

Master's Thesis

Database guided detection of anatomical landmark points in 3D images of the heart

Thesis Committee:

Prof.Dr. M. J.T. Reinders
Dr. D.M.J. Tax
Dr E. Hendriks
Dr J. G. Bosch
ir. K.Y.E. Leung

Name	Thomas Karavides
Email	t.karavides@gmail.com
Student Number	1366475
Thesis Supervisors	Dr E. Hendriks Dr J. G. Bosch ir. K.Y.E. Leung
Date	14 th October 2009



Preface

Automated landmark detection may prove important for the examination and automatic analysis of real-time three-dimensional (3D) echocardiograms. By detecting 3D anatomical landmark points, the standard anatomical views can be extracted automatically in 3D ultrasound images of left ventricle, for better standardization and objective diagnosis. Furthermore, the landmarks can serve as an initialization for other analysis methods, such as segmentation. In this thesis we describe an algorithm that iteratively applies landmark detection in perpendicular planes of the 3D dataset. The landmark detection exploits a large database of expert annotated images, using an extensive set of Haar wavelet-like features for classification, resulting in fast detection times suitable for real-time applications. The detection is performed using two cascades of Adaboost classifiers, that work in different 2D planes, in a coarse to fine scheme. The method is evaluated by measuring the total detection error for the landmark points between the detected positions and the manual ones.

Acknowledgements

I would like to express my appreciation to those who have helped me during the development of this project. For his permanent confidence, advice and support, I would like to thank my supervisors at Erasmus MC Dr. J. G. Bosch and ir. K.Y.E. Leung. This learning process has been richer and broader everyday thanks to their constant interest. To my TU Delft supervisor, Dr E. Hendriks, all my gratitude for his patience and for providing a fresh perspective to the research project. Many thanks to G. van Burken, who was always available and willing to help me with any hardware or data processing related problem, and to P. Paclik for his help in understanding complex classification problems that appeared. I would also like to express my appreciation to everyone at the Erasmus MC Thoraxcenter for providing me with a perfect working environment. Finally, I am very grateful to my fellow classmates at ICT group for keeping me company and helping me improve in my presentational skills.

Contents

1. Introduction.....	1
1.1 The Heart.....	1
1.2 Ultrasound Medical Imaging.....	1
1.3 Echocardiography.....	3
1.4 Study Goal.....	5
2. Landmark Detection in Ultrasound Images.....	7
2.1 General.....	7
2.1.1 Registration.....	7
2.1.2 Segmentation.....	8
2.1.3 Feature Detection.....	9
2.2 Classification Methods.....	11
3. Database Guided Landmark Detection.....	15
3.1 Iterative Scheme.....	15
3.2 Landmark Detection.....	17
3.2.1 Features.....	17
3.2.2 Classifier.....	21
3.2.3 Cascade.....	23
3.2.4 Marginal Space Learning.....	25

4. Experimental Setup	28
4.1 Data Representation	28
4.2 Preprocessing	29
4.3 Experimental Setup	29
4.3.1 Training	29
4.3.2 Tuning the Classifier	31
4.3.3 Detecting	32
5. Evaluation	33
5.1 Two Chamber View Plane	33
5.2 Four Chamber View Plane	38
6. Discussion	42
6.1 Multilevel Approach	42
6.2 Choice of Classifier Parameters	43
6.3 Comparison with Literature	44
6.4 Limitations	45
6.5 Conclusions	46
7. Future Research	48

1 Introduction

1.1 The heart

The heart is the muscular organ of the circulatory system that constantly pumps blood throughout the body (Figure 1.1). Approximately the size of a clenched fist, the heart is composed of cardiac muscle tissue that is very strong and able to contract and relax rhythmically throughout a person's lifetime. The heart has four separate compartments or chambers. The upper chamber on each side of the heart, which is called the atrium, receives and collects the blood coming to the heart. The atrium then delivers blood to the powerful lower chamber, called the ventricle, which pumps blood away from the heart through powerful, rhythmic contractions. The human heart is actually two pumps in one. The right side receives oxygen-poor blood from the various regions of the body and delivers it to the lungs. In the lungs, oxygen is absorbed in the blood. The left side of the heart receives the oxygen-rich blood from the lungs and delivers it to the rest of the body. In this thesis, we will focus on the left ventricle of the heart and its anatomical landmarks.

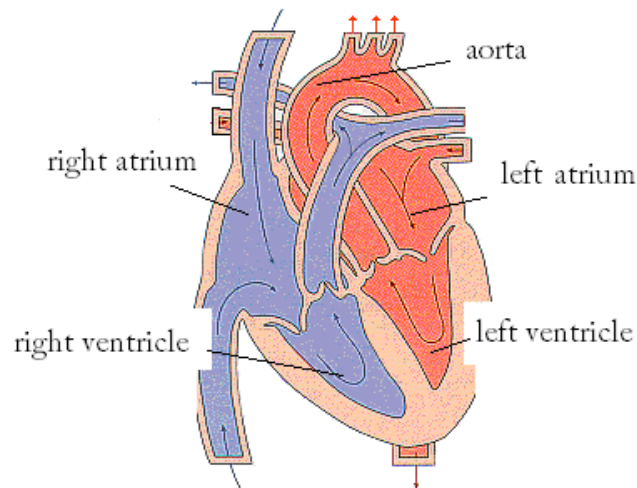


Figure 1.1 The main anatomical parts of the human heart

1.2 Ultrasound medical imaging

Ultrasound (sound in the MHz frequency range) has been used for many decades now for medical imaging. The technology is relatively inexpensive and portable, especially when compared with modalities such as magnetic resonance imaging (MRI) and computed tomography (CT). Some of the success and popularity of acoustic imaging is closely related to its safe and risk-free nature. In addition to being completely noninvasive, the biologic effects of ultrasound, as used in routine clinical applications, pose minimal risks for the patients. Medical sonography has been used in cardiology, endocrinology, gastroenterology, gynecology, obstetrics, ophthalmology and urology, among other fields, without documentations of serious adverse events.

The biologic effects of ultrasound depend on the total energy applied on a specific region and they are related to the production of heat and cavitation effects. With pulsed ultrasound, the relatively short periods of pulsing makes excessive heat generation unlikely[1]. A metric associated with the tissue heating bio-effect is given by the thermal index, which is the ratio of the power used to that required to cause a maximum temperature increase of 1°C. The second physical effect of ultrasound is cavitation. This term refers to the formation and behavior of gas bubbles, produced when high intensity ultrasound penetrates into tissue. These bubbles can implode and inadvertently destroy biological tissue. To prevent this phenomenon, it is made sure that the mechanical index, defined by the peak negative pressure and the center frequency, stays below a certain threshold.

The instrument used to create an ultrasound image is called an echograph. It has all the necessary electronics needed to transmit, receive, process and interpret ultrasound pulses and generate an image. It uses a so called probe or transducer to generate and receive ultrasound waves. The echograph determines how long it took the echo to be received from the time the sound was transmitted and from this deduces the image depth. How strong the echo was determines in what intensity level each pixel will light up on the screen.



Figure 1.2 An ultrasound probe

The use of ultrasound for imaging became practical with the technological evolution of piezoelectric transducers. Piezoelectric materials have the ability to produce electric voltage in response to applied mechanical stress. The effect is reversible and results in deformation when a voltage is applied to the material. In this way, these materials can generate ultrasound waves, and convert the reflected sound energy to an electric impulse.

An ultrasound probe mainly consists of an array of piezoelectric ceramic elements (Figure 1.2). The frequency of the ultrasound beam depends on their thickness, while their shape determines the beam shape. Other important components of the probe are the backing material and the lens. The backing material shortens the ringing response of the piezoelectric material after the brief electronic impulse. For most clinical applications, the ultrasound beam is both focused and steered electronically, through the use of phased-

array transducers. The wave front of the beam consists of the sum of the individual wavelets produced by each element. Manipulating the time of excitation of each element results in both steering and focusing of the beam [1].

1.3 Echocardiography

An echocardiogram is an ultrasound image of the heart. Also known as cardiac ultrasound, echocardiography uses ultrasound techniques to enable analysis of the heart. Echocardiography is a significant tool in the hands of physicians. By measuring the heart volume the cardiologist can tell a lot about the patient's condition. For example, left ventricular ejection fraction (the fraction of blood pumped out of a ventricle with each heart beat) and end-diastolic and end-systolic volumes (maximal and minimal volume per beat respectively) serve as a predictor for long-term survival of patients with for example congestive heart disease or dilated cardiomyopathy. Most attention has been given to detecting the endocardium (blood pool/tissue border) to allow for estimation of left ventricular volumes and derived measures such as the ejection fraction, and for regional wall motion assessment. In particular, these measures are used in diagnosis and assessment of ischemic heart disease.

The majority of echocardiographic examinations are performed via the transthoracic route. In this case, the echocardiography probe is placed on the thorax of the subject, and images are taken through the chest wall. The views used in transthoracic echocardiography are the parasternal, the apical and the subcostal view, corresponding with different positions of the ultrasound probe on the patients chest [1],[2]. Slight turns of the transducer give us further subdivisions of these main views like the apical two and four chamber view (Figure 1.3).

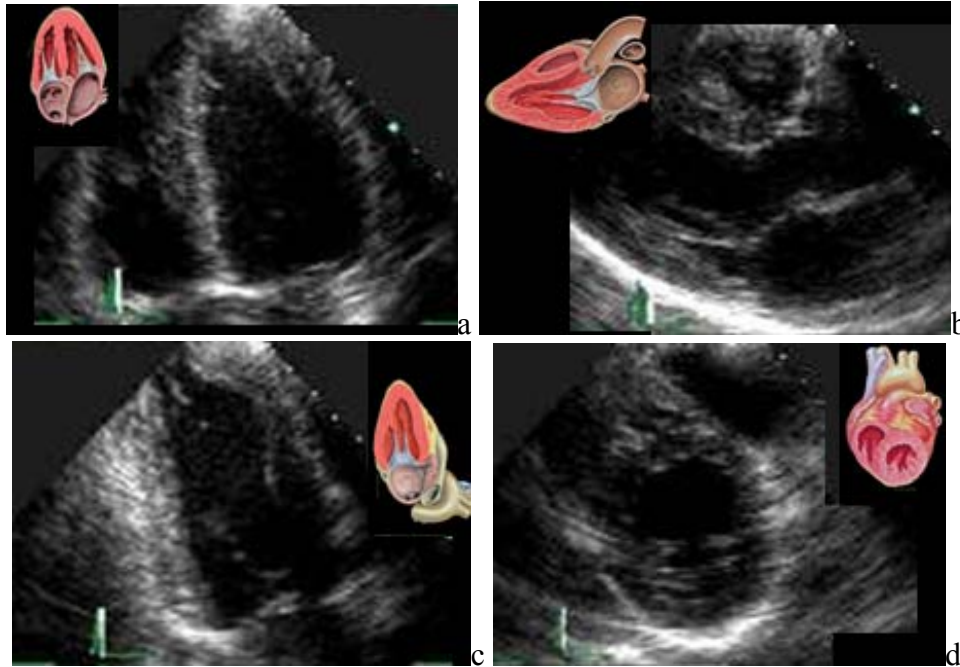


Figure 1.3 Different views of the heart. a) the four chamber view, b) parasternal long axis view, c) the two chamber view and d) the parasternal short axis view.

As already mentioned, one of the most important applications of ultrasound imaging is measurement of the heart volume and specifically of the left ventricle. The latter is traditionally calculated by delineating the endocardium in two perpendicular 2D views, such as the four and two chamber, and then applying Simpson's rule [3]. Evidently, the results may vary a lot as assumptions are needed with regards to the geometry of the left ventricle. Also, apical foreshortening may occur due to imperfections in image plane positioning (the 2D images do not transect the true left ventricular apex). 3D echocardiography may overcome these drawbacks. The eventual goal is real time, 3D display of cardiac function.

Realization of 3-D echocardiography requires facilities for the acquisition of the data set and the display of the three dimensional images. There have been two approaches concerning the acquisition of the images. The first one is based on acquiring multiple series of 2-D images of different planes, storing them and finally processing them in order to construct the 3-D images. The second method is volumetric imaging. A transducer is constructed with a two dimensional array of crystals, capable of collecting pyramidal images. For displaying the image there are also two main approaches: the whole 3-D image can be rendered as a volume, or 2-D slices of the volume are displayed, giving the opportunity of an inside look at any plane desired [4-8] (Figure 1.4).

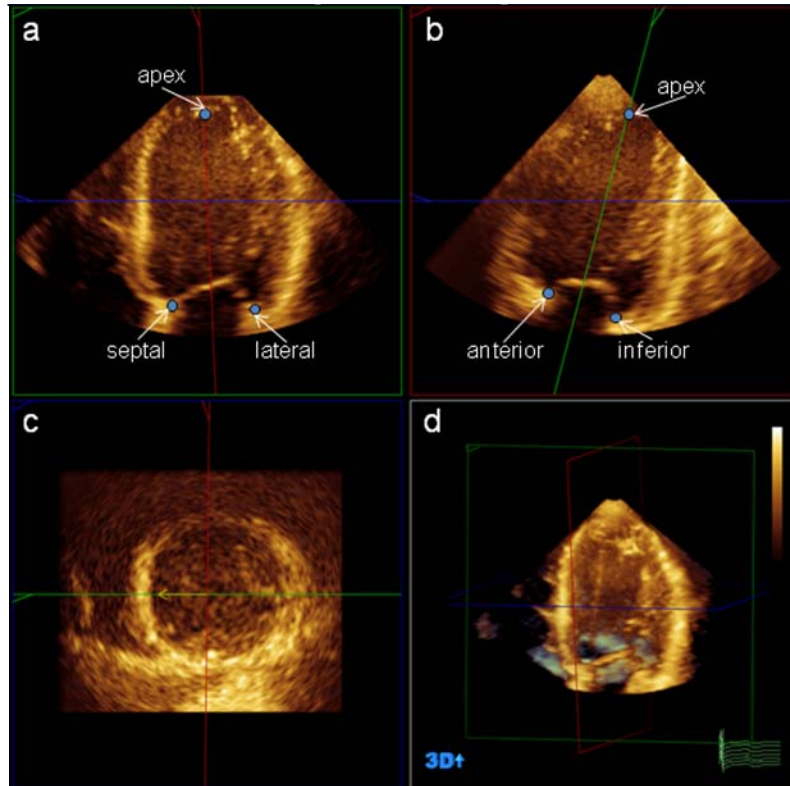


Figure 1.4 Example of a 3D image and landmarks. (a) 2D four-chamber view, (b) two-chamber view (c) short axis view, and (d) 3D rendered volume.

1.4 Study goal

The goal of this master thesis project is to detect the 3D landmark points that define the position of the left ventricle (LV). This has proved to be challenging due to large amount of noise and signal drop-out that is common in ultrasound images. Additionally, echocardiography suffers from large image variation depending on the acquisition procedure. Factors as patient breathing, probe position or the anatomical differences among patients, give a large variety of images that make standardization difficult. The anatomical landmark points, however, can be used to extract the standard anatomical views in apically acquired 3D ultrasound images of LV, for better standardization and objective diagnosis. Furthermore, automatic detection of these landmarks would serve as an initialization for fully automated 3D analysis methods, such as segmentation.

The anatomical views are defined based on the position of the LV long axis, which is defined as the line through the LV apex and the centre of the mitral valve (Figure 1.5). The correct orientation of the four-chamber (4C) view is defined as the largest cross-section through the right ventricle (RV), which can be found by detecting the attachment points of the RV. The three-chamber (3C) view is found by locating the aortic outflow tract (AO, Figure 1.5). The current project will focus on finding these landmarks.

