NTNU

Norwegian University of
Science and Technology

Segmentation of Right Ventricle in 3D Ultrasound Recordings

Asbjørn Breivik Engås

Master of Science in Engineering Cybernetics

Submission date:	June 2008
Supervisor:	Hans Torp, ITK
Co-supervisor:	Fredrik Orderud, IDI

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The purpose of this thesis is to develop a deformable subdivision model for the right ventricle in the human heart. This model should subsequently be tested in various three-dimensional cardiac ultrasound recordings with the use of Kalman filter based real-time tracking. The model's ability to adjust to the shape of the right ventricle should be evaluated and compared to manually segmented right ventricles.

Assignment given: 07. January 2008
Supervisor: Hans Torp, ITK

Preface

This thesis has been written in the 10th semester of the Master of Technology program at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology. The thesis has been performed at the Department of Circulation and Medical Imaging, during the spring of 2008.

I would especially like to thank my supervisor Fredrik Orderud, for very helpful guidance and feedback. He is also the creator of the excellent Real-Time Contour Tracking Library. I will also like to thank Professor Hans Torp and the TVI/Strain group at the Department of Circulation and Medical Imaging for useful input and suggestions. Finally, I would thank my fellow master students at the Department of Circulation and Medical Imaging for creating a nice and constructive working environment.

Trondheim, June 8, 2008

Asbjørn Breivik Engås

Abstract

This thesis presents segmentation of the right ventricle of the heart in real-time tracking of 3D ultrasound recordings. A simple deformable model for the right ventricle is developed based on statistical data from manual segmentations, and the model has been tested out in a set of 3D ultrasound recordings and compared to manually segmented right ventricular volumes. The manual segmentation method with volume approximation is also developed. The segmentation tests on the recordings are performed with an already present Kalman filter based real-time contour tracking framework.

The ability of the models to fit to the shape of the right ventricle has been evaluated, and the resulting volume curves have been inspected. Deformable models of the right ventricle are constructed by placing nodes in an initial three-dimensional mesh, and subdivision schemes are applied to make smooth surfaces. There have also been experimented with models of different resolutions and initial positions.

A background study of right ventricular anatomy, subdivision, model-based segmentation and Kalman filter theory is included, and clinical applications of volume measurements in real-time are suggested.

The results of the segmentation are promising, and indicate that models adjust to the right ventricle during the heart beat.

Contents

1 Introduction 1

I Background theory 3

2 Heart physiology 5
2.1 Heart function . . . 5
2.2 Heart anatomy . . . 5
2.3 The heart chambers . . . 7
2.4 Right ventricle . . . 9
2.4.1 Anatomy . . . 9
2.4.2 Physiology . . . 12
2.4.3 Volume . . . 13
2.4.4 Potential Clinical Applications of Real-Time Volume Measurements . . . 16

3 Medical imaging 19
3.1 Ultrasound . . . 19
3.1.1 Echocardiography . . . 20
3.1.2 Transducers . . . 21
3.1.3 3D imaging . . . 22
3.1.4 Scanconversion . . . 24
3.1.5 Image quality . . . 26
3.2 MRI . . . 28
3.3 Right ventricle in echocardiography . . . 28

4 Model-based segmentation 31
4.1 The concept of model-based segmentation . . . 31
4.2 Deformable models . . . 32
4.3 Active contour models . . . 33

5 Smooth parametric surfaces **39**
5.1 Splines 40
5.2 Subdivision 41
5.3 Subdivision schemes 44
5.3.1 Catmull-Clark 46
5.3.2 Doo-Sabin 49

6 State-space modeling and estimation **53**
6.1 State-space model 53
6.2 Kalman filter 54
6.2.1 Extended Kalman Filter 57

II Methods and Tools **59**

7 Manual segmentation **61**
7.1 Extracting slices from the volume 61
7.2 Outlining Right Ventricular Contours Manually 63
7.3 Approximating the volume from slices 67

8 Three-dimensional geometrical model of the right ventricle **70**
8.1 Geometrical models 70
8.2 Manually shaped model 70
8.2.1 Normalizing the models 72
8.2.2 Mean shape 73

9 Real-Time Contour Tracking Library **75**
9.1 State estimation with deformable model 75
9.2 Edge measurements 77
9.3 Attractors 78

III Results **81**

10 Tracking of the right ventricle **83**
10.1 Training Set 83
10.2 Mean Shape Of The Manually Segmented Ventricles 84
10.3 Tracking of the right ventricle 84
10.3.1 Manual shaped models 85
10.3.2 Mean shape model 89
10.3.3 Edge profile experiment 92
10.3.4 Initial Position Experiment 94

10.3.5 Tracking of the right ventricle together with the left ventricle 96
10.4 Experiment with attractors 97

IV Discussion and conclusion 98

11 Discussion 99

12 Conclusion 103

13 Further work 105

Bibliography 107

V Appendices 112

A MATLAB-script for area calculation of polygon and volume approximation of manually segmented contour 113

B Node positions and face list for the mean right ventricle model 117

Chapter 1

Introduction

Cardiovascular diseases have become increasingly common and is today the number one cause of death in Norway [30]. In the recent years many studies have shown that the right ventricular function can be of a prognostic value for such diseases as right ventricular myocardial infarction, congenital heart disease and coronary artery disease [12].

Volume measurements of the right ventricle in two-dimensional echocardiography is limited due to the complex and the heavily varying shapes of the right ventricle. This is in great contrast to the much more easily modeled left ventricle. The right ventricle has an asymmetrical and crescentic shape and can often appear difficult to view in standardized two-dimensional echocardiographical views [17]. This has led to a stagnation in the development of the right ventricular volume measurements methods.

The development of three-dimensional echocardiography have created a new effort and interest in the attempt to measure the volume of the right ventricle. The 3D approach makes the volume measurements possible without any geometric assumptions, and the volume measurements has been an extremely active area of recent investigation [28]. In the United States, The National Heart, Lung and Blood Institute identified the right ventricular physiology as a priority in cardiovascular research [13].

The main goal for this thesis is to segment the right ventricle during the heart cycle in three-dimensional ultrasound recordings. This is based on the framework for real-time segmentation of the left ventricle developed by Fredrik Orderud [32]. The focus is pointed toward the development of initial models of the right ventricle which is being used with the approach of model-based segmentation. The model-based method have some advantages over more traditional low-level methods, because of the inherited knowledge of the geometrical shape that should be segmented and the segmentation can not be carried out only with the use of image features [44]. This is one step on the long way of automated computer vision in medical imaging.

This thesis investigates the effects of different initial deformable models of the right ventricle and includes a procedure for making several geometrical models of the right ventricle based on three-dimensional ultrasound recordings.

The report is divided into four parts: Background theory, Methods and Tools, Results, and Discussion and Conclusion. The background theory presents some theory about the heart with special emphasis on the right ventricular geometry and physiology, ultrasound theory with emphasis on three-dimensional echocardiography imaging. Further it contains chapters about model-based segmentation theory, subdivision surfaces, and state-space modeling together with Kalman filter theory. The method part presents a method for manual segmentation of the right ventricle and a method for volume approximation from manual segmented two-dimensional contours. Three-dimensional geometrical models are presented and prepared for automatic segmentation. A chapter about the Real-Time Contour Tracking Library (RCTL) which is used as a tool for segmentation is also included. The result part includes validation of the automated three-dimensional segmentation against two-dimensional manual segmentation, the effect of the number of edge detectors, and the effect of attractors in the segmentation. Finally the discussion, conclusion and suggestions for further work in the area of model-based real-time right ventricle segmentation is included.

Part I

Background theory

Chapter 2

Heart physiology

2.1 Heart function

The heart is one of the most essential organs in the human body, and it is the key organ for life. The four main functions of the heart as listed in [38] is

- **Generating blood pressure**. The blood pressure in the body is generated when the heart contracts, and provides the flow of the blood through the body.
- **Routing blood**. The heart directs the blood such that blood that is rich in oxygen goes through the body, and that blood with low oxygen content is routed to the lungs for oxygen filling.
- **Ensuring one-way blood flow**. The heart valves ensures that the blood only goes one way through the body
- **Regulating blood supply**. The heart rate changes due to the metabolic need in the body, during rest or exercise and other changes in the body.

2.2 Heart anatomy

The heart, shown in Figure 2.1 and 2.2, is an oval shaped organ. The main features are the *apex* at the tip of the heart and the *base* on the opposite

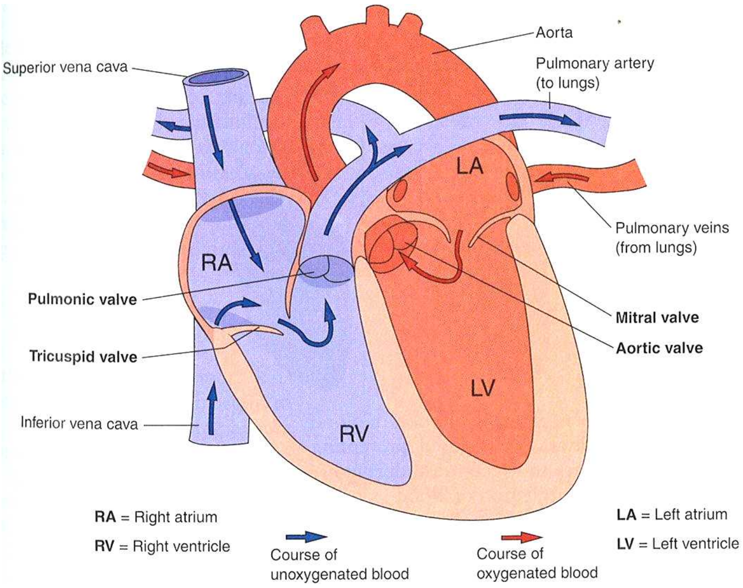


Figure 2.1: The human heart. The blue color indicates blood that has a low oxygene content, while the red color shows blood that has a high oxygen content. The right ventricle (RV) is shown on the lower left side of the figure. Figure from [2].

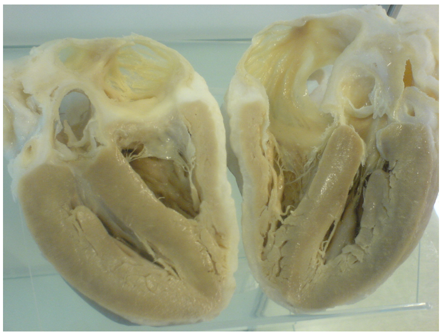


Figure 2.2: Photography of a human heart. This photography is of a specimen stored at the anatomical collection at St. Olavs Hospital in Trondheim. (Photo: Fredrik Orderud.)

side. The heart lies in the *pericardial cavity* which is sited behind the ribs and slightly to the left of the midline of the body, see Figure 2.4. The *pericardium*, which is a double-layered sac, encloses the heart. The two layers in the pericardium are the *fibrous pericardium* and the *serous pericardium*.

The fibrous pericardium is the outer layer and the serous pericardium is the innermost layer. The serous pericardium can again be divided into the *parietal pericardium* and the *epicardium* which lies in toward the heart. Between the parietal pericardium and the epicardium there is a pericardial cavity filled with *pericardial fluid*.

2.3 The heart chambers

The heart consists of four chambers; the right and left *atria* and the right and left *ventricles*, as shown in 2.1. The atria and the ventricles are separated from each other by the *coronary sulcus*.

The *superior vena cava* and the *anterior vena cava* delivers blood from the body to the right atrium from the top of the body and the bottom of the body respectively. The *pulmonary veins* delivers blood from the lungs to the left atrium, while the *pulmonary arteries* transports blood from the right ventricle to the lungs. The aorta extends from the left ventricle and delivers blood to the body.

The ventricles are the two chambers with the main responsibility for pumping the blood. The left ventricle is the stronger of the two ventricles and pumps the blood out in to the aorta where the blood continues further out in the body. The right ventricle pumps blood with low oxygen content into the lungs. The two ventricles are separated from each other by the *interventricular septum*.

Between each ventricle and its respective atrium the *atrioventricular valves* are located. The right ventricle and atrium is separated with the *tricuspid valve* (see Figure 2.1). This valve has three tips from which it has its name. The left ventricle and atrium is separated with the *mitral valve*. The valves allow blood to flow from the atria to the ventricles, but prevent flow the other way. This is achieved by the fact that the valves has a shape that makes it close when the pressure is higher in the ventricle than in the atrium, and open when the case is the opposite.

Between the aorta and the left ventricle the *aortic semilunar valve* is placed, and between the right ventricle and the pulmonary artery the *pulmonary semilunar valve* is placed. The functions of these valves are analogous to the function of the atrioventricular valves; they control that the flow in the veins goes in only one direction.

The cardiac cycle, i.e. the blood flow through the heart and the body is shown in Figure 2.3. The blood from the body enters the right atrium, which causes it to expand. Then the right atrium contracts, which make the triscupid valve to open, and the blood flows in to the right ventricle. The right ventricle then contracts and the blood flow in to the lungs through the pulmonary vein. An analogous procedure happens when the blood flows from the lungs to the left atrium and then further out in the left ventricle. It is important to note that the contractions of the two atria happen at the same time, and the contractions of the ventricles happen at the same time. The contraction of the atria is called *atrial systole*, and the relaxation are called *atrial diastole*. The contraction of the ventricles is referred to as *ventricular systole*, and similar the relaxation are called *ventricular diastole*. When we simply refer to the *systole* or *diastole* we refer to ventricular contraction or relaxation [38].

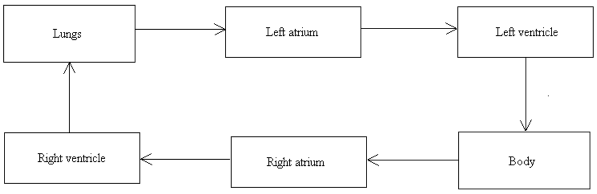


Figure 2.3: Illustration of the blood flow in the cardiac cycle. The blood is in the upper and the lower row at the same time.

The amount of blood pumped out by each of the ventricles, is called the *cardiac output (CO)* which is measured in milliliters per minute (ml/min). The amount of blood pumped for each ventricle during each diastole is called *stroke volume (SV)*, and is measured in milliliters per beat ($ml/beat$). The number of times the heart beats in one minute is called the *heart rate (HR)*. The relation between the three parameters is:

$$CO = SV \times HR \tag{2.1}$$

The stroke volume can be described as the difference between the volume in the ventricle in the end of the diastole and the volume in the end of the systole. A parameter which describes how much of the blood that is pumped out is called the *ejection fraction* and is defined as:

$$\text{Ejection fraction} = \frac{\text{Stroke volume}}{\text{End diastole volume}} \tag{2.2}$$

2.4 Right ventricle

The right ventricle is situated anterior on the heart and, in the normal situation, it lies right behind the sternum. The right ventricle is separated from the left ventricle by the interventricular septum. Figure 2.4 shows the position of the right ventricle in connection with the sternum and the ribs.

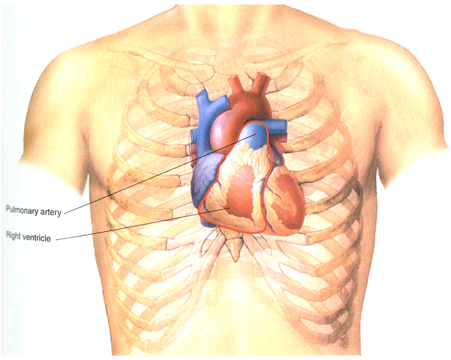


Figure 2.4: The position of the heart and the right ventricle in the body. The right ventricle lies right behind the sternum. This can make echogardiographic examinations difficult. Figure from [2].

2.4.1 Anatomy

The right ventricle are located at the ventral side of the heart. The overall shape of the right ventricle is triangular when it is viewed from the side, and crescent in the cross-sectional view, it is also asymmetric and heavy trabeculated. This gives a much more complex shape than the left ventricle which can be approximated as an ellipsoid. Some segmented right ventricles from CT images are illustrated in figure 2.5.

It is possible to describe the right ventricle by the notion of three parts [12], and this is illustrated in figure 2.6. Part 1 in figure 2.6 is the outflow part, also called the infundibulum or the conus arteriosus, and is a tubular muscular formation with the pulmonary valve on the top. The infudibulum is limited anferiorly by the free ventricular wall, posteriorly by the ventriculoinfundibular fold and by the outlet portion of the interventricular septum.

The part situated superior in the ventricle is the inflow part which contains the tricuspid valve, *chordae tendineae*, and papillary muscles, this part is shown as part 2 in figure 2.6. The chordae tendineae are thin but strong

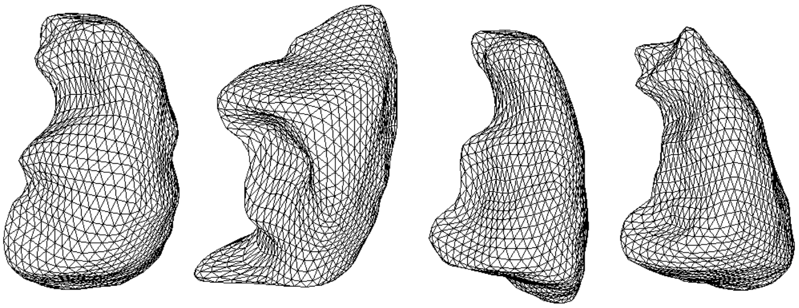


Figure 2.5: Segmented right ventricles from CT images. Notice the complex shape and the large variations between the ventricles. (The volumes are manually segmented ventricles taken from GE's cardiac atlas).

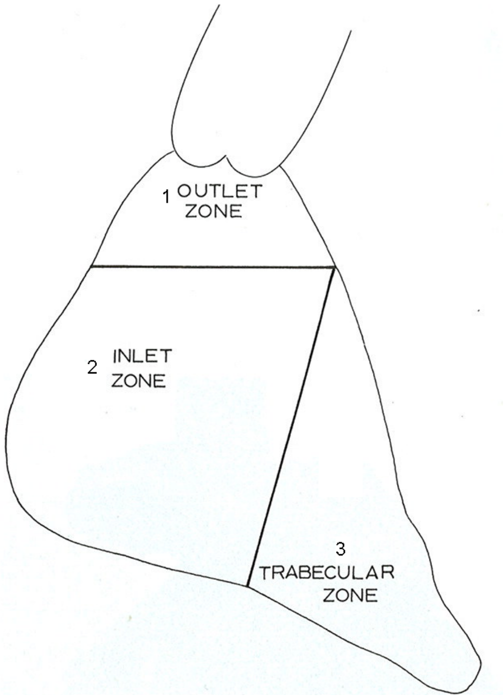


Figure 2.6: The right ventricle divided into three parts: 1) the outflow, 2) the inlet, 3) the trabeculated zone. The outflow part, or infundibulum, is a tubular muscular formation with the pulmonary valve on top. The inlet includes the tricuspid valves and extends until the implantation line of the papillary muscles. The trabuleculated zone extends from the apex to the insertion of the papillary muscles. Figure from [47].

connective tissue strings which attach the papillary muscles to the tricuspidal valve, see Figure 2.7.

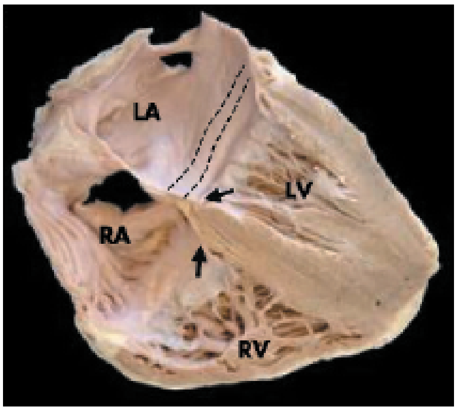


Figure 2.7: This figure illustrates the papillary muscles in the right ventricle. The papillary muscles are attached to the tricuspid valve and are responsible for valve movement. It illustrates also that the tricuspid valve is attached closer to the cardiac apex than the left ventricle. (RV = Right ventricle, RA = Right atrium, LV = Left ventricle, LA = Left atrium). Figure from [16].

The part nearest the apex is the trabeculated apical myocardium, which has very coarse trabeculations. Such as the large papillary muscles which is illustrated in Figure 2.7.

There are also other means of dividing the right ventricle into subparts [12], such as by means of the walls (free wall, septal wall and inferior). The free wall is the anterior wall of the ventricle, while the septal wall is the lateral wall and corresponds to the interventricular septum. The walls of the right ventricle are thinner than the walls of the left ventricle.

Between the tricuspid and the pulmanory semilunar valve, there are a ventriculoinfundibular fold which separates the two valves, see Figure 2.9. This is a characteristical feature of the right ventricle as the left ventricle has continuity between the mitral and the aortic semilunar valve.

Another characteristic feature of the right ventricle is the *moderator band* or septomarginal trabecula. The moderator band is one distinctive muscular band in the ventricle and is attached to the anterior papillary muscle. It lies on the right side of the interventricular septum and has two limbs embracing the body of the supraventricular crest, see Figure 2.8.

The internal features of the right ventricle is illustrated in figure 2.9. Notice the external pyramidal shape, and the large papillary muscles in the apex.

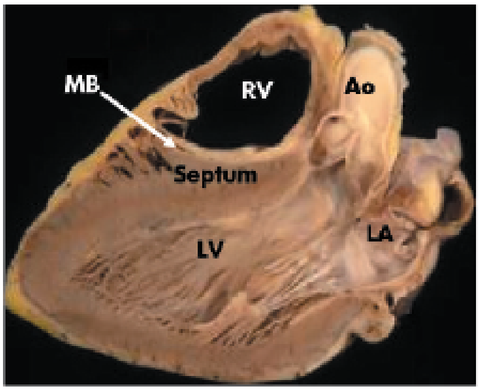


Figure 2.8: The moderator band (MB), or the septomarginal trabecula, is a distinctive muscular band in the ventricle. The moderator band is often a prominent feature in echocardiographic images. Figure from [16].

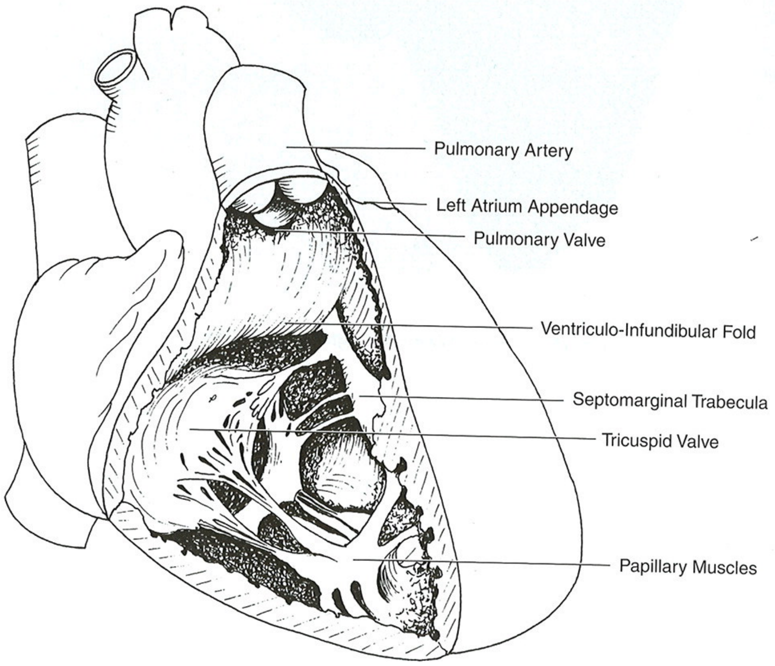


Figure 2.9: Internal features of the right ventricle. The prominent papillary muscles is seen in the apex part. The overall external shape appears as a rounded pyramidal shape, which are flat in the inferior part and more rounded in the superior part. Figure from [47].

2.4.2 Physiology

The right ventricle's primary function is to pump blood into the pulmonary trunk. When the right ventricle contracts, the pressure increases to approximately one-fifth of the pressure in the left ventricle [38], the two chambers does however pump the same average stroke volume due to the fact that they are connected in series. However the anatomical coupling of the right ventricle to the interventricular septum is complicating the understanding of the right ventricular function [37].

The contraction of the right ventricle is performed by three separate mechanisms [12]:

1. Inward movement of the free wall.
2. Contraction of the longitudinal fibers. The long axis shortens and pulls the tricuspid annulus against the apex.
3. Free wall traction at the attachment points secondary to left ventricular contraction.

The separate anatomic components contract asynchronously [28] and starts with the contraction of the inflow and the trabeculated myocardium part (Figure 2.6) and ends with the contraction of the infundibulum (outflow section). The infundibulum contraction lasts longer than the inflow region contraction [12]. There are no significant twisting and rotational movements in the right ventricle contraction, this is in contrast to the left ventricular contraction. In the pattern of the right ventricle contraction the predominant shortening is apex to base [49]. This often appears in echocardiography images as the apex has a very small movement.

