

Real-time segmentation of 3D echocardiograms using
a state estimation approach with deformable models

Fredrik Orderud

Abstract

Cardiac ultrasound, or *echocardiography*, is considered to be the quickest and most cost effective imaging modality for assessment of cardiac function. The modality is unique in that it allows for real-time imaging of the heart, using portable equipment. The latest generation of echocardiography scanners are capable of acquiring dense image volumes in either real time, or over a few heart beats. Methods for analyzing these images, however, are lagging behind, with all existing methods for segmentation of 3D ultrasound data considered too computationally intensive to operate at acquisition rate. Availability of methods for analysis of 3D ultrasound in real-time could open up possibilities for very quick and simple measurement of cardiac function, potentially conducted during image acquisition. Results from automatic methods would also lessen the amount of inter- or intra-examiner differences introduced during the analysis, which in turn could lead to more reproducible results. The left ventricle of the heart has traditionally been the chamber of most interest within the field cardiac image analysis. This thesis has therefore focused on the left ventricle, and investigated ways to measure different aspects of the chamber, such as chamber volumes, myocardial strain and myocardial mass, by means of state estimation techniques.

The main goal of this thesis has been to explore the possibilities of using state estimation methods for segmentation and tracking of structures in volumetric data with deformable models. The ability of using non-iterative estimators, such as the Kalman filter, for fitting deformable models to image structures would radically reduce the computationally effort required for performing a 3D segmentation, and open up for real-time usage. An existing Kalman tracking framework have therefore been extended to operate in volumetric data. The framework has been successfully demonstrated to fit ellipsoids, spline surfaces, active-shape surfaces and subdivision surfaces to image data. Furthermore, theory for the simultaneous tracking of several models have been developed. Finally, methods for combining edge-detection measurements with speckle-tracking measurements have been shown, with the potential advantages of material tracking with the lack of drift in edge detection.

The framework has been demonstrated to successfully conduct high quality segmentation and tracking of the left ventricle, while operating in real-time without user intervention. Processing times are in the range of milliseconds per frame on standard computer hardware, which is orders of magnitude faster than the state-of-the-art methods. Applications of the framework on 3D echocardiograms has been investigated to prove the feasibility of the framework for automatic analysis of various aspects of cardiac function, such as view alignment, measurement of chamber volumes, myocardial muscle volume and regional myocardial strain. The results for view alignment are considered state of the art, and the results for volume measurements are close to the results of similar studies with semi-automatic tools.

Preface

This thesis is submitted as partial fulfillment of the degree *Philosophiæ Doctor* (PhD) in computer science at the Norwegian University of Science and Technology (NTNU). The presented work was carried out at the department of computer science (IDI) and the department of circulation and medical imaging (ISB) between August 2004 and March 2009.

Acknowledgments

First, I want to thank the department of computer science (IDI) and faculty of informatics, mathematics and electronics (IME) for financing my PhD through an open department grant with few strings attached. This grant gave me full freedom to pursue my ideas within image analysis over an extended period of time. This has indeed been a rare opportunity that I will always be grateful for.

A special thank goes to my main supervisor Stein I. Rabben for sharing his insight and giving valuable suggestions in discussions regarding image analysis approaches with deformable models, for all his thorough manuscript reviews, and for inviting me to the R&D team in GE Vingmed Ultrasound. Also, I want to thank my second supervisor Hans Torp for giving me a warm welcome to the ultrasound group despite not being a formal member, and always being available and helpful for questions and discussions on all issues related to ultrasound imaging and speckle-tracking.

My co-author Jøger Hansegård also deserves a special thank for always being helpful upon request, for conducting reference segmentations used for validation, for all his matlab scripts, and for manuscript reviews. Stian Langeland and Espen Remme also deserve a thanks for creating the ultrasound simulations of infarcted heart ventricles, as well as a thank to Lasse Løvstakken, for all his contributions on ultrasound simulation and spectral analysis. I also want to thank Olivier Gerard for many helpful discussions and suggestions over the last year.

On the clinical side, I want to thank Brage Amundsen for always being helpful with providing patient data for method evaluation, as well as Asbjørn Støylen, Charlotte Ingul and Anders Torstensen for interesting discussions on cardiac applications for my technology. I also want to thank all my colleagues during my stay at the department of circulation and medical imaging for welcoming me into this group, and making my stay at the group as interesting and enjoyable.

Finally, I want to thank all the employees at GE Vingmed Ultrasound in Trondheim for always being helpful when facing scanner-related problems.

Contents

I	Introduction	5
1	Background and motivation	7
2	Aims of study	9
3	Summary of presented work	11
4	Discussion	13
4.1	Theoretical contributions	14
4.2	Software implementation	15
4.3	Alternative approaches	16
4.4	Applications	16
4.5	Future extensions	18
4.6	Conclusions	20
5	Publication list	21
II	Background material	23
6	Cardiac function	24
6.1	Global function	24
6.2	Regional function	25
6.3	Remodeling	26
7	Medical ultrasound	28
7.1	Image formation	28
7.2	3D imaging challenges	31
7.3	Acquisition coordinate system	31

<i>CONTENTS</i>	4
8 State modeling and estimation	33
8.1 General state space models	33
8.2 Linear state space models	33
8.3 Sequential Bayesian estimation	34
8.4 Kalman filter	35
8.5 Extended Kalman filter	37
8.6 Information filter	38
8.7 Multiple independent measurements	39
8.8 Measurement likelihood	40
8.9 Iterated Kalman filter	40
8.10 Kalman smoother	41
9 Model-based image segmentation	43
9.1 Deformable models	43
9.2 Deformation schemes	46
9.3 Temporal properties	47
9.4 Fitting algorithm	47
9.5 Image measurements	49
9.6 The role of regularization	50
III Publications	60

Part I

Introduction

This thesis consists of three parts: First, an introductory part that contains motivation and aims of the thesis, together with discussions and conclusions to summarize and tie the papers together. Secondly, background theory on cardiac function, state estimation, as well as model-based image analysis, which together serves as background material and context for the papers to make the thesis self-contained. Finally, the publications that constitutes the individual contributions of the thesis.

Chapter 1

Background and motivation

In Europe, cardiovascular diseases is the cause of over 4.3 million deaths each year, or 48% of all deaths [1]. This makes cardiovascular diseases the main cause of death and most prominent threat to public health in Europe, even when compared to cancer. The widespread occurrence of cardiovascular diseases makes rapid and reproducible methods for evaluation of cardiac function an important objective. Cardiac ultrasound, or *echocardiography*, is considered to be the quickest and most cost effective imaging modality for assessment of cardiovascular diseases. The modality is unique in that it allows for real-time imaging of the heart, using portable equipment.

Over the last few years, cardiac ultrasound has moved from 2D to 3D, and the latest generation of echocardiography scanners are capable of acquiring dense image volumes in either real time, or over a few heart beats. Methods for analyzing these images, however, are lagging behind. At the beginning of my PhD period, all known existing methods for segmentation of 3D ultrasound data were considered too computationally intensive to operate at acquisition rate. Parts of the reason for this gap between imaging data rate and processing speed is related to the fact that most approaches for cardiac image analysis are based on deformable geometric models that are fitted to the image using iterative *optimization* methods, that gradually converge to a good fit. While such methods are known to produce good results, they are also computationally intensive, due to the need for many refinement iterations for each image frame.

An alternative, although less commonly pursued approach to optimization methods is to use *state estimation* methods [2]. With state estimation, the fitting problem is modeled as the estimation of a limited set of parameters related to the geometric model being fitted. It is then possible to use a class of computationally efficient algorithms that can compute an optimal fitting of the model to the image data in a least squares sense in a single iteration [3, 4, 5]. At the start of my PhD, prior art only existed on the application of state estimation methods for fitting deformable models to 2D image data, whereas the application within volumetric 3D data appeared unexplored.

Availability of methods for analysis of 3D ultrasound in real-time could open up possibilities for very quick and simple measurement of cardiac function, potentially conducted during image acquisition. Furthermore, any real-time method must be

robust enough to operate with little or no user interaction, since the examiner will be busy scanning the patient. Results from automatic methods would also lessen the amount of inter- or intra-examiner differences introduced during the analysis, which in turn could lead to more reproducible results.

Chapter 2

Aims of study

The main goal of this thesis has been to explore the possibilities of using state estimation methods for segmentation and tracking of structures in volumetric data with deformable models. The ability of using non-iterative estimators, such as the Kalman filter, for fitting deformable models to image structures would radically reduce the computational effort required for performing a 3D segmentation, and open up for real-time usage. Ultrasound is currently the only medical imaging modality capable of acquiring dense image volumes in real-time, so 3D echocardiography has served as the main modality of interest. The left ventricle of the heart has traditionally been the chamber of most interest within the field cardiac image analysis. This thesis has therefore focused on the left ventricle, and investigated ways to measure different aspects of the chamber, such as chamber volumes, myocardial strain and myocardial mass, by means of state estimation.

This goal is motivated by a more long-term vision of automatic cardiac diagnosis in ultrasound recordings. Successful tracking of high quality will hopefully enable fully automatic extraction of clinical parameters from the recordings, which can then be used to detect a range of different cardiac diseases. The development of real-time methods places additional constraints on the performance, compared to offline methods. There is little time available for user interaction when operating during image acquisition, so effort must be made to ensure a high degree of robustness, in addition to efficiency and accuracy.

Application of state estimation algorithms for real-time sequential frame-by-frame segmentation of echocardiography constitutes a novel approach, with only a handful known prior publications [6, 7, 8, 9]. Prior art is limited to 2D echocardiography, with 3D extension only listed as an “*interesting challenge*” in discussion of future work [8, 9]. Furthermore, no thorough theory for the connection to deformable models, with separation of local shape parameters from global pose parameters, could be found in the literature.

To summarize, the primary goals of this thesis have been to:

- Extend the video tracking framework of Blake & al. [3] to volumetric data, and

generalize the framework to support a range of different deformable models and image measurement sources.

- Demonstrate that the framework is capable of conducting high quality 3D segmentation and tracking of the left ventricle in real-time, with the robustness required for fully automatic behavior.
- Use the framework to measure fully automatic and in real-time different aspects of left ventricular function, such as chamber volumes, myocardial strain, as well as myocardial mass.

Emphasis has been placed on ensuring that the developed methods and algorithms are general, so they can be applicable to most imaging modalities capable of generating 2D or volumetric images. 3D echocardiography has still remained the primary application, since this is the only modality capable of acquiring dense image volumes in real time.

To demonstrate the applicability of the developed tracking framework, it has been prioritized to publish papers that cover a range of different extensions and applications to the framework. Focus have primarily been on technical feasibility, which means that the experiments has been conducted rigorously, but not excessively. More thorough validation of clinical usefulness has been left for future studies.

Chapter 3

Summary of presented work

List of presented works, together with their respective contribution:

A *A Framework for Real-Time Left Ventricular Tracking in 3D+T Echocardiography, Using Nonlinear Deformable Contours and Kalman Filter Based Tracking*, IEEE Computers in Cardiology 2006:

This paper extends the Kalman-filter based video tracking framework of Blake et al. [3, 5] to volumetric 3D data, using a truncated ellipsoid as crude model of the left ventricle.

B *Real-time Tracking of the Left Ventricle in 3D Echocardiography Using a State Estimation Approach*, Medical Image Computing and Computer Assisted Intervention 2007:

This paper extends the tracking framework to support deformable spline surfaces. It also introduces methodology for clear separation of global pose parameters from local shape parameters in the processing chain.

C *Real-Time Active Shape Models for Segmentation of 3D Cardiac Ultrasound*, Conference on Computer Analysis of Images and Patterns 2007:

This paper extends the tracking framework to support statistical shape models [10], where shape deformations are restricted to a linear subspace of the full deformation space to achieve physiologically realistic shapes with a compact parametrization.

D *Real-time 3D Segmentation of the Left Ventricle Using Deformable Subdivision Surfaces*, IEEE Conference on Computer Vision and Pattern Recognition 2008:

This paper extends the tracking framework to support deformable subdivision surfaces [11], which generalize spline surfaces to arbitrary topology. Also, this is the first known publication of segmentation with subdivision surfaces directly evaluated without recursive refinement.

E *Real-time Left Ventricular Speckle-Tracking in 3D Echocardiography With Deformable Subdivision Surfaces*, MICCAI workshop on Analysis of Medical Images 2008:

This paper extends the tracking framework with 3D block matching instead of edge-detection to update both the shape and parametric density of models with the objective of tracking tissue deformation and detect regional strain differences.

F *Combining Edge Detection With Speckle-Tracking for Cardiac Strain Assessment in 3D echocardiography*, IEEE Ultrasonics symposium 2008:

This paper extends the tracking framework to combine edge-detection with block-matching measurements to update the models in an integrated manner that compensates for the inherent drift problem associated with sequential block-matching.

G *Automatic coupled segmentation of endo- and epicardial borders in 3D echocardiography*, IEEE Ultrasonics symposium 2008:

This paper provides an extension of tracking framework to simultaneous tracking of multiple models, arranged in a hierarchy of transformations. The concept is applied to simultaneous tracking of the endocardium and the epicardium of the left ventricle, using a coupled thick-wall model.

H *Automatic Alignment of Standard Views in 3D Echocardiograms Using Real-time Tracking*, SPIE Medical Imaging conference 2009:

This paper describes an application of tracking framework for extraction of standardized views of the heart, based on landmarks from connected models corresponding to specific cardiac structures.

Chapter 4

Discussion

The Kalman filter [12] was first developed by R. E. Kalman in 1960 as a state-space solution to the minimal mean-square filtering problem. The filter gives a non-iterative least squares linear algebra solution to the discrete filtering problem, that allows for efficient computer implementation. Engineers quickly discovered the relevance of the filter as a practical solution to various estimation problems within navigation. Gradually, navigation computers used for everything from airplanes, missiles and spacecrafts, to handheld GPS devices and RADAR tracking systems were equipped with Kalman filters to estimate position and velocity [2]. Later, engineers outside the field of navigation discovered that the filters unique ability to combine a kinematic model with incomplete measurements from different sources also made it well-suited for other applications, such as robotic navigation and video tracking [3].

At the beginning of my PhD in 2004, a special issue of Proceedings of the IEEE was published on the topic “*Sequential state estimation: From Kalman filters to particle filters*” [13] that focused on the limitations of Kalman filters, like the assumption of Gaussian distributions that are inherently unable of handling multimodal distributions, divergent behavior when facing strong non-linearities and so on. By reading this issue, one gets the impression that Kalman filters are more or less abandoned in favor of more computationally intensive methods that are either based on stochastic Monte-carlo particle filter algorithms [14] or more the deterministic multi-sample unscented or sigma-point filtering approaches [15].

Still, I felt that the clear computational advantages of using a Kalman filter, compared to both iterative and multi-sample approaches, could outweigh the limitations in certain applications. The emergence of real-time 3D imaging in ultrasound had opened up a gap between the imaging rate and processing performance of current segmentation algorithms, which were unable to analyze the acquired images in real-time. I therefore started extending the Kalman-tracking framework of Blake et al. [3, 5] to handle 3D data, as well as generalizing it to handle several types of deformable models and image inputs. Nobody had much faith in this approach at first, but my colleagues and later industry partners gradually started taking an interest when I managed to generate some crude results that proved the feasibility of the approach.

4.1 Theoretical contributions

First and most importantly, the contour tracking framework of Blake et al. [3, 5] have been extended to operate in 3D data with deformable surface models [16]. This includes generalizing the formulas for mapping normal displacement measurements against model parameters with a Kalman filter in a manner that is invariant to the dimensionality of the dataset, so that the same algorithm can be used for tracking in both 2D and 3D image data. Also, it includes decoupling of the parametrization of the model from the overall framework [17], so that different types of deformable models can be plugged into the same tracking framework, without any changes to the way the measurement and filtering process operate.

To demonstrate the applicability and flexibility of the framework, theory on how to support a range of different models, including ellipsoid surfaces [16], spline surfaces [17], subdivision surfaces [18] and statistical shape models [19] have been developed. Subdivision surfaces are a class of surface types that combines the inherent smoothness and compact parametrization of spline surfaces with the unconstrained topology and ease of modeling of flat polygon surfaces. The subdivision paper is also believed to be the first application of direct non-recursive evaluation of subdivision surfaces using Stam's method [20] within image segmentation.

In addition, a clear separation between global pose transformation and local shape deformations of the models has been developed in a manner that supports a wide range of possible parametrization. Furthermore, the framework have been extended to support a full hierarchy of coupled deformable models [21]. This provides a simple and intuitive way of combining different models by defining transforms that map the coordinate system between different models. Following this approach, it is simple to constrain models to move relative to each other, in a manner similar to scene-graphs in computer graphics [22].

Finally, support for direct tracking of material points by means of speckle-tracking measurements has been incorporated. Such measurements generate full 3D displacement measurements, and are therefore conceptually different from normal displacement measurements. A clear advantage of tracking material points compared to edge-detection, is that the measurements can be used to update the parametric density of the model, and not only to adjust the shape. The ability of the framework to detect regional differences in strain in a simulation of an infarcted heart ventricle was demonstrated in [23], by using 3D speckle-tracking to update a subdivision surface. Furthermore, in [24] the approach was extended to also incorporate edge-detection measurement, by means of a two-step update process in the Kalman filter that combines edge-detection for efficient shape fitting with speckle-tracking for tracking of the residual deformations. This combination compensates for the inherent problem of drift associated with sequential tracking of material points from frame to frame, and allows for both faster and more accurate tracking with smaller search windows.

4.2 Software implementation

During the initial phase of my PhD period, a search for existing segmentation software to use as basis for my work was conducted. Unfortunately, little or no software for computationally efficient segmentation with support for non-cartesian data were discovered. I therefore decided to write my own library, named the *Real-time Contour Tracking Library* (RCTL).

This library has been developed continuously throughout my PhD, and has been used for all experiments to test various aspects of the tracking framework. The library has not only been designed to be computationally efficient, but also modular and extensible, so that it is suitable for a range of different applications. Specifically, emphasis has been placed on the following aspects in the design of the library:

- Independence of imaging geometry and modality:
 - Direct support image data in cartesian and polar coordinates, as well as sparse slice-based data, without initial conversion to cartesian coordinates.
 - Support for ultrasound data in 2D, multi-plane and volumetric configurations. Support for cardiac MR in sparse slice configurations consisting of both longitudinal and short-axis slices. Support for dense CT volume data.
- Modular design:
 - Support for a range of different edge detection criterias, feature tracking criterias, different types of deformable models, as well as transforms for model orientation.
 - Support for several estimation algorithms, including the extended Kalman filter, iterated Kalman filter, and distributed Kalman filtering approaches.
 - Support for simultaneous tracking of multiple coupled models, arranged in a hierarchical "scene graph"-like structure.
- Computationally efficiency:
 - Efficient vectorized and parallelized C++ code, with minimum sacrifice of readability.
 - Segmentations that consumes approximately 5-10 ms processing time per model per frame on a standard dual-core processor.

Originally, the "RCTL" library was intended to be a publicly available open-source project, and it was registered on the SourceForge project hosting webpage for some time. Later, when I started co-operating closely with GE, the library had to be withdrawn from SourceForge. This had some obvious disadvantages for the availability of the software to the research community. But still, I believe that my research will have an impact on cardiac image analysis in the time to come.

4.3 Alternative approaches

The recent emergence of general-purpose programmable graphics processing units (GPU) and widespread adoption of multi-core processors has led to a lot of focus on adopting existing algorithms to exploit the computational power and parallelism offered by these chips. GPU's features computational powers that far surpasses what is available in current processors. An approach for real-time segmentation could therefore be to accelerate current optimization algorithms by adapting them to run on a GPU, and solving the real-time segmentation problem brute force [25].

Many recently published methods are based on *machine learning* approaches, where the problem is solved by iteratively exploring a state space with a classifier that has been automatically trained, based on a collection of annotated examples. This reportedly leads to good results [26, 27], but the resulting classifier is more like a black-box that is not parametrized by physical units. Tuning or extension of the classifiers might therefore not necessarily be very intuitive, as compared to the more analytic approaches pursued in this thesis.

More recently, iterative refinement methods have been reported to achieve real time processing speeds, with a processing time of 33 ms per frame in 3D echocardiograms [28]. Still, this is approximately 10 times longer than the methods presented in this thesis, and the reported average mesh distance to a reference segmentations was 4.0 mm, whereas this thesis reports average errors of 2.2 mm [18].

An attractive property of Kalman filter-based estimators is its ability to be parametrized by physical units and spatial uncertainty. Measurement noise values, which are used to weight the influence of measurements, are specified as a variance in squared *meters*, which is a physical unit that has an intuitive meaning. This means that measurement noise values can be determined from image statistics, or by comparing edge detection criterias against a manual tracing and computing the standard deviation of the error. Similarly, the trace of covariance matrix for the state estimate gives a measure of the variance of each estimated parameter. These uncertainties can furthermore be propagated back to the deformable model, to compute the spatial uncertainty of each point on the surface of the deformable model. An example application of this could be to color-code surface regions of high uncertainty, that might e.g. be caused by low edge evidence.

4.4 Applications

The proposed tracking framework has been designed to be general enough to enable a wide range of configurations and support several applications. Within 3D echocardiography, the most obvious applications include:

Automatic view alignment

Orientation and navigation in volumetric images can be a time-consuming task that has traditionally been conducted manually as an initialization step for analysis. With

fully automatic segmentation, it is possible to automate this step, and generate standardized image views based on landmarks from the fitted models, as was presented in [29]. This includes both intersection slices through the volume, as well as cropping and orientation of volume renderings. This application has the advantage that it does not rely on 100% accurate segmentation, as long as the models are positioned and oriented approximately correctly. Real-time segmentation also enables dynamic views that are updated continuously to follow tissue movement. The reported agreements to manual alignment for the state-estimation approach is also favorable when comparing to prior art based on iterative optimization [30] and machine learning [31], with similar or better agreement for most metrics of long-axis and circumferential rotation error.

Chamber volume measurements

The heart's main function is to pump blood and maintain circulation of oxygenated blood throughout the body. Analysis of the pumping function is therefore of utmost importance. Blood volume measurements can be calculated by computing the volume of the surface models fitted to the endocardial boundary. These volumes can be traced over time, used for assessment of global heart function, and to derive clinically important parameters, such as end-systolic and end-diastolic volume, stroke volume, cardiac output and ejection fraction. Simultaneous volume measurement in several chambers can also be used for timing assessment, and detection of dyssynchrony. Several papers on left ventricular volume measurements based on different classes of deformable models are included in this thesis [16, 17, 18, 19]. Usage of subdivision surfaces reported the best agreement, with an agreement of 3.6 ± 21.4 ml for EDV, 9.0 ± 17.4 ml for ESV and -5.9 ± 11.1 % for EF. These results are almost as good as results from similar studies with semi-automatic tools [32].

Myocardial mass measurement

Enlarged myocardial muscle mass is closely related to some diseases such as hypertrophy [33]. Measurement of myocardial mass requires detection of both endocardium and epicardium simultaneously, which can be difficult since parts of the epicardium often have poor visibility. One can, however, couple the two models and utilize volume conservation to constrain the segmentation and enable measurement of myocardial mass, based on fitting the surfaces in a sequence of frames. Multiplication of the myocardial volume with the density will then give a measure of myocardial mass. This concept was introduced in [21], which reported an average underestimation of myocardial volume of 10.7 ml (8.0%) with all 5 cases within the 68% confidence interval of the reference segmentations. The reported myocardial volume measurements are furthermore within the interval of previous studies conducted with semi-automatic tools [34, 35].

Myocardial strain measurement

Methods for analysis of global heart function are excellent for assessing if something is wrong with the heart. More localized pin-pointing of the source of reduced function does, however, require methods that are capable of analyzing function on a regional scale. One such example is the detection of infarcted areas within myocardium [36]. One can track material points within the myocardium by means of speckle-tracking techniques, and use the displacement vectors to update a model which captures both shape and regional contraction. The parametric density of the model can then be used to infer regional strain differences throughout the heart cycle. This combination of the Kalman tracking framework with speckle-tracking was introduced in [23, 24], and good qualitative agreement for regional strain was reported against ultrasound simulations of an infarcted heart. Strain measurements in in-vivo data did, however, turn out to be a more challenging task than originally anticipated, since 3D echo has significantly lower spatiotemporal resolution than 2D echo. Limited faith in the feasibility of getting results of any clinical value therefore led to no results from in-vivo data being published. Later validations of a commercial 3D strain tool have also reported very low correlation for segmental strain values, when compared to strain from 2D echo [37].

4.5 Future extensions

The methods presented in this thesis have potential for further development. Several possibilities are open for extending the framework into both new processing modes, as well as new applications still remains. Some of the most interesting areas of improvement include:

Edge detection

The endocardial edge is covered in a trabeculated structure that is partially filled with blood. This trabeculation is filled with blood during the diastole, whereas blood is squeezed out during systole. To most correctly measure blood volumes, it is therefore common convention to include the trabeculated area in end-diastole, and exclude it from the volume in end-systole [38]. The current edge detection criteria has problems adapting to this convention, and tends to consistently detect the outermost edge of the trabeculation at both end-diastole and end-systole, which leads to an underestimation of the ejection fraction. This problem can hopefully be improved by the usage of more advanced, possibly cardiac phase-dependent edge criterias that compensate for this effect. Utilization of different edge criterias for each cardiac segment might also have potential to further improve the edge detection accuracy.

Distributed and multi-model tracking

Simultaneous endo- and epicardial detection should only be considered as a starting point. The ability to combine several models in a tracking hierarchy can also poten-

tially be used for simultaneous detection of all four cardiac chambers in a manner where the models assist the search range of each one another. This could be useful for detecting dilated chambers, as well as dyssynchrony in the contraction pattern between the chambers. Multi-model tracking also yields an interesting problem of distribute filtering, since the models are independent with the exception of the shared transform nodes in the tracking hierarchy. The linear algebra calculations in the Kalman filter scales rather poorly with the number of parameters in the state vector, since matrix multiplications requires a cubic number of calculations relative to the state vector. Any decoupling of the partially independent tracking problem into smaller sub-problems could therefore help improve the computational efficiency when dealing with larger multi-model setups.

Strain measurement

Tracking of material points within the myocardium for strain measurement in 3D echocardiography have received a lot of interest the past few years, since it would open up for rapid and comprehensive evaluation of regional heart function from a single dataset. Several approaches for material tracking in 3D echocardiography, based on block-matching, optical flow and elastic registration, have recently been published, and most approaches report promising results from simulations. A commercial tool for 3D strain assessment have also been released recently [39]. The current generation of 3D ultrasound scanners do, however, place limitations on the spatio-temporal image quality obtainable that makes reliable tracking a difficult problem. Further investigation and improvements will therefore be required in order to make block-matching within the proposes tracking framework successful in in-vivo data.

Bidirectional and smoothing estimators

Another opportunity for improvements would be to track bidirectionally, meaning both forwards and backwards in time simultaneously, at the expense of not being capable of operating in real-time anymore. This would allow for more accurate segmentation and tracking by incorporating image information from both previous and future images when fitting the models in each frame [40]. The inherent problem of drift faced in sequential block-matching would also be alleviated with bidirectional tracking, since periodicity could be more strictly enforced.

