Automatic localisation and segmentation of the Left Ventricle in Cardiac Ultrasound Images

Presented by:
Esther PUYOL
IG 3A F4B and MR 2A SISEA
2013/2014

Supervisor:
Paolo PIRO

Academic supervisor:
Guy CAZUGUEL

Company:
MEDISYS - PHILIPS RESEARCH PARIS

University:
TELECOM BRETAGNE

17th March - 12th September 2014
Abstract

Echocardiography is a common non-invasive diagnostic image modality that uses ultrasound to capture the structure and the function of the heart. During the last years there has been a growing need to automate the process of cardiac ultrasound images, involves many tasks, which such as image view classification, wall motion analysis, automatic placement of the Doppler gate over the valves, etc. Specifically, the delineation of the left ventricle of the heart in ultrasound data is an important tool to produce a quantitative assessment of the health of the heart. In this study, we propose a processing chain for the localisation and segmentation of the left ventricle in 2D-echocardiography in apical 2 chamber, 3 chamber and 4 chamber views. The system is built based on a machine learning approach that extracts knowledge from an annotated database. To reduce the complexity of the problem it has been divided into two parts, a pose estimation of the left ventricle and a non-rigid segmentation of the contour. The pose estimation problem is presented as a binary classification problem based on Boosting algorithms with Haar-like features. The main idea of Boosting is to combine the output of several weak classifiers to produce a powerful decision making committee. The non-rigid segmentation is based on a cascade regression framework, where every regressor learns the relationship between the local neighbourhood and the displacement from the true feature location in order to progressively refine the shape initialization given by the rigid detection.

Résumé

L’échographie cardiaque est une technique d’imagerie médicale non invasive qui utilise des ultrasons pour étudier la structure et le fonctionnement du cœur. Lors des dernières années il y a eu un besoin croissant d’automatisation dans le traitement des images cardiaques, ce qui implique de nombreuses tâches telles que la classification du type de vue, l’analyse du mouvement de la paroi, le positionnement automatique de la porte Doppler sur les valves, etc. En particulier, a segmentation du contour du ventricule gauche dans les images échographiques est un outil très important afin d’obtenir une quantification reproductible de la santé du cœur. Dans cette étude, nous proposons une chaîne de traitement pour la segmentation automatique du ventricule gauche dans les images échocardiographiques 2D faisant appel à des méthodes d’apprentissage supervisé. Pour réduire la complexité de la problématique, le problème a été divisé en deux parties : une première partie consacrée à la détection et à la localisation du ventricule gauche et une deuxième partie dédiée à la segmentation non rigide du contour de l’épicarde ou de l’endocarde. Le problème de localisation est défini comme un problème de classification binaire dans le cadre des approches de type Boosting, et abordé en faisant appel à l’algorithme AdaBoost avec des descripteurs de Haar. L’idée principale du Boosting est de combiner la sortie de nombreux classifiants faibles pour produire un classifieur plus puissant. La segmentation non rigide est basée sur une chaîne de régresseurs en cascade, où chaque régresseur apprend la relation entre le voisinage local et le déplacement de la position réelle de la fonction pour affiner progressivement la forme initiale donnée par le détecteur rigide.
Acknowledgements

I would like to express my deep gratitude to my advisor, Paolo Piro, for his support, continuous guidance, meticulous suggestions, immense knowledge and useful remarks through my learning process in this master’s thesis.

I am also thankful to Nicolas Villain and Vincent Auvray, for giving me the opportunity to do my internship at Philips.

Special thanks are given to Medisys team for their encouragement, insightful comments, and questions during the presentations, and for the great working environment in the laboratory. My thanks to Nathalie Bailly for her flawless organization and administration of this internship.

I would also like to express my special appreciation and thanks to my co-advisor, Professor Guy Cazuguel, and to my master’s director, Valerie Burdin. You have been my mentors since I began at Telecom Bretagne.

I am also grateful to the current teachers at Telecom Bretagne, for their various forms of support during my graduate study.

Finally, my greatest regards to the others interns and the PhD students, with whom I had the pleasure to spend 6 months, having great moments during the breaks.
# Contents

Abstract ii

Acknowledgements iii

1 Introduction 1

1.1 Clinical Context 2

1.1.1 Hearth anatomy 2

1.1.2 Transthoracic Echocardiogram (TEE) 4

1.1.3 Challenges 6

1.2 Literature Review and Theoretical Background 8

1.2.1 Overview of Pose estimation 8

1.2.2 Overview of Segmentation methods 9

2 Methodology 12

2.1 LV Pose estimation 13

2.1.1 Theoretical approach 13

2.1.2 Proposed approach 15

2.2 Supervised LV Segmentation 24

2.2.1 General approach 24

2.2.2 Proposed approach 25

3 Results 29

3.1 Database and settings 29

3.1.1 Settings for pose estimation 31

3.1.2 Settings for Segmentation 32

3.2 LV bounding box detection 34

3.2.1 AdaBoost 34

3.2.2 Boosted cascade 38

3.2.3 Comparison 41

3.3 LV Segmentation 42

4 Conclusion and Future Work 44

4.1 Thesis summary 44

4.2 Future Work 45

4.2.1 View classification problem 45

4.2.2 Rotation-invariance for the boosted cascade with a Pyramidal detector 45

4.2.3 Multi-scale classifier 46

4.2.4 Multi-scale Gabor features 46

4.2.5 Multivariable regression 46

4.3 Personal conclusion 46

Bibliography 48
List of Figures

1.1 Heart anatomy \(^1\) ......................................................... 3
1.2 Parasternal view \([25]\) ....................................................... 4
1.3 Subcostal and Subprastenal view \([25]\). ............................... 5
1.4 Apical view \([25]\). .......................................................... 6
1.5 Missing boundaries in ultrasound images on the Philips database. .... 7
1.6 Segmentation of the epicardium (green) and the endocardium (pink) in three images from the same patient on the Philips database. .... 7

2.1 Schema of the proposed approach for left-ventricle localisation. .... 16
2.2 Haar-like features \([17]\). ....................................................... 18
2.3 Schematic description of Boosted Cascade \([17]\). A window has to pass all the classifiers in the cascade to be considered as a positive (T). If a window is failed by any classifier, it is regarded as a negative (F). 20
2.4 Gabor feature construction. ................................................. 27
2.5 Module of convolution image with Gabor filter for multiple \(\sigma\) with \(\theta=-45^\circ\) on the Philips database. ................................. 27

3.1 Segmentation of the epicardium (yellow) and the endocardium (green) on the Philips database. ........................................... 30
3.2 Histogram of LV pose parameters \(c_x, c_y, s_x, s_y, \theta\) and their fitting with a Gaussian. .................................................. 30
3.3 Distribution map of the bounding box. ................................... 31
3.4 Contour points tracked along Echocardiogram sequence with \([33]\). .... 32
3.5 Variance PCA for shape and Gabor feature vector. ....................... 33
3.6 Jaccard index (\(J\)), Dice index (\(D\)) and number of points inside the bounding box (nbPoints) for estimate bounding box (yellow) and ground truh (pink). 35
3.7 Precision and recall curve versus number of iterations. ................. 36
3.8 Examples of LV localisation with AdaBoost. On the left, the image with ground truth (pink) and the proposed bounding box (yellow), on the right the corresponding probabilistic map. The first pair of images is a 2 chamber view, the second a 3 chamber view while the last is a 4 chamber view. .... 37
3.9 A2C localisation. On the left, the image with ground truth (red) and the proposed bounding box (green), on the right the corresponding probabilistic map. On the top Boosted Cascade with four pose estimation and on the bottom with five pose estimation. ......................... 40
3.10 A3C localisation. On the left, the image with ground truth (red) and the proposed bounding box (green), on the right the corresponding probabilistic map. On the top Boosted Cascade with four pose estimation and on the bottom with five pose estimation. ......................... 40
3.11 A4C localisation. On the left, the image in the first stage with ground truth (red) and the proposed bounding box (green), on the right the same image on the last stage. On the top Boosted Cascade with four pose estimation and on the bottom with five pose estimation. .................................................. 41

3.12 Segmentation example, in green the ground truth, in yellow the mean shape and in pink the the predicted shape. ................................................................. 43

4.1 Cakebdar for the internship ................................................................. 47

List of Tables

3.1 Jaccard and Dice index in AdaBoost and number of points of the contour inside the estimate bounding box. ................................................................. 34
3.2 Jaccard and Dice index in AdaBoost and number of points of the contour inside the estimate bounding box in Figure 3.8 ................................. 37
3.3 Jaccard and Dice index in Boosted cascade with 4 parameters estimation $(c_x, c_y, s_x, s_y)$ ................................................................. 38
3.4 Jaccard and Dice index in Cascade Boosted with 5 parameters estimation $(c_x, c_y, s_x, s_y, \theta)$ ................................................................. 39
3.5 Parameters for Cascade Boosted ................................................................. 39
3.6 Hausdorff distance (HD) and mean Euclidean distance (MED) in training and in test ................................................................. 42
Chapter 1

Introduction

My final internship is carried out at Philips France, an international electronics company which is headquartered in Suresnes near Paris. Philips was founded in Eindhoven, Netherlands, in 1891 by brothers Gerard and Anton Philips, and their father Frederik Philips. In 1914 they created their first research laboratory to develop x-ray and radio technology. Over the years, the list of Philips’s inventions has grown to include many breakthroughs that continue to enrich people’s everyday lives in the areas of healthcare, consumer lifestyle and lighting.

My internship take part of the Philips Healthcare group which is focused on delivering the most technologically advanced products and solutions for disease prevention and detection, clinical diagnosis, and vital function monitoring. Philips Healthcare aims at simplified medical care, by focusing on the needs of patients and healthcare professionals to streamline care throughout the process. The Healthcare industry is organized around four strategic business units: Imaging Systems, Patient Care and Clinical Informatics, Home Healthcare Solutions, and Healthcare Transformation Services.

Philips Healthcare France has a research team called Medisys which is comprised of thirty researchers and few PhD students dedicated to research in medical image processing in all modalities: Magnetic Resonance Imaging, X-Ray Computed Tomography, Fluoroscopy and Ultrasound. Their focus areas are:

- Diagnostic imaging
- Image-guided intervention and therapy
- Patient care
- Clinical decision support
- Home and personal healthcare
- Healing environments and services

These areas are established for cardiology, oncology, cardio-pulmonary, and women’s health. More specifically, my internship is associated with the cardiology team that has a project called Cardus which has the aim of developing an application to characterize and predict pathologies.
Within Cardus project, my internship is focused on the left ventricle localisation and segmentation problems of 2D Echocardiography. The main task is the localisation and detection of the left ventricle and its presentation in three standard views in order to assist cardiologists in the detection of cardiac diseases. The goal is to use this knowledge as an initial step for the segmentation of the endocardium and epicardium in a cardiac sequence. There are several methods to solve these tasks and some of them will be presented later in this introduction. First, an overview of the clinical context and main challenges are given in general, as well as a bibliography study. Following this introductory chapter on challenges and objectives of this thesis, Chapter 2 presents a framework that enables the localisation and segmentation of the left ventricle. Chapter 3 will be dedicated to present the validation and some results of the previous implementation. In conclusion, Chapter 4 will summarize this thesis’ findings and suggest ideas for future research.

