| - 1} \sum_{\mathbf{d} \in S_c^l} \left(\mathbf{d} - \frac{1}{|S_c^l|} \sum_{d' \in S_c^l} \mathbf{d}' \right)^2 \right) \tag{6.13}$$

For the class label uncertainty, we choose the average entropy of the class label distributions over the child nodes. Note that minimizing the average entropy is equivalent to maximizing information gain.

$$I_c(S) = -|S^{left}| \cdot \sum_c p(c|S^{left}) - |S^{right}| \cdot \sum_c p(c|S^{right}) \tag{6.14}$$

Every split node randomly selects 6.13 or 6.14 to optimize either for class label c or the continuous parameters $\mathbf{d}_c$.

6.3.3 Testing

Given a new unseen image I, all voxels of that image are passed through the trees of the forest. Each voxel traverses the tree until it reaches a leaf node l. The voxel then votes for the location of the plane $\mathbf{b} = (b_1, b_2, b_3)$ using distributions $p(c)$ and $p(\mathbf{d}|c_i)$ from the leaf. The parametrization of the plane $\mathbf{b}$ is expressed via the stored distance vectors $\mathbf{d}$ using equation 6.5 and Fig. 6.3 from section 6.2.3:

$$\mathbf{b} = \mathbf{d} \left(\frac{\mathbf{v} \cdot \mathbf{d}}{|\mathbf{d}|^2} + 1 \right)$$

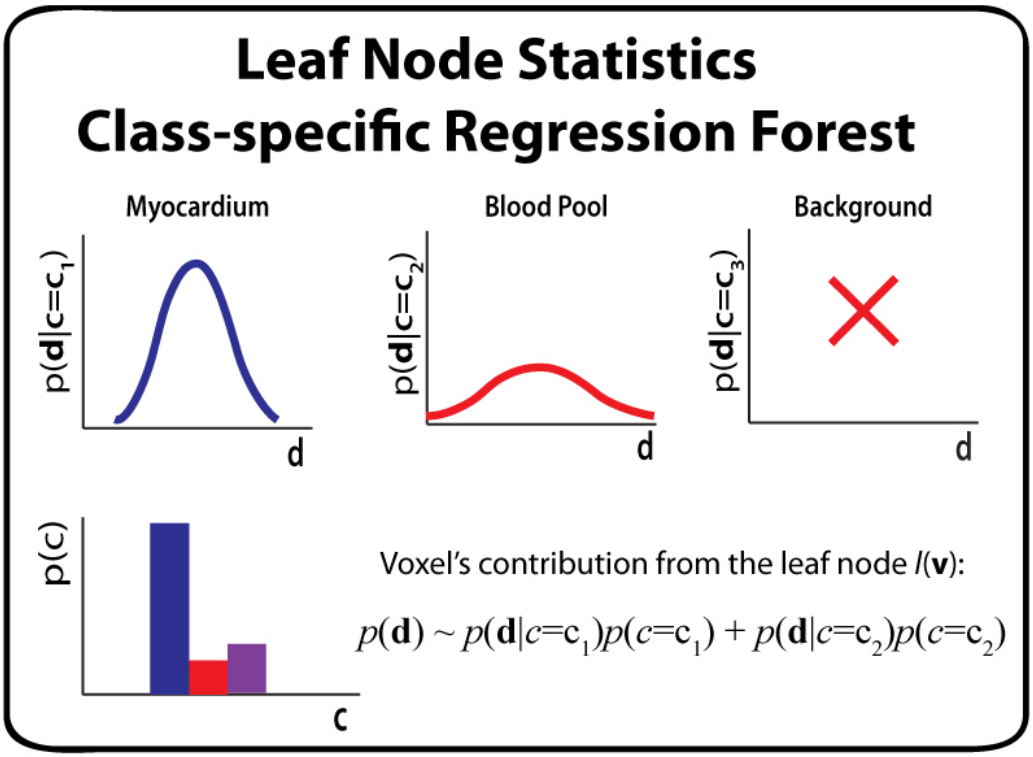


Figure 6.6: Each leaf node of the Class-Specific Regression Forest contains a discrete distribution of class labels $p(c)$ and one three-dimensional distribution

We are interested in the probability $p(\mathbf{b}|l(\mathbf{v}))$, which can be decomposed as follows:

$$p\left(\mathbf{b}|l\left(\mathbf{v}\right)\right) = \sum_{c} p\left(\mathbf{b}|c,l\left(\mathbf{v}\right)\right) p\left(c|l\left(\mathbf{v}\right)\right) = \tag{6.15}$$

$$\sum_{c} p\left(\mathbf{b} = \mathbf{d}_c\left(\frac{\mathbf{v}\cdot\mathbf{d}_c}{|\mathbf{d}_c|^2}+1\right)|c,l\left(\mathbf{v}\right)\right) p\left(c|l\left(\mathbf{v}\right)\right) \tag{6.16}$$

The distribution $p\left(c|l\left(\mathbf{v}\right)\right)$ is estimated as the proportion of voxels per class c at the leaf, while $p\left(\mathbf{b}|c,l\left(\mathbf{v}\right)\right)$ is estimated by a sum of Dirac measures:

$$p\left(\mathbf{b}|l\left(\mathbf{v}\right)\right) = \sum_{c} p(c|l(\mathbf{v}) \cdot \frac{1}{|S_c^{l(\mathbf{v})}|} \sum_{\mathbf{d}\in S_c^{l(\mathbf{v})}} \delta_d\left(\mathbf{d}\left(\frac{\mathbf{v}\cdot\mathbf{d}}{|\mathbf{d}|^2}+1\right)\right) \tag{6.17}$$

Having separate class-conditional distributions $p(\mathbf{d}|c)$ allows for exclusion of distributions with low confidence or distributions with low presence $p(c)$ from voting. We choose the sample variance of $\mathbf{d}_c$ as the uncertainty measure and set a threshold K on it to control the maximum uncertainty of the participating distributions $p(\mathbf{d}|c_i)$. An additional threshold