Other imaging modalities

There is nothing with the tracking framework that limits its applicability to ultrasound. Other imaging modalities, such as nuclear Magnetic Resonance Imaging (MRI) and Computed x-ray Tomography (CT) generate beautiful images of the heart with more uniform intensity levels and less noise compared to ultrasound. The problem of model fitting with edge-detection in these modalities should therefore be simpler, compared to ultrasound. Still, the temporal resolution of the modalities are significantly lower, and acquisition time much longer than for ultrasound. Algorithms capable

of operating in real-time might therefore not be that important. Still, the inherent robustness of the framework, the long convergence radius, as well as the applicability to a range of different types of deformable models might be attractive properties also in other modalities.

4.6 Conclusions

I have successfully extended the video tracking framework of Blake & al. [3] to volumetric data, and demonstrated the use of the framework to fit ellipsoids, spline surfaces, active-shape surfaces and subdivision surfaces to echocardiographic data. Furthermore, I have developed theories for the simultaneous tracking of several models. Finally, I have shown how to combine edge-detection measurements with speckle-tracking measurements to update the models in a manner that combines the advantages of material tracking with the lack of drift in edge detection.

The framework has been demonstrated to successfully conduct high quality segmentation and tracking of the left ventricle, while operating in real-time without user intervention. Processing times are in the range of milliseconds per frame on standard computer hardware, which is orders of magnitude faster than the state-of-the-art methods.

Applications of the framework on 3D echocardiograms has been investigated to prove the feasibility of the framework for automatic analysis of various aspects of cardiac function, such as view alignment, measurement of chamber volumes, myocardial muscle volume and regional myocardial strain. The results for view alignment are considered state of the art, and the results for volume measurements are close to the results of similar studies with semi-automatic tools.

Chapter 5

Publication list

The following is a list of all research papers and patent applications published during my PhD where I have contributed as a main or co-author. Some of them fall outside the scope of my PhD, and are therefore not included as part of this thesis.

Peer-reviewed conference proceeding papers

1. F. Orderud, G. Kiss, S. Langeland, E. Remme, H. Torp, S.I. Rabben: *Real-time Left Ventricular Speckle-Tracking in 3D Echocardiography With Deformable Subdivision Surfaces*, in proceedings of the MICCAI workshop on Analysis of Medical Images - f-MICCAI 2008, p. 41-48.
2. F. Orderud, S.I. Rabben: *Real-time 3D Segmentation of the Left Ventricle Using Deformable Subdivision Surfaces*, in proceedings in the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR 2008, p. 1-8.
3. F. Orderud, J. Hansegård, S.I. Rabben: *Real-time Tracking of the Left Ventricle in 3D Echocardiography Using a State Estimation Approach*, in proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention - MICCAI 2007, p. 858-865.
4. J. Hansegård, F. Orderud, S.I. Rabben: *Real-Time Active Shape Models for Segmentation of 3D Cardiac Ultrasound*, in proceedings of the 12th International Conference on Computer Analysis of Images and Patterns - CAIP 2007, p. 157-164.
5. F. Orderud: *Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements*, in proceedings of Scandinavian Conference on Simulation and Modeling - SIMS 2005.

Other conference proceeding papers

1. F. Orderud, H. Torp, S.I. Rabben: *Automatic Alignment of Standard Views in 3D Echocardiograms Using Real-time Tracking*, in proceedings of the SPIE Medical Imaging conference 2009.
2. F. Orderud, G. Kiss, S. Langeland, E. Remme, H. Torp, S.I. Rabben: *Combining Edge Detection With Speckle-Tracking for Cardiac Strain Assessment in 3D echocardiography*, in proceedings of the IEEE Ultrasonics symposium - IUS 2008.
3. F. Orderud, G. Kiss, H. Torp: *Automatic coupled segmentation of endo- and epicardial borders in 3D echocardiography*, in proceedings of the IEEE Ultrasonics symposium - IUS 2008.
4. L. Løvstakken and F. Orderud, and H. Torp: *Real-time Indication of Acoustic Window for Phased-array Transducers in Ultrasound Imaging*, in proceedings of IEEE Ultrasonics symposium - IUS 2007, p. 1549-1552.
5. F. Orderud: *A Framework for Real-Time Left Ventricular Tracking in 3D+T Echocardiography, Using Nonlinear Deformable Contours and Kalman Filter Based Tracking*, in proceedings of Computers in Cardiology - CinC 2006, p. 125-128.

Abstracts

1. F. Orderud, J. Hansegård, B. Amundsen, H. Torp, S.I. Rabben: *Real-time Left Ventricular Volume Measurements in 3D Echocardiograms*, presented abstract on EuroEcho 2007.

Patent applications

1. A. Heimdal, S. Langeland, F. Orderud: *System and method for displaying ultrasound motion tracking information*, submitted USPTO application 2009.
2. F. Orderud, V. Lundberg, H. Torp, S.I. Rabben: *Method and apparatus for automatically identifying image views in a 3D dataset*, submitted USPTO application 2009.
3. F. Orderud, J. Hansegård, S.I. Rabben: *Methods for using deformable models for tracking structures in volumetric data*, submitted USPTO application 2008.
4. F. Orderud: *Method for real-time tracking of cardiac structures in 3D echocardiography*, USPTO application 2008/0069436.
5. H. Torp, F. Orderud, L. Løvstakken: *Method and system for determining contact along a surface of an ultrasound probe*, USPTO application 2007/0010742.

Part II

Background material

Chapter 6

Cardiac function

The heart is a small muscular organ, whose main purpose is to pump blood through the body. It consists of four chambers, the *right atrium*, the *right ventricle*, the *left atrium* and the *left ventricle*. The atriums on each side of the heart is connected to their adjacent ventricles through valves, which opens for blood-stream in only one direction. Blood circulation is divided between the two sides of the heart: Deoxygenated blood from the body is pumped through the right side of the heart, towards the lungs, and fresh oxygenated blood from the lungs is pumped through the left side of the heart, towards the rest of the body. Figure 6.1 shows an overview of the anatomy of the cardiac chambers, and the direction of the blood flow throughout the heart.

Blood is pumped through the heart in rhythmic strokes, divided into two stages. The first stage, *systole*, happens as the ventricles contract, leading to a decrease in chamber volume, which in turn pumps blood out of the heart. This is followed by the *diastole*, where the ventricles relax to expand, while at the same time the atriums contract, pumping new blood into the heart.

6.1 Global function

Examinations of heart function usually starts with an evaluation of overall heart function, refereed to as global function. Such evaluation focuses on the size and pumping ability of the heart. For the left ventricle, two commonly used clinical parameters often used for diagnosis are stroke volume and ejection fraction. *Stroke volume* (SV) is the difference between the *end diastolic volume* (EDV) and the *end systolic volume* (ESV), and measures how much blood the heart pumps in each stroke. *Ejection fraction* (EF) is the ratio between the stroke volume and the end diastolic volume, and measures the percentage of the ventricle being emptied in each stroke:

$$SV = EDV - ESV$$

$$EF = \frac{SV}{EDV}$$

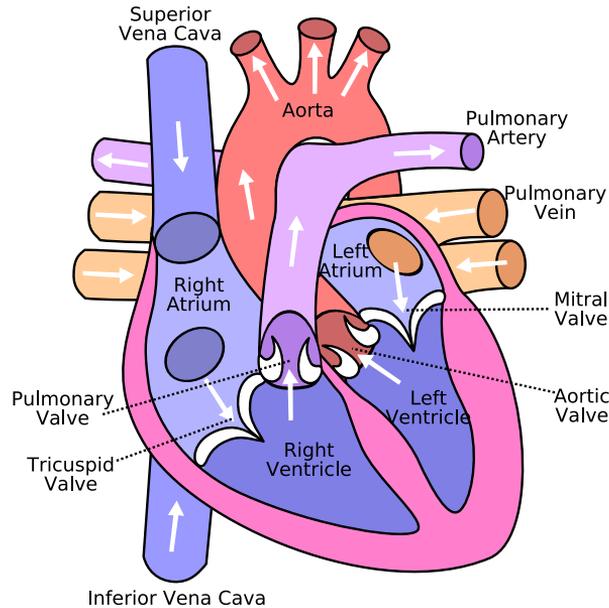


Figure 6.1: *Longitudinal* view of the heart that shows all chambers and valves. The tip of the heart at the bottom of the figure is referred to as the *apex*, whereas the the area around the valve are referred to as *base*. Figure courtesy of Eric Piercing.

Ejection fraction is generally considered as a good indicators of global heart function, since many cardiac diseases lead to hearts that does not contract as efficiently [41].

6.2 Regional function

To get more localized information about cardiac function, one can also analyze the contraction pattern locally across the heart muscle, which is referred to as *myocardium*. Historically, this has often been done through *wall motion* assessment, where experienced cardiologists look at ultrasound images of the myocardium while it contracts. Myocardial contraction leads to a combination of wall-thickening and radial displacement of the myocardium towards the blood pool, which can be assessed visually, and used to score the degree of contraction in different areas across the heart wall. This approach is, however, highly qualitative, and the results will depend on the training of the operator conducting the examination.

Over the last years, quantitative methods for evaluation of regional function have appeared. These methods estimate the deformation and contraction across the heart, based on following the speckle pattern inside the myocardium, or by measuring the velocity along the ultrasound beams. Local myocardial contraction is measured in *strain* (ϵ) [36, 42], which is a measure of fractional contraction of local myocardial dimensions (l) relative to associated dimensions at rest (l_0):

$$\epsilon = \frac{l - l_0}{l_0} .$$

The most common class of cardiac diseases are the ones associated with blockage of the coronary arteries that supply the myocardium with blood, such as ischemia and myocardial infarction. These diseases manifest themselves through reduced or ceased contractility in the affected regions of the myocardium. Regional assessment of myocardial strain can be used to identify regions that are affected by the disease and to which extent, whereas ejection fraction alone will only indicate that something is wrong with the contraction pattern, but not provide any localization information.

The heart has historically been imaged in 2D along the *longitudinal* apex-base direction, and the *circumferential* short-axis direction. This has led to a common convention of decomposing cardiac strain into longitudinal, circumferential and radial components, due to the limitations of the imaging techniques. With 3D imaging, we are no longer restricted to evaluating 2D projections of the strain, but can more directly measure all 6 component of the the full 3D strain tensor [33]. This tensor can be difficult to interpret directly, but by projecting the measured strain onto a surface model it is possible to compute the amount of contraction locally that contributes to the cardiac contraction, namely the *area strain* [43].

Most existing methods for evaluating regional function in 2D images suffer from two limitations: First, the images have to be properly aligned to either the longitudinal or circumferential axis of the heart, so that measurements can be related to the anatomic areas of the heart. This can be resolved by either carefully guiding the acquisition, or by extracting aligned views from an acquired 3D image volume of the heart. Secondly, the contraction of the heart causes out-of-plane movement of the myocardium through the image slice, so that different parts of the heart muscle is imaged at different phases in the cardiac cycle. This is especially a problem in the basal part of the heart, which moves considerably during the cardiac cycle.

6.3 Remodeling

Cardiac remodeling refers to changes in size, shape and muscle mass of the heart, caused by cardiac injury, altered blood pressure, or other factors [44]. Remodeling can affect the cardiac chamber differently, depending on disease type. Increased diastolic filling pressure in the left ventricle will e.g. affect the size of the left atria more than the ventricular size [45]. Remodeling can also be a temporal phenomenon, such as in cases of acute pressure increases in the right side of the heart [46]. The septum muscle between the ventricles will then shift sideways towards the left ventricle, causing a simultaneous dilation of the right ventricle and reduction of the left ventricular size.

Dyssynchrony in the contraction of the heart, caused by failures in the electrical activation of the heart muscle, will also lead to remodeling over time [47]. The shape of the ventricles will then change to become more spherical, and the amount of myocardial muscle mass will increase to compensate for the reduced pumping efficiency. In

such cases, it can be useful to study the shape of the ventricles, by means of sphericity analysis [48], and to measure the thickness of the myocardium to derive a measure of the total amount of myocardium muscle mass.

Chapter 7

Medical ultrasound

Medical ultrasound is a diagnostic imaging modality used to visualize the interior of the body. It is based on transmitting focused, high frequency ultrasound pulses from a transducer through the body, and then processing the backscatter echo received by the transducer. The pulse frequencies used are typically in the range of 1.5 - 10MHz, depending on the depth of the organ being imaged. Different organs, blood and tissue exhibit different acoustical properties, which in turn yields different echo responses that can be used to create images. When ultrasound is used to image the heart it is referred to as *echocardiography*. Imaging is achieved by transmitting and receiving ultrasound beams steered in different directions, so that they together form a plane or image volume. The resulting volume can then be visualized on a display. Unlike other medical imaging modalities like Magnetic Resonance Imaging (MRI) and X-ray Computed Tomography (CT), there are no radiation or strong magnetic fields involved, only sound waves. Ultrasound is therefore considered to be the safest and least harmful imaging modality used for medical diagnosis [49]. For this reason, ultrasound is the only modality routinely used for fetal imaging.

7.1 Image formation

Cardiac ultrasound images are commonly acquired using *phased array* ultrasound transducers, which are hand-held probes consisting of a 2D matrix or 1D array of piezoelectric elements that convert electric energy to acoustic pressure waves during transmission, and acoustic pressure back to electric signals during receive. When transmitting pulses, electrical pulses are applied to the transducer elements, causing the piezoelectric structures to vibrate and generate pressure waves. Upon receive, acoustical vibrations induce electrical signals across the crystals that is sampled and used for image formation.

Echoes are continuously generated as the transmitted pulse propagate through tissue, depending on differences in acoustic impedance in the images tissue, and the fraction of the echo that propagates in the direction back to the transducer is utilized

for imaging. The time between pulse transmission and the received echoes increase linearly with the distance from the transducer to the origin of the echo due to increasing propagation distances. By assuming a constant wave propagation speed c , the delay time t can be converted into radial distance r :

$$r(t) = \frac{1}{2}ct ,$$

where the division by two originates from the fact that sound waves have to travel both down into the body and be reflected back up again, before being received, leading to an effective doubling of propagation distance.

The relation between propagation speed c of ultrasound beams through tissue with density ρ and compressibility κ is $c = 1/\sqrt{\rho\kappa}$. For soft human tissue, c is approximately 1540 m/s. When neglecting absorption, the transmitted ultrasound waves propagate through tissue approximately follows the scalar wave equation [50]:

$$\frac{\partial^2 p(\mathbf{r}, t)}{\partial t^2} = c^2 \nabla^2 p(\mathbf{r}, t) ,$$

where $p(\mathbf{r}, t)$ is the acoustic pressure at spatial position \mathbf{p} on time t . For a constant c , impulses from a point source yields a sphere wave propagating without generating any reflected echo signal. Localized perturbations in tissue density and compressibility, however, interacts with the wave propagation in such a way that some of the wave will be scattered and reflected back as echoes to enabling imaging. Structures that are relatively large compared to the wavelength, such as muscle fibers generate specular reflections, much like a mirror, where the amount of reflected echo depends on the angle of incident. Smaller sources of disturbance, with size smaller than the wavelength generate more omnidirectional scattering, that does not depend on angle. In addition to being scattered, the ultrasound signal is also absorbed while propagating through tissue. This absorption is highly frequency dependent, and increases rapidly with frequency and propagation distance. The total attenuation from absorption and scattering is usually assumed to be approximately 0.5 dB/cmMHz.

Image volumes are constructed by sequentially transmitting beams that are steered in different directions to cover a desired image sector. The response of each transducer element is not focused in any particular direction, but focusing of the ensemble of transmitted beams from all elements can still be accomplished by individually delaying the pulse emitted from each element. A coherent wavefront focused at a particular point are generated by gradually adding delays on the transducer elements depending on the distance between the elements and the focal point such that the elements farthest away from the focal point are excited first, as shown in figure 7.1. The combination of all pulses emitted from each element will then form a narrow focused beam steered in one direction that converge at a certain focal point. This procedure is performed during transmission to generate steered beams with a fixed focus at the approximate center depth of the object being imaged. On receive, the procedure is reversed to listen to echoes originating from the same direction. The focal depth is also continuously updated by means of dynamic focusing during reception to match the

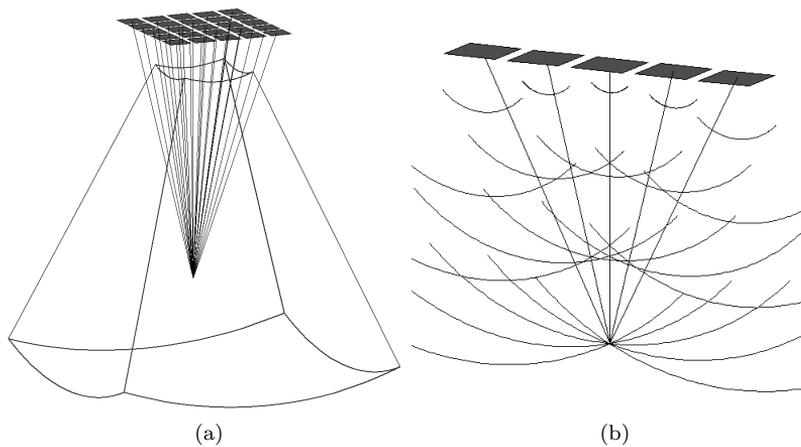


Figure 7.1: Illustration of ultrasound beamforming with phased array transducer. Signals from the transducer elements (top) are individually delayed based on their distance from the focal point. (a) illustrates 3D focusing within an imaging sector (outlined). (b) illustrates how pulses from transducer elements are individually delayed to form a wavefront which converges into a focused pulse at a desired focal point.

current depth of the propagating beam, as shown in figure 7.1 (a) for a matrix array transducer.

Ultrasound images are characterized by highly anisotropic resolution, with much higher resolution in beam direction compared to the lateral direction. The resolution in the beam propagation direction, often referred to as radial direction, is primarily determined by the wavelength λ of the ultrasound beam and the fractional bandwidth B of the imaging system

$$RES_{rad} = \frac{\lambda}{2B} .$$

The division by two due to the fact that the pulses has to propagate back and forth, which effectively doubles the radial resolution compared to one-way imaging systems. In the lateral directions, the resolution in focus is determined by the the ratio between the transducer diameter D and the imaging depth $|\mathbf{r}|$. The lateral resolution is commonly assumed to be [51]:

$$RES_{lat} = \frac{\lambda|\mathbf{r}|}{D} ,$$

based on computing the distance to the first zero in the two-way intensity response when assuming a rectangular transducer aperture with uniform apodization.

These are, however, theoretical limitations that assumes very short pulses and ultrasound beams that are spaced close enough to avoid undersampling of the image. In

practice, the radial resolution degrades with the length of the transmitted pulse and the lateral resolution can be lower if optimal spatial resolution is sacrificed for higher framerate.

7.2 3D imaging challenges

A major challenge faced in 3D echocardiography is the limited propagation velocity of sound in tissue. With a propagation speed of 1540 m/s, it takes approximately $200 \mu\text{s}$ for a single beam to propagate 15 cm down through the heart and back again. For cardiac imaging, it is desirable with framerates in excess of 25 fps to image the heart's dynamics with sufficient temporal resolution. This leaves only time for a little more than 200 transmitted beams per frame. Volume imaging of reasonable quality requires thousands of beams to picture the heart with sufficient detail, which leaves a gap between the number of image beams required and the number of transmit beams available. Two recent innovations in ultrasound imaging have helped alleviate this problem:

First, modern ultrasound scanners therefore uses a technique referred to as *parallel beamforming* to generate several imaging beams per transmitted beam. Parallel beamforming works by transmitting broader beams to illuminate a slightly wider portion of the imaging sector at a time. Several receive beamformers are then used in parallel to simultaneously listen for backscatter echo in slightly different directions within the illuminated area of the imaging sector. This leads to increased imaging rate, which in turn can greatly increase the temporal resolution.

Secondly, gated *subvolume imaging* is used to acquire only a part of the entire heart volume, a subvolume, in each cardiac cycle. The remaining subvolumes are retrieved from previous heartbeats, based on electrocardiography (ECG) gating, so that they together form a complete volume that appears to come from a single heart beat. This is possible because the cardiac cycle is a cyclic process that repeats itself over and over with little variation between beats. After a full volume is acquired, the subvolumes can be replaced in an alternating fashion to update the volume, one subvolume per heartbeat.

Together, parallel beamforming and subvolume imaging can be used to enable real-time 3D echocardiography imaging with both high spatial resolution and sufficient framerate for cardiac imaging.

7.3 Acquisition coordinate system

Due to the directional focusing of phased array transducers, ultrasound images are acquired in polar coordinates. This coordinate system is parametrized by lateral *azimuth* and *elevation* angles, corresponding to rotations relative to the probe, as well as *range* (radial) coordinates for distances from the transducer. The nature of the phased-array beamforming used in ultrasound, where focused beams are transmitted

and received in directions of interest, therefore leads to image data in polar coordinates.

Conversion from polar to Cartesian coordinates, known as *scanconversion*, are required before displaying images on computers. It is also not uncommon to do scanconversion prior to image analysis, since many analysis tools work exclusively with cartesian data. Scanconversion does, however, have the disadvantage that it inevitable leads to some conversion loss, especially in the uppermost areas of the image, where the spatial resolution of ultrasound is highest. The amount of image data also grows significantly, since the pyramidal sector-shape of the ultrasound volume must be padded to form a rectangular volume block.

Chapter 8

State modeling and estimation

A state space model is a mathematical model of a process, where the process' *state* \mathbf{x} is represented by a numerical vector of finite dimensionality. State space models usually consists of two separate models: The first is a kinematic *process model*, which describes how the state propagates in time based on external influences, such as input and sources of uncertainty, referred to as noise. The second model is the *measurement model*, which describe how a vector of numerical measurements \mathbf{z} are taken from the process, typically simulating noisy and indirect measurements of an unknown state.

8.1 General state space models

The most general form of state-space models is the nonlinear model. This model is here described using two functions, f and h :

$$\begin{aligned}\mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k)\end{aligned}$$

which govern state propagation and measurements, respectively. \mathbf{u} is process input, and \mathbf{w} and \mathbf{v} are stochastic state and measurement noise vectors, respectively while k is the discrete time.

State-space models are remarkably usable for modeling almost all sorts of processes within a range of different fields. f and h are usually based upon a set of discretized differential equations, governing the dynamics of and observations from the process.

8.2 Linear state space models

A linear state-space model is a model where the functions f and h are linear in both state and input. The functions can then be expressed by using the matrices \mathbf{F} , \mathbf{B} and

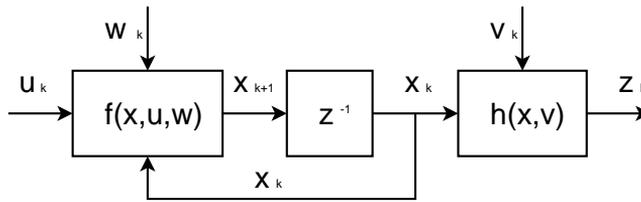


Figure 8.1: A general state-space model. z^{-1} is the unit delay function known from the Z-transform in digital signal processing.

H, reducing state propagation calculations to linear algebra. Overall this results in the following linear state-space model:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k\end{aligned}$$

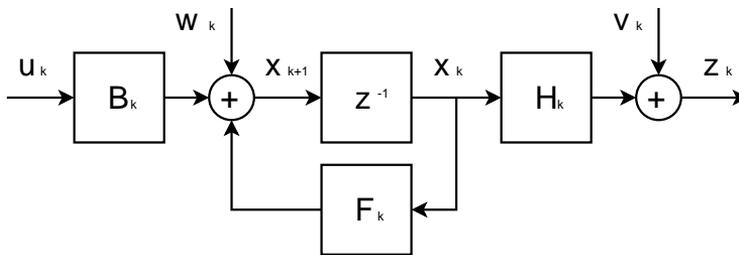


Figure 8.2: A linear state-space model

This linear model is easier both to calculate and analyze, enabling modelers to investigate properties such as controllability, observability and frequency response [52]. Linear state models are either based on inherently linear processes, or simply a linearized version of a nonlinear process by means of a first order Taylor approximation.

8.3 Sequential Bayesian estimation

State estimation concerns the problem of estimating the probability density function (*pdf*) for the state of a process which is not directly observable. This typically involves both predicting the next state based on the current, and updating/correcting this prediction based on noisy measurements taken from the process.

The most general form for state estimation is known as sequential Bayesian estimation [14]. This is the optimal way of predicting a state pdf for any process, given a system and a measurement model. In this section we will discuss this estimator, which recursively calculates a new estimate for each time step, based on the estimate for the previous time step and new measurements.

Sequential Bayesian estimation works by simulating the process, while at the same time adjusting it to account for new measurements \mathbf{z} , taken from the real process. The calculations are performed recursively in a two step procedure. First, the next state is predicted, by extrapolating the current state onto next time step using state propagation belief $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ obtained from function f . Secondly, this prediction is corrected using measurement likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ obtained from function h , taking new measurements into account.

The Chapman-Kolmogorov equation is used to calculate a prior pdf for state \mathbf{x}_k , based on measurements up to time $k - 1$:

$$p(\mathbf{x}_k|\mathbf{z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1}) d\mathbf{x}_{k-1}$$

Bayes rule is then used to calculate the updated pdf for state \mathbf{x}_k , after taking measurements up to time k into account:

$$p(\mathbf{x}_k|z_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{k-1}) d\mathbf{x}_k}$$

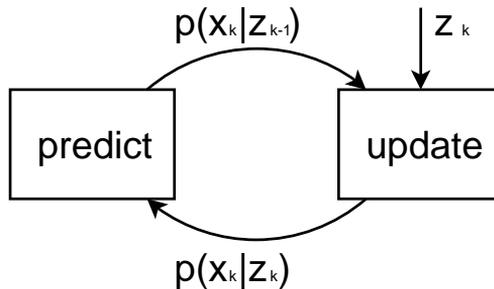


Figure 8.3: *Sequential Bayesian estimator loop.*

Unfortunately, this general method does not scale very well in practice, mainly due to the large state space for multidimensional state vectors. Calculating the prior probability of each point in this state space involves a multidimensional integral, which quickly becomes intractable as the state space grows. Computers are also limited to calculation of the pdf in discrete point in state space, requiring a discretization of the state space. This technique is therefore mainly considered as a theoretic foundation for state estimation in general. Bayesian estimation by means of computers is only possible if either the state space can be discretized efficiently, or if certain limitations apply for the model.

8.4 Kalman filter

The problem of state estimation can be made tractable if we put certain constraints on the system models, by requiring both f and h to be linear functions, and the noise

terms \mathbf{w} and \mathbf{v} to be uncorrelated, white Gaussian processes with zero mean value. Put in mathematical notation, we then have the following constraints:

$$\begin{aligned} f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) &= \mathbf{F}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \\ h(\mathbf{x}_k, \mathbf{v}_k) &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \\ \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q}_k) \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \\ E(\mathbf{w}_i \mathbf{w}_j^T) &= \mathbf{Q}_i \delta_{i-j} \quad E(\mathbf{v}_i \mathbf{v}_j^T) = \mathbf{R}_i \delta_{i-j} \\ E(\mathbf{w}_k \mathbf{v}_k^T) &= \mathbf{0} \end{aligned}$$

where \mathbf{Q} and \mathbf{R} are covariance matrices, describing the second-order properties of the state- and measurement noise, that restricts the allowable rate of change. The constraints described above reduce the state model to:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

where \mathbf{F} , \mathbf{B} and \mathbf{H} are matrices, possible time dependent.

As the model is linear and input is Gaussian, we know that the state and output will also be Gaussian [53]. The state and output pdf will therefore always be normally distributed, where mean and covariance are sufficient statistics. This implies that it is not necessary to calculate a full state pdf any more, a mean vector $\hat{\mathbf{x}}$ and covariance matrix $\hat{\mathbf{P}}$ for the state will suffice.