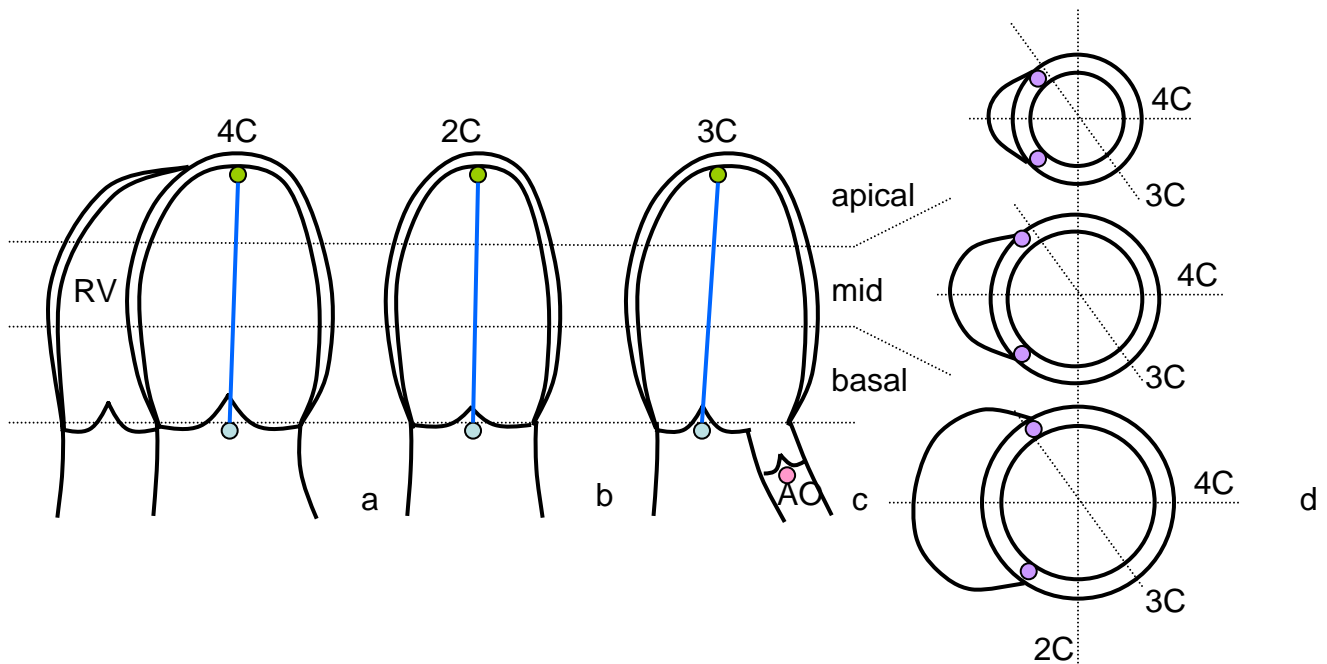


Figure 1.5 Diagram of (a) four chamber (4C), (b) two chamber (2C), which is defined as the perpendicular to the 4C, (c) three chamber (3C) long axis views, and (d) basal, mid and apical short axis views.

Since these standard anatomical views are required for simple visualization and initialization purposes, the detection method needs to be relatively fast. Also, because of the challenging image quality and the variation in LV appearance, robust methods are required. Additionally, special attention should be paid to preprocessing the images, to reduce speckle noise. So far, from our experience, we found that landmarks like the attachment points of the RV and the LV apex are difficult to define and to detect. More salient structures, like the AO and the mitral valve annulus, may be more reliable. Therefore, the method should let the more salient structures guide the segmentation, combining prior knowledge on these anatomies. This may suggest a multilevel approach, in which the global locations of the landmarks are found first, and then refined.

2 Landmark Detection in Ultrasound Images

2.1 General

Object detection in ultrasound images is a task of great interest in medical diagnostic procedures. Detecting heart volumes, fetus or the prostate are only some applications of object detection algorithms. In clinical practice, physicians mentally integrate information from different images acquired from a patient, often with different imaging modalities. Images are shown in various orientations and positions and at different scales. Semi-interactive detection methods rely on the expert's ability to interactively select corresponding slices using anatomical knowledge. Current research is mostly focused on making these methods fully automatic and thus operator independent. To do so, registration, segmentation and feature detection have been proposed. In this chapter we briefly discuss these topics. However, it is important to keep in mind that specific methods may use combinations of these techniques.

Data sets acquired by imaging a scene or object at different times, or from different perspectives, or using a different physical modality, will result in images that are sampled in different coordinate systems. Image *registration* is the process of transforming two or more images from one or more physical modalities into one coordinate system. In image processing, *segmentation* refers to the process of partitioning an image into several regions. The goal of this process is to divide the image in different sections and it results either into a set of regions that collectively cover the whole image or into a set of contours extracted from the image. Image segmentation is a procedure typically used for object and boundary detection [9]. In medical imaging methods of this kind are used for detecting tumors, tissue wall computation, study of anatomical structure and other diagnostic purposes. Finally, in computer vision and image processing, the concept of *feature detection* refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions. There is no universal or exact definition of what constitutes a feature, and the exact definition often depends on the problem or the type of application.

2.1.1 Registration

Image registration is a significant asset to object detection algorithms. By aligning an image which contains information about an object, to a second image, information can be projected. We can group the different registration techniques by the type of deformation they want to estimate. Rigid transformations can be described by a simple parametric transformation consisting of translation and rotation. An affine model, on the other hand, includes translation, rotation, scaling, and shear. If the transformation maps parallel lines onto parallel lines it is called affine. If it maps lines onto lines, it is called projective. Finally, if it maps lines onto curves, it is called non-rigid or elastic[10]. Non-rigid transformations use polynomial warping, interpolation of smooth basis functions (thin-plate splines), and physical continuum models to represent a smooth deformation field

[9]. Non-rigid registration of medical images can also be used to register a subject's data to an anatomical atlas. Large datasets of images can be mapped into a common coordinate system. That way we can study inter-population differences or intra-population variability.

A transformation is called global if it applies to the entire image, and local if subsections of the image each have their own transformations defined. The type of registration to use depends on the application of interest. So, for intrasubject differences focus is placed on small changes and global transformations are applied. An example is registration of rest to stress cardiac images [11, 12]. On the other hand when large changes are required, like for example monitoring tumor growth [13], local transformations are necessary. For intersubject and patient-to-model registration the appropriate transformation must be able to capture large anatomical differences.

Apart from the type of transformation, the main components of registration are the image similarity measure to use, the optimizer and the interpolator. Interpolation is required when an image needs to be translated, rotated, scaled, warped, or otherwise deformed before it can match a reference image or an atlas [14]. The choice of the image similarity measure depends on the nature of the images to be registered. Common examples of image similarity measures include cross-correlation, sum of square differences and ratio image uniformity. These are commonly used for registration of images of the same modality [9]. For images of different modalities, the mutual information measure is commonly used [11]. The metric of choice has to be optimized, i.e. maximized or minimized depending on the metric, during registration. Optimization algorithms commonly used are the simplex method, simulated annealing, gradient descent etc. [15].

2.1.2 Segmentation

Ultrasound image segmentation is strongly influenced by the quality of the data. Speckle and attenuation are some of the artifacts typical for ultrasound images that make the task of contour detection in such images a difficult task. However, as transducer technologies evolved and preprocessing algorithms were introduced, the image quality improved significantly, resulting in more possibilities for ultrasound image segmentation, not only in 2-D but in 3-D as well [16]. Over the years, medical image segmentation techniques have become more sophisticated, replacing methods like region growing, thresholding and watershed algorithms with deformable models, level sets and statistical models.

In echocardiography segmentation methods are mainly used for border and volume detection, mostly of the left or the right ventricle. The most popular approaches have been those that use deformable models (snakes) to detect the endocardium border. A snake can be viewed as a curve that deforms under the influence of external and internal forces. Internal forces are determined by the snake's elasticity and curvature, while external forces are derived from image properties [17].

An evolved version of active contour border detection is level set based methods. Instead of manipulating the contour directly, the contour is embedded as the zero level set of a

higher dimension function ψ called the level set function. The level set function is then controlled through a differential equation (Figure 2.1). At any time during evolution, the contour can be obtained by extracting the zero level set from the output. The level set method makes it very easy to follow shapes that change topology, for example when a shape splits in two, develops holes, or the reverse of these operations. Level sets have been extensively used in medical imaging [18], [19], as it is easily adaptable by using image based features, like image intensities, gradients etc. in the governing differential equation.

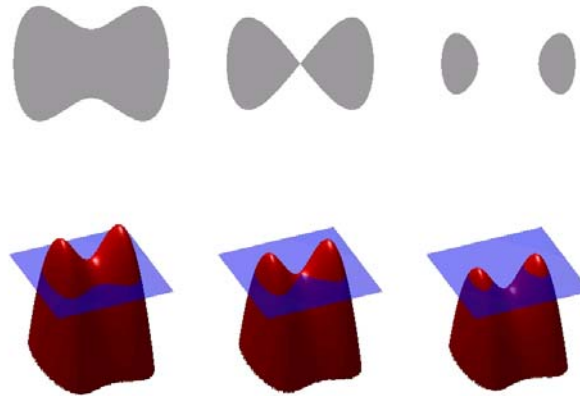


Figure 2.1 In the upper-left corner we see a shape, that is, a bounded region with a well-behaved boundary. Below it, the red surface is the graph of a level set function ψ determining this shape, and the flat blue region represents the z plane. The shape itself is the set of points on the surface for which the function is positive or zero.

Other important segmentation methods are active shape models and active appearance models. These are model-based approaches that integrate expert knowledge with computer vision by constructing statistical models of the object. The models, built using Principal Component Analysis (PCA), can be used to generate new realistic instances of the object, which can be matched to new images that are not yet interpreted. To synthesize a complete image of an object or structure, shape and texture (i.e. the intensity in the region of the object) can be modeled [20]. Active shape models take only the statistical shape variation into account, whereas active appearance models combine the shape and texture model into one appearance model, to describe the complete image.

2.13 Feature detection

A feature is defined as an interesting part of an image. Edges, blobs, specific points can all be considered as features. Feature detection is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features. Many image processing algorithms use feature detection as the initial step, so as a result, a very large number of feature detectors have been

developed. These vary widely in the types of feature detected, the computational complexity and the repeatability. Repeatability is strongly related to whether the selected features are invariant to image scaling, rotation, and to change in illumination between the different images.

One feature detection technique widely used in image processing has been proposed by Lowe [21]. In this work a method is introduced to extract oriented keypoint features that can be used for further object identification (Figure 2.2). The approach has been named Scale Invariant Feature Transform (SIFT), as it transforms image data into scale-invariant coordinates relative to local features.

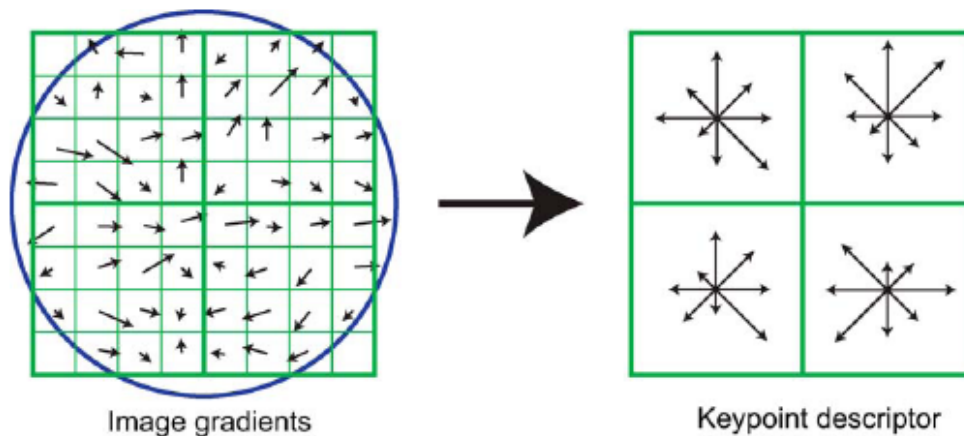


Figure 2.2 A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the key point location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub-regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples [21].

A frequently used method in ultrasound is based on the Hough transform (HT). The HT converts the problem of extracting collinear point sets in the image space into maxima detection in the Hough space. Although the original formulation allows the detection of lines and circles, the method has been extended to extract other parametric curves [22]. So, using the HT it is possible to detect approximate circular structures, such as the aorta and the endocardium in short-axis slices, in an edge-enhanced image. The algorithm has been used several times for border detection in ultrasound and MRI images [23, 24], [25].

As an alternative to the HT for contour initialization in image segmentation, the Fourier Mellin Transform (FMT) [26] is proposed by the authors of [27]. The main idea behind this approach is to build an appearance template of the object of interest, and to express both the template and the search image in a rotation and scale invariant representation using the FMT (Figure 2.3). This way, the estimation of rotation, scale and position converts into a cascade of linear shift detections, enabling a fast estimate of the object pose, and the subsequent generation of an initial contour from the template.

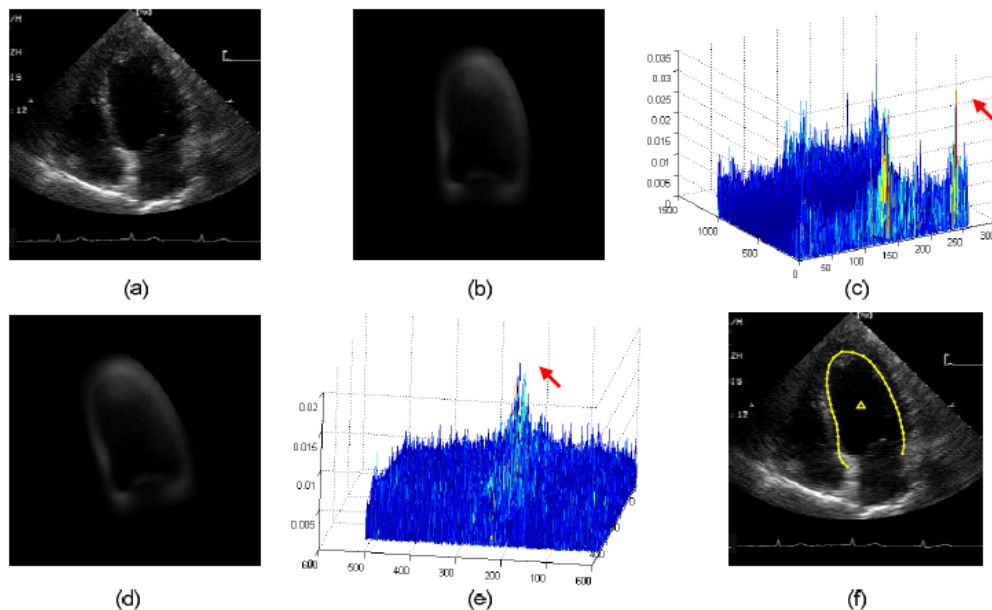


Figure 2.3 (a) 4 chamber view echocardiogram (ED phase). (b) Template image (c). The peak matrix after matching a and b; the location of the highest peak (indicated by the arrow) indicates the orientation and scaling factor. (d) Based on the scale and orientation detected in (c), the template is rotated and scaled to match the search image (e). By matching (a) and (d), the position of the left ventricle is detected (indicated by the arrow). (f) An average contour is placed on the detected location serving as an initialization of a segmentation algorithm [27].

The term classification is used to underline the process of implicitly encoding the prior knowledge embedded in expert annotated databases. Classification has been proven to solve ultrasound problems similar to ours. These techniques use a database of training images to train a classifier which can distinguish between different classes of objects (e.g. apples and pears). Objects are represented by their features, as described in section 2.1.2. Usually, the classifier automatically selects from a large set of features computed at different image scales, some relevant ones. Recently, an interesting classification approach has been proposed by Georgescu et al. [28]. In the following section, we will discuss this approach in more detail.

2.2 Classification methods

Georgescu et al. [28] introduce a classification method, which they named database-guided paradigm, that directly exploits expert annotation of structures of interest in large medical databases. They select the features using a boosted cascade of weak-classifiers. This is used to detect the possible locations of the object of interest. A similar approach is used by the authors of [29] for the detection and measurement of fetal structures in ultrasound images.