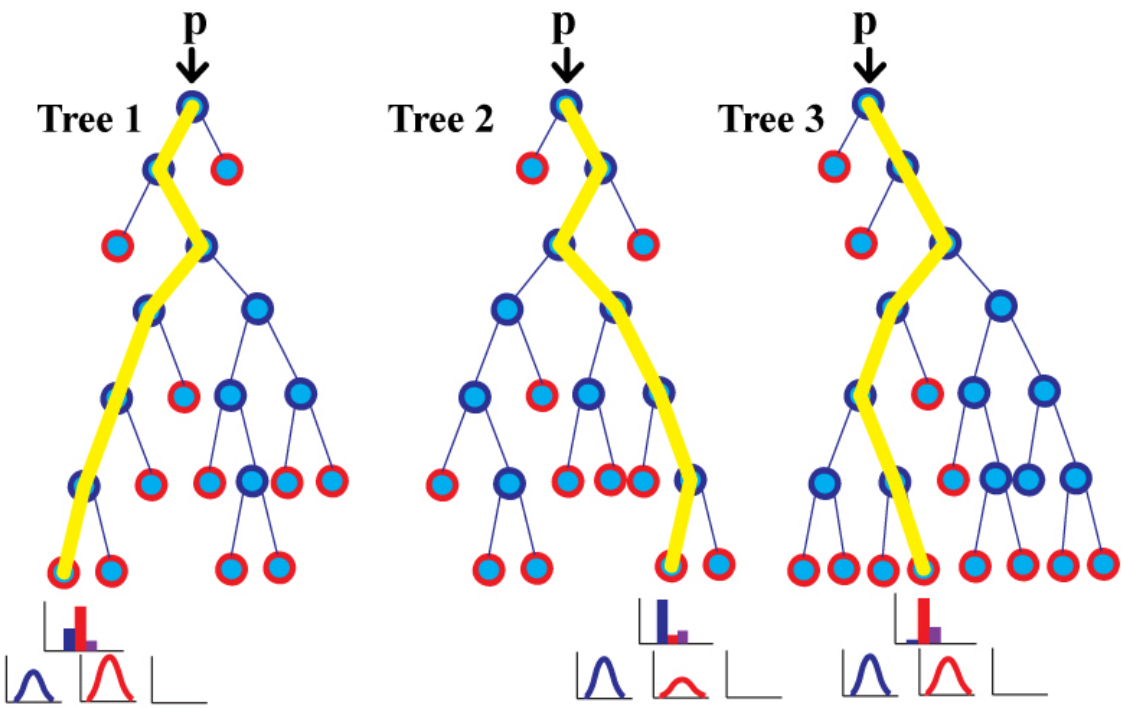


Figure 6.7: Testing stage of the Class-Specific Regression Forest. All voxels traverse the trees of the forest and the votes from the leaf nodes they reached are collected to produce a single posterior probability distribution $p(\mathbf{d}|Im)$, equation 6.15.

M on $p(c)$ is set to control the minimum presence of the class c at the leaf. The parameters K and M are found by cross-validation. An example in Fig. 6.6 explains a possible voting scenario from a single voxel $\mathbf{v}$.

Once one or more class-conditional distributions $p(\mathbf{d}|c)$ are excluded from voting, the sum in equation 6.15 ceases to be a probability measure since it does not integrate to one. However, we are only interested in the maximum of $p(\mathbf{b}|I)$ in equation 6.18, which can be obtained without a normalization factor.

The votes from all voxels and all trees are collected to produce a single posterior probability distribution:

$$p(\mathbf{b}|I) \propto \sum_{\mathbf{v}} \sum_{t=1}^{T} \sum_{\hat{c}} p(\hat{c}|l(\mathbf{v})) \cdot \frac{1}{|S^l_{\hat{c}}(\mathbf{v})|} \sum_{d \in S^{l(\mathbf{v})}_{\hat{c}}} \delta_d \left(\mathbf{d} \left(\frac{\mathbf{v} \cdot \mathbf{d}}{|\mathbf{d}|^2} + 1 \right) \right) \tag{6.18}$$

where $\hat{c}$ is the class label whose corresponding distribution $p(\mathbf{d}|\hat{c}, l)$ is included in the voting. The measure $p(\mathbf{b}|I)$ in equation 6.18 is further smoothed with a Gaussian kernel G

and the parameters of the plane are extracted using equation 6.10.

6.4 Features

Several standard low level local appearance features are deployed in this work, namely Rectanglular3D, Haar3D, and Difference3D features (86; 84; 83). The Position3D feature is also employed to capture the absolute position of the voxels in the image.

6.5 Experiments

The dataset used in this study comprised 25 end-diastolic 3D echocardiograms from healthy subjects. A Philips iE33 ultrasound system was used to acquire the volumes with dimension of (224x208x208) voxels and an average of $0.88mm^3$ spatial resolution. The myocardium and the blood pool of all volumes, along with the 2C, 3C and 4C standard views were manually segmented/annotated by an expert.

To investigate the performance of the proposed algorithms for standard views detection, three classic Regression Forests and three Class-Specific Regression Forests were trained to extract 2C, 3C and 4C view planes from 3D echocardiograms. The results of the two methods on each plane detection task are reported in section 6.6.

6.5.1 Model Parameters

During training, aside from choosing a feature f and a threshold τ for each split node, there are six parameters to be defined for each of the classic Regression Forests and seven parameters for each of the Class-Specific Regression Forests:

1. Number of trees in the forest T
2. Depth of the trees D
3. Number of candidate features to be tested at a split node m
4. Minimum number of datapoints at a split node N

5. A threshold $K_{classic_RF}$ on the sample variance of the output predictors $p(\mathbf{d}|\mathbf{v})$ for the classic Regression Forests, and a threshold $K_{class-specific_RF}$ on the sample variance of the class-conditional output predictors $p(\mathbf{d}|c_i,\mathbf{v})$ for the Class-Specific Regression Forests

6. A Gaussian kernel bandwidth σ^2 for the Parzen density estimator of 6.9 and 6.18

7. Thresholds M_i on the presence of the class c_i at a node, controlling whether the corresponding class-conditional distribution of the plane parameters $p(\mathbf{d}|c_i)$ will participate in the voting. This parameter is inherent only to the Class-Specific Regression Forest.