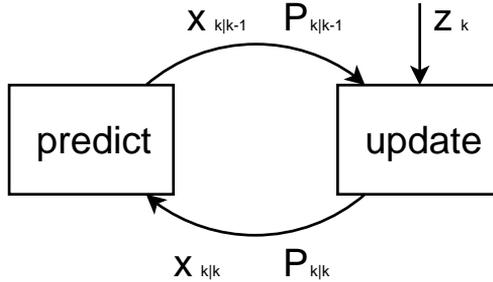


Figure 8.4: *Kalman filter loop*

The sequential Bayesian estimation problem is then reduced to the *Kalman filter* [12], where f and h are replaced by the matrices \mathbf{F} , \mathbf{B} and \mathbf{H} . The Kalman filter is, just as the Bayesian estimator, decomposed into two steps: *predict* and *update*. The actual calculations required are:

Predict next state at time k , before measurements are taken:

$$\begin{aligned} \bar{\mathbf{x}}_k &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k \\ \bar{\mathbf{P}}_k &= \mathbf{F}_k \hat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned}$$

Update state at time k , based on measurements taken:

$$\begin{aligned}\mathbf{K}_k &= \bar{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \bar{\mathbf{x}}_k) \\ \hat{\mathbf{P}}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \bar{\mathbf{P}}_k\end{aligned}$$

where \mathbf{K} is the Kalman gain matrix, used in the update observer, and $\hat{\mathbf{P}}$ is the covariance matrix for the state estimate, containing information about the accuracy of the estimate. More details and background for this filter can be found in [54]. The Kalman filter allows for efficient computer implementations, due to the fact that it is mostly linear, except for a matrix inversion. It can also be proved that the Kalman filter is an optimal estimator of process state in a *minimum mean square error* (MMSE) sense, meaning it minimizes the expected quadratic error.

8.5 Extended Kalman filter

Most processes in real life are unfortunately not linear, and therefore needs to be linearized before they can be estimated by means of a Kalman filter. The extended Kalman filter (EKF) [54] solves this problem by calculating the Jacobian¹ of f and h around the estimated state, which in turn yields a trajectory of the model function centered on this state.

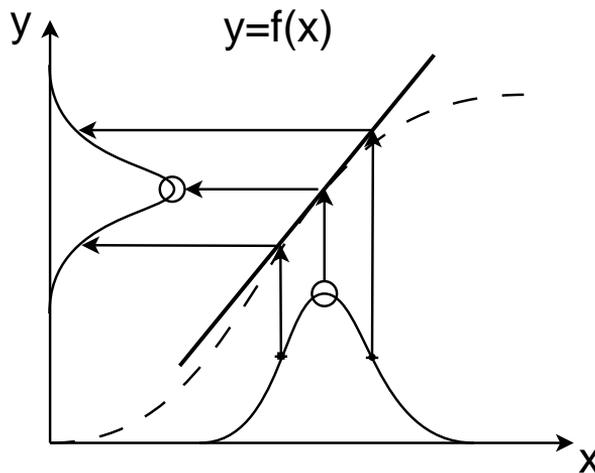


Figure 8.5: *Illustration of how the Extended Kalman filter linearizes a nonlinear function around the mean of a Gaussian distribution, and thereafter propagates the mean and covariance through this linearized model*

¹The Jacobian is the matrix of all partial derivatives of a vector with regards to the state vector.

$$\mathbf{F}_k = \mathbf{J}_x f(\mathbf{x}, \mathbf{u}, \mathbf{w}) = \left. \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k, \mathbf{u}_k, 0}$$

$$\mathbf{H}_k = \mathbf{J}_x h(\mathbf{x}, \mathbf{v}) = \left. \frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}_k, 0}$$

The extended Kalman filter works almost like a regular Kalman filter, except for \mathbf{F} and \mathbf{H} , which vary in time based on the estimated state $\hat{\mathbf{x}}$. The actual calculations required are:

Predict next state, before measurements are taken:

$$\bar{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0)$$

$$\bar{\mathbf{P}}_k = \mathbf{F}_k \hat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

Update state, after measurements are taken:

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

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - h(\bar{\mathbf{x}}_k, 0))$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \bar{\mathbf{P}}_k$$

where \mathbf{K} is the Kalman gain matrix, used in the update observer, and \mathbf{P} is the covariance matrix for the state estimate, containing information about the accuracy of the estimate.

8.6 Information filter

There exists an alternative formulation of the discrete Kalman filter, namely the *information filter*. This form, which is algebraically equivalent to the standard form, is often advantageous in situations where the number of measurements exceed the state dimension, since it avoids the problem of inverting matrices of the size of the measurement covariance. Instead, it requires inversion of matrices with dimensions equal to the state dimension. The information filter also enables efficient processing when many independent measurements are taken at each time step [55].

Just like the standard Kalman filter, this variant starts by predicting the state at the next time step. The equations are equal to those above:

$$\bar{\mathbf{x}}_k = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k$$

$$\bar{\mathbf{P}}_k = \mathbf{F}_k \hat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

The measurement update equations, however, are quite different. They are expressed on a *summing form*, where prior knowledge from the prediction is added to the knowledge from the measurements:

$$\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k$$

$$\hat{\mathbf{P}}_k^{-1} \hat{\mathbf{x}}_k = \bar{\mathbf{P}}_k^{-1} \bar{\mathbf{x}}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k$$

Inversion of the resulting sums are required to acquire the updated state estimate and covariance:

$$\begin{aligned}\hat{\mathbf{P}}_k &= (\bar{\mathbf{P}}_k^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{P}}_k (\bar{\mathbf{P}}_k^{-1} \bar{\mathbf{x}}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k)\end{aligned}$$

By expanding the last equation above and comparing it to the original Kalman equations we discover an alternative form of the Kalman gain, namely:

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

This form collects everything related to the measurements into $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ and $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}$, which are referred to as the *information matrix*, and *information vector*, respectively.

8.7 Multiple independent measurements

One often encounters situations where several independent scalar measurements are taken at each time step. The information filter formulation of the Kalman filter can often be computationally more efficient in such situations, as will be shown below.

The calculation of information-matrices and -vectors for independent measurements yield equations involving inversion of the diagonal measurement covariance matrix. The structure of the equations can then be exploited to convert it into efficient summations:

$$\begin{aligned}\mathbf{H}^T \mathbf{R}^{-1} \mathbf{z} &= \begin{bmatrix} | & | & | \\ \mathbf{h}_1 & \cdots & \mathbf{h}_N \\ | & | & | \end{bmatrix} \begin{bmatrix} r_1^{-1} & 0 & 0 \\ 0 & \cdots & 0 \\ 0 & 0 & r_N^{-1} \end{bmatrix} \begin{bmatrix} z_1 \\ \cdot \\ z_N \end{bmatrix} \\ &= \sum_i \mathbf{h}_i r_i^{-1} z_i \\ \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} &= \begin{bmatrix} | & | & | \\ \mathbf{h}_1 & \cdots & \mathbf{h}_N \\ | & | & | \end{bmatrix} \begin{bmatrix} r_1^{-1} & 0 & 0 \\ 0 & \cdots & 0 \\ 0 & 0 & r_N^{-1} \end{bmatrix} \begin{bmatrix} - & \mathbf{h}_1^T & - \\ - & \cdots & - \\ - & \mathbf{h}_N^T & - \end{bmatrix} \\ &= \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T\end{aligned}$$

This formulation replaces the expensive measurement covariance matrix inversion with simple scalar divisions. It also involves fewer multiplications. The principal advantage of this formulation, is that it makes it conceptually simple to fuse information from multiple independent scalar measurements together, simply by summing information-matrices and -vectors together.

This form of measurement processing is an alternative to *sequential processing* of scalar measurements [56] which avoids the need for recursively updating the estimate covariance for each new measurement.

8.8 Measurement likelihood

Measurements are modeled as stochastic processes, affected by multivariate Gaussian noise. The probability density distribution for single measurements hence becomes a scalar Gaussian probability density function centered around the predicted position, with a variance equal to the measurement noise.

$$z_i \sim \mathcal{N}(\bar{z}, s) = \mathcal{N}(\mathbf{h}_i^T \mathbf{x}_i, \mathbf{h}_i^T \bar{\mathbf{P}} \mathbf{h}_i + r)$$

This leads to the following measurement likelihood function:

$$p(z_i | \mathbf{x}_i) = \frac{1}{(2\pi)^{1/2} |\mathbf{h}_i^T \bar{\mathbf{P}} \mathbf{h}_i + r|^{1/2}} \exp\left(-\frac{1}{2}(z_i - \mathbf{h}_i^T \mathbf{x}_i)^T (\mathbf{h}_i^T \bar{\mathbf{P}} \mathbf{h}_i + r)^{-1} (z_i - \mathbf{h}_i^T \mathbf{x}_i)\right)$$

Knowledge of the likelihood associated with the measurements can be used to filter out improbable measurements, which can help improve tracking robustness.

The *de facto* way of performing this is by using the measurement innovation and error covariance to calculate the *Normalized Innovation Squared* (NIS) as defined by Bar-Shalom et al. in [2]. With scalar measurements, the NIS becomes:

$$NIS_i \propto v_i^T (\mathbf{h}_i^T \bar{\mathbf{P}} \mathbf{h}_i + r)^{-1} v_i$$

We clearly see that NIS grows proportional to the square of the innovations. Large measurement innovations will therefore typically lead to large NIS values, which can be rejected by simple thresholding.

8.9 Iterated Kalman filter

Nonlinear measurement models can, when coupled with state prediction errors, lead to large estimate errors. To limit this effect, it is possible to modify the measurement update step to iteratively compute an approximate maximum a posteriori (MAP) estimate instead of the regular conditional mean estimate [2].

The approach is to iterative maximize the conditional pdf of $\hat{\mathbf{x}}$ with the Newton-Raphson algorithm. This leads to a second order Taylor-expansion of log-likelihood of the covariance, that is minimized iteratively. Altogether, this results in a three step algorithm as follows:

Relinearize the measurement matrix for each iteration, based on the current estimate:

$$\mathbf{H}^i \equiv \frac{\partial \mathbf{h}(\hat{\mathbf{x}}^i)}{\partial \mathbf{x}}$$

Compute a updated state estimate for each iteration with a modified formula:

$$\begin{aligned}\hat{\mathbf{x}}^0 &= \bar{\mathbf{x}} \\ \hat{\mathbf{x}}^{i+1} &= \hat{\mathbf{x}}^i + \hat{\mathbf{P}}^i \mathbf{H}^{iT} \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}^i)) - \hat{\mathbf{P}}^i \bar{\mathbf{P}}^{-1} (\hat{\mathbf{x}}^i - \bar{\mathbf{x}}) \\ \hat{\mathbf{x}} &= \hat{\mathbf{x}}^N\end{aligned}$$

Compute a updated covariance matrix for each iteration with the regular formula:

$$\begin{aligned}\hat{\mathbf{P}}^i &= \bar{\mathbf{P}} - \bar{\mathbf{P}} \mathbf{H}^{iT} (\mathbf{H}^i \bar{\mathbf{P}} \mathbf{H}^{iT} + \mathbf{R}^{-1})^{-1} \mathbf{H}^i \bar{\mathbf{P}} \\ \hat{\mathbf{P}} &= \hat{\mathbf{P}}^N\end{aligned}$$

This process can either be conducted a fixed number of iterations, or until some convergence criteria is met.

The iterative Kalman filter provides a mean for iterative refinement of the estimate, by iteratively taking new measurements relative to the previous estimate. This can be useful if the measurement model contains strong non-linearities, or if the measurement values are truncated in some way. Relevant examples of measurement truncation within image analysis are the application of finite search windows for edge-detection and block matching. The benefits of iterative refinement does, however, come at the cost of increased computational complexity, since both the measurements and the Kalman update step has to be conducted over again for each iteration.

8.10 Kalman smoother

All regular Kalman filter variants incorporate all available information up until time k when computing an estimate for time k . No information about the future, whether available or not, is not exploited. This property makes the Kalman filter suited for real-time applications, where data is continuously processed, but the filter might be a sub-optimal choice in situations of post-processing data acquired in advance.

The Kalman smoother [2] solves the problem of estimating the state at time k , based on all available information, both before and after. This form of estimation is not a part of sequential state estimation in a strict sense, but is still included since it is closely related to the ordinary Kalman filter, and can be useful in several situations.

It turns out that a smoothed Kalman estimate can be computed by combining the estimates from a forward and a backwards running Kalman filter operating from time 1 to N [40]. Denoting $\hat{\mathbf{x}}_{k+}$, $\hat{\mathbf{P}}_{k+}$ for the forward running estimate, and $\hat{\mathbf{x}}_{k-}$, $\hat{\mathbf{P}}_{k-}$ for the backwards running estimate at time step k , they can be combined into a smoothing estimate that incorporates all available knowledge as follows:

$$\begin{aligned}\hat{\mathbf{x}}_{k|N} &= \mathbf{P}_{k|N}^{\hat{}} \left(\hat{\mathbf{P}}_{k-}^{-1} \hat{\mathbf{x}}_{k-} + \hat{\mathbf{P}}_{k+}^{-1} \hat{\mathbf{x}}_{k+} \right) \\ \mathbf{P}_{k|N}^{\hat{}} &= \left(\hat{\mathbf{P}}_{k-}^{-1} + \hat{\mathbf{P}}_{k+}^{-1} \right)^{-1}\end{aligned}$$

The backwards running filter should start with infinite covariance at the terminal time. This leads to an unchanged estimate for the last time, while all previous estimates are corrected to account for measurements from the future.

The Kalman smoother can be used to enforce cyclic behavior by wrapping the end of the data series back to the beginning. It can also be used to counteract the inherent inertia of the estimates from Kalman filters due to temporal regularization that only pulls the estimates back in time, but not forward.

Chapter 9

Model-based image segmentation

This chapter is intended to give a brief overview and categorization of common approaches for image segmentation that are relevant within medical imaging. The overview is limited to approaches based on deformable models in 3D, and focuses on the most commonly reported approaches. For more in-depth reviews, consult some of the excellent reviews available on model-based image analysis in general [33, 57, 58], and the ones that are more geared towards echocardiography in specific [59, 60].

Image segmentation deals with the process of analyzing and interpreting images, with the aim of separating the image into the different parts or structures it consists of, referred to as *segments*. Within medical imaging, segments typically correspond to specific parts of the body, such as organs, or other anatomic structures. It is very common, although not strictly necessary in all cases, to utilize prior knowledge about the structure of interest to guide the segmentation process, such as e.g. position, shape, texture and motion.

Most segmentation techniques that rely on prior knowledge for image segmentation can conceptually be decomposed into three main modules, namely *deformable model*, *fitting algorithm* and *image measurements*, as is shown in Fig. 9.1. These modules handles different parts of the processing, and their content can vary widely between the different approaches. The remaining parts of this chapter will walk through the aspects of these modules.

9.1 Deformable models

Deformable models act as prior knowledge for the shape and geometry of the structure being searched for. They come in many varieties, but typically incorporate some information about the shape, size, means of deformation, and possibly texture or boundary contrast. This information is utilized by the fitting algorithm to both guide the search for the structure in the image, as well as regularizing the result.

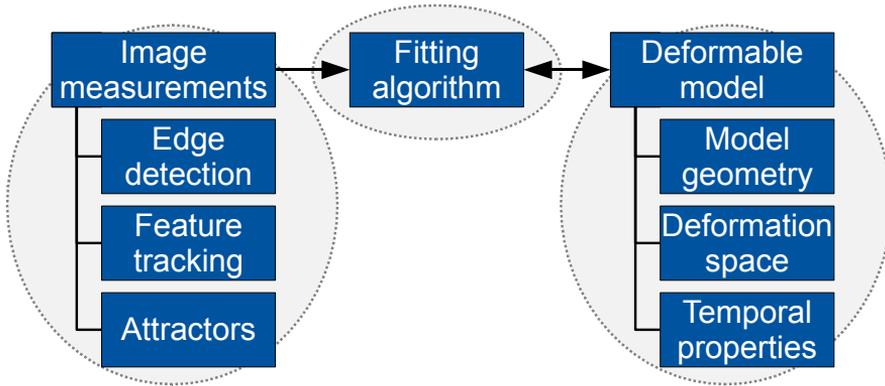


Figure 9.1: Relationship between the main modules involved in image segmentation with deformable models. The fitting algorithm in the center utilizes input from the image measurement sources to update various aspects of the deformable model.

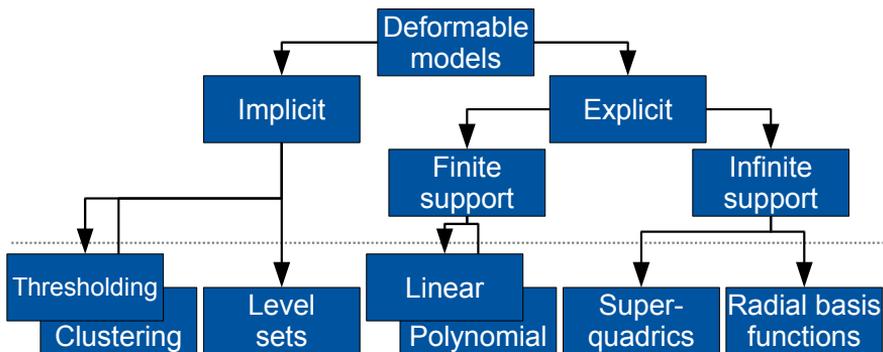


Figure 9.2: Categorization of some common types of deformable models.

Fig. 9.2 shows an overview over some of the most common types of deformable models, in a manner similar to the categorization in [57]. As shown, most types of deformable models can be categorized as belonging in one of two main groups, namely *implicit* or *explicit*, based on their parametrization.

Explicit models

Explicit models are models where a direct mapping from parametric model locations to spatial surface coordinates exists, depending on model parameters. Examples of explicit models include polygonal and higher order surfaces, such as subdivision and radial basis-function surfaces. Explicit models can be further decomposed based on their area of support and interpolation scheme.

Polygon and simplex meshes, spline surfaces and subdivision surfaces are characterized by *finite support* of their basis functions. This means that surface points are only dependent on parameters corresponding to nodes in a local neighborhood of the point, with the size of the neighborhood dependent on the interpolation degree. The effect of any change in a surface parameter will then be constrained to only affect a localized region of the model. The locality of the region depends on the *interpolation* degree, with linear polygonal surfaces having the narrowest area of support.

Radial basis-function surface thin-plate splines [61] and superquadrics [62], however, are examples of *infinite support* surfaces, where surface points are dependent on the location of every single node that parametrize the model. Any change to a surface parameter will then have a global effect on the entire model to some extent.

Finite support is often considered a desirable property, since it leads to more intuitive parametrization that corresponds to the different regions of the model. The sparse relationship between surface points and parametrization can also be utilized to improve the computational efficiency of the fitting process in many situations.

Implicit surfaces

Implicit surfaces, however, do not have a direct mapping between shape and parametrization. Examples of implicit models include level set surfaces [63], as well as various clustering and thresholding approaches, such as for instance watershed segmentation [64]. The lack of direct mapping between shape and parametrization has the advantage that it allows for a wider range of shapes and topologies to be represented, compared to explicit surfaces. This includes graceful handling of large deformation, adoption to complex geometry, handling of convoluted and self-intersecting surfaces and changes in topology over time, all of which can pose problems for explicit surfaces.

It is, however, more difficult to incorporate prior shape knowledge into implicit surfaces, compared to explicit surfaces, due to the lack of direct mapping between shape and parametrization. Adoption of implicit surfaces can also be slow, since implicit method often require processing of the entire image volume, whereas explicit models can be adapted by sparsely evaluating the image data in proximity of the model. One notable exception to processing of the entire image volume is narrow-band level sets

[65], where the computations are restricted to a thin band of image data near the surface.

Notable differences

Implicit surfaces usually focus on features of the object interior, such as color or texture, whereas explicit models are usually more geared against features of the object boundary, such as gradients and edges. Objects are usually surrounded by clear edges in images, that makes them stand out from their surroundings. Detection of edges therefore provides a strategy for object detection in images, which makes them detectable. Most fitting algorithms for explicit models therefore utilize edge-detection algorithms to search for edges present in proximity to the contour. This is often done by searching for edges in the normal direction of a predicted contour at regularly spaced intervals. The resulting distance between contour and edge is referred to as *normal displacement* [3], and can be used to update the model.

An important advantage of explicit models compared to implicit models is reduced computational complexity, because only the object boundary needs to be examined, and not the entire interior. This does, however, require objects with clearly present edges, that can be used to detect the object. Together, this results in what is primarily an edge-detection problem, instead of a region detection problem.

9.2 Deformation schemes

There are several different ways of allowing explicit models to deform. Some of the most common ways include the following:

- *Free-form* deformations, where all parameters in the model are allowed to vary freely without any constraints. A common example of free-form deformation models previously applied to cardiac segmentation are polygon or simplex-meshes [66, 67]. Other examples include spline or subdivision surfaces where the coordinates of the control point grid for a spline surface are used as parameters.
- *Linear subspace* deformations, where the valid range of deformations fall within a linear subspace of the allowable deformation space with free-form deformations. Active-shape models (ASM) is probably the most common type of linear subspace deformations [10]. For ASM's, the deformation subspace is chosen to be the eigenvectors of the most dominant eigenvalues, based on computing the spatial inter-point covariance between a set of reference shapes. There are also simpler approaches to determining the deformation subspace, such as by restricting nodes to only deform in the surface-normal direction. An other example of linear subspace models are active appearance models (AAM) [68, 69], that combine subspace deformations of polygonal models with matching of a gray level texture subspace, corresponding to the object boundary.

- More *general subspace* deformation, extend subspace deformations to nonlinear manifolds [70]. This class of deformations does not suffer from the limitations of linear functions, and can therefore better adapt to big deformations and deformations involving rotations.
- *Subspace-constrained* methods, is an intermediate class where the model is regularized towards the deformation subspace instead of being strictly enforced to the subspace [9]. This class of methods has the advantage that the predetermined subspace does not need to encompass all types of deformations that can be encountered. Segmentation can therefore still succeed when encountering new types of heart diseases that were not present during method tuning.

9.3 Temporal properties

Analysis of image sequences over time usually do not process each frame independently. Instead, it is common to exploit similarities between successive image frames to guide the search and regularize the analysis process. Some of the most common aspects of temporal modeling includes:

- *Initialization* of the analysis of each image frame. This can for instance be based on the result from the previous frame.
- It is common to model *temporal smoothness* to achieve smooth deformations over time and prevent unrealistic behavior such as shaking between frames. Newtonian model with mass and damping parameters is a common way of introducing temporal smoothness [71]. Temporal properties can also be modeled as a fourth dimension in the deformation pattern for subspace deformation, as is common for active appearance models [72].
- For analysis of *periodic* deformations, such as the heart cycle, one can also exploit the cyclicity to assume that the net deformation after a full cycle is zero. In addition, one can also assume a repeating deformation pattern for cardiac segmentation, so that the phase of the cardiac cycle can be used to predict deformation.

9.4 Fitting algorithm

There are many different types of fitting algorithms in use. The choice of fitting algorithm depends on a combination of the model representation, the problem at hand, the computational resources available, as well as convention and personal preference. Most algorithms can be categorized to belong in either of the following groups:

Iterative refinement

These methods are by large the most common group of methods, and encompass a wide group of approaches. They are typically based on iterative minimization of energy function that relates to the quality of the fitting. Some of the most common approaches in this group include:

- *Force-based refinement* schemes for marginal update of each model parameter until force equilibrium is reached [73]. The simplex mesh scheme [67] is a typical example of a force-based scheme.
- *Gradient descent* and similar methods, like the deformable snake update scheme [73], and in part also the active shape scheme [10]. These methods are based on computing the local gradient of the energy function, and walk in the direction of the gradient vector to update model parameters.
- *Marginal-space refinement* methods is a class of methods where only parts of the state space is explored on each refinement iteration [26].
- *Expectation-maximization algorithm* [74], where the update iteration are divided between a expectation prediction stage and a maximization fitting stage .
- *Iterated Kalman filter* updates, that combine Kalman fitting with iterative refinement to reduce the effect of non-linearities and measurement truncation [2].

Direct fitting

This group of methods are capable of fitting models to measurements directly, either through analytic solutions or least squares approximations. Some common approaches of this include:

- *Analytic fitting* methods for simple geometric shapes, such as superquadrics [62].
- Optimal fitting of a deformable contour with *dynamic programming* [75]. This approach is, however, mostly relevant for 2D contours, since it does not extend easily into 3D surfaces.
- *Least squares fitting* of spline surfaces.
- *Kalman filter* variants, such as the extended Kalman filter [5] and related information fusion approaches [9].

Sampling-based schemes

Most direct fitting and iterative refinement methods will converge to the nearest local minimum. More global optimal results can be accomplished by using more general class of Bayesian estimation methods referred to as sampling-bases schemes. Such

schemes use sequential Monte Carlo methods [14, 76] to stochastically search for a global minimum in a more general multi-modal parameter probability distributions.

The *condensation algorithm* by Isard and Blake [77] uses a statistical sampling-based scheme, known as Sampling Importance Resampling (SIR), to track deformable contours in video streams. This algorithm uses edge measurements to weight the likelihood of each contour sample, and then randomly perturbing the most promising samples to explore the state space. The advantage of this approach is that it is capable of tracking of several objects in a cluttered environments, where objects might disappear and overlap. It also capable of handling several candidates for the most likely edge in each edge profile due to its multi-modal nature. The downside of SIR approaches is that they are very computationally intensive, due to the requirement for propagating a large number of parallel samples in order to ensure good results.

A limitation of the basic sampling schemes is that they do not exploit any gradient or Jacobian information to selectively steer the samples in the direction of high likelihood, as is done in Kalman filtering. This can, however, be achieved by combining the sampling scheme with a Kalman filter to compute a sample prediction distribution, and to correct the samples based on image measurements [78].

An alternative to stochastic sampling is to use the *unscented Kalman filter* [15]. This filter replaces the Jacobian calculations in the Kalman filter with a limited set of deterministically chosen samples that are used to propagate the mean and covariance of the state estimate. This results in better behavior for highly non-linear systems, due to the elimination of linearized Jacobian matrices.

9.5 Image measurements

Many different types of input measurements can be used to update the models. Most of them can, however, be categorized into the following categories:

Edge and pattern detection

Edge detection refers to the process of detecting edges in the image, that corresponding to the surface of the model. This is usually accomplished by some sort of search for strong gradients in the image, which results in measurements in absolute coordinates. Edge detection does therefore not suffer from drift over time. It should be noted that edge measurements are usually unable of measuring displacements along the model surface, and does therefore not necessarily follow material points when applied over a sequence of frames.

Pattern or texture detection is a process similar to edge detection. The difference is that that instead of detecting positions of high gradient, it detects edges by matching them against predefined edge patterns by means of texture matching.

Feature tracking

Feature tracking refers to the process of tracking the movement of image features between successive frames. This is often accomplished by means of block matching [79], optical flow tracking [80, 81], or elastic registration [82], and gives measurements of relative coordinates between image frames. Feature tracking measurements does therefore suffer from drift due to cumulative buildup of error over time, which must be counteracted by either combining them with absolute measurements, or by some sort of temporal regularization. The advantage of feature tracking is that it follows material points, and therefore can be used for regional strain measurement, in addition to tracking of shape.