1.1 Clinical Context

Cardiovascular disease is one of the leading causes of mortality worldwide, according to the World Health Organization. For this reason, the evaluation of heart functioning is an important research area in the medical field. Nowadays, multiple techniques exist to reduce cardiovascular risks. One of the most useful is the technique of Echocardiography which is routinely used in the diagnosis, management, and follow-up care of patients with any suspected or known heart diseases. It can provide a wealth of helpful information, including the size and shape of the heart, pumping capacity, and the location and extent of any tissue damage. In addition, it is one of the least invasive methods used in diagnostic tests in cardiology. An Echocardiogram can also give physicians other estimates of heart functioning, such as a calculation of cardiac output, ejection fraction, and diastolic functioning. More specially, the transthoracic echocardiogram (TEE) is a painless test similar to an X-ray but without the radiation, employing instead the same technology used to evaluate a baby’s health before birth. A hand-held device called a transducer is placed on the chest and transmits high frequency sound waves (ultrasound). These sound waves bounce off the heart structures, producing images and sounds that can be used by the doctor to detect heart damage and disease. During an Echocardiographic exam the transducer captures different anatomical sections of the heart from different viewpoints. Diagnostic analysis of these images requires manual intervention in both imaging and interpretation.

1.1.1 Hearth anatomy

The human heart is enclosed in a double-walled sac called the pericardium. The outer wall of the human heart is composed of three layers; the outer layer is called the epicardium, which is the inner wall of the pericardium. The middle layer is called the myocardium and is composed of contractile cardiac muscle. The inner layer is called the endocardium and is in contact with the blood that the heart pumps (see Figure 1.1).
Heart has four chambers, two superior atria and two inferior ventricles (see Figure 1.1). The atria are the receiving chambers and the ventricles are the discharging chambers. A wall of muscle called the septum separates the left and right atria and the left and right ventricles.

Figure 1.1: Heart anatomy.

1.1.2 Transthoracic Echocardiogram (TEE)

The transthoracic echocardiogram has been largely standardized across institutions. Those images are obtained from 4 separate standard transducer positions which allow for different portions of the heart to be visualized in detail. The first standard positions are the parasternal position, which has both long axis and short axis views, which placed the transducer approximately on the 3rd intercostal space on the left of the parasternal.

- **Parasternal long-axis view (PLAX):** The right (RV) and the left ventricle (LV), as well as the aortic bulb (Ao) and the left atrium (LA) can be displayed. (see Figure 1.2b)

- **Parasternal short-axis view (PSAX):** The transducer is turn 90 degrees clockwise from PLAX position, this rotation allow displaying the apex of the left (LV) and right (RV) ventricles. (see Figure 1.2a)

![Parasternal short-axis view (PSAX).](image1)

![Parasternal long-axis view (PLAX).](image2)

(a) Parasternal short-axis view (PSAX).
(b) Parasternal long-axis view (PLAX).

Figure 1.2: Parasternal view [25]

The second standard position is the subcostal where the patient is placed flat on their back. The structures seen are roughly similar to the apical four chambers, however, now the ultrasound beam is more perpendicular to the atrial and ventricular septum (See Figure 1.3a)

The third standard view is the suprasternal view. The patient lies completely flat without anything under their head. Then the transducer is placed in the suprasternal notch with the probe pointed at the left shoulder at about 2 o’clock. This view usually gives a nice look at the arch, from which the 3 major vessels may be seen to arise. The right pulmonary artery and brachiocephalic vein which drains into the Superior Vena Cava can also be seen. (See Figure 1.3b)
The last standard view is the apical position, which is obtained by placing the probe at the point of maximal impulse. The patient is positioned on their left side, but not as far as is the case when using the parasternal window. The apical window is usually found in the fifth intercostal space. There are numerous views that can be obtained from this position: including the 4 chamber view and the 5 chamber view, which includes the LV outflow tract, the 2 chamber view and the 3 chamber view all by rotation and minor manipulation of the probe. The apical windows are the most useful for assessing flow across the aortic, tricuspid and mitral valves with Doppler and this information can be used to assess for degree of stenosis and regurgitation, assessing diastolic function and evaluating LV segmental wall motion and LV thrombus.

- **Apical 2-chamber view (A2C):** In this view only the LV and the left atrium are visible. This view is excellent for assessing wall motion of the inferior wall, anterior wall and apex and mitral valve. (see Figure 1.4a)

- **Apical 3 chamber view or apical long-axis (A3C):** The structures seen are the LV, the left atrium and the RV outflow track. This view is similar to the parasternal long axis except now the LV apex is well visualized and is in the near field, and since there is complete visualization of the left atrium, its contents, area, and size are can be completely evaluated. (see Figure 1.4b)

- **Apical 4-chamber view (A4C):** The four-chamber view is probably the most important view in echocardiography. In this view is possible to see the LV, the RV and the two atria. The septum, which separates the cavities of the heart, is placed in the middle of the image. The interventricular septum tends to contract toward the left ventricular cavity during systole because of the higher pressure requirements of that chamber. This view is used to assess RV size and function, atrial size, abnormal intra-atrial and interventricular septal movement, as well as diastolic function. (see Figure 1.4c)
Nowadays, there is a growing need to automate the process of cardiac ultrasound image analysis that involves many tasks such as cardiac left ventricle delineation, view classification, wall motion analysis, automatic placement of Doppler fate overt the valves, etc. Among all these tasks, the delineation of the left ventricle is the first step to achieve automation of the other tasks, and also it is an important tool to produce a quantitative assessment of the health of the hearth. However, the localisation and segmentation of left ventricle in ultrasound images has some intrinsic problems, some of them due to the poor quality of the ultrasound images, other to the heart structure. Next section describes the challenges of ultrasound images that will condition the choice of the model.

1.1.3 Challenges

Ultrasonography is a very appealing modality for imaging both soft tissue and bone structures. Among the advantages of ultrasonography there are low cost, real time imaging, no side effects, no invasion, and ease of use. Despite the advantages of diagnostic ultrasound, there are also two fundamental limitations: trade-off between resolution and attenuation, and the presence of speckle noise and artefacts. Speckle is a random, deterministic, interference pattern in an image formed with coherent radiation of a medium containing many sub-resolution scatterers. The texture of the observed speckle pattern is not related to the underlying structure. The local brightness of the speckle pattern, however, does reflect the local echogenicity of the underlying scatterers. This will be a problem that I ought to take into account when choosing which features to include during the classification and segmentation process.

Another challenge is that cardiac ultrasound imaging can contain missing parts in the edge of the cardiac wall, missing regions from areas of the heart cavity where the wall is not present, or hidden by speckle noise (see Figure 1.5). Normally these regions are situated near the septum, but they can be in many other locations. These areas can cause a problem for the segmentation process because it will be difficult to distinguish them since they won’t show any difference of intensity.
Another issue, which is mostly related with the annotated database I uses, is the non-accurate delineation of the Epicardium and the Endocardium. The epicardial border is the delineation between the pericardium and the myocardium, which in many case leads to high heterogeneities in the annotation. In most of the cases, the delineation along the epicardium membrane is done in the right side of the gradient of the myocardium. However there are some annotations at the left. For example, in Figure 1.6 which shows three exams for the same patient, in the first image the annotation of epicardium contour at right is placed after the bright zone surrounding the myocardium, however, for the second image it is located in the center and in the third case it is again in the right.

![Figure 1.5: Missing boundaries in ultrasound images on the Philips database.](image)

![Figure 1.6: Segmentation of the epicardium (green) and the endocardium (pink) in three images from the same patient on the Philips database.](image)

In the literature exist several techniques to achieve our goals: object localization and segmentation but no all the methods are compatible with ultrasound challenges that are describes above, next section draw a short review of different approaches that are commonly used for pose estimation and segmentation taking in account our challenges.
1.2 Literature Review and Theoretical Background

This section will be divided into two parts, the first one will describe the most relevant articles in the state of the art of view object detection problems and the second part will be devoted to the segmentation process of the epicardium and endocardium.

1.2.1 Overview of Pose estimation

Object detection is the process of finding instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Object localisation is a more subtle problem; it consists in identifying the smallest rectangular region of the image that contains the object of interest. And finally segmentation is the process of contour delineation of the object. Typically these methods use extracted features and learning algorithms to recognize instances of an object category.

Detecting anatomical structures is important for medical image understanding, specifically, cardiac structure detection and localisation has received a lot of attention in image processing, especially in echocardiography since it is an important diagnostic aid in cardiology for the morphological and functional assessment of the heart. Thus, extracting viewpoint information in cardiac echo videos is an important step for echocardiogram interpretation, a great deal of attention has been driven towards the automatic classification problems.

In 2D-echocardiography, different techniques of Machine Learning have already proved their efficiency in this application, the two more used are Support Vector Machine ([8], [21], [37]) and Boosting ([40], [30], [32], [41]).

The method of Ebadolhahi et al. [8] used a generic cardiac chamber template to locate heart chambers, and they represented the spatial arrangement of the chambers using Markov Random Field (MRF)- based regional graphs. The collection of the energies obtained from comparing a test image to the model of the different views is used as the input to multi-class SVM classifier. This was one of the first methods proposed in the cardiac view recognition field, achieving accuracies of up to 88.35% if they consider clinical similarities, frames that even human experts could not discriminate. Nevertheless when taking into account missed false chambers, the accuracy drops to 34%. Since the chamber detection process is a weak process for primary identification, it is sensitive to ultrasound noise where small variations can lead to large errors in the graph formation.

In 2006, Zhou et al. [40] present a new method based on boosting algorithms, where they trained a boosted cascade of two multiclass classifiers to differentiate between A2C and A4C views achieving an accuracy of up to 90%. The method only deals with a two class problem. However, in the same year, Otey et al. [30] proposed a two-level hierarchical classification approach combined with a single dimension reduction approach. At the top level, it classifies an input sequence into apical class or parasternal class, and then it further classifies the sequences into one of the four final views at the second level (A2C, A4C, PLAX, PSAX). Their approach obtains 92% classification accuracy in testing.
Lately, Park et al. [32] proposed a system which first of all trains for each view type an LV detector based on MLBoost, then the LV candidates are used to construct multi-class view classifiers. They perform the final classification by combining the multiple classification results from the view classifiers. They achieve classification accuracy over 96% in both training and test data sets, but their method cannot be used to detect views in which the LV region is absent, such as the suprasternal view.

Zhou et al. [41] argued that scanning the whole image for detecting anatomical structures in medical images is unnecessary since the problem domain offers strong contextual information for localizing the targets. They propose an image-based boosting ridge regression algorithm (IBRR), where the regression-based detection algorithm replaces the exhaustive search by a sparse scanning, achieving improved accuracy with significantly less computational requirements.

Afterwards, Kumar et al. [21] used scale-invariant video features based on optical flow and the image edge maps to train multi-class SVMs, obtaining an average recognition rate of 81%. They also presented comparison with the classification method proposed in [32] having recognition rate of 74.9%. However they only differentiate between the A2C and A4C views, and the approach requires a pre-processing phase in which the left ventricle is identified by a human.

Finally, the most recent work on echo view recognition is that of Wu et al. [37], which does not require external grating to select a specific image, nor image-level annotations and classification of specific heart structures. They use GIST feature transform [28], which provides a compact, robust and discriminating model of a scene as input for SVM classifier. Their method achieves 94.85% accuracy using a single frame, and 98.51% using 4 consecutive frames. GIST feature transform consists of generating spectrograms from a grid of non-overlapping image blocks. In every block several oriented Gabor filters of different scales are applied to model structures at varying scales. Features obtained in every block are concatenated to produce the final vector representation. In brief, GIST feature transform computes the spectral energy of the image producing a global description, using a single feature vector.