Two parameters were used as the stopping criteria for growing the trees in each forest:

1. The depth of the tree D

2. The minimum number of datapoints at a split node N.

In our experiments we used $N = 50$. As it was mentioned in chapter 5, adding extra trees to the forest may only improve the overall accuracy of the regression forest, and therefore, we used $T = 20$ trees for all experiments and showed that increasing the number of trees in the forest would not improve the overall accuracy of the method. As for the number of candidate features per split node, we used $m = 100$. By testing a small proportion of features m per node we injected randomness into the algorithm. This ensured that all the trees in the forest have different structure yielding a good generalisation. During training, the offset to calculate the local appearance features was selected within the $20x20x20$ voxels neighbourhood. For all these features, any side of the feature 'box' could not be greater than 20 voxels or less than 4 voxels.

Thus, there are three parameters to be found for the classic Regression Forest, and four parameter for the Class-Specific Regression Forest. These parameters were found using 5-fold cross-validation for each of the six experiments on the available 25 echocardiographic volumes. The data was randomly split into 5 non overlapping groups, 5 echocardiographic images in each group. One of the five groups was left out for validation and the remaining four groups were used for training, the procedure was repeated five times, selecting different folders for validation and the remaining folders for training. The optimal values of the parameters corresponded to the minimum average validation error and are reported in

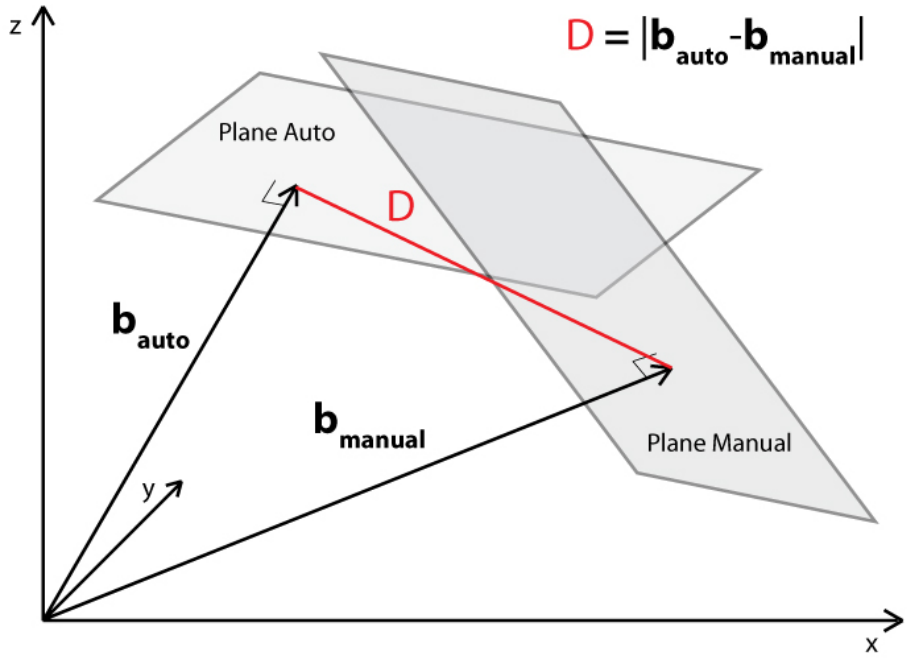


Figure 6.8: Distance D is used to compare the performance of the classic Regression Forest with the performance of the Class-Specific Regression Forest.

Table 5.1 in the results section 6.6.

6.5.2 Validation Metrics

To measure the similarity between two planes, two error metrics were utilized, namely, distance D and angle θ.

The distance between two planes is defined as follows:

$$D = |\mathbf{b}_{automatic} - \mathbf{b}_{manual}| \tag{6.19}$$

where $\mathbf{b}_{automatic}$ and $\mathbf{b}_{manual}$ are the distances from the origin to the automatically detected plane and the manually annotated plane, Fig. 6.8.

The angle between two planes is measured as the angle between the normals to the

planes:

$$\theta = cos^{-1} \frac{\mathbf{n}_a \mathbf{n}_m}{|\mathbf{n}_a||\mathbf{n}_m|} \tag{6.20}$$

The performance of the Class-Specific Regression Forest summarized in Table 6.1. The results were obtained with the maximum tree depth $D = 18$, number of trees $T = 20$, width of the Gaussian kernel for all cases was $\sigma^2 = 2mm$, $m = 100$ randomly chosen features where tested at each node during training. For all Class-Specific Regression Forests, the thresholds M_c for $p(c = myocardium)$ and $p(c = blood_pool)$ were greater than 0.5, leaving only one of the class-conditional distributions $p(\mathbf{d}|c_i)$ per node to participate in the voting.