To do so the image is considered to consist of two parts, a foreground S which contains the sought object and a background region B. The foreground image S is determined by the following vector

$$\theta = (x, y, \alpha, \sigma_x, \sigma_y) \quad 2.1$$

where the parameters (x, y) represent the top left region position in the image, α denotes orientation, and (σ_x, σ_y) the region scale (Figure 2.4). In the training phase, the features within this foreground box are used to build the classifier. In the detection phase, the idea is to find the position of the foreground box in a new image, by evaluating the image features at candidate positions of the foreground box.

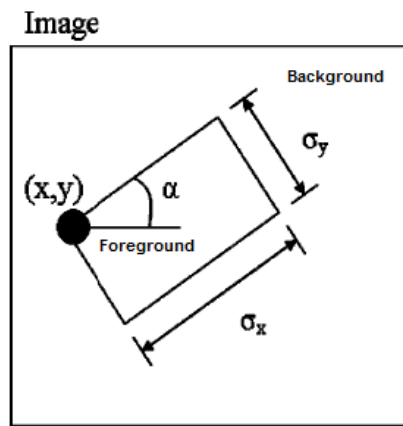


Figure 2.4 Foreground (rectangular) image region with five parameters [29].

The appearance of the image region is represented with Haar-like features, which are derived from wavelets. A simple rectangular Haar-like feature is defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. These features can be calculated very quickly. Haar wavelets have been successfully used for face detection [30] as well as for left ventricle detection in ultrasound images [28].

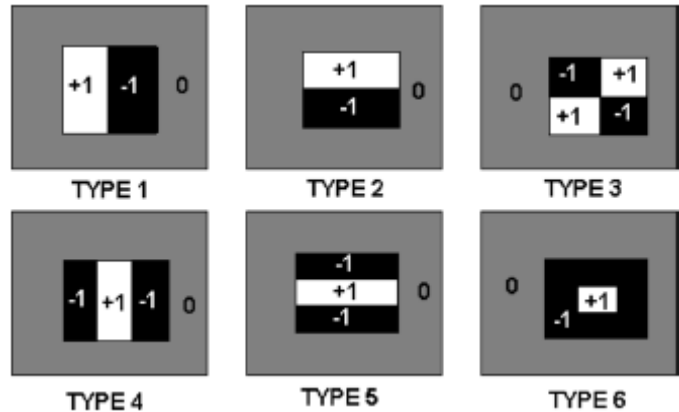


Figure 2.5 Haar image feature types used. The feature value is the sum of the pixels in the white region minus the sum of the pixels in the black region. Notice that the gray area represents the foreground region S [29].

As we already mentioned the method depends on a classifier, based on the idea that one can combine a number of weak classifiers and, by giving them appropriate weights, construct a strong one. The classifier is a probabilistic boosting tree that is able to cluster the data automatically, allowing for binary classification of data sets. The algorithm recursively learns a classification tree. At each node, a strong classifier is learned using a boosting algorithm which automatically learns a strong classifier by combining weak classifiers. The algorithm selects the weights by minimizing the probability of error.

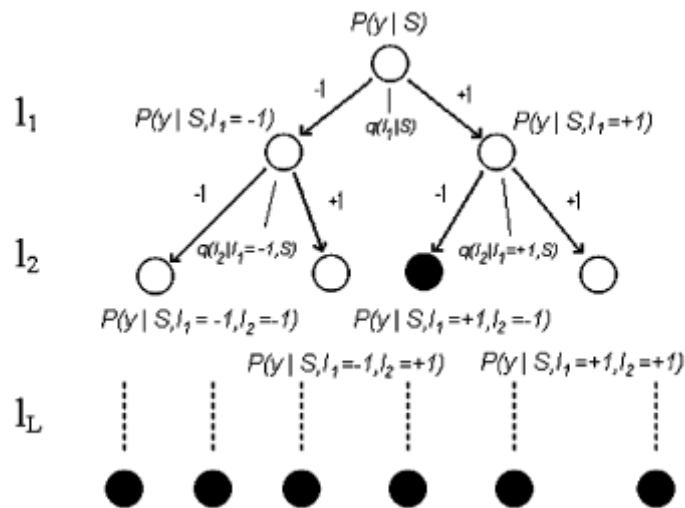


Figure 2.6 Illustration of the probabilistic boosting tree. The dark nodes are the leaf nodes. Each level of the tree corresponds to an augmented variable. Each tree node is a strong classifier. The classifier computes the probability $P(y | S)$ where $y \in \{-1, 1\}$ with $P(y = -1 | S)$ representing the probability that image region S contains background information and $P(y = 1 | S)$ that it contains the sought structure [29].

The original probabilistic boosting tree algorithm may suffer from overfitting of the training data in the nodes close to the leaves (Figure 2.6), as well as in long training and detection procedures. The authors of [29] suggest a two part solution to these problems. To simplify the training and detection procedures they divide the parameter space into subspaces. In addition, they set a limit to the allowed growth of the tree by limiting the height and number of nodes. They train three different classifiers, where the classification problem grows in terms of complexity. The main difference between this approach and the cascade scheme is that the first stages are trained with a subset of the initial set of parameters instead of a subspace of the full parameter space. Given that the classification problem of each classifier is less complex than the original problem, the height and the number of tree nodes can be constrained.

The main disadvantage of this approach is that it may fail in terms of robustness if the database is not large enough. On the other hand it is a generic method, which makes it easy to expand it to detect other interesting structures as well (RV, 2C and 4C views, left ventricle long axis).

3 Database Guided Landmark Detection

3.1 Iterative scheme

After considering all our options, it seems that a database guided approach would be the most promising. It may give us a fast, robust and complete solution, since all landmarks could be detected at once. It has been previously demonstrated in ultrasound images [28, 29] and it exploits prior knowledge from a database, which was one of our goals. Rather than extending this method to 3D, we opt for staying close to the original 2D method and replacing our existing manual framework for landmark definition.

Previously, we proposed a manual method to find the LAX, in which perpendicular planes from 3D data set were annotated iteratively [31, 32]. By annotating the apex and mitral valve centers in perpendicular long-axis planes, the 3D LAX can be found easily (Figure 3.1, Figure 3.2). In this study we propose to replace the manual annotation in 2D cross-sections using the fully automated landmark detection. This can be achieved by iteratively applying a database-guided approach similar to the one described in [29] for detecting 2D landmark points in fetal structures. In addition, the orientation of the 4C, 2C, 3C can be found by matching 2D short-axis views on several levels, yielding the rotation, which can then be used to update the LAX views again. The 2D templates for the database can be generated simply by taking differently oriented 2D slices of the available 3D images with manually defined LV surfaces. The method will need a few iterations before it can estimate an accurate LAX position, but it is still expected to reach a solution faster than a Hough transform approach for example. Similar to [31, 32] the method will be applied to single 3D images at end-diastole (ED) only, although use of temporal features from the complete cycle may also be possible.

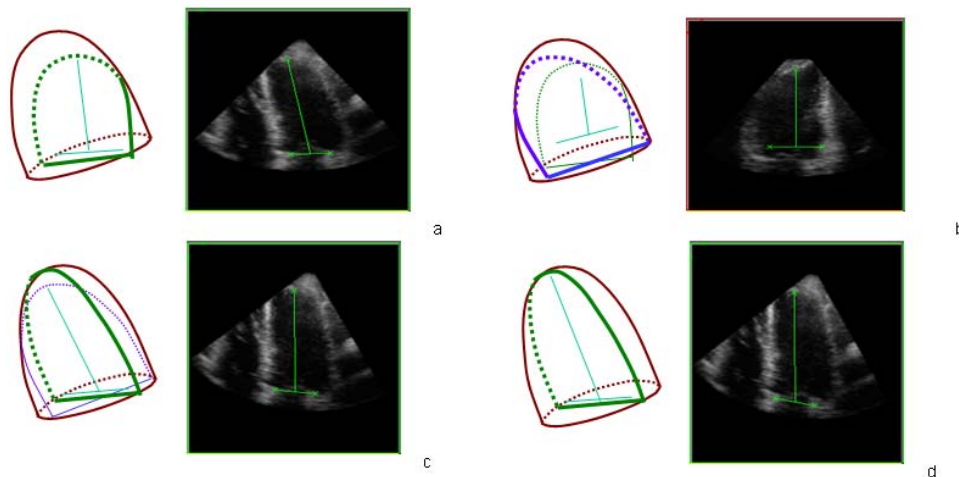


Figure 3.1 Manual method for LAX determination by iterative annotation of landmarks. a) After the apex and mitral valve markers are placed in an approximate 4C plane, the LAX is calculated, b) the perpendicular 2C approximation plane is taken and the LAX is adjusted, c) the perpendicular 4C approximation is used to adjust the LAX again, d) step b) and c) are repeated until we have the LAX.

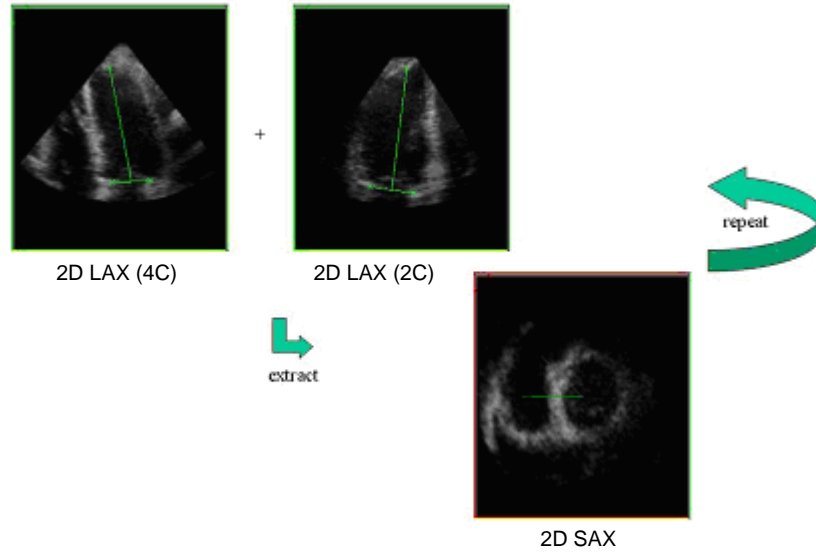


Figure 3.2 Block diagram of the iterative scheme.

Instead of detecting the landmark points individually, we try to locate the region where these landmarks are at fixed positions from its borders (Figure 3.3c). The typical image features are learned using a classification approach, previously used in applications like face detection [30] where the resulted implementations were used in digital cameras since they were suitable for real time applications (in terms of speed, accuracy, robustness).

First, we train classifiers to identify the left ventricle position, by using 2D example images in which these landmarks have been manually annotated. In a new test image, the region of interest is detected by extracting candidate regions from the test image and feeding this into the trained classifier. The classifier will determine for each candidate whether it contains the sought structure. The method will work in a coarse to fine approach, using three classifiers. First a rough scan over the 2D slices taken from the 3D dataset will give us the primal “hits” in which the left ventricle is present. These will be used to generate new candidates, by applying translations and rotations to the first “hits”, in which the apex and mitral valve center are sought. After classifying them with the second classifier, the new “hits” will be used to give us new input candidates, by applying finer translation, rotation, scale and shear, to create regions where all landmark points are sought (Figure 3.3). The third classifier is used to give us the final estimate of the landmark positions.

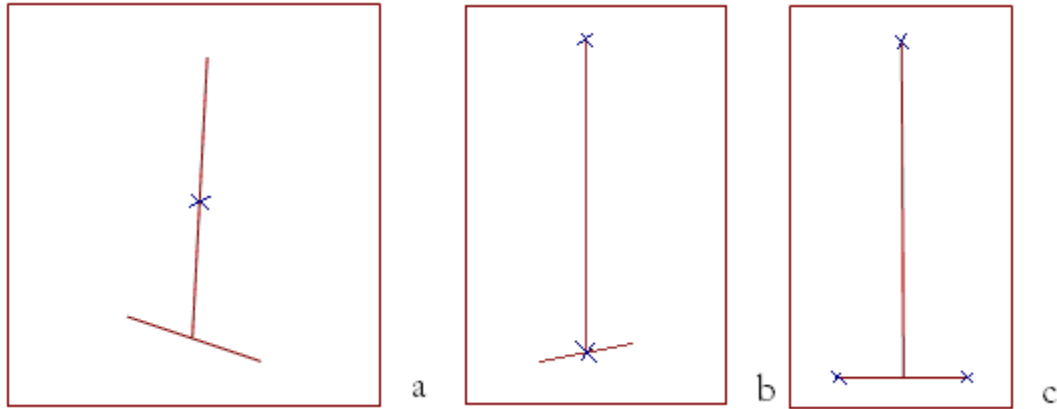


Figure 3.3 Drawings of the templates. a) Shows the template of the first classifier where only the left ventricle center is at a fixed position and b) shows a template with fixed positions for the apex and mitral valve center and c) with fixed positions for all sought anatomical landmark points.

We can split the whole method in three steps: training the classifiers, tuning them and the actual detection step. Before we get into any details an overview of the method can be found in Figure 3.4

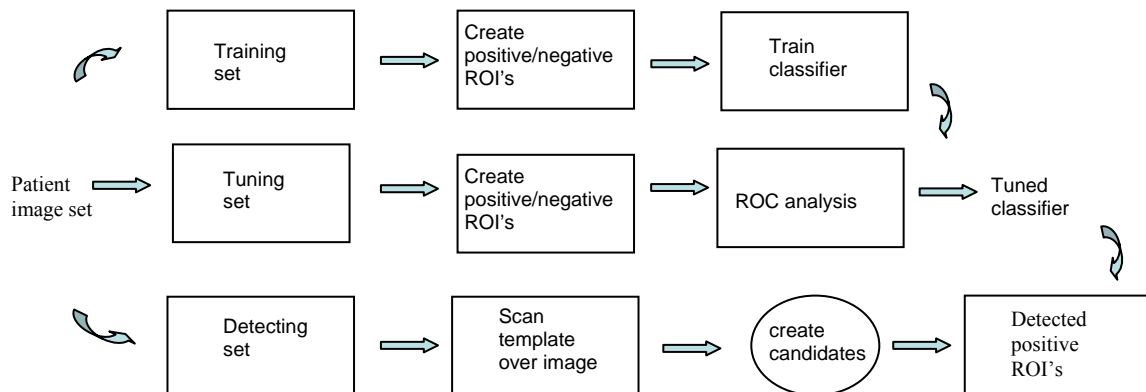


Figure 3.4 Overview of classifier creation and use.

3.2 Landmark detection

Our landmark detection algorithm relies on using expert annotated images to train classifiers. The image features are used to train our classifiers. We train three classifiers, with which we create a cascade that works in a coarse-to-fine scheme.

3.2.1 Features

We use a very large and general pool of simple Haar-like features. The advantage is that they provide a robust description of the sought structures. Also, a very fast computational scheme exists which allows calculation at any position in any scale at the same speed for

all sizes. Papageorgiou et al. in [33] proposed an over-complete feature pool that we combined with a very fast computation scheme proposed by Viola and Jones in [30].

A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. Viola and Jones [30] defined 6 types of these features (Figure 2.5). Each feature type can indicate the existence of certain characteristics in the image, such as edges or corners. For example, a feature of type 2 shown in Figure 2.5 can indicate the horizontal border between a dark region and a light region.

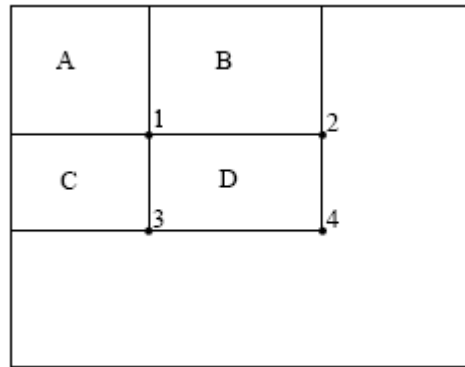


Figure 3.5 The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A+B, at location 3 is A+C, and at location 4 is A+B+C+D. The sum within D can be computed as $4+1-(2+3)$ [30].