1.2.2 Overview of Segmentation methods

The automatic segmentation of the left ventricle of the heart in ultrasound images has been a core research topic in medical image analysis. Most of the solutions are based on low-level segmentation methods (top-down segmentation), which use an a-priori model of the appearance of the left ventricle, such as level set techniques (Cootes et al. [4]), snakes and active contours (Kass et al. [18]), active shape and appearance models (Paragios et al. [31]). These methods assume that the myocardium is displayed brighter, and the blood pool in the LV is displayed darker than other structures in the image, and the main problem with this approach is that the violation of these assumptions may lead to incorrect segmentation.
Recently, pattern recognition methods have become more robust to imaging conditions by automatically building an appearance model from training images. These methods rely on a non-rigid top-down visual segmentation based on machine learning approach which has been traditionally divided into two procedures that are run in the following sequences: first, rigid detection and second non-rigid segmentation. The fundamental reason for having this first step is the reduction of complexity of the training and search mechanisms.

Some of the pattern recognition methods that have been proved to be efficient are Deep belief networks ([27], [26],[3]), Boosted Regression ([19],[2], [39],[5]) and structural SVM [38].

In active shape models any set of feature detectors can be used, the original algorithm used eigenmodels profiles of the texture for each of the individual feature points. Cristinacce et al. [5] propose to use boosted feature detector instead of eigen model profiles, the aim is to learn the relationship between the displacement of the true feature localisation and the texture appearance of the local neighbourhood around each point. To this end they use the GentleBoost logistic regression method described by Friedman et al. [11]. GentleBoost algorithm works as a boosted classifier. However the outputs are local displacement values in the training image frame, instead of class labels.

In the same year, Zhou et al. [39] proposed a shape regression machine based on the same idea of [5] for left ventricle segmentation. However, instead of using GentleBoost logistic regression they used Image-Based Boosting regression and they parametrized the contour using 17 landmarks points and PCA to reduce the shape dimensionality. Their solution outperforms other state-of-art methods while computation is significantly faster.

Afterwards, Cao et al. [2] design an explicit shape regression framework based on boosted regression with pixel-difference feature, i.e., the intensity difference of two pixels in the image. To let the pixel-difference achieve geometric invariance, they index a pixel by the local coordinate with respect to a landmark in the normalized object. To better explore the huge feature space in a short time and generate good candidate ferns, they exploit the correlation between features and the regression target. They compare their method with previous work (LFRW, LFW87, Helen, BioI) and in all cases they have better score in terms of mean square error.

The most recent approach was proposed by Kazemi et al. in 2014 [19] based in cascade of regressors with pixel-difference features. With the aim of improving the previous algorithms they introduce in pixel-difference feature a new term that corresponds to an exponential priori over the distance between pixels used in a split to encourage closer pixels pair to be chosen, also for every stage of the cascade they use gradient tree boosting with a sum of square error loss. Comparing it with [2] this method suffers from a higher error, but surpasses most of the other previously presented methods.
The performances of the cascade regression methods can decrease when the initialization provided by the rigid detection is not accurate enough. To handle this problem Yan et al. [38] propose to generate multiple hypotheses, and then learn to combine them to get the final result. The parameters to combine the hypotheses are learnt via structural SVM framework, which is an algorithm that generalizes the Support Vector Machine (SVM) and the main difference is SVM framework allows training a classifier for general structured output labels. The key point of the article is that, as other articles [19], [2], they try to minimize a regression problem by cascade regressor framework but to render more robust the initialization points, they combine the multiple hypotheses as a SVM problem. Thanks to this improvement, they overcome most of the methods of the state-of-the-art performances.
Chapter 2

Methodology

The delineation of the left ventricle (LV) in ultrasound images is an important tool to produce a quantitative assessment of the health of the hearth. The automation of the LV delineation is desirable in clinical setting since it can increase patient throughput while reducing inter-user variation in the LV delineation procedure.

In this work, we address the LV segmentation problem as a supervised learning problem consisting of two steps: 1) LV pose estimation via localization of the smallest oriented rectangle containing the LV; 2) segmentation of the endocardial/epicardial border. The first step is treated as a binary classification step, whereas the second as a regression step. The main difference between them is that classification attempts to assign each input value to one of a given set of classes and regression assigns a real-valued output to each input. Both can be described as $y = f(x)$ where $x$ is the input feature vector and $y$ output prediction, however in classification $Y \in \{0, 1, ..., J\}$ where $J$ is the number of classes and in regression $Y \in \mathbb{R}$.

The fundamental reason for having these two steps is based on the fact that the rigid detection can be run in a lower dimensionality space than the original contour space, and its results can then be used to constrain the non-rigid segmentation [26]. More specifically, assuming that the explicit representation of the segmentation contour consists of $S$ 2-D points, the complexity of the exhaustive search would be $O(K^{2S})$, where $K$ denotes the number of samples in each of the $2S$ dimensions. The introduction of the rigid detection procedure allows for a significant reduction of $K$ and also constrains the search for visual object borders within a small window.

Usually, the rigid detector finds the center, scale and orientation of the visual objects. Namely, in order to model the unaligned LV in real images, we use a 5D parametrization $\theta = (c_x, c_y, s_x, s_y, \alpha)$, where $(c_x, c_y)$ denotes the LV center, $(s_x, s_y)$ the scale and $\alpha$ the orientation, as proposed in [41].

During my internship I mostly concentrated on the first stage which is crucial for effectively reducing the complexity of the training and search mechanisms. Furthermore, the resulting output is used to initialize and constrain the non-rigid segmentation. For this first part we carried out an exhaustive validation, whereas in the second part we only performed a feasibility study, due to time constraints. We evaluated the potential of the proposed approach and we determined its viability before proceeding with an exhaustive validation.
In this chapter, we describe the method we propose in order to automatically estimate the pose of the left ventricle and predict the endocardial/epicardial border. The chapter is organized as follows: Section 2.1 describes the proposed approach for LV pose estimation which is based on an ensemble learning method, namely on Boosting. In section 2.2 we propose an approach for addressing the segmentation process by exploiting the prediction coming from pose estimation.

2.1 LV Pose estimation

We treat our localization task as a binary classification problem that discriminates the region of interest, i.e. the LV, from the other anatomical structures. As a classifier we propose to use a boosting algorithm that combines the output of many ”weak” classifiers in order to produce a powerful strong classifier having higher classification rate than any single weak classifier.

In this section we will first describe the main idea of boosting, then we will give the specific details for our implementation and how we take into account the challenges described in subsection 1.1.3

2.1.1 Theoretical approach

Ensemble Learning refers to the procedures employed to train multiple learning machines and combine their outputs, treating them as a committee of decision makers. The principle is that the strong decision, with all weak classifiers combined appropriately, should have better overall accuracy than any individual committee member. Numerous empirical and theoretical studies have demonstrated that ensemble models very often attain higher accuracy than single models [35], [22]. The members of the ensemble may predict real-valued numbers, class labels, posterior probabilities, rankings, clustering. Therefore, their decisions can be combined in several ways, such as using averaging, voting, and probabilistic methods. The majority of ensemble learning methods are generic, applicable across broad classes of model types and learning tasks. Some examples are Bagging and Random Forest, which are ensemble methods for classification, where a combination of trees casts a vote for the predicted class, and Boosting where a committee of weak learners is trained iteratively and the votes from theses learners are cast into a weighting rule. Boosting arose from a theoretical question posed by Kearns [20]: ”Can a set of weak learners create a single strong learner?” where a weak learner is defined to be a classifier which is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

Historically Boosting algorithms can be viewed as a way of fitting an additive model of the form:

$$f(x) = \sum_{i=1}^{N} \beta_i G(x; \gamma_i)$$  \hspace{1cm} (2.1)
where $x$ is input data, $\beta_i$ are the expansion coefficients, i.e. the weights of the contributions of weak classifiers. $G(x; \gamma_i)$ are the basis functions, in Boosting these are the individual classifiers $G_m(x) = G(x; \gamma_i) \in \{-1, 1\}$, where $\gamma_i$ is the parametrization of the classifiers. Additive models are fit by minimizing a loss function $L(y, f(x))$ averaged over the training data:

$$
\min_{\{\beta_m, \gamma_m\}_{m=1}^M} \sum_{i=1}^N L \left( y_i, \sum_{m=1}^M \beta_m G(x_i; \gamma_m) \right).
$$

(2.2)

where $y_i$ is the prediction for the $i$th example

In forward stagewise modelling the learning procedure starts with $f_0(x) = 0$ and adds new basis functions sequentially, without adjusting the parameters and coefficients of those that have already been added. Hence, at iteration $m$ the algorithm finds the new expansion coefficient $\beta_m$ and the parameters $\gamma_m$ of the classifier $C_m$ by solving:

$$
(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta G(x_i))
$$

(2.3)

where $f_m(x) = f_{m-1}(x) + \beta_m G(x)$.

The most popular commonly used version of boosting algorithm is AdaBoost, which was proposed by Schapire and Freund in 1997 [10].

AdaBoost considers a two-class problem, with the output variable codes as $Y \in \{-1, 1\}$. Giving a vector of prediction variables $X$, a classifier $G(X)$ produces a prediction taking one of the two values $\{-1, 1\}$. The prediction from all of them are combined through a weighted majority vote to produce the final prediction $G(X)$. The data modification at each boosting step consists of applying weights $\omega_i$ to each training observation. Algorithm 1 shows the details of the AdaBoost algorithm [15].

**Algorithm 1** AdaBoost [15]

1. Initialize the observation weights $\omega_i = 1/N, i = 1, 2, ..., N$.
2. for $m = 1$ to $M$ do
3. Fit a classifier $G_m(x)$ to the training data using weights $\omega_i$.
   
   $G_m = \arg \min_G \sum_{i=1}^N \omega_i^{(m)} \cdot I(y_i \neq G_m(x_i))$
4. Compute
   
   $err_m = \frac{\sum_{i=1}^N \omega_i \cdot I(y_i \neq G_m(x_i))}{\sum_{i=1}^N \omega_i}$
5. Compute:
   
   $\alpha_m = \log((1 - err_m)/err_m)$
6. Set $\omega_i \leftarrow \omega_i \cdot \exp[\alpha_m I(y_i \neq G_m(x_i))]$, $i = 1, ..., N$
7. end for
8. Output $G(x) = \text{sign}[\sum_{m=1}^M \alpha_m \cdot G_m(x)]$
AdaBoost is equivalent to forward stagewise additive modelling using the exponential loss function \( L(y, f(x)) = \exp(-yf(x)) \) in Eq. 2.2.

Using the exponential loss function, we solve:

\[
(\beta_m, G_m) = \arg \min_{\beta, G} \sum_{i=1}^{N} \exp\left[-y_i f_{m-1}(x_i) + \beta G(x_i)\right]
\]  (2.4)

If we change \( \exp(-\beta y_i G(x_i)) \) by \( w_i^{(m)} \), \( w_i^{(m)} \) depends neither on \( \beta \) nor \( G(x) \), it can be regarded as a weight that is applied to each observation. This weight depends on \( f_{m-1}(x) \), and so the individual weight values change at each iteration \( m \). Equation 2.4 can be rewritten as:

\[
G_m = \arg \min_{G} \sum_{i=1}^{N} \omega_i^{(m)} \exp\left[-\beta y_i G(x_i)\right]
\]  (2.5)

Comparing line 3 in 1 and equation 2.5, we can see that AdaBoost fits effectively an additive model.