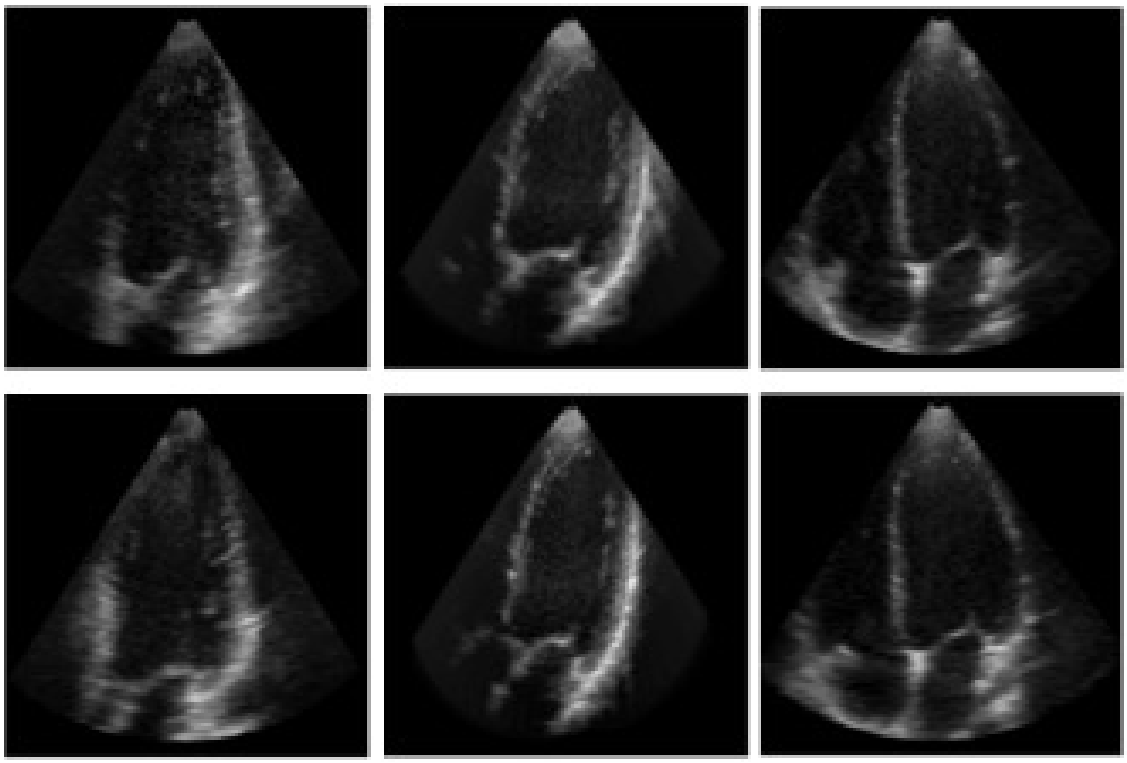
The regression forests were implemented in C++. Testing one volume requires approximately 10 seconds using Intel Xeon 2.8GHz computer with 12 cores and 48GB RAM running Win7. With a parallel tree implementation, training required about 9 hours (i.e. training time per tree) and only needed to be done once.

6.6 Results

We report the mean $\pm$ standard deviation of distance D and angle θ in Table 6.1.The table shows the comparison between the classic Regression Forest and the Class-Specific Regression Forest on 2C, 3C and 4C standard views detection. Visual examples of the detection results are shown in Fig. 6.9.

		2 chamber	3 chamber	4 chamber
Distance (mm)	Traditional RF	14.0 ± 10.7	15.0 ± 8.7	13.8 ± 9.5
	Class-Specific RF	5.5 ± 4.0	6.1 ± 4.0	5.2 ± 3.6
Angle (degrees)	Traditional RF	6.4 ± 5.7	8.3 ± 6.3	6.2 ± 5.4
	Class-Specific RF	3.2 ± 2.2	3.4 ± 3.1	3.2 ± 2.0

Table 6.1: Distance in millimetre and angle in degrees between the planes extracted by the classic and the Class-Specific Regression Forests for the 2C, 3C and 4C view.



(a) 2-chamber (b) 3-chamber (c) 4-chamber

Figure 6.9: Visual comparison between manual 2C, 3C and 4C views (top row) and the automatically detected 2C, 3C and 4C views (bottom row) using the Class-Specific Regression Forests.

6.7 Discussion

We have presented a new method to extract planes from 3D images by using continuous parameterization of the plane and learning a non-linear mapping to these parameters from the original input space of voxels. In the context of standard planes detection from 3D echocardiography, we proposed a Class-Specific Regression Forest, where additional structural information in the form of voxel class label was injected into the training step of the algorithm in order to collects votes only from semantically relevant parts of the image with respect to the plane and ignores the regions of the image which are irrelevant to it. Compared to the classic Regression Forest, a significant improvement has been observed. The boost in performance can be explained by the ability of the Class-Specific Regression Forest to localize the areas with meaningful to the problem information and ignore the extensive amount of irrelevant data from outside of the left and right ventricles. Future work includes incorporation of the right ventricle voxel classes into the algorithm and validation

of the algorithm on a larger dataset.

Chapter 7

Conclusions and Future Work

7.1 Summary

This doctoral thesis has explored new ways to model and analyse 3D echocardiography data using machine learning methods with a key focus on Decision Random Forests. The core contributions of this work are summarised below:

In Chapter 3, we proposed four methods for feature extraction and wall motion classification of 2D stress echocardiography. All four methods demonstrated superior accuracy compared to state-of-the-art. We first described the process of candidate features extraction, 30 features per patient - 15 features from the rest sequence and 15 features from the stress sequence, section 3.5. The four separate methods were further applied to select the best set of features out of the pool of 30 features to classify the cardiac wall motion. The first two methods were based on Support Vector Machines and Relevant Vector Machines, section 2.3, which were used to evaluate various sets of features. In section 3.7 we proposed a novel way to search the space of all possible feature subsets, and extracted the optimal set of features to maximise the accuracy of wall motion classification. The other two methods were so called embedded methods, where the feature selection procedure was encoded into the training phase, section 3.8. These methods were based on: (i) Lasso algorithm and Regularised Least Squares, and (ii) Elastic Net regularisation and Regularised Least Squares.

In Chapter 4, we described the application of the Classification Random Forest to segment the myocardium from 3D echocardiographic images of the left ventricle, section 4.2. We further demonstrated that more reliable and ultrasound specific descriptors had to be included in order to achieve desirable results.

Therefore, **in Chapter 5**, we proposed to improve the segmentation results achieved in Chapter 4 by choosing better descriptors for 3D echocardiographic images. We first described a procedure to reinforce the standard classic features such as Rectangular3D, Difference3D, Haar3D and DifferenceHaar3D features, by globally registering the echo volumes. We introduced a new modified Hough transform for circles in section 5.3.1 to detect the epicardial and endocardial centres in short-axis slices and extract the centre lines of the left ventricle, which were used to register the volumes, section 5.3.2. We also employed a new set of features, which were extracted from feature asymmetry images - FARectangular3D, FADifference3D, FAHaar3D and FADifferenceHaar3D features, section 5.4. The COFA measure was further introduced, section 5.5.3, to enable extraction of novel *LoCo* and GloCo features with local and global shape constraints on coupled endo- and epicardial boundaries of the left ventricle, section 5.5. The new features showed the boost in performance by approximately 15%.

In Chapter 6, we developed a novel regression voting algorithm to detect planes from 3D images based on the Regression Random Forest, with the application to extract the standard views from 3D echocardiography, section 6.2. We further proposed to incorporate voxel class label information into the training phase to reinforce the learning from semantically relevant to the problem classes, yielding the Class-Specific Regression Forests, section 6.3. The Class-Specific Regression Random Forest showed significant increase in performance compared to the classic Regression Random Forest, and produced results comparable to the manual annotations, section 6.6.