2.4.3 Volume

Lorenz et. al [24] obtained the following quantities for right ventricular volume with cine MRI:

- RV end diastolic volume: 138 ± 40 ml (males: 157 ± 35 ml, females: 106 ± 24 ml)
- RV end systolic volume: 54 ± 21 ml (males: 63 ± 20 ml, females: 40 ± 14 ml)
- RV ejection fraction: $61 \pm 7\%$ (males: $60 \pm 7\%$ ml, females: $63 \pm 8\%$)
- RV stroke volume: 84 ± 24 ml (males: 95 ± 22 ml, females: 66 ± 16 ml)

The parameters are expressed with mean value ± one standard deviation. The stroke volume would be approximately the same as for the left ventricle due to the closed loop system of the heart and veins. However, the ejection fraction does not necessarily have to be the same as for the left ventricle, because the chambers end diastolic volume does not need to be the same.

2D echo volume measurements methods

There is no simple geometrical shape such as pyramidical, ellipsoidal, or cylindrical which can represent the right ventricle shape in a satisfying matter. The accuracy of volume estimation with two-dimensional echocardiography and geometrical assumption has been limited, both for methods based on Simpson's rule and area and length calculation [28]. The anatomical and physiological limitation of the right ventricle makes the volume measurement of the right ventricle challenging.

Several suggestions for calculating volume from two-dimensional echocardiography has been proposed:

Munoz et. al. developed a method for reconstructing the three-dimensional shape of the right ventricle from three orthogonal standard echocardiographic planes [28]. In [39] they state that RV volumes can be estimated from the projection of the chamber in two perpendicular planes

Other forms for estimating volume is by means of angiography, in where an area-length algorithm has been proposed [39]. In [7] they propose methods that make geometrical assumptions such as elliptical cross-section, pyramidal shape and prism shape assumptions.

Some attempts to make mathematical models which describes the geometric shape of the right ventricle. Because of large variations in the ventricle these does not yield exact results when just based on two-dimensional parameters. They could however be used as an basis for a deformable model used in model-based segmentation. These are the *crescentic cross-sectional* model proposed by Czegledy [7], and the *bent-ellipsoid model* proposed by Cao [5].

The crescentic cross-sectional model proposed by [7] is based on two circles with different radii. One circle specifies with radius R specifies the septum, and one other circle of radius r specifies the right ventricle free wall. Figure 2.10 illustrates this model, shown from the same view as figure 7.9. The free wall defining circle are changing in radius as a function of the length, making the curved shape of the free wall. The model can be constructed by specifying the length of the ventricle, the radius in the base and the radius of the two circles, all of these are parameters which can be obtained by two-dimensional echocardiography. The volume can then be calculated by a given formula [7].

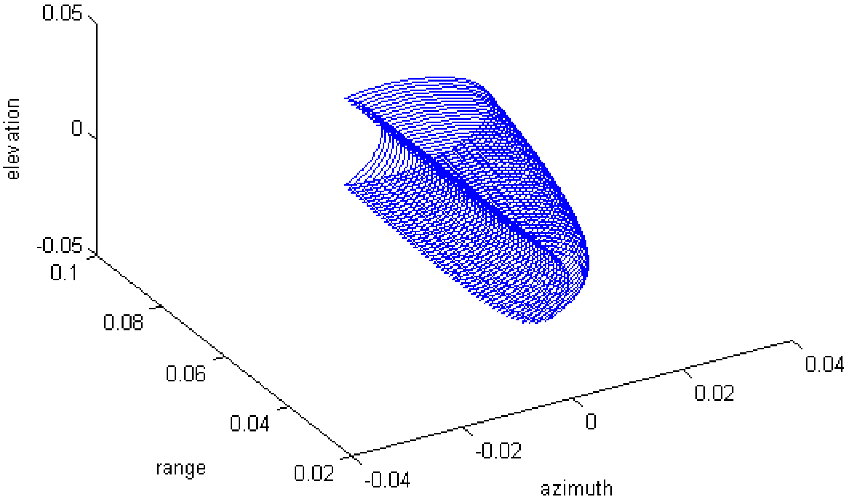


Figure 2.10: The crescentic cross-sectional model, translated along the long-axis to obtain the same view as in figure 7.9. The model consists of two circles, one which defines the septum and one which defines the free wall.

The bent ellipsoid model [5], uses a modified version of the ellipsoid formula:

$$\frac{(x - kz^2 - ly^2)^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \text{for } y \geq 0 \tag{2.3}$$

This model is specified by the variables a the equatorial radius along the x-axis, b the polar radius along the y-axis, c the equatorial radius along the z-axis, k which specifies how far the apex should be from the y-axis, and l which specifies how far the corners of the base should be from the y-axis. The bent ellipsoidal model is shown in figure 2.11.

3D echocardiography

With the development of three-dimensional echocardiography the use of this technology to assess the right ventricular volume and function has been suggested and are under investigation because there is no need for geometric assumptions [49]. The complex crescent shape in the short axis view of the right ventricle has made estimation of right ventricular volumes based on

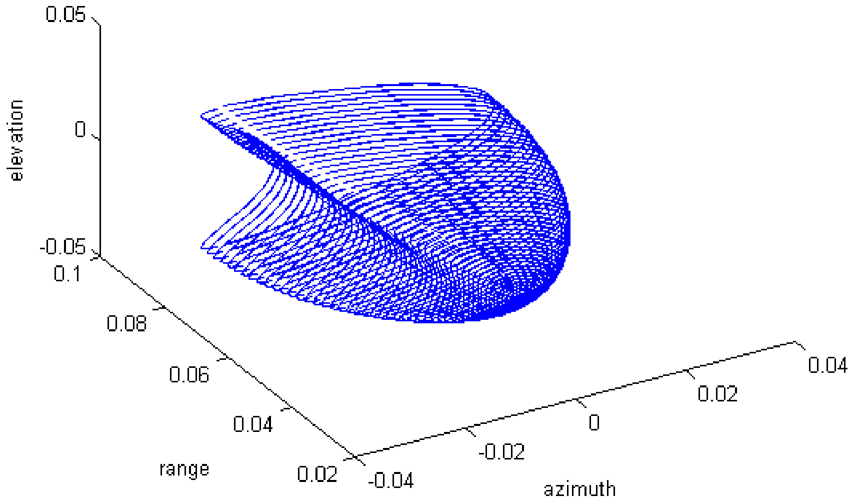


Figure 2.11: The bent ellipsoidal model, translated along the long-axis to obtain the same view as in figure 7.9. The model can be constructed with parameters obtained by two-dimensional echocardiography.

two-dimensional imaging and geometric modeling especially challenging. The ability of the three-dimensional echocardiogrophy to assess the volume directly without any geometrical assumptions have improved the accuracy and reproducibility than with previously proposed two-dimensional techniques [19].

2.4.4 Potential Clinical Applications of Real-Time Volume Measurements

In the past the importance of the right ventricular function has been underestimated in the evaluation of cardiovascular disease, but in the recent years many studies have shown a prognostic value [13].

In [28] several clinical applications of three-dimensional measurements of the right ventricle by echocardiography. These applications include monitoring of right ventricular function in children with lesions associated with right ventricular volume overload and/or pulmonary hypertension. Importance in the quantification of the effects of pulmonary hypertension and specific therapies on right ventricular function is also included as potential applications.

Further it is proposed that a monitoring of the right ventricular shape, function and volume can be used to assess some clinical conditions, such as ventricular ischemia, single ventricle, hypoplastic right ventricle, and post-operative tetralogy of Fallot [28].

These claims are backed up by [49] where it is proposed that knowledge of RV function has potential use in several clinical conditions such as pulmonary hypertension, congenital heart disease, and right ventricular dysplasia/cardiomyopathy. In addition [39] has proposed that it could be used for finding disorders such as ischemic heart disease, congenital heart disease, and pulmonary hypertension. In [37] the following diseases and disorders are listed: chronic obstructive pulmonary disease, congenital defects, and congestive heart failure.

Chapter 3

Medical imaging

3.1 Ultrasound

Ultrasound imaging is based on sending an acoustic pulse with a high frequency and then listening for echoes from structures where the acoustic impedance is different from the carrier medium. Ultrasound frequencies ranges from 20 kHz and up, these are frequencies that are beyond the audible area of the human ear. Ultrasound used for medical diagnostics has typically frequencies from 2 MHz to 10 MHz [1].

The ultrasound waves can be described by its *frequency*, *wavelength*, *period*, *amplitude*, *power*, *pulse length* and *intensity*. The period and amplitude is shown in Figure 3.1. The frequency is given as the reciprocal of the period. The wavelength is given by:

$$\lambda = \frac{c}{f} \tag{3.1}$$

where c is the speed of sound and f is the frequency. The power is the energy transfer in the beam and is proportional with the amplitude squared, A^2. The intensity is the average power per unit area that is incident perpendicular to the direction of propagation ($P = \frac{P_a v}{A}$) [45].

The *speed of sound* is a property of the medium, and is given by the relationship:

$$c = \frac{1}{\sqrt{\rho \kappa}} \tag{3.2}$$

where ρ is the mass density in the medium and κ is the volume compressibility. In diagnostic ultrasound the medium is human tissue, and the average

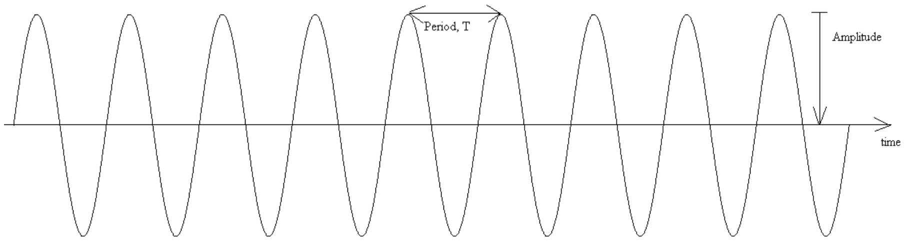


Figure 3.1: Soundwave parameters. An ultrasound wave is a sound wave with frequencies over $20kHz$. The frequency is given as the reciprocal of the period.

speed of sound in tissue is 1540 m/s. Another property of the medium is the acoustic impedance which is given by:

$$Z = \rho c \tag{3.3}$$

where ρ is the mass density of the medium and c is the speed of sound. The fact that different sorts of tissue have different acoustic impedance is the fundamental property of diagnostic ultrasound. As mentioned earlier, when the ultrasound wave hit structures with different acoustic impedance a part of the wave will be reflected and this can be exploited to identify structures in the tissue. The range from the ultrasound transducer to the echo-giving structure can be calculated from the time it takes from the pulse was transmitted and until it is received. This can be stated by the following expression [1]:

$$r = \frac{c\tau}{2} \tag{3.4}$$

where τ is the time from the transmission to the receiving and the factor $\frac{1}{2}$ originating from the fact that the wave has to travel the distance two times.

3.1.1 Echocardiography

Echocardiography is the application of diagnostic ultrasound on the heart. There are several advantages with using ultrasound for inspection of the heart versus other imaging modalities, such as MRI or CT. The ultrasound does not harm the patient neither does it cause any discomfort for the patient. The equipment can be portable and can be used under changing conditions. Echocardiography does also allow real-time imaging which makes it possible for the clinician to inspect the heart beats. The real-time recordings are also a crucial part of the results of this thesis.

Bones have very high acoustic impedance and will reflect a substantial part of the ultrasound waves and thus hide the tissue structures behind it. One of the main challenges when performing echocardiography is to be able to avoid the ribs to direct the waves directly on the heart.

The frequencies used in echocardiography are typically $2 - 2.5\ MHz$ for adults and $5 - 7.5\ MHz$ for pediatric inspection [31]. Higher frequencies give higher resolution, but the penetration depth decreases.

3.1.2 Transducers

The ultrasound waves are generated by an ultrasound transducer. A transducer is a device that converts one type of energy to another. In the diagnostical ultrasound case it is electrical energy to mechanical energy and vice versa, which means that the transducer can be used both to transmit and to receive an ultrasound signal. This is done with piezoelectric elements which have the ability that they expand and contract according to the voltage applied to them. If an oscillating voltage is applied, the elements will vibrate and produce waves in the medium they are applied to. Reversely, an incoming mechanical wave causes the elements to vibrate which again creates an electrical pulse.

The elements in the transducer can be arranged in an array. The elements in the array can vibrate independently, and according to Huygens principle [25] the sum of each wave created from each element will make up a full wave. Many different types of ultrasound transducers is on the market. The most common transducers in medical imaging are the linear array, curvilinear array and phased arrays [43].

The linear array are build up of many elements (300-400) placed contiguous on a line, but with only a few functioning at a time. These active elements make up the active aperture. Once one element has received the echo it is switched off and the next element in the array is switched on. The linear array is often used where large areas of skin is available such as imaging of the carotid artery. The curvilinear array is a sort of linear array where the arrays are curved slightly. With this arrangement it is possible to obtain a wide field, and therefore this sort of probe is used in fetal imaging.

For cardiac imaging the phased array is used. The ultrasound beams have to be steered between the ribs to access the area of interest, the opening between the ribs is approximately 20 mm. The phased array has a small aperture which means that the beams can be transmitted through a narrow area. For cardiac imaging it is important that the beams is steered in large angles to cover a sufficiently big area. The phased array transducer has few, but small elements.

3.1.3 3D imaging

A 3D ultrasound transducer consists of a two dimensional array of piezoelectric elements, such a transducer is illustrated in Figure 3.2. Each of these elements transmits and receives each own pulse, and this can be exploited to steer the ultrasound beam in the wanted direction [1].

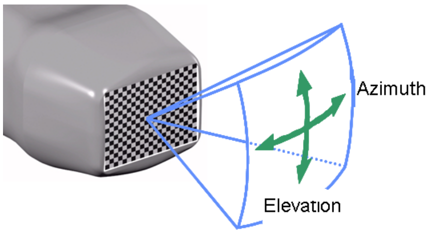


Figure 3.2: An ultrasound transducer used for three-dimensional imaging. The azimuth and elevation direction are indicated. The direction along the beam is called the radial or the range direction. Figure adapted from [46].

3D ultrasound recordings are recorded in a coordinate system which is referred to as *beamspace*. Beamspace is very similar to regular spherical coordinates, where the middle of the probe is defined as the origin. Figure 3.3 illustrates this coordinate system. The angle az between the origin and each beam direction is defined as the azimuth angle and it describes the azimuth direction. The angle el is called the elevation angle and the distance r is referred to as the range.

When we refer to the pulses in each azimuth direction we call them *beams*. In the elevation direction we refer to them as planes. The time from the pulse is transmitted until each echo is received is measured and inserted in equation 3.4, and the distance obtained is called the *range*. Thus each plane can be arranged as a table of beams and ranges.

The ultrasound recordings used in this thesis are stored in a four-dimensional table. In this table the first dimension describes each beam, the second each range, the third each elevation plane and the fourth describes each point in

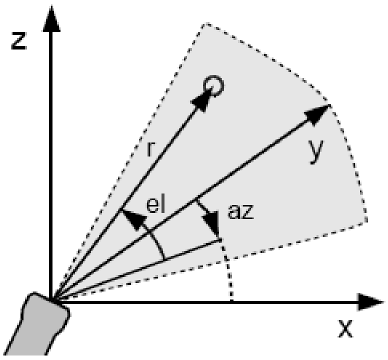


Figure 3.3: Illustration of the two coordinate systems; beamspace and probespace. These coordinate systems are used in ultrasound recordings. (Figure: Fredrik Orderud).

time the image is made. This fourth dimension is called the *frame*. The data does also contain information about the azimuth angle for each beam, the elevation angle between each plane, the number of frames per second, and the minimum and maximum range.

The temporal resolution is a great challenge in three-dimensional ultrasound, due to the relatively slow sound velocity c in human tissue. This imposes a problem in echocardiography where it is a need for a good temporal resolution to image the heart cycle. Therefore some techniques has been developed to attempt to deal with these difficulties.

Multiple line acquisition

Multiple line acquisition is a three-dimensional acquisition method for increasing the temporal solution. With the multiple line acquisition technique one receives multiple beams for every transmitted beam. The transmitted beam are broad to cover a specific sector in the human body. Then the transducer receives echoes from different directions, which increases the possible number of beams per time. Thus the frame rate is increased while the resolution is preserved.

Full volume acquisition

In full volume acquisition one subvolume is acquired for each new frame, this is an attempt to obtain large volumes without reducing the resolution. This volumes are then stitched together to form a full volume over a heart cycles. This method repeats and replaces the oldest subvolumes for each heartbeat. It is possible to make a full volume because the heartbeats are essentially equal. The subvolume acquisition is often performed by trigging it to an electrocardiogram (ECG).

3.1.4 Scanconversion

If we just plot the obtained table with signals for one plane and one frame as it is stored we get an image like the one seen to the left in Figure 3.4. This gives a wrong impression of the geometry of the received signal. This is because the ultrasound data is recorded in a spherical coordinate system, while it is desired to display the data in a cartesian coordinate system, refer to Figure 3.3. The desired image is shown to the right of Figure 3.4. The process of converting the signal in beamspace into the coordinate system *probespace* is called *scanconversion* and it is illustrated in Figure 3.5. Probespace is a cartesian coordinate system with the origin placed in the middle of the ultrasound transducer. Its definition is slightly different from an ordinary coordinate system, because it is desired to have positive y-axis in the forward direction from the transducer.

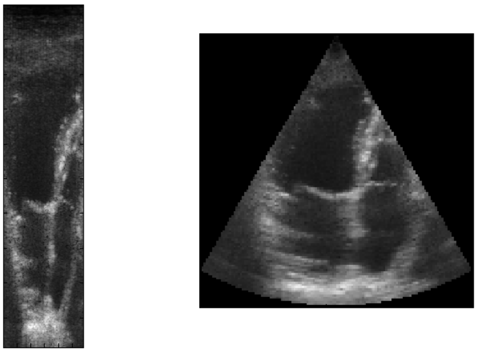


Figure 3.4: Illustration of the result of scanconversion. To the left an image taken directly from the acquired signals in beamspace. To the right the scanconverted image, illustrating the correct geometry

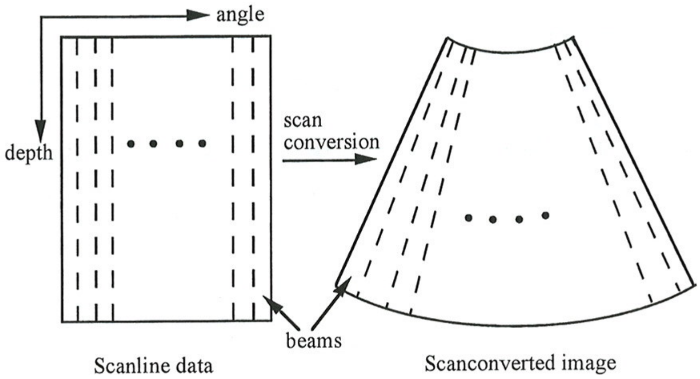


Figure 3.5: The concept of scanconversion. The beams recorded can not be displayed as shown to the left, but has to be placed in the geometry shown to the right to illustrate the correct geometry. Figure from [1].

The conversion from beamspace to probespace can be performed by converting the data acquired in spherical coordinates to a cartesian frame from the following equations:

$$x = r \cos el \sin az \tag{3.5}$$

$$y = r \sin el \tag{3.6}$$

$$z = r \cos el \cos az \tag{3.7}$$

where (x, y, z) is a coordinate in probespace, and (r, az, el) is a coordinate in beamspace.

Because the data acquired in beamspace is from a discrete rectangular grid, and the conversion from beamspace to probespace often does not yield integer coordinates, direct conversion from beamspace to probespace would yield a image which is not satisfactory. This can particularly be observed in the outer points where the image contains "gaps". Figure 3.6 shows these effects.

The solution to these problems is interpolation. In Figure 3.4 *linear interpolation* has been used. The concept of linear interpolation is that at each probespace coordinate the intensity will be a weighted sum of the 8 nearest samples in the beamspace sample grid. The weights are based upon the distance from the point in probespace to the integer point in the beamspace sample grid.

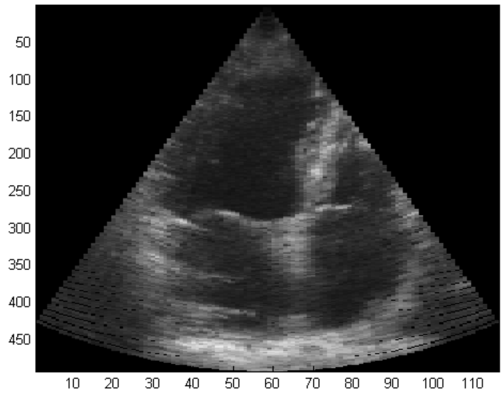


Figure 3.6: Scanconversion without interpolation. Notice the black areas in the lower part of the image. Compare with the image to the right in Figure 3.4.

3.1.5 Image quality

Ultrasound are inferior to other imaging modalities when it comes to image quality. Because of the fact that the ultrasound wave have to go through human tissue and this creates a serious amount of distortion in the ultrasound image. The human tissue consists of different types of tissue that has different speed of sound and this have an can effect on the focus of the beam. This effect is called aberration. The different types of tissue can also lead to multiple reflections, so called reverbations. Bones and muscles have a large acoustic impedance and can introduce shadows in the ultrasound image. This is especially a problem in echocardiography where shadowing from the lungs are a severe problem.

Speckle

When the beam hits objects that have a size that is much less than the wavelength of the beam, a phenomenon called speckle occurs. This means that the beam is reflected in many directions and interfering with beams reflected from other objects [1]. Speckle makes the characteristic granular look of the ultrasound image. An illustration of speckle is shown in Figure 3.7.

The speckle effect is deterministic, i.e. if the environment for two experiments is the same, the same speckle pattern will occur. One problem with

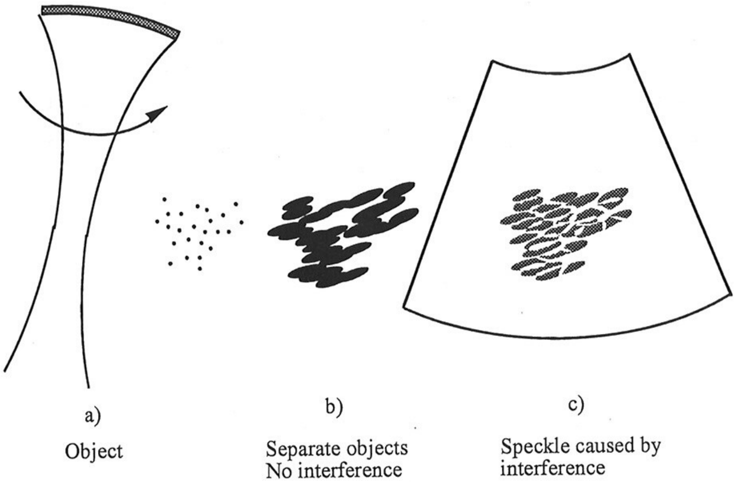


Figure 3.7: The speckle effect. When the ultrasound beam hits structures that are smaller than the wavelength of the beam, the beam is reflected in all directions, and the beams interferes with each other. Figure taken from [1].

the speckle effect is that the artifacts can appear as real structures for the viewer of the images.

Resolution

The resolution of the ultrasound image is decided by the point spread function and is a measure for how small details that can be imaged. The point spread function is the system's ability to separate small adjacent objects. There are two different kinds of resolution, lateral and radial. The lateral resolution is the resolution across the ultrasound beam, in the azimuth direction of Figure 3.2. This resolution is given by [50]:

$$u_R = f_{\#}\lambda = \frac{R}{a}\frac{c}{f_0} \tag{3.8}$$

where $f_{\#}$ is called the f-number, R is focal depth, a is the aperture, λ is the wavelength, c is the speed of sound, and f_0 is the center frequency for the transmitted ultrasound pulse. From this equation it is appearent that a high frequency gives a good resolution in the lateral and elevation direction. The radial resolution is given by:

$$u_z = \frac{c\tau}{2} = \frac{c}{2B} \tag{3.9}$$

where τ is the pulse length, c is the speed of sound, and B is the bandwidth. This resolution equation corresponds to equation 3.4. We can see that a high bandwidth gives a good radial resolution.

3.2 MRI

Magnetic resonance imaging (MRI) is often viewed upon as the "gold standard" when it comes to imaging structures of the body into great detail. The magnetic resonance images does not contain any speckle artifacts and the images has a clear appearance compared to ultrasound images. MRI is however a very expensive form of medical imaging and is not portable because it requires a large apparatus.

When magnetic resonance imaging are applied, the patient is placed in a strong static magnetic field created by a large magnet, which often has a high strength as 1.5 Teslas. A rotating field at radio frequency is applied in a perpendicular plane to the static field. This field has a frequency which is equal to a property of the material. This frequency is called the Larmor frequency. The rotating field excites the atoms in the body, because of spin in the nucleus. When the atom has been excited the rotating field is shut off and the original magnetic moment of the nucleus goes back to equilibrium while it emits a signal. Two relaxation constants are detected and these are used for image formation and for discriminating between different tissue types. The resolution of the MRI is typically 1 mm [43].