In the literature there exist several variations of boosting algorithms: LPBoost, TotalBoost, BrownBoost, MadaBoost, LogitBoost, and others [11],[7],[12]. The main difference between these algorithms is the kind of cost function they optimize on the training set, but all of these rely on performing a generic "greedy" gradient descent in the function space by optimizing a convex cost function like in Eq. 2.2.

### 2.1.2 Proposed approach

In this subsection, we will describe in detail all the steps involved in dealing with the localisation problem. The first one consists of a preprocessing step that normalizes all the images and extracts some positive windows, which contain the left ventricle, and some negative windows, which do not contain the left ventricle from every ultrasound image. In the second step, we calculate the Haar-like features for every window; these features will be the input of the classifier. As classifier we will test AdaBoost and Boosted cascade. Figure 2.1 is a schematic of the proposed approach.
Chapter 2. Methodology

Figure 2.1: Schema of the proposed approach for left-ventricle localisation.

a) Building the training set

In each sequence, the first frame has the epicardium and the endocardium annotated as well as three anatomical key points (the apex, and the two valves see Figure 3.1), which represent the ground truth. Using the epicardial borders and the three key points it is possible to determinate the minimum bounding box that contains the left ventricle. This bounding box is defined as the box with the smallest measure in terms of area within which all the points lie. This bounding box is defined by the five parameters discussed in the previous section (center: $c_x, c_y$, scale: $s_x, s_y$ and rotation: $\theta$).

In order to determinate the minimum oriented bounding boxes we use the method proposed by Gottschalk [14] based on the covariance matrix of the contour points $p=(p_1, ..., p_N)$ where $N$ is the number of contour points. The first step is to compute the mean position $m$ and covariance matrix $C$ of the points.

$$m = \frac{1}{N} \sum_{i=1}^{N} p_i$$  \hspace{1cm} (2.6)

$$C = \frac{1}{N} \sum_{i=1}^{N} (p_i - m)(p_i - m)^T$$  \hspace{1cm} (2.7)

Then, we extract the eigenvectors $V_0, V_1$ of the covariance matrix via a single value decomposition (SVD).
Then we build a transformation matrix $T$ using the mean position and eigenvectors as:

$$T = \begin{pmatrix} V_{0x} & V_{1x} & 0 \\ V_{0y} & V_{1y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Then, we transform each point $p_i = (x_i, y_i)$ into the local frame of the box using $T^{-1}$ and we calculate minimum $p_{\text{min}}$ and maximum coordinates $p_{\text{max}}$ in each direction. Hence, the half-extents of the bounding box is $\delta = (s_x, s_y) = \frac{p_{\text{max}} - p_{\text{min}}}{2}$, and the center of the bounding box is $p_{\text{cen}} = (c_x, c_y) = \frac{p_{\text{max}} + p_{\text{min}}}{2}$ multiplying by $T$. Whereas the orientation matrix is $[V_0, V_1]$, from which we can obtain $\theta$.

According to the computed bounding box we rotate the image to obtain a rotation-invariant window, which will be considered as the ground-truth bounding box in all this section.

Given an echocardiographic sequence and its LV annotation, we extract the minimum bounding box which will be considered as a positive window, and we also collect additional positives windows by moving this bounding box randomly by a small amount around its center. Negative training images are sampled by extracting windows at significantly far away locations from positive examples. (see Figure 2.1).

In order to determine which of sampled windows are to be considered as positive/negative, we use the Jaccard similarity coefficient (2.12), which measures the closeness between a sampled window and the ground-truth bounding box, and it is defined as the size of the intersection divided by the size of the union of the windows. Thus, a window is considered as positive if $J$ is above a certain threshold and negative otherwise.

All examples windows used for training are variance normalized to minimize the effect of different lighting conditions.

b) Haar-like features

There are many reasons for using features instead of raw pixel values to represent image appearance. The major one is that image features provide low-level descriptions of image contents, which are valuable for many image analysis applications. Many image features have been proposed in the literature, such as Canny edge detector [1], steerable filters [9], and SIFT features [24]. However, not all of them can handle ultrasound problems described in Sec. 1.1.3.

In our approach we use Haar-like features as proposed by Viola and Jones [17] for three reasons. First, Haar-like features can be computed very quickly, enabling efficient object detection algorithms. Second, in an image region there is a large number of Haar-like features, each providing weak evidence of the image content. Third, the Haar-like local rectangles coarsely filter components in various directions and resolutions. However, the main drawback of Haar-like features is that they are not invariant to lighting conditions, rotation and translation.
Haar features are simple and inexpensive image features based on intensity differences between rectangle-based regions that share similar shapes with the Haar wavelets.

More specifically, the most used are the two-rectangle feature, which computes the difference between the sum of pixels within two rectangular regions that have the same size and shape and are horizontally or vertically adjacent, the three-rectangle feature, which computes the sum within two exterior rectangles subtracted from the sum in a center rectangle, and the four-rectangle features, which computes the difference between diagonal pairs of rectangles (See Figure 2.2).

![Figure 2.2: Haar-like features [17].](image)

Haar features can be computed very rapidly using the integral image where the value at any point \((x, y)\) is the sum of the pixels above and to the left of \((x, y)\) inclusive. Moreover, the summed area table can be computed efficiently in a single pass over the image, using the fact that the value in the summed area table at \((x, y)\) is just:

\[
I(x, y) = i(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1)
\]

Once the summed area table has been computed, the task of evaluating any rectangle can be accomplished in constant time with just four array reference. However Haar-like feature are not invariant with respect to rotation. To handle with this problem is very important the pre-processing step that assures that in all the training images the LV has coarsely the same orientation.

Before computing Haar-like features the training and test set of positive and negative images are downsampled in order to reduce the dimensionality of the input of the AdaBoost classifier. This is an important point because thanks to the downsampled we can extract features at a scale where they are less affected by noise and more sensitive to the relevant anatomical landmarks, while reducing the dimensionality. As Haar features are just differences of pixels, the amount of output features increase dramatically with the bounding box size. Working on a coarser scale due to downsampling is not critical for our pose estimation problem, as we only aim at coarsely localizing the LV before addressing the true segmentation.

For pose estimation purpose we tested two different implementations of Boosting algorithms, the first one is AdaBoost presented before and the second one is a generalization of AdaBoost called Boosted cascade that was proposed by Viola and Jones in 2001 [17] and has shown to be one of the best performing techniques for face detection.
c) AdaBoost classifier

We consider our pose estimation problem as a binary classification problem, since we aim to separate the object of interest from the background. In our framework, each example consists of a rectangular, oriented bounding box of given parameters $c_x, c_y, s_x, s_y$ and $\theta$ and is labelled as positive or negative according to the amount of its intersection with the ground truth LV bounding box.

We address this binary learning task in the boosting framework, which has proven successful for several object detection/localization applications. In particular, we propose to learn our classification rule according to the AdaBoost model, that take as input a training set $(x_1, y_1), \ldots, (x_N, y_N)$ where each $x_i$ belongs to Haar-like feature domain $X$, and each label $y_i$ correspond to the binary class (LV or background). AdaBoost combine a collection of weak learners to form a strong classifier. Each weak classifier is a given function which returns a boolean value, the result is 1, if $x$ is classified as positive example and 0 otherwise. Whereas the weak classifiers only need to be slightly better than a random guessing, the combined strong classifier typically produces good results [10]. After each round of learning, the examples are reweighted in order to increase the importance of those which were incorrectly classified by the previous weak classifiers. The final strong classifier takes the form of a perceptron, i.e. a weighted combination of weak classifiers followed by a threshold as described in algorithm 1.

Throughout our work we used the variant of the AdaBoost algorithm proposed by Viola and Jones [36] for their face detection approach. This variant restricts the weak classifier to depend on single-valued features $f_m$ only. Each weak classifier has the form

$$G_j(x) = \begin{cases} 
1 & \text{if } p_m f_m(x) < p_m \theta_m \\
0 & \text{otherwise}
\end{cases}$$

where $\theta_m$ is a threshold and $p_m$ is either -1 or 1 and this representing the direction of the inequality. The algorithm determines for each weak classifier $G_m(x)$ the optimal values for $\theta_m$ and $p_m$, such that the number of misclassified training examples is minimized. To achieve this, it considers all possible combinations of both $\theta_m$ and $p_m$, whose number is limited since only a finite number of training examples is given:

$$\left( p_m, \theta_m \right) = \operatorname*{argmin}_{(p_m, \theta_m)} \sum_{i=1}^{N} |G_i(x_n) - y_n|$$

(2.8)

where $x$ is a windows of an image.
d) Boosted cascade classifier

Grounding on the AdaBoost framework, Paul Viola and Michael Jones proposed [17] a classification approach for visual object detection which is still capable of processing images extremely fast but achieves better detection rates than regular AdaBoost.

Namely, they proposed a cascade structure, as shown in Figure 2.3, that focuses on image regions having high probability of containing objects of interest. The cascade is formed by a set of successively more complex and hence more discriminative Adaboost classifiers. For a window to be considered as a positive, namely containing an object of interest, it must pass all stages of the cascade. The computational efficiency is gained by the fact that the low-level classifiers can filter out a large number of negatives with a low computational cost. Only a small number of candidates, having high likelihood of containing object of interest, advance to higher levels, which involve more computation to differentiate between positives and negatives.

Figure 2.3: Schematic description of Boosted Cascade [17]. A window has to pass all the classifiers in the cascade to be considered as a positive (T). If a window is failed by any classifier, it is regarded as a negative (F).

Boosted cascade provides a high constant sensitivity and a growing specificity through the nodes obtained by connecting more simple classifiers with high sensitivity and sufficient specificity. Giving a trained cascade of classifiers, the false positive rate of the cascade is:

$$F = \prod_{i=1}^{K} f_i$$  \hspace{1cm} (2.9)

where F is the false positive rate of the cascade classifier, K is the number of classifiers, and $f_i$ is the false positive rate of the $i$th classifier on the examples that get through it. And the detection rate is:

$$D = \prod_{i=1}^{K} d_i$$  \hspace{1cm} (2.10)

where D is the detection rate of the cascaded classifier, K is the number of classifiers, and $d_i$ is the detection rate of the $i$th classifier on the examples that get through it.
False negative rate and detection rate are defined as:

\[ f_i = \frac{FP}{FP + FN} = 1 - SPC, \]
\[ SPC = \frac{TP}{TP + FP}, \]
\[ d_i = TPR = \frac{TP}{TP + FN} \quad (2.11) \]

where FR is the false positive, FN false negative, TP is true positive, TF is false negatives, SPC is specificity and TPR is the selectivity.

Algorithm 2 shows the learning algorithm for Boosted cascade.

**Algorithm 2** Boosted cascade [17]

1. Given examples images \((x_1, y_1), \ldots, (x_n, y_n)\) where \(y_i = 0, 1\) for negative and positive examples respectively.
2. Initialize the observation weights \(\omega_i = 1/N, i = 1, 2, \ldots, N\), so what \(\omega_i\) is a probabilistic function.
3. for \(t = 1\) to \(T\) do

   4. For each feature, \(j\), train a classifier \(h_j\) which is restricted to using a single feature.
      The error is evaluated with respect to \(\omega_i, \varepsilon_t = \sum_i \omega_i^{(m)} \cdot |h_j(x_i) - y_i|\)
   5. Choose the classifier, \(h_y\), with the lowest error \(\varepsilon_t\).
   6. Update the weight:
      \[ \omega_{t+1,i} = \omega_{t,i} \beta_t^{1-\varepsilon_t} \]
      where \(e_i = 0\) if examples \(x_i\) is classified correctly, \(e_i = 1\) otherwise, and
      \[ \beta_t = \frac{e_t}{1-\varepsilon_t} \]
   7. end for
   8. The final strong classifier is:
      \[ h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases} \]
      where \(\alpha_t = \log \frac{1}{\beta_t}\)

Boosted cascade presented by Viola and Jones [17] was the first algorithm that achieved real-time detection speed and high accuracy on face images, and it is still one of the most successful and widespread techniques for face detection. Then, this method was used with some modifications in generic object recognition such as in [29] where complex objects like bikes, persons or cars in different poses and scales are learned with recognition rate 87%.