7.2 Future Work

This section highlights a number of interesting directions in which the work in this thesis could be extended.

7.2.1 Larger Datasets

The dataset of 3D echocardiographic images used in this thesis comprises 25 sequences of healthy subjects. Since all the experiments treated each voxel as a datapoint, the amount of data to be processed was very large. However, the statistical variability was still 25 subjects. Therefore, to produce more accurate and reliable results, a larger dataset needs to be deployed. Furthermore, since all the subjects from the dataset were healthy individuals, the methods proposed in this thesis might not perform as well on diseased subjects. At the same time the diseased hearts tend to have larger variability in shape and motions, and the data might have lower quality. Therefore, more dataset would be required to overcome these challenges.

7.2.2 GPU Implementation

Although different cores were used to execute different trees in this thesis (both at the training stage and testing stage), the true parallelism would involve GPU implementation of the random forest technique. Since the training has to be done once, the focus of the GPU implementation shall lie on the testing stage to enable real time myocardium segmentation and the standard views extraction from 3D echocardiography.

7.2.3 Wall Motion Classification of 3D Echocardiography and MRI Cardiac Data

The methods for wall motion classification described in Chapter 3 could be extended to assess the wall motion from 3D stress echocardiography. However, since the proposed methods for wall motion classification highly depend on the provided segmentation of the

myocardium, reliable automatic methods to segment heart muscle from 3D stress echocardiograms have to be developed. As mentioned earlier, the proposed methods are not modality dependant and could equally be applied to cardiac MRI. Due to superior spatial resolution of MRI data, the input error of myocardium segmentation would be much lower compared to the ultrasound case allowing for more accurate validation of the proposed methods.

7.2.4 Oblique Classification Random Forests for Segmentation of 3D Echocardiography

As stated in chapter 4, the conventionally used in images analysis features become poor descriptors when applied to ultrasound data. Therefore, during the training stage of the classic Classification Random Forest, there is not much information about the class label gained at each split. The classifier shall perform a large number of splits to get a reasonable confidence in its predictors. However, due to limited training data, the splitting normally terminates when the minimum number of data points is reached at a node and not due to low uncertainty in the class label predictors. More elaborate, non-linear functions, i.e. hypersurfaces, or linear functions of two or more features (hyperplanes, or oblique functions) can be used to separate the highly non-linear ultrasound data. Applying a non-linear split function at each node would require a lot of computational power, and in many cases would be impractical. For the oblique split functions, with every additional feature added as an argument, the space for search for the best set of features with the corresponding thresholds grows exponentially, and therefore, smart methods to search this space would be the core of this research.

7.2.5 Oblique Class-Specific Regression Random Forests with *LoCo* and *GloCo* features for the Standard Views Extraction

The proposed regression voting for the plane detection task from 3D images could be reinforced by (i) making changes into the training procedure of the regressor and the use of the oblique functions (or more complex non-linear functions) to split the ultrasound data at a node, (ii) registering the echo volumes prior to training, (iii) adding FA, LoCo and GloCo features into the learning stage of the regressor. Furthermore, additional class label information such as right ventricle blood pool and myocardium of the right ventricle may improve the accuracy of the Class-Specific Regression Forest, and therefore, could be investigated for inclusion.

Bibliography

[1] L. S. Lilly, *Braunwald's heart disease: a textbook of cardiovascular medicine*, vol. 1. Elsevier Health Sciences, 2012.

[2] S. B. Barnett, G. R. Ter Haar, M. C. Ziskin, H.-D. Rott, F. A. Duck, and K. Maeda, "International recommendations and guidelines for the safe use of diagnostic ultrasound in medicine," *Ultrasound in medicine & biology*, vol. 26, no. 3, pp. 355–366, 2000.

[3] T. Marwick, A.-M. D'hondt, T. Baudhuin, B. Willemart, W. Wijns, J.-M. Detry, and J. Melin, "Optimal use of dobutamine stress for the detection and evaluation of coronary artery disease: combination with echocardiography or scintigraphy, or both?," *Journal of the American College of Cardiology*, vol. 22, no. 1, pp. 159–167, 1993.

[4] Y. Matsumura, T. Hozumi, K. Arai, K. Sugioka, K. Ujino, Y. Takemoto, H. Yamagishi, M. Yoshiyama, and J. Yoshikawa, "Non-invasive assessment of myocardial ischaemia using new real-time three-dimensional dobutamine stress echocardiography: comparison with conventional two-dimensional methods," *European heart journal*, vol. 26, no. 16, pp. 1625–1632, 2005.

[5] W. F. Armstrong and W. A. Zoghbi, "Stress echocardiographycurrent methodology and clinical applications," *Journal of the American college of cardiology*, vol. 45, no. 11, pp. 1739–1747, 2005.

[6] W. F. Armstrong and T. Ryan, *Feigenbaum's echocardiography*. Lippincott Williams & Wilkins, 2012.

[7] C. Jenkins, J. Chan, L. Hanekom, and T. H. Marwick, "Accuracy and feasibility of online 3-dimensional echocardiography for measurement of left ventricular parameters," *Journal of the American Society of Echocardiography*, vol. 19, no. 9, pp. 1119–1128, 2006.

[8] C. Jenkins, S. Moir, J. Chan, D. Rakhit, B. Haluska, and T. H. Marwick, "Left ventricular volume measurement with echocardiography: a comparison of left ventricular opacification, three-dimensional echocardiography, or both with magnetic resonance imaging," *European heart journal*, vol. 30, no. 1, pp. 98–106, 2009.

[9] E. G. Caiani, C. Corsi, J. Zamorano, L. Sugeng, P. MacEneaney, L. Weinert, R. Battani, J. L. Gutierrez, R. Koch, L. Perez de Isla, *et al.*, "Improved semiautomated quantification of left ventricular volumes and ejection fraction using 3-dimensional echocardiography with a full matrix-array transducer: comparison with magnetic resonance imaging," *Journal of the American Society of Echocardiography*, vol. 18, no. 8, pp. 779–788, 2005.