Attractors

Most fitting algorithms will eventually fail at some point. It is therefore common to allow for manual adjustment to the model, based on user input to place surface attractors to correct the model. This correction can either occur interactively during the fitting procedure, or be incorporated as a validation and correction step after model fitting has first been performed. Attractors might not strictly be regarded as a type of image measurement, but they can still be treated as such by the fitting algorithm, since they connect surface points to spatial image coordinates. Attractors are typically considered as a strong model constraint associated with a low spatial uncertainty, since the user will want the model to intersect his input points.

9.6 The role of regularization

Regularization plays a different role in the different modules of an image segmentation framework. The roles act complimentary, and can together aid the segmentation process by improving both accuracy and robustness.

Input images are usually assumed to form a smooth intensity field, except in areas between objects, where abrupt changes are expected. Direct smoothing of the fields or image measurements should therefore be conducted with caution. Instead, outlier rejection should be employed to eliminate anomalies from the measurement field. Outliers are measurement anomalies that does not follow the regular noise statistics. They can therefore severely degrade the result if not eliminated from the processing chain at an early stage as possible.

Usage of *geometric models* often impose inherent smoothness constraints. The field of image measurements are effectively smoothed when projected into to the model. This smoothing is, however, limited to the surface, and possible interior, of the model. Not smoothing needs to be applied across the model surface, where no assumption of continuity is assumed.

Temporal properties are used to regularize towards smooth behavior over time. They can also enforce cyclicity in cases where this is relevant, such as the heart cycle.

Subspace *deformations* can be used to regularize shape deformations towards physiological plausible shapes, thereby avoiding or lessening the chance of unrealistic results.

Bibliography

- [1] S. Allender, P. Scarborough, V. Peto, and M. Rayner, *European cardiovascular disease statistics*. European Heart Network, 2008. [Online]. Available: <http://www.heartstats.org/uploads/documents%5Cproof30NOV2007.pdf>
- [2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.
- [3] A. Blake, R. Curwen, and A. Zisserman, “A framework for spatiotemporal control in the tracking of visual contours,” *International Journal of Computer Vision*, vol. 11, no. 2, pp. 127–145, Oct. 1993. [Online]. Available: <http://dx.doi.org/10.1007/BF01469225>
- [4] A. Blake, M. Isard, and D. Reynard, “Learning to track the visual motion of contours,” *Artificial Intelligence*, vol. 78, no. 1-2, pp. 179–212, Oct. 1995. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TYF-4031CCJ-7/2/da4ffae4d4f1c900ad57df1aea34e9c2>
- [5] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998. [Online]. Available: <http://www.robots.ox.ac.uk/~contours/>
- [6] G. Jacob, J. Alison Noble, M. Mulet-Parada, and A. Blake, “Evaluating a robust contour tracker on echocardiographic sequences,” *Medical Image Analysis*, vol. 3, no. 1, pp. 63–75, Mar. 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/B6W6Y-44G8C12-5/2/5dc5c34244792353437cf706861e176b>
- [7] G. Jacob, J. A. Noble, A. D. Kelion, and A. P. Banning, “Quantitative regional analysis of myocardial wall motion,” *Ultrasound in Medicine & Biology*, vol. 27, no. 6, pp. 773–784, Jun. 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TD2-438BW6D-6/2/b0d328727cff728f791b4a1e2391e17a>
- [8] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning, “A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography,” *Medical Imaging, IEEE Transactions on*, vol. 21, no. 3, pp. 226–238, 2002.

- [9] D. Comaniciu, X. S. Zhou, and S. Krishnan, "Robust real-time myocardial border tracking for echocardiography: An information fusion approach," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 7, pp. 849–860, 2004.
- [10] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The use of active shape models for locating structures in medical images," in *Proc. Conf. Information Processing in Medical Imaging*, July 1993, pp. 33–47.
- [11] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, no. 6, pp. 356–360, Nov. 1978. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TYR-481DYSP-1GD/2/c1ae713b367b3b2e077fc74b0d84d389>
- [12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [13] "Sequential state estimation: From kalman filters to particle filters," *Proceedings of the IEEE*, 2004.
- [14] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [15] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Mar. 2004.
- [16] F. Orderud, "A framework for real-time left ventricular tracking in 3D+T echocardiography, using nonlinear deformable contours and kalman filter based tracking," in *Computers in Cardiology*, 2006.
- [17] F. Orderud, J. Hansegård, and S. I. Rabben, "Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, ser. LNCS, vol. 4791. Springer, 2007, pp. 858–865.
- [18] F. Orderud and S. I. Rabben, "Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces," *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, 2008. [Online]. Available: <http://www.idi.ntnu.no/~fredrior/files/orderud08cvpr.pdf>
- [19] J. Hansegård, F. Orderud, and S. Rabben, "Real-time active shape models for segmentation of 3D cardiac ultrasound," in *Computer Analysis of Images and Patterns - CAIP*, 2007, pp. 157–164.
- [20] J. 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*. New York, NY, USA: ACM Press, 1998, pp. 395–404.

- [21] F. Orderud, G. Kiss, and H. G. Torp, "Automatic coupled segmentation of endo- and epicardial borders in 3D echocardiography," *to appear in Ultrasonics Symposium, 2008. IEEE*, 2-5 Nov. 2008.
- [22] H. Barrow and J. Tenenbaum, "Computational vision," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 572–595, May 1981.
- [23] F. Orderud, G. Kiss, S. Langeland, E. W. Remme, H. G. Torp, and S. I. Rabben, "Real-time left ventricular speckle-tracking in 3d echocardiography with deformable subdivision surfaces," in *MICCAI 2008 Workshop on Analysis of Functional Medical Images*, 2008, pp. 41–48.
- [24] F. Orderud, G. Kiss, S. Langeland, E. Remme, H. Torp, and S. Rabben, "Combining edge detection with speckle-tracking for cardiac strain assessment in 3d echocardiography," in *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, Nov. 2008, pp. 1959–1962.
- [25] A. Lefohn, J. Cates, and R. Whitaker, "Interactive, gpu-based level sets for 3d segmentation," pp. 564–572, 2003. [Online]. Available: <http://www.springerlink.com/content/kk9bax62dc0666a0>
- [26] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Fast automatic heart chamber segmentation from 3d ct data using marginal space learning and steerable features," in *IEEE 11th International Conference on Computer Vision*, 2007. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4408925&isnumber=4408819>
- [27] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu, "3d ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.
- [28] Q. Duan, E. Angelini, S. Homma, and A. Laine, "Real-time segmentation of 4d ultrasound by active geometric functions," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, May 2008, pp. 233–236.
- [29] F. Orderud, H. Torp, and S. I. Rabben, "Automatic alignment of standard views in 3d echocardiograms using real-time tracking," S. A. McAleavey and J. D'hooge, Eds., vol. 7265, no. 1. SPIE, 2009, p. 72650D. [Online]. Available: <http://link.aip.org/link/?PSI/7265/72650D/1>
- [30] K. Y. E. Leung, M. van Stralen, A. Nemes, M. M. Voormolen, G. van Burken, M. L. Geleijnse, F. J. ten Cate, J. H. C. Reiber, N. de Jong, A. F. W. van der Steen, and J. G. Bosch, "Sparse registration for three-dimensional stress echocardiography," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 11, pp. 1568–1579, Nov. 2008.
- [31] X. Lu, B. Georgescu, Y. Zheng, J. Otsuki, and D. Comaniciu, "Autompr: Automatic detection of standard planes in 3D echocardiography," *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pp. 1279–1282, May 2008.

- [32] J. Hansegard, S. Urheim, K. Lunde, S. Malm, and S. Rabben, "Semi-automated quantification of left ventricular volumes and ejection fraction by real-time three-dimensional echocardiography," *Cardiovascular Ultrasound*, vol. 7, no. 1, p. 18, 2009. [Online]. Available: <http://www.cardiovascularultrasound.com/content/7/1/18>
- [33] A. Frangi, W. Niessen, and M. Viergever, "Three-dimensional modeling for functional analysis of cardiac images, a review," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 1, pp. 2–5, Jan 2001.
- [34] M. Takeuchi, T. Nishikage, V. Mor-Avi, L. Sugeng, L. Weinert, H. Nakai, I. S. Salgo, O. Gerard, and R. M. Lang, "Measurement of left ventricular mass by real-time three-dimensional echocardiography: Validation against magnetic resonance and comparison with two-dimensional and m-mode measurements," *Journal of the American Society of Echocardiography*, vol. 21, no. 9, pp. 1001–1005, Sep. 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/B6WMB-4TBG8GS-8/2/348048345b1a912a7fdac81a1e63b0e6>
- [35] A.-C. Pouleur, J.-B. le Polain de Waroux, A. Pasquet, B. L. Gerber, O. Gerard, P. Allain, and J.-L. Vanoverschelde, "Assessment of Left Ventricular Mass and Volumes by Three-Dimensional Echocardiography in Patients with or without Wall Motion Abnormalities: Comparison against Cine Magnetic Resonance Imaging," *Heart*, p. hrt.2007.123711, 2007. [Online]. Available: <http://heart.bmj.com/cgi/content/abstract/hrt.2007.123711v1>
- [36] J. D'hooge, B. Bijnens, J. Thoen, F. Van de Werf, G. Sutherland, and P. Suetens, "Echocardiographic strain and strain-rate imaging: a new tool to study regional myocardial function," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1022–1030, Sep 2002.
- [37] F. Maffessanti, H.-J. Nesser, L. Weinert, R. Steringer-Mascherbauer, J. Niel, W. Gorissen, L. Sugeng, R. M. Lang, and V. Mor-Avi, "Quantitative evaluation of regional left ventricular function using three-dimensional speckle tracking echocardiography in patients with and without heart disease," *The American Journal of Cardiology*, vol. 104, no. 12, pp. 1755 – 1762, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/B6T10-4XV98J2-12/2/3f2edb4e355573bdf73a75cc1d31d4df>
- [38] T. Papavassiliu, H. P. Kühl, M. Schröder, T. Süselbeck, O. Bondarenko, C. K. Böhm, A. Beek, M. M. B. Hofman, and A. C. van Rossum, "Effect of endocardial trabeculae on left ventricular measurements and measurement reproducibility at cardiovascular mr imaging," *Radiology*, vol. 236, no. 1, pp. 57–64, 2005. [Online]. Available: <http://radiology.rsna.org/content/236/1/57.abstract>
- [39] Y. Seo, T. Ishizu, Y. Enomoto, H. Sugimori, M. Yamamoto, T. Machino, R. Kawamura, and K. Aonuma, "Validation of 3-Dimensional Speckle Tracking Imaging to Quantify Regional Myocardial Deformation," *Circ Cardiovasc Imaging*, p. CIRCIMAGING.109.858480, 2009. [Online]. Available: <http://circimaging.ahajournals.org/cgi/content/abstract/CIRCIMAGING.109.858480v1>

- [40] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *Automatic Control, IEEE Transactions on*, vol. 14, no. 4, pp. 387–390, Aug 1969.
- [41] R. M. Lang, M. Bierig, R. B. Devereux, F. A. Flachskampf, E. Foster, P. A. Pellikka, M. H. Picard, M. J. Roman, J. Seward, J. Shanewise, S. Solomon, K. T. Spencer, M. St. John Sutton, and W. Stewart, "Recommendations for chamber quantification," *Eur J Echocardiogr*, vol. 7, no. 2, pp. 79–108, 2006. [Online]. Available: <http://ejechocard.oxfordjournals.org/cgi/content/abstract/7/2/79>
- [42] A. Heimdal, A. Støylen, H. Torp, and T. Skjærpe, "Real-time strain rate imaging of the left ventricle by ultrasound, , ,," *Journal of the American Society of Echocardiography*, vol. 11, no. 11, pp. 1013 – 1019, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/B6WMB-4HG NR27-2/2/b01b934a72f8f2248b1e9abe38cda3e2>
- [43] H. Azhari, J. L. Weiss, W. J. Rogers, C. O. Siu, and E. P. Shapiro, "A noninvasive comparative study of myocardial strains in ischemic canine hearts using tagged mri in 3-d," *American journal of physiology. Heart and circulatory physiology*, vol. 268, no. 5, pp. 1918–, 1995. [Online]. Available: http://x-port-sfx.uio.no/sfx_ubit?sid=google;auinit=H;aulast=Azhari;atitle=A%20noninvasive%20comparative%20study%20of%20myocardial%20strains%20in%20ischemic%20canine%20hearts%20using%20tagged%20MRI%20in%203-D;title=American%20journal%20of%20physiology.%20Heart%20and%20circulatory%20physiology;volume=268;issue=5;date=1995;spage=1918;issn=0363-6135
- [44] J. N. Cohn, R. Ferrari, and N. Sharpe, "Cardiac remodeling—concepts and clinical implications: a consensus paper from an international forum on cardiac remodeling," *Journal of the American College of Cardiology*, vol. 35, no. 3, pp. 569 – 582, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/B6T18-3YMW36V-5/2/2efcb67bfebb092128a12f81921f19ee>
- [45] C. Jenkins, K. Bricknell, and T. H. Marwick, "Use of real-time three-dimensional echocardiography to measure left atrial volume: Comparison with other echocardiographic techniques," *Journal of the American Society of Echocardiography*, vol. 18, no. 9, pp. 991 – 997, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/B6WMB-4H28W9X-18/2/69b89a77854cbc8203453ab6fc547a86>
- [46] F. Haddad, S. A. Hunt, D. N. Rosenthal, and D. J. Murphy, "Right Ventricular Function in Cardiovascular Disease, Part I: Anatomy, Physiology, Aging, and Functional Assessment of the Right Ventricle," *Circulation*, vol. 117, no. 11, pp. 1436–1448, 2008. [Online]. Available: <http://circ.ahajournals.org>
- [47] S. Kapetanakis, M. Kearney, A. Siva, N. Gall, M. Cooklin, and M. Monaghan, "Real-Time Three-Dimensional Echocardiography: A Novel Technique to Quantify Global Left Ventricular Mechanical Dyssynchrony," *Circulation*, vol. 112, no. 7, pp. 992–1000, 2005. [Online]. Available: <http://circ.ahajournals.org/cgi/content/abstract/112/7/992>

- [48] T. Kono, H. N. Sabbah, P. D. Stein, J. F. Brymer, and F. Khaja, "Left ventricular shape as a determinant of functional mitral regurgitation in patients with severe heart failure secondary to either coronary artery disease or idiopathic dilated cardiomyopathy." *American journal of cardiology*, vol. 68(4), no. 0002-9149 (Print), pp. 355–9, 1991.
- [49] S. B. Barnett, G. R. T. Haar, M. C. Ziskin, H.-D. Rott, F. A. Duck, and K. Maeda, "International recommendations and guidelines for the safe use of diagnostic ultrasound in medicine," pp. 355–366, Mar. 2000. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0301562900002040>
- [50] M. E. Anderson and G. E. Trahey, "k-space applied to medical ultrasound," *Department of Biomedical Engineering, Duke University*, 2006.
- [51] B. Angelsen and H. Torp, *Ultrasound Imaging*, 2000.
- [52] C.-T. Chen, *Linear System Theory and Design, third edition*. Oxford University Press, 1999.
- [53] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Prentice Hall, 1992.
- [54] A. Gelb, *Applied Optimal Estimation*. The MIT Press, 1974.
- [55] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering, 3rd Edition*. Prentice Hall, 1996.
- [56] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [57] J. Montagnat, H. Delingette, and N. Ayache, "A review of deformable surfaces: topology, geometry and deformation," *Image and Vision Computing*, vol. 19, no. 14, pp. 1023 – 1040, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/B6V09-44B23HY-1/2/5f661092cd907748e3e1a253a2b86d2b>
- [58] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001. [Online]. Available: citeseer.ist.psu.edu/moeslund01survey.html
- [59] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: A survey," *Medical Imaging, IEEE Transactions on*, vol. 25, no. 8, pp. 987–1010, 2006.
- [60] A. Hammoude, "Endocardial border identification in two-dimensional echocardiographic images: review of methods," *Computerized Medical Imaging and Graphics*, vol. 22, no. 3, pp. 181 – 193, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/B6T5K-3TPDRVW-1/2/5d211a4678e29c476803dcdf16c4bc8b>
- [61] J. Duchon, "Splines minimizing rotation-invariant semi-norms in sobolev spaces," *Constructive Theory of Functions of Several Variables*, pp. 85–100, 1977. [Online]. Available: <http://dx.doi.org/10.1007/BFb0086566>

- [62] D. Terzopoulos and D. Metaxas, "Dynamic 3d models with local and global deformations: Deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703–714, 1991.
- [63] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti, "Left ventricular volume estimation for real-time three-dimensional echocardiography," *Medical Imaging, IEEE Transactions on*, vol. 21, pp. 1202–1208, 2002.
- [64] S. Beucher and F. Meyer, "The morphological approach of segmentation: the watershed transformation," *Mathematical Morphology in Image Processing*, pp. 433–481, 1992.
- [65] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, pp. 269–277, May 1995.
- [66] O. Gerard, A. Billon, J.-M. Rouet, M. Jacob, M. Fradkin, and C. Allouche, "Efficient model-based quantification of left ventricular function in 3-d echocardiography," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1059–1068, Sept. 2002.
- [67] H. Delingette, "General object reconstruction based on simplex meshes," *International Journal of Computer Vision*, vol. 32, no. 2, pp. 111–146, September 1999.
- [68] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. Eur. Conf. Computer Vision*, vol. 2. Springer-Verlag, 1998, pp. 484–498.
- [69] J. Bosch, S. Mitchell, B. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 11, pp. 1374–1383, Nov. 2002.
- [70] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. [Online]. Available: <http://www.sciencemag.org/cgi/content/abstract/290/5500/2319>
- [71] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, Jun. 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/B6W6Y-45JKKXN-6/2/0594c7d0323a96476b360f5da8b97044>
- [72] J. Hansegård, S. Urheim, K. Lunde, and S. I. Rabben, "Constrained active appearance models for segmentation of triplane echocardiograms," *IEEE Trans. Med. Imaging*, vol. 26, no. 10, pp. 1391–1400, 2007.
- [73] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan. 1988. [Online]. Available: <http://dx.doi.org/10.1007/BF00133570>

- [74] M.-P. Jolly, "Automatic segmentation of the left ventricle in cardiac mr and ct images," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 151–163, Nov. 2006. [Online]. Available: <http://dx.doi.org/10.1007/s11263-006-7936-3>
- [75] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 9, pp. 855–867, 1990.
- [76] P. Perez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, Mar 2004.
- [77] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, August 1998. [Online]. Available: citeseer.ist.psu.edu/isard98condensation.html
- [78] R. van der Merwe, A. Doucet, N. D. Freitas, and E. Wan, "The unscented particle filter," in *Proceedings of Neural and Information Processing Systems.*, 2000.
- [79] V. Behar, D. Adam, P. Lysyansky, and Z. Friedman, "Improving motion estimation by accounting for local image distortion," *Ultrasonics*, vol. 43, no. 1, pp. 57–65, Oct. 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/B6TW2-4BWV9WD-1/2/ea608bebdc0d9391fdde93314301b8c1>
- [80] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI81*, 1981, pp. 674–679. [Online]. Available: citeseer.ist.psu.edu/lucas81iterative.html
- [81] J. Meunier, "Tissue motion assessment from 3D echographic speckle tracking," *Physics in Medicine and Biology*, vol. 43, no. 5, pp. 1241–1254, 1998. [Online]. Available: <http://stacks.iop.org/0031-9155/43/1241>
- [82] A. Elen, D. Loeckx, H. F. Choi, H. Gao, P. Claus, F. Maes, P. Suetens, and J. D'hooge, "P4a-5 3D cardiac strain estimation using spatio-temporal elastic registration: In silico validation," *Ultrasonics Symposium, 2007. IEEE*, pp. 1945–1948, 28-31 Oct. 2007.

Part III

Publications

A Framework for Real-Time Left Ventricular Tracking in 3D+T Echocardiography, Using Nonlinear Deformable Contours and Kalman Filter Based Tracking

F Orderud

Norwegian University of Science and Technology (NTNU), Norway

Abstract

This paper presents a new framework for automatic real-time left ventricular (LV) tracking in 3D+T echocardiography. The framework enables usage of existing biomechanical deformation models for the heart, with nonlinear modes of deformation, combined with edge models for the endocardial boundary.

Tracking is performed in a sequential state estimation fashion, using an extended Kalman filter to recursively predict and update contour deformations in real-time. Contours are detected using normal-displacement measurements from points on the predicted contour, and are processed efficiently using an information-filter formulation of the Kalman filter.

Promising results are shown for LV-tracking using a truncated ellipsoid contour model, with deformation parameters for translation, orientation, scaling and bending in all three dimensions. The tracking framework automatically detects LV position initially, even in situations where it is partially outside the volume. It also successfully tracks the dominant motion and shape changes throughout the heart cycle in real-time. A collection of 21 3D echocardiography recordings of good quality demonstrates that the framework is capable of automatically identifying and tracking the left ventricle in 90% of the recordings without any user input.

1. Introduction

There is a clinical need for real-time monitoring of cardiac function during invasive procedures and intensive care. Real-time tracking of the left ventricle (LV) would hence be beneficial in such situations. The last few years, 3D echocardiography has been introduced. However, no method for real-time tracking or segmentation of such data is currently available.

Most tracking approaches in 2D echocardiography have been based on traditional deformable models as introduced by Kass [1], which facilitate free-form deformation. However, these methods tend to be too slow for real-time applications and also have to be initialized close to the LV boundaries. The problem can, however, be made tractable by restricting the allowable deformations to certain predefined modes. This both regularizes the problem to make tracking more robust, and allows for real-time implementations based on sequential state estimation.

This state estimation approach was first presented by Blake et al. in [2], [3] and [4], which used a Kalman filter to track B-spline models deformed by linear transforms within a model subspace referred to as *shape space*. Later, the framework was applied for real-time left ventricular tracking in long-axis 2D-echocardiography by Jacob et al. in [5], [6] and [7]. All these papers were using a B-spline representation, deformed by a trained linear principal component analysis (PCA) deformation model. The papers also discuss the possibility of extending the framework to 3D echocardiography, which is what has been done in this paper. The proposed framework also allows for nonlinear modes of deformations.

2. Methods

Contour model

The tracking framework is based on a contour deformation model. This model is a function $\mathcal{D}(\dots)$ that transforms points on a contour template \mathbf{p}_0 into deformed points \mathbf{p} using a state vector \mathbf{x} as parameter:

$$\mathbf{p} = \mathcal{D}(\mathbf{p}_0, \mathbf{x})$$

This parameterization puts very few restrictions on the allowable deformation, so virtually “any” deformation model can be used, including biomechanical models. One must, however, be able to derive all partial derivatives of the position as a function of the deformation parameters. Deformation of contour normals also requires calculation of the spatial derivatives [8]. This approach differs from the linear *shape space* deformations used by Blake and Jacob [2],[3],[4],[5], [6] and [7], where all deformations had to be linear in the state vector.

Pursuing the approach by Park et al. [9] we use a truncated ellipsoid as contour template in this paper, and allow for the following deformation parameters:

- Translation (t_x, t_y, t_z) .
- Scaling (s_x, s_y, s_z) .
- Rotation/orientation (r_x, r_y) .
- Bending/curving (c_x, c_y) .

In total, these parameters form the state vector below.

$$\mathbf{x} = [t_x \ t_y \ t_z \ s_x \ s_y \ s_z \ r_x \ r_y \ c_x \ c_y]^T$$

Kinematic model

To enable modeling of motion in addition to position, the state vector is augmented to contain the last two successive state estimates [3]. The kinematic model for the predicted state $\bar{\mathbf{x}}$ at timestep $k + 1$ then becomes:

$$\bar{\mathbf{x}}_{k+1} = \mathbf{A}_1 \hat{\mathbf{x}}_k + \mathbf{A}_2 \hat{\mathbf{x}}_{k-1} + \mathbf{B}_0 \mathbf{w}_k$$

Tuning of kinematic properties like damping, regularization and prediction uncertainty for all deformation parameters can now be accomplished by adjusting the coefficients in matrices \mathbf{A}_1 , \mathbf{A}_2 and \mathbf{B}_0 .

Edge measurements

Edge measurements are used to guide the contour towards the object being tracked. This is performed by measuring the distance between contour points and measured edges in normal direction, called *normal displacement* [4], for points along a predicted contour inferred from the measurement model.

The normal displacement between a predicted contour point \mathbf{p} with associated normal vector \mathbf{n} and a measured edge point \mathbf{p}_{obs} is:

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

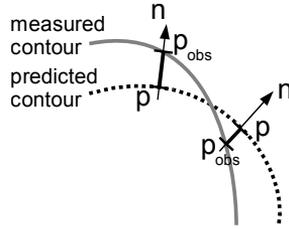


Figure 1. Normal displacement measurements along normal vectors of a predicted contour.

This inner-product form is dimensionally invariant, thus function just as good for measurements in 3D-data as 2D. The associated measurement noise r can either be constant for all edges, or dependent on edge-strength or other measure of uncertainty.

Edge model

The high levels of noise and speckle in ultrasound recordings makes edge detection difficult. Robust edge detectors that minimize the chance of detecting spurious edges in noisy areas are therefore desired.

An edge-model that exhibits robust characteristics is the *step model* [10]. 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 squared errors between the model and the data.

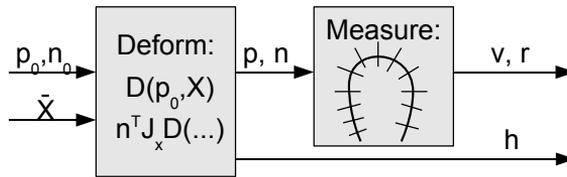


Figure 2. Overview over the contour deformation and edge measurement process. The figure shows how points on the initial contour p_0, n_0 are first deformed using a predicted state \bar{x} , yielding a deformed contour p, n and measurement vector h . Edges are then measured relative to the predicted contour, resulting in normal displacements v with associated measurement error variances r .

Measurement linearization

Normal displacement measurements can be used as measurement model in a *Kalman filter* model for the tracking problem. This is possible by linearizing the nonlinear deformation model around the predicted deformation state and using an *extended Kalman filter* [11] implementation. Altogether, this results in a measurement vector h that is based on the state-space Jacobian of the measurement model, meaning all partial derivatives of contour position with regard to all state dimensions, evaluated at the predicted state.

The linearized measurement vector then becomes the normal vector projection of the Jacobian ma-

trix:

$$\mathbf{h}^T \equiv \mathbf{n}^T \frac{\partial \mathcal{D}(\mathbf{p}_0, \mathbf{x})}{\partial \mathbf{x}}$$

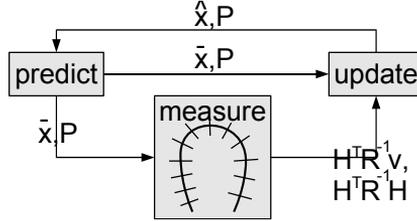


Figure 3. Overall framework structure.

Measurement processing

Assumption of independent measurements allows measurements to be summed together efficiently in *information space* [11], since independent measurements lead to diagonal measurement covariance matrices. All measurement information can then be summed into an information vector and matrix of dimensions invariant to the number of measurements:

$$\begin{aligned} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} &= \sum_i \mathbf{h}_i r_i^{-1} v_i \\ \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} &= \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T \end{aligned}$$

This is the same form as was used in [4].