3.3 Right ventricle in echocardiography

Fully identifying the right ventricle in echocardiography is challenging. This is the case for both two-dimensional and three-dimensional imaging. Even evaluation of the right ventricular function is challenging because of the complex geometry and the ventricle's thin walls. These are limitations that have affected the assessment of the right ventricular function in routine clinical controls [49].

The greatest challenges of the right ventricle in echocardiography is the thin walls, the complex shape and the heavy trabeculation. The thin walls makes wall-motion imaging difficult, the effects of the thin walls can also be seen in the image in Figure 7.6 where the free wall of the right ventricle has completely fallen out. The complex shape does not allow any geometric assumption for volume calculation or ejection fraction, which is possible in the left ventricle. The heavy trabeculation gives problems for outlining the endocardial borders accurately. These facts has resulted in that right ventricular function is not routinely inspected by echocardiography.

Segmentation methods for the right ventricle faces great challenges, the complex shape mentioned in the previous paragraph impose huge challenges. Segmentation methods should also include strategies for avoiding the papillary muscles, the moderator band and the trabeculations. From apical views reliable methods for finding the free wall is very difficult [29].

Chapter 4

Model-based segmentation

4.1 The concept of model-based segmentation

The scientific field of image processing and machine vision has experienced a very slow progress since it appearance in the mid-1960s. The impact of digital image processing upon its potential fields of application still remains below the expectations compared to other fields of computer science, e.g. programming languages [35] [51]. This can come from the fact that machine vision is a very ambitious science compared to the other fields, which goals are more down to earth. As stated in [51] the discipline consists of two types of skills; systematic knowledge of well-established techniques with very predictable performances, and heuristical talent, experience, and a good sense for the kind of procedure and for the sequence of procedures that are well-promising for coping with a given problem. The first type of skills can be acquired by reading textbooks and attending university courses, while the latter type is difficult to learn and is best acquired by experience by working together with experienced personnel. In other words; the disciplines shortcomings comes from the extremely powerful nature of human vision. So to most people which are capable of solving vision tasks much quicker and better than a computer, progress in the field of machine visions seems slow.

Image segmentation stands as one of the many remaining challenges in automatic computer vision. When it comes to automation of computer-based image segmentation the task has been very passing although the human visual system is generally capable of precise segmentation. One of the problems with automated image segmentation is that there have been a view that the image is self contained, i.e. all the necessary information needed for segmentation are inherent in the image. In fact Székely [44] states that this assumption is fundamentally wrong. Algorithms based on this fact, so-called *low-level methods* [26], are many. Examples of low-level methods are linear filtering, Fourier transform, thresholding, watershed methods and region growing algorithms [11].

Image segmentation problems in the real world, such as in medical imaging, does very often require much a priori information to be able to perform a satisfying segmentation. This kind of information contains a complexity that can be difficult to include in algorithms. Segmentation algorithms that uses some kind of model of the feature to extract for improvement of the segmentation are often called *model-based segmentation*. The models can be both deterministic and statistical. The use of this methods can be viewed as a step away from pure segmentation algorithms to pattern recognition problems.

Frangi et. al. states a few advantages of model-based methods of the right ventricle (and the left) over low-level methods [10].

1. The model itself can constrain the segmentation process that is illposed in nature owing to noise and image artifacts
2. Segmentation, image analysis and shape modeling are simultaneously addressed in a common framework.
3. Models can be coarse or detailed depending on the desired degree of abstraction
4. In some approaches, most of the chamber's shape can be explained with a few comprehensible parameters which can subsequently be used as cardiac indexes.

4.2 Deformable models

In medical imaging we are often exposed to large datasets, with complicated and often noisy structures. To make use of image recognition in medical images, segmentation has to be used efficient and accurate in order to end up with probable models. Human organs often has large variations, and the fact that one segmentation looks perfect is no guarantee that it is anatomically correct. Because of the presence of noise and image acquisition artifacts in many routinely acquired ultrasound images image information can be ill defined, unreliable, or missing. This is a problem for low-level methods which in this case would yield results that are not likely or useful.

Manually outlining of a contour can be considered as a "‘gold standard"’ for segmentation of features in medical images. This approach is however very time-consuming and labor-intensive. A contour as drawn by an expert human observer may not always correspond to the location of the strongest local image evidence [27], this motivates the approach of model-based segmentation in the segmentation problem of the right ventricle.

With a deformable model we specify a shape resembling the shape we want the segmented feature to detect. By use of geometry constraints, physics and approximation-theory, this model can be adjusted to our image. McInerney [26] states several advantages for the use of deformable models:

1. They have the ability to segment, match and track images directly from image features (a bottom-up approach), and from *a priori* knowledge about location, size and form.
2. Deformable models are able to handle rather large variations that usually are present in human organs
3. The models can also handle greater complexity of the shape, than the low-level methods.

Thus deformable models are a good choice for ultrasound images, that sometimes can be very affected by noise. This noise is compensated for by the inherent continuity and smoothness of the models. This leads to greater robustness, accuracy and reproducibility in the image segmentation and can compensate for gaps, and other irregularities.

For the right ventricle this will be of special importance in the heavily trabeculated part of the ventricle (see chapter 2.4), which are a very noisy part of the ventricle in ultrasound images. The deformable model approach is also very useful for approximating the free wall of the ventricle which frequently can be missing from the ultrasound picture.

4.3 Active contour models

This sections presents a collection of different ways to make deformable models that can be used in model-based segmentation.

Snake models

In 1988 Kass et. al. presented a model called snake model [18], this model has been used extensively in the field of model-based segmentation.

This model is based on a parametric contour, which is a contour that is represented by parameters. An example is the circle which can be parametrized as:

$$x(s) = x_c + rcos(2\pi s) \tag{4.1}$$

$$y(s) = y_c + rsin(2\pi s) \tag{4.2}$$

for $s \in \left[0, 1\right]$, where x_c and y_c is the center and r is the radius, the parameter is in this equation s. The discretization of the parametric contour, can be performed by specifying a finite number of points on the contour specified by $\left[x(s), y(s)\right]$. Parametrization can also be performed by splines, which are explained further in chapter 5.1.

The basis of the segmentation by snake models is by using forces. The snake models has three kinds of forces; internal forces, image forces and external forces. The internal forces controls the stretching and bending of the snake. The images forces attract the snake to prominent features in the image, this can be done by using the intensity or gradients. The external forces are used for further control and constraints of the deformations of the snake.

In the traditional snake models energy computations were used instead of forces. The energy would be expressed as a sum of the internal energy and the image energy:

$$E(v) = E_{internal}(v) + E_{image}(v) \tag{4.3}$$

The goal is to minimize the snake energy, as this will stabilize the snake.

The internal energy is given by:

$$E_{internal} = \frac{1}{2}(\alpha(s)\left|\tfrac{\partial v}{\partial s}\right|^2 + \beta(s)\left|\tfrac{\partial^2 v}{\partial s^2}\right|^2) \tag{4.4}$$

where α is a constant that controls the elasticity in the snake, the constant β controls the rigidity in the snake and $v(s)$ is a point on the discrete snake.

The image energy can be given by many means, for example the intensity, the negative intensity, the negative gradient of the intensity and a blurred version of the intensity. A blurred version of the gradient of the intensity is usual, this gives the following expression for the image energy:

$$E_{image} = -\left|G_\sigma * \nabla^2 I\right|^2 \tag{4.5}$$

where G_σ is a Gaussian of standard deviation in σ.

The total energy of the snake can then be expressed as:

$$E(v) = \int_0^1 \frac{1}{2}(\alpha(s)\left|\tfrac{\partial v}{\partial s}\right|^2 + \beta(s)\left|\tfrac{\partial^2 v}{\partial s^2}\right|^2) + E_{image}(v)ds \tag{4.6}$$

The local minima of this function represents stable snake positions. The global minimum is not interesting as it just represent a point. By calculus of variations it can be found that the minima of the energy is given by:

$$-\alpha(s)\frac{\partial^2 v}{\partial s^2} + \beta(s)\frac{\partial^4 v}{\partial s^4} + \nabla E_{image}(v) = 0 \tag{4.7}$$

A solution for the minima is out of the scope of this text but can be found in [18].

Fully discrete snake models

Another version of an deformable model is the fully discrete snake, this is a model which not yields as exact results as the traditional snake, does not have the mathematical foundation, and are harder to implement. The advantage is however that it is easier understood and derived. Further description of the functionality of the fully discrete snake can be found in [22].

An illustration of the fully discrete snake model is shown in Figure 4.1. The goal of the internal forces of this snake model is to minimize the local curvature while still maintaining the snake length.

Statistical models

The shapes of the human internal organs vary a great deal in size and shape. This can make the task of segmenting the various structures challenging. If we have models of a variety of structures the desired organ can take, the task can be manageable. A desired way of doing this is to have a statistically based technique for modeling the different shapes.

A way of obtaining statistical data is to manually segment the desired feature in a number of images, and then representing them in a consistent way. This will be considered a training set, from which other models will be developed. It is important that the points in the different models of the

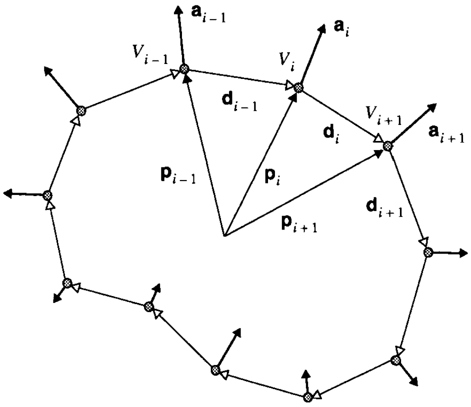


Figure 4.1: Illustration of the fully discrete snake. The model consists of a set of vertices V_i which are connected by edges d_i. Deformation is caused by the acceleration forces a_i acting on the vertices. Figure from [22]

training data set is consistently described, i.e. they have to be at approximately the same position in the model relative to the other points in the model.

A way of expressing these variations is with the use of principal component analysis (PCA) [8]. The concept is to reduce a data set with a high dimension to one of a lower dimension. The PCA is a least squares projection of the higher dimension dataset down on the eigenvectors of the sets covariance matrix [6]. The points in the model can be described as plane points as described in chapter 8.2.2, or by Fourier descriptors [6], [40].

The higher dimensional model is a statistical model obtained by examining N three-dimensional images. From a model with n nodes, the shapes are represented with an $3n$-dimensional vector. By this representation a mean model is also possible to make, this is described in chapter 8.2.2.

It is also possible to make a $3n \times 3n$ covariance matrix by:

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{4.8}$$

where $\bar{\mathbf{x}}$ is the mean shape from equation 8.5. The eigenvalues and eigenvectors of the covariance matrix, expresses the modes of variations. There will be $3n$ eigenvalues and eigenvectors, and the eigenvectors are orthogonal on each other. The eigenvalues λ and eigenvectors $\mathbf{p}_k$ of the covariance matrix is found by:

$$\mathbf{S}\mathbf{p}_k = \lambda_k \mathbf{p}_k,\ k = 1...3n \tag{4.9}$$

We sort the eigenvalues such that

$$\lambda_k \geq \lambda_{k+1} \tag{4.10}$$

Then we pick out the l eigenvalues that describes a sufficient part of the variations of the data set. The total variance of the data set can be expressed by:

$$\lambda_T = \sum_{k=1}^{3n} \lambda_k \tag{4.11}$$

In most cases it turns out that a large amount of the variation can be expressed with a few number of eigenvalues. The eigenvalue λ_k affects coordinate $\mathbf{p}_l$ by moving it parallel with $(\mathbf{p}_l - \bar{\mathbf{p}_l})$ of the first l eigenvectors.

The shape statistics can be expressed by:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{4.12}$$

where $P = \left[\mathbf{p}_1 \mathbf{p}_2 ... \mathbf{p}_l\right]$ is a matrix consisting of the l first eigenvectors sorted with respect to the eigenvalues λ_k. The vector $\mathbf{b} = \left[b_1 b_2 ... b_l\right]$ contains weights that is applied on each eigenvector.

Shape-space models

The theory in this section is taken from [3]. The shape-space models a curve with the notion of a shape-vector $\mathbf{X} \in \mathcal{S}$, where $\mathcal{S}$ is a shape-space. The shape-space is a linear parametrization of the set of allowed deformation of a curve, the linear requirement is stated for simplicity. This linear requirement is a limitation for complex object, such as three-dimensional objects with articulated parts. Models in the human body is often of such a kind and therefore there have been made non-linear extensions to the shape-space models [32]. This is implemented in the Real-Time Contour Tracking Library (chapter 9). It is possible to deal with articulated parts in the linear case too, but this has the cost that some geometrical constraints are relaxed.

The precise definition of shape-space is that a shape-space $\mathcal{S} = \mathcal{L}(W, \mathbf{Q_0})$ is a linear mapping of a "shape-space vector" $\mathbf{X} \in \mathbb{R}^N_X$ to a spline-vector $\mathbf{Q} \in \mathbb{R}^N_Q$:

$$\mathbf{Q} = W\mathbf{X} + \mathbf{Q}_0. \tag{4.13}$$

where W is a $N_Q \times N_X$ "shape-matrix". The vector $\mathbf{Q}$ is a vector of control points for a spline (see chapter 5.1). The vector $\mathbf{Q}_0$ is a template curve which one measures the shape variations against.

For an planar affine two-dimensional shape-space it is possible to represent a parametric spline curve $\mathbf{r}(s) = \big[x(s), y(s)\big]$ as:

$$\mathbf{r}(s) = \mathbf{u} + M\mathbf{r}_0(s) \tag{4.14}$$

where $\mathbf{u} = (u_1, u_2)^T$ is a two-dimensional translation vector, and M is a matrix which together with $\mathbf{u}$ represents 6 degrees of freedom in the space. M can be represented by a scaling matrix multiplied with a rotational matrix:

$$\mathbf{R} \cdot \mathbf{S} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} = \begin{bmatrix} s_x\cos\theta & -s_y\sin\theta \\ s_x\sin\theta & s_y\cos\theta \end{bmatrix} \tag{4.15}$$

For a shape-space representation as in equation 4.13. The shape-matrix W can be representated as:

$$W = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{Q_0^x} & \mathbf{0} & \mathbf{0} & \mathbf{Q_0^y} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{Q_0^y} & \mathbf{Q_0^x} & \mathbf{0} \end{bmatrix} \tag{4.16}$$

Where Q_0^x and Q_0^y are the x- and y-coordinates of the control points of the splines. The shape-space vector $\mathbf{X}$ is:

$$\mathbf{X} = \begin{bmatrix} u_1 & u_2 & M_{11} - 1 & M_{22} - 1 & M_{21} & M_{12} \end{bmatrix}^T \tag{4.17}$$

where u_1 and u_2 represents translation in the horizontal and vertical direction respectively. For example does $\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}^T$ represent a doubling of the template size and $\mathbf{X} = \begin{bmatrix} 0 & 0 & \cos\theta - 1 & \cos\theta - 1 & -\sin\theta & \sin\theta \end{bmatrix}^T$ represent a rotation of the template through the angle $\cos\theta$ The framework presented in chapter 9 accounts for non-linear deformations as well.

Chapter 5

Smooth parametric surfaces

Human organs can rarely be described by straight lines, but rather by smooth surfaces. The right ventricle is no exception to this. The process of making models of smooth surfaces can be a very time-consuming task, requiring much of the modeler. A method for making smooth surfaces out of connected straight curves is desired. Important properties for such a method are listed in [52]:

- **Efficiency**. The method should be efficient; the refined points should be obtained by a small amount of calculations. Splines are more efficient than subdivision.
- **Local definition**. The rules used to determine where new points go should not depend on ”far away” places
- **Affine invariance**. An affine transformation (i.e. scaling, translation or rotation) on the original set of points should yield the same transformation in the resulting shape.
- **Simplicity**. The rules should be determined off line and the number of rules should be small
- **Continuity**. There should be available mathematical proofs of convergence and continuity.
- **Arbitrary topology**. The methods should be able to represent any kind of geometrical shape. This is an important feature in the modeling of the heart chambers for contour tracking.

All of these points is also valid for spline surfaces, in fact spline surfaces can sometimes yield better performance of the properties

5.1 Splines

A spline is a piecewise polynomial function, which consists of concatened polynomial segments, each of polynomial order d [3]. The polynomial segments are joined together at breakpoints. Normal orders of splines are quadratic ($d = 3$) and cubic ($d = 4$). Where the order of a polynomial is the number of its coefficients. If the polynomial order is fixed and low, the spline has computational stability and simplicity. This is true even for complex geometric shapes.

B-splines

One common type of splines is the B-splines, other types is Bézier-splines, Hermite splines, cardinal splines and several others [14]. To make up a parametric curve of B-splines we use the vector

$$\mathbf{r}(s) = \begin{bmatrix} x(s) & y(s) \end{bmatrix} \tag{5.1}$$

This curve is constructed by the use of basis functions $B_n(s)$, $n = 0, ..., N_B - 1$. $\mathbf{r}(s)$ is then constructed as a weighted sum of the basis-functions. Each of the basis functions are defined over a span of the s-axis. The basis-functions are constructed by recursion, defined for a regular spline by:

$$B_{n,1}(s) = \begin{cases} 1 \text{ if } n \leq s < n+1 \\ 0 \text{ otherwise} \end{cases} \tag{5.2}$$

$$B_{n,d}(s) = \frac{(s-n)B_{n,d-1}(s) + (n+d-s)B_{n+1,d-1}(s)}{d-1} \tag{5.3}$$

For each basis function a control point is defined as $\mathbf{q}_n = \begin{bmatrix} q_n^x & q_n^y \end{bmatrix}^T$, and the curve is expressed as a weighted sum of the control points:

$$\mathbf{r}(s) = \sum_{n=0}^{N_B - 1} B_n(s)\mathbf{q}_n \text{ for } 0 \leq s \leq L \tag{5.4}$$

where L is the number of spans and N_B is the number of basis-functions. An illustration of a regular, periodic, quadratic spline is illustrated in Figure 5.1.

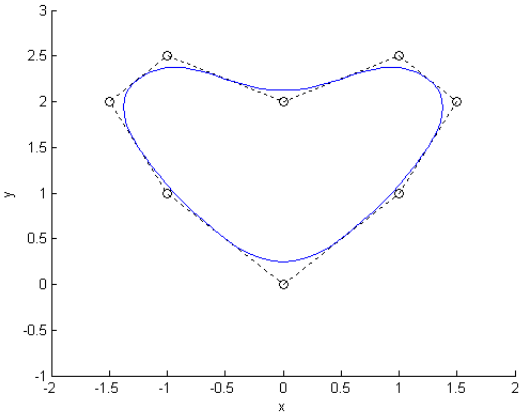


Figure 5.1: A spline curve with its control points, illustrating that the curve is a smooth approximation of its *control polygon* (dotted line). The spline has $L = 8$ spans and $N_B = 8$ basis functions. The spline is approximating because the curve does not have to go through the control points.

Spline surfaces

Spline surfaces are splines described with two sets of orthogonal spline curves [14]. Thus to define a spline surface it is necessary to specify the two sets by a mesh of control points over some spatial region. The spline surfaces can be expressed by $(N_{B_u}) \times (N_{B_v})$ control points $\mathbf{q}_{n_u,n_v}$ arranged in a mesh. The spline surface can then be calculated by:

$$\mathbf{R}(u,v) = \sum_{n_u=0}^{N_{B_u}-1} \sum_{n_v=0}^{N_{B_v}-1} \mathbf{q}_{n_u,n_v} B_{n_u}(u) B_{n_v}(v) \tag{5.5}$$

The parameters u and v can vary depending on what type of splines are used, but a common choice is that they can vary between 0 and 1. The spline surfaces exhibits the many of the same properties as the component spline curves, but they do also contain topological limitations.

5.2 Subdivision

One method for smoothing surfaces is by subdivision, which is a generalisation of spline surfaces to arbitrary topology. The idea behind subdivision is to calculate more and more points in between the initial points, and this will eventually converge to a smooth surface or curve. There are proposed several methods for deciding where the new points should be positioned.

Peter Schröder and Denis Zorin describes subdivision as [52]:

Subdivision defines a smooth curve or surface as the limit of a sequence of successive refinements

Thus, the idea is to describe the smooth surfaces or curves as connected straight line segments which are iteratively refined.

Figure 5.2 shows an example of subdivision in 2D where three lines represented by four vertices are refined into a smooth segment. We can see that the refinement improves the smoothness of the curve rather quick and that after only a few refinements a rather smooth segment is obtained. Figure 5.3 shows subdivision in 3D where an initial mesh defines a contour and subdivision on the surfaces makes a smooth picture.

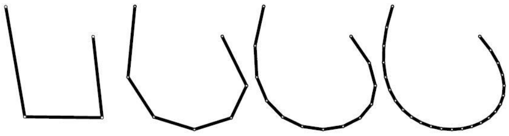


Figure 5.2: Subdivision in 2D. Initial control points are shown to the left, and the successive result of subdivision are shown from left to right. Figure adapted from [52].

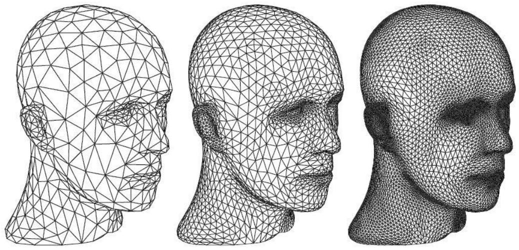


Figure 5.3: Subdivision in 3D. Initial control points are shown to the left, and the successive result of subdivision are shown from left to right. Figure adapted from [52].

The basic building block of a subdivision procedure is the *univariate subdivision step* [36]. The lines are initially represented by points called *control points*, and these points joined together defines a *control polygon*.

The control points are expressed as a set:

$$P^n = \{P_i^n\}_i \in \mathbb{R}^s \tag{5.6}$$

To generate a refined set of control points constituting a refined control polygon we use the *subdivision step* S with the use of a *subdivision mask* with finitely many non-zero entries.

$$P_i^{n+1} = SP^n = \{P_i^{n+1}\} \tag{5.7}$$

$$P_i^{n+1} = \sum_k a_{i-2k} P_k^n \tag{5.8}$$

where the superscript denotes the level of refinement and the subscript denotes the number of the control point. Equation 5.8 is called the *subdivision rule.* The subdivision rule splits into two formulas, one for even entries and one for odd entries.

$$P_{2i}^{n+1} = \sum_{k \in \mathbb{Z}} a_{2i-2k} P_k^n = \sum_{k \in \mathbb{Z}} a_{2k} P_{i-k}^n \tag{5.9}$$

$$P_{2i+1}^{n+1} = \sum_{k \in \mathbb{Z}} a_{2i+1-2k} P_k^n = \sum_{k \in \mathbb{Z}} a_{2k+1} P_{i-k}^n \tag{5.10}$$

where the set $\mathbb{Z}$ is the set of all integer numbers.

As an example; for cubic splines the coefficients are

$$a_0 = \frac{1}{8}, a_1 = \frac{1}{2}, a_2 = \frac{6}{8}, a_3 = \frac{1}{2}\text{and } a_4 = \frac{1}{8} \tag{5.11}$$

giving the rules:

$$P_{2i}^{n+1} = \frac{1}{8} P_i^n + \frac{6}{8} P_{i-1}^n + \frac{1}{8} P_{i-2}^n \tag{5.12}$$

$$P_{2i+1}^{n+1} = \frac{1}{2} P_i^n + \frac{1}{2} P_{i-1}^n \tag{5.13}$$

There are two methods of subdivision: *interpolating* and *approximating.* Interpolating methods keeps the control points, while approximating methods allows the control points to be positioned on the outside of the resulting curve or surface. The subdivision rule shown in Figure 5.4 is approximating, where even points at refinement level $n + 1$ are local averages of points at level n.

In subdivision surfaces control points are defined as in the case of the curves. In addition the control points are arranged in a mesh, where it is specified which of the other control points in the set who is connected to the other control points, see Figure 5.2 and Figure 5.3. The refinement process on these meshes results in finer meshes. This process is called *vertex insertion* because each edge are replaced by two edges by the insertion of a new vertex.

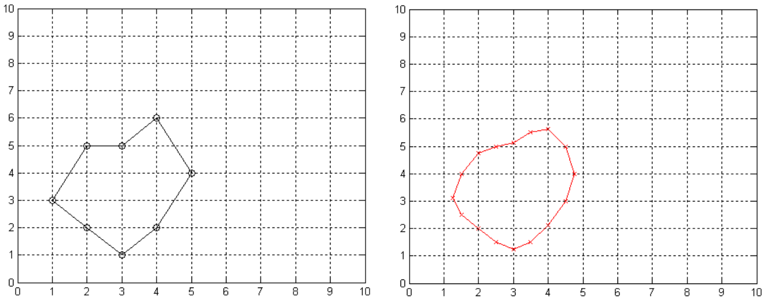


Figure 5.4: Illustration of one refinement of the cubic spline subdivision rule. The original control points are shown to the left, and the refined curve is shown to the right. The subdivision rule is approximating, in which the curve does not have to go through the control points.