[10] H. S. Yang, P. A. Pellikka, R. B. McCully, J. K. Oh, J. A. Kukuzke, B. K. Khandheria, and K. Chandrasekaran, "Role of biplane and biplane echocardiographically guided 3-dimensional echocardiography during dobutamine stress echocardiography," *Journal of the American Society of Echocardiography*, vol. 19, no. 9, pp. 1136–1143, 2006.

[11] R. M. Lang, V. Mor-Avi, L. Sugeng, P. S. Nieman, and D. J. Sahn, "Three-dimensional echocardiographythe benefits of the additional dimension," *Journal of the American College of Cardiology*, vol. 48, no. 10, pp. 2053–2069, 2006.

[12] M. J. Monaghan, "Role of real time 3d echocardiography in evaluating the left ventricle," *Heart*, vol. 92, no. 1, pp. 131–136, 2006.

[13] J. Hung, R. Lang, F. Flachskampf, S. K. Shernan, M. L. McCulloch, D. B. Adams, J. Thomas, M. Vannan, and T. Ryan, "3d echocardiography: a review of the current status and future directions," *Journal of the American Society of Echocardiography*, vol. 20, no. 3, pp. 213–233, 2007.

[14] L. P. Badano, E. Dall'Armellina, M. J. Monaghan, M. Pepi, M. Baldassi, M. Cinello, and P. M. Fioretti, "Real-time three-dimensional echocardiography: technological gadget or clinical tool?," *Journal of Cardiovascular Medicine*, vol. 8, no. 3, pp. 144–162, 2007.

[15] K. Chykeyuk, D. A. Clifton, and J. A. Noble, "Feature extraction and wall motion classification of 2d stress echocardiography with support vector machines.," in *SPIE Medical Imaging*, pp. 79630H–79630H, International Society for Optics and Photonics, 2011.

[16] K. Chykeyuk, D. A. Clifton, and J. A. Noble, "Feature extraction and wall motion classification of 2d stress echocardiography with relevance vector machines," in *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pp. 677–680, IEEE, 2011.

[17] K. Chykeyuk, M. Yaqub, and J. A. Noble, "Novel context rich loco and gloco features with local and global shape constraints for segmentation of 3d echocardiograms with random forests," in *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pp. 59–69, Springer, 2013.

[18] K. Chykeyuk, M. Yaqub, and J. A. Noble, "Class-specific regression random forest for accurate extraction of standard planes from 3d echocardiography," in *Medical Computer Vision. Large Data in Medical Imaging*, pp. 53–62, Springer, 2014.

[19] R. C. Houck, J. E. Cooke, and E. A. Gill, "Live 3d echocardiography: a replacement for traditional 2d echocardiography?," *American Journal of Roentgenology*, vol. 187, no. 4, pp. 1092–1106, 2006.

[20] B. J. Krenning, M. M. Voormolen, J. R. Roelandt, *et al.*, "Assessment of left ventricular function by three-dimensional echocardiography," *Cardiovasc Ultrasound*, vol. 1, no. 1, p. 12, 2003.

[21] A. Fenster, D. B. Downey, and H. N. Cardinal, "Three-dimensional ultrasound imaging," *Physics in medicine and biology*, vol. 46, no. 5, p. R67, 2001.

[22] L. Sugeng, L. Weinert, K. Thiele, and R. M. Lang, "Real-time three-dimensional echocardiography using a novel matrix array transducer," *Echocardiography*, vol. 20, no. 7, pp. 623–635, 2003.

[23] E. Picano, "Stress echocardiography: a historical perspective," *The American journal of medicine*, vol. 114, no. 2, pp. 126–130, 2003.

[24] P. Coupé, P. Hellier, C. Kervrann, and C. Barillot, "Nonlocal means-based speckle filtering for ultrasound images," *Image Processing, IEEE Transactions on*, vol. 18, no. 10, pp. 2221–2229, 2009.

[25] B. Farhang-Boroujeny, *Adaptive filters: theory and applications*. John Wiley & Sons, Inc., 1998.

[26] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 165–168, 1980.

[27] A. Lopes, R. Touzi, and E. Nezry, "Adaptive speckle filters and scene heterogeneity," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, no. 6, pp. 992–1000, 1990.

[28] M. Karaman, M. A. Kutay, and G. Bozdagi, "An adaptive speckle suppression filter for medical ultrasonic imaging," *Medical Imaging, IEEE Transactions on*, vol. 14, no. 2, pp. 283–292, 1995.

[29] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 7, pp. 629–639, 1990.

[30] Y. Yu and S. T. Acton, "Speckle reducing anisotropic diffusion," *Image Processing, IEEE Transactions on*, vol. 11, no. 11, pp. 1260–1270, 2002.

[31] Y. Yu, J. A. Molloy, and S. Acton, "Three-dimensional speckle reducing anisotropic diffusion," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, vol. 2, pp. 1987–1991, IEEE, 2003.

[32] Q. Sun, J. A. Hossack, J. Tang, and S. T. Acton, "Speckle reducing anisotropic diffusion for 3d ultrasound images," *Computerized Medical Imaging and Graphics*, vol. 28, no. 8, pp. 461–470, 2004.

[33] K. Krissian, C.-F. Westin, R. Kikinis, and K. G. Vosburgh, "Oriented speckle reducing anisotropic diffusion," *Image Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 1412–1424, 2007.

[34] D. L. Donoho, "De-noising by soft-thresholding," *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, 1995.

[35] R. R. Coifman and D. L. Donoho, *Translation-invariant de-noising*. Springer, 1995.

[36] L. Gagnon and A. Jouan, "Speckle filtering of sar images: a comparative study between complex-wavelet-based and standard filters," in *Optical Science, Engineering and Instrumentation'97*, pp. 80–91, International Society for Optics and Photonics, 1997.

[37] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. Wells Jr, "Noise reduction using an undecimated discrete wavelet transform," *Signal Processing Letters, IEEE*, vol. 3, no. 1, pp. 10–12, 1996.

[38] X. Zong, A. F. Laine, and E. A. Geiser, "Speckle reduction and contrast enhancement of echocardiograms via multiscale nonlinear processing," *Medical Imaging, IEEE Transactions on*, vol. 17, no. 4, pp. 532–540, 1998.