One of the contributions of Viola and Jones was to use so-called integral images. Integral images can be defined as a 2D lookup table in the form of a matrix with the same size as the original image. Each element of the integral image ii contains the sum of all pixels located within the upper-left region of the original image (in relation to the element's position). The integral image at location (x,y) contains the sum of the pixel values i above and to the left of (x, y) , inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad 3.1$$

This allows computing sum of arbitrary rectangular areas in the image, at any position or size, using only 4 lookups (Figure 3.5).

The Haar types that we used were the same as in Carneiro et al. [29] (Figure 2.5). The number and the size of the feature types to use vary, depending on the classification stage. We did so in order to follow the coarse-to-fine scheme.

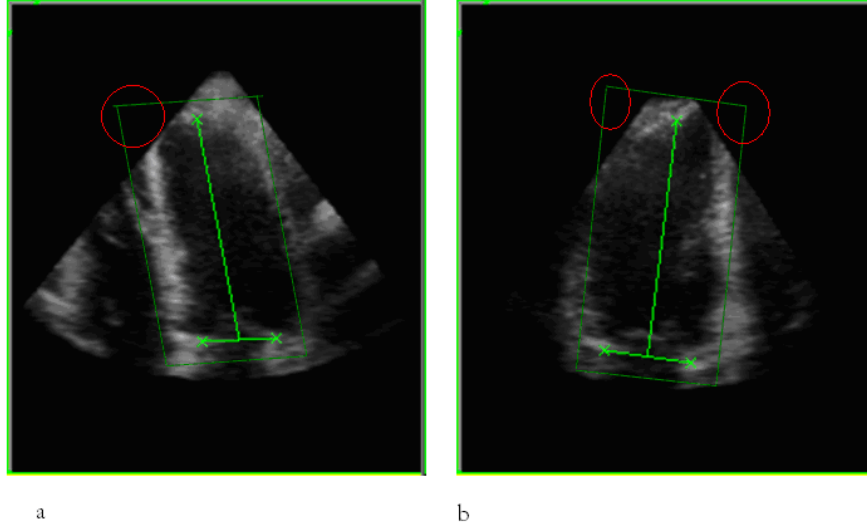


Figure 3.6 Annotated images of a) 4C view and b) 2C view along with the resulting template. The points where the template exceeds the image are marked by circles

Finally, a very important issue related to the calculation of our features had to do with missing image information, in regions where the template exceeds the actual image boundaries (Figure 3.6, Figure 3.7). Carneiro et al. [29] do not discuss this problem. On the other hand, Georgescu et al. [28] use a mask to exclude invalid pixel values. They claim that if the valid image mask is available we can use it to eliminate the contribution of the invalid pixels to the feature value. The mask can be inferred from the data, or is directly available when images are taken in controlled environment, which is our case. If we set to zero the intensity for the invalid pixels, the rectangle sum will no longer be influenced by incorrect values. However due to the missing data the sum will be “unbalanced”. To deal with this, we correct the rectangle sum using the number of valid pixels. The number of valid pixels can be easily found by first computing an equivalent map: the “integral mask” im :

$$im(x, y) = \sum_{x' \leq x, y' \leq y} m(x', y') \quad 3.2$$

where $m=1$ within the valid image region and $m=0$ for invalid pixels.

Similar to the integral image the number of valid pixels in a rectangle can be computed from the integral mask in the same number of operations (Figure 3.5). The equivalent feature value will be given by a weighted difference between the sums of the intensities i in the “light” and “dark” image regions. If we denote by R_+ the region where the pixels intensities contribute with a positive value (light region) and by R_- with a negative value (dark region), the feature value f is:

$$f = \begin{cases} \frac{N}{n_+} \sum_{(x,y) \in R_+} i(x, y) - \frac{N}{n_-} \sum_{(x,y) \in R_-} i(x, y) & n_-, n_+ > 0 \\ 0 & otherwise \end{cases} \quad 3.3$$

where n_- , n_+ denote the number of valid pixels for dark and light regions respectively, each containing N pixels.

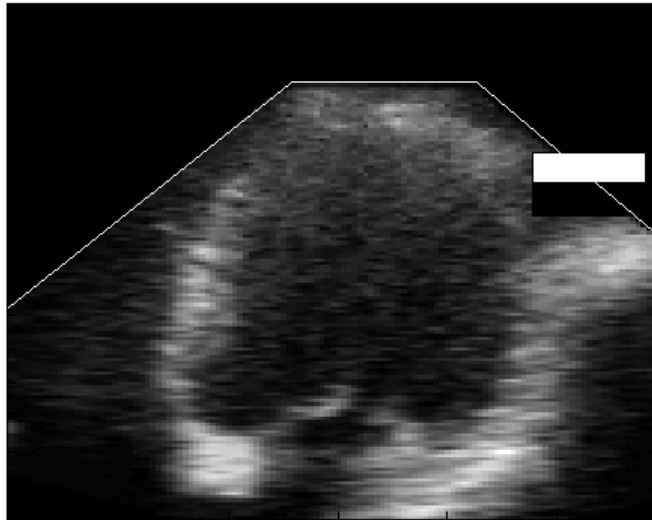


Figure 3.7 the image boundaries problem. The black-white rectangle is our Haar feature. The region within the ultrasound sector (under the white line) is valid and the one outside of it invalid.

The use of integral masks improved the classifiers' performance, but invalid pixels were still used by them. Mapping the pixels used clearly showed that the masks were not efficient enough (Figure 3.8). So, apart from using the integral masks to avoid invalid regions, we also explicitly excluded most of them. By summing all positive examples from all patients, we used this summed template to know which regions were invalid for all our patients (Figure 3.9). We used this information to exclude the feature values of these regions from both our training and testing set.

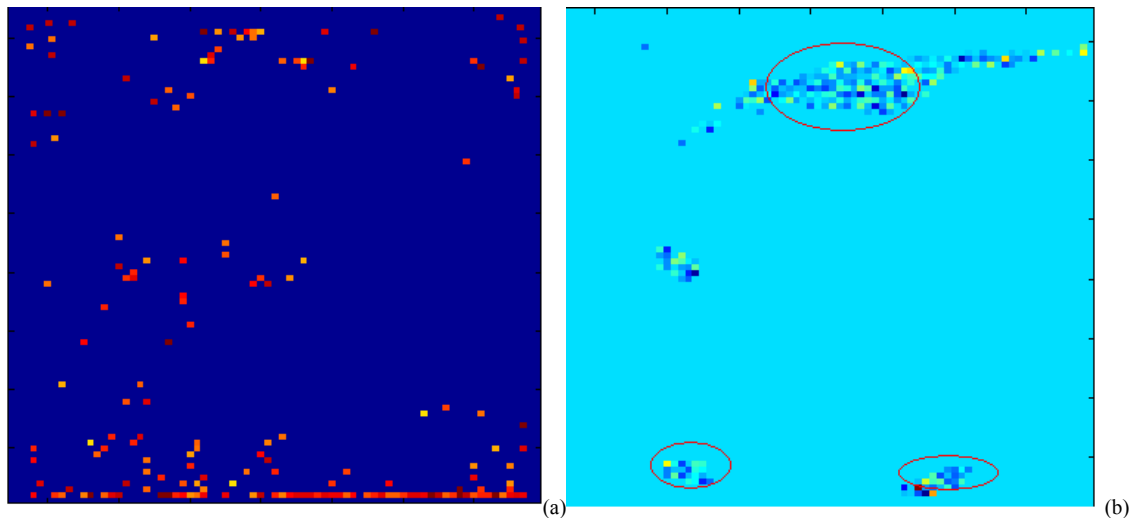


Figure 3.8 Examples of pixels used by classifiers. (a) Only image masks are used, (b) after explicitly excluding pixel values. It is clear that the classifier focuses near the anatomical landmark locations (red circles), since there are many pixels used from the top part (apex) and two location in the base part (mitral valve points) of the image.

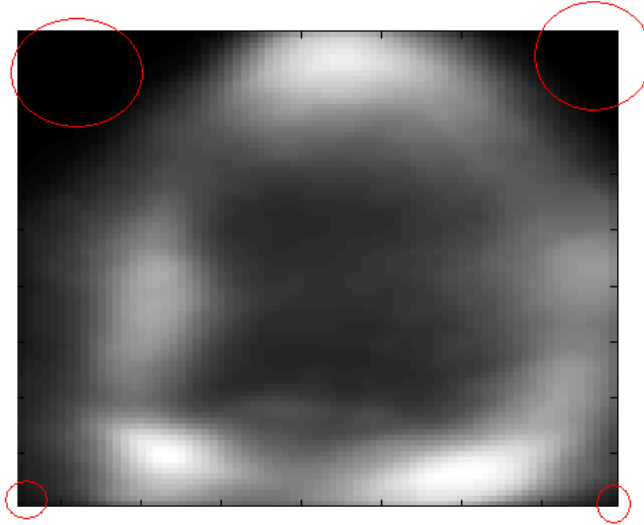


Figure 3.9 Sum of templates for the two chamber view.

3.2.2 Classifier

At this point it is important to describe and discuss the characteristics of the classifier we used. We chose to work with Adaboost because it has been applied to problems similar to ours with the use of Haar features [29].

“Boosting” is a general method for improving the performance of any learning algorithm by iteratively applying weights on misclassified data. In theory, boosting can be used to significantly reduce the error of any “weak” learning algorithm that consistently generates classifiers which need only be a little bit better than random guessing. Despite the potential benefits of boosting promised by the theoretical results, the true practical value of boosting can only be assessed by testing the method on “real” learning problems. Freund and Schapire [34] introduced a new boosting algorithm that they named AdaBoost.

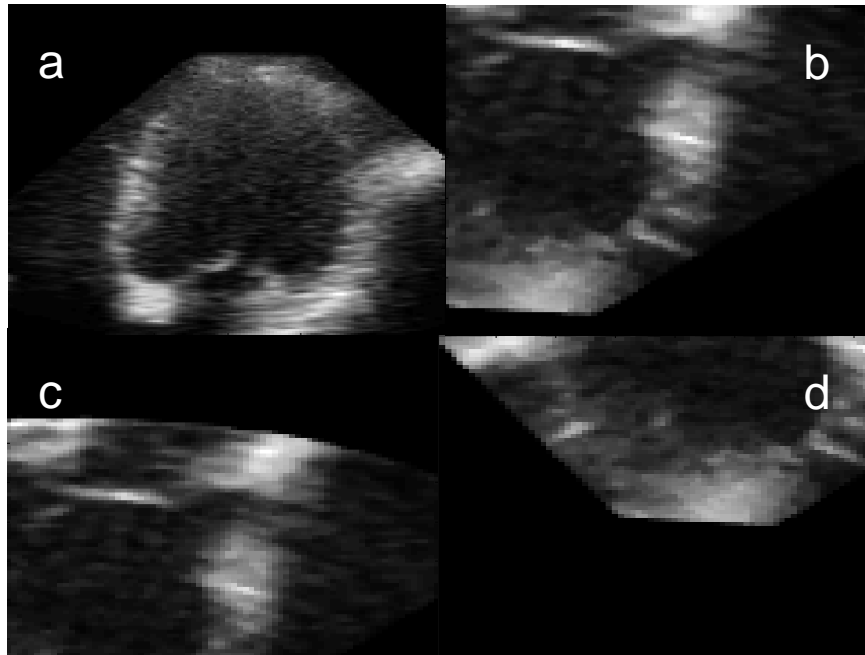


Figure 3.10 Examples of (a) positive sub-images and (b), (c), (d) negative ones taken from the same patient.

AdaBoost is a machine learning boosting algorithm capable of constructing a strong classifier through a weighted combination of weak classifiers (a weak classifier classifies correctly in only a little bit more than half the cases). It is important to make clear that the AdaBoost theory uses the term positive examples, to indicate the image regions that contain the sought object, and negative examples for the ones that don't (Figure 3.10).

Starting with training samples all with equal weights, the AdaBoost builds a strong classifier, using a few features. If a training sample is misclassified, the weight of that training sample is increased (boosted). A second classifier is built using the new weights, which are no longer equal. Again, misclassified training data have their weights boosted and the procedure is repeated. Typically, this is repeated 500-1000 times. A score is assigned to each classifier, and the final classifier is defined as a linear combination of the classifiers from each boosting round, based on their scores. We can see an overview of the training algorithm in Figure 3.11.

In our application we used more complicated classifiers as weak classifiers. We used the `treec` function of `prTools`, which trains a tree of simple classifiers (each feature is the simple classifier here). We tested several tree levels, as well as different classifiers.

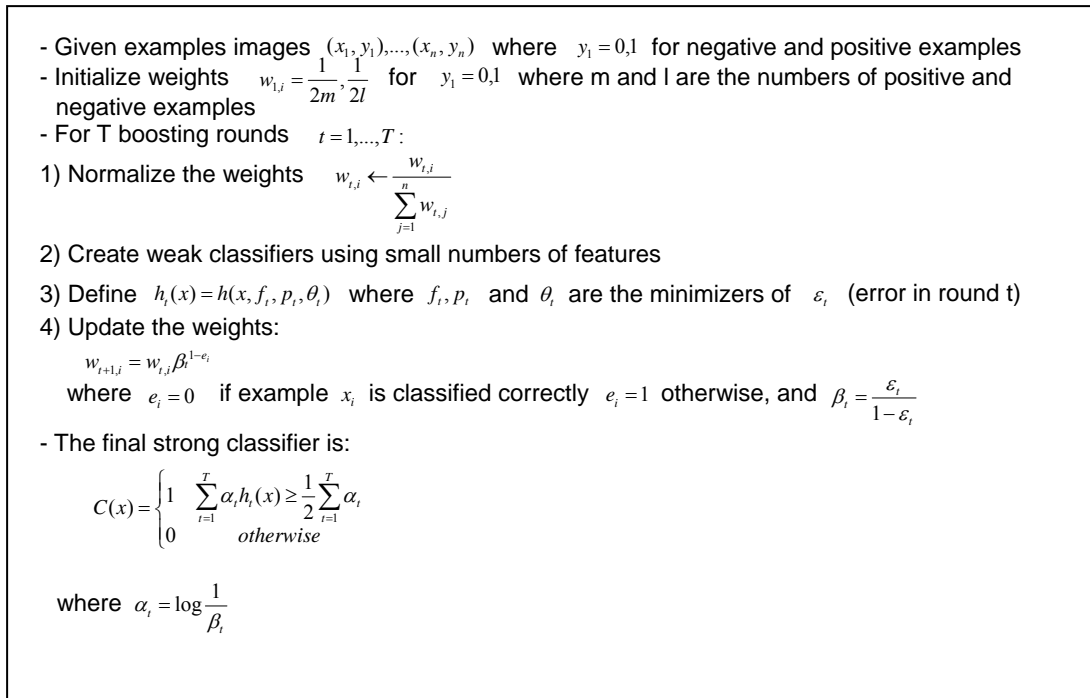


Figure 3.11 The training algorithm for the AdaBoost classifier

3.2.3 Cascade

For the actual classification problem, we chose to use a cascade of classifiers, which achieves increased detection performance while reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-regions while detecting almost all positive instances (i.e. the threshold of a boosted classifier can be adjusted so that the false negative rate is close to zero). Simpler classifiers are used to reject the majority of sub-regions before more complex classifiers are called upon to achieve low false positive rates. The overall form of the detection process is that of a degenerate decision tree, what we call a “cascade” (Figure 3.12). A positive result from the first classifier is a candidate for evaluation in the second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier is a candidate for the third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-region.