Measurement update equations

Contour tracking forms a special problem structure, since the number of measurement typically far exceed the number of state dimensions. Ordinary Kalman gain calculation will then be computationally intractable, since they involve inverting matrices with dimensions equal to the number of measurements. An alternative approach, presented by Blake and Isard in [4], avoids this problem by altering the measurement update step in the Kalman filter. This is accomplished by utilizing that the Kalman gain $\mathbf{K}_k \equiv \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1}$, and reformulating to account for measurements on information filter [11] form. The updated state estimate $\hat{\mathbf{x}}$ for timestep k then becomes:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + \mathbf{K}_k \mathbf{v}_k \\ \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + \hat{\mathbf{P}}_k (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v}_k) \end{aligned}$$

Measurement innovations are here efficiently summed into a measurement vector with dimension equal to the state dimension.

The error covariance update equations can similarly be performed in information space to avoid inverting large matrices:

$$\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$$

This form only requires inversion of matrices with dimensions equal to the state dimension.

3. Results

A collection of 21 apical 3D-echocardiography recordings served as independent validation of the method. The same configuration were used for all recordings, with an initial LV contour automatically placed at a depth of approximately 80mm in the first frame as shown in figure 4. The tracking was then run for a couple of heartbeats to give the contour enough time to lock on to the LV.

The experiments were performed using an LV-contour consisting of 426 contour points. Edge detection was performed in the normal direction of each of these points, using 20 samples spaced 1mm apart. Real-time tracking in 25fps 3D echocardiography datasets yielded a modest CPU load of approximately 18%¹.

The results are summarized in table 1. One can see that the tracking was performed with subjectively *good* or *adequate* quality in 90% of the 21 recordings present in the dataset. Figure 5 and 6 shows examples of *good* tracking in an ordinary recording, and a recording with apex outside the volume. Figure 4 shows the initial contour used in all recordings, as well as the contour after tracking for a couple of heartbeats.

Quality	Count	Description
Good	16	Tracking performed well.
Adequate	3	Tracking with reduced accuracy.
Fair	1	Tracking with low accuracy.
Poor	1	Unable to automatically track

Table 1. Overall performance of the automatic tracking. Subjectively scored by the author.

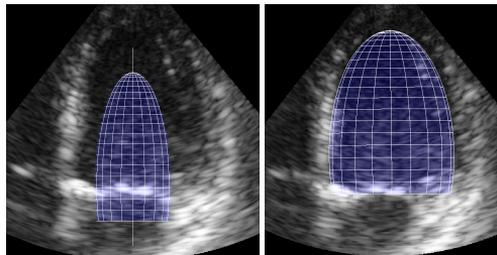


Figure 4. Azimuth view of the initial contour (left), and tracking results after the contour has locked on the LV after a couple of heartbeats (right).

4. Discussion and conclusions

A novel framework for real-time contour tracking in 3D echocardiography using sequential state estimation has been proposed. The framework builds upon previous work by Blake et al. [4], and enables tracking of of deformable contours with nonlinear modes of deformation. The feasibility of the framework has been demonstrated by automatic tracking in several recordings using a truncated

¹The tracking were then performed using a C++ implementation on a 3GHz Intel Pentium 4 processor. Visualization were disabled for CPU benchmarking.

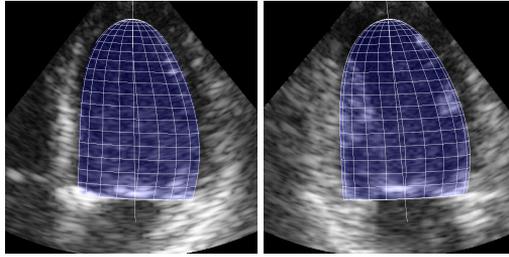


Figure 5. Azimuth and elevation view of a recording with *good* tracking.

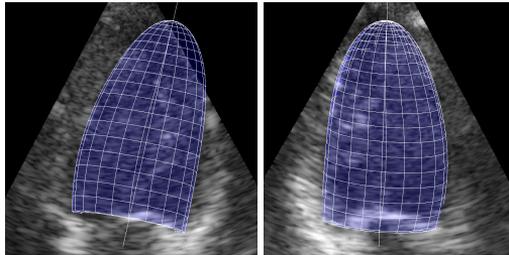


Figure 6. Azimuth and elevation view of tracking when LV is partially outside the volume.

ellipsoid model. The tracking framework was found to automatically detect LV position initially, even in situations where the LV is partially outside the acquired volume.

It can be argued that the evaluation procedure performed is too subjective and should have been performed by a medical clinician. However, the principal objective was not to get an accurate segmentation of the LV suitable for clinical diagnosis, but merely to demonstrate the ability to track the dominant motion and shape changes throughout the heart cycle. Further research will focus on perfecting the method, and striving towards tracking that is both robust and accurate.

Traditional free-form deformation models are not capable of operating in real-time in volumetric data. The proposed framework instead sacrifices accuracy for the capability of automatic real-time tracking within a limited shape space. This limited shape space also serves to regularize the problem, thus making tracking more robust.

The general deformation formulation also places few restrictions on the modes of deformations allowed. It is therefore believed that the truncated ellipsoid model can be replaced with a more realistic biomechanical model for the LV. This is likely to yield better model fitting to the data, and hence improve tracking accuracy.

Acknowledgements

I want to thank Stein Inge Rabben and Sigmund Frigstad at GE Vingmed Ultrasound for guidance and most helpful feedback, as well as Brage Amundsen at the Norwegian University of Science and Technology (NTNU) for access to the datasets used.

References

- [1] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. *International Journal of Computer Vision* 1988;1(4):321–331.
- [2] Blake A, Curwen R, Zisserman A. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision* 1993;11(2):127–145.
- [3] Blake A, Isard M, Reynard D. Learning to track the visual motion of contours. *Artificial Intelligence* 1995;78(1-2):179–212.
- [4] Blake A, Isard M. *Active Contours*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998. ISBN 3540762175.
- [5] Jacob G, Noble JA, Mulet-Parada M, Blake A. Evaluating a robust contour tracker on echocardiographic sequences. *Medical Image Analysis* 1999;3(1):63–75.
- [6] Jacob G, Noble JA, Kelion AD, Banning AP. Quantitative regional analysis of myocardial wall motion. *Ultrasound in Medicine Biology* 2001;27(6):773–784.
- [7] Jacob G, Noble JA, Behrenbruch CP, Kelion AD, Banning AP. A shape-space based approach to tracking myocardial borders and quantifying regional left ventricular function applied in echocardiography. *IEEE Transactions on Medical Imaging* 2002; 21(3):226–238.
- [8] Barr AH. Global and local deformations of solid primitives. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1984; 21–30.
- [9] Park J, Metaxas D, Young A, Axel L. Deformable models with parameter functions for cardiac motion analysis from tagged mri data. *IEEE Transactions on Medical Imaging* June 1996;15(3):290–298.
- [10] Rabben SI, Torp AH, Støylen A, Slørdahl S, Bjørnstad K, Haugen BO, Angelsen B. Semiautomatic contour detection in ultrasound M-mode images. *Ultrasound in Medicine Biology* 2000;26(2):287–296.
- [11] Bar-Shalom Y, Kirubarajan T, Li XR. *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2002. ISBN 0471221279.

Address for correspondence:

Fredrik Orderud

Department of Computer and Information Science / NTNU

Sem Sælands vei 7-9; NO-7491 Trondheim

fredrik.orderud@idi.ntnu.no

Real-time Tracking of the Left Ventricle in 3D Echocardiography Using a State Estimation Approach

Fredrik Orderud¹, Jøger Hansgård², and Stein I. Rabben³

¹ Norwegian University of Science and Technology, Norway
fredrik.orderud@idi.ntnu.no

² University of Oslo, Norway jogerh@ifi.uio.no

³ GE Vingmed Ultrasound, Norway stein.rabben@med.ge.com

Abstract. In this paper we present a framework for real-time tracking of deformable contours in volumetric datasets. The framework supports composite deformation models, controlled by parameters for contour shape in addition to global pose. Tracking is performed in a sequential state estimation fashion, using an extended Kalman filter, with measurement processing in information space to effectively predict and update contour deformations in real-time. A deformable B-spline surface coupled with a global pose transform is used to model shape changes of the left ventricle of the heart.

Successful tracking of global motion and local shape changes without user intervention is demonstrated on a dataset consisting of 21 3D echocardiography recordings. Real-time tracking using the proposed approach requires a modest CPU load of 13% on a modern computer. The segmented volumes compare to a semi-automatic segmentation tool with 95% limits of agreement in the interval 4.1 ± 24.6 ml ($r = 0.92$).

1 Introduction

The emergence of volumetric image acquisition within the field of medical imaging has attracted a lot of scientific interest over the last years. In a recent survey, Noble *et al.* [1] presented a review of the most significant attempts for automatic segmentation within the field of ultrasonics. However, all of these attempts are limited to being used as postprocessing tools, due to extensive processing requirements, even though volumetric acquisition may be performed in real-time with the latest generation of 3D ultrasound technology. Availability of technology for real-time tracking and segmentation in volumetric datasets would open up possibilities for instant feedback and diagnosis during data acquisition. Automatic tracking of the main chamber of the heart, the left ventricle (LV), would here serve as an excellent application.

Orderud [2] recently presented a tracking approach that allows for real-time tracking of rigid bodies in volumetric datasets. The approach treats the tracking problem as a state estimation problem, and uses an extended Kalman filter to

recursively track global pose parameters using a combination of state predictions and measurement updates. Experimental validation of LV tracking in 3D echocardiography, using a simple truncated ellipsoid model, was performed to demonstrate the feasibility of the approach.

This state estimation approach is based on prior work by Blake *et al.* [3, 4], who used a Kalman filter to track B-spline contours deformed by linear transforms within a model subspace referred to as *shape space*. Later, the same approach was applied to real-time LV tracking in 2D echocardiography by Jacob *et al.* [5–7]. All these papers did, however, lack a separation between global pose and local shape. They were also restricted to linear deformations, such as principal component analysis deformation models, and to tracking in 2D datasets.

This paper extends [2] to support contours with composite deformation models, consisting of both local shape changes and global pose. We also propose to use a 3D B-spline surface model to circumvent the limitations of rigid ellipsoidal models for LV tracking. This model is coupled with a global pose transform, and successful LV tracking in 3D echocardiography is demonstrated.

This model somewhat resembles the deformable superquadrics model [8], and a 3D active shape model [9], but is spline-based, and allows for free-form shape deformations [10] by letting the control points move independently perpendicular to the surface.

2 Deformable Contour

The tracking framework is based upon a contour deformation model \mathbf{T} , which is decomposed into local deformations and global transformations. Local shape deformations are used to alter contour shape, by deforming points on a shape template \mathbf{p}_0 into intermediate contour shape points \mathbf{p}_l , using a local shape deformation model \mathbf{T}_l with local state vector \mathbf{x}_l as parameters:

$$\mathbf{p}_l = \mathbf{T}_l(\mathbf{p}_0, \mathbf{x}_l) . \quad (1)$$

The intermediate contour shape points \mathbf{p}_l are subsequently transformed into final contour points \mathbf{p} , using a global pose transformation model \mathbf{T}_g with global state vector \mathbf{x}_g as parameters:

$$\mathbf{p} = \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g) . \quad (2)$$

A composite deformation model \mathbf{T} is then constructed by combining local and global deformations of shape template points into a joint model, as shown in Fig. 1. This yields a composite state vector $\mathbf{x} \equiv [\mathbf{x}_g^T, \mathbf{x}_l^T]^T$ consisting of N_g global and N_l local deformation parameters. Calculation and propagation of associated contour normals \mathbf{n} through \mathbf{T} must also be performed, since they are required to specify the search direction for the later edge-detection.

This separation between local and global deformations is intended to ease modeling, since changes in shape are often parameterized differently compared to deformations associated with global position, size and orientation. It also

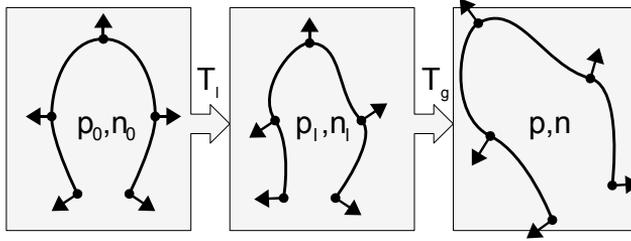


Fig. 1. Overview over the contour deformation and transformation process: The shape template $\mathbf{p}_0, \mathbf{n}_0$ is first deformed locally into $\mathbf{p}_l, \mathbf{n}_l$, followed by a global transformation into the final contour \mathbf{p}, \mathbf{n} .

puts very few restrictions on the allowable deformations, so a wide range of parameterized deformation models can be used, including nonlinear models, as opposed to previous *shape space* models that are limited to linear deformations.

3 Tracking Framework

The proposed tracking framework follows the processing chain of the Kalman filter, starting by predicting contour state using a motion model (section 3.1). The shape template is subsequently deformed to form a predicted contour, and associated normal vectors and state-space Jacobi matrices are computed (section 3.2). Edge detection is then performed locally along the normal vectors. Finally, all measurement information is assimilated in information space [11], and combined with the predicted contour to compute an updated contour state estimate (section 3.3). Figure 2 shows an overview over these steps.

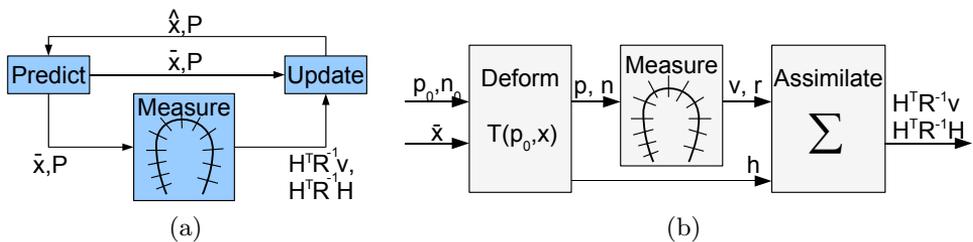


Fig. 2. (a) Overview of the tracking scheme. (b) Illustration of how points on the shape template $\mathbf{p}_0, \mathbf{n}_0$ are first deformed using a predicted state $\bar{\mathbf{x}}$, yielding a deformed contour \mathbf{p}, \mathbf{n} and measurement vector \mathbf{h} . Edges are then measured relative to the predicted contour, resulting in normal displacements v with associated measurement error variances r . The edge measurements are finally assimilated into an information vector and matrix for efficient processing.

3.1 Motion Model

Modeling of motion, in addition to position, is accomplished in the prediction stage of Kalman filter by augmenting the state vector to contain the last two successive state estimates. A *motion model* which predicts the state $\bar{\mathbf{x}}$ at timestep $k + 1$, is then expressed as:

$$\bar{\mathbf{x}}_{k+1} - \mathbf{x}_0 = \mathbf{A}_1(\hat{\mathbf{x}}_k - \mathbf{x}_0) + \mathbf{A}_2(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_0) , \quad (3)$$

where $\hat{\mathbf{x}}_k$ is the estimated state from timestep k . Tuning of properties like damping and regularization towards the mean state \mathbf{x}_0 for all deformation parameters can then be accomplished by adjusting the coefficients in matrices \mathbf{A}_1 and \mathbf{A}_2 . Prediction uncertainty can similarly be adjusted by manipulating the process noise covariance matrix \mathbf{B}_0 used in the associated covariance update equation. The latter will then restrict the change rate of parameter values.

3.2 Edge Measurements

Processing of edge measurements using an extended Kalman filter [12] requires *state-space* Jacobi matrices to relate changes in contour point positions to changes in contour state. Separate Jacobi matrices for each contour point must therefore be calculated. The choice of composite deformations leads to state-space Jacobi matrices consisting of two separate parts, namely of global and local derivatives:

$$\frac{\partial \mathbf{T}(\mathbf{p}_0, \mathbf{x})_i}{\partial \mathbf{x}} \equiv \left[\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)_i}{\partial \mathbf{x}_g}, \sum_{n \in x, y, z} \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)_i}{\partial \mathbf{p}_{l,n}} \frac{\partial \mathbf{T}_l(\mathbf{p}_0, \mathbf{x}_l)_n}{\partial \mathbf{x}_l} \right] . \quad (4)$$

The global Jacobian becomes a $3 \times N_g$ matrix, while the Jacobian for the local shape deformations becomes the product of a 3×3 matrix by a $3 \times N_l$ matrix using the chain rule for multivariate calculus.

Edge detection is performed by measuring the distance between points \mathbf{p}_i on a predicted contour inferred from the motion model, and edges $\mathbf{p}_{obs,i}$ detected by searching in normal direction \mathbf{n}_i of the contour surface. This type of edge detection is referred to as *normal displacement* [4], and is calculated as follows:

$$v_i = \mathbf{n}_i^T (\mathbf{p}_{obs,i} - \mathbf{p}_i) . \quad (5)$$

Each normal displacement measurement is coupled with a measurement noise value r_i that specifies the uncertainty associated with the edge, which may either be constant for all edges, or dependent on edge strength, or other measure of uncertainty.

Linearized measurement models [12], which are required in the Kalman filter for each edge measurement, are constructed by transforming the state-space Jacobi matrices the same way as the edge measurements, namely by projecting them onto the normal vector:

$$\mathbf{h}_i^T = \mathbf{n}_i^T \frac{\partial \mathbf{T}(\mathbf{p}_0, \mathbf{x})_i}{\partial \mathbf{x}} . \quad (6)$$

This yields a separate *measurement vector* \mathbf{h}_i for each normal displacement measurement, to relate normal displacements to changes in contour state.

3.3 Measurement Assimilation and State Update

Contour tracking forms a special problem structure, since the number of measurements far exceeds the number of state dimensions. Ordinary Kalman gain calculations then becomes intractable, since they involve inverting matrices of dimensions equal to the number of measurements. An alternative approach is to assimilate measurements in *information space* prior to the state update step. This enables very efficient processing if we assume that measurements are independent. All measurement information can then be summed into an information vector and matrix of dimensions invariant to the number of measurements:

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{h}_i r_i^{-1} v_i \quad (7)$$

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T . \quad (8)$$

Measurements in information filter form require some alterations to the state update step in the Kalman filter. This can be accomplished by using the information filter formula [11] for the updated state estimate $\hat{\mathbf{x}}$ at timestep k :

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1} \mathbf{v}_k . \quad (9)$$

The updated error covariance matrix $\hat{\mathbf{P}}$ can similarly be calculated in information space to avoid inverting matrices with dimensions larger than the state dimension:

$$\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} . \quad (10)$$

4 Experiments

We used a quadratic B-spline surface consisting of 24 control points arranged in a prolate spheroid grid as a LV model. Shape deformations were enabled by allowing the control points to move perpendicularly to the surface. This was combined with a global model for translation and scaling in three dimensions, as well as rotation around two of the axes. In total, this resulted in a deformation model consisting of 32 parameters. Edge detection was performed in the normal direction of approximately 450 contour points distributed evenly over the surface, using 30 samples spaced 1 mm apart. A simple edge detector based upon the transition criterion with variable height [13] is used to determine the position of the strongest edge along each normal. This detector assumes edges to form a transition in image intensity, from a dark cavity to a bright myocardium, and calculates the edge position that minimizes the sum of squared errors between a transition model and the data. Weak and outlier edges were automatically rejected to improve robustness.

4.1 Results

A collection of 21 apical 3D echocardiography recordings of adult patients, of which half were diagnosed with heart diseases, were used to validate the method.

The recordings were acquired with a Vivid 7 scanner (GE Vingmed Ultrasound, Norway), using sub-volume stitching to form a contiguous wide-angle recording of the LV. The exact same tracking configuration was used for all recordings, with an initial LV contour placed at a predefined position in the first frame. The tracking were then run for a couple of heartbeats to give the contour enough time to lock on to the LV. Tracking accuracy was evaluated by comparing the segmentations with the results from a custom-made semiautomatic segmentation tool (GE Vingmed Ultrasound), that facilitates manual editing. The reference segmentations were validated and, if needed, edited by an expert operator.

Figure 3 shows the overall volume correspondence throughout the cardiac cycle, for all frames in all of the 21 recordings. Bland-Altman analysis of the volume points in Fig. 3(a) yields a 4.1 ml bias, with 95% limits of agreement in the interval 4.1 ± 24.6 ml, and a strong correlation ($r = 0.92$). The average point to surface distance between the contour and the reference surfaces was 2.7 mm.

The volume curves were usually very similar in shape compared to the reference method. However, some per-recording bias was seen. This is illustrated in Fig. 3(b), where each volume curve is more or less parallel to the identity line, but have different offset. If we subtract the 'per-recording' bias from each volume curve, the limits of agreement is improved to ± 11.0 ml, indicating that much of the disagreement stems from the varying bias from recording to recording.

Figure 4 shows the intersection between the segmented contour and several slices throughout the volume in one of the recordings, as well as a plot of the volume of the segmented contour throughout the cardiac cycle.

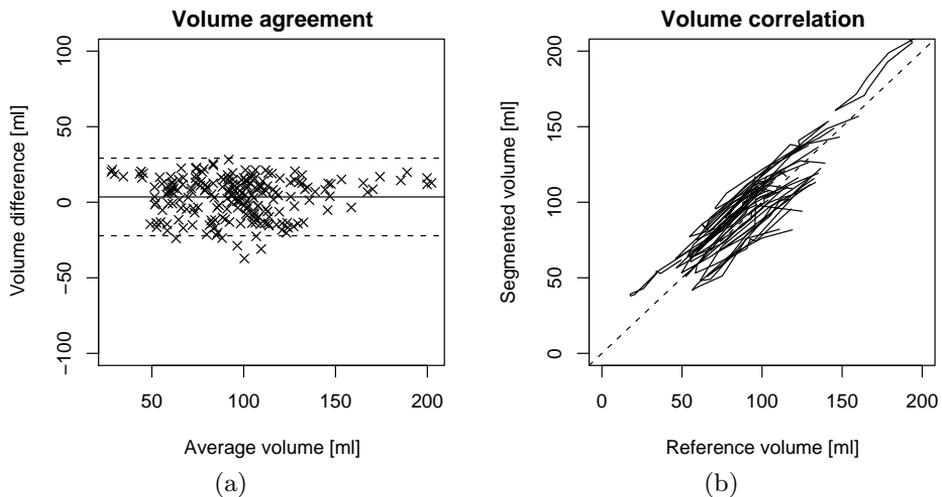


Fig. 3. Bland-Altman plot for the overall correspondence between automatically real-time segmented volumes and the reference (a), along with the mean volume difference (solid) and 95% limits of agreement (dashed). Associated volume correlation plot (b), where segmented volume curves from each recording is compared to the reference and shown as separate lines, along with the identity line (dashed).

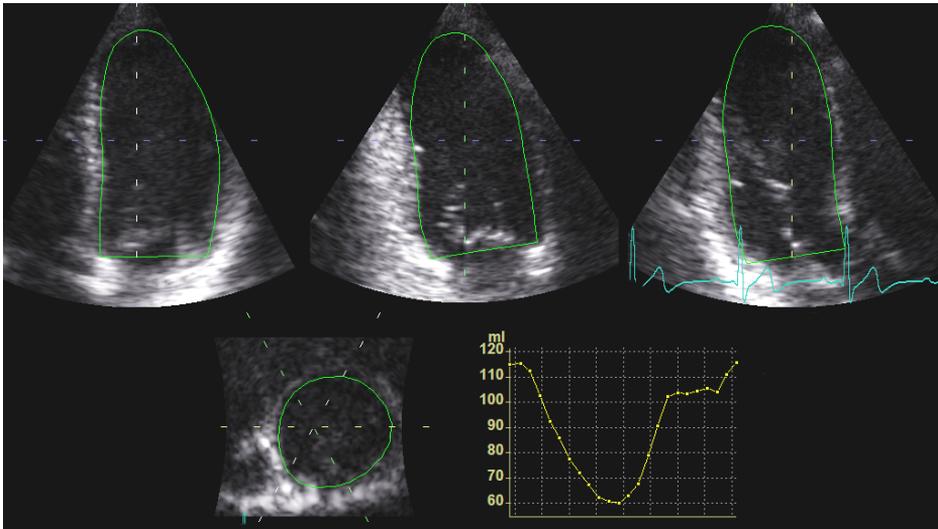


Fig. 4. Example of successful real-time tracking, showing intersections between the segmented contour and several slices through the volume, as well as a plot of the volume of the segmented contour for one cardiac cycle.

The CPU load required to maintain real-time segmentation at 25 frames per second (fps) was approximately 13% when visualization was disabled⁴.

5 Discussion and Conclusion

A novel framework for real-time tracking of deformable contours in volumetric datasets using a state estimation approach has been proposed. It extends prior work [2] by enabling tracking of contours with both local and global modes of deformation, both of which may be nonlinear. Compared to traditional deformable model based segmentation methods, the non-iterative Kalman filter algorithm leads to outstanding computational performance.

The feasibility of real-time LV tracking in 3D echocardiography has been demonstrated by successful tracking and segmentation in 21 recordings using a quadratic B-spline model coupled to a global pose transform. This represents a significant improvement over [2], that solely tracked chamber position, orientation and size, using an ellipsoid model, and were thus inherently unable of segmenting shape changes of the LV. The framework is well suited for rapid analysis of LV volumes and global function, since it operates without user interaction. Other applications, like patient monitoring, and automatic initialization of other methods are also possible.

Tracking accuracy is primarily limited by the difficulty of edge detection in echocardiography recordings, which suffers from inherently poor image quality.

⁴ The experiments were performed using a C++ implementation on a 2.16 GHz Intel Core 2 duo processor.

It is in fact difficult, even for experts, to accurately determine the endocardial border in such recordings. Replacement of the simplistic transition criterion with a more advanced edge-detector, specifically designed for the ultrasound modality, is therefore believed to give better tracking accuracy.

Acknowledgment: The authors would like to thank Brage Amundsen at the Norwegian University of Science and Technology for providing the 3D echocardiography datasets.