[39] A. Pizurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single-and multiband image denoising," *Image Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 654–665, 2006.

[40] A. Achim, A. Bezerianos, and P. Tsakalides, "Novel bayesian multiscale method for speckle removal in medical ultrasound images," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 8, pp. 772–783, 2001.

[41] S. Foucher, G. B. Bénié, and J.-M. Boucher, "Multiscale map filtering of sar images," *Image Processing, IEEE Transactions on*, vol. 10, no. 1, pp. 49–60, 2001.

[42] S. Gupta, R. Chauhan, and S. Sexana, "Wavelet-based statistical approach for speckle reduction in medical ultrasound images," *Medical and Biological Engineering and computing*, vol. 42, no. 2, pp. 189–192, 2004.

[43] M. I. H. Bhuiyan, M. O. Ahmad, and M. Swamy, "Wavelet-based despeckling of medical ultrasound images with the symmetric normal inverse gaussian prior," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, pp. I–721, IEEE, 2007.

[44] M. Elad, "On the origin of the bilateral filter and ways to improve it," *Image Processing, IEEE Transactions on*, vol. 11, no. 10, pp. 1141–1151, 2002.

[45] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 60–65, IEEE, 2005.

[46] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 4, pp. 425–441, 2008.

[47] M. Mulet-Parada and J. A. Noble, "2d+ t acoustic boundary detection in echocardiography," *Medical Image Analysis*, vol. 4, no. 1, pp. 21–30, 2000.

[48] P. Kovesi, "Symmetry and asymmetry from local phase," in *Tenth Australian joint conference on artificial intelligence*, vol. 190, Citeseer, 1997.

[49] M. C. Morrone and R. Owens, "Feature detection from local energy," *Pattern Recognition Letters*, vol. 6, no. 5, pp. 303–313, 1987.

[50] P. Kovesi, "Image features from phase congruency," *Videre: Journal of computer vision research*, vol. 1, no. 3, pp. 1–26, 1999.

[51] D. Boukerroui, J. A. Noble, and M. Brady, "On the choice of band-pass quadrature filters," *Journal of Mathematical Imaging and Vision*, vol. 21, no. 1-2, pp. 53–80, 2004.

[52] K. Rajpoot, V. Grau, and J. A. Noble, "Local-phase based 3d boundary detection using monogenic signal and its application to real-time 3-d echocardiography images," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pp. 783–786, IEEE, 2009.

[53] M. Felsberg and G. Sommer, "The monogenic signal," *Signal Processing, IEEE Transactions on*, vol. 49, no. 12, pp. 3136–3144, 2001.

[54] X. Ye, J. A. Noble, and D. Atkinson, "3-d freehand echocardiography for automatic left ventricle reconstruction and analysis based on multiple acoustic windows," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1051–1058, 2002.

[55] V. Grau and J. A. Noble, "Adaptive multiscale ultrasound compounding using phase information," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pp. 589–596, Springer, 2005.

[56] K. Rajpoot, J. A. Noble, V. Grau, C. Szmigielski, and H. Becher, "Multiview rt3d

echocardiography image fusion," in *Functional Imaging and Modeling of the Heart*, pp. 134–143, Springer, 2009.

[57] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[58] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[59] G. I. Sanchez-Ortiz, G. J. Wright, N. Clarke, J. Declerck, A. P. Banning, and J. A. Noble, "Automated 3-d echocardiography analysis compared with manual delineations and spect muga," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1069–1076, 2002.

[60] V. Zagrodsky, V. Walimbe, C. R. Castro-Pareja, J. X. Qin, J.-M. Song, and R. Shekhar, "Registration-assisted segmentation of real-time 3-d echocardiographic data using deformable models," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 9, pp. 1089–1099, 2005.

[61] K. Rajpoot, V. Grau, J. Alison Noble, H. Becher, and C. Szmigielski, "The evaluation of single-view and multi-view fusion 3d echocardiography using image-driven segmentation and tracking," *Medical image analysis*, vol. 15, no. 4, pp. 514–528, 2011.

[62] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 2, pp. 158–175, 1995.

[63] E. D. Angelini, S. Homma, G. Pearson, J. W. Holmes, and A. F. Laine, "Segmentation of real-time three-dimensional ultrasound for quantification of ventricular function: a clinical study on right and left ventricles," *Ultrasound in medicine & biology*, vol. 31, no. 9, pp. 1143–1158, 2005.

[64] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing,

and bayes/mdl for multiband image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 9, pp. 884–900, 1996.

[65] T. F. Chan and L. A. Vese, "Active contours without edges," *Image processing, IEEE transactions on*, vol. 10, no. 2, pp. 266–277, 2001.

[66] N. Lin, W. Yu, and J. S. Duncan, "Combinative multi-scale level set framework for echocardiographic image segmentation," *Medical Image Analysis*, vol. 7, no. 4, pp. 529–537, 2003.

[67] Y. Zhu, X. Papademetris, A. Sinusas, and J. S. Duncan, "Segmentation of myocardial volumes from real-time 3d echocardiography using an incompressibility constraint," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pp. 44–51, Springer, 2007.

[68] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[69] T. F. Cootes and C. J. Taylor, "Active shape modelssmart snakes," in *BMVC92*, pp. 266–275, Springer, 1992.

[70] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Training models of shape from sets of examples," in *BMVC92*, pp. 9–18, Springer, 1992.

[71] T. F. Cootes, G. J. Edwards, C. J. Taylor, *et al.*, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[72] J. G. Bosch, S. C. Mitchell, B. P. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 11, pp. 1374–1383, 2002.

[73] J. Hansegård, F. Orderud, and S. I. Rabben, "Real-time active shape models for segmentation of 3d cardiac ultrasound," in *Computer analysis of images and patterns*, pp. 157–164, Springer, 2007.

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

[75] K. E. Leung, M. van Stralen, G. van Burken, N. de Jong, and J. G. Bosch, "Automatic active appearance model segmentation of 3d echocardiograms," in *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pp. 320–323, IEEE, 2010.