5.3 Subdivision schemes

There are several different methods for subdivision of surfaces. They build upon different techniques to obtain the desired result. We can classify the different by several criteria as listed in [52]:

1. The type of refinement rule
2. The type of generated mesh
3. Approximating or interpolating
4. Degree of smoothness of the limit surfaces

A *subdivision scheme* S is made up of iterations of subdivision steps which produces sequences of control point sets and makes sequences of piecewise linear control polygons. A subdivision scheme S produces a sequence of control point sets in the following manner:

$$(S^n P^0)_i = P_i^n = (S^n S^{n-1} ... S^2 S^1 P^0)_i \tag{5.14}$$

where P^0 is the initial sets of control points. All the different subdivision schemes is based upon one basic rule, the *regular subdivision scheme*, i.e. the rule that is performed on the regular topology. For the schemes to be able to describe arbitrary topologies, additional rules are required. These rules describe how subdivision should be performed on edges, creases and other irregular topologies. One advantage of subdivision is that it only uses local control points information for all the computations.

Once the subdivision is performed we need to specify how the original vertex should be connected to the new vertex. There are two ways of doing this; *face split* and *vertex split.* Face split, shown in Figure 5.5, divide each face into four new faces; this is the case for both quadrilateral meshes and triangular meshes. Each of the new faces is defined by one old and three new vertices for the quadrilateral case and one old and two new vertices for the triangular case. These vertices are placed on the edges, and for the quadrilateral case one vertex is placed in the old face.

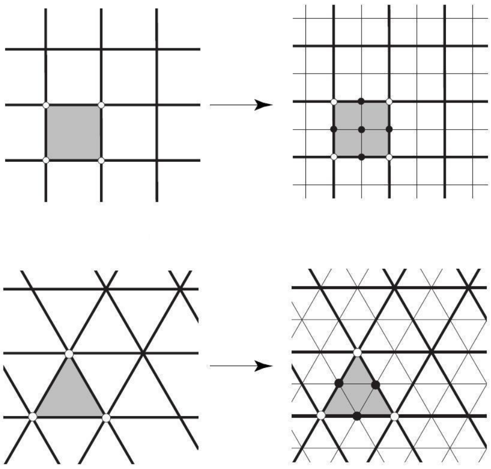


Figure 5.5: Illustration of the face split approach to subdivision. Each face is divided into four new faces. The top picture show face split from quadrilateral faces, and the bottom shows face split for triangular faces. Figure adapted from [52].

When the vertex split rule is used, each node is split into new nodes, one for each adjacent face. The old vertex is deleted and the new vertexes make up a new face. Figure 5.6 visualize the idea of this subdivision rule. When a quadrilateral mesh is used, the new vertices will get valence four, i.e. four adjacent nodes. With triangular meshes there will be made hexagons. Vertex split in triangular meshes is therefore not frequently used.

As stated above, subdivision schemes could be interpolating or approximating. From Figure 5.6 we can see that the vertex split rule is approximating. None of the original vertices are kept in the refining process. The face split rule can however be defined as interpolating, if the original set of vertices are defined as control points, we can define the same set of vertices as control points for the refined surface. Interpolating schemes have an advantage in

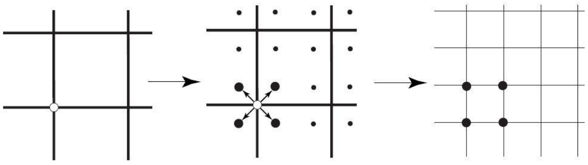


Figure 5.6: Illustration of the vertex split approach to subdivision. Each vertex is split into four new vertices. Figure adapted from [52].

that the user can specify an initial mesh with control points that will be kept during the whole refinement process, and hence will have more control over the shape of the surface. However, interpolating rules does not converge as fast as approximating rules and the surfaces will not be as accurate.

This and the previous section treated the subject of subdivision and presented a way of classifying the different subdivision schemes. Several schemes are proposed and they all possess different properties and abilities. Examples of the different schemes are: the Loop scheme, the modified butterfly scheme, the Catmull-Clark scheme, the Kobbelt scheme, the Doo-Sabin scheme, and the Midedge scheme. This theory summary will treat two of them; Catmull-Clark and Doo-Sabin.

5.3.1 Catmull-Clark

Regular case of order 4

The Catmull-Clark scheme [52] is an expansion of bi-cubic B-spline surfaces, see equations 5.12 and 5.13. It is an approximating, face-split scheme with quadrilateral faces. For a rectangular mesh, the control points are arranged and named in the following manner. v^i is a vertex of the grid produced after the i-th subdivision step, e^i_k for $k = 1, ..., 4$ is the vertex' four neighbors in the grid, these points are called *edge points*, the four points that constitutes the missing vertices to form the full grid is called *face points* and are denoted as f^i_l for $l = 1, ..., 4$. An illustration of this are shown in Figure 5.7. The new vertex points are generated by applying the masks shown in Figure 5.8. This gives the equation for the new vertex point:

$$v^{i+1} = \frac{1}{64}(v^i + 6e^i_1 + 6e^i_2 + 6e^i_3 + 6e^i_4 + f^i_1 + f^i_2 + f^i_3 + f^i_4) \tag{5.15}$$

The new edge points are calculated by applying the masks shown in Figure 5.9. The left masks is applied on the edge points on the vertical edges and

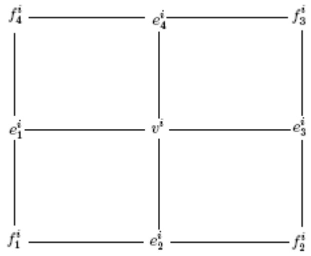


Figure 5.7: Definition of grid points.

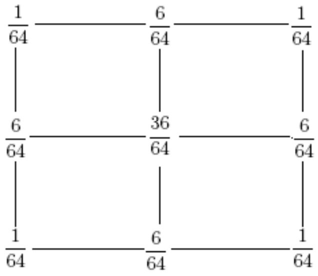


Figure 5.8: Bicubic mask for vertex points in the Catmull-Clark scheme for topology of order 4.

the right masks on the edge points on the horisontal edges. This gives the

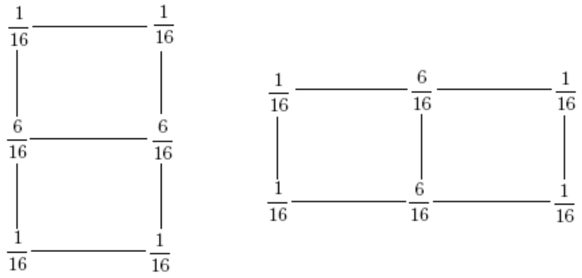


Figure 5.9: Bicubic mask for edge points in the Catmull-Clark scheme for topology of order 4.

equation for the new edge points:

$$e_1^{i+1} = \frac{1}{16}(6v^i + 6e_1^i + e_2^i + e_4^i + f_1^i + f_4^i) \tag{5.16}$$

$$e_2^{i+1} = \frac{1}{16}(6v^i + 6e_2^i + e_1^i + e_3^i + f_1^i + f_2^i) \tag{5.17}$$

$$e_3^{i+1} = \frac{1}{16}(6v^i + 6e_3^i + e_2^i + e_4^i + f_2^i + f_3^i) \tag{5.18}$$

$$e_4^{i+1} = \frac{1}{16}(6v^i + 6e_4^i + e_1^i + e_3^i + f_3^i + f_4^i) \tag{5.19}$$

Finally the new face points are computed by the mask in Figure 5.10. This gives the equation for the new face points:

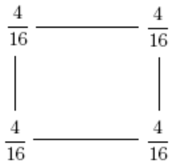


Figure 5.10: Bicubic mask for face points in the Catmull-Clark scheme for topology of order 4.

$$f_1^{i+1} = \frac{1}{16}(4v^i + 4e_1^i + 4e_2^i + 4f_1^i) \tag{5.20}$$

$$f_1^{i+1} = \frac{1}{16}(4v^i + 4e_2^i + 4e_3^i + 4f_2^i) \tag{5.21}$$

$$f_1^{i+1} = \frac{1}{16}(4v^i + 4e_3^i + 4e_4^i + 4f_3^i) \tag{5.22}$$

$$f_1^{i+1} = \frac{1}{16}(4v^i + 4e_4^i + 4e_1^i + 4f_4^i) \tag{5.23}$$

All the above equations is applied on every control point, and will generate a new refined surface. It is clear that this scheme only needs local control point information. By using vectors it is possible to express this in a concise form:

$$V^{i+1} = S_4 V^i \tag{5.24}$$

with the *local subdivision matrix* given by

$$S_4 = \frac{1}{16}\begin{bmatrix} 9 & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 6 & 1 & 6 & 1 & 0 & 1 & 1 & 0 & 0 \\ 6 & 0 & 1 & 6 & 1 & 0 & 1 & 1 & 0 \\ 6 & 1 & 0 & 1 & 6 & 0 & 0 & 1 & 1 \\ 4 & 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & 4 & 4 & 0 & 0 & 4 & 0 & 0 \\ 4 & 0 & 0 & 4 & 4 & 0 & 0 & 4 & 0 \\ 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \end{bmatrix} \tag{5.25}$$

and the points are collected in a vector

$$V^i = \begin{bmatrix} v^i & e_1^i & e_2^i & e_3^i & e_4^i & f_1^i & f_2^i & f_3^i & f_4^i \end{bmatrix}^T \tag{5.26}$$

When the local grid is generated for all the vertices v^i each new face point is connected to all those new edge points which correspond to edges bounding this face. This creates the new global control grid.

General case of order N

The vertex mask (Figure 5.8), can be generalized by noting that the new vertex point is a convex combination of the old vertex point, the average of all new face points, and the average of all end points of the edges meeting in the old vertex. This gives the generalization:

$$v^{i+1} = \frac{1}{N}((N-2)v^i + \frac{1}{N}\sum_j e_j^i + \frac{1}{N}\sum_j f_j^{i+1}) \tag{5.27}$$

where N is the order, i.e. the number of vertices attached to the vertex. The points f_j^{i+1} is calculated by the same rule as for the regular case. This is because the rule just calculates the centroid of the face, i.e. the average of all its corners. The mask for the edge points can be generalized by noting that it computes the average of the two centroids of the two incident four-sided faces, and the endpoints of the edge. This gives:

$$e_j^{i+1} = \frac{1}{4}(v^i + e_j^i + f_{j-1}^{i+1} + f_j^{i+1}) \tag{5.28}$$

where the f_j^{i+1} are given as explained for equation 5.27. When all the vertex, edge and face points are calculated the edges of the new grid are placed by connecting each new face point with the new edge points of the edges defining the old face and connecting each new vertex point to the new edge points for all old edges sharing the old vertex point.

The Catmull-Clark scheme produces a C^2-continuous surface, except in extraordinary vertices where the surface is C^1-continuous.

5.3.2 Doo-Sabin

The Doo-Sabin scheme is an expansion of bi-quadric B-splines to arbitrary topology. It is a quadrilateral, vertex splitting scheme, the rule will result in a C^1-continuous surface.

Regular case of order 4

The mask used for the Doo-Sabin subdivision scheme for quadrilateral faces is shown in figure 5.11. If v_k^i for $k = 1, ...4$ defines the vertices for a quadrilateral face after the i-th subdivision; the Doo-Sabin yields the following rule for calculating the next refinement level, written on matrix form:

$$\begin{bmatrix} v_1^{i+1} \\ v_2^{i+1} \\ v_3^{i+1} \\ v_4^{i+1} \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 9 & 3 & 1 & 3 \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ 3 & 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} v_1^i \\ v_2^i \\ v_3^i \\ v_4^i \end{bmatrix} \tag{5.29}$$

The generated face is connected to the other new generated faces across the old edges and the old vertices. This generates additional faces in the new mesh.

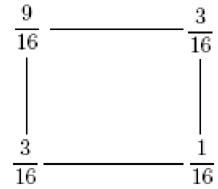


Figure 5.11: The vertex mask for Doo-Sabin subdivision for quadrilateral faces.

General case of order N

The generalization for faces consisting of N vertices are in the Doo-Sabin scheme given by:

$$\begin{bmatrix} v_1^{i+1} \\ v_2^{i+1} \\ \vdots \\ v_N^{i+1} \end{bmatrix} = \begin{bmatrix} \frac{N+5}{4N} & \frac{3+\cos\frac{2\pi}{N}}{4N} & \cdots & \frac{3+\cos\frac{2\pi(N-1)}{N}}{4N} \\ \frac{3+\cos\frac{2\pi}{N}}{4N} & \frac{N+5}{4N} & \cdots & \frac{3+\cos\frac{2\pi(N-2)}{N}}{4N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{3+\cos\frac{2\pi(N-1)}{N}}{4N} & \frac{3+\cos\frac{2\pi(N-2)}{N}}{4N} & \cdots & \frac{N+5}{4N} \end{bmatrix} \begin{bmatrix} v_1^i \\ v_2^i \\ \vdots \\ v_N^i \end{bmatrix} \tag{5.30}$$

To construct new control grids from the refined faces, first each N-vertex face is connected together. Then with reference to Figure 5.12 we can observe that each new vertex corresponds to an old vertex and its intersecting old edges. From each new vertex there are two other vertices that comes from the same old vertex and that is associated with each one of the two old edges. These two new vertices are connected across the old edges. Doo-Sabin subdivision on a sphere is illustrated in Figure 5.13.

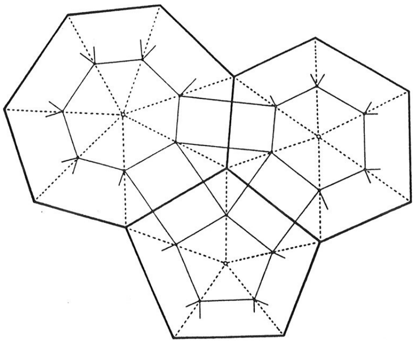


Figure 5.12: Illustration of the construction of new control points in the Doo Sabin subdivision scheme. Figure is taken from [36].

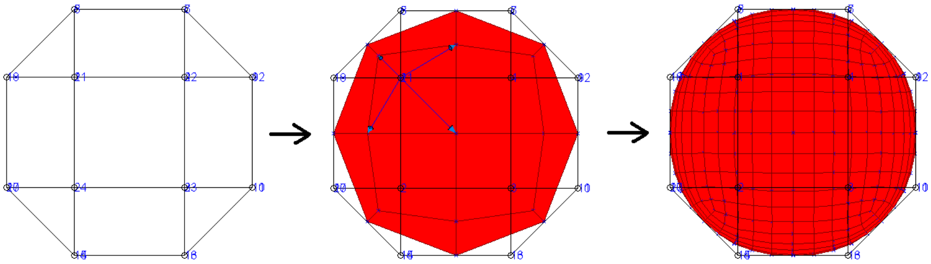


Figure 5.13: Doo-Sabin subdivision on a sphere. Each node in the initial mesh, shown to the left, is divided in to four new nodes and making new surfaces. The final sphere is shown to the right.

Direct evaluation for arbitrary topology

Stam [41] has introduced a non-iterative method for efficient evaluation of the Catmull-Clark subdivision surfaces, this method has been modified in [34] to work for Doo-Sabin surfaces. This method uses Doo-Sabin subdivision as a matrix division.

The control points are expressed in a matrix $\mathbf{Q}_0$ of the size $N_Q \times 3$, this matrix can be multiplied with a $N_Q + 7 \times N_Q$ subdivision matrix $\mathbf{S}$, as explained in the previous chapter. Both for extraordinary vertices and regular vertices can this multiplication be used as it creates at most one irregular face regardless of the topology of the mesh, the other three faces would always be regular.

It is possible to pick out a subset of the control vertices for each subpatch $k \in \{0, \cdots, 3\}$ by:

$$\mathbf{Q}_{n+1,k} = \mathbf{P}_k \mathbf{S} \mathbf{Q}_n \tag{5.31}$$

where $\mathbf{P}_k$ is a $N_Q \times N_Q + 7$ "picking matrix" for selecting N_Q control points and storing them in $\mathbf{Q}_{n+1,k}$.

Subdivision can then be performed until a desired point is on the outside of an extraordinary patch, and this can be expressed by:

$$\mathbf{Q}_{n,k} = \mathbf{P}_k \mathbf{S} \mathbf{S}_0^{n-1} \mathbf{Q}_0 \tag{5.32}$$

There are n number of subdivision steps which are required to be able to get the desired point out of the extraordinary patch. This number n is determined by:

$$n = \lfloor -\log_2(\max\{u, v\}) \rfloor \tag{5.33}$$

where u and v is the parametric position of the point within a patch. The rules for witch new sub-patch k to pick after each subdivision is given by:

$$k = \begin{cases} 1 \text{ if } u > \frac{1}{2^{n+1}} \text{ and } v < \frac{1}{2^{n+1}} \\ 2 \text{ if } u > \frac{1}{2^{n+1}} \text{ and } v > \frac{1}{2^{n+1}} \\ 3 \text{ if } u < \frac{1}{2^{n+1}} \text{ and } v < \frac{1}{2^{n+1}} \end{cases} \tag{5.34}$$

It is now possible to calculate the basis functions to the original control points by:

$$\mathbf{b}(u, v)|_{\Omega_k^n} = (\mathbf{P}_k \mathbf{S} \mathbf{S}_0^{n-1})^T \tilde{\mathbf{b}}(\mathbf{t}_{k,n}(u, v)) \tag{5.35}$$

where $\tilde{\mathbf{b}}$ are regular bi-quadric B-spline basis functions, $\mathbf{S}$ is the subdivision matrix, $\mathbf{P}_k$ is the "picking matrix" and Ω_k^n is a subdivision mapping function which determines the number of subdivision steps required. The transformation $\mathbf{t}_{k,n}$ maps the interval (u, v) to the interval within a desired subpatch according to a rule presented in [34].

The evaluation of the surface position is then given by:

$$\mathbf{p}(u, v) = \mathbf{Q}_0^T \mathbf{b}(u, v) \tag{5.36}$$

Chapter 6

State-space modeling and estimation

6.1 State-space model

A discrete state-space model can be expressed in the following way [9]:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{6.1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \tag{6.2}$$

The model is a discretisation of a set of first-order differential equations. These equations describes the dynamics of a state vector $\mathbf{x} = (x_1, ..., x_n)^T$. And incorporates the effect of a input $\mathbf{u} = (u_1, ..., u_p)$. The z is the measurement vector. The vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are process noise and measurements noise respectively.

The functions $\mathbf{f}$ and $\mathbf{h}$ can be non-linear. It is however quite usual to linearize the function thus obtaining the important system class, the *linear time-invariant system.*

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \tag{6.3}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{v}_k \tag{6.4}$$

A block diagram of the system is illustrated in figure 6.1.

For the purpose of image segmentation it is possible to represent a deformable model using a state vector. This is proposed in [3] and incorporated in the Real-Time Contour Tracking Library described in chapter 9. This approach uses a concatenation of local deformations and global transform as a state vector, this is explained further in chapter 9.

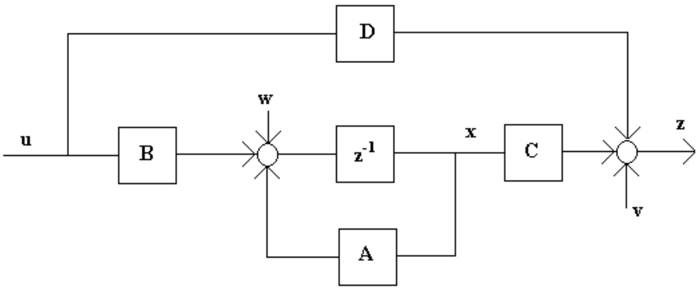


Figure 6.1: Block diagram of the linear time-invariant system. The z^{-1} is a unit time delay represented by the z-transform.

6.2 Kalman filter

In the previous chapter it was assumed that all the state variables where available for state feedback. This is an assumption that may not hold in practice. This can stem from many reasons such as the states are not directly accessible, or measurements devices are difficult to obtain. These states are still needed for system control, therefore a need for state estimation occurs.

Measurement and process noise often corrupts the data which are available from the system. This is a problem that a state estimator needs to account for. System measurements and the mathematical model are the available data from the system, and these data can be used for estimating the missing states.

There are several methods for state estimation, such as minimum variance and least-squared error methods [21]. One estimation method is the celebrated Kalman filter, which can be viewed upon as a covariance matrix to capture state uncertainty. The Kalman filter is a recursive algorithm based on minimizing the mean of the squared estimation error.

In the case of image segmentation it is used to couple edge measurements with a kinematic model, allowing for robust real-time segmentation (chapter 9).

The following theory is adapted from [4].

A random process is assumed to have the general model:

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{w}_k \tag{6.5}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{6.6}$$

where $\mathbf{x}_k = \mathbf{x}(t_k)$, ϕ_k relates the previous sample to the next sample, $\mathbf{w}_k$ are white noise, and $\mathbf{v}_k$ is measurement error assumed to be white noise with known covariance:

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & \text{if } i = k \\ 0, & \text{if } i \neq k \end{cases} \tag{6.7}$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k, & \text{if } i = k \\ 0, & \text{if } i \neq k \end{cases} \tag{6.8}$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0, \text{for all} k, i \tag{6.9}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices for $\mathbf{w}$ and $\mathbf{v}$ respectively. Equation 6.9 shows that the noise sequences are uncorrelated. If we have a *a priori* estimate $\hat{\mathbf{x}}_k^-$, i.e. an estimate based on our knowledge of the process prior to t_k, and that the covariance matrix of $\hat{\mathbf{x}}_k^-$ is known. Then we can define the estimation error:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \tag{6.10}$$

with the covariance matrix:

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \tag{6.11}$$

If the measurements are used it is possible to set up an expression for the *a posteriori* (updated) estimate.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \hat{\mathbf{x}}_k^-) \tag{6.12}$$

The $\mathbf{K}_k$ is a blending factor, and the goal for the Kalman-filter is to find the $\mathbf{K}_k$ which minimizes the mean-square of the estimation error in equation 6.10.

Analog to the covariance matrix of the a priori estimate 6.11 it is possible to express the covariance matrix of the updated error.

$$\begin{aligned}
\mathbf{P}_k &= E[\mathbf{e}_k\mathbf{e}_k^T] && (6.13)\\
&= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] && (6.14)\\
&= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \hat{\mathbf{x}}_k^-))(\mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \hat{\mathbf{x}}_k^-)^T] && (6.15)\\
&= E[((\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k\mathbf{v}_k)((\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k\mathbf{v}_k)^T] && (6.16)\\
&= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T](\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T && (6.17)\\
&- (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)\mathbf{v}_k^T]\mathbf{K}_k^T - \mathbf{K}_kE[\mathbf{v}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T](\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T && (6.18)\\
&+ \mathbf{K}_kE[\mathbf{v}_k\mathbf{v}_k^T]\mathbf{K}_k^T && (6.19)\\
&= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R_k}\mathbf{K}_k^T && (6.20)
\end{aligned}$$

The last equality comes from the fact that the a priori estimation error is uncorrelated to the measurement noise. To obtain the $\mathbf{K}_k$ that minimizes the mean-square estimation error, we differentiate $\mathbf{P}_k$ with respect to $\mathbf{K}_k$ and set the derivate to zero, we obtain:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{6.21}$$

By inserting the Kalman gain in equation 6.21 into equation 6.12, the following expression for the updated covariance is obtained:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{6.22}$$

The next a priori estimate can now be obtained via the Kalman filter gain:

$$\hat{\mathbf{x}}_{k+1}^- = \phi_k\hat{\mathbf{x}}_k^- \tag{6.23}$$

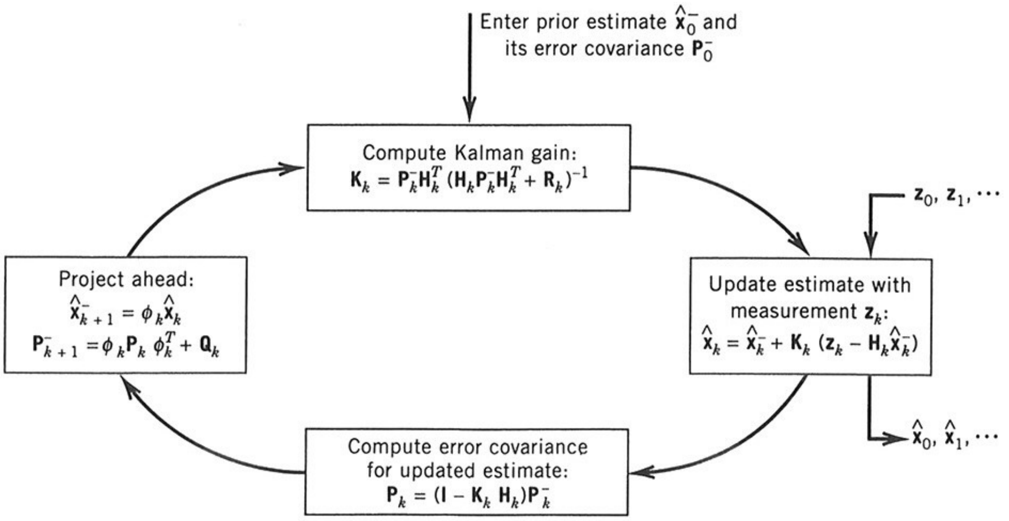
the a priori estimation error then becomes:

$$\begin{aligned}
\mathbf{e}_{k+1}^- &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- && (6.24)\\
&= (\phi_k\mathbf{x}_k + \mathbf{w}_k) - \phi_k\hat{\mathbf{x}}_\mathbf{k} && (6.25)\\
&= \phi_k\mathbf{e}_k + \mathbf{w}_k && (6.26)
\end{aligned}$$

and the covariance matrix:

$$\begin{aligned}
\mathbf{P}_{k+1}^{-} &= E[\mathbf{e}_{k+1}^{-}\mathbf{e}_{k+1}^{-T}] && (6.27)\\
&= E[(\phi_k\mathbf{e}_k + \mathbf{w}_k)(\phi_k\mathbf{e}_k + \mathbf{w}_k)^T] && (6.28)\\
&= \phi_{\mathbf{k}}E[\mathbf{e}_k\mathbf{e}_k^T]\phi_k^T + \phi_k E[\mathbf{e}_k\mathbf{v}_k^T] + E[\mathbf{v}_k\mathbf{e}_k^T]\phi_k + E[\mathbf{v}_k\mathbf{v}_k^T] && (6.29)\\
&= \phi_k\mathbf{P}_k\phi_k^T + \mathbf{Q}_k && (6.30)
\end{aligned}$$

The equations 6.12, 6.21, 6.22, 6.23, and 6.30 gives the Kalman Filter. The sequence of the Kalman Filter is illustrated in Figure 6.2.