References

1. Noble, J.A., Boukerroui, D.: Ultrasound image segmentation: A survey. *Medical Imaging, IEEE Transactions on* **25**(8) (2006) 987–1010
2. Orderud, F.: A framework for real-time left ventricular tracking in 3D+T echocardiography, using nonlinear deformable contours and kalman filter based tracking. In: *Computers in Cardiology*. (2006)
3. Blake, A., Curwen, R., Zisserman, A.: A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision* **11**(2) (October 1993) 127–145
4. Blake, A., Isard, M.: *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1998)
5. Jacob, G., Alison Noble, J., Mulet-Parada, M., Blake, A.: Evaluating a robust contour tracker on echocardiographic sequences. *Medical Image Analysis* **3**(1) (March 1999) 63–75
6. Jacob, G., Noble, J.A., Kelion, A.D., Banning, A.P.: Quantitative regional analysis of myocardial wall motion. *Ultrasound in Medicine & Biology* **27**(6) (June 2001) 773–784
7. Jacob, G., Noble, J.A., Behrenbruch, C., Kelion, A.D., Banning, A.P.: A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography. *Medical Imaging, IEEE Transactions on* **21**(3) (2002) 226–238
8. Park, J., Metaxas, D., Young, A.A., Axel, L.: Deformable models with parameter functions for cardiac motion analysis from tagged MRI data. *Medical Imaging, IEEE Transactions on* **15**(3) (June 1996) 278–289
9. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - Their training and application. *Computer Vision and Image Understanding* **61**(1) (January 1995) 38–59
10. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid registration using free-form deformations: application to breast MR images. *Medical Imaging, IEEE Transactions on* **18**(8) (1999) 712–721
11. Comaniciu, D., Zhou, X.S., Krishnan, S.: Robust real-time myocardial border tracking for echocardiography: An information fusion approach. *Medical Imaging, IEEE Transactions on* **23**(7) (2004) 849–860
12. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience (2001)
13. Rabben, S.I., Torp, A.H., Støylen, A., Slørdahl, S., Bjørnstad, K., Haugen, B.O., Angelsen, B.: Semiautomatic contour detection in ultrasound M-mode images. *Ultrasound in Med. & Biol.* **26**(2) (Feb. 2000) 287–296

Real-Time Active Shape Models for Segmentation of 3D Cardiac Ultrasound

Jøger Hansgård¹, Fredrik Orderud², and Stein I. Rabben³

¹ University of Oslo, Norway, jogerh@ifi.uio.no

² Norwegian University of Science and Technology, Norway,
fredrik.orderud@idi.ntnu.no

³ GE Vingmed Ultrasound, Norway, stein.rabben@med.ge.com

Abstract. We present a fully automatic real-time algorithm for robust and accurate left ventricular segmentation in three-dimensional (3D) cardiac ultrasound. Segmentation is performed in a sequential state estimation fashion using an extended Kalman filter to recursively predict and update the parameters of a 3D Active Shape Model (ASM) in real-time. The ASM was trained by tracing the left ventricle in 31 patients, and provided a compact and physiological realistic shape space. The feasibility of the proposed algorithm was evaluated in 21 patients, and compared to manually verified segmentations from a custom-made semi-automatic segmentation algorithm. Successful segmentation was achieved in all cases. The limits of agreement (mean \pm 1.96SD) for the point-to-surface distance were 2.2 \pm 1.1 mm. For volumes, the correlation coefficient was 0.95 and the limits of agreement were 3.4 \pm 20 ml. Real-time segmentation of 25 frames per second was achieved with a CPU load of 22%.

1 Introduction

Left ventricular (LV) volumes and ejection fraction (EF) are among the most important parameters in diagnosis and prognosis of heart diseases. Recently, real-time three-dimensional (3D) echocardiography was introduced. Segmentation of the LV in 3D echocardiographic data has become feasible, but due to poor image quality, commercially available tools are based upon a semi-automatic approach [1,2]. Furthermore, most reported methods are using iterative and computationally expensive fitting schemes. These factors make real-time segmentation in 3D cardiac ultrasound challenging.

Prior work by Blake *et al.* [3,4] and Jacob *et al.* [5,6], have shown that a state estimation approach is well suited for real-time segmentation in 2D imagery. They used a Kalman filter, which requires only a single iteration, to track the parameters of a trained deformable model based on principal component analysis (PCA), also known as Active Shape Models (ASMs) [7]. ASMs can be trained on manually traced LV contours, resulting in a sub-space of physiologically probable shapes, effectively exploiting expert knowledge of the LV anatomy

and function. For segmentation of 3D cardiac data, Van Assen *et al.* [8] introduced the 3D ASM. However, there are to our knowledge no reports of real-time implementations of 3D ASMs.

Based on the work in [3,4,5,6], real-time LV segmentation of 3D cardiac ultrasound was recently introduced by Orderud [9]. He used an extended Kalman filter for robust tracking of a rigid ellipsoid LV model. Later this framework has been extended to use a flexible spline-based LV model coupled with a global pose transform to improve local segmentation accuracy [10]. However, expert knowledge of LV anatomy could not be modeled directly.

To utilize expert knowledge of LV anatomy during segmentation, we propose to use a 3D ASM for real-time segmentation of 3D echocardiograms, by extending the framework described in Orderud [9]. The 3D ASM, trained on LV shapes traced by an expert, gives a compact deformable model which is restricted to physiologically realistic shapes. This model is fitted to the target data in real-time using a Kalman filter. The feasibility of the algorithm is demonstrated in 21 patients, where we achieve real-time segmentation of the LV shape, and instantaneous measurements of LV volumes and EF.

2 Shape Model

A set of 496 triangulated LV training meshes were obtained from 31 patients using a custom-made segmentation tool (GE Vingmed Ultrasound, Norway). The training tool provides manual editing capabilities. When necessary, the user hence did manual editing of the segmentation to make it equivalent to manual tracing.

Building the ASM requires pair-wise point correspondence between shapes from different patients [7,8]. We developed a reparametrization algorithm for converting triangulated LV training shapes into quadrilateral meshes. This algorithm produced meshes with 15 longitudinal and 20 circumferential segments, with vertices approximately identifying unique anatomical positions. The meshes were aligned separately to remove trivial pose variations, such as scaling, translation and rotation.

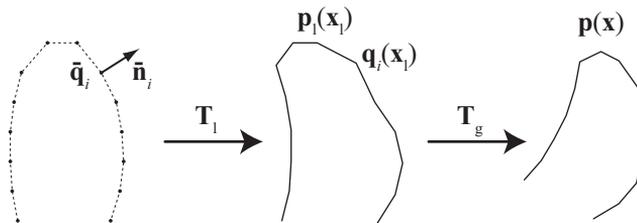


Fig. 1. A point $p(x)$ on the ASM is generated by first applying a local transformation T_l described by the ASM state vector x_1 on the mean shape, followed by a global pose transformation T_g to obtain a shape in real-world coordinates.

From the aligned training set, the mean vertex position $\bar{\mathbf{q}}_i$ was computed, and PCA was applied on the vertex distribution to obtain the N_x most dominant eigenvectors. In normalized coordinates, the ASM can be written on the form

$$\mathbf{q}_i(\mathbf{x}_1) = \bar{\mathbf{q}}_i + \mathbf{A}_i \mathbf{x}_1 , \quad (1)$$

where the position of a vertex \mathbf{q}_i is expressed as a linear combination of the associated subspace of the N_x most dominating eigenvectors combined into the $3 \times N_x$ deformation matrix \mathbf{A}_i . Here, \mathbf{x}_1 is the local state vector of the ASM. The expression for the ASM can be optimized assuming that the deformation at vertex $\bar{\mathbf{q}}_i$ is primarily directed along the corresponding surface normal $\bar{\mathbf{n}}_i$ of the average mesh. This is done by projecting the deformation matrix \mathbf{A}_i onto the surface normal, giving an N_x -dimensional vector of projected deformation modes $\mathbf{A}_i^\perp = \bar{\mathbf{n}}_i^T \mathbf{A}_i$. The optimized expression for the ASM can now be written on the form

$$\mathbf{q}_i(\mathbf{x}_1) = \bar{\mathbf{q}}_i + \bar{\mathbf{n}}_i (\mathbf{A}_i^\perp \mathbf{x}_1) , \quad (2)$$

reducing the number of multiplications by a factor of three.

Due to the quadrilateral mesh structure of the ASM, a continuous surface is obtained using a linear tensor product spline interpolant. An arbitrary point on the ASM in normalized coordinates can be expressed as $\mathbf{p}_1(\mathbf{x}_1) = \mathbf{T}_1|_{(u,v)}$ where (u, v) represents the parametric position on the surface, and the local transformation \mathbf{T}_1 includes the deformation and interpolation applied to the mean mesh. By coupling this model with a global pose transformation \mathbf{T}_g with parameters \mathbf{x}_g including translation, rotation, and scaling, we obtain a surface

$$\mathbf{p}(\mathbf{x}) = \mathbf{T}_g(\mathbf{p}_1(\mathbf{x}_1), \mathbf{x}_g) \quad (3)$$

in real-world coordinates, with a composite state vector $\mathbf{x}^T \equiv [\mathbf{x}_g^T, \mathbf{x}_1^T]$. An illustration showing the steps required to generate the ASM is shown in Fig. 1. In our experiments we used 20 eigenvectors, describing 98% of the total variation within the training set.

3 Tracking Algorithm

The tracking algorithm extends prior work by Orderud [9,10], to enable usage of 3D ASMs in the Kalman filter for real-time segmentation. This is accomplished by using the ASM shape parameters \mathbf{x}_1 directly, in addition to the global pose parameters \mathbf{x}_g , in the Kalman filter state vector.

3.1 Motion Model

Modeling of motion in addition to position can be accomplished in the prediction stage of the Kalman filter by augmenting the state vector to contain the last two

successive state estimates. A *motion model* which predicts the state $\bar{\mathbf{x}}$ at timestep $k + 1$, is then expressed as

$$\bar{\mathbf{x}}_{k+1} - \mathbf{x}_0 = \mathbf{A}_1(\hat{\mathbf{x}}_k - \mathbf{x}_0) + \mathbf{A}_2(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_0) , \quad (4)$$

where $\hat{\mathbf{x}}_k$ is the estimated state from timestep k . Tuning of properties, like damping and regularization towards the mean state \mathbf{x}_0 for all deformation parameters, can then be accomplished by adjusting the coefficients in matrices \mathbf{A}_1 and \mathbf{A}_2 . Prediction uncertainty can similarly be adjusted by manipulating the process noise covariance matrix \mathbf{B}_0 used in the associated covariance update equation. The latter will then restrict the change rate of parameter values.

3.2 Measurement Processing

Edge-detection is based on *normal displacement* measurements v_i [4], which are calculated by measuring the radial distance between detected edge-points $\mathbf{p}_{\text{obs},i}$ and the contour surface \mathbf{p}_i along selected search normals \mathbf{n}_i . These displacements are coupled with associated measurement noise r_i to weight the importance of each edge, based on a measure of edge confidence. Measurement vectors are calculated by taking the normal projection of the composite state-space Jacobian for the contour points

$$\mathbf{h}_i^T = \mathbf{n}_i^T \left[\frac{\partial \mathbf{T}_g(\mathbf{p}_1, \mathbf{x}_g)_i}{\partial \mathbf{x}_g} \quad \frac{\partial \mathbf{T}_g(\mathbf{p}_1, \mathbf{x}_g)_i}{\partial \mathbf{x}_1} \right] , \quad (5)$$

which is the concatenation of a global and a local state-space Jacobi matrix. The global Jacobian is trivially the state-space derivative of the global pose transformation, while the local Jacobian has to be derived, using the chain-rule for multivariate calculus, to propagate surface points on the spline through mesh vertices, and finally to the ASM shape parameters:

$$\frac{\partial \mathbf{T}_g(\mathbf{p}_1, \mathbf{x}_g)}{\partial \mathbf{x}_1} = \sum_{n \in x,y,z} \frac{\partial \mathbf{T}_g(\mathbf{p}_1, \mathbf{x}_g)}{\partial \mathbf{p}_{1,n}} \sum_{j \in 1..N_q} \left(\frac{\partial \mathbf{T}_1(\mathbf{x}_1)_n}{\partial \mathbf{q}_j} \cdot \bar{\mathbf{n}}_k \right) \mathbf{A}_j^\perp . \quad (6)$$

Here, $\partial \mathbf{T}_g(\mathbf{p}_1, \mathbf{x}_g) / \partial \mathbf{p}_1$ is the spatial derivative of the global transformation, and $\partial \mathbf{T}_1(\mathbf{x}_1) / \partial \mathbf{q}_j$ is the spatial mesh vertex derivative of the spline interpolant.

3.3 Measurement Assimilation and State Update

All measurements are assimilated in *information space* prior to the state update step. Assumption of independent measurements leads to very efficient processing, allowing summation of all measurement information into an information vector and matrix of dimensions invariant to the number of measurements:

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{h}_i r_i^{-1} v_i \quad (7)$$

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T . \quad (8)$$

The updated state estimate $\hat{\mathbf{x}}$ at timestep k can then be computed by using the information filter formula for measurement update [11], and the updated error covariance matrix $\hat{\mathbf{P}}$ is calculated directly in information space:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} \quad (9)$$

$$\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} . \quad (10)$$

Using this form, we avoid inversion of matrices with dimensions larger than the state dimension.

4 Evaluation

4.1 Data Material

For evaluation of the proposed algorithm, apical 3D echocardiograms of one cardiac cycle from 21 adult patients (11 diagnosed with heart disease) were recorded using a Vivid 7 scanner (GE Vingmed Ultrasound, Norway) with a 3D transducer (3V). In all patients, meshes corresponding to the endocardial boundary were determined using a custom-made semi-automatic segmentation tool (GE Vingmed Ultrasound, Norway). The segmentations were, if needed, manually adjusted by an expert to serve as independent references equivalent to manual tracing.

4.2 Experimental Setup and Analysis

Edge measurements were done perpendicular to the mesh surface within a distance of ± 1.5 cm to the surface at approximately 450 locations, using a simple edge model based on the transition criterion [12]. The ASM was initialized to the mean shape, and positioned in the middle of the volume in the first frame. Segmentation was performed on the evaluation set by running the algorithm for a couple of heartbeats, to give the ASM enough time to lock on to the LV.

The accuracy of the ASM was assessed using the mean of absolute point-to-mesh distances between the ASM and the reference, averaged over one cardiac cycle. Volume differences (bias) between the ASM and the reference were calculated for each frame. End-diastolic volume (EDV), end-systolic volume (ESV), and EF $((\text{EDV} - \text{ESV})/\text{EDV} \cdot 100\%)$ were compared to the manually verified reference (two-tailed t-test assuming zero difference), with 95% limits of agreement (1.96 standard deviations (SD)). EDV and ESV were computed as the maximum and minimum volume within the cardiac cycle respectively.

5 Results

We observed that common challenges with 3D cardiac ultrasound, such as drop-outs, shadows, and speckle noise were handled remarkably well, and segmentation was successful in all of the 21 patients. Some examples are shown in Fig. 2(b-d).

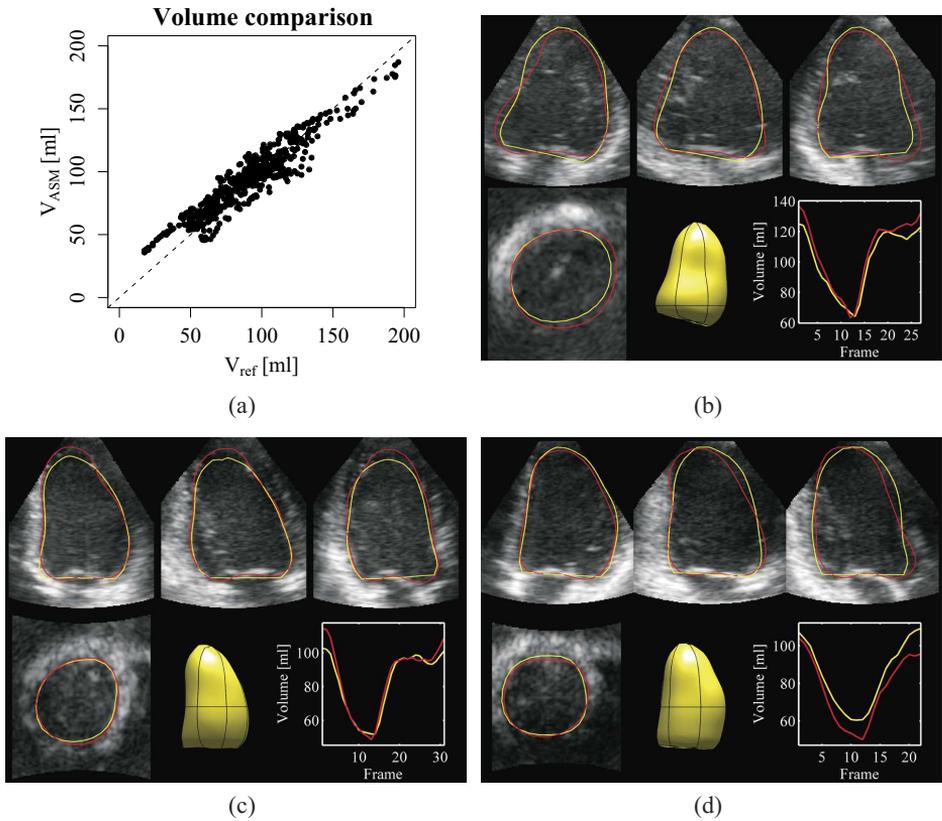


Fig. 2. LV volumes obtained by the ASM (V_{ASM}) in all 21 patients is compared to the reference (V_{ref}) and shown with the identity line (dashed) in (a). In (b-d), the end-diastolic segmentation (yellow) in three patients is compared to the reference (red) and shown along with the volume curve for one cardiac cycle.

The limits of agreement ($\text{mean} \pm 1.96\text{SD}$) for the point-to-surface distance were 2.2 ± 1.1 mm, indicating good overall agreement between the ASM and the reference. From Tab. 1, column 2, we see that the limits of agreement for volumes

Table 1. Segmentation results showing ASM versus reference.

	Volume [ml]	EDV [ml]	ESV [ml]	EF [%]
Difference ($\text{mean} \pm 1.96\text{SD}$)	$3.4^* \pm 20$	$-5.9^* \pm 21$	$6.2^* \pm 19$	$-7.7^* \pm 12$
Correlation coeff. (r)	0.95	0.91	0.91	0.74

* Significantly different from 0, $p < 0.05$.

were 3.4 ± 20 ml, with a strong correlation ($r=0.95$). The volume correspondence between the ASM and the reference is shown in Fig. 2(a).

We also found a strong EDV correlation ($r=0.91$), with a bias and 95% limits of agreement of -5.9 ± 21 ml (Tab. 1, column 3). The correlation in ESV was 0.91, with limits of agreement of 6.2 ± 19 ml, (Tab. 1, column 4), while the correlation in EF was 0.74, with limits of agreement of $-7.7 \pm 12\%$ (Tab. 1, column 5).

The CPU load required to maintain real-time segmentation at 25 frames per second (fps) was approximately 22% on a 2.16 GHz Intel Core 2 Duo processor.

6 Discussion

We have presented a fully automatic real-time algorithm for robust and accurate LV segmentation in 3D cardiac ultrasound. This was achieved by combining a 3D ASM with a Kalman filter based tracking algorithm. The feasibility of the algorithm was demonstrated in 21 patients.

Computational performance was excellent with a CPU load of 22% at 25 fps. Compared to traditional ASM update schemes, the Kalman filter gives good segmentation in a single iteration, allowing real-time implementations.

Contours detected by the ASM showed good overall agreement with the reference shapes, both with respect to point-to-mesh distances and volumes. There was a significant bias in estimated EDV, ESV, and EF, but with relatively narrow 95% limits of agreement. We speculate if the bias is primarily caused by the simple edge detector used, and better results are expected using a more advanced edge detector. Robustness was high in the evaluation set, with successful segmentation in all patients.

Since no user-interaction is required, the algorithm provides rapid analysis of LV function, and it can potentially provide higher reproducibility than semi-automatic methods. Adding means of manual corrections when segmentation fails will be subject for further studies.

The algorithm was evaluated on a population with varying image quality, but for evaluation of clinical applicability, the algorithm must be tested on a larger population. Also, an inherent challenge when using ASMs for clinical applications, is that care must be taken when assembling the training set to ensure inclusion of a sufficiently wide range of pathologies.

Traditional applications where our algorithm fits well includes rapid analysis of LV volumes, EF, and regional function. With real-time segmentation, we expect new applications to emerge, such as patient monitoring, and automated operator guidance.

7 Conclusion

We have developed a fully automatic algorithm for real-time segmentation of the left ventricle in 3D cardiac ultrasound. Initial evaluation in 21 patients is very promising, suggesting that this method is applicable in a clinical setting.

Acknowledgment: The authors would like to thank Brage Amundsen at the Norwegian University of Science and Technology (NTNU) for providing the 3D echocardiography datasets.

References

1. Jacobs, L.D., Salgo, I.S., Goonewardena, S., Weinert, L., Coon, P., Bardo, D., Gerard, O., Allain, P., Zamorano, J.L., de Isla, L.P., Mor-Avi, V., Lang, R.M.: Rapid online quantification of left ventricular volume from real-time three-dimensional echocardiographic data. *European Heart Journal* **27** (November 2006) 460–468
2. Sugeng, L., Mor-Avi, V., Weinert, L., Niel, J., Ebner, C., Steringer-Mascherbauer, R., Schmidt, F., Galuschky, C., Schummers, G., Lang, R.M., Nesser, H.J.: Quantitative assessment of left ventricular size and function. Side-by-side comparison of real-time three-dimensional echocardiography and computed tomography with magnetic resonance reference. *Circulation* **114** (August 2006) 654–661
3. Blake, A., Curwen, R., Zisserman, A.: A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision* **11**(2) (October 1993) 127–145
4. Blake, A., Isard, M.: *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1998)
5. Jacob, G., Noble, J.A., Kelion, A.D., Banning, A.P.: Quantitative regional analysis of myocardial wall motion. *Ultrasound in Medicine & Biology* **27**(6) (June 2001) 773–784
6. Jacob, G., Alison Noble, J., Mulet-Parada, M., Blake, A.: Evaluating a robust contour tracker on echocardiographic sequences. *Medical Image Analysis* **3**(1) (March 1999) 63–75
7. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - Their training and application. *Computer Vision and Image Understanding* **61**(1) (January 1995) 38–59
8. van Assen, H.C., Danilouchkine, M.G., Behloul, F., Lamb, H.J., van der Geest, R., Reiber, J.H.C., Lelieveldt, B.P.F.: Cardiac LV segmentation using a 3D active shape model driven by fuzzy inference. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. Volume 2878 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg (2003) 533–540
9. Orderud, F.: A framework for real-time left ventricular tracking in 3D+T echocardiography, using nonlinear deformable contours and kalman filter based tracking. In: *Computers in Cardiology*. (2006)
10. Orderud, F., Hansegård, J., Rabben, S.I.: Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach. Submitted to: *Medical Image Computing and Computer-Assisted Intervention - MICCAI* (2007)
11. Comaniciu, D., Zhou, X.S., Krishnan, S.: Robust real-time myocardial border tracking for echocardiography: An information fusion approach. *Medical Imaging, IEEE Transactions on* **23**(7) (2004) 849–860
12. Rabben, S.I., Torp, A.H., Støylen, A., Slørdahl, S., Bjørnstad, K., Haugen, B.O., Angelsen, B.: Semiautomatic contour detection in ultrasound M-mode images. *Ultrasound in Med. & Biol.* **26**(2) (Feb. 2000) 287–296

Real-time 3D Segmentation of the Left Ventricle Using Deformable Subdivision Surfaces

Fredrik Orderud

Norwegian University of Science and Technology
Trondheim, Norway

fredrik.orderud@idi.ntnu.no

Stein Inge Rabben

GE Vingmed Ultrasound
Oslo, Norway

stein.rabben@med.ge.com

Abstract

In this paper, we extend a computationally efficient framework for real-time 3D tracking and segmentation to support deformable subdivision surfaces. Segmentation is performed in a sequential state-estimation fashion, using an extended Kalman filter to estimate shape and pose parameters for the subdivision surface. As an example, we have integrated Doo-Sabin subdivision surfaces into the framework. Furthermore, we provide a method for evaluating basis functions for Doo-Sabin surfaces at arbitrary parameter values. These basis functions are precomputed during initialization, and later used during segmentation to quickly evaluate surface points used for edge detection.

Fully automatic tracking and segmentation of the left ventricle is demonstrated in a dataset of 21 3D echocardiography recordings. Successful segmentation was achieved in all cases, with limits of agreement ($\text{mean} \pm 1.96\text{SD}$) for point to surface distance of 2.2 ± 0.8 mm compared to manually verified segmentations. Real-time segmentation at a rate of 25 frames per second consumed a CPU load of 8%.

1. Introduction

The emergence of volumetric image acquisition within the field of medical imaging has attracted a lot of scientific interest over the last years. In a recent survey, Noble et al. [1] presented a review of the most significant attempts for 3D segmentation within the field of ultrasonics. However, all of these attempts are limited to being used as post processing tools, due to extensive processing requirements, even though volumetric acquisition may be performed in real-time with the latest generation of 3D ultrasound technology. Availability of technology for real-time tracking and segmentation in volumetric data sets would open up possibilities for instant feedback and diagnosis during data acquisition.

Orderud et al. has recently presented a framework for real-time tracking and segmentation in volumetric data [2]. This framework treats the tracking problem as a state estimation problem, and uses an extended Kalman filter to recursively track global pose and local shape parameters using a combination of state predictions and measurement updates. In [2], a deformable spline model was used to track left ventricular (LV) shape segmentation 3D echocardiography. Later, in [3] the framework was combined with a trained active shape model with predefined deformation modes to improve segmentation accuracy.

This state estimation approach is based on prior work by Blake et al. [4] in 2D, who used a Kalman filter to track B-spline contours deformed by linear transforms within a model subspace referred to as shape space. Later, the same approach was applied to real-time LV tracking in 2D echocardiography by Jacob et al. [5]. Similar efforts have later been published by Comaniciu et al. [6], who focused on the information fusion problem encountered in state-space tracking. A state-based approach for cardiac deformation analysis has also recently been published by Liu & Shi in [7].

Usage of spline models for shape segmentation does, however, imply some inherent topological restrictions, since the control vertices of a spline surface are restricted to form a regular quadrilateral structure. The LV model in [2] were for instance based on a cylindrical topology, with a hole at both the apex and base that required ad-hoc steps to form a closed surface. Polygonal models coupled with subspace deformation [3] does not suffer from any of these topological restrictions, but instead requires much higher mesh resolution in order to form smooth surfaces, which implies higher computational complexity and a more complex surface description.

1.1. Contributions

In this paper, we extend the Kalman tracking framework from [2] to support a wider class of smooth deformable surfaces known as subdivision surfaces [8, 9], that generalize spline surfaces to support meshes of arbitrary topology. This allows for more flexible modeling of arbitrary mesh structures, without the inherent topological restrictions associated with spline surfaces. We focus on a specific type of subdivision surfaces known as Doo-Sabin surfaces, which have some properties that make them suitable for low resolution cardiac modeling. Details for exact evaluation of surface points, as well as partial derivatives, for Doo-Sabin subdivision surfaces is also presented, to provide a means of evaluating basis functions for arbitrary surface points required for the edge detection process.

A low resolution Doo-Sabin subdivision model is then used to model the left ventricle of the heart. This model has adjustable control vertices to allow alteration of the shape, and is coupled with a global pose transform to position and orient it within the volume. Successful real-time LV segmentation in 3D echocardiography is finally demonstrated using the proposed subdivision model in conjunction to the Kalman filter based tracking framework.

1.2. Nomenclature

Scalars are expressed in italic, vectors in boldface and matrices in uppercase boldface. Control vertices are denoted ' \mathbf{q} ', displacement directions ' \mathbf{d} ', surface points ' \mathbf{p} ' and surface normal vectors ' \mathbf{n} '. State vectors are denoted ' \mathbf{x} '. Discrete time is denoted with subscript ' k ' for the Kalman filter, and control vertex or surface point indices are denoted with subscript ' i '.

2. Evaluation of Doo-Sabin Surfaces

Doo-Sabin surfaces [8] is a type of subdivision surface that generalize bi-quadratic B-spline surfaces to arbitrary topology. Following the same approach as Stam for Catmull-Clark surfaces [10], we define Doo-Sabin subdivision as a matrix operation. Each surface patch can be subdivided into four new sub-patches by multiplying the $N_q \times 3$ control vertex matrix \mathbf{Q}_0 with the $(N_q + 7) \times N_q$ subdivision matrix \mathbf{S} , as is shown in Fig. 1. The content of this matrix originates from the regular Doo-Sabin subdivision rules, which are outlined in appendix A. Control vertices for the region of support for each sub-patch $k \in \{0, 1, 2, 3\}$ of choice can then be extracted from the subdivided control vertices using a picking

matrix \mathbf{P}_k , such that $\mathbf{Q}_{n+1,k} = \mathbf{P}_k \mathbf{S} \mathbf{Q}_n$.