[76] V. Lempitsky, M. Verhoek, J. A. Noble, and A. Blake, "Random forest classification for automatic delineation of myocardium in real-time 3d echocardiography," in *Functional Imaging and Modeling of the Heart*, pp. 447–456, Springer, 2009.

[77] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[78] M. Verhoek, M. Yaqub, J. McManigle, and J. A. Noble, "Learning optical flow propagation strategies using random forests for fast segmentation in dynamic 2d & 3d echocardiography," in *Machine Learning in Medical Imaging*, pp. 75–82, Springer, 2011.

[79] V. Vapnik, *The nature of statistical learning theory*. springer, 2000.

[80] C. M. Bishop *et al.*, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[81] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *The journal of machine learning research*, vol. 1, pp. 211–244, 2001.

[82] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," 2008.

[83] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[84] M. Yaqub, M. K. Javaid, C. Cooper, and J. A. Noble, "Improving the classification accuracy of the classic rf method by intelligent feature selection and weighted voting of trees with application to medical image segmentation," in *Machine Learning in Medical Imaging*, pp. 184–192, Springer, 2011.

[85] A. Montillo, J. Shotton, J. Winn, J. E. Iglesias, D. Metaxas, and A. Criminisi, "Entangled decision forests and their application for semantic segmentation of ct images," in *Information Processing in Medical Imaging*, pp. 184–196, Springer, 2011.

[86] E. Geremia, B. H. Menze, O. Clatz, E. Konukoglu, A. Criminisi, and N. Ayache, "Spatial decision forests for ms lesion segmentation in multi-channel mr images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pp. 111–118, Springer, 2010.

[87] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu, "Regression forests for efficient anatomy detection and localization in ct studies," in *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pp. 106–117, Springer, 2011.

[88] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 617–624, IEEE, 2011.

[89] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Decision Forests for Computer Vision and Medical Image Analysis*, pp. 143–157, Springer, 2013.

[90] C. Leistner, A. Saffari, J. Santner, and H. Bischof, "Semi-supervised random

forests," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 506–513, IEEE, 2009.

[91] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1393–1400, IEEE, 2009.

[92] B. H. Menze, B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht, "On oblique random forests," in *Machine Learning and Knowledge Discovery in Databases*, pp. 453–469, Springer, 2011.

[93] N. Payet and S. Todorovic, "^ 2–random forest random field," in *Advances in Neural Information Processing Systems*, pp. 1885–1893, 2010.

[94] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli, "Decision tree fields," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1668–1675, IEEE, 2011.

[95] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[96] E. Konukoglu, B. Glocker, D. Zikic, and A. Criminisi, "Neighbourhood approximation forests," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 75–82, Springer, 2012.

[97] P. Yin, A. Criminisi, J. Winn, and I. Essa, "Tree-based classifiers for bilayer video segmentation," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[98] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[99] A. Statnikov and C. F. Aliferis, "Are random forests better than support vector machines for microarray-based cancer classification?," in *AMIA annual symposium proceedings*, vol. 2007, p. 686, American Medical Informatics Association, 2007.

[100] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, ACM, 2006.

[101] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 415–422, IEEE, 2011.

[102] A. Criminisi, J. Shotton, and S. Bucciarelli, "Decision forests with long-range spatial context for organ localization in ct volumes," in *MICCAI Workshop on Probabilistic Models for Medical Image Analysis*, 2009.

[103] L. B. J. F. R. Olshen and C. J. Stone, "Classification and regression trees," *Wadsworth International Group*, 1984.

[104] J. R. Quinlan, *C4. 5: programs for machine learning*, vol. 1. Morgan kaufmann, 1993.

[105] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends® in Computer Graphics and Vision*, vol. 7, no. 2–3, pp. 81–227, 2012.

[106] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[107] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[108] P. Clarysse, M. Han, P. Croisille, and I. E. Magnin, "Exploratory analysis of the spatio-temporal deformation of the myocardium during systole from tagged mri," *Biomedical Engineering, IEEE Transactions on*, vol. 49, no. 11, pp. 1328–1339, 2002.

[109] A. Suinesiaputra, A. F. Frangi, H. J. Lamb, J. H. Reiber, and B. P. Lelieveldt, "Automatic prediction of myocardial contractility improvement in stress mri using shape morphometrics with independent component analysis," in *Information Processing in Medical Imaging*, pp. 321–332, Springer, 2005.

[110] K. E. Leung and J. G. Bosch, "Localized shape variations for classifying wall motion in echocardiograms," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pp. 52–59, Springer, 2007.

[111] S. Mansor, N. P. Hughes, and J. A. Noble, "Wall motion classification of stress echocardiography based on combined rest-and-stress data," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pp. 139–146, Springer, 2008.

[112] S. Mansor and J. A. Noble, "Local wall motion classification of stress echocardiography using a hidden markov model approach," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pp. 1295–1298, IEEE, 2008.

[113] R. M. Lang, M. Bierig, R. B. Devereux, F. A. Flachskampf, E. Foster, P. A. Pellikka, M. H. Picard, M. J. Roman, J. Seward, J. Shanewise, *et al.*, "Recommendations for chamber quantification," *European Heart Journal-Cardiovascular Imaging*, vol. 7, no. 2, pp. 79–108, 2006.

[114] A. N. Tikhonov, V. I. Arsenin, and F. John, *Solutions of ill-posed problems*. Winston Washington, DC, 1977.

[115] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection

using wavelet templates," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 193–199, IEEE, 1997.

[116] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu, "Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 9, pp. 1342–1355, 2008.

[117] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[118] M. G. Roberts, T. F. Cootes, and J. E. Adams, "Automatic location of vertebrae on dxa images using random forest regression," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 361–368, Springer, 2012.

[119] O. Pauly, B. Glocker, A. Criminisi, D. Mateus, A. M. Möller, S. Nekolla, and N. Navab, "Fast multiple organ detection and localization in whole-body mr dixon sequences," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pp. 239–247, Springer, 2011.

[120] B. Glocker, O. Pauly, E. Konukoglu, and A. Criminisi, "Joint classification-regression forests for spatially structured multi-object segmentation," in *Computer Vision–ECCV 2012*, pp. 870–881, Springer, 2012.