Stages in the cascade are constructed by training classifiers using AdaBoost and then adjusting the threshold to minimize false negatives. We do so with the help of receiver operator characteristic (ROC) curves. In classifiers theory, a ROC curve is a graphical plot of the sensitivity vs. (1 - specificity) for a binary classifier system as its discrimination threshold is varied. The ROC can also be represented equivalently by plotting the fraction of true positives (TPR = true positive rate) vs. the fraction of false positives (FPR = false positive rate), also known as a Relative Operating Characteristic curve, because it is a comparison of two operating characteristics (TPR & FPR) as the criterion changes [35]. In a two-class prediction problem (binary classification), in which

the outcomes are labeled either as positive (p) or negative (n) class, there are four possible outcomes from a binary classifier. If the outcome from a prediction is p and the actual value is also p, then it is called a true positive (TP); however if the actual value is n then it is said to be a false positive (FP). Conversely, a true negative has occurred when both the prediction outcome and the actual value are n, and false negative is when the prediction outcome is n while the actual value is p (Figure 3.12).

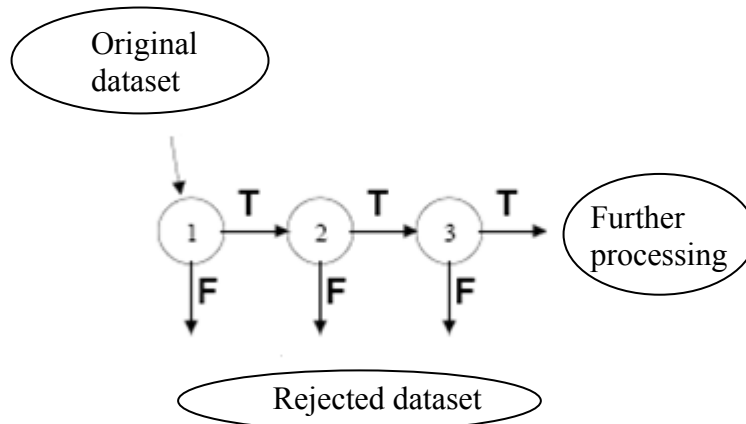


Figure 3.12 An example of cascaded classifiers

Note that the default AdaBoost threshold is designed to yield a low error rate (high accuracy) on the training data. In general a lower threshold yields higher detection rates (higher p') and higher false positive rates. We try to choose the appropriate threshold, that will give the optimal number of “hits” (predicted positives) that we want.

		<u>True class</u>			
		p	n		
<u>Hypothesized class</u>	p'	True Positives	False Positives	fp rate = $\frac{FP}{N}$	tp rate := $\frac{TP}{P}$
	n'	False Negatives	True Negatives	precision = $\frac{TP}{TP+FP}$	recall = $\frac{TP}{P}$
<u>Column totals:</u>		P	N	accuracy = $\frac{TP+TN}{P+N}$	
				F-measure = $\frac{2}{1/\text{precision}+1/\text{recall}}$	

Figure 3.13 Confusion matrix and common performance metrics calculated from it for a typical ROC curve.

3.2.4 Marginal Space Learning

Besides a faster computation, the cascade approach gave us the opportunity to apply a coarse-to-fine scheme for our problem, by first finding the region of the left ventricle and then the apex and mitral valve points. Zheng et al. [36] proposed a similar method, which they named marginal space learning.

As we have already said, for affine 2D object detection, there are 6 degrees of freedom: two for translation and four for rotation, scaling and shear. The object is found by scanning the region exhaustively over all possible combinations of locations, orientations, and scales. Exhaustive search makes the system robust under local minima. However, it is challenging to exhaustively search for all parameters simultaneously, since the number of parameter combinations increases exponentially with respect to the number of parameters. To efficiently localize the object, we perform parameter estimation in a series of parameter spaces (i.e. marginal spaces) with increasing dimensionality. To be specific, the task is split into three steps: object position estimation, position-orientation estimation, and position-orientation-scale-shear estimation (Figure 3.14). We approached the detection problem like Carneiro et al. did [29], defining a rectangle region of interest where the landmark points have known positions. We also included shear in our implementation though, to achieve fixed positions for the landmark points (Figure 3.3c). After each step, several candidates are retained for the following estimation step. At the end, the candidates are aggregated to provide the final detection result.

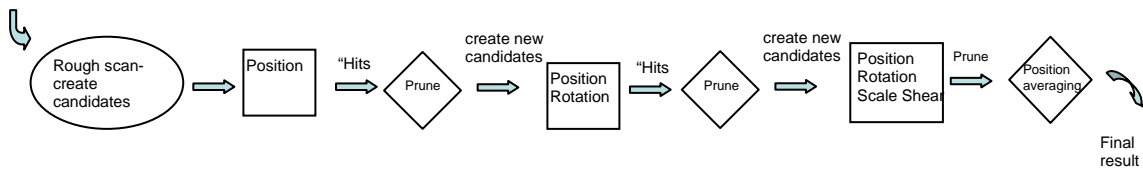


Figure 3.14 Block diagram of the marginal space learning algorithm [36].

So, the first classifier performs a rough scanning with a region of fixed size on the 2D slice from the 3D set, considering only translation. At this point we only want to detect the region where the left ventricle is present, regardless of its scale or rotation. As a result a rectangular image that contains the left ventricle and is taken from the plane with the three landmark points would be sufficient for a positive example (Figure 3.3a). From each patient, we can generate several positive examples, limiting ourselves with the condition that the three landmark points should always be visible. In the same way, negative images can be generated by translating the center of the template to a much greater extent (Figure 3.15-3.16). This will give us the opportunity to use large number of images for the training phase, which is important because a robust classifier, as a rule of thumb, needs a database that contains ten times more images than the features it uses.

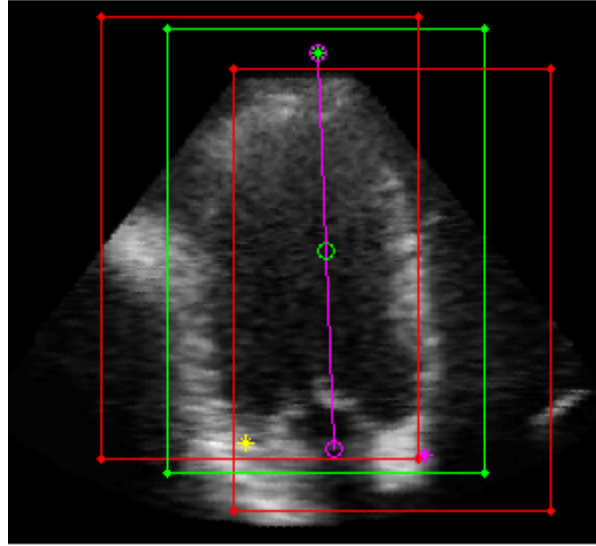


Figure 3.15 Example of 2 chamber view with templates. With green color we see the original template. With red color we see negative samples

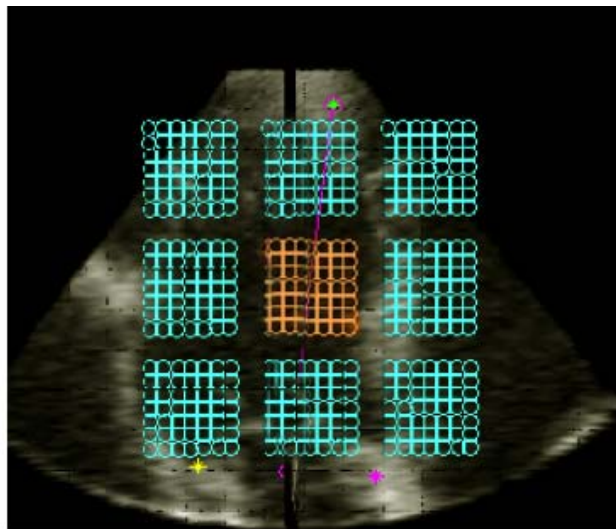


Figure 3.16 Centers of positive and negative examples used for training. The orange ones are the positive examples.

The second classifier will try to locate the actual position of the apex and the mitral valve center within the regions that were classified as “hits” at the previous stage. To do so we have to change our template. We use a rectangular template with the ventricle center as its center, but this time where the apex and mitral valve center will have fixed distances from the borders (Figure 3.3b). The negative images are generated again by translations of the image center. Here we also rotate the images, creating new set of candidates.

The third classifier tries to locate all three landmark points. What differentiates it from the second one is the addition of scale and shear. So, in order to have fixed distances between each point in the mitral valve ring and the template borders, we apply shear to

the template of the previous stage (Figure 3.3c). Additionally, the ranges of the other parameters (rotation, scale) are set to finer numbers, to achieve better accuracy.

Another important issue has to do with the features that we use. We experimented with different Haar types and in different sizes, since this has not been reported in literature. That way we can exploit the fact that we are using a cascade and approach the coarse-to-fine scheme with the additional help of the features: larger sizes in the beginning where less detail is important and smaller sizes in later stages of the cascade.



Figure 3.17 Types of feature used for training the classifiers.

4 Experimental Setup

4.1 Data representation

Our data set consists of 3D end-diastolic images of the left ventricle, which were acquired using the Philips Sonos 7500 system with a four matrix array transducer. The typical dimensions were 160x144x208pix with a resolution of 1x1.2x0.7mm/pix. We annotated these volumes, defining the landmark points and the orientation angles for the different 2D views (two, three, four chamber and short axis view) using 3DStressView, a software package for analyzing 3D echocardiograms (Thoraxcenter Biomedical Engineering, Erasmus MC Rotterdam, the Netherlands). The annotation was as follows:

- Indicate the lateral and the septal points on the mitral valve ring along with the apex in an approximate 4-chamber view
- Indicate the anterior and the inferior points along with the apex in a view perpendicular to the previous one (approximate 2-chamber)
- Repeat until the long axis is found
- Indicate the orientation angles in the short axis view (SAX)

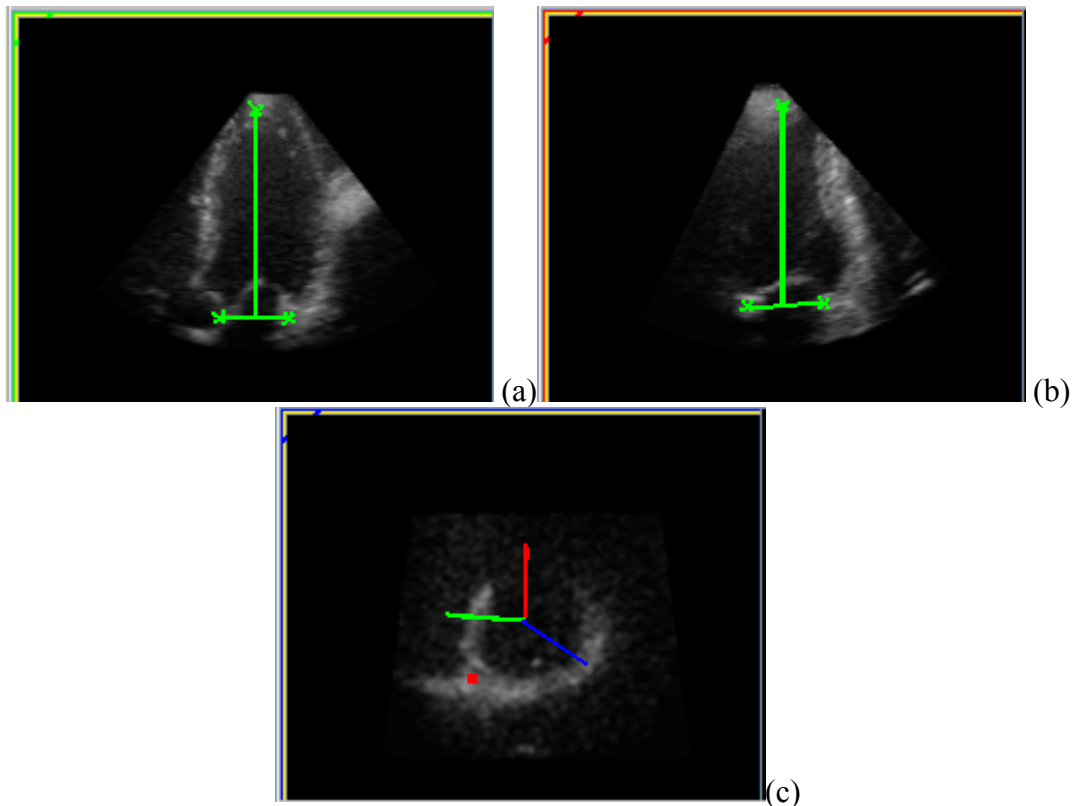


Figure 4.1 Annotated images of good quality a) 4C view, b) 2C view, c) SAX view. The orientation angles of the apical views are also indicated in the SAX view. With red color we see the 2C view, with green the 4C and with blue the 3C view.

The sets are taken from 85 patients which we split into a training, tuning and detection set. We used 60 patients to generate positive and negative examples, with which we trained the classifiers. The remaining 25 were used both for tuning and detection. By tuning we mean the procedure of tuning the operating points of our classifier using ROC analysis. By detection, we mean to apply the classifiers to detect the landmark points. This will be explained in detail later on. The 25 patients were split into groups of 5. Each time 20 were used for tuning and 5 for detection. This was done five times and at the end the results were averaged. We tried to use images of different quality. So, all three groups were containing images of good medium and bad quality.

4.2 Preprocessing

Some preprocessing is applied to the images. Ultrasound images are more difficult to interpret compared with other medical imaging modalities (CT, MRI) due to speckle noise and other artifacts. Additionally, the acquisition procedure of ultrasound images is highly dependent on the user and varies from patient to patient. To cope with the above we tried to find the best preprocessing steps for our set of patient data.

First of all we tried to get a feeling of the various speckle removing algorithms described in literature [37, 38]. To do so, we processed a set of 20 images using the programming environment MeVisLab (version 1.6.1, MeVis Research GmbH, Bremen, Germany), suitable for medical image processing. Gaussian smoothing, Median filtering and Anisotropic diffusion were applied.

The three methods can be used to suppress speckle. Gaussian smoothing may blur salient structures, Anisotropic diffusion required more time than the other methods and Median filtering may introduce spurious artifacts (i.e., in specific regions, edges appeared that were not obvious in the original image). We chose to use Median filtering, because it is fast and the artifacts that are introduced are not significant, when small kernels are used.

4.3 Experimental setup

As we have already mentioned, we split the task in three steps, training the classifiers, tuning them by choosing an operating point (tradeoff between errors in our classes) and finally detection in real life situation.

4.3.1 Training



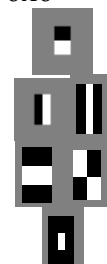
As already mentioned, we trained three Adaboost classifiers, using simple tree classifiers (prTools treec classifier) as weak ones. Having in mind the size of our templates and that each feature value should be used once, we used 50 boosting rounds, while the tree level was set at 5 leaf nodes, since more boosting rounds or deeper level would result require bigger templates. We used three types of classifiers, with three different configurations to get the desired results. We always had in mind the coarse-to-fine scheme and the level of accuracy we wanted to have in each step.

So, in the beginning we search for the region of interest in a rough way. The level of detail is not too deep and only translation is considered. As a consequence, the translation step was set to 5 pixels and all translations within ± 15 pixels were considered to give a positive template. The Haar features that we used were relatively big, set at 12 pixels, and four types were used. At this point each pixel is 1mm, and the image is isotropic. Usually, choosing the type of the Haar features is part of the “boosting”, but we decided not to work this way due to memory limitations. The Haar values were chosen based on experiments we did in Matlab and using OpenCV. OpenCV is a library of programming functions mainly aimed at real time computer vision. It contains an implementation of the Viola and Jones [30] face detector, which is a similar to our application. They use Haar like features to train the classifiers, and then use the best types in terms of sizes and type.

In this step of the cascade we also included some slightly rotated versions of our positive examples. We did so in order to add greater variation in the left ventricle types. If we had an enormous database of patients, we wouldn't need to. Including some rotation variation should make our classifier more robust.

The second classifier was trained to detect rotation and find translation more accurately than the first. Since accuracy was one of the goals, the Haar features were smaller than before. Also, the translation steps were smaller and a different template was used (Figure 3.13). Finally the third classifier used even more fine steps, resulting in small features and translations/rotations, while shear was also included. The above are summarized in Table 1.