Regardless of the topology of \mathbf{Q}_n , all sub-patches $\mathbf{Q}_{n+1,k}$ will at most consist of a single irregular face in addition to three quadrilaterals. Successive subdivision operations on $\mathbf{Q}_{n+1,k}$ will then yield a single irregular patch, while the three others becomes regular bi-quadric spline patches that can be evaluated directly. By assuming, without loss of generality, that the irregular face in \mathbf{Q}_{n+1} is located top-left, then the picking matrix \mathbf{P}_k gives an regular 3×3 bi-quadric control vertex mesh when $k \neq 0$, and a irregular mesh consisting of N_q control vertices when $k = 0$. This relation can be exploited by performing repeated subdivisions n times until the desired surface point is no longer within an extraordinary patch ($k \neq 0$). Denoting $\mathbf{S}_0 = \mathbf{P}_0 \mathbf{S}$, we can express this as $\mathbf{Q}_{n,k} = \mathbf{P}_k \mathbf{S} \mathbf{S}_0^{n-1} \mathbf{Q}_0$.

The number of subdivision steps n required depends on the logarithm of (u, v) , while the sub-patch to pick after the final subdivision is determined using the following criterions:

$$n = \lfloor -\log_2(\max\{u, v\}) \rfloor \quad (1)$$

$$k = \begin{cases} 1 & \text{if } 2^n u > 1/2 \text{ and } 2^n v < 1/2 \\ 2 & \text{if } 2^n u > 1/2 \text{ and } 2^n v > 1/2 \\ 3 & \text{if } 2^n u < 1/2 \text{ and } 2^n v > 1/2 \end{cases} \quad (2)$$

Direct evaluation of surface points can then be performed for any patch location (u, v) except $(0, 0)$, by subdividing sufficient number of times, until the new subdivided patch below (u, v) no longer contains a extraordinary face, and treating the resulting sub-patch as a ordinary bi-quadric spline surface. For locations near $(0, 0)$, an approximate surface evaluation can be obtained by perturbing (u, v) slightly to prevent n from growing beyond a predefined upper limit. Basis functions with regards to the original non-subdivided control vertices can similarly be computed using the same approach:

$$\mathbf{b}(u, v)|_{\Omega_k^n} = (\mathbf{P}_k \mathbf{S} \mathbf{S}_0^{n-1})^T \tilde{\mathbf{b}}(\mathbf{t}_{k,n}(u, v)), \quad (3)$$

where Ω_k^n is the subdivision mapping function described above, that determines the number of subdivision steps required based on (u, v) [10]. $\tilde{\mathbf{b}}$ is the regular bi-quadric B-spline basis functions defined in appendix B, and $\mathbf{t}_{k,n}$ is a domain mapping function used to map the parametric interval (u, v) to the parametric interval within the desired sub-patch:

$$\mathbf{t}_{k,n}(u, v) = \begin{cases} (2^{n+1}u - 1, 2^{n+1}v) & \text{if } k = 1 \\ (2^{n+1}u - 1, 2^{n+1}v - 1) & \text{if } k = 2 \\ (2^{n+1}u, 2^{n+1}v - 1) & \text{if } k = 3 \end{cases} \quad (4)$$

Partial derivatives of the basis functions, \mathbf{b}_u and \mathbf{b}_v , are similarly computed by replacing $\tilde{\mathbf{b}}(u, v)$ with the respective derivatives of the B-spline basis functions in the formula. Surface positions can then be evaluated as an inner product between the control vertices and the basis functions

$$\mathbf{p}(u, v) = \mathbf{Q}_0^T \mathbf{b}(u, v). \quad (5)$$

Note that this approach is not dependent on diagonalization of the subdivision matrix, as in [10]. Instead, repeated matrix multiplication performed n times will result in exactly the same result. The associated increase in computational complexity associated with this repeated multiplication will not be a burden if evaluation of basis functions is performed only once, and later re-used to compute surface points regardless of movement of the associated control vertices.

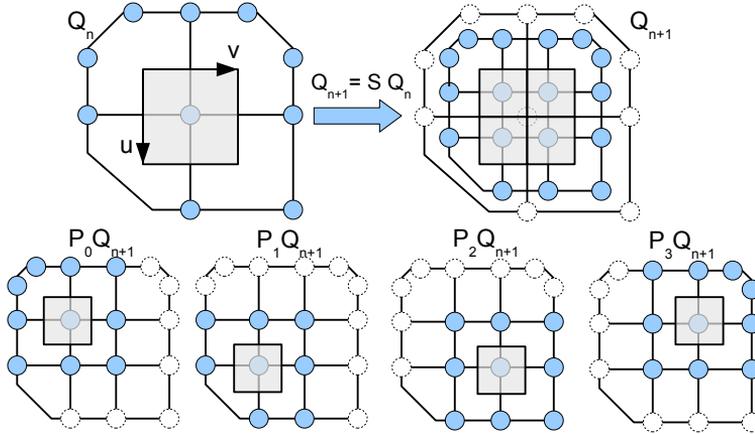


Figure 1. Illustration of the Doo-Sabin subdivision process. The control vertices \mathbf{Q}_n that define the initial surface patch (upper left) are subdivided into new control vertices \mathbf{Q}_{n+1} (upper right) by multiplying \mathbf{Q}_n with the subdivision matrix \mathbf{S} . Application of the picking matrix \mathbf{P}_k on \mathbf{Q}_{n+1} further divides the subdivided mesh into four sub-patches that together span the same surface area as the original patch.

3. Deformable Subdivision Model

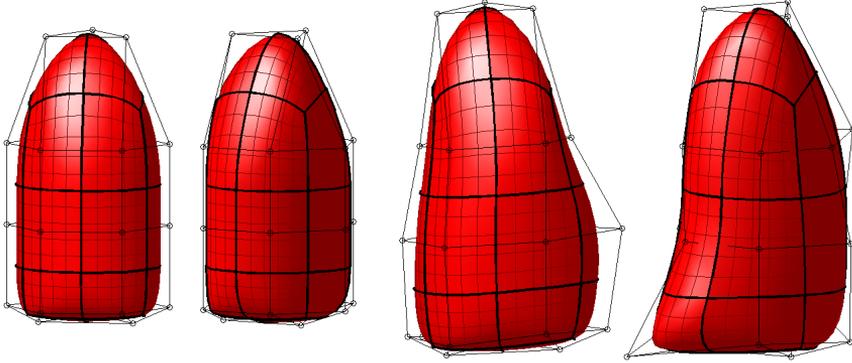
This section explains how deformable subdivision models, such as the LV model shown in Fig 2, can be incorporated into the Kalman tracking framework. The subdivision models consists of control vertices \mathbf{q}_i for $i \in \{1 \dots N_q\}$ with associated displacement direction vectors \mathbf{d}_i that defines the direction in which the control vertices are allowed to move. Displacement directions are typically based on surface normals, since movement of control vertices in this direction results in the greatest change of shape. In addition to the control vertices, the topological relationships between the control vertices have to be defined in a list $C(c)$. This list maps surface patches $c \in \{1 \dots N_c\}$ to enumerated lists of control vertex indices that define the region of support for each surface patch.

We denote the local deformations $\mathbf{T}_l(\mathbf{x}_l)$ to our deformable model as the deformations obtainable by moving the control vertices of the subdivision model. These local deformations are combine with a global transform $\mathbf{T}_g(\mathbf{x}_g, \mathbf{p}_l)$ to position, scale and orient the model within the image volume where the tracking takes place.

After creation of the model, a set of surface points have to be defined, which are to be used for edge detection measurements. This set consists of parametric coordinates (including patch number) for each of the points $(u, v, c)_l$, and are typically distributed evenly across the model surface to ensure robust segmentation. By restricting the distribution of these edge profiles to fixed parametric coordinates throughout the tracking, then basis functions for each edge profile can be precomputed during initialization. These basis functions are independent of the position of the control vertices, and can therefore be re-used during tracking to efficiently generate surface points regardless of local shape deformations.

3.1. Calculation of Surface Points

The Kalman filter framework requires the creation of a set of surface points \mathbf{p}_l with associated normal vectors \mathbf{n}_l and Jacobi matrices \mathbf{J}_l , based on a predicted state vector $\bar{\mathbf{x}}_l$. The creation of these objects can



(a)

Figure 2. Orthogonal views of the initial undeformed subdivision surface (left), as well as the same model deformed to fit the LV after tracking in some frames (right). The model consists of 24 surface patches, that each are outlined by a bold black border and subdivided into a 5 x 5 quadrilateral grid for visualization. The encapsulating wire-frame mesh illustrates the control vertices (circles) that define the surface.

be performed efficiently following the steps below:

1. Update position of control vertices \mathbf{q}_i based on the state vector: $\mathbf{q}_i = \bar{\mathbf{q}}_i + x_i \mathbf{d}_i$, where $\bar{\mathbf{q}}_i$ is the mean position of the control vertex and x_i is the parameter corresponding to this control vertex in the state vector for each control vertex. \mathbf{d}_i is the displacement direction for control vertex \mathbf{q}_i . The full state vector for the model then becomes the concatenation of the state parameters for all control vertices $\mathbf{x}_l = [x_1, x_2, \dots, x_{N_l}]^T$. One can here chose to force certain vertices to remain stationary during tracking without altering the overall approach. This would both reduce the deformation space, as well as the number of parameters to estimate.
2. Calculate surface points \mathbf{p}_l as a sums of control vertices weighted with their respective basis functions within the surface patch of each surface point: $\mathbf{p}_l = \sum_{i \in C(c_l)} \mathbf{b}_i \mathbf{q}_i$.
3. Calculate surface normals \mathbf{n}_l as the cross product between the partial derivatives of the basis functions with regards to parametric values u and v within the surface patch of each surface point: $\mathbf{n}_l = \sum_{i \in C(c_l)} (\mathbf{b}_u)_i \mathbf{q}_i \times \sum_{i \in C(c_l)} (\mathbf{b}_v)_i \mathbf{q}_i$.
4. Calculate Jacobian matrices for the local deformations \mathbf{J}_l by concatenating the displacement vectors multiplied with their respective basis functions: $\mathbf{J}_l = [\mathbf{b}_{i_1} \mathbf{d}_{i_1}, \mathbf{b}_{i_2} \mathbf{d}_{i_2}, \dots]_{i \in C(c_l)}$. The Jacobian matrix will here be padded with zeros for columns corresponding to control vertices outside the region of support for the surface patch of each surface point.

Precomputation of basis functions enables the operations above to be performed very quickly, which is crucial for enabling real-time implementations.

3.2. Global Transform

The composite object deformation $\mathbf{T}(\mathbf{x}) = \mathbf{T}_g(\mathbf{T}_l(\mathbf{x}_l), \mathbf{x}_g)$ is obtained by combining the local deformations of the subdivision model with a global transform to create a joint model. This leads to a

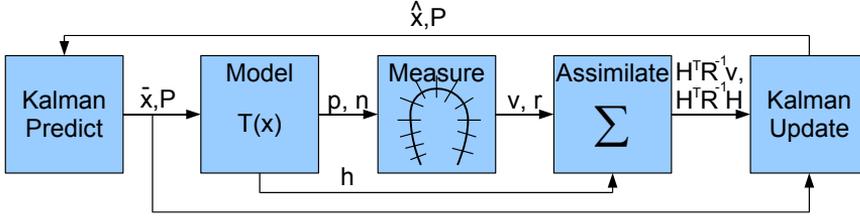


Figure 3. Overview over the processing chain in the Kalman filter based tracking framework. All five steps are performed only once for each new frame.

composite state vector $\mathbf{x} = [\mathbf{x}_g^T, \mathbf{x}_l^T]^T$ consisting of N_g global and N_l local deformation parameters. This separation between local and global deformations is intended to ease modeling, since changes in shape are often parametrized differently compared to deformations associated with global position, size and orientation.

We denote \mathbf{p}_l , \mathbf{n}_l and \mathbf{J}_l for the surface points created from the subdivision surface with local deformations $\mathbf{T}_l(\mathbf{x}_l)$. These points are subsequently transformed by means of a global *pose transform* \mathbf{T}_g , that translates, rotates and scales the model to align it correctly within the image volume. Surface points are trivially transformed using \mathbf{T}_g , whereas normal vectors must be transformed by multiplying with the normalized inverse spatial derivative of \mathbf{T}_g to remain surface normals after the global transform [11]:

$$\mathbf{p}_g = \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g) \quad (6)$$

$$\mathbf{n}_g = \left| \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \right| \left(\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \right)^{-T} \mathbf{n}_l \quad (7)$$

The Jacobian matrices for the composite deformations then becomes the concatenation of both global and local state-space derivatives. The local part is created by multiplying the 3×3 spatial Jacobian matrix for the global transform with the $3 \times N_l$ local Jacobian matrix for the deformable model, as follows from the chain-rule of multivariate calculus:

$$\mathbf{J}_g = \left[\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{x}_g}, \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \mathbf{J}_l \right]. \quad (8)$$

4. Kalman Tracking Framework

The tracking framework is decomposed into the 5 separate steps shown in Fig. 3.

4.1. State Prediction

Incorporation of temporal constraints is accomplished in the prediction stage of the Kalman filter by augmenting the state vector to contain the last two successive state estimates. A *motion model* which predicts state $\bar{\mathbf{x}}$ at time step $k + 1$, with focus on deviation from a mean state \mathbf{x}_0 , can then be expressed as:

$$\bar{\mathbf{x}}_{k+1} - \mathbf{x}_0 = \mathbf{A}_1 (\hat{\mathbf{x}}_k - \mathbf{x}_0) + \mathbf{A}_2 (\hat{\mathbf{x}}_{k-1} - \mathbf{x}_0), \quad (9)$$

where $\hat{\mathbf{x}}_k$ is the estimated state from time step k . Tuning of properties like damping and regularization towards the mean state \mathbf{x}_0 for all deformation parameters can then be accomplished by adjusting the

coefficients in matrices \mathbf{A}_1 and \mathbf{A}_2 . Prediction uncertainty can similarly be adjusted by manipulating the process noise covariance matrix \mathbf{B}_0 that is used in the associated covariance update equation for $\bar{\mathbf{P}}_{k+1}$. The latter will then restrict the rate of which parameter values are allowed to vary.

4.2. Evaluation of Deformable Model

Creation of surface points \mathbf{p} , normals \mathbf{n} and Jacobian matrices \mathbf{J} , based on the predicted state $\bar{\mathbf{x}}_k$. This is performed as described in section 3.

4.3. Edge Measurements

Edge measurements are used to guide the model toward the object being tracked. This is done by measuring the distance between points on a predicted model inferred from the motion model, and edges found by searching in normal direction of the model surface. This type of edge detection is referred to as *normal displacement* [4], and is calculated as the normal projection of the distance between a predicted edge point \mathbf{p} with associated normal vector \mathbf{n} and a measured edge point \mathbf{p}_{obs} :

$$v = \mathbf{n}^T (\mathbf{p}_{obs} - \mathbf{p}) . \quad (10)$$

Each normal displacement measurement is coupled with a *measurement noise* r value that specifies the uncertainty associated with the edge. This value is typically dependent on edge strength or other measure of uncertainty. This choice of normal displacement measurements with associated measurements noise enables usage of a wide range of possible edge detectors. The only requirement for the edge detector is that it must identify the most promising edge candidate for each search profile, and assign an uncertainty value to this candidate.

Linearized measurement models [12], which are required in the Kalman filter for each edge measurement, are constructed by transforming the state-space Jacobi matrices the same way as the edge measurements, namely taking the normal vector projection of them:

$$\mathbf{h}^T = \mathbf{n}^T \mathbf{J} . \quad (11)$$

This yields a separate *measurement vector* \mathbf{h} for each normal displacement measurement, that relates the normal displacements to changes in the state vector.

4.4. Measurement Assimilation

State-space segmentation forms a special problem structure, since the number of measurements typically far exceeds the number of state dimensions. Ordinary Kalman gain calculation will then be computationally intractable, since they involve inverting matrices with dimensions equal to the number of measurements.

An alternative approach is to assimilate measurements in *information space* [12] prior to the state update step. This enables very efficient processing if we assume that the measurements are uncorrelated, since uncorrelated measurements lead to a diagonal measurement covariance matrix \mathbf{R} . All measurement information can then be summed into an information vector and matrix of dimensions invariant to the number of measurements:

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{h}_i r_i^{-1} v_i \quad (12)$$

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T . \quad (13)$$

Distance [mm]	EDV [ml]	ESV [ml]	EF [%]
2.2 ± 0.8	3.6 ± 21.4	9.0 ± 17.4	-5.9 ± 11.1

Table 1. Bland-Altman analysis of the segmentation results compared to the reference segmentation. Results are expressed as mean difference $\pm 1.96SD$.

4.5. Measurement Update

Measurements in information filter form require some alterations to the state update step in the Kalman filter. This can be accomplished by utilizing that the Kalman gain $\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1}$, and reformulating the equations to account for measurements in information space. The updated state estimate $\hat{\mathbf{x}}$ for time step k then becomes:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1} \mathbf{v}_k . \quad (14)$$

The updated error covariance matrix $\hat{\mathbf{P}}$ can similarly be calculated in information space to avoid inverting unnecessary large matrices:

$$\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} . \quad (15)$$

This form only requires inversion of matrices with dimensions equal to the state dimension.

5. Results

The proposed tracking framework were evaluated by performing fully automatic tracking in 21 unselected 3D echocardiography recordings of the heart, recorded with a Vivid 7 ultrasound scanner (GE Vingmed Ultrasound, Norway) using a matrix array transducer (3V). Exactly the same initialization were used in all of the recordings, and the resulting segmentations were compared to semi-automatic segmentation using a custom made segmentation tool (GE Vingmed Ultrasound, Norway). The reference segmentations were conducted by an expert, and whenever needed manually adjusted to serve as a validated reference comparable to manual tracing.

A manually constructed Doo-Sabin model consisting of 24 surface patches, as shown in Fig 2, were used as basis for the LV segmentation. This deformable model were combined with a global pose transform that featured parameters for translation, rotation and isotropic scaling of the model to position and orient it within the image volume. 384 edge profiles distributed evenly over the surface were used for edge detection. Each of these profiles consisted of 30 image samples spaced 1 mm apart. Edge detection was then performed in each profile, based on a transition criterion [2], with edge weighting based on the transition height. These edge measurements were combined with an outlier removal step which discarded edges with normal displacement value differencing significantly for that of its neighbors.

Figure 4 and table 1 shows the results from the comparison with the reference segmentation, using Bland-Altman analysis of the average point to surface distance between the meshes, difference in end diastolic volumes (EDV), end systolic volumes (ESV) as well as differences in the computed ejection fraction (EF)¹. Fig. 5 shows orthogonal intersection slices of the segmentations at end diastole and end systole from four of the recordings to illustrate the typical segmentation quality obtained.

¹Ejection fraction is the ratio of LV contraction. It is commonly used for assessment of global ventricular function, and is computed as $(EDV-ESV)/EDV$.

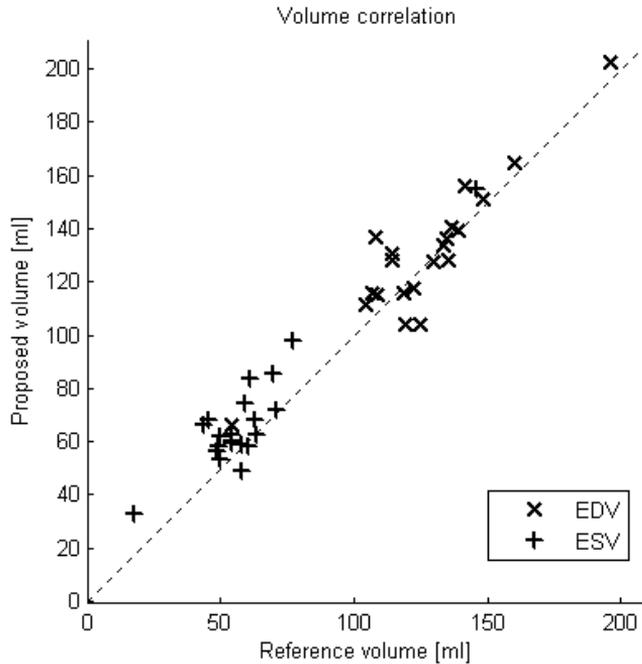


Figure 4. Volume correlation plot for the proposed segmentation against the reference method at end-diastole (EDV) and end-systole (ESV) in each of the 21 recording.

Compared to previous efforts on real-time Kalman segmentation [2, 3], this indicates slightly more accurate segmentations with smaller distances between the meshes and less bias in the ejection fraction numbers. The tracking and segmentation was performed using a C++ implementation on a 2.16 GHz Intel Core 2 duo processor, were real-time segmentation at a rate of 25 frames per second consumed approximately 8% CPU power.

The tracking converged in 2-4 frames after initialization in most of the recordings. A typical convergence for one of the recordings is shown in Fig. 6. A single recording did, however, require a half heartbeat to converge, because to the basal edge profiles were not long enough to reach the base of the heart before the heart was maximally contracted at end systole.

6. Discussion

In this paper, we have extended prior work [2, 3] to enable fully automatic and real-time LV tracking and segmentation in dense volumetric data using deformable subdivision surfaces.

6.1. Subdivision Model

Usage of subdivision models for segmentation has some desirable properties that makes them suitable for modeling of cardiac structures, in that they combine the inherent smoothness and continuity of surface derivatives of spline surfaces with the support of arbitrary topology from flat polygonal meshes. This enables more flexible modeling, since control vertices of the surface models are no longer restricted

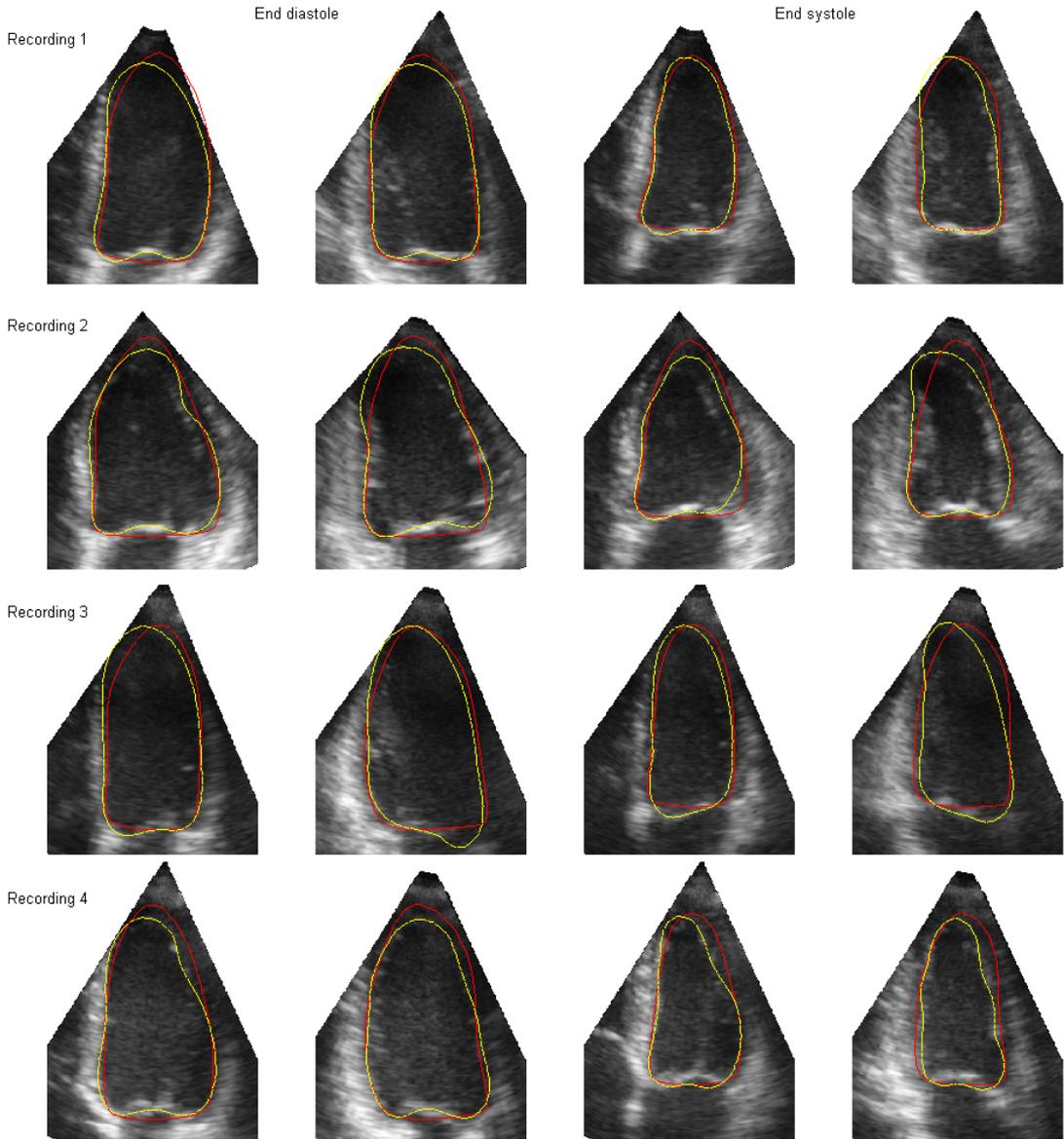


Figure 5. Example segmentation results from four of the recordings (rows). Orthogonal intersection slices through each volume shows the segmentation result at end-diastole (left) and end systole (right). The red intersection lines show the proposed Kalman segmentation, and the yellow lines the reference segmentation.

to the quadrilateral structure known from spline surfaces. Subdivision models that form closed surfaces and surfaces with complex geometries can therefore be constructed in a simple and intuitive fashion.