In the training process, the input parameters of the cascade are:

- Matrix of Haar-like training features \(x=x_{1m}, \ldots, x_{nm}\), where \(n\) is the number of examples to train and \(m\) the number of features by example.
- Train label vector \(y=y_1, \ldots, y_n\).
- Matrix of haar-like test features \(x=x_{1z}, \ldots, x_{kz}\), where \(z\) is the number of images for test.
• Test label vector $y = y_1, \ldots, y_k$.
• Minimum specificity in every stage $f_i$ described in 2.9.
• Minimum global selectivity $D$ described in 2.10.
• Minimum global specificity $F$ described in 2.9.
• Number of iterations $k$ in every stage which is equivalent to the number of weak learners.
• Type of features used (edge features, line or four-rectangle features, or the same rotated as explained in Sec. 2.2).
• Output classifier $h_t$.
• Previous classifier $h_{t-1}$

In every stage the classifier outputs the specificity $1 - f_i$ and the selectivity $d_i$.

e) LV parameters estimation

A common way to localize an object in an image is to consider all the possible object configurations in the image by performing exhaustive window scanning over the input image as it was proposed by Viola and Jones [17]. To avoid searching the whole configuration space which hampers real-time detection, we can use the a priori knowledge of LV pose parameters $(c_x, c_y, s_x, s_y, \theta)$ to draw randomly multiple windows, assuming that at least one bounding box localises is closer enough to the ground truth pose.

From every parameter we calculate the probability distribution function (pdf) which is used as a prior for drawing randomly a given number of windows from that constitute a dictionary set. We scale down the dictionary set to the same size than training examples, and then we extracted their Haar-like feature, and we use this features, and we used this features as input for the classifier previously trained.

With the aim of calculating the performances in the validation, all the test examples should be labelled as positive or negative. To this end, we use both the Jaccard similarity coefficient (2.12) and the Sørensen–Dice index (2.12), which normalizes the intersection area by the area of the two sample sets.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad S(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$  (2.12)
Boosting classifiers output a discriminant function (strong classifier) \( G \) (Eq. 2.13), where the meaning of \( G \) is that the higher the \( l \)th element \( G_l(x) \) of \( G(x) \), the more likely is that the real label index of \( x \) is \( l \). The vector-valued classifier \( G \) is formally obtained by simply thresholding the elements of \( G(x) \) at 0.

\[
G(x) = \text{sign}\left[ \sum_{m=1}^{M} \alpha_m \cdot G_m(x) \right]
\]  

(2.13)

where \( G(x) \) is a vector that contains the score of the predicted class.

To determine the winning windows several methods have been proposed in the literature. Some of these methods are based on choosing the example with the highest score, whereas others calculate a probabilistic map from the strong classifier output \( G \) and use maximum a posteriori.

In our case, from the discriminant function \( G \), we calculate a probabilistic map from the windows as positives. Then, from this probabilistic map we extract the minimum bounding box that contains more than a fixed percentage of the points of the probabilistic map.

- 6. Challenges

Boosting has been applied to many tasks but is particularly successful for face/object detection/localization. Our domain is even more challenging due to several problems of ultrasound images, like slow signal-to-noise ratio, edge dropout and shadows. We address these problems by proposing to adapt the AdaBoost framework to the domain of cardiac ultrasound images. To this end we have introduced several modifications that improve the recognition performances, which we will describe in next paragraphs.

Haar features outside the ultrasound cone

2-D echocardiographic images typically consist of a cone-shaped region in the center of the image, which contains the ultrasound signal, while the surrounding pixels do not contain any signal. Since the windows are sampled at arbitrary locations in the image, they may intersect the boundary of the cone, thus resulting in fictitious gradient-based features. In order to deal with this problem we only calculate the Haar features whose rectangular support is inside the cone, treating all the remaining features as missing not to bias the classification results.

Initial weights

By default the Boosting algorithm 1 initialises the observation weights uniformly. However, in our project we deal with an unbalanced database, i.e., the number of negative windows is much higher than the number of positive windows. This means that initializing the weights uniformly would give much more importance to the negative examples that to the positive ones. In order to equalize the weights of positive and negative examples, we use the same technique proposed in [34], which redefines the initial weights \( \omega_i = 1/N, i = 1, 2, ..., N \) as: \( w_{1,i} = \frac{1}{2m}, \frac{1}{2l} \), where \( m \) and \( l \) are the number of negatives and positives respectively.
Then after every iteration of AdaBoost we re-normalize the weights:

\[ w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}} \]  

(2.14)

Rotation-invariance for the boosted cascade

The original Boosted cascade proposed by Viola and Jones [17] only detects objects with fixed pose (estimation of the center of the object \( c_x, c_y \)). In ultrasound images LV may be captured by transducer in arbitrary poses, thus giving rise to large variations in terms of scale and orientation. While the fast evaluation of local rectangle features is applicable when dealing with scale variation, it is not straightforward when applying it to different orientations. A simple approach is to train a classifier by pooling data from all possible orientations.

To deal with this problem we decided to rotate the image for a given set of orientations and compute the corresponding integral image for each orientation. This approach is only possible when the size the image is reduced, otherwise it is computationally prohibitive.

2.2 Supervised LV Segmentation

In the previous section we have presented our approach to LV pose estimation, which is the first stage in our two-step automatic LV segmentation approach. In this section we describe how we can address the endocardium/epicardium segmentation as a regression task by relying on the LV estimated pose. First of all, we describe the approach according to the general supervised segmentation framework, and then we explain the specific algorithm we implemented. For this purpose, we ground on ideas related to the Active Shape Models [4] which is a statistical model of the shape of one or more objects, that can be deformed in order to fit similar objects in other images, as well as on the regression framework proposed in [38] for learning the segmentation of LV.

It is important to note that the segmentation is performed separately for every type of view, as the textural information significantly depends on the type of view.

2.2.1 General approach

The Active Shape Model (ASM) has been proposed by Cootes and Taylor in 1995 [4] as a method for fitting a set of global feature detectors to an object while simultaneously taking into account global shape information. The allowed shape deformations are learnt from a manually labelled training set to produce a linear shape model from a statistical analysis such as the Principal Component Analysis (PCA). Hence, the object is defined by a sequence of points belonging to the contour object, called landmarks. The ASM algorithm starts the search for landmarks from the mean shape aligned to the position and the size of object as estimated by a global object detector. Then it repeats the following two steps until converging: first, it looks for a local improvement of the location of each patch in the image; then, it updates the model parameters in order to best match these new positions.
The segmentation problem can be cast as a shape alignment issue in a regression framework. This enables to take advantage from an annotated dataset of images for learning in a supervised way the mapping between visual appearance features extracted at local patches and the landmark positions defining the shape. The input is an ultrasound image, while the output is a predicted shape parametrized via the coordinates of each landmark. In the training process we have N samples $I_i, S_i, S_0^i$, where $I_i$ is the image, $S_i$ the ground truth and $S_0^i$ is the initialisation of $S_i$, which corresponds to the mean shape. We aim at learning a regression function $f$ that minimizes the following mean square error:

$$\argmin_f \sum_{i=1}^{N} \| f(S_0^i, I_i) - S_i \|_2$$

(2.15)

The direct regression is very complex, due to the high dimensional output and complex non-linear relationship. However, alternative optimization techniques have proven successful in solving 2.15, such as the cascade regression approach, which can be viewed as a generalization of the boosting framework for regression. In this work, we ground on the algorithm proposed by [38], splitting $f$ into a series of simpler regression functions. The sub-regression function is defined by a feature transform procedure

$$S_t^i = f(S_{t-1}^i, I_i) = W_t \cdot \Phi(S_{t-1}^i, I_i)$$

(2.16)

where $\Phi(S_t^i - 1, I_i)$ is the feature transform encoding the appearance of Image $I_i$ around shape $S_t^i - 1$. An example of features transform can be HOG [6], Gabor filters, etc. The equation 2.15 can be redefined as:

$$W_t = \argmin_{W_t} \sum_{i=1}^{N} \| S_i - W_t \cdot \Phi(S_{t-1}^i, I_i) \|_2$$

(2.17)

where $W_t$ can be solved as a multivariate least square problem.

### 2.2.2 Proposed approach

Before applying the segmentation algorithm and in order to improve the invariance to the pose parameters (position, scale and orientation), we perform a rigid registration of training images as proposed in [2].

During a 2D echo exam, the traducer position is varied to elicit important information about the heart function and its anatomy. The goal of the image registration is to recover the geometric transformation that brings the images into alignment. This mapping aims at determining the optimal rotation and translation parameters for aligning the LV endocardial/epicardial borders arising from two different images. To this end we calculate the average contour as the centroid of all the contour shapes and we center each shape on its centroid. The mean shape is defined as:

$$m = \sum_{i=1}^{N} \frac{1}{N} \hat{S}_i$$

(2.18)

where $\hat{S}_i$ is the registration points of $S_i$. 
Then we calculate the rotation matrix $M$ by a singular value decomposition. Matrix $M$ which is a $m \times m$ real square matrix with positive determinant that can be factorized of the form:

$$M = U\Sigma V^*$$ (2.19)

Where $\Sigma$ can be regarded as a scaling matrix and, $U, V^*$ can be viewed as rotation matrix. Thus, it can be interpreted as a composition of three geometric transformations: a rotation, a scaling, and another rotation.

The rigid registration is only used in training, then in test we will use the information provided by the classification approach to set the mean shape in the bounding box. In both cases, training and test, after the image registration we perform a contrast normalization to reduce speckle noise and improve image quality for the segmentation propose.

a) Gabor filter

As a feature transform we select Gabor features, which not only have the advantage of including the Gaussian operator, that reduces the effects of ultrasound noise but also are defined as band-pass filters at multiple orientations, thus giving rise to effective edge detectors [8]. Indeed, Gabor features are constructed from responses of 2D Gabor filters that correspond to a Gaussian kernel function modulated by a sinusoidal plane wave. Namely, a 2D Gabor filter is defined as follows:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$ (2.20)

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$, where $\lambda$ is the wavelength of the cosine factor of the Gabor filter kernel and herewith the preferred wavelength of this filter, $\theta$ is the parameter that specifies the orientation of the normal to the parallel stripes of a Gabor function, $\psi$ is the phase offset, $\sigma$ is the sigma/standard deviation of the Gaussian envelope and $\gamma$ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

For our purpose, we calculate K Gabor 2D filters with different orientations, with the same frequency and variance of the Gaussian for all the orientations, where $K$ corresponds to the number of orientation. The features are extracted as the value of the landmark point in the new image that corresponds to the convolution between the image and the filter with the nearest orientation of the normal orientation of the landmark $X$. As the ground truth may be imprecise, and the position of the landmark slightly shifted, we convolve locally for every landmark the image with a Gaussian filter to reduce the effect such annotation errors.(see Figure 2.4).