Table 1 Training settings

	Image size	positive		negative		features
Step 1 ROI	150x100 Template: Figure 3.13 (a)	Transform	Range	Transform	Range	12x12 
		Translation Rotation	-15:5:15 -10 ⁰ :5 ⁰ :10 ⁰	Translation	-15:5:15 With an extra translation of ±45	
Step 2 Coarse	120x80 Template: Figure 3.13 (b)	Transform	Range	Transform	Range	10x10 
		Translation Rotation	-6:2:6 -10 ⁰ :5 ⁰ :10 ⁰	Translation Rotation Scale	-12:3:12 With an extra translation of ±25 -10 ⁰ :5 ⁰ :10 ⁰ 40% to 10% and 160% to 200% with step of 10%	
Step 3 Fine	120x80 Template: Figure 3.13 (c)	Transform	Range	Transform	Range	6x6 
		Translation Rotation Scale Shear	-4:1:4 -3 ⁰ :1 ⁰ :3 ⁰ 95% to 105% of d with step of 2% Maximum angle of 30 ⁰	Translation Rotation Scale Shear	-6:2:6 With an extra translation of ±15 -3 ⁰ :1 ⁰ :3 ⁰ 68% to 80% and 150% to 162% with step of 4% Maximum angle of 30 ⁰	

4.3.2 Tuning the classifier

The tuning part had to do firstly with organizing our classifiers and visualizing their performance and secondly with choosing the operating points that would perform best. We do so, as already mentioned, using the receiver operator characteristic (ROC) curves (Figure 4.2). The basic idea of ROC analysis is very simple: For a given trained classifier

and a labeled test set define a set of possible operating points and estimate different type of errors at these points. In our application instead of using the typical ROC curves (sensitivity vs 1-specificity) we used a different configuration (error on positive class vs error on negative class). We did so because such a curve was easier to interpret. It is obvious that the closer we get to point (0, 0), the better our results will be.

Since we are using a coarse-to-fine approach our objectives at each step are different. So, at first we want to be sure that all our positive examples (images with the region of interest) will pass to the next stage. As a consequence, we are more interested in having zero error on the negative class: the false positives will be discarded in later stages.

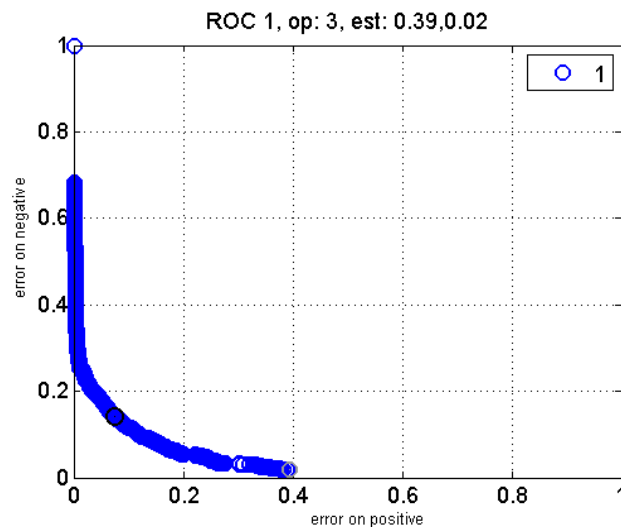


Figure 4.2 Adapted Receiver Operator Curve of the second classifier. On x axis we have the error on the positive examples and on y the negative examples. In the title we can see that among all operating points the third one is chosen, which gives an error of 0.39 on positive ($\frac{FP}{N}$, Figure 3.11) and 0.02 on negative examples ($\frac{FN}{P}$, Figure 3.11).



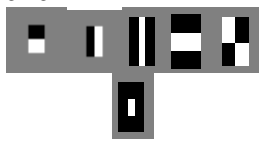
However, as we conducted the experiments we saw that opting for zero error in negative class resulted in large number of hits. To avoid this, as it meant more candidates in later stages and more computational time, we decided to optimize the performance, allowing some errors in first class, but not higher than 0.4 so we could be certain that most of the true positives would pass to the next stage of the cascade. This would not affect the overall result, since the number of positive examples that we created was redundant. After choosing an operating point, the classifier works on that configuration and is ready to be used for detection.

4.3.3 Detecting

In the last step of the evaluation setup, the actual detection is performed. This step is combined with the previous one, so from the 25 patients, each time 20 are used for the ROC analysis and 5 for detection.

Since we are using different configurations for training the classifiers, the detection configurations also vary. So, first we do a rough scan on the 2D slice that we extract from the 3D dataset. Here only translation is considered, while the size of the Haar features is fixed and the same as we used in the training phase. The “hits” (detected positives) of this step are pruned, based on a maximum distance from their average to remove outliers, and then are used to generate new region candidates for the second classifier. The detected regions are rotated, translated and scaled. The translation steps are smaller than previously, but the rotation and scale are still in the rough phase. The new hits are again pruned and fed to the final classifier. Here no translation is needed and we have the fine stage for both rotation and scale. Shear is also introduced, to add accuracy to our final candidates. The above are summarized in Table 2.

Table 2 Detecting settings (d is the pixel size).

	Image size	Search		features
Step 1 ROI	150x100 d= 1mm	Transform	Step size	12x12 
		Translation	6 pixels	
Step 2 Coarse	120x80 d=1mm	Transform	Range	10x10 
		Translation	-6:2:6 pixels	
		Rotation Scale	-10 ⁰ :5 ⁰ :10 ⁰ 0,8*d :0,1*d: 1,5*d	
Step 3 Fine	120x80 d=1mm	Transform	Step size	6x6 
		Rotation	-4 ⁰ :1 ⁰ :4 ⁰	
		Scale Shear	0,95*d:0,02*d:1,05*d 10 steps maxAngle=30 ⁰	

5 Evaluation

As already explained, the evaluation is done in each stage in two steps: first we conduct ROC analysis to set the operating point of the classifier and then we test the classifiers' ability to detect the sought structure on that point. As a consequence, we will present the results, in these two steps for the four and two-chamber views.

5.1 Two chamber view plane

A first impression of the classifiers' performance can be given from the ROC curves, which we use to tune among the possible classifier operation points. So, Figures 5.1-5.3 show the curves that we get for the first batch of patients. We show here only one of the batches because the ROC-like curves of the batches are very similar. At this point we should highlight that the third classifier originally suffered from overtraining. The image variations that were included in the training process resulted in two-point ROC curves, an indication of overtraining (Figure 5.4). These classifiers would classify all input templates either as positives or negatives in the detection step. We solved that issue by using less complex weak classifiers, setting the boosting rounds to 30 and allowing the trees to grow until the third level.

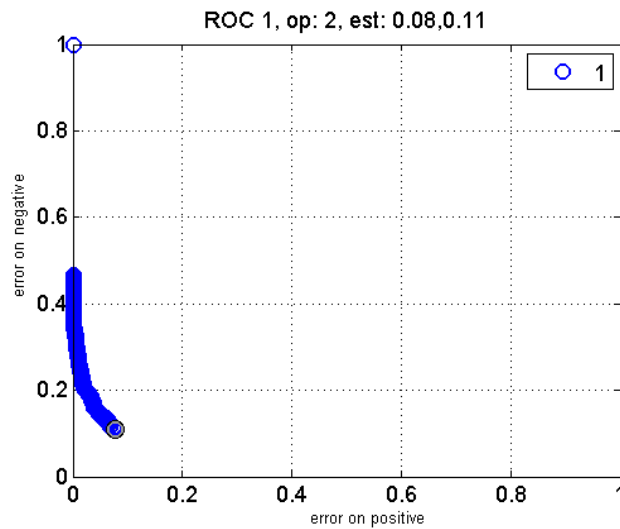


Figure 5.1 ROC curve of the first 2C classifier for the first group of patients

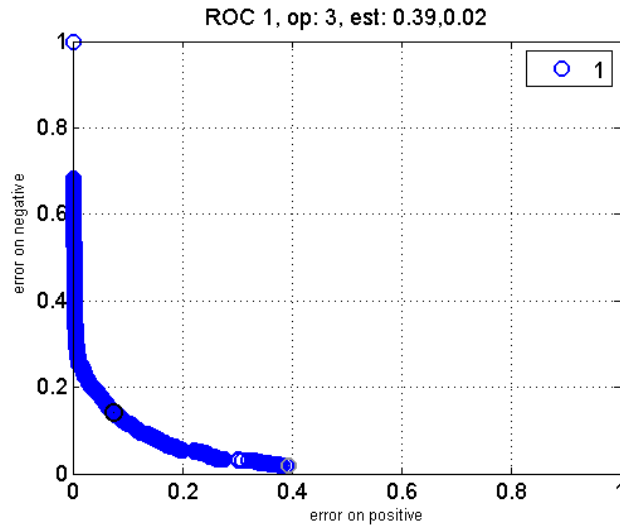


Figure 5.2 ROC curve of the second 2C classifier for the first group of patients

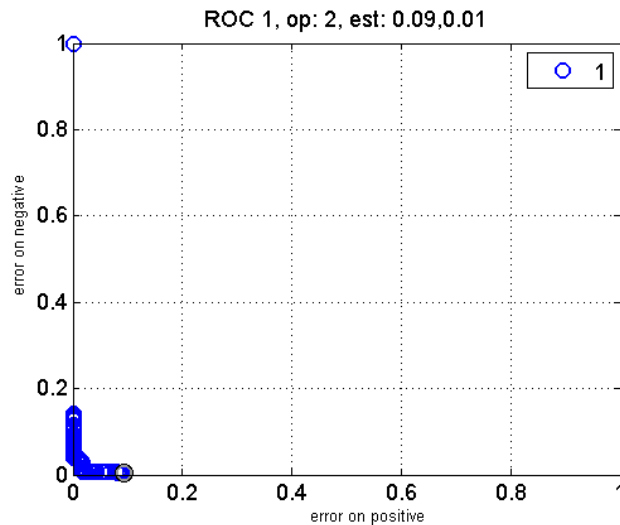


Figure 5.3 ROC curve of the third 2C classifier for the first group of patients

The curves indicate that all three classifiers can work with relatively small errors. To check that other simple and easier to implement classifiers would not give similar results, we used a Gaussian classifier, applying a PCA to the training data and then use a threshold to classify the data used for tuning. The resulted ROC curve is shown in Figure 5.5. Obviously, a classifier like that would give poor results since the classification error of the negative examples is high, regardless of the operating point. Finally, we used ROC curves to test whether our effort to expand our database generating artificial patient variation is justified. The results are presented in Figure 5.6 where we have the ROC curve of a first classifier that does not include rotated versions of the images. We see operating points that give larger errors in the positive class, which was an indication that this classifier would give poorer results.

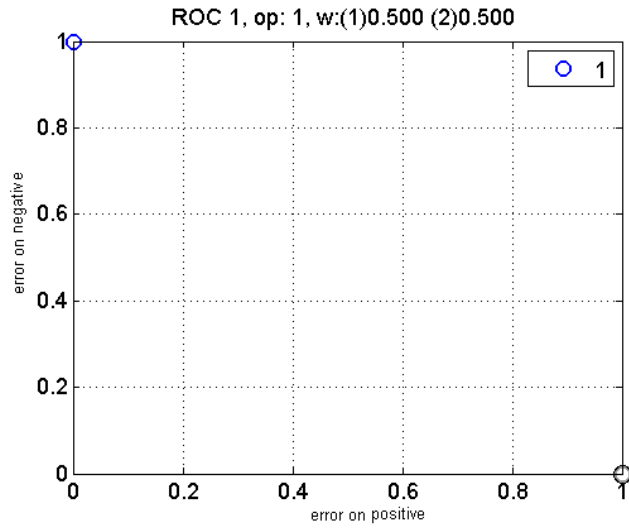


Figure 5.4 ROC curve of a third 2C classifier in the case of overtraining.

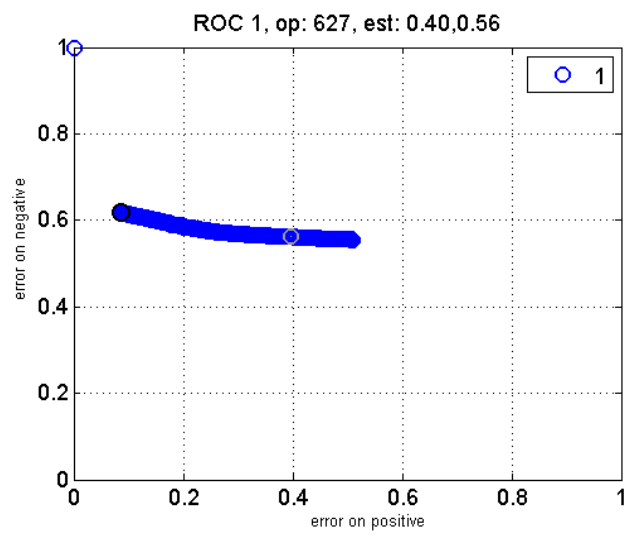


Figure 5.5 ROC curve of a Gaussian classifier on the first 2C classifiers training data

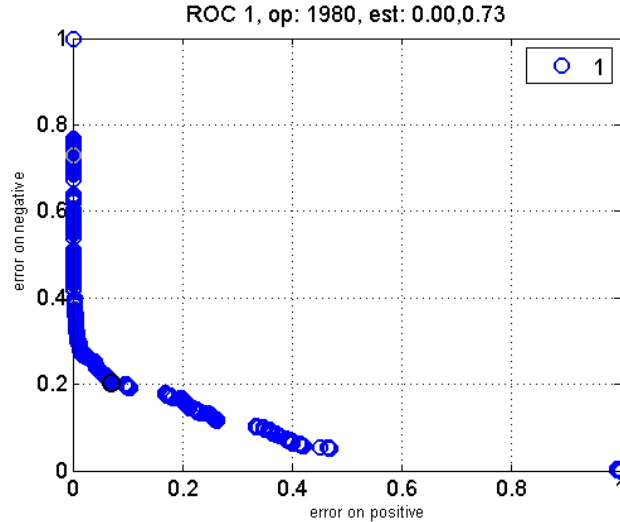


Figure 5.6 ROC curve of the first 2C classifier for the first group of patients with no rotations.

The detection performance was evaluated by measuring the Euclidian point-to-point errors, meaning the distance between the manually annotated position of the sought point and the one indicated by the classifier. At the first stage, from a total of 885 candidate regions for each of the 25 patients we got after pruning 15-20 candidate positions, depending on the image quality of the 3D sets (Figure 5.7), as more candidates were obtained for images of lower quality. For each patient we calculated a mean value from all the “hits”. The distance between the manually annotated centers of the region of interest and the means was on average 6.5 ± 2.5 mm. These detected regions were translated, rotated and scaled resulting in $7 \times 7 \times 5 \times 8 = 1960$ new candidates, which we classify using the classifier that we trained with the help of the template shown in Figure 3.3b. This allows us to detect the apex and mitral valve center position. The resulted mean distance between the true apex and the mean detected one equal to 7.2 ± 7.0 mm, while for the mitral center the error was 5.6 ± 5.2 mm, while the paired t-test noted statistically significant improvement in the accuracy of the detected image center. The candidates indicated from the second classifier as possibly containing the left ventricle were further rotated, scaled and sheared giving us $9 \times 6 \times 11 = 594$ new images to classify. The last classifier used was trained with templates like the one in Figure 3.3c, so all three landmark points could be predicted. The resulted errors were at the rate of 7.1 ± 6.7 mm for the apex, 5.8 ± 3.5 mm for the anterior and 4.5 ± 3.1 mm for the inferior point. It is important to note here that the accuracy from the second to the third classifier does not always change significantly (Figure 5.8).