Enter prior estimate $\hat{x}_0^-$ and its error covariance P_0^-
Compute Kalman gain: $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
$z_0, z_1, \cdots$
Update estimate with measurement z_k: $\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-)$
$\hat{x}_0, \hat{x}_1, \cdots$
Compute error covariance for updated estimate: $P_k = (I - K_k H_k) P_k^-$
Project ahead: $\hat{x}_{k+1}^- = \phi_k \hat{x}_k$ $P_{k+1}^- = \phi_k P_k \phi_k^T + Q_k$

Figure 6.2: The Kalman Filter Loop. An initial estimate with error covariance is entered into the algorithm at the beginning, and measurements are inserted for each iteration. The output of the algorithm are the estimated states of the system, based on minimization of the mean of the squared estimation error. Figure from [4].

6.2.1 Extended Kalman Filter

The Kalman filter described in the previous chapter assumes a linear model. In many cases a linear model is not apparent, and we are facing a non-linear model. The extended Kalman filter accounts for this non-linearity [15]. The state model can be expressed as:

$$\mathbf{x}_{k+1} = f(x_k, u_k) + \mathbf{w}_k \qquad (6.31)$$

$$\mathbf{z}_k = h(x_k) + \mathbf{v}_k \tag{6.32}$$

Linearization by application of a first-order Taylor expansion, gives a linearized measurement function around the a-priori state estimates:

$$z_k = h_k(\hat{\mathbf{x}}_k^-) + \mathbf{v}_k + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \tag{6.33}$$

$$\mathbf{H}_k = \frac{\partial h_k}{\partial x_k^T}(\hat{\mathbf{x}}_k^-) \tag{6.34}$$

A priori estimation of the measurements can be obtained by the expected value of the measurement based on all the previous measurements Z_{k-1}

$$\hat{\mathbf{z}}_k^- = E[\mathbf{z}_k | Z_{k-1}] = h_k(\hat{\mathbf{x}}_k^-) \tag{6.35}$$

The covariance matrix of the error between the measurement and the a priori estimated measurements can be expressed as:

$$\begin{aligned}
\hat{\mathbf{P}}_{k|k-1} &= E[(\mathbf{z}_k - \hat{\mathbf{z}}_k^-)(\mathbf{z}_k - \hat{\mathbf{z}}_k^-)^T] && (6.36)\\
&= E[(\mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{v}_k)(\mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{v}_k)^T] && (6.37)\\
&= \mathbf{H}_k\hat{\mathbf{P}}_k^-\mathbf{H}_k^T + \mathbf{R}_k && (6.38)
\end{aligned}$$

Now it is possible to set up an expression for the a posteriori state estimation, based upon the a priori estimates, the measurements, the Kalman gain and the mathematical model.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k(\hat{\mathbf{x}}_{\mathbf{k}}^-)] \tag{6.39}$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{P}}_k^- \tag{6.40}$$

$$\mathbf{K}_k = \hat{\mathbf{P}}_k^-\mathbf{H}_k^T(\mathbf{H}_k\hat{\mathbf{P}}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} = \hat{\mathbf{P}}\mathbf{H}_k^T\mathbf{R}_k^{-1} \tag{6.41}$$

The state space model can now be expressed as:

$$\mathbf{x}_{k+1} = f_k(\hat{\mathbf{x}}_k, \mathbf{u}_k) + \mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \mathbf{w}_k \tag{6.42}$$

$$\mathbf{A}_k = \frac{\partial f_k}{\partial x_k^T}(\hat{\mathbf{x}}_{\mathbf{k}}, \mathbf{u}_k) \tag{6.43}$$

With the a priori estimated states given by:

$$\hat{\mathbf{x}}_{k+1}^- = f(\hat{\mathbf{x}}_k, \mathbf{u}_k) \tag{6.44}$$

Finally the state estimation error and its covariance matrix:

$$\hat{\mathbf{e}}_{k+1}^- = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{w}_k \tag{6.45}$$

$$E[\hat{\mathbf{e}}_{k+1}^-\hat{\mathbf{e}}_{k+1}^{-T}] = \mathbf{A}_k\hat{\mathbf{P}}_k^-\mathbf{A}_k^T + \mathbf{Q}_k \tag{6.46}$$

Part II

Methods and Tools

Chapter 7

Manual segmentation

7.1 Extracting slices from the volume

In order to understand and visualize the geometry of the right ventricle, there is a need to extract two-dimensional planes from the three-dimensional data in order to show them on two-dimensional computer screens. The data acquired from three-dimensional ultrasound would be a pyramidal-like volume (see chapter 3.1.3). It is necessary to extract one arbitrary two-dimensional plane from the volume, this is illustrated in Figure 7.1. The approach taken is called *any plane display* [1], and allows two-dimensional displays of freely selected planes through the data set.

A two-dimensional plane is fully defined by its four corner points [42]. These corner points are placed in a counter-clockwise manner as shown in Figure 7.3. Figure 7.2 shows the section shown in Figure 7.1 taken from the recording in Figure 3.4. Arbitrary two-dimensional cross sections can now be displayed on the screen, even cross-sections that can not be obtained from ordinary two-dimensional scans. This allows fast examination of the object under examination,

The approach taken is called bi-linear interpolation, and the algorithm which is used for achieving any plane display is illustrated in the following pseudocode, this is called bi-linear interpolation:

1. Choose a resolution RES, i.e. the number of points in each direction in the image that is desired displayed.

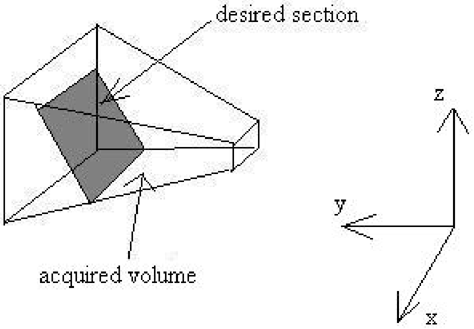


Figure 7.1: Illustration of a two-dimensional section in a acquired volume from a three-dimensional ultrasound recording. The desired plane can be fully described by its four corners.

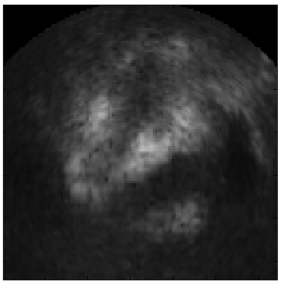


Figure 7.2: The two-dimensional section shown in Figure 7.1 from the recording in Figure 3.4.

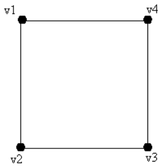


Figure 7.3: Positions of the corner vertices in a two-dimensional section from a three-dimensional volume. Notice that the vertices are placed counter-clockwise.

2. Compute the cartesian coordinates for all the points in the plane specified by the four corner vertices shown in Figure 7.3. It is important that the vertices are positioned counter clockwise.

```
    (a) For each $i$ from 1 to $RES$
    (b) $pos_{12} = \frac{(i-1)v_2+(RES-i)v_1}{RES-1}$ and $pos_{43} = \frac{(i-1)v_3+(RES-i)v_4}{RES-1}$
    (c) For each $j$ from 1 to $RES$
    (d) $locations(i,j) = \frac{(j-1)pos_{43}+(RES-j)pos_{12}}{(RES-1)}$
    (e) Next $j$
    (f) Next $i$
3. Scanconvert the points specified in $locations$ as explained in chapter 3.1.4.
```

Step 2b calculates the positions by each row according to the desired resolution, and step 2d does the same thing for each column and stores it in a 2 × 2 matrix.

All the ultrasound images displayed in this thesis are generated with this method.

7.2 Outlining Right Ventricular Contours Manually

For manual segmentation of the right ventricle from the ultrasound recordings, a number of short axis slices is used. The short axis plane is shown in figure 7.4. The same cross section from a human heart is shown in figure 7.5, and a short axis ultrasound image of the heart is shown in figure 7.6.

In order to figure out which ranges that covers the right ventricle, a long-axis view of the heart is obtained from the three-dimensional data. From this view it is possible to mark the position of the apex and the base along the y-axis in the right ventricle.

The range number is calculated from the formula:

$$range = \frac{x_i r_{inc} + r_{min}}{\frac{r_{inc}}{res_r}} \tag{7.1}$$

where x_i is the location of the point in the image which has the size $RES \times RES$ and

$$r_{inc} = \frac{r_{max} - r_{min}}{RES} \tag{7.2}$$

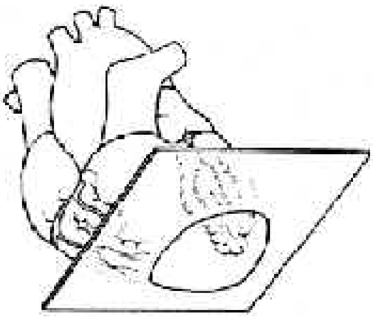


Figure 7.4: Illustration of the two-dimensional short-axis plane. In this cross-sectional view the right ventricle have a crescent shape, which is a characteristic feature of the right ventricle. Figure taken from [20].

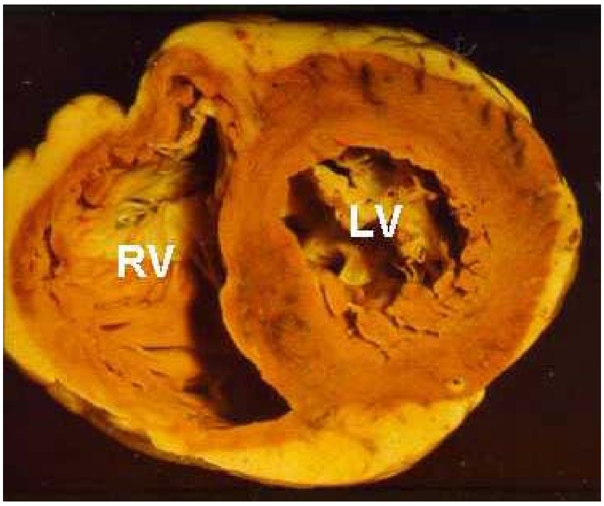


Figure 7.5: Transverse cross section of a human heart, this section corresponds to the plane in 7.4 for echocardiography imaging. Figure from [48].

where r_{max} and r_{min} are the maximum and minimum distances which are stored in the recording.

$$res_r = \frac{Total\ Ranges}{RES} \tag{7.3}$$

where RES is the resolution in the image from the algorithm in chapter 7.1.

The short-axis slices are calculated by keeping the range constant when specifying the vertices in the algorithm described in in chapter 7.1. This is possible when the data is obtained from a probe in the apical position. The vertices used for obtaining the slice in figure 7.8 is: $v_1 = (az_{min}, r, el_{min})$,

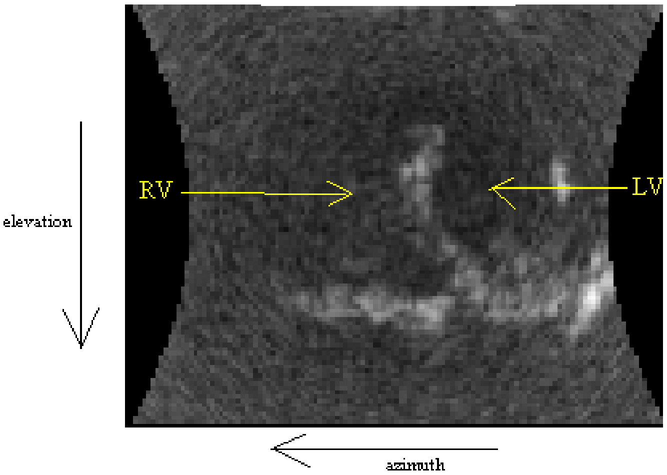


Figure 7.6: Short axis plane from echocardiography. The chambers are indicated. Notice the weak representation of the right ventricle free wall, which is a problem in echocardiography and introduce challenges in automatic segmentation of the right ventricle. The picture is rotated to the same orientation as figure 7.5.

$v_2 = (az_{min}, r, el_{max})$, $v_3 = (az_{max}, r, el_{max})$, $v_4 = (az_{max}, r, el_{min})$

When the maximum and minimum ranges are determined, short-axis slices of the right ventricle are extracted with a distance of 10 ranges between each contour. This is a spacing of approximately $3 - 4$ mm, and is sufficiently small to be able to visualize the shape of the right ventricle which has a length of about $7 - 10$ cm.

From the short-axis slices the contour around the ventricle is outlined by manually positioning 20 points at important features around the right ventricle and thereby outlining the short axis contour. This is illustrated in figure 7.8, notice the problems with finding clear contours in the right ventricular free wall. In figure 7.8 the 20 points are connected with straight lines, forming a closed polygon. The contours are obtained both from the end-systole and the end-diastole frame. Since the points are placed manually, there is no guarantee that they are equally spaced around the contour. This is however not required because it is not the points itself that are used further but rather the contours made by the straight lines between the points. The outlined contours from all the short-axis slices in the end-diastole and end-systole of one right ventricle is illustrated in figure 7.9.

End-diastole is defined as the frame where the right ventricle has the maximum volume, the end-systole is defined as the frame where the right

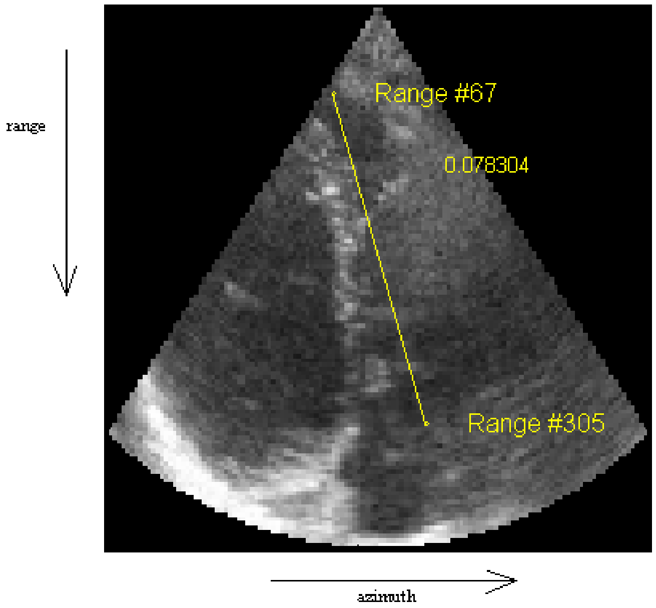


Figure 7.7: Long-axis view to obtain the range of the apex and the base of the right ventricle. The plane in the middle of the recording is used ($azimuth = 0$). The number in the middle of the figure is the distance between the two points expressed in meters.

ventricle has the minimum volume.

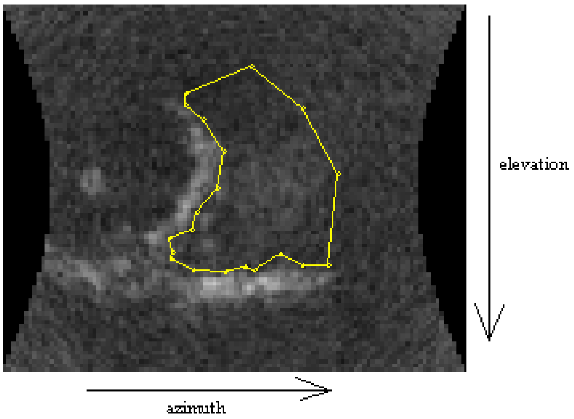
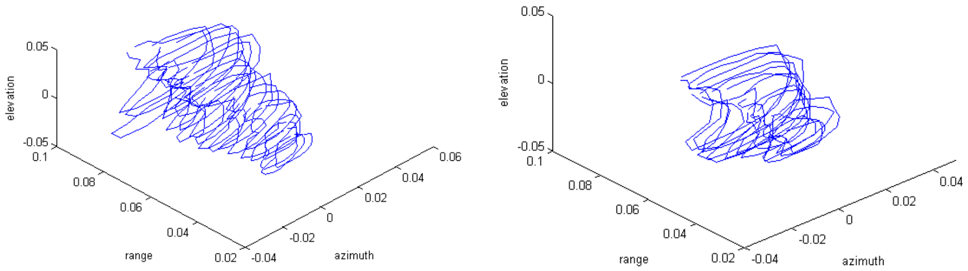


Figure 7.8: The right ventricle manually outlined in a short-axis slice. 20 points are manually placed at important features. Notice the problems of finding good points in the free wall.



(a) End-diastole

(b) End-systole

Figure 7.9: Manually segmented contours of the right ventricle in end-diastole and end-systole. Notice the way the septum bulges in to the ventricle, and that the main changes during the beat is the long-axis shortening.

7.3 Approximating the volume from slices

The contours obtained in the previous chapter are used for volume approximations of the manually segmented right ventricle. The approach taken is to calculate the area of each contour and then multiply it with the distance to the next contour. This is an approximation in which it divides the ventricle into discrete disks. The distance between each contour is however relatively short so the approximation does not introduce severe errors.

Green's plane theorem can be used as a basis to calculate the area of a polygon [14], shown in Figure 7.8 and a general version in Figure 7.10.

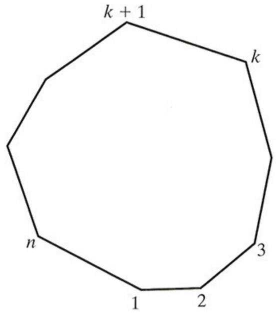


Figure 7.10: A polygon with n vertices in the xy-plane. The vertices are also referred to as control points. Figure from [14].

Green's plane theorem:

$$\int\int_{area} (\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}) dxdy = \oint_C f_1 dx + f_2 dy \qquad (7.4)$$

where f_1 and f_2 are two continuous functions defined in the planar area bounded by the curve C. The area of a planar region can be expressed by setting $f_1 = 0$ and $f_2 = x$.

$$\int\int_{area} dxdy = \oint_C xdy \tag{7.5}$$

It is possible to express the polygon as a parametric equation, using the control points.

$$x = x_k + (x_{k+1} - x_k)u \tag{7.6}$$

$$y = y_k + (y_{k+1} - y_k)u \tag{7.7}$$

for $0 \leq u < 1$, $k = 1, 2, ..., n$ where $x_{n+1} = x_1$ and $y_{n+1} = y_1$. $(x_{k+1} - x_k)$ expresses the difference between to adjacent control points, and u is the parameter. If the equation for the y-coordinate are differentiated with respect to u it becomes:

$$\frac{dy}{du} = (y_{k+1} - y_k) \tag{7.8}$$

and thus

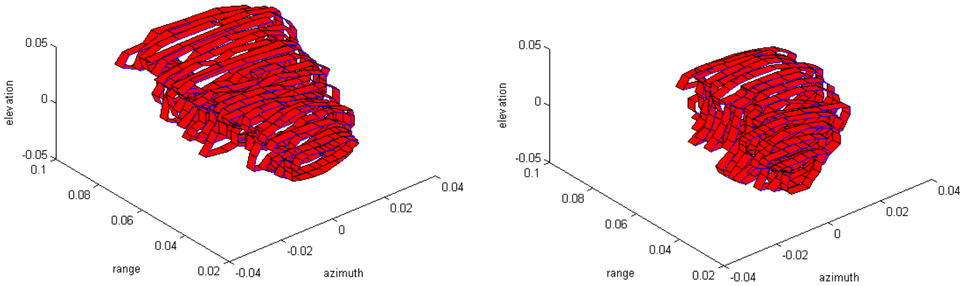
$$dy = (y_{k+1} - y_k)du \tag{7.9}$$

Inserting this and equation 7.7 into equation 7.5 an expression for the area can be obtained.

$$\begin{aligned} A &= \oint_C xdy && (7.10)\\ &= \sum_{k=1}^{n} \int [x_k + (x_{k+1} - x_k)u](y_{k+1} - y_k)du && (7.11)\\ &= \sum_{k=1}^{n} (y_{k+1} - y_k)[x_k + \frac{(x_{k+1} - x_k)}{2}] && (7.12)\\ &= \frac{1}{2}\sum_{k=1}^{n} (x_k y_{k+1} - x_k y_k + x_{k+1} y_{k+1} - x_{k+1} y_k) && (7.13) \end{aligned}$$

This is a telescoping series in which every second and third term cancels with each successive k. This gives the following expression for the area of the polygon in Figure 7.10.

$$A = \frac{1}{2}\sum_{k=1}^{n} (x_k y_{k+1} - x_{k+1} y_k) \tag{7.14}$$



(a) End-diastole

(b) End-systole

Figure 7.11: Right ventricle represented by disks for volume approximation.

This area is calculated for each slice and then multiplied with the distance between each slice. The distance between each slice is found by:

$$h = \frac{(r_{max} - r_{min})}{\frac{(R_{MAX} - R_{MIN})}{10}} \tag{7.15}$$

where r_{max} and r_{min} are the ranges in meter and R_{MAX} and R_{MIN} are the distances in range numbers. The factor 10 comes from that each disk is ten ranges thick. The expression for the volume is then given by:

$$V = \sum_i A_i h, \text{ where } i \text{ is the set of all disks} \tag{7.16}$$

The volume of these disks are summed together to form the whole volume. Figure 7.11 illustrates the approximated right ventricular volume. The equation 7.16 is a left Riemann sum and would converge to an integral if the distances h are infinitely small [23]. The MATLAB code used for this volume calculation is attached in the appendix.

Chapter 8

Three-dimensional geometrical model of the right ventricle

8.1 Geometrical models

In contrast to the left ventricle does not the right ventricle possess a simple geometrical shape, and thus modeling is more challenging. The right ventricle have many variations in shape and it has irregular trabeculations, a separate infundibulum and variations in shape from different loading conditions [39]. The right ventricular anatomy is described in chapter 2.4.1. The changes during the systole can also be troublesome to incorporate in to a model.

The models are made by an initial mesh for which subdivision has been performed. The subdivision scheme used is the direct Doo-Sabin scheme for arbitrary topology (chapter 5.3.2).

8.2 Manually shaped model

The right ventricle is manually segmented from short-axis slices as described in chapter 7.2 and shown in Figure 7.9. These contours is consequently obtained from constant ranges.

To make a model of the right ventricle the manually segmented slices are used. The shape of the right ventricle are made up of a mesh, which consists of nodes connected to each other and this is forming faces. The model are based upon a basic shape, which is a modification of the truncated ellipsoid model used for left ventricle segmentation as described in [34]. This shape is illustrated in Figure 8.2. The modification is that the left side has a bulge in to account for the ventricular septum, giving the crescent shape in the short-axis view. The thick black borders in Figure 8.2 shows the surface patches. In contrast to the model described in [34] this model has 27 surface patches, from corresponding 27 nodes. These nodes are placed in successive rings around the model. The nodes are placed tighter towards the base to account for the geometrical change in shape during systole [39], and the sparse resolution in the apex is to account for the heavy trabeculation and noise present in images in the right ventricle. An illustration of the need of having lower resolution in the apex part is illustrated in Figure 8.1. This can be viewed as an attempt to try to incorporate the division of the ventricle into three parts as illustrated in Figure 2.6.