The resulting Gabor feature vector $\Phi = \Phi_1, ..., \Phi_N$ can be defined as:

$$\Phi_i = f(x, y; \lambda, \theta, \psi, \sigma, \gamma) \otimes H_G(x, y)$$ (2.21)

where $H_G(x, y)$ correspond to the 2D Gaussian filter and $f$ can be defined as:

$$f(x, y; \lambda, \theta, \psi, \sigma, \gamma) = g(x, y; \lambda, \theta, \psi, \sigma, \gamma) \otimes I$$ (2.22)
where \( g(x, y; \lambda, \theta, \psi, \sigma, \gamma) \) is a filter bank consisting of Gabor filters with various scales and rotations, \( I \) is the image and \( x, y \) are the landmark points (see Figure 2.4 and 2.5).

Figure 2.5 shows some examples of information that return the convolution of the Gabor filter with the image \( f(x, y; \lambda, \theta, \psi, \sigma, \gamma) \), we can see that for smaller values of \( \sigma \) the filtered image is similar to the original with the contours slightly more pronounced, whereas for higher values of \( \sigma \) the image is more blurred and we can only differentiate between a bright area and a grey area, if we compare different Gabor image convolution from different patients we note that the left ventricle wall is always lighted however the apex and the right ventricle in many image is not present. As the annotation is not always precise we want the filter support to be sufficiently large and comparable to the average thickness of the myocardium wall. Is important to note, that the textural information obtain by Gabor is normally situated in the left wall of the LV and the shape is place thanks to this information and to the a priori information of the shape, but in the case that there are many variations in the right wall in many cases it is impossible to have an accurate segmentation as the textural information is not visible in Gabor.
b) Dimension reduction via PCA

As we explained in the Section 2.2 the main idea is to model the mapping between the geometrical information of the shape of the contour and the texture information of the image. To this end we formulate the problem as the estimation of the unknown matrix $W$ in Eq. 2.17 via least square optimization. The problem is that the matrix to estimate has dimensionality $K \times N$, where $K$ is the number of features, which in our approach match with the number of landmarks, and $N$ is the total number of contour point coordinates.

In order to reduce the dimensionality we represent both the features and the geometric annotation via the firsts coefficients of principal component analysis, this is one of the most important point in this section, because the number of training examples is reduce and it will be impossible to estimate robustly a high dimensionality matrix with few data.

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables, the principal components. These components are chosen such that the first has the largest variance, the second has the second largest variance, etc. The assumption when using a PCA method is that the data set has a multivariate normal distribution.

With the aim of continuing to reduce the dimension of the matrix, we will apply PCA algorithm to the shape database and to the Gabor feature vector, in both cases we will take the $N$ first principal components that retain at least 90% of total variance which can be considered as a sufficient approximation of the data.

c) Training and Test initialisation

For the training purpose we combine multiple initialisations to improve the robustness of the algorithm. The multiple initialisations are done by slightly deforming the first components of the PCA algorithm for the shape; these deformations create new shapes that are similar to the original but with slight variations. Thanks to the multiple initialisations the algorithm is more robust against possible non-accurate annotations. Also, this process increases the training database and gives more examples for learning the regressor. In order to ensure that the new shape is similar to the ground truth we force that the distance between different initialisations to be less than two pixels.

For testing purpose, we use the estimated bounding box returned by Boosting classifier to rotate and translate the image in order to have the same centroid than in training. Then, we place the mean shape calculated from all the training shapes in the estimated bounding box and we use this initialisation as the input for the cascade regression framework.
Chapter 3

Results

This chapter is focused on the validation process, a quantitative analysis of the results obtained by running the proposed methods on a clinical database. Section 3.1 is focused on presenting the dataset and the settings used during my internship, the section 3.2 presents results of the LV pose estimation and the section 3.3 shows a feasibility study of the regression-based.

3.1 Database and settings

We tested our localisation and segmentation approach on a database of ultrasound cardiac sequences in the three apical views (2 chambers, 3 chambers and 4 chambers) provided by Philips Medisys [13], with the ground truth manually annotated on the first frame of every sequence, consisting of the epicardial and endocardial contours. An example of the three image classes is shown in Figure 3.1

This database is composed of 39 sequences, each one related to a different patient and containing few cardiac cycles in each of three apical views. After database verification, we retained 121 annotated sequences (42 for A2C view, 37 for A3C view and 42 for A4C view).

In each sequence, the first frame was annotated with a semi-automatic segmentation tool by a cardiologist, which built a ground truth database. The workflow is as follows. First, the physician clicks some points of interest on the first image of the sequence: in one of the apical views, these points correspond to the position to the apex (Ap), and the two extremities of the myocardium, near the valves (end valve points $V_1$, $V_2$). An interactive tool suggests a segmentation of the myocardium, that is to say the endocardial and epicardial borders on the first frame, using the manually initialized points (Ap,$V_1$, $V_2$). An edit function allows for the correction of possible errors in the boundary segmentation. Figure 3.1 shows an example of such annotation for the classes of apical views.
Chapter 3. Results

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

Figure 3.1: Segmentation of the epicardium (yellow) and the endocardium (green) on the Philips database.

Thanks to the epicardial borders it is possible to determinate the minimum bounding box that contains the left ventricle, which is defined by the five pose parameters $(c_x, c_y, s_x, s_y, \theta)$. We consider these parameters as statistically independent and distributed according to a Gaussian distribution $\mathcal{N}(\text{mean}, \text{variance})$:

- $c_x \sim \mathcal{N}(311.4, 17.1)$ and $c_y \sim \mathcal{N}(174.4, 15.0)$
- $s_x \sim \mathcal{N}(197.0, 17.4)$ and $s_y \sim \mathcal{N}(271.1, 22.7)$
- $\theta \sim \mathcal{N}(-12.2, 8.4)$

Figure 3.2 shows the histogram of LV pose parameters $c_x, c_y, s_x, s_y, \theta$ and their fitting with a Gaussian, we can see that the most dispersed parameter is $\theta$, whereas the center and the size have low variance and can be more easily detected on our database.

Figure 3.2: Histogram of LV pose parameters $c_x, c_y, s_x, s_y, \theta$ and their fitting with a Gaussian.
Figure 3.3 shows the superposition of the entire ground truth-bounding box as well as the edges of the ultrasound cone. We can see that the region that may contain the LV is a wide area of the image. This variability will make LV localization difficult because we need to look for in a large region and the estimation of pose parameters will be complex.

3.1.1 Settings for pose estimation

For AdaBoost we used the ground-truth bounding box as a positive example and five randomly sampled rectangular windows as negative examples. However, in Boosted cascade, we used three positive windows versus twenty negative windows to ensure that at every stage there are enough negative examples, as suggested by Viola and Jones [17].

The choice of negative windows in training is semi-automatic: first we sample the pose parameter randomly from the prior distribution learned on training images 3.2, then we chose manually what the most convenient windows for training are.

This was an important point because as we sample the parameters randomly, there may be some cases where the distribution of the parameters are similar, therefore two windows are nearly equal, and the idea is to have a representative database of negative examples that are distributed uniformly in regions that do not contain the LV.

As we discussed in section 2.1.2 we assigned positive and negative labels to the sampled windows according to the Jaccard (J) and Dice (S) indices. Namely, we label a window positive if $J > 0.75$ or if $S > 0.8$, otherwise as negative. These parameters are chosen thanks to the validation process that will be explained in the next section. This process can be expensive in terms of computational time, but it is only done once to obtain the training set and then it will be the same for all the test phase.

Then, the training set composed of positive and negative windows is scaled down to size 14x24, and we extract Haar features from these subsampled windows as it is described in subsection 2.1. To this end, we calculate the integral image to reduce computational time, and then we calculate five types of Haar-like features: edge features, line features and four-rectangle features in all the positions, and all possible scales.
In total 17875 features are extracted for every image, 4210 for horizontal edge features, 4309 for vertical edge features, 5096 for four-rectangle feature, 2025 for horizontal line feature and 2235 for vertical line feature. These features form the vector representation for an example and are fed to the classification algorithm. We didn’t use rotation Haar features, as they increased dramatically the number of features in our matrix without significantly improving the results.

3.1.2 Settings for Segmentation

With the purpose of making even a more robust segmentation algorithm, we decide to use not only the information of the first frame of the echocardiographic exam, but the whole sequence. The main problem is that the ground truth is only available for the first frame, thus in order to construct an annotation for the other frames, we use a tracking algorithm to find the LV contours of interest all along the sequence. The tracking algorithm is developed by Philips [33] where motion estimation is based on a fast daemons-like algorithm focusing on myocardial tissue. Figure 3.4 shows an example of a contour tracked along an echocardiogram sequence.

![Figure 3.4: Contour points tracked along Echocardiogram sequence with [33].](image)

Both databases, the original containing only the first end-diastolic frame, and the expanded, containing all the frames of each sequence can be defined as the pair \((I_1, S_1), \ldots, (I_N, S_N)\), where \(I_i\) is a cardiac ultrasound image and \(S_i\) is the shape vector \(S_i = S_{i1}, \ldots, S_{iN}\), where \(S_{ik} = (x, y)\) is a landmark point.

Segmentation algorithm 2.17 is solved by multivariate least square problem, which corresponds to calculate the pseudo-inverse matrix:

\[
W_t = \Phi(S^{t-1}, I)S^T \cdot (\Phi(S^{t-1}, I)\Phi(S^{t-1}, I)^T)^{-1}
\]  

(3.1)

Where \(W_t\) is the estimated matrix of size \(NxN\).

Specifically, our training database for A2C is composed by 42 echocardiogram images and the shape vector contains \(N=20\) landmarks, consequently the estimated matrix \(W_t\) is a 20x20 matrix.
With the aim of reducing the complexity of the algorithms, and as we explain in section 2.2.2 we perform an image registration shape in order to improve the invariance to the pose parameters (position, scale and orientation). Then we calculate the PCA of the shape vector which can be represented by the first four principal components, which correspond to 92% of total variance (see Figure 3.5a).

Thanks to this procedure we have reduced the output dimensionality from 20 landmarks coordinates to 4 PCA coefficients. The mean square error reconstruction is less than 1 pixel comparing the reconstructed shape and the ground truth shape. Gabor features are represented by the first twelve principal components (see Figure 3.5b), which means that the estimation matrix will have a size of 4 x 12, compared to the initial one having size 20 x 20.

Figure 3.5: Variance PCA for shape and Gabor feature vector.
3.2 LV bounding box detection

For the pose estimation purpose we use two different boosting algorithms, the first one is an implementation of AdaBoost-SAMME \[42\] in Python provided by the library scikit-learn \[^{1}\]. The second one, is a MatLab implementation of boosting cascade provided by Philips Medisys, which corresponds to the algorithm originally proposed by of Viola and Jones \[17\].

The evaluation of the performances in the pose estimation problem is done by a quantitative analysis based on leave-k(patient)-out cross validation, it means that the data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. In our case as our database only contains 39 patients, and we want a robust classifier, we decided to perform a leave-one-out cross validation that is a particular case of leave-k-out cross-validation with k = 1. That means that every time one exam is considered as test set and the remaining ones are used for training.

In order to evaluate the classification performances, we carried out an incremental strategy, by solving pose estimation with increasing difficulty: first, we fix the orientation, width and height, thus reducing the pose estimation problem to the prediction of the LV center. Then, we add the two scale parameters as unknowns, by fixing the bounding box orientation as vertical. And finally we estimate all the five pose parameters.