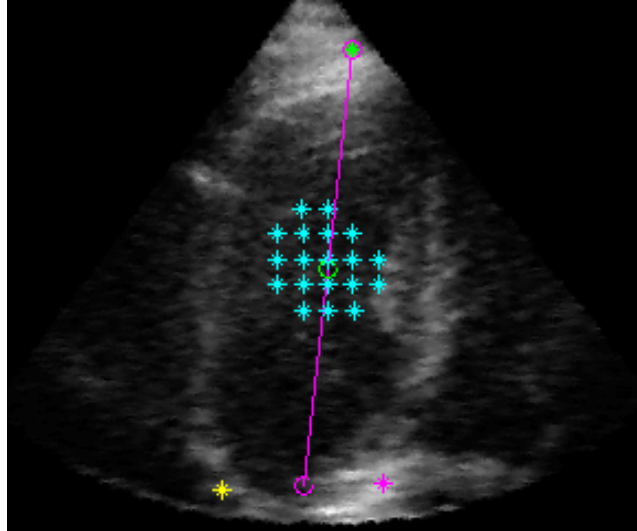


Figure 5.7 Example of the centers for the 2 chamber view detected by the first classifier. The true center is marked with a green circle and the hits with cyan asterisks. The inferior (magenta), anterior (yellow) and apex (cyan) markers are also indicated.

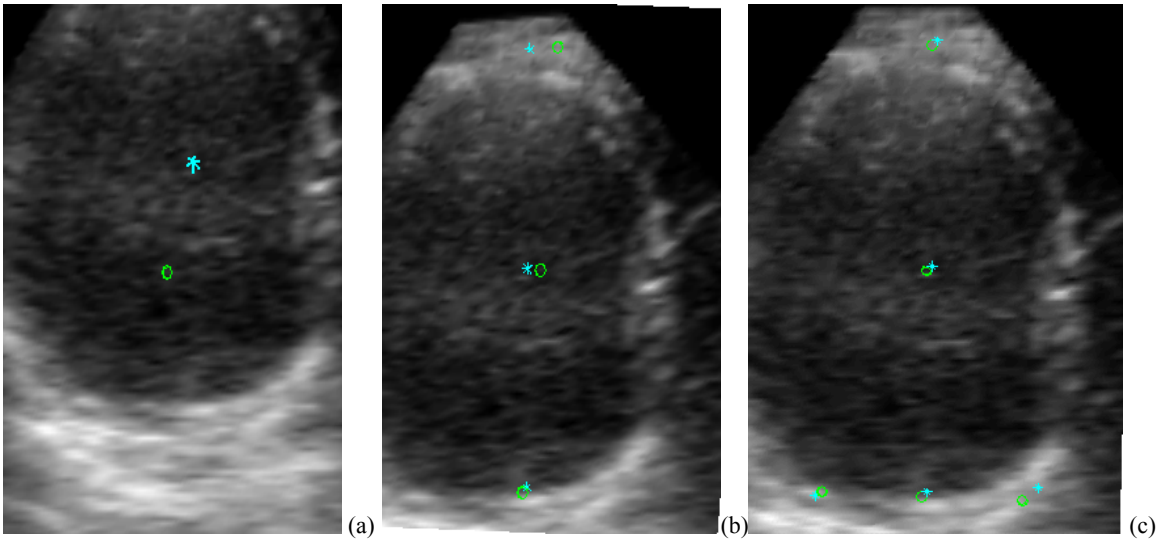


Figure 5.8 Example of detected regions by (a) the first (b) the second and (c) the third classifier. With green circles are depicted the detected points, while with cyan asterisks the true ones.

Table 3 Results on 2C view. * denotes statistically significant ($p < 0.05$) difference with respect to the previous classifier. † denotes statistically insignificant difference with respect to the previous classifier.

	Number of "hits" after pruning	Average error on points	
Step 1 ROI	15-20	Image center	6.5±2.5mm
Step 2 Coarse	1000-2000	Image center	5.8±2.4mm*
		Apex	7.2±7.0mm
		Mitral valve center	5.6±5.2mm
Step 3 Fine	150	Apex	7.1±6.7mm†
		Mitral valve center	5.2±2.8mm†
		Anterior	5.8±3.5mm
		Inferior	4.5±3.1mm

5.2 Four chamber view plane

In the four chamber view the ROC curves are used once again to give a first impression of the classifiers' performance (Figure 5.9-5.11).

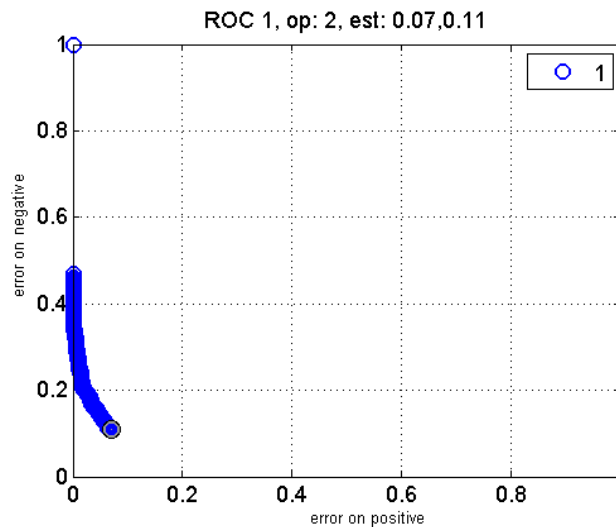


Figure 5.9 ROC curve of the first 4C classifier for the first group of patients

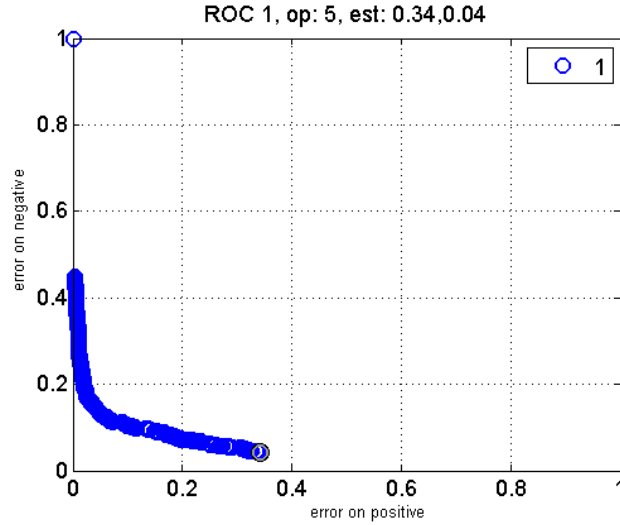


Figure 5.10 ROC curve of the second 4C classifier for the first group of patients

The same sequence of classifiers was once again used. The resulted errors are presented at Table 4. At the first stage, again from a total of 885 candidate regions for each of the 25 patients we got 10-15 candidate positions, depending on the image quality of the 3D sets. The candidates were less than in the 2C view case probably because there were less outliers in this view. The mean distance between the manually annotated center of the region of interest and the mean detected one was here on average 7.9 ± 4.1 mm. These detected regions were once again translated, rotated and scaled, like described in the 2C case, resulting in 1960 new candidates. The resulted mean distance between the true apex and the mean detected one equal to 8.2 ± 4.7 mm, while for the mitral center the error was 8.0 ± 5.0 mm. The candidates indicated from the second classifier as possibly containing the left ventricle were further rotated, scaled and sheared giving us $9 \times 6 \times 11 = 594$ new images to classify. The last classifier used was once again trained with templates like the one in Figure 3.3c, so all three landmark points could be predicted. The resulted errors were 7.9 ± 7.1 mm for the apex, 5.6 ± 2.7 mm for the septal and 4.0 ± 2.6 mm for the lateral point.

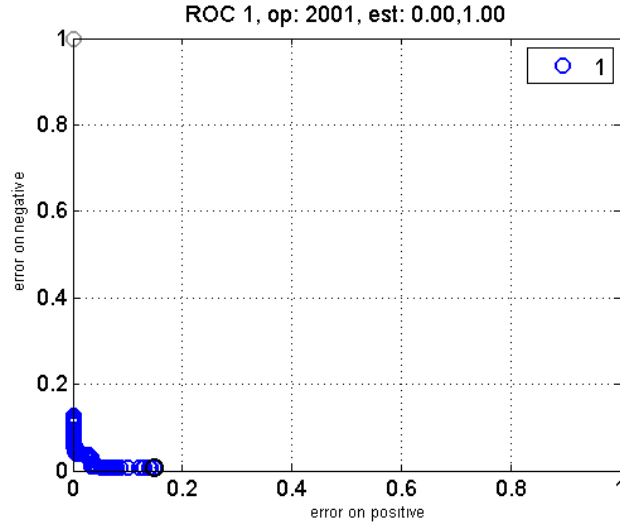


Figure 5.11 ROC curve of the third 4C classifier for the first group of patients

Table 4 Results on 4C view. * denotes statistically significant ($p < 0.05$) with respect to the previous classifier. † denotes statistically insignificant difference with respect to the previous classifier.

	Number of “hits” after pruning	Average error on points	
Step 1 ROI	10-15	Image center	$7.9 \pm 4.1 \text{mm}$
Step 2 Coarse	1500-2500	Image center	$7.6 \pm 4.8 \text{mm}^*$
		Apex	$8.0 \pm 5.0 \text{mm}$
		Mitral valve center	$8.2 \pm 4.7 \text{mm}$
Step 3 Fine	150	Apex	$7.9 \pm 7.1 \text{mm}^\dagger$
		Mitral valve center	$4.8 \pm 2.3^*$
		Septal	$5.6 \pm 2.7 \text{mm}$
		Lateral	$4.0 \pm 2.6 \text{mm}$

6 Discussion

The goal of this thesis was to detect the anatomical landmark points that define the position of the left ventricle in ultrasound images. This can be used to extract the standard anatomical views of 3D echocardiography, which, in consequence, can be used to initialize segmentation methods. Due to the fact that these standard anatomical views are required for simple visualization and initialization purposes, the detection method should be relatively fast. Also, because of the challenging image quality and the variation in LV appearance, robust methods are required. We intend to use this automatic detection method to automate an existing manual method for view extraction, which is also based on manually indicated landmarks [31, 32]. In that method, we find the left ventricle long-axis by locating the apex along with the mitral valve points.

The research method is used to detect the left ventricle landmark points in a fully automated way, without the need for initialization or interaction. The method may be applied for detecting landmark points in multiple 2D views of the 3D set (approximate two-chamber, four-chamber, short axis). This makes it suitable for our intended use, resulting in a 3D application, which has not been done before in similar approaches.

6.1 Multilevel approach

We decided to follow a multilevel approach, in which the global location of the ventricle with the landmarks is found first, and then rotation and scaling are found and further refined, to locate the landmark positions. We tested the detection accuracy in two of the standard anatomical views of echocardiography: the two (2C) and four chamber (4C) views.

The method produces a reasonably accurate estimation for the locations of the points. We can detect the points in both the 2C and 4C views. There is an acceptable error of 7.1 ± 6.7 mm for the apex, 5.6 ± 2.7 mm for the septal, 4.0 ± 2.6 mm for the lateral, 5.8 ± 3.5 mm for the anterior and 4.5 ± 3.1 mm for the inferior point. The multilevel approach is justified, since the results are improving from stage to stage. However, we noticed that the improvement from the second to the third classifier in some cases is minimal (Figure 5.8, Table 3, Table 4). The second classifier is already quite accurate, so that it is difficult for the fine classifier to improve the scores. Additionally, generating new candidates from the second to the third stage of the cascade in fine steps resulted in image differences that were hard even for an expert to distinguish and classify. If further improvement would be needed, the parameter choices or features for the third stage should be reconsidered.

Compared to research conducted in similar applications [28, 29], one of the challenges was to overcome the limited image database, as we had relatively few 3D sets of patients to cover the expected image variability. In order to do so, we added artificial variations and noticed that the ROC curve with the variation (Figure 5.1) was better versus the one without it (Figure 5.6). This may be enough for the purposes of a thesis project, but based on what we see in literature [36, 39] methods like ours generally need large databases, if

they are to be applied in real life situations. However, although limited, we tried to include in our database images of different quality, to make it suitable for clinical practice (where differences in image quality can be expected).

The current implementation is very slow. In general, a cascade of classifiers is used to give faster detection rates [30], [29]. In our case, the training and detection times were fast, but going from the one stage of the cascade to the next was time consuming, because of the amount of new candidates that were generated each time and current implementation issues. We wanted to be as general as possible in our approach, to allow an easy change of the classifier parameters, thus we did not seek for an optimal implementation in terms of speed. However, in order to accelerate the detection steps, we pruned away a lot of candidate regions, based on knowledge about the left ventricle's size. This resulted in faster detection, along with better results, since many of the outliers were discarded.

6.2 Choice of classifier parameters

One important aspect for the method's overall performance had to do with the choice of our training and detecting parameters (Table 1, Table 2). We chose the training parameters to achieve a certain detection accuracy; the parameters define the maximum accuracy that we can reach. So, for example, adding artificial variation of $\pm 4\text{mm}$ translation for the third classifier means that we can expect an error approximately at 4mm, which, compared to our ground truth, was considered realistic enough. Unfortunately, since only one manual annotation was used it is unclear what the quality and accuracy of our annotation is. It is our impression that future approaches should start with calculating the interobserver and intraobserver errors on the used database, to get an indication on the quality of the manual annotation, while that way we could also result in different parameter choices.

One basic difference between this implementation and similar ones, like the one proposed by Viola and Jones [30], is that instead of filtering our positive and negative sub-images with as many Haar feature types as possible and in as many sizes as possible, we used specifically six of them in three fixed sizes (smaller size as we proceed in the cascade in a coarse to fine scheme). This was done, as explained, to simplify things and reduce computational demands. When we tried to be more general and let the classifier decide on which sizes to use, we run either into memory issues (too much data for MATLAB to handle) or long training times. Unfortunately, there is no certain way to know which can perform best and in what size. There is no indication upon that on literature, so we based our choice on experiments that we conducted using OpenCV, which uses many types of Haar features and in all possible sizes (depending on the size of the given image). However, the nature of the problem is such that adding different Haar feature types would probably not result in significant improvement. The regions that we are looking for are of limited variation, so complex Haar features probably would just complicate things. One could opt, though, for better performing feature categories. We know, for example, that Lu et al. [39] used also steerable features. We chose to use Haar-like features because they have been previously used in similar applications, and can be computed

fast. The first indications we got when we tried other types widely used in image processing (SIFT, HOG), were poor due to speckle noise.

6.3 Comparison with literature

In Table 5 we compare our method to the one of Van Stralen [40], using exactly the same database. In all comparison tables we are giving our best performing results. Van Stralen’s method uses a combination of the Hough transform and dynamic programming, which results in a 3D application (whereas ours is 2D). In terms of accuracy we see that we performed better. But a fair comparison is difficult here. Van Stralen’s method detects the long axis and mitral valve center in 3D. First the LAX is detected and based on that the mitral valve plane is taken as the perpendicular circular ring, touching the left ventricle bottom endocardial border. The fact that the plane is considered always perpendicular to the LAX, which is not the case, is probably responsible for the difference in the accuracy of detecting the mitral valve center and the other mitral valve points (Table 5, Table 6). Moreover, Van Stralen’s method does not detect rotation. As a consequence, the orientation angles for the 2C and 4C planes are not known and a standard angle at 0° and 90° is always considered as the right one. Our method was evaluated in optimal 2D planes, and a 2D distance from the true landmarks is reported. To compensate this difference, we project the detected points from Van Stralen’s method on the correct plane. Our current implementation is much slower than Van Stralen’s method that needed only a few seconds to estimate the landmark positions.