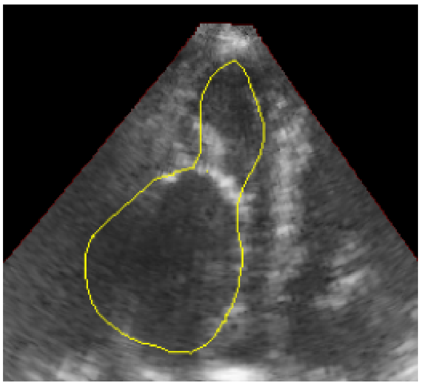


Figure 8.1: Right ventricle tracking using a model with higher resolution in the apex part. Notice that the model attaches to the moderator band making the tracking erroneous.

The nodes are placed according to the manually segmented contours. Each of the nodes are moved such that they fit to the contour. Where the nodes are placed such that the rightmost node is on the rightmost place in the contour, the next node is the node in ”the corner” on the upper side of the ventricle, following by a node at the upper corner of the crescent shape, one node in the deepest bulge from the ventricular septum, the next is at the lower corner of the crescent shape, and the last one is down to the right in the ventricle.

By following these rules it is possible to calculate a mean shape based on these nodes. And the relative placement between the nodes is important to prevent a malformed model, which an arbitrary node positioning is more likely to do. Figure 8.3 shows a ventricle that is shaped in the way described.

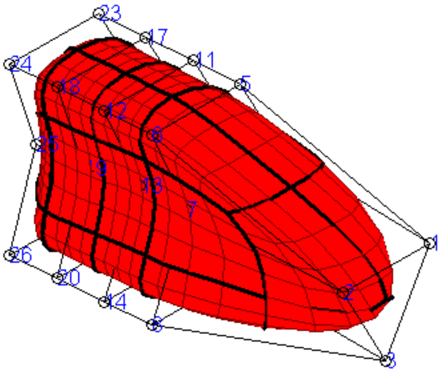


Figure 8.2: The modified truncated ellipsoid model used as a basis for manually shaping a right ventricular model. The long distance from the apex nodes to the next ring of nodes is to cope with the heavy trabeculated area in the apex. There are three nodes in the apex to round of the tip, which have the tendency to become too sharp if only a single apex node is used.

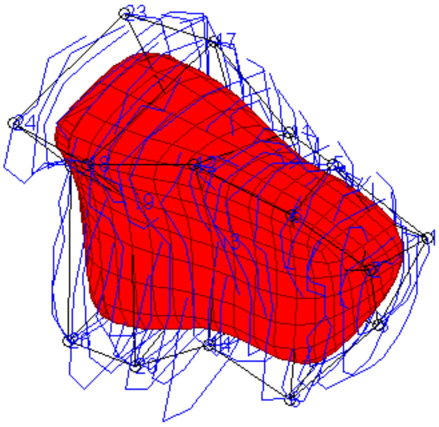


Figure 8.3: A manually shaped model of the right ventricle plotted together with the manually outlined contours of the right ventricle. The subdivision scheme used is the Doo-Sabin scheme incorporated in the RCTL application. The smooth model is much smaller than the contours, this is due to the approximating nature of this scheme. However it is possible to scale the model to get the same size at the contours.

8.2.1 Normalizing the models

Before calculating a mean model of the of the manually fitted right ventricular models, by using equation 8.5 for calculating the mean model. The model has to be adjusted to avoid that the position in space has an effect on the mean model. The issue of the mean shape is the nodes relative position to each other and not relative to the coordinate system. The method used for avoiding this is to translate the model so that its middle point lies in the origin. The middle point (x_0, y_0, z_0) is defined as:

$$x_0 = \frac{x_{max} + x_{min}}{2}, y_0 = \frac{y_{max} + y_{min}}{2}, z_0 = \frac{z_{max} + z_{min}}{2}, \tag{8.1}$$

where x_{max}, y_{max}, and z_{max} are the maximal values of the coordinates and x_{min}, y_{min}, and z_{min} are the minimal values.

To move the model so that the middle point (x_0, y_0, z_0) gets in the origin, a translation matrix is applied [14]. The translation matrix is given by:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.2}$$

where $t_x = -x_0$, $t_y = -y_0$, $t_z = -z_0$. By expressing the coordinates for one node as a vector $\mathbf{x} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}$, the new position can be written:

$$\mathbf{x'} = \mathbf{Tx} \tag{8.3}$$

where $\mathbf{x'} = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix}$ is the coordinates for the node when the middle point lies in the origin.

8.2.2 Mean shape

The mean shape of the models from a training set of manually fitted models, can be a good suggestion for a model used for model-based segmentation of the right ventricle.

For a model with n nodes, the shapes are represented with an $3n$-dimensional vector:

$$\mathbf{x}_i = \begin{bmatrix} x_{i0} & y_{i0} & z_{i0} & \dots & x_{in} & y_{in} & z_{in} \end{bmatrix}^T \tag{8.4}$$

where the j'th node in the i'th shape has coordinate (x_{ij}, y_{ij}, z_{ij}).

The mean model is determined by averaging the $\mathbf{x}_i$ vectors:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \tag{8.5}$$

Chapter 9

Real-Time Contour Tracking Library

The *Real-Time Contour Tracking Library* (RCTL) is a framework initially made for real-time tracking of the left ventricle in three-dimensional ultrasound. The tracking is performed as a sequential state estimation with an extended Kalman filter, see chapter 6.2.1 and [32]. The tracking represents a connection between subdivision surfaces and the Kalman filter. This framework is inspired by a suggestion in [3] and is described in several papers by Orderud [32] [33] [34].

In this thesis the RCTL is used for performing the real-time three-dimensional segmentation of the right ventricle based on the model described in this thesis. The segmentation is performed both exclusively on the right ventricle and together with the segmentation of the left ventricle. In the latter case the two models are connected to each other by a global pose transform.

9.1 State estimation with deformable model

The model has two notions for denoting the deformation and placement, the local deformations and the global transform. The local deformations $\mathbf{T}_l(\mathbf{x})_l$ describes the deformations by moving the control vertices, while the global transform $\mathbf{T}_g(\mathbf{x}_g, \mathbf{p}_l)$ represents the position, scale and orientation of the model. These two are connected together to a composite object deformation $\mathbf{T}(\mathbf{x}) = \mathbf{T}_g(\mathbf{T}_l(\mathbf{x}_l), \mathbf{x}_g)$. A state vector is made up of the global transform and the local deformation $\mathbf{x} = [\mathbf{x}_g^T, \mathbf{x}_l^T]^T$, this state vector is used in the framework for tracking.

The state estimation is based upon a kinematic model which predicts an a priori state $\hat{\mathbf{x}}_{k+1}^-$, with covariance matrix $\hat{\mathbf{P}}_{k+1}$ This model calculates a difference between the predicted state and a mean state $\mathbf{x}_0$, from the previous estimated states:

$$\hat{\mathbf{x}}_{k+1}^- - \mathbf{x}_0 = \mathbf{A}_1(\hat{\mathbf{x}}_k - \mathbf{x}_0) + \mathbf{A}_2(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_0) + \mathbf{B}_0\mathbf{w}_k \tag{9.1}$$

where B_0 is the covariance of the process noise, and together with $\mathbf{A}_1$, $\mathbf{A}_2$ this can be used for tuning the kinematic properties by adjusting the coefficients of the matrices.

The deformable model is represented in the framework by its control vertices $\mathbf{q}_i$ for $i \in \{1 \cdots N_q\}$, which are connected together in faces, for representing the topology. The faces are placed in a face list $C(c), c \in \{1 \cdots N_c\}$. Each control vertex has an associated displacement vector $\mathbf{d}_i$ which defines the direction for which the control vertices are allowed to move. The deformable models are constructed by the use of the Doo-Sabin extension to arbitrary topology (see chapter 5.3.2).

The framework requires the surface points $\mathbf{p}_i$, the normal vectors $\mathbf{n}_i$ and the Jacobi-matrices $\mathbf{J}_i$ to be able to predict the a priori state estimate $\hat{\mathbf{x}}_l^-$. It is possible to calculate the mentioned properties by first calculating the control points:

$$\mathbf{q}_i = \bar{\mathbf{q}}_i + x_i\mathbf{d}_i \tag{9.2}$$

where $\bar{\mathbf{q}}_i$ is the mean position of the control vertices, x_i is each vertices state and $\mathbf{d}_i$ is the displacement vector. The surface points can then be calculated as:

$$\mathbf{p}_l = \sum_{i \in C(c_l))} \mathbf{b}_i\mathbf{q}_i \tag{9.3}$$

where the $\mathbf{b}_i$ is the basis function for each control vertex within the surface patch of each surface point, see chapter 5.3.2. The normal vectors can then be expressed by a cross product between partial derivatives of the basis function in each direction u and v within the surface patch:

$$\mathbf{n}_l = \sum_{i \in C(c_l))} (\mathbf{b}_u)_i\mathbf{q}_i x \sum_{i \in C(c_l))} (\mathbf{b}_v)_i\mathbf{q}_i \tag{9.4}$$

Finally the Jacobian matrices can be described by:

$$\mathbf{J}_l = [\mathbf{b}_{i1}\mathbf{d}_{i1}, \mathbf{b}_{i2}\mathbf{d}_{i2}, \cdots]_{i \in C(c_l)} \tag{9.5}$$

in which the Jacobian matrices are padded by zeros for columns corresponding to control vertices outside the region of support for the surface patch of each surface point.

The real-time property of this framework is supported by the fact that the possibility to precalculate the basis functions. This allows fast tracking of the contour.

9.2 Edge measurements

The edge measurements are performed by a search in the normal direction for each surface point. These measurements are compared against the predicted model from the kinematic model. This leads to a normal displacement expressed as a normal projection of the distance between the observed point and the predicted point, through the normal vector.

$$v = \mathbf{n}^T(\mathbf{p}_{obs} - \mathbf{p}) \tag{9.6}$$

where $\mathbf{n}^T$ is each points normal vector, $\mathbf{p}_{obs}$ is the edge measurement observed and $\mathbf{p}$ is the predicted edge point.

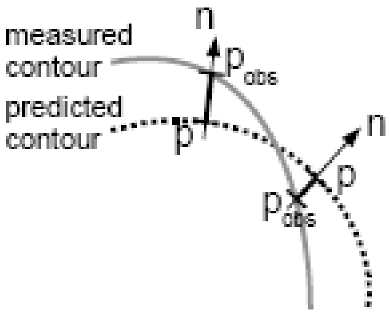


Figure 9.1: Edge measurements in the Real-Time Contour Tracking Library. The edge is detected along the normal of each surface point, and then the observed point is compared to the predicted surface point. Figure from [32].

Ultrasound images are often noisy and blurry and this complicates the edge detection. The edge detection implemented in RCTL is one called the step model. This model assumes edges to form a transition in image intensity, for one plateau to another, and calculates the edge position that minimizes the sum of the squared errors between the model and the data [32]. There is also an uncertainty measure connected to each normal displacement, called the measurement error r, which can be coupled to edge strength or other uncertainty measure in the edge detection.

To be able to use the normal displacement in the Kalman filter based tracking framework a linearization of the nonlinear deformation model is used. This can then be provided to the extended Kalman filter (see chapter 6.2.1). The linearization is made from a normal projection of the Jacobian matrix:

$$\mathbf{h}^T = \mathbf{n}^T\mathbf{J} \tag{9.7}$$

One challenge is the fact that we often have much more measurements than states, this can not be put into the ordinary Kalman filter, and the notion of the information filter are used [3]. The measurements are assumed to be independent and the linearized measurements with noise can be expressed as:

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{v} = \sum \mathbf{h}_i r_i^{-1} v_i \tag{9.8}$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} = \sum \mathbf{h}_i r_i^{-1} h_i^T \tag{9.9}$$

The measurement updates are used with the Kalman gain:

$$\mathbf{K}_k = \hat{\mathbf{P}}_\mathbf{k}\mathbf{H}^T\mathbf{R}^{-1} \tag{9.10}$$

And the new state estimation can be expressed by:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \hat{\mathbf{P}}_k\mathbf{H}^T\mathbf{R}^{-1}\mathbf{v}_k \tag{9.11}$$

with the covariance matrix:

$$\hat{\mathbf{P}}_k^{-1} = (\hat{\mathbf{P}}_k^-)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \tag{9.12}$$

The Real-Time Contour Tracking Library framework can be summarized as in Figure 9.2.

A screenshot from the RCTL application is shown in Figure 9.3.

9.3 Attractors

A new feature in RCTL is the concept of attractors. The attractors are a kind of virtual edge points which overrules the normal edge detection for the nearest edge point. This concept can be useful for correcting parts of the

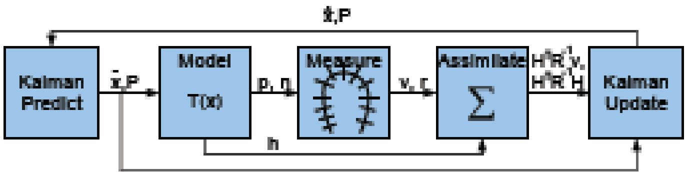


Figure 9.2: The contour tracking loop implemented in the Real-time Contour Tracking Library. The Kalman filter is used for prediction of the states of the model, which then is updated and compared against the measurements. Figure from [34].

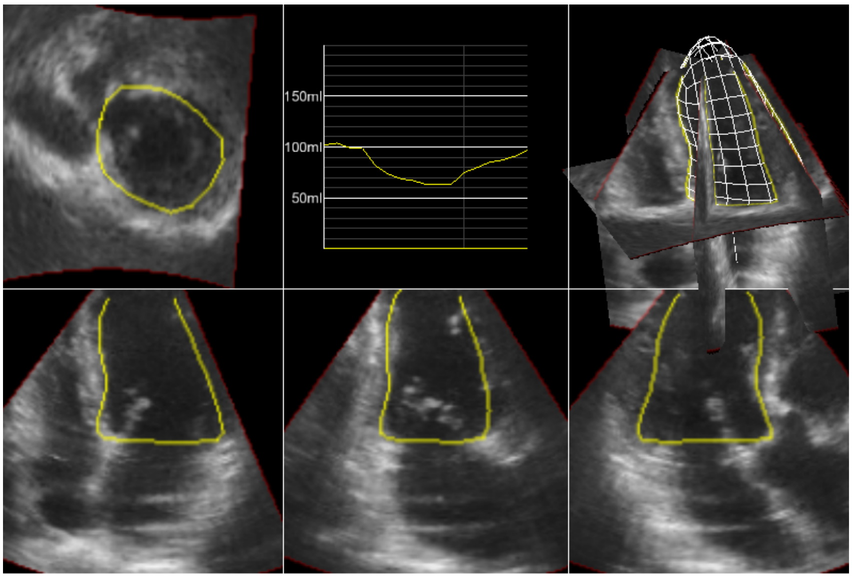


Figure 9.3: Screenshot from the Real-Time Contour Tracking Library. The measured volume curve are shown in the second column of the first row. In the third column of the first row a three-dimensional visualization is shown with four planes and a model of the segmented left ventricle. These four planes are shown in the other figures of the screenshot.

deformable model in areas of unclear or missing edges, which is difficult to detect with normal edge detection. An example of this is the apex of the right ventricle, which is not moving substantially through the heart beat. The attractors is however not so useful for positions which have a lot of movement through the beat such as the free outer wall of the right ventricle, this is because they are at the present time only implemented at fixed spatial positions.

Part III

Results

Chapter 10

Tracking of the right ventricle

10.1 Training Set

Figure 10.1 shows the training set made up of $N = 12$ manually shaped ventricles from N different recordings. The segmentation has been performed in the *end diastole* frame, i.e. when the ventricle is as it largest (see chapter 7.2). Figure 10.1 illustrates how the right ventricle can change from different individuals.

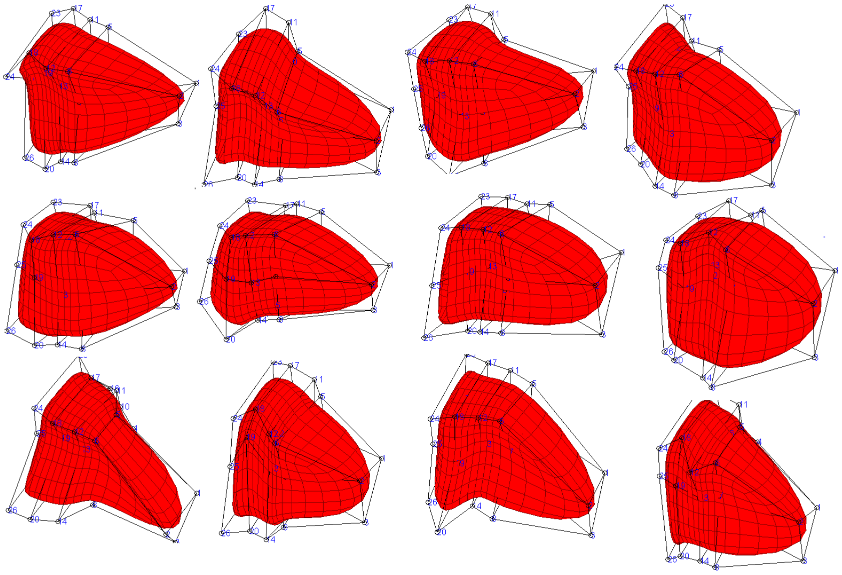


Figure 10.1: Set of manually shaped right ventricles. The procedure used is described in chapter 8.2.

10.2 Mean Shape Of The Manually Segmented Ventricles

In Figure 10.2 the mean shape of the right ventricle is illustrated. This model is obtained by applying equation 8.5 on the training set in Figure 10.1. The nodes are placed according to the rules described in chapter 8.2, this avoids malforming of the shape of the model. No malforming is apparent in Figure 10.2. This model is used further in the tracking of the right ventricle. The positions of the nodes are given in appendix B.

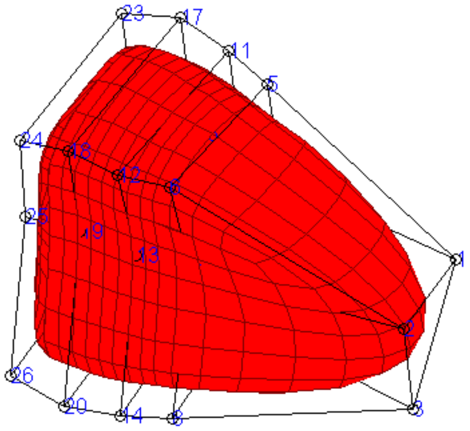


Figure 10.2: Mean shape calculated from the training set in Figure 10.1. The shape resembles an ellipsoid with a crescent shaped side and a slightly bending of the apex towards the left ventricle. The model is also somewhat flatter at the bottom side of the illustration than at the top.

10.3 Tracking of the right ventricle

The purpose of this section is to test how the various initial models affect the segmentation. Only 8 of the recordings in the training set contained a whole satisfactory cycle of the heart. Therefore only these eight have been used in the tracking of the right ventricle. There are used four types of initial models:

1. Model that is manually fitted for each recording according to chapter 8.2. All the models have edge resolution of 675 edge profiles.

2. The mean shape model in Figure 10.2 with edge resolution of 675 edge profiles.

3. The mean shape model with the smallest edge resolution possible, where one edge corresponds to one node (27 edges).

4. The mean shape model with a high edge resolution (2700 edges).

All of the segmentations has been performed by using the RCTL tool (chapter 9). The models has been placed according to the following configuration in the .xml-file for RCTL.

```
<transform type="full" state="0.01 0.045  0  0.06 0 0 0"
 damping="0.8  0.5  0.8" regularization="0.05  0.15  0.10"
 noise="0.10 0.20 3.13">
<!-- RV model -->
<model type="cell" parameters="RV-Model.h5" state="" stiffness="0"
     resolution="5" damping="0.8"
     regularization="0.2" noise="2">
<edge type="step" samples="30" spacing="0.0010" noise="0.0400"
  threshold="15" neigh_threshold="0.01"/>
</model>
```

10.3.1 Manual shaped models

The result from the segmentation with the manually shaped models, shown in Figure 10.1, to each respective recording is shown in the following tables and figures. In table 10.1 the clinical parameters from automatic segmentation of the right ventricle with manually shaped right ventricles are presented and compared with the results from the manual segmentation.

The manual segmented volume by the disk method explained in chapter 7.3 is plotted against the automatic segmented volume from manually shaped models to its respective recordings. The following four tables shows the plots for end diastolic volume, end systolic volume, stroke volume, and ejection fraction, together with a line produced by linear regression to the points.

Table 10.1: Result of the segmentation of the right ventricle using the RCTL application with manually fitted models, expressed by the clinical parameters, *End Diastolic Volume* (EDV) [*ml*], *End Systolic Volume* (ESV) [*ml*], *Stroke Volume* (SV) [*ml*], and *Ejection Fraction* (EF) [%]

	Manual segmentation	Automatic segmentation
Recording	EDV ESV SV EF	EDV ESV SV EF
7A9BIB00	132.8 74.0 58.8 44.3	124.3 66.7 57.6 46.3
7A9F7JG2	101.9 63.4 38.5 37.8	105.5 62.1 43.4 41.1
7A9F7P84	91.3 62.3 29.0 31.8	85.8 50.3 35.5 41.4
7A9F7900	118.1 80.9 37.2 31.5	116.8 69.2 47.6 40.8
7A9FFM06	104.8 76.3 28.5 27.2	98.8 59.9 27.3 27.6
7A9FFR88	87.1 54.4 32.7 37.5	87.2 55.2 32.0 36.1
7A9FPBOA	103.1 65.2 37.9 36.8	102.7 53.7 49.0 47.7
7A9FPLGC	129.4 66.8 62.6 48.4	129.8 68.9 60.9 46.9

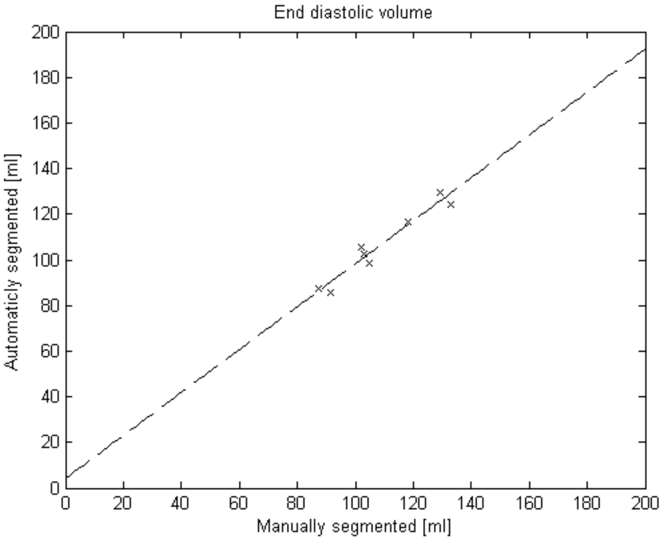


Figure 10.3: End diastolic volume with manually shaped models with each respective recording. Linear regression yields: $y = 0.9405x + 4.2546$. Correlation coefficient: $r^2 = 0.9414$

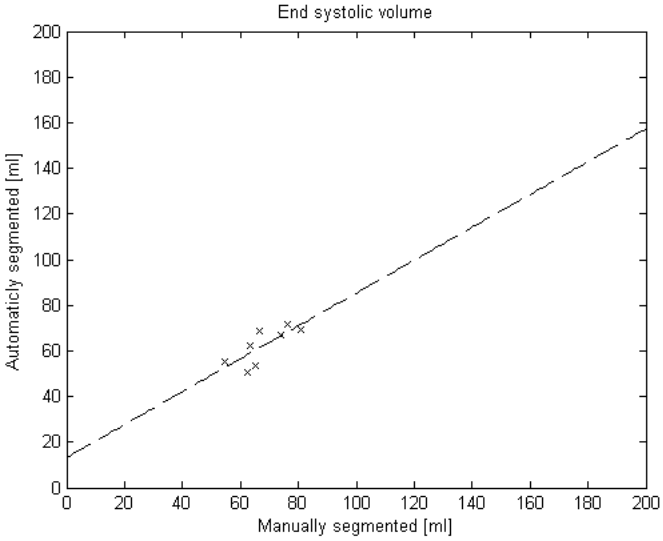


Figure 10.4: End systolic volume with manually shaped models with each respective recording. Linear regression yields: $y = 0.7199x + 13.3073$. Correlation coefficient: $r^2 = 0.5805$

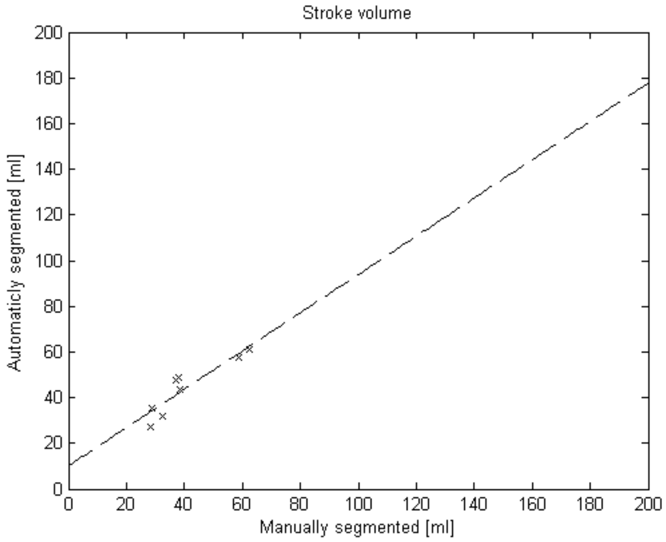


Figure 10.5: Stroke volume with manually shaped models with each respective recording. Linear regression yields: $y = 0.8379x + 10.1031$. Correlation coefficient: $r^2 = 0.8265$

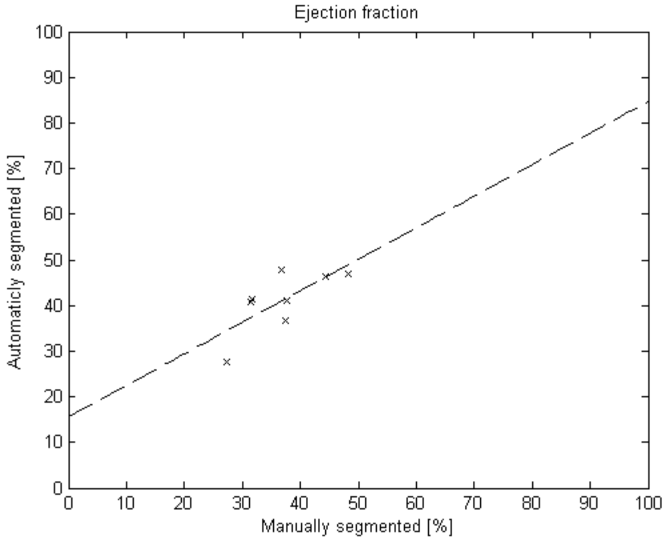


Figure 10.6: Ejection fraction with manually shaped models with each respective recording. Linear regression yields: $0.6908x + 15.5795$. Correlation coefficient: $r^2 = 0.5258$

Table 10.2: Result of the segmentation of the right ventricle using the RCTL application with the mean shape model shown in Figure 10.2, expressed by the clinical parameters, *End Diastolic Volume* (EDV) [*ml*], *End Systolic Volume* (ESV) [*ml*], *Stroke Volume* (SV) [*ml*], and *Ejection Fraction* (EF) [%]

	Manual segmentation	Automatic segmentation
Recording	EDV ESV SV EF	EDV ESV SV EF
7A9BIB00	132.8 74.0 58.8 44.3	122.1 68.9 53.2 43.5
7A9F7JG2	101.9 63.4 38.5 37.8	99.7 48.7 50.9 51.1
7A9F7P84	91.3 62.3 29.0 31.8	91.2 51.7 35.4 40.8
7A9F7900	118.1 80.9 37.2 31.5	115.2 62.5 52.6 45.7
7A9FFM06	104.8 76.3 28.5 27.2	101.7 60.6 41.1 40.4
7A9FFR88	87.1 54.4 32.7 37.5	85.5 41.1 44.4 52.0
7A9FPBOA	103.1 65.2 37.9 36.8	101.7 52.6 49.0 48.2
7A9FPLGC	129.4 66.8 62.6 48.4	124.8 56.8 67.9 54.4

10.3.2 Mean shape model

The mean shape model, shown in Figure 10.2, has been used for automatic segmentation on the same recordings as in the previous chapter. Figure 10.7 shows one example of the segmentation, showing the edge detections.