3.2.1 AdaBoost

The results presented for AdaBoost have been obtained for all the five parameters (center, height, width and rotation) with leave-one-out-cross validation. Different criteria can be used to compare the ground truth windows and the obtained windows, however, we will calculate the performances in terms of Jaccard index and Dice index \[^{2}\] and we also calculate the number of ground truth contour points or landmarks (N=20) that are inside the bounding box. Table 3.1 summarize the results obtained for every type.

<table>
<thead>
<tr>
<th>Jaccard index</th>
<th>Dice index</th>
<th>Nb points inside bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2C 81.8%</td>
<td>A2C 95.1%</td>
<td>A2C 99.5%</td>
</tr>
<tr>
<td>A3C 80.2%</td>
<td>A3C 94.3%</td>
<td>A3C 98.7%</td>
</tr>
<tr>
<td>A4C 84.3%</td>
<td>A4C 99.5%</td>
<td>A4C 95.4%</td>
</tr>
<tr>
<td>Average 82.1%</td>
<td>Average 96.3%</td>
<td>Average 97.9%</td>
</tr>
</tbody>
</table>

Table 3.1: Jaccard and Dice index in AdaBoost and number of points of the contour inside the estimate bounding box.

\[^{1}\]http://scikit-learn.org/stable/
We can see that in the three viewtypes (A2C, A3C and A4C) almost 98% of the landmarks are on average inside the bounding box. This result is very important because pose estimation step aims at reducing the search space for segmentation algorithm, and the most important is that all the contour points of the LV are contained in the bounding box. If we increase 5% the size of the bounding box in the three cases (A2C, A3C and A4C), the estimate bounding box will contain all the points of the LV contour.

After results 3.1 we note an important difference between Jaccard and Dice index, this difference is due that Jaccard index is related to the precision and Dice index to the recall which measures how good is the detection. For examples, if we have a bounding box too large or too rotate, but it contains the left ventricle, the Dice index will be higher because is well detected, but Jaccard index will be lower as it is not precise.

Figure 3.6 shows three examples of bounding boxes that contain the LV but in every case the pose parameters are different. First image shows a bounding box that is too large and rotate, and in this case Dice index is 100% however Jaccard index is only 55%, which means that has a good detection but bad localization. Second image shows a bounding box slightly scaled, and in this case Jaccard index is just 76%. Last image shows an accurate bounding box, and in this case Dice and Jaccard index are nearly 95%, which can be considered as a good detection and localisation.

Figure 3.6: Jaccard index (J), Dice index (D) and number of points inside the bounding box (nbPoints) for estimate bounding box (yellow) and ground truh (pink).

On average, we have a good compromise between localization and precision, Dice index is above 95% and Dice index above 82%, and we can consider the estimate bounding box a good initialization for segmentation algorithm.

It is important to note that AdaBoost algorithm does not directly minimizes Jaccard or Dice indices, as it minimizes the performances in terms of sensitivity and specificity. In order to evaluate the performances of the classifier and to estimate which is the optimal number of weak learners, we perform a leave-one-out cross validation exam as a function of the number of iterations, and for each case we calculate the precision or specificity and the recall or selectivity (see equation 2.11). Precision measures the accuracy of the localization while recall measures the accuracy of the detection. Figure 3.7 shows the precision and recall curves base on the number of iterations.
Chapter 3. Results

We note that the precision increases with the number of iterations while the recall slightly decreases (from 63% to 58%): this means that about one window annotated as positive is recognized as positive. It is important to note, that in the test we label the windows thanks to Jaccard and Dice index, but this measure may not be precise. For example, there are several cases where the bounding box is considered as positive because the Jaccard index is over 75% but the bounding box is shifted so that one or more segments of the epicardium are not included in it, and in classification this bounding box will be considered as negative.

To choose the number of iterations is important to consider that more iterations mean more weak learners and higher complexity, therefore longer computational time. We need to have a compromise between good precision/recall and as few as possible weak learners to avoid over-fitting while achieving reasonable computational time. Finally, we decided to take 20 iterations as the precision is over 80% and recall over 60%, and computational time is acceptable. With this parameter we obtain the results shown in Table 3.1.

Figure 3.8 presents one example of LV pose estimation for each type of view using AdaBoost. On the left, we have the ultrasound image with the bounding box and the ground truth contour in pink, and in yellow the proposed bounding box, on the right we have the corresponding probabilistic map with the ground truth and the bounding box that contains 70% of the points. In all the images almost all the points of the contour are inside the bounding box even if the bounding box does not perfectly fit to the ground truth annotation due to translation, scaling or rotation. We observed that the difference between the ground-truth bounding box and the estimated one is just few pixels.

The images presented in 3.8 are extracted from a leave-one-out exam, where the training database is composed of 121 images, from every image of the database we extract one positive and six negatives windows. We used twenty weak learners to train the classifier with non-uniformly initial weights and in the test we drew randomly 500 corresponding to different random combinations of the pose parameters.
Table 3.2 shows the performances for the examples presented in Figure 3.8.

<table>
<thead>
<tr>
<th>Jaccard index</th>
<th>Dice index</th>
<th>Nb points inside bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2C: E03</td>
<td>77.8%</td>
<td>A2C: E03</td>
</tr>
<tr>
<td>A3C: E16</td>
<td>79.5%</td>
<td>A3C: E16</td>
</tr>
<tr>
<td>A4C: E27</td>
<td>83.4%</td>
<td>A4C: E27</td>
</tr>
</tbody>
</table>

Table 3.2: Jaccard and Dice index in AdaBoost and number of points of the contour inside the estimate bounding box in Figure 3.8.

Figure 3.8: Examples of LV localisation with AdaBoost. On the left, the image with ground truth (pink) and the proposed bounding box (yellow), on the right the corresponding probabilistic map. The first pair of images is a 2 chamber view, the second a 3 chamber view while the last is a 4 chamber view.
3.2.2 Boosted cascade

With the aim of improving the performances obtained with Adaboost we tested Boosted cascade. The key is insight that the first nodes of the cascade have a simpler task, rejecting the most distinguishable background tissue regions, while the last ones are specialized to discriminate between actual LV and the background tissue.

The implementation that we use performs an exhaustive research in test, which mean that a fixed bounding box look over all the image, then we construct a probabilistic map to determine the most suitable center of the cavity. However, unlike our AdaBoost implementation, the Boosted Cascade implementation we used cannot handle missing features like the Haar features having support outside the ultrasound cone, as explained in 1.1.3.

As in AdaBoost we calculate the performances in terms of Jaccard and Dice index with a leave-one-out-cross validation. In this case, we have first tested the performances assuming that the rotation is fixed, so we estimate four of the five parameters \((c_x, c_y, s_x, s_y)\). Then we have increased the complexity estimating the five parameters \((c_x, c_y, s_x, s_y, \theta)\).

Before presenting the results it is important to remark that after a study the number of cascade needed to obtain the best performances in terms of Jaccard and Dice index is two, which means that we only have two AdaBoost classifiers concatenated. The problem is that depending on the exam the number of cascades needed may change, and we take the average to automatize the classification but there are some outliers that cannot be handling. Each classifier needs in average seventeen weak learners for training, which is more or less the same result that we have obtained with AdaBoost in previous subsection (Figure 3.8).

Table 3.3 summarizes the results obtained with Boosted cascade when we only estimate four parameters \((c_x, c_y, s_x, s_y)\).

<table>
<thead>
<tr>
<th></th>
<th>Jaccard index</th>
<th>Dice index</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2C</td>
<td>76.1%</td>
<td>A2C 85.6%</td>
</tr>
<tr>
<td>A3C</td>
<td>70.7%</td>
<td>A3C 84.7%</td>
</tr>
<tr>
<td>A4C</td>
<td>76.0%</td>
<td>A4C 83.9%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>74.3%</strong></td>
<td><strong>Average 84.8%</strong></td>
</tr>
</tbody>
</table>

Table 3.3: Jaccard and Dice index in Boosted cascade with 4 parameters estimation \((c_x, c_y, s_x, s_y)\)

For estimating the rotation parameter we use the same trained Boosted cascade than before, but in the test we rotate the image for a set of chosen orientations and then we calculate the results for every orientation. This methodology is slower because we have to do N test for a single image, which means that in total we should do N*K test where N are the number of orientations tested and K the number of exams. Table 3.4 shows the performances for the five pose parameter estimation \((c_x, c_y, s_x, s_y, \theta)\).
Chapter 3. Results

### Table 3.4: Jaccard and Dice index in Cascade Boosted with 5 parameters estimation $(c_x, c_y, s_x, s_y, \theta)$

<table>
<thead>
<tr>
<th>Parameters Estimated</th>
<th>A2C Jaccard</th>
<th>A2C Dice</th>
<th>A3C Jaccard</th>
<th>A3C Dice</th>
<th>A4C Jaccard</th>
<th>A4C Dice</th>
<th>Average Jaccard</th>
<th>Average Dice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters estimated</td>
<td>76.9%</td>
<td>89.2%</td>
<td>75.9%</td>
<td>87.6%</td>
<td>76.7%</td>
<td>88.7%</td>
<td>76.5%</td>
<td>88.5%</td>
</tr>
</tbody>
</table>

After the obtained results, we can see that when we try to estimate the five parameters $(c_x, c_y, s_x, s_y, \theta)$ with Boosted cascade the results are only slightly better than when we only try to estimate the fourth $(c_x, c_y, s_x, s_y)$, and we take the original image which has posed rotation. The problem is that with five pose parameter estimation the computational time increases considerably.

Figures 3.9 and 3.10 present one example of LV pose estimation for A2C and A3C views using Cascade Boosted. On the left, we have the ultrasound image with the ground truth bounding box in red, and in green the proposed bounding box, on the right we have the corresponding probabilistic map. The two left images correspond to the same patient but on the top with four pose parameter estimation and on the bottom with five pose estimation. Table 3.5 summarizes the results obtained when we estimate four and five parameters. The bottom-right image in Figure 3.10 correspond to the probabilistic map in the first stage of the cascade however the upper-right image corresponds to the same probabilistic map but in the latest stage. If we compare the two probabilistic maps, we can see that in the first stage the classifier has only rejected few zones, while in the last stage the probabilistic map only contains two probable zones, and in these areas there are only few light points that correspond to the most probable center of the bounding box.

Figure 3.11 shows the same image of A4C view on the top with four pose parameter estimation and on the bottom with five pose estimation, on the left we have the image in the first cascade stage and on the right in the last stage of the cascade. If we compare them, we can see how the performances improve. In the first stage the bounding box is placed roughly at the proper localisation, but in the last stage the localisation is more precise. Contrasting the results obtained with and without rotation estimation we can see that the localisation is more accurate if we allow rotation, and this fact is reflected in the index of Jaccard. Last row at right of table 3.5 show the Jaccard index in the first stage and in the last stage for the two cases.

In general, the localisation is slightly more accurate in two chamber view than in the others view types.

<table>
<thead>
<tr>
<th>Parameters estimated</th>
<th>A2C</th>
<th>A3C</th>
<th>A4C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters estimated</td>
<td>$c_x, c_y, s_x, s_y, \theta$</td>
<td>$c_x, c_y, s_x, s_y, \theta$</td>
<td>$c_x, c_y, s_x, s_y, \theta$</td>
</tr>
<tr>
<td>Exam</td>
<td>E01</td>
<td>E05</td>
<td>E17</td>
</tr>
<tr>
<td>Cascade stage</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Jaccard index</td>
<td>78%</td>
<td>87.3%</td>
<td>78.8%</td>
</tr>
</tbody>
</table>

Table 3.5: Parameters for Cascade Boosted
Figure 3.9: A2C localisation. On the left, the image with ground truth (red) and the proposed bounding box (green), on the right the corresponding probabilistic map. On the top Boosted Cascade with four pose estimation and on the bottom with five pose estimation.