Table 5 Compare results with Van Stralen [40]

	Proposed 2C	Van Stralen [40] 2C	Proposed 4C	Van Stralen [40] 4C
Apex	7.1±6.7mm	14.7±8.6mm	7.9±7.1mm	15.0±8.3mm
Septal/Anterior	5.8±3.5mm	11.2±6.0mm	5.6±2.7mm	13.7±7.7mm
Lateral/Inferior	4.5±3.1mm	13.0±5.7mm	4.0±2.6mm	13.3±6.8mm

Lu et al. [39] presented a data driven approach for the detection of standard view planes (apical 4C, 2C, 3C and short axis basal, middle and apex plane), similar to ours. They also use a cascade of three classifiers, which are trained based on boosting techniques. The coarse classifier is using Haar wavelet-like feature types, like we do, but the next two are using steerable features [41]. They claim that the latter are at the detailed level computationally inexpensive, compared to Haar features. Finally, their cascade was structured in a different way. The coarse classifier is used as a detector for the apical 4C view and its outcome is used to limit the search region for fine parameter estimation for in the other 2D planes individually. They compute initial plane parameters for the sought planes combining the detected parameters of the coarse classifier and empirical statistics of their pose parameters. Finer classifiers are then used for each plane separately. Lu’s method results in smaller errors than our method does. This may be due to better performance of the steerable features or the structure of the cascade. We have to note though, that their database was much larger (326 patients on which a four-fold cross-

validation was performed) and as consequence, adding artificial variation was not necessary.

Furthermore, some approaches which also identify the sought landmark points in addition to the long axis or the apical views' orientation angles have been published. Leung et al. [31] presented an approach for registration of the left ventricle in stress echocardiography that also detected the sought landmark points and Orderud et al. [42] presented an approach for alignment of standard apical and short-axis slices in 3D echocardiography. Compared to the results given by Orderud et al. [42] and Van Stralen [40] we find lower errors for the apex and mitral valve center (Table 6). Leung et al. [31] outperform our method, but we have to take into consideration that they aimed at registering markers in stress images, given the markers in the rest image, instead of detecting the standard views, which might have influenced the results. Once more we have to note that our approach works in 2D cross-sections, while all the rest are applied on the 3D data, so our result may be a bit biased.

Table 6 Comparison with existing methods

	Apex	Mitral Valve
Proposed	7.1±6.7mm	4.8±2.3mm
Orderud [42]	8.4±3.5mm	3.6±1.8mm
Lu [39]	4.5±3.5mm	3.6±3.1mm
Van Stralen [40]	14.7±8.6mm	8.4±5.7mm
Leung [31]	7.6±4.8mm	4.5±2.9mm
Interobserver [31]	7.1±2.9mm	3.8±1.5mm
Intraobserver [31]	5.2±2.0mm	3.3±1.5mm

We also compare our results to the interobserver and intraobserver errors given by Leung et al. [31]. They note that two independent observers indicate the long axis and 4C direction, as well as the aorta and the RV-attachment in end-diastole and end-systole. The intraobserver variability in the apex and mitral valve centre is defined as the average of Euclidean distances between points annotated twice by the same observer. However, these observer errors were measured on a smaller database (20 patients). The interobserver variability is similarly defined, as the average of differences in the mean annotation of each observer. The fact that our errors are close to those provided by the interobserver and intraobserver variability is very promising, since these are the errors in our ground truth of manual annotation (following the same protocol as in this study).

6.4 Limitations

An important issue for our project was to create an algorithm that could work relatively fast, thus suitable for real life applications. The time for both processing of the images and evaluating the method varies, depending on the classifier stage. The first classifier

needs 30 minutes for training, while the second and third one need at around 45. However, preprocessing (calculating feature values for training and tuning/testing set) is taking half a day for the first classifier and one day for the others. The second and third stages need 2-3 days to evaluate (calculating the point-to-point errors for the detecting set). The evaluation time was greatly influenced by the number of candidates that were passing from each stage to the other (Table 3, Table 4), since each time the Haar values are being calculated for all the candidates. However, as we have already mentioned, we kept our method's frame as general as possible, so we could research more on the classifier parameters, rather than speeding up the process. An implementation aimed at a specific application would use standard classifiers. We can easily map which features the classifiers use, so we could calculate only those Haar values. The current implementation, is calculating, for instance, at the case of the second classifier, 4 Haar feature types, for a template of 120x80 pixels resulting in 38400 calculations, while in the end 250 pixels are used. Going from 38400 to 250 calculations is already speeding the process at a factor of 150. Furthermore, our work with OpenCv showed that the computational time using C++ code can be greatly reduced: OpenCV needs a couple of hours to train the classifiers and a few seconds for detecting, including the time to calculate all feature values. We are confident that a real time application is feasible with this approach, since similar methods for face detection are being used in digital cameras.

Overfitting was another hurdle that we had to overcome. As discussed in literature for similar projects [29], complex classification problems may suffer from overtraining. In our case the third classifier, where the differences were too detailed, was overtrained at first, thus incapable of classifying our evaluation sets (Figure 5.4). As already explained, reducing the classifier complexity solved that issue, but we should always keep that in mind for future applications.

Finally, another limitation of our work lies in the fact that we are always working on the ideal 2D image plane, defined by the manual annotated 3D landmark points. We have trained the cascades to work in many different translations, rotations and scales so we can expect that they will detect the most probable locations for the landmark points in random planes. Although we don't have any specific indication concerning the accuracy of that detection, we believe that using these classifiers in an iterative scheme would result in the right planes, thus accurate detection rates. We base that belief on our experience using the manual method.

6.5 Conclusions

We showed that using our proposed approach, the anatomical landmark points can be detected accurately. If we consider that our errors were comparable to the interobserver and intraobserver variability, it appears that our approach is really promising. The algorithm proved to be accurate and robust enough in both planes, so it could be combined with the already existing manual technique to extract the views and result to a fully automated method of detecting the left ventricle's location. This would be possible if we include in our method a third cascade of classifiers, trained to work in the short axis plane, which would work as a detector for the orientation angles of the 2C and 4C planes.

Our original goal was to propose a method that can work in the three planes, but this proved to be too optimistic for the time span of a master thesis project. Additionally, we were opting for a fast application, which is possible but was not achieved yet due to implementation choices.

7 Future Research

Some possible improvements for the method have already been discussed. One of the most important improvements has to do with the size of the database to use. Using a larger database may improve the cascade's performance. As already explained, since we could not do so, we decided to create artificial variation. We believe that our classifier became more robust that way, but this also resulted in a reduction in accuracy. The differences among our examples were not natural, and maybe harder to distinguish, which probably led to overfitting. So, using larger databases will probably result in more robust classifiers and furthermore will help us avoid overfitting issues.

A next step would be to implement a similar cascade for the short-axis plane. In that way a full three step scheme would be ready for use in combination with the previously proposed manual method [31]. Apart from the three cascades, trained to work on the exact standard anatomical views' planes (2C, 4C, SAX), additional cascades for random planes could be examined. Normally, the manual methods needs 4-6 iterations before the sought views are found [31]. The annotated images produced by these intermediate steps are available, and a classifier trained to work in these intermediate steps may prove to be more accurate and robust than the final one. However, our first impression is that the current classifiers can work on the intermediate steps as well, as they will give the most probable position for the landmark points in the given plane. These could be used for further correcting the extracted plane, until the algorithm converges.

Working with different feature types could also be considered. Lu et al. [39] propose a method that performs better than ours and one of the differences is that they do not rely only on Haar like feature types. However, even if we consider Haar features as reliable enough, experimenting on their size, or their type would also be interesting. In any case, issues like how their size or their type influences the classifier's accuracy and robustness have not been covered by literature.

Finally, the overall cascade structure could be different. As we explained the accuracy improvement from the second to the third classifier was not so significant. This means that perhaps different parameters could be used for both of them, or only the second one may be used. Calculating interobserver and intraobserver errors for the specific database would facilitate that procedure. The ground truth would be different, changing the accuracy that we aim at.

References

- [1] H. Feigenbaum, *Echocardiography*, 4th ed. Philadelphia: Lea and Febiger, 1986.
- [2] N. C. Nanda, J. Kisslo, R. Lang, N. Pandian, T. Marwick, G. Shirali, and G. Kelly, "Examination protocol for three-dimensional echocardiography," *Echocardiography*, vol. 21, pp. 763-768, 2004.
- [3] C. C. Jaffe, P. J. Lynch, and P. Simon, "Atlas of Echocardiography ": Yale University, 1999.
- [4] G. Kan, C. A. Visser, K. I. Lie, and D. Durrer, "Early two-dimensional echocardiographic measurement of left ventricular ejection fraction in acute myocardial infarction " *European Heart Journal*, vol. 5, pp. 210-217, July 1984.
- [5] R. S. Bardelen, H. P. K uhl, S. Mohr-Kahaly, and A. Franke, "Second-generation real-time three-dimensional echocardiography," *Zeitschrift f ur Kardiologie*, vol. 93, pp. 56-64, 2004.
- [6] A. F. Frangi, W. J. Niessen, and M. A. Viergever, "Three-dimensional modeling for functional analysis of cardiac images, a review," *IEEE Trans. Med. Imag.*, vol. 20, pp. 2-25, Jan 2001.
- [7] R. M. Lang, V. Mor-Avi, L. Sugeng, P. S. Nieman, and D. J. Sahn, "Three-dimensional echocardiography: the benefits of the additional dimension," *J Am Coll Cardiol*, vol. 48, pp. 2053-69, Nov 21 2006.
- [8] J. A. Panza, "Real-time three-dimensional echocardiography: an overview," *International Journal of Cardiovascular Imaging*, vol. 17, pp. 227-235, 2001.
- [9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed.: Pearson Prentice Hall, 2008.
- [10] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, pp. 1-36, Mar. 1998.
- [11] R. Shekhar and V. Zagrodsky, "Mutual information-based rigid and non-rigid registration of ultrasound volumes," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 9-22, Jan. 2002.
- [12] R. Shekhar, V. Zagrodsky, M. J. Garcia, and J. D. Thomas, "Registration of real-time 3-D ultrasound images of the heart for novel 3-D stress echocardiography," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 1141-1149, sept 2004.
- [13] D. Rueckert, L. I. Sonoda, C. Hayes, A. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: applications to breast MR images," *IEEE Transactions on Medical Imaging*, vol. 18, pp. 712-721, Aug. 1999.
- [14] T. M akel a, P. Clarysse, O. Sipil a, N. Pauna, Q. C. Pham, T. Katila, and I. E. Magnin, "A review of cardiac image registration methods," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1011-1021, Sept. 2002.
- [15] J. A. Little and D. J. Hawkes, "The registration of multiple medical images acquired from a single subject: why, how, what next?," *Statistical Methods in Medical Research*, vol. 6, pp. 239-265, 1997.
- [16] J. A. Noble and D. Boukerroui, "Ultrasound Image Segmentation: A Survey," *IEEE Trans. Med. Imag.*, vol. 25, pp. 987-1010, Aug 2006.

- [17] E. D. Angelini, A. F. Laine, S. Takuma, J. W. Holmes, and S. Homma, "LV volume quantification via spatiotemporal analysis of real-time 3-D echocardiography," *IEEE Trans Med Imaging*, vol. 20, pp. 457-69, Jun 2001.
- [18] E. D. Angelini, S. Homma, G. Pearson, J. W. Holmes, and A. F. Laine, "Segmentation of real-time three-dimensional ultrasound for quantification of ventricular function: a clinical study on right and left ventricles," *Ultrasound Med Biol*, vol. 31, pp. 1143-58, Sep 2005.
- [19] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 158 - 175, Feb 1995.
- [20] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681-685, Jun 2001.
- [21] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision* vol. 60, pp. 91-110, November 2004.
- [22] C. Kimme, D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," *Commun ACM*, vol. 18, pp. 120 - 122 1975.
- [23] B. Solaiman, B. Burdsall, and C. Roux, "Hough transform and uncertainty handling. Application to circular object detection in ultrasound medical images," *ICIP 98*, vol. 3, pp. 828-831, 1998.
- [24] S. Golemati, J. Stoitsis, E. G. Sifakis, T. Balkizas, and K. S. Nikita, "Using the Hough transform to segment ultrasound images of longitudinal and transverse sections of the carotid artery," *Ultrasound Med Biol*, vol. 33, pp. 1918-32, Dec 2007.
- [25] A. Muller, A. Neitmann, N. Merkle, J. Wohrle, V. Hombach, and H. A. Kestler, "Contour detection of short axis slice MR images for contraction irregularity assessment," *Computers in Cardiology*, pp. 21- 24, Sept. 25-28 2005.
- [26] Q. Chen, M. Defrise, and F. Deconinck, "Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1156-1168, , Dec. 1994.
- [27] M. Ma, J. G. Bosch, J. H. C. Reiber, and B. P. F. Lelieveldt, "Fully Automatic Estimation of Object Pose for Segmentation Initialization: Application to Cardiac MR and Echocardiography Images," *SPIE Medical Imaging*, vol. 6512, p. 65123J, Feb 2007.
- [28] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta, "Database-Guided Segmentation of Anatomical Structures with Complex Appearance," *Proc. IEEE Computer society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 429-436, 20-25 June 2005.
- [29] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu, "Detection and Measurement of Fetal Anatomies from Ultrasound Images using a Constrained Probabilistic Boosting Tree," *IEEE Transactions on Medical Imaging*, vol. 27, pp. 1342-1355, September 2008.
- [30] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Proc. IEEE Conf Computer Vision and Pattern Recognition*, 2001.

- [31] K. Y. E. Leung, M. van Stralen, A. Nemes, M. M. Voormolen, G. van Burken, M. L. Geleijnse, F. J. ten Cate, J. H. C. Reiber, N. de Jong, A. F. W. van der Steen, and J. G. Bosch, "Sparse Registration for Three-Dimensional Stress Echocardiography," *Medical Imaging, IEEE Transactions on*, vol. 27, pp. 1568-1579, 2008.
- [32] A. Nemes, K. Y. E. Leung, G. van Burken, M. van Stralen, J. G. Bosch, O. I. I. Soliman, B. J. Krenning, W. B. Vletter, F. J. ten Cate, and M. L. Geleijnse, "Side-by-Side Viewing of Anatomically Aligned Left Ventricular Segments in Three-Dimensional Stress Echocardiography," *Echocardiography*, vol. 26, pp. 189-195, February 2009.
- [33] C. P. Papageorgiou, "A trainable system for object detection in images and video sequences," Massachusetts Institute of Technology, 2000, p. 1.
- [34] Y. Freund and E. R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*: Springer-Verlag, 1995.
- [35] K. H. Zou, A. J. O'Malley, and L. Mauri, "Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models," *American Heart Association*, vol. 115, pp. 654-657, 2007.
- [36] Y. Zheng, X. Lu, B. Georgescu, A. Littmann, E. Mueller, and D. Comaniciu, "Automatic left ventricle detection in MRI images using marginal space learning and component-based voting," 2009, p. 725906.
- [37] T. Loupas, W. N. McDicken, and P. L. Allan, "An adaptive weighted median filter for speckle suppression in medical ultrasonic images," *Circuits and Systems, IEEE Transactions on*, vol. 36, pp. 129-135, 1989.
- [38] J. Montagnat, M. Sermesant, H. Delingette, G. Malandain, and N. Ayache, "Anisotropic filtering for model-based segmentation of 4D cylindrical echocardiographic images." vol. 24: Elsevier Science Inc., 2003, pp. 815-828.
- [39] X. Lu, B. Georgescu, Y. Zheng, J. Otsuki, and D. Comaniciu, "AutoMPR: Automatic detection of standard planes in 3D echocardiography," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, 2008, pp. 1279-1282.
- [40] M. Van Stralen, K. Y. E. Leung, M. M. Voormolen, N. De Jong, A. F. W. Van der Steen, J. H. C. Reiber, and J. G. Bosch, "Time Continuous Detection of the Left Ventricular Long Axis and the Mitral Valve Plane In 3-D Echocardiography," *Ultrasound in Medicine & Biology*, vol. 34, pp. 196-207, Feb 2008.
- [41] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Fast Automatic Heart Chamber Segmentation from 3D CT Data Using Marginal Space Learning and Steerable Features," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, pp. 1-8.
- [42] F. Orderud, H. Torp, and S. I. Rabben, "Automatic alignment of standard views in 3D echocardiograms using real-time tracking," *Ultrasonic Imaging and Signal Processing, SPIE*, vol. 7265, p. 72, Feb 2009.