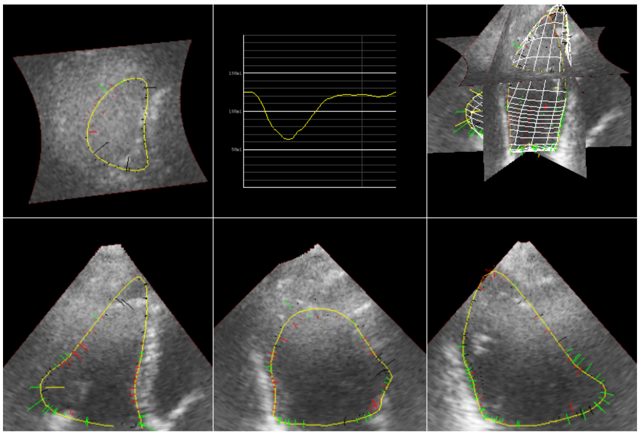


Figure 10.7: Screenshot of tracking of the right ventricle with the mean model with a medium high resolution (675 edges)

The results for the clinical parameters for the automatic segmentation of the right ventricle is expressed in the following table and figures.

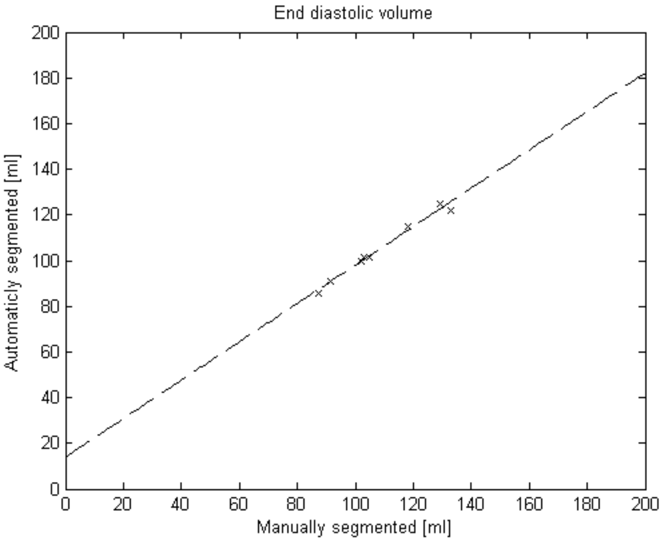


Figure 10.8: End diastolic volume with the mean model. Linear regression yields: $y = 0.8404x + 14.0049$. Correlation coefficient: $r^2 = 0.9825$.

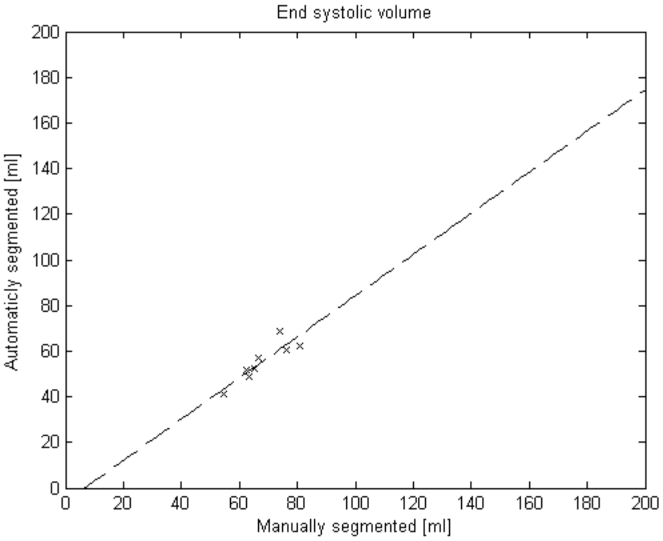


Figure 10.9: End systolic volume with the mean model. Linear regression yields: $y = 0.9020x - 5.8971$. Correlation coefficient: $r^2 = 0.7932$.

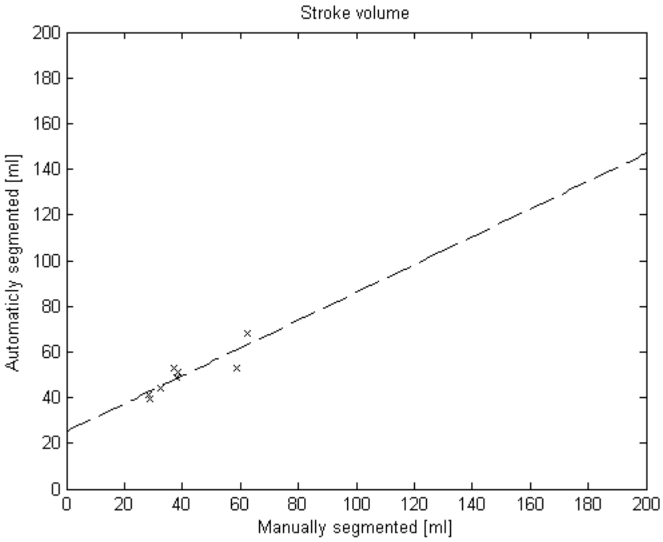


Figure 10.10: Stroke volume with the mean model. Linear regression yields: $y = 0.6085x + 25.1392$. Correlation coefficient: $r^2 = 0.7759$.

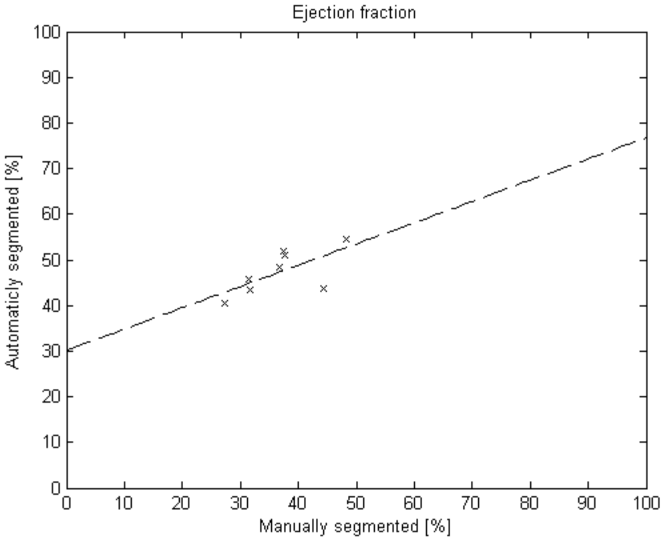


Figure 10.11: Ejection fraction with the mean model. Linear regression yields: $y = 0.4655x + 30.1833$. Correlation coefficient: $r^2 = 0.4335$.

10.3.3 Edge profile experiment

In Figure 10.12 a screenshot of the RCTL application for segmentation of the right ventricle with a model with a few number of edge profiles. Notice the jagged profile of the volume curve, this indicates the random nature of the segmentation when one have to few edge detections. The volume measurements are very unstable.

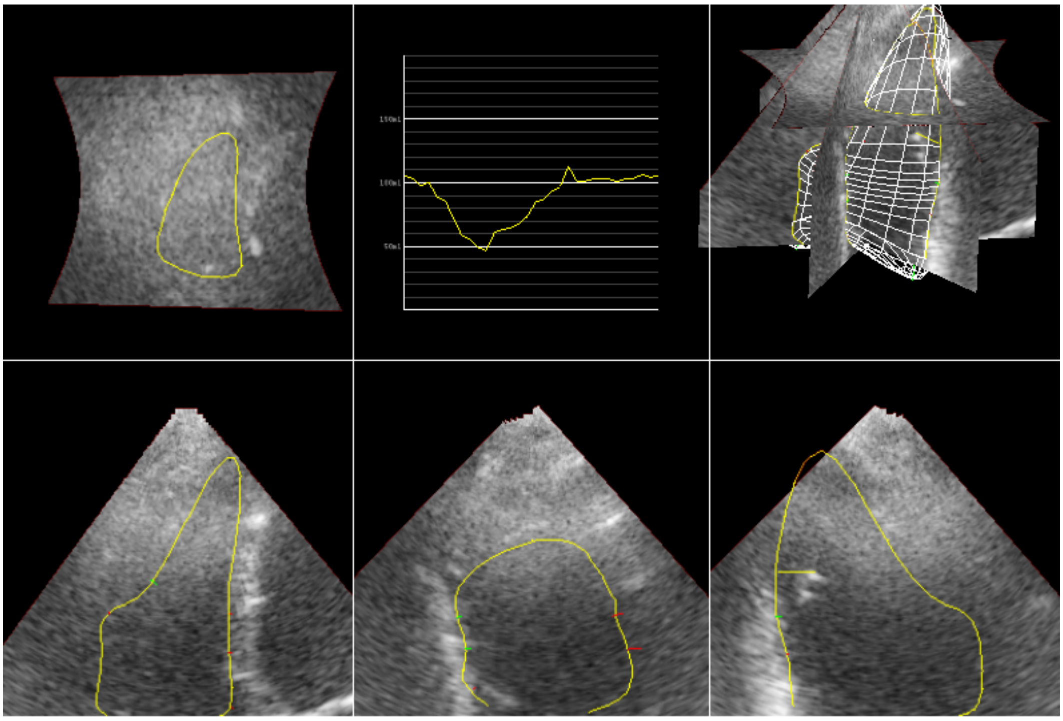


Figure 10.12: Screenshot of tracking of the right ventricle with a model with a few number of surface points (27 edges). Notice the jagged shape of the volume curve.

In Figure 10.13 a screenshot of the RCTL application for segmentation of the right ventricle with a model with many edge profiles. Notice the smooth profile of the volume curve, this indicates the more stable nature of the volume measurements, because of the high number of edge detections.

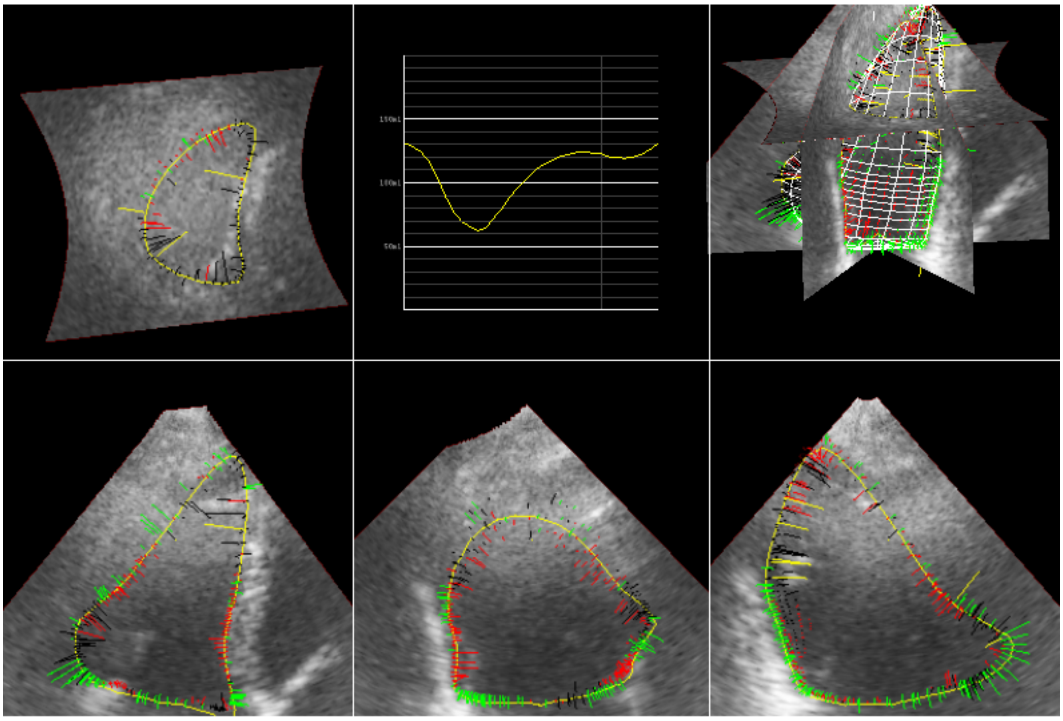


Figure 10.13: Screenshot of tracking of the right ventricle with a model with a high number of surface points (1227 edges). Notice the very smooth shape of volume curve.

10.3.4 Initial Position Experiment

The effect of the initial position in the global transform has been investigated. Figure 10.14 shows a fairly good segmentation with the initial position $x = 0.01, y = 0.045, z = 0$. The initial position is an important factor when using the automatic segmentation. Figure 10.15 and 10.16 shows what happens when the initial position gets to far out to the left and the right. The two positions in the two figures are the smallest deviation from the position in Figure 10.14 for which the model tracks erroneous features, such as the left ventricle (Figure 10.15) and outside of the heart (Figure 10.16).

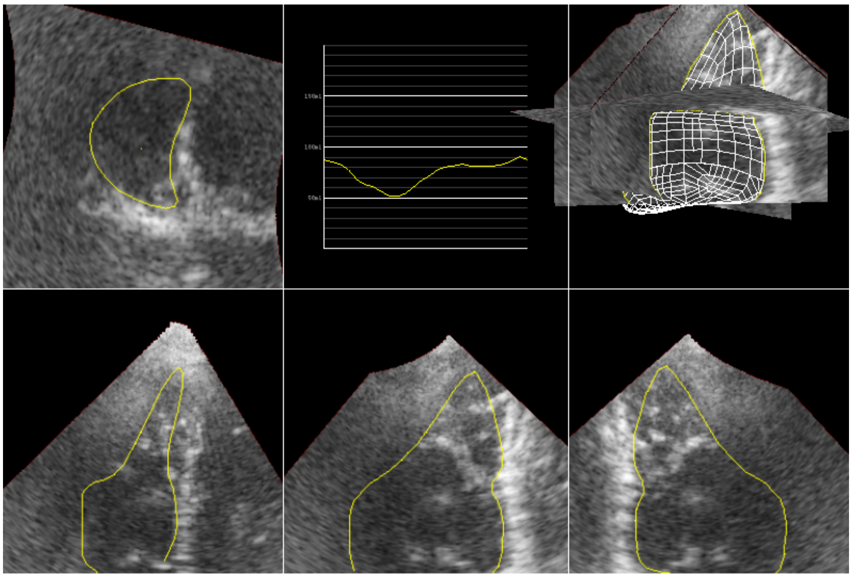


Figure 10.14: A fairly good segmentation with the initial position in the global transform $x = 0.01, y = 0.045, z = 0$

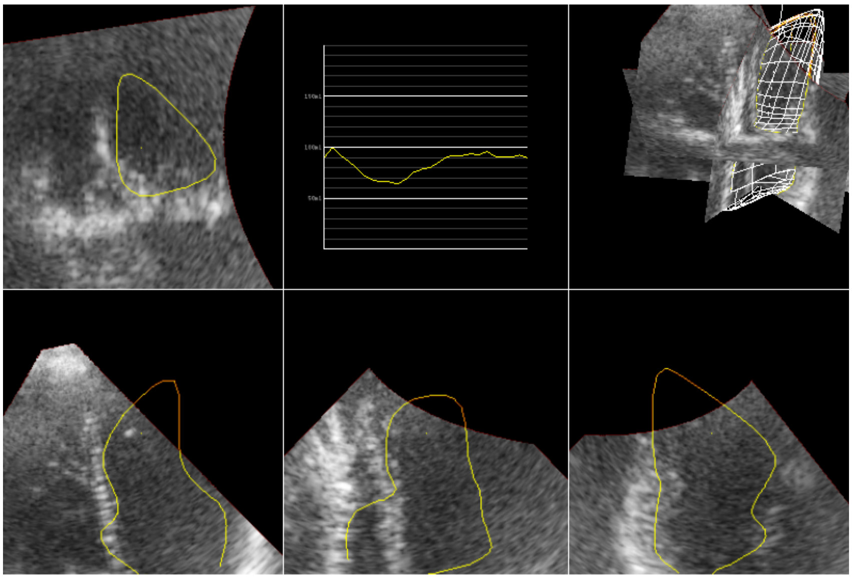


Figure 10.15: A segmentation with the initial position in the global transform $x = -0.03, y = 0.045, z = 0$. The contour is to far to the right and tracks the left ventricle.

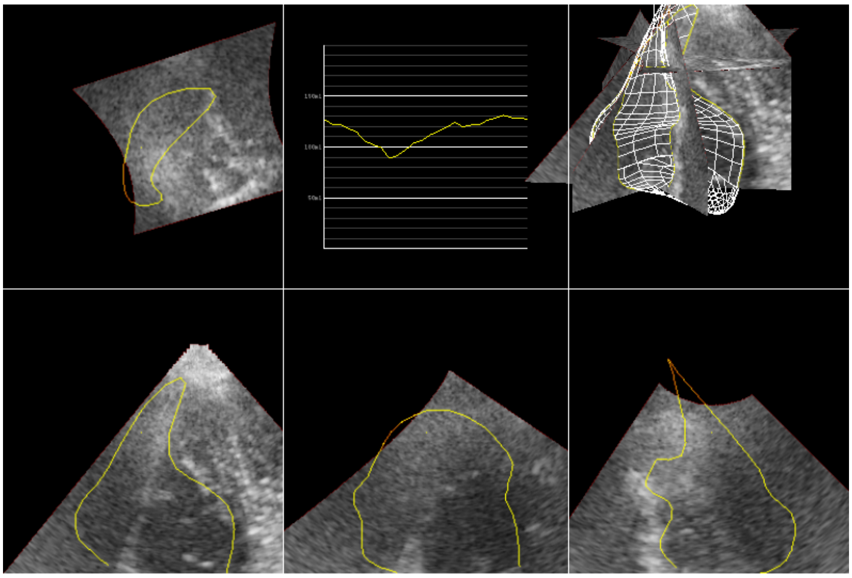


Figure 10.16: A segmentation with the initial position in the global transform $x = 0.03, y = 0.045, z = 0$. The contour is to far to the left and the apex part of the model tracks features on the outside of the heart.

10.3.5 Tracking of the right ventricle together with the left ventricle

The overall purpose over time with the framework of the Real-Time Contour Tracking Library is to be able to track two or more heart chambers simultaneously. This thesis is a small part on the road to this vision. Figure 10.17 shows a screenshot from the simultaneous segmentation of the two ventricles in the heart. For the left ventricle a truncated ellipsoid model is used [34], for the right ventricle the mean shape model in Figure 10.2 is used.

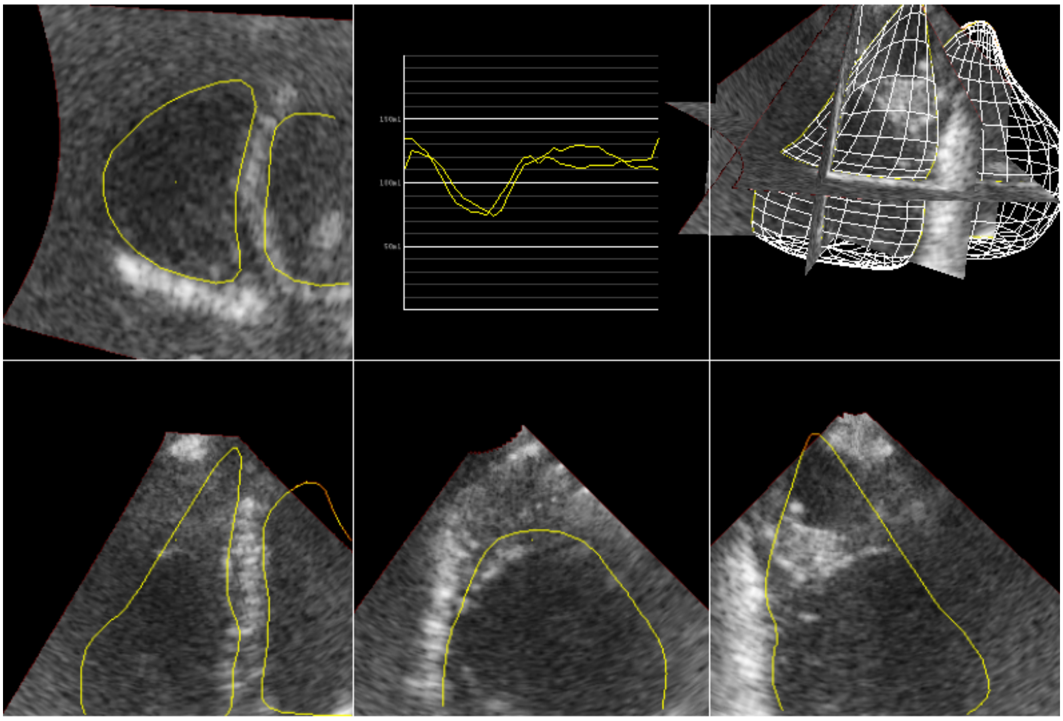


Figure 10.17: Segmentation of the left and right ventricle simultaneously. The left ventricle is partly outside of the image, but maintains its shape due to the initial model constraints. The volume curves are approximately the same for the two ventricles.

10.4 Experiment with attractors

To try out the new feature of the attractors in the Real-Time Contour Tracking Library, a segmentation with two attractors in the trabeculated apex part of the right ventricle. The same segmentation as shown in Figure 10.14 is used for the attractor trial. Figure 10.18 shows a better attraction to the apex of the right ventricle, the attractors are placed manually by right clicking in the window in the RCTL application.