Figure 3.10: A3C localisation. On the left, the image with ground truth (red) and the proposed bounding box (green), on the right the corresponding probabilistic map. On the top Boosted Cascade with four pose estimation and on the bottom with five pose estimation.
3.2.3 Comparison

Comparing the two proposed approaches AdaBoost and Boosted Cascade, first of all we note that the optimal number of stages in the Boosted Cascade is only two, which means that the benefit of cascading multiple classifiers of increasing complexity and specificity is only marginal on our database. We consider that this result is mainly due to the limited number of images on which the classifiers are trained, thus we expect the Boosted Cascade approach much more convenient when scaling the database size up to thousands images.

If we compare the performances we can see that AdaBoost classifier outperforms Boosted Cascade results for all the types of views with a significant margin. The most suitable reason is that an intrinsic problem to cardiac ultrasound imaging is that the pixels outside the cone-shaped region do not contain signal, and the features extracted in these regions have a strong contrast that can bias the classification results. With the Boosted Cascade implementation used is not possible to handle this problem and for this reason the accuracy of the method decreases.

The results obtained with AdaBoost are encouraging and confirm that this technique can be successfully be used as a preliminary stage for the segmentation problem. We can see that its Jaccard index is above 85%, and almost all the points of the contour are contained in the bounding box.
3.3 LV Segmentation

We present here a quantitative analysis of the obtained segmentations, on the two databases the original one and the augmented one. Each predicted shape is compared to the ground truth contour using the Hausdorff $d_H$ and the mean Euclidean distance $d_E$ that can be defined as follows:

$$d_H(X,Y) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a,b), \sup_{b \in B} \inf_{a \in A} d(a,b) \right\}$$

(3.2)

$$d_E(X,Y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

(3.3)

As in the previous validation, we perform a leave one out cross validation to calculate the performances of the segmentation algorithm.

Before starting with the validation is important to note that thanks to the image registration algorithm and to the knowledge of the center of the cavity, if we place the mean shape in the centroid of the image the average distance between the ground truth and the mean shape is about 9 pixels. It means that if the pose estimation provided by AdaBoost is accurate the corresponding mean shape constitutes a good initialization, and the segmentation algorithm is able to ”deform” this shape to obtain a good segmentation. However, as we have seen in previous section the winner bounding box is not always perfect, and in this case, the segmentation algorithm should not only refine the LV shape but also the pose.

We calculate the performances in terms of Hausdorff distance and mean Euclidean distance for the database of 2 chamber apical view, which contains 42 images that correspond to 14 patients. In order to have an idea of the quality of the performances, we calculate the distances for the estimated shape and for the mean shape. Table 3.6 summarizes the distances between the ground truth and the estimate, and also the ground truth with the mean shape in training and in test.

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th></th>
<th>Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate contour</td>
<td>MED</td>
<td>HD</td>
<td>MED</td>
<td>HD</td>
</tr>
<tr>
<td></td>
<td>5.87</td>
<td>6.27</td>
<td>19.92</td>
<td>33.27</td>
</tr>
<tr>
<td>Mean Shape</td>
<td>6.31</td>
<td>7.98</td>
<td>20.22</td>
<td>35.82</td>
</tr>
</tbody>
</table>

Table 3.6: Hausdorff distance (HD) and mean Euclidean distance (MED) in training and in test

Figure 3.12 shows an example of Endocardium segmentation with the ground truth in green, the mean shape in yellow and the predicted shape in pink. We can see that the predicted shape compared to the mean shape is deformed in order to come near the ground-truth, however the result obtained is not enough accurate.
Figure 3.12: Segmentation example, in green the ground truth, in yellow the mean shape and in pink the predicted shape.

After the results, we can see that due to the small size of the training database, the results with our method are slightly better than only using the mean shape initialization. Indeed, even after dimension reduction with PCA the matrix to be estimated is $4 \times 12$, but we only have 13 patients for training. Furthermore, there is significant intra-class variability in this dataset, which makes the training even more challenging.
Chapter 4
Conclusion and Future Work

The work described in this thesis is focused on the development of an image segmentation algorithm for 2D-echocardiography within a supervised learning framework. The approach is articulated in two steps: the first one addresses the estimation of the left ventricle pose using a Boosting algorithm, whereas the second uses this pose information for learning the segmentation of the Epicardium and Endocardium. We carried out the experimental evaluation of the AdaBoost pose estimation algorithm, which showed robust performances on a database of apical views, and a feasibility study of our supervised segmentation technique, based on regression, whose preliminary results are promising although further validation on larger datasets will be needed.

4.1 Thesis summary

In Chapter 1, we have first presented a general introduction of the subject and clinical context of decision support systems for cardiology. With medical imaging playing an increasingly prominent role in the diagnosis and treatment of disease, the problem of extracting clinically useful information has become crucial. For example, the automatic delineation of the left ventricle helps to assist physicians in various states of treatment of cardiovascular diseases performing quantitative or qualitative analysis. We also provided a review of the state-of-the-art algorithms for recognition in cardiac ultrasound images. So as for the pose estimation task, we have distinguished three categories of approaches, based respectively on: (a) support vector machines, (b) tracking, and (c) boosting techniques, whereas for the segmentation task we discussed model-based approaches and pattern recognition techniques.

In Chapter 2, we have provided a detailed description of the proposed pose estimation and segmentation method. The pose estimation model consists of a boosting classifier with Haar-like features that determines the most suitable bounding box that contains the LV. Moreover, we have proposed a cascade regression framework based on a least square algorithm addressing for the segmentation task.

Chapter 3 provided experimental results and quantitative assessment to validate the performance of the new algorithm. We tested the localisation algorithm for the Boosted cascade and AdaBoost approaches and we provided the performance analysis in term of Jaccard and Dice indices according to leave-one-out cross validation.
The segmentation was performed by taking the output information from the localisation algorithm as an input for the cascade regressor. The performances for the LV segmentation were evaluated in terms of mean Euclidean distance and Hausdorff distance on a database containing all the frames of each sequence. Experimental results on real images from the databases were shown and evaluated in Chapter 3.

4.2 Future Work

Although the results presented here have demonstrated the effectiveness of the pose estimation and segmentation approach, it could be further developed in a number of ways:

4.2.1 View classification problem

Considering the future availability of a larger database, containing up to thousands images, training the classifier for pose localization will result in more robust performances, while enabling the use of the same approach for the viewtype recognition task, as presented in [40], [30], [32], [41]. The main idea is to train an AdaBoost classifier for every type of view (A2C, A3C and A4C) and then to decide the winning class according to the output classification score. Thus, the classifier with the highest score will decide for the predicted class. We can also envisage a more complex, hierarchical learning framework of AdaBoost classifiers whether dealing with more categories is needed. In this case, one could first train an AdaBoost classifier in order to detect the type of windows (apical, parasternal, subparasternal and subcostal), and then for apical and parasternal one could train another AdaBoost classifiers to classify the sequence into one of the fifth final views (PLAX/PSAX or A2C/A3C/A4C).

Another possible approach for viewtype recognition would be to use template matching, i.e., finding areas of an image that match to a learned template image. We would have tree template images at least, one for every type of view, and we would find the similarities between the source image and the template image. These similarities are windows extracted from the entire that match the template; in our case the small regions could be the other cavities, which are unique for every type of view.

4.2.2 Rotation-invariance for the boosted cascade with a Pyramid detector

In order to improve the results obtained for the five-parameters pose estimation in Boosted cascade we should increase the complexity of the cascade by using a Pyramid detector as was proposed by Li et al. [23] for face detection. The pyramid detector consists of 3 levels, the first level of the pyramid works on faces at all poses; the second level detects faces between $-90^\circ$ and $-30^\circ$, between $-30^\circ$ and $30^\circ$, and between $30^\circ$ and $90^\circ$, respectively; the third level detects faces at 7 finer angles. Once a test window passes one level of the detector, it will be passed to all the children nodes for further decision. The main idea would be to use and adapt this detector four our cavity detection.
4.2.3 Multi-scale classifier

For the pose estimation purpose we have optimised all the parameters in order to obtain the most accurate results. With the aim of improving the current performances we have started to work with a multi-scale framework. The main idea is to train a first AdaBoost classifier as the current one to delineate a zone that should contain the left ventricle and then split the bounding box into four zones, and for everyone train a new AdaBoost classifier with size $14 \times 24$ to refine the localisation. This new approach was tested for few examples and the accuracy slightly increased more extensive validation is needed in order to find the optimal parameter settings.

4.2.4 Multi-scale Gabor features

With the aim of improving the performances of segmentation algorithms we could use cascade regression with multi-scale Gabor features, so as to deal with the large pose variations early stages by using Gabor features extracted at coarser scales, then refining the detected shape at later stages using Gabor filters at finer scales.

We observed that the performances of the cascade regression decreases when the initialisation provided by the localisation algorithm is not accurate enough. To handle this problem, one could generate multiple hypotheses in test and combine them to get the final result as proposed in [38] and [16].

4.2.5 Multivariable regression

The proposed approach for segmentation is based on multivariable regression. In literature we can find many techniques that solve multivariable regression, finally we decided to use the simple linear least-squares model for our feasibility study. However there are some limitations of this approach and for future work it would be more suitable to use more complex vectorial regression techniques, such as the popular Random Forests.

4.3 Personal conclusion

The past months of my internship at Philips Medisys have been very instructive and invaluable for me, in both, from both the scientific and personal point of view.

From the scientific viewpoint, I was part of an excellent team that gave me the opportunity to put into practice the skills acquired throughout my Master in Medical Imagery. To be involved in a real application project made me feel excited and in a certain way nervous about being able to perform in a satisfactory way. During this internship I gained a lot of experience that will be very useful for my PhD.

Also, after this professional experience, I realized that one of the main points to succeed in completing the tasks is organization in a large sense. From the very beginning of my internship I established a calendar with all the documents (and deadlines) to be sent to Telecom Bretagne. The figure 4.1 shown below is a graphic representation of the main dates and deliverables that I had to meet.
A second extremely important aspect when working in a professional environment is communication. During my internship I had plenty of follow up meetings with my supervisor in order to discuss about the status of the project. These meetings were really helpful because they allowed me to discuss with my supervisor about my motivations and my doubts, and he always helped me to continue with my project. Moreover, we had monthly a review meeting with all the research team, during which I had the opportunity to give a presentation of my ongoing work, and receive significant feedbacks about the most challenging points of my project. These meetings were very important for me because it was the moment where I learnt about the work of my co-workers, and sometimes some ideas used by one trainee were useful for my project, also I had advice from other researchers.

From the personal viewpoint, this internship has offered me a research whole approach, from understanding the clinical problematic, establishing the state of the art and the possible methods to implement and validate a method. Moreover, the fact of doing my internship in a foreign country has helped me a lot improve my language skills.

I strongly believe that Philips Medisys is a great place to work and it has been with no doubt the best place to do my final degree internship.
Bibliography


48


[38] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Learn to combine multiple hypotheses for accurate face alignment. In Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, pages 392–396. IEEE, 2013. 10, 11, 24, 25, 46