The inherent smoothness also makes it possible to represent cardiac geometries using low resolution

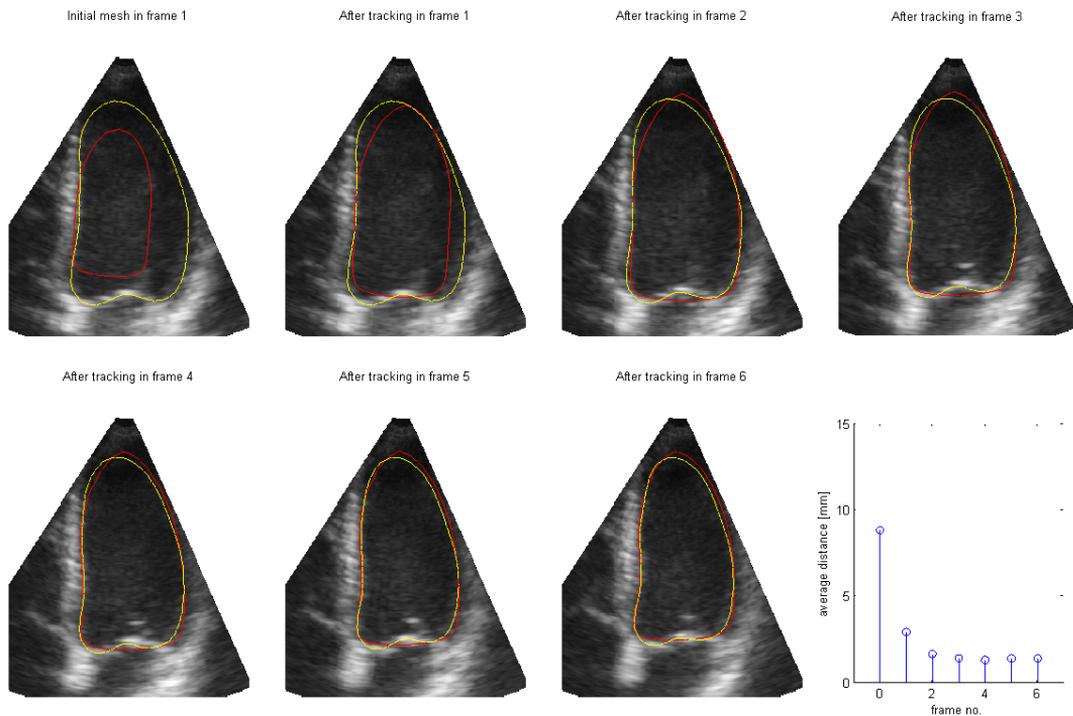


Figure 6. Intersection slices through the volume showing the convergence rate for the tracking. The initial mesh before tracking is started is shown top-left (red for proposed and yellow for reference), followed by deformed meshes after tracking in the first 6 frames. A plot of the average surface to surface distance between the deformed meshes and the reference segmentation for each frame is shown bottom-right.

models, consisting of few control vertices, as shown in this paper. This makes for more robust segmentation compared to high resolution models, since fewer shape parameters have to be estimated during tracking. Papillary muscles and trabecular structures, that are known to disrupt segmentation [1], are also handled robustly, since the low-resolution model are unable to represent the local sharp discontinuities in shape these structures represents.

Doo-Sabin is chosen in favor of the more common Catmull-Clark surfaces, that generalize bi-cubic B-spline surfaces, because bi-quadric surfaces has a narrower region of support compared to bi-cubic surfaces. This makes them more suitable for segmentation with low resolution models, where the range of support for each surface patch should be restricted to a local area, and not be so large that it covers a significant portion of the entire model. The proposed segmentation approach is, however, not restricted to Doo-Sabin surfaces in particular, so other subdivision schemes, such as Catmull-Clark and Loop could also be used without altering the overall approach. The only requirement is that the subdivision process can be expressed as a linear operation on matrix form.

Usage of subdivision models for cardiac tracking have also been presented in [13], but this paper focuses on image registration using iterative gradient descent algorithms, and not on segmentation per se. It therefore depends on manual surface initialization, and is thus not suitable for fully automatic

behavior.

6.2. Segmentation Results

The results shown in table 1 indicate that usage of subdivision surfaces leads to improved segmentation accuracy compared to spline and active-shape models [2, 3], even though fewer edge profiles were used for edge detection. This is believed to be caused by the subdivision model is more capable of capturing the shape and deformation pattern of the LV.

The segmented ventricles showed good overall agreement with the reference segmentations, both with respect to point to surface distances and for the computed volumes. Segmentation accuracy is primarily limited by the difficulty of edge detection in echocardiography recordings, which suffers from inherently poor image quality. It is difficult, even for experts, to accurately determine the endocardial border in such recordings. Perfect agreement with reference segmentations might therefore never be achieved. The simple transition criteria used in this paper is chosen primarily because it behaves robustly and has a long radius of convergence. More advanced criterias might very well yield more accurate segmentation in areas of weak and unclear edges, but state of the art edge detection is not the main focus of this paper.

Tracking convergence seems to primarily be limited by the length of the edge profiles. There is, however, an inherent trade-off between convergence speed, and edge detection robustness/outlier frequency here, since longer edge profiles might lead to the detection of more outlier edges that might disrupt the tracking. When disregarding the first few frames during tracking, the Kalman tracker seemed to respond fast enough to capture changes in pose and shape between successive frames, even though it only used a single refinement iteration per frame.

Precalculation of basis functions for the subdivision model during initialization also lead to a more computationally efficient implementation compared to [2, 3], even when compensating for the reduction in the number of edge profiles used.

6.3. Kalman Filter Approach

Most segmentation approaches used in medical imaging, such as active shape segmentation and simplex mesh segmentation [1], are based on modeling of forces acting upon a deformable model with semi-realistic physical properties. Segmentation is then performed by using iterative optimization algorithms to determine an equilibrium state between internal shape forces and external image forces. Segmentation typically requires hundreds of iterations to converge using this approach, which makes real-time 3D segmentation using this approach computationally intractable.

State-estimation based tracking instead uses a non-iterative algorithm, based on a Bayesian least squares solution of the linearized tracking problem, namely the extended Kalman filter. The model is segmented by computing a solution to a system of equations to fit the model to the detected edges, while at the same time regularizing the fitting by incorporating a kinematic model to restrict the rate of change for shape and pose parameters. This leads to outstanding computational performance compared to iterative algorithms, and enables real-time tracking and segmentation in volumetric datasets. Usage of extended Kalman filters for segmentation does, however, imply the making of some assumptions with regards to Gaussian distributions and linearity:

Firstly, the framework assumes that the normal displacement values are independent and follow a Gaussian distribution. This, however, only accounts for the normal displacement values only, and not to the shape of the underlying edge profiles or details in the edge detector, in which are not assumed

to follow any given distribution in the Kalman filter. Secondly, the extended Kalman filter assumes all deformations to be linear. Except for global rotation, which is inherently non-linear, every other mode of deformation is linear. This includes global translation, scaling and the local shape deformations, which due to the tensor product formulation of the polynomial basis functions are linear in the position of the control vertices. There is very little change in global rotation of the heart between successive frames, so the linearization approximation penalty is believed to be small.

7. Conclusion

In this paper, we have proposed a Kalman filter based framework for fully automatic real-time tracking and segmentation in volumetric data, using deformable subdivision surfaces. Usage of subdivision surfaces enables simple modeling of closed surfaces, and surfaces with complex topology, without any of the limitations associated with spline surfaces. In addition, a method for exact evaluation of surface points at arbitrary parameter values for Doo-Sabin surfaces is provided, to enable efficient precalculation of basis functions used to extract edge profiles.

The results indicate that usage of subdivision surfaces leads to improved segmentation accuracy compared to spline and active-shape models [2, 3]. Precalculation of basis functions also significantly reduces the computational complexity. The combination of subdivision models with a Kalman filter tracker thus enables 3D segmentation that is both robust and capable of operating in real-time.

A. Doo-Sabin Subdivision Matrix

The subdivision weights used for faces consisting of n vertices are used as defined by Doo & Sabin [8]:

$$\alpha_n^j = \frac{\delta_{j,0}}{4} + \frac{3 + 2 \cos(2\pi j/n)}{4n}, \quad (16)$$

where $\delta_{i,j}$ is the Kronecker delta function which is one for $i = j$ and zero elsewhere. Subdivision of the control vertices within a single face can then be expressed as a linear operation using a subdivision matrix \mathbf{S}_n :

$$\mathbf{S}_n = \begin{bmatrix} \alpha_n^0 & \alpha_n^1 & \alpha_n^2 & \dots & \alpha_n^{-1} \\ \alpha_n^{-1} & \alpha_n^0 & \alpha_n^1 & \dots & \alpha_n^{-2} \\ \alpha_n^{-2} & \alpha_n^{-1} & \alpha_n^0 & \dots & \alpha_n^{-3} \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_n^1 & \alpha_n^2 & \alpha_n^3 & \dots & \alpha_n^0 \end{bmatrix}. \quad (17)$$

Subdivision of whole patches is accomplished by combining \mathbf{S}_n for all four faces in a patch into a composite subdivision matrix \mathbf{S} . The structure of this matrix depends on the topology and control vertex enumeration scheme employed, but construction should be straightforward.

B. Basis functions for Quadratic B-splines

The 9 tensor product quadratic B-spline functions can be expressed as a product of two separable basis polynomials for the parametric value u and v ($i = 0, \dots, 8$):

$$\tilde{b}_i(u, v) = P_{i\%3}(u) P_{i/3}(v), \quad (18)$$

where “%” and “/” denotes the division remainder and division operators respectively. $P_i(t)$ are the basis polynomials for quadratic B-splines with uniform knot vectors:

$$2P_0(t) = 1 - 2t + t^2 \quad (19)$$

$$2P_1(t) = 1 + 2t - 2t^2 \quad (20)$$

$$2P_2(t) = t^2 \quad (21)$$

References

- [1] J. A. Noble and D. Boukerroui, “Ultrasound image segmentation: A survey,” *Medical Imaging, IEEE Transactions on*, vol. 25, no. 8, pp. 987–1010, 2006.
- [2] F. Orderud, J. Hansegård, and S. I. Rabben, “Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, vol. 4791 of *LNCS*, pp. 858–865, Springer, 2007.
- [3] J. Hansegård, F. Orderud, and S. Rabben, “Real-time active shape models for segmentation of 3D cardiac ultrasound,” in *Computer Analysis of Images and Patterns - CAIP*, pp. 157–164, 2007.
- [4] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.
- [5] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning, “A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography,” *Medical Imaging, IEEE Transactions on*, vol. 21, no. 3, pp. 226–238, 2002.
- [6] D. Comaniciu, X. S. Zhou, and S. Krishnan, “Robust real-time myocardial border tracking for echocardiography: An information fusion approach,” *Medical Imaging, IEEE Transactions on*, vol. 23, no. 7, pp. 849–860, 2004.
- [7] H. Liu and P. Shi, “State-space analysis of cardiac motion with biomechanical constraints,” *Image Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 901–917, April 2007.
- [8] D. Doo and M. Sabin, “Behaviour of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, vol. 10, pp. 356–360, Nov. 1978.
- [9] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, pp. 350–355, Nov. 1978.
- [10] J. 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*, (New York, NY, USA), pp. 395–404, ACM Press, 1998.
- [11] A. H. Barr, “Global and local deformations of solid primitives,” in *SIGGRAPH ’84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 21–30, ACM Press, 1984.
- [12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.

- [13] R. Chandrashekhara, R. Mohiaddin, R. Razavi, and D. Rueckert, "Nonrigid image registration with subdivision lattices: Application to cardiac mr image analysis," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, vol. 4791 of *LNCS*, pp. 335–342, Springer, 2007.

Real-time Left Ventricular Speckle-Tracking in 3D Echocardiography With Deformable Subdivision Surfaces

Fredrik Orderud¹, Gabriel Kiss¹, Stian Langeland², Espen W. Remme³,
Hans Torp¹, and Stein I. Rabben²

¹ Norwegian University of Science and Technology (NTNU), Norway

² GE Vingmed Ultrasound, Norway

³ Department of Cardiology, Rikshospitalet, Norway

Abstract. In this paper, we extend a computationally efficient framework for real-time tracking of deformable subdivision surfaces in 3D echocardiography with speckle-tracking measurements to track material points. Tracking is performed in a sequential state-estimation fashion, using an extended Kalman filter to update the subdivision surface based on displacement vectors from 3D block-matching in the left ventricular wall. Fully automatic tracking is demonstrated in two simulations of an infarcted ventricle, as well as in a set of 21 in-vivo 3D echocardiograms. Credible tracking results were achieved in all cases, with an average drift ratio of $12.08 \pm 2.09\%$ ($2.7 \pm 1.0\text{mm}$). The infarcted regions were also correctly identified in both of the simulations. Due to the high computational efficiency of the method, it is capable of operating in real-time.

1 Introduction

With the introduction of 3D echocardiography, rapid and low-cost acquisition of volumetric images of the left ventricle (LV) has become feasible. Tools for assessment of global function, based on semi-automatic shape segmentation of the endocardial boundary, have appeared over the last few years [1]. However, in order to evaluate regional function of the LV, methods that also estimate myocardial deformation by tracking material points are required.

The distinctive speckle pattern found in ultrasound images has often been considered an undesirable image artifact, since it reduces the apparent image quality. However, this pattern has the fortunate property that it, despite being gradually decorrelated, moves in the same manner as the underlying tissue being imaged [2]. This property can be exploited to track the LV myocardial deformation field by means of speckle-tracking techniques. Existing approaches for 2D and 3D speckle-tracking include: Horn-Schunck optical flow speckle-tracking [3], band-pass Gabor filtering prior to block-matching [4], tracking single speckle points with motion coherence regularization of the velocity field [5] and elastic volume registration using B-splines coupled with a mutual-information metric [6]. To our knowledge, none of these approaches have been demonstrated to achieve real-time processing times when applied to volumetric data.

This paper describes a fully automatic, real-time method for LV tracking of material points in 3D echocardiography. It extends the subdivision model based Kalman-filter method of [7], with speckle-tracking to capture the full myocardial deformation pattern, and not only shape changes. A combination of integer voxel displacement estimation using block matching, coupled with optical flow correction of the best match, is used to achieve sub-voxel displacement estimation.

2 Methods

The tracking framework is based on a deformable subdivision surface, consisting of control vertices \mathbf{q}_i for $i \in \{1 \dots N_g\}$ that are allowed to move to alter the shape and parameter-space density of the surface. Unlike in [7], where shape segmentation was the objective, we allow the control vertices to move freely in any direction, and not just in the surface-normal direction. In addition to the control vertices, the topological relationships between the control vertices have to be defined in a list $C(c)$, that maps surface patches $c \in \{1 \dots N_c\}$ to enumerated lists of control vertex indices that define the control vertices influencing each surface patch.

We denote the local deformations $\mathbf{T}_l(\mathbf{x}_l)$ to our deformable model as the deformations obtainable by moving the control vertices of the subdivision model. These local deformations are combined with a global transform $\mathbf{T}_g(\mathbf{x}_g, \mathbf{p}_l)$ to position, scale and orient the model. This leads to a composite state vector $\mathbf{x} = [\mathbf{x}_g^T, \mathbf{x}_l^T]^T$ consisting of N_g global and N_l local deformation parameters.

A manually constructed Doo-Sabin subdivision surface [8] consisting of 20 control vertices is used to represent the LV. A distribution of approximately 450 surface points, spread evenly across the surface as in [7], are used as shown in Fig. 1(a). These points are used as a basis for both edge-detection and speckle-tracking measurements, and consists of parametric coordinates (including patch number) for each of the surface points.

The tracking framework is decomposed into the 5 separate steps shown in Fig. 1(b). Most of the steps are similar to [7], with the exception of the measurement step, where edge-detection is replaced with 3D speckle-tracking to update the model. Edge-detection is instead used solely to automatically initialize the model to the endocardial boundary, prior to speckle-tracking. This initialization leads to shape deformations, by moving control vertices inwards and outwards in the direction perpendicular to the surface, but does not impose any deformations along the surface to alter the parameter-space density, as speckle-tracking does.

The Kalman prediction step and Kalman update steps are identical as in [7], and therefore not covered in this paper. Instead this paper focuses on what is new, namely evaluation of the deformable model, speckle-tracking measurements and assimilation of 3D displacement vectors:

2.1 Evaluation of Deformable Model

Calculation of Local Surface Points: The Kalman filter framework requires the creation of a set of surface points \mathbf{p}_l and Jacobi matrices \mathbf{J}_l , based on a pre-

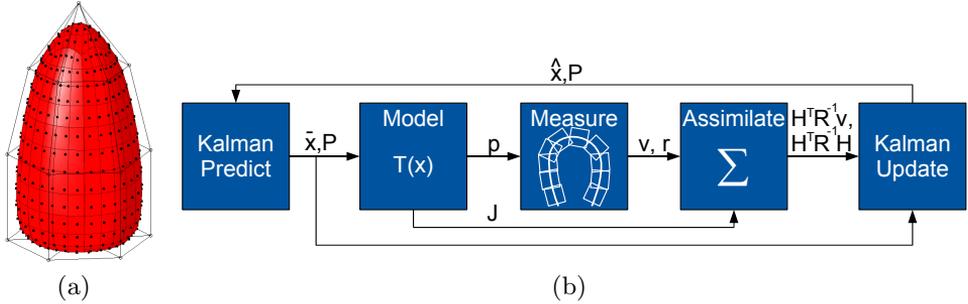


Fig. 1. (a) The Doo-Sabin subdivision surface used for tracking, which consists of 20 control vertices shown in the encapsulating wire-frame mesh. The speckle-tracking distribution is illustrated with black dots on the surface. (b) Overview over the processing chain for each new frame in the Kalman filter tracking framework.

dicted state vector $\bar{\mathbf{x}}_l$. The creation of these objects can be performed efficiently following the steps below:

1. Update position of control vertices \mathbf{q}_i based on the state vector: $\mathbf{q}_i = \bar{\mathbf{q}}_i + x_{(3i)}\mathbf{v}_x + x_{(3i+1)}\mathbf{v}_y + x_{(3i+2)}\mathbf{v}_z$, where $\bar{\mathbf{q}}_i$ is the initial position of the control vertex, \mathbf{v}_x , \mathbf{v}_y , \mathbf{v}_z are unit vectors along the x, y and z axis respectively, and $x_{(3i)}$, $x_{(3i+1)}$, $x_{(3i+2)}$ are the parameters in the state vector corresponding to this control vertex. The full state vector for the model then becomes the concatenation of the state parameters for all control vertices $\mathbf{x}_l = [x_0, x_1, \dots, x_{(3N_q-1)}]^T$.
2. Calculate surface points \mathbf{p}_l as a sum of control vertices weighted with their respective basis functions within the surface patch of each surface point: $\mathbf{p}_l = \sum_{i \in C(c_l)} \mathbf{b}_i \mathbf{q}_i$.
3. Calculate Jacobian matrices for the local deformations \mathbf{J}_l by concatenating the unit vectors multiplied by their respective basis functions: $\mathbf{J}_l = [\mathbf{b}_{i_1} \mathbf{v}_x, \mathbf{b}_{i_1} \mathbf{v}_y, \mathbf{b}_{i_1} \mathbf{v}_z, \mathbf{b}_{i_2} \mathbf{v}_x, \dots]_{i \in C(c_l)}$. The Jacobian matrix will here be padded with zeros for columns corresponding to control vertices outside the region of support for the surface patch of each surface point.

Basis functions for these points can be precomputed during initialization if we restrict the parametric coordinate distribution of the surface points to be constant throughout the tracking. This allows the above operations to be performed very quickly, which is crucial for enabling real-time implementations.

Global Transform: We denote \mathbf{p}_l and \mathbf{J}_l for the surface points created from the subdivision surface with local deformations $\mathbf{T}_l(\mathbf{x}_l)$. These points are subsequently transformed by means of a global *pose transform* \mathbf{T}_g , that translates, rotates and scales the model to align it correctly within the image volume:

$$\mathbf{p}_g = \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g) . \quad (1)$$

The Jacobian matrices for the composite deformations then becomes the concatenation of both global and local state-space derivatives. The local part is created by multiplying the 3×3 spatial Jacobian matrix for the global transform with the $3 \times N_l$ local Jacobian matrix for the deformable model, as follows from the chain-rule of multivariate calculus:

$$\mathbf{J}_g = \left[\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{x}_g}, \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \mathbf{J}_l \right]. \quad (2)$$

2.2 Speckle-tracking Measurements

Speckle-tracking measurements are performed 1mm outside the endocardial surface, in order to track deformations inside the myocardium. There, 3D displacement vectors $\mathbf{v} = [v_x, v_y, v_z]^T$ for local motion are inferred by matching predicted surface points \mathbf{p} in the current frame to associated surface point from the updated model in the previous frame. The measurements are computed by first performing 3D block-matching using a sum of absolute differences (SAD) metric to determine integer displacements. This is followed by translative Lucas-Kanade optical flow estimation [9] on the best integer voxel match to correct for sub-sample displacements, as was done for 2D tracking in [10].

Implementation of this matching can be done efficiently on modern processors by using vector instructions and multi-core parallelization of the SAD operations. Furthermore, preprocessing of the data is avoided by doing the tracking directly on the "raw" grayscale ultrasound data acquired in spherical coordinates. Tracking is performed in data decimated in the beam propagation direction by a factor of four to reduce window sizes, since ultrasound image resolution is significantly higher in this direction compared to the two lateral directions. A kernel size of $4 \times 4 \times 4$ voxels is used, while the search window has an adaptive size, based on image depth, to make its cartesian dimensions approximately constant regardless of image depth.

Associated measurement noise values r , for the spatial uncertainty of the displacement measurement, are computed based on the ratio between the best and average SAD matching value for each point. After computing the measurements, simple outlier rejection is performed, based on the measurement noise values and a comparison with neighboring displacement vectors in a local search area. Parameters were adjusted by trial and error.

2.3 Measurement Assimilation

The measurements can be efficiently assimilated in information space if we assume that they are uncorrelated [11], since uncorrelated measurements lead to a diagonal measurement covariance matrix \mathbf{R} . All measurement information can then be summed into an information vector and matrix of dimensions invariant to the number of measurements:

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{H}_i^T \mathbf{v}_i r_i^{-1} \quad (3)$$

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{H}_i^T \mathbf{H}_i r_i^{-1}. \quad (4)$$

Usage of unit vectors in x, y and z-direction for displacing control vertices enables direct usage of the Jacobian matrices as measurement matrices in the Kalman filter, since $\mathbf{H}^T = [\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z]^T \mathbf{J} = \mathbf{J}$. A covariance matrix for the measurement can also be used instead of scalar measurement noise values if one desires to capture any non-isotropy in the spatial uncertainty of the displacement, but this is not done in this paper.

3 Experimental Validation & Results

In order to show the feasibility of the method and validate its performance, the method was applied to both simulated and in-vivo data sets:

3.1 Data Description

Two volumetric ultrasound datasets was generated for the experiment, based on a finite element simulation of a left ventricle with an antero-apical infarction. The first simulation used an ellipsoidal shape for the myocardium, while the second used the average shape of five canine ventricles. The motion and deformation of the ventricle was determined by modeling internal systolic contraction forces and external forces from the cavity pressure, and the infarcted area was modeled by abolishing contractile forces in the antero-apical region, as in [12]. A *k-space* ultrasound simulator, described in [13], was then used to create realistic 3D ultrasound simulations based on scatter positions extracted from the finite element model.

In addition to the simulations, a collection of 21 apical 3D echocardiography recordings of adult patients, of which half were diagnosed with heart diseases, were used for in-vivo validation of the method. These recordings were acquired with a Vivid 7 scanner (GE Vingmed Ultrasound, Norway) using a matrix array transducer (3V). The exact same configuration was used to initialize tracking in all in-vivo recordings.

3.2 Simulated Data Results

After tracking, the subdivision surfaces were re-meshed into a grid-structured mesh in a manner that preserves material points. Area strain $\epsilon = (a - a_0)/a_0$ values were then computed locally across the surface by comparing the area of each quadrilateral during tracking with associated end-diastolic (ED) areas. These strain values represents variations in the parameter-space density of the subdivision surface, which should correspond to the total muscle contraction locally in a manner that combines the effect of longitudinal and circumferential strain.

Figure 2(b) shows end-systolic (ES) area strain values across the tracked surface, together with ground truth strain values from both simulations. The infarcted regions exhibit small contraction or stretching, and therefore show up as blue and green, while healthy myocardium is contracting, and therefore shows

up as red. One can clearly see that the infarcted regions are correctly identified in the tracked meshes, although the strain values in the infarcted regions are underestimated compared to the ground truth, especially in the second simulation which has a more complex geometry.

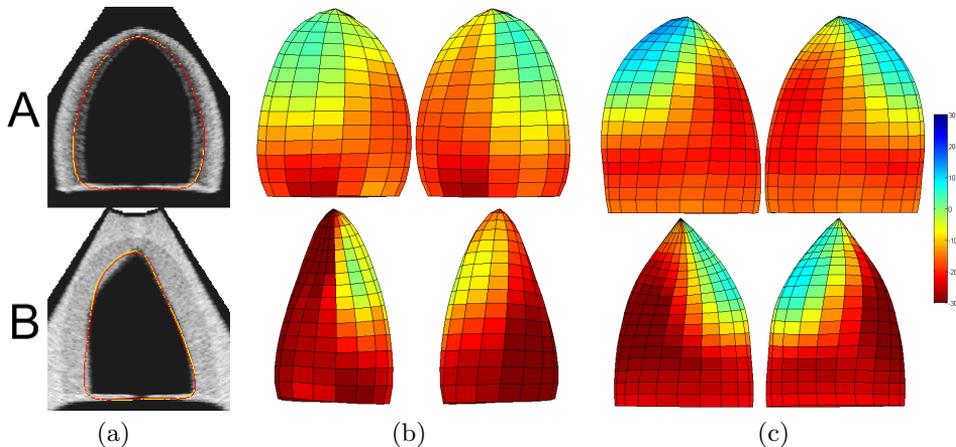


Fig. 2. Results from tracking in the simulated data, showing (a) intersection slices through the simulations, as well as front & back views of color-coded area strain meshes at ES from (b) the tracked strain meshes and (c) ground truth values. Results from the infarcted ellipsoid (simulation 'A') are shown in the top row, and from the infarcted dog-heart (simulation 'B') in the bottom row.

3.3 In-vivo Results

For the in-vivo data, no ground truth was available, so tracking was instead evaluated by computing the average ratio between the drift of surface points after tracking an entire cardiac cycle, and the walked path distance for the same point.

Tracking in the 21 in-vivo recordings yielded an average drift ratio of $12.08 \pm 2.09\%$, which in absolute values corresponds to $2.7 \pm 1.0\text{mm}$. As a comparison, the drift ratio in the simulated recordings were 8.58% and 10.59% , with absolute drift values of 0.58 and 0.70mm . Fig. 3 shows orthogonal intersection slices of the tracking results at ED and ES in two of the recordings.

Tracking in both the simulated and in-vivo recordings consumed approximately 37ms of processing time per frame on a 2.2GHz Intel Core 2 duo processor. This makes the method capable of operating in real-time, given a typical frame-rate of 25fps .

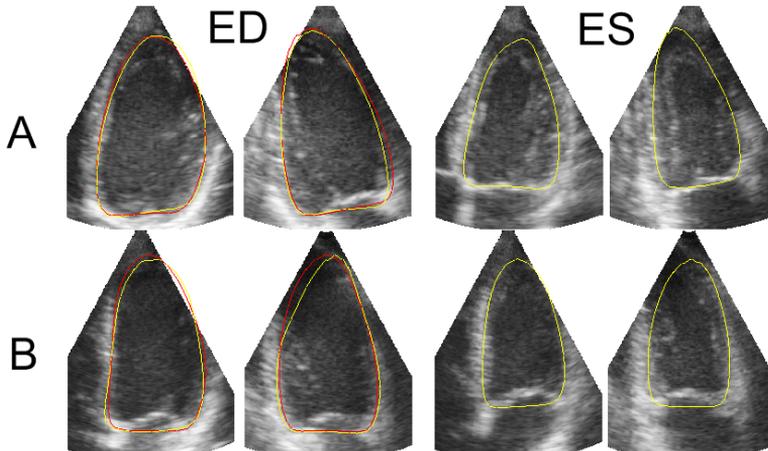


Fig. 3. Orthogonal image slices of the tracked mesh in two of the in-vivo recordings (patient 'A' and 'B'), at both ED and ES. The slices show the initialized mesh in red (only at ED), and tracked meshes in yellow (both at ED and ES). The difference between the tracked and initialized mesh at ED constitutes the drift after tracking in an entire cardiac cycle.

4 Discussion

We have presented a new approach for LV tracking of material points in 3D echocardiography, using a Kalman filter to fit a deformable subdivision model to 3D speckle-tracking measurements. The method is automatically initialized using endocardial edge-detection, and is capable of operating in real-time due to its strong computational efficiency