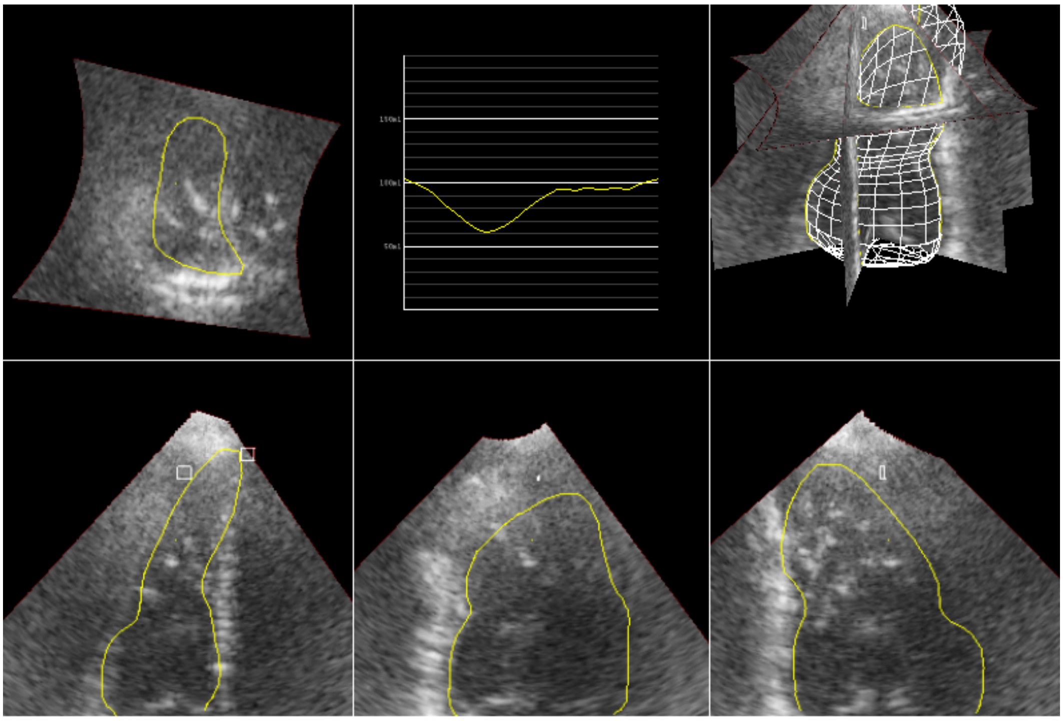


Figure 10.18: Screenshot of a segmentation with two attractors (chapter 9.3) in the trabeculated apex part of the right ventricle. The attractors are placed manually and is shown as two white squares in the bottom left image.

Part IV

Discussion and conclusion

Chapter 11

Discussion

The results suggest that there are substantial individual differences in the shape and size of the right ventricle. This, together with noisy images, makes the tracking of the right ventricle challenging. However, the use of a deformable model used to track the right ventricle is promising as the tracking results in this report proposes.

The mean shape turned out to be a good initial model that made fairly well segmentation in all the recordings in which it was tested. The shape of the mean shape is similar to that of the bent ellipsoid model, described in chapter 2.4.3, and with some work this model could possibly be used as an initial model for the right ventricular tracking. The mean shaped model obtained in this report has no mathematical foundation and is possible a little to heuristic, especially from the rather limited training set used.

The small number of recordings used in the development of the shapes comes from problems with getting good recordings of the right ventricle, which is not a prioritized chamber in clinical examinations. Many physicians also claims that it is hard to get good echocardiography recordings of the right ventricle. This small number imposes a problem for the statistical analysis of the feasibility of the segmentation. However, an attempt was made to try to express some statistical results of the right ventricular tracking.

The manual segmentation performed in this thesis is performed by the author, who has no clinical training so they are based upon my limited anatomical knowledge of the right ventricle, prominent features in the image and common sense. This yields, of course, not clinical useful results but are rather an indication of the volume of the right ventricle. The manual segmentation does however avoid noise features as the moderator band and the large papillary muscle in the apex, and does also capture the free wall to some extent where it is missing. So the relation between the automatic segmentation and the manual does show promising tendencies in the automatic real-time tracking. The comparison of the manual segmentation to the automatic segmentation can be viewed as an attempt to try to evaluate the automatic computer vision of the tracking against the human vision. In the future, the volume measurements should be validated against MRI images, which are considered to be the gold standard for volume measurements of the heart chambers.

The manually fitted models had a good fit to the manual segmentation, with a very good correlation coefficient ($r^2 = 0.9414$), for the end diastolic volume. The model was shaped from the contours of the manually segmented ventricle in the diastole, so this result is not so surprising, but still promising. The same, maybe even better ($r^2 = 0.9825$), good relation between the manual and the automatic segmentation using the mean shape model is shown. This can suggest that the mean model is absolutely feasible for the tracking of the ventricle in the diastole.

The results for the end systolic volume for both the mean shape model and the manually fitted model are however not so optimistic. This can be a sign of that the model did not incorporate the changes during heart cycle good enough, and some more effort could be made with this matter if the time has allowed it. Again, the small number of data can also be an important factor for the poor correlation, as one poor recording would yield a great negative effect on the correlation.

The results for the stroke volume and the ejection fraction shows that the errors in the end diastolic volume and in the end systolic volume between the manual and the automatic segmentation does not cancel each other, but rather have a negative effect on the stroke volume and the ejection fraction correlation.

The edge profile testing shows that it is necessary to have a certain number of edge profiles to get stable results. The results with the lowest number of edge profile gave very unstable volume measurements, while the high number of each profiles gave a very smooth, and pleasing segmentation. The number of edge profiles in the mean model seemed however to be sufficient to yield a good enough segmentation. A high number of edge profiles demands high computational power.

Initial positioning of the model was also investigated. In the segmentations in this project the initial positions are placed manually. Figure 10.15

and 10.16 shows examples of the effect of wrongfully placing the initial models. The contours tracked are not the right ventricle, but instead other features in the heart, there were a quite large area in which it was possible to position the model were the tracking was satisfactory (approximately $4-5cm$) this makes the tracking somewhat robust against erroneous initial positions. The results does however suggest that there should be a mechanism to find the right initial position. This is partly accomplished in the RCTL application when tracking two models similarly; the models are placed relative to each other which imposes a restriction of which features that is tracked. However, initial positioning for the whole system of contours is still needed. In the future an automatic initial position tracking mechanism could be a feature of the RCTL application.

Attractors which are a new and yet undocumented feature of the RCTL turned out to be a nice feature for adjusting the segmentation in difficult areas which contains little movement, such as the heavy trabeculated apex area.

The greatest challenges for segmentation did, as expected, turn out to be the trabeculations in the apex area and the poor imaging of the free outer wall of the ventricle. One additional concern were also the moderator band, which appeared as a very prominent feature in the images. The problems with the apex segmentation were avoided by having a very low resolution in this part of the ventricle. This was possible due to the way the ventricle contracted during the heart beat. The most prominent change from the end diastole to the end systole were the shortening along the long axis, while the apex remained to a great extent unchanged. This allowed the low resolution in the apex part. A relatively high resolution was applied toward the base, i.e. the outflow and inflow parts in Figure 2.6. The physiology of the contraction can be somewhat understood by inspecting Figure 2.1, where it is clear that the blood does not have to move all the way into the apex to get from the inflow tract to the outflow tract. The differences in resolution from the apex to the base of the ventricle has the disadvantage that it can attend a ”pear-shape” as can be seen in Figure 10.14.

Subdivision seems to be an easy and effective way of making geometrical models, automating the time-consuming problem of representing a smooth model. Some problem with perhaps too low volumes could possibly be improved by using another subdivision scheme which is interpolating instead of approximating. This can also be avoided with choosing another method for construction of the models, which does not place the control points on the contours, but rather fit the smooth surface to the contour. However the problem is not as substantial, as the model acts inflating when it is placed within the heart chamber. Before the segmentation the models where enlarged to be able to lock on to the endocardial border instead of to features within the chamber. The segmentation did from the smooth models have a tendency to round out sharply protruding edges, such as the narrow apex and the outflow tracts.

A challenge in developing the tracking of the right ventricle is to get good enough recordings. It can be stated that the recordings used in this project work was not good enough to yield very accurate results. Nevertheless, the purpose of the project was to investigate if it were possible to make an initial model, which could lead to good segmentations of the right ventricle. For the purpose of an initial model, the recordings were good enough.

Chapter 12

Conclusion

Segmentation of the right ventricle using model-based segmentation in real-time has been performed, and a simple model of the right ventricle has been developed. The segmentation was accomplished using the Real-Time Contour Tracking Library with a deformable subdivision model for the right ventricle.

The results of the tracking of the right ventricle are promising. The models adjust to the right ventricle and could to some extent account for missing right ventricular free wall and crescent cross-sectional shape. Quantitative validation of the volume calculations and the segmentations shows promising results. However, challenges still remains, especially in the systolic segmentation of the ventricle.

The right ventricle segmentation was connected with simultaneous left ventricle segmentation. And the volume curves were changing according to the theory of the change in volume in the two heart chambers during the heart cycle.

Segmentation of the right ventricle based on a model-based approach seems to yield feasible results and encourages further work about the subject.

Chapter 13

Further work

Further development of the right ventricular model could be performed. This includes a larger statistical foundation for the construction of the mean shape model, a more sophisticated way of incorporating the physiology during the systole which could avoid the ”‘pear shaped”’ form that sometimes occurred during the segmentation.

Improvement on the edge detection in the RCTL application could be an important and interesting improvement of the whole subject of segmentation of the right ventricle.

The segmentation of the right ventricle in RCTL is one step on the road to accomplishing the vision that in the future the application could be used for tracking all four heart chambers simultaneously. The promising results of the simultaneous left ventricle and right ventricle tracking encourage further research toward this goal.

Bibliography

[1] Bjørn Angelsen and Hans Torp. *Ultrasound Imaging - Waves, Signals and Signal processing in Medical Ultrasonics, Vol. I and Vol. II.* www.ultrasoundbook.com, 2000.

[2] Lynn S. Bickley. *Guide to Physical Examination and History Taking.* Lippincott Williams and Wilkins, 8th edition, 2003.

[3] Andrew Blake and Michael Isard. *Active Contours.* Springer Verlag, 1998.

[4] R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied Kalman filtering.* John Wiley & Sons, 3rd edition, 1997.

[5] Chunguang Ken Cao and Timothy S. Newman. A new framework for recovery of shape of the right ventricle from gbp spect images. In *ACM Southeast Regional Conference (2)*, 2005.

[6] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–356, 1994.

[7] F. P. Czegledy and J. Katz. A new geometric description of the right ventricle. *J. Biomed. Eng.*, 15(5), 1993.

[8] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, Inc., 2nd edition, 2001.

[9] Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation for Automatic Control.* Marine Cybernetics AS, 1st edition, 2002.

[10] Alejandro F. Frangi, Wiro J. Niessen, and Max A. Viergever. Three-dimensional modeling for functional analysis of cardiac images: A review. *IEEE Transactions on Medical Imaging*, 20(1), 2001.

[11] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing.* Prentice-Hall, Inc., 2nd edition, 2001.

[12] François Haddad, Sharon A. Hunt, David N. Rosenthal, and Daniel J. Murphy. Right ventricular function in cardiovascular disease, part i: Anatomy, physiology, aging and functional assessment of the right ventricle. *Circulation*, 117(11), 2008.

[13] François Haddad, Sharon A. Hunt, David N. Rosenthal, and Daniel J. Murphy. Right ventricular function in cardiovascular disease, part ii: Pathophysiology, clinical importance, and management of right ventricular failure. *Circulation*, 117(13), 2008.

[14] Donald Hearn and M. Pauline Baker. *Computer Graphics with Open GL.* Pearson Prentice Hall, 3rd edition, 2004.

[15] Rolf Henriksen. *Stokastiske systemer: analyse, estimering og regulering.* Institutt for teknisk kybernetikk, Fakultet for elektroteknikk og telekommunikasjon, Norges teknisk-naturvitenskapelige universitet, 1998.

[16] S. Y. Ho and P. Nihoyannopoulos. Anatomy, echocardiography, and normal right ventricular dimensions. *Heart*, 92(supplement 1), 2006.

[17] Leng Jiang, Samuel Siu, Mark D. Handschumacher, J. Luis Guererro, Jose Antonio Vazquez de Prada, Mary Etta Kin, Michael H. Picard, Arthur E. Weyman, and Robert A. Levine. Three-dimensional echocardiography. in vivo validation for right ventricular volume and function. *Circulation*, 89(5), 1994.

[18] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[19] Roberto M. Lang, Victor Mor-Avi, Lissa Sugeng, Petra S. Nieman, and David J. Sahn. Three-dimensional echocardiography - the benefits of the additional dimension. *Journal of the American College of Echocardiography*, 48(10), 2006.

[20] M. A. Lawson, G. G. Blackwell, N. D. Davis, M. Roney, L. J. Dell'Itelia, and G. M. Pohost. Accuracy of bi-plane long-axis left ventricular volume determined by cine magnetic resonance imaging in patients with regional and global dysfunction. *American Journal of Cardiology*, 77, 1996.

[21] Lennart Ljung. *System Identification. Theory for the user.* Prentice Hall PTR, 2nd edition, 1999.

[22] Steven Lobregt and Max A. Viergever. A discrete dynamic contour model. *IEEE Transactions On Medical Imaging*, 14(1), 1995.

[23] Lisa Lorentzen, Arne Hole, and Tom Lindstrøm. *Kalkulus med én og flere variabler.* Universitetsforlaget, 1st edition, 2003.

[24] Christine H. Lorenz, Eloisa S. Walker, Victoria L. Morgan, Stacy S. Klein, and Thomas P. Graham. Normal human right and left ventricular mass, systolic function and gender differences by cine magnetic resonance imaging. *Journal of Cardiovascular Magnetic Resonance*, 1, 1999.

[25] Michael Mansfield and Colm O'Sullivan. *Understanding Physics.* John Wiley and Sons Ltd, 1st edition, 1999.

[26] T. McInerney and D. Terzopoulos. *Handbook of Medical Imaging.* Academic Press, 2002.

[27] Steven C. Mitchell, Boudewijn P. F. Lelieveldt, Rob J. van der Geest, Hans G. Bosch, Johan H. C. Reiber, and Milan Sonka. Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac mr images. *IEEE transactions on medical imaging*, 20(5), 2001.

[28] Ricardo Munoz, Edward Marcus, Guillermo Palacio, Kimberlee Gauvreau, David L. Wessel, and Steven D. Colan. Reconstruction of 3-dimensional right ventricular shape and volume from 3 orthogonal planes. *Journal of the American Society of Echocardiography*, 13(5), 2000.

[29] J. Alison Noble and Djamal Boukerroui. Ultrasound image segmentation: A survey. *IEEE Transactions on Medical Imaging*, 25(8), 2006.

[30] Sosial og helsedepartementet. Kvinners helse i norge. *NOU*, 13, 1999.

[31] Jae K. Oh, James B. Seward, and A. Jamil Tajik. *The echo manual.* Lippincott Williams & Wilkins, 3rd edition, 2006.

[32] Fredrik Orderud. A framework for real-time left ventricular tracking in 3d+t echocardiography, using nonlinear deformable contours and kalman filter based tracking. *Computers in Cardiology*, 33:125–128, 2006.

[33] Fredrik Orderud, Jøger Hansgård, and Stein I. Rabben. Real-time tracking of the left ventricle in 3d echocardiography using a state estimation approach. In *MICCAI 2007*, pages 858–865. Springer-Verlag Berlin Heidelberg, 2007.

[34] Fredrik Orderud and Stein Inge Rabben. Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[35] Theo Pavlidis. Why progress in machine vision is so slow. *Pattern Recognition Letters*, 13(4), 1992.

[36] Ewald Quak. *Subdivision surfaces: a survey of basic concepts and algorithms.* Oslo: SINTEF, Applied Mathematics, 1997.

[37] M. S. Sacks, C. J. Chuong, G. H. Templeton, and R. Peshock. In vivo 3-d reconstruction and geometric characterization of the right ventricular free wall. *Annals of Biomedical Engineering*, 21(3), 1993.

[38] Rod R. Seeley, Trent D. Stephens, and Philip Tate. *Essentials of Anatomy & Physiology.* McGraw-Hill International Edition, 6th edition, 2007.

[39] F. G. Spinale, B. A. Carabello, and F. A. Crawford Jr. Right ventricular function and three-dimensional modeling using computer-aided design. *Journal of Applied Physiology*, 68(4), 1990.

[40] Lawrence H. Staib and James S. Duncan. Model-based deformable surface finding for medical images. *IEEE transactions on medical imaging*, 15(5), 1996.

[41] Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH '98: Proceedings of the 25th annual conference on computer graphics and interactive techniques*, 1998.

[42] Gilbert Strang. *Linear algebra and its applications.* Thomson Brooks/Cole, 4th edition, 2006.

[43] Thomas L. Szabo. *Diagnostic Ultrasound Imaging.* Elsevier Academic Press, 2004.

[44] Gábor Székely and Guido Gerig. Model-based segmentation of radiological images. *Künstliche Intelligenz*, 3:18–23, 2000.

[45] Paul A. Tipler and Gene Mosca. *Physics for scientists and engineers.* W.H. Freeman and Company, 5th edition, 2004.

[46] Hans Torp. Unpublished lecture notes in the course mfel1010, medicine for non-medicinists, 2006.

[47] Renan Uflacker. *Atlas of Vascular Anatomy an Angiographic Approach.* Lippincott Williams and Wilkins, 1st edition, 2007.

[48] P. M. J. van der Berg. *Extension of a finite element model of left ventricular mechanics with a right ventricle.* Eindhoven, 2002.

[49] Jianwen Wang, Kalpana Prakasea, Chandra Bomma, Harikrishna Tandri, Darshan Dalal, Cynthia James, Crystal Tichnell, Mary Corretti, David Bluemke, Hugh Calkins, and Theodore P. Abraham. Comparison of novel echocardiographic parameters of right ventricular function with ejection fraction by cardiac magnetic resonance. *Journal of the American Society of Echocardiography*, 20(9), 2007.

[50] J. Nelson Wright. Image formation in diagnostic ultrasound. In *IEEE International Ultrasonics Symposium*, 1997.

[51] P. Zamperoni. Plus ça va, moins ça va. *Pattern Recognition Letters*, 17(7), 1996.

[52] Denis Zorin and Peter Schröder. Subdivision for modeling and animation. In *SIGGRAPH2000 Course Notes*, 2000.

Part V

Appendices

Appendix A

MATLAB-script for area calculation of polygon and volume approximation of manually segmented contour

```
clear all
hold on

% The number of vertices in each polygon
L = 20;

% The frame and the maximum and minimum ranges manually segmented and
% stored in a .mat file called filename.
filename = '7A9BIB00';
frame = 12;
MAX_RANGE = 240;
MIN_RANGE = 60;

% Gets the polygon control points saved for the ventricle in a matrix with
% the control points for each disk saved after each other.
A = GetPolygons(filename,MIN_RANGE,MAX_RANGE,frame);

% Calculates the distance between the disks
h = (A(end,2)-A(1,2))/((MAX_RANGE-MIN_RANGE)/10)
```

```
i = 1; j = 1;
area = zeros(20,1);
volume = zeros(20,1);

% Traversing through the short-axis slices of the whole manually
% segmented ventricle
for z = A(1,2):h:A(end,2)
    for sigma = 1:L

        % Gets the control for each short axis slice
        Qx = A(i:i+19,1);
        Qy = A(i:i+19,3);

        % Plots the slice
        plot3(Qx,ones(1,20)*z,Qy)

        % Illustrates the disks by making patches
        points1 = [Qx (ones(1,20)*z)’  Qy];
        points2 = [Qx (ones(1,20)*z+h)’  Qy];
        points = [points1;points2];
        faces = [ 1  2 22 21;  2  3 23 22;  3  4 24 23;  4  5 25 24;
         5  6 26 25;  6  7 27 26;  7  8 28 27;  8  9 29 28;
         9 10 30 29; 10 11 31 30; 11 12 32 31; 12 13 33 32;
          13 14 34 33; 14 15 35 34; 15 16 36 35; 16 17 37 36;
          17 18 38 37; 18 19 39 38; 19 20 40 39; 20 1 21 40];
        patch(’Faces’,faces,’Vertices’,points,’FaceColor’,’red’)
        view([-40 50]);
        axis([-0.04 0.04 0.02 0.1 -0.05 0.05])
    end

    % Store the area for the current disk
    area(j,1) = PolygonArea(Qx,Qy);

    % Store the volume for the current disk
    volume(j,1) = area(j,1)*h;

    % Control point number
    i = i+20;

    % Disk number
    j = j+1;
```

```
end

% Summarize the volume and expresses it in ml.
total_volume = sum(volume);
Volume = total_volume*1e6


function area = PolygonArea(Qx,Qy)
% Calculates the area of a polygon with n vertices
% Made by Asbjørn Engås, May 2008
%
% area = PolygonArea(Qx,Qy)
%
% Takes in the vertices of a polygon and returns the area
%
% Qx and Qy are the x- and y-coordinates of the polygon
%
% Returning variable:
% Area = The area of the polygon

% Number of vertices in the polygon
n = size(Qx,1);

% Periodicity of a polygon
Qx = [Qx; Qx(1)];
Qy = [Qy; Qy(1)];

% The polygon formula
area = 0;
for k = 1:n
    area = area + (Qx(k)*Qy(k+1)-Qx(k+1)*Qy(k));
end
area = 0.5*abs(area);


function [points] = GetPolygons(filename,MIN_RANGE,MAX_RANGE,frame)
% Gets the vertices of all the contours of a manually segmented ventricles
% Made by Asbjørn Engås, May 2008
%
% filename is the filename of the recording which are manually segmented
% MIN_RANGE are the minimum range
% MAX_RANGE are the maximum range
% frame is the frame segmented
%
% Returns all the control points in three coordinates.

% Finds the file and gets the info from the corresponding h5-file.
filepath = 'C:\Ultralydopptak\2007 Brage RV\';
file = strcat(filepath,filename,'.h5');
[fGeom fData] = GetFrames_h5(file);
file2 = strcat('EndSystole\',filename,'ContourRange');

% Calculates the attributes of the recording in probespace.
RES = 128;
range_min = fGeom.depthstart;
range_max = fGeom.depthend;
az_min = (range_max-range_min)*sin(min(fGeom.az_angles));
az_max = (range_max-range_min)*sin(max(fGeom.az_angles));
el_min = (range_max-range_min)*sin(min(fGeom.el_angles));
el_max = (range_max-range_min)*sin(max(fGeom.el_angles));
az_inc = (az_max-az_min)/RES;
el_inc = (el_max-el_min)/RES;

% Traverse through the file and store all the stored points in the
% returning variable
points = [];
for b = MIN_RANGE:10:MAX_RANGE
    filename = strcat(file2,num2str(b),'Frame',num2str(frame));
    load(filename);
    points = [points;(punkt(:,1)*az_inc+az_min)
              (punkt(:,2)) (punkt(:,3)*el_inc+el_min)];
end
```

Appendix B

Node positions and face list for the mean right ventricle model

The columns are the x, y, and z coordinates for the i'th node in the i'th row, respectively.

```
nodes =

   -0.0044   -0.0378    0.0227
   -0.0229   -0.0378    0.0232
   -0.0199   -0.0378    0.0010
    0.0178    0.0067   -0.0019
    0.0125    0.0067    0.0266
   -0.0218    0.0067    0.0340
   -0.0140    0.0067    0.0059
   -0.0210    0.0067   -0.0228
    0.0026    0.0067   -0.0211
    0.0227    0.0167   -0.0039
    0.0177    0.0167    0.0247
   -0.0211    0.0167    0.0315
   -0.0146    0.0167    0.0058
   -0.0205    0.0167   -0.0276
    0.0051    0.0167   -0.0269
    0.0254    0.0271   -0.0081
    0.0206    0.0271    0.0247
   -0.0190    0.0271    0.0298
   -0.0147    0.0271    0.0064
   -0.0203    0.0271   -0.0306
    0.0085    0.0271   -0.0310
    0.0247    0.0378   -0.0073
    0.0198    0.0378    0.0213
   -0.0161    0.0378    0.0244
   -0.0141    0.0378    0.0038
   -0.0190    0.0378   -0.0300
    0.0079    0.0378   -0.0298
```

The face list specifies how the nodes are topologically connected together.

```
faces =        {[ 1  2  3]
               [ 1  4  5]
               [ 1  5  6  2]
               [ 2  6  7]
               [ 2  7  8  3]
               [ 3  8  9]
               [ 3  9  4  1]
               [ 4 10 11  5]
               [ 5 11 12  6]
               [ 6 12 13  7]
               [ 7 13 14  8]
               [ 8 14 15  9]
               [ 9 15 10  4]
               [10 16 17 11]
               [11 17 18 12]
               [12 18 19 13]
               [13 19 20 14]
               [14 20 21 15]
               [15 21 16 10]
               [16 22 23 17]
               [17 23 24 18]
               [18 24 25 19]
               [19 25 26 20]
               [20 26 27 21]
               [21 27 22 16]
               [22 27 26 25 24 23]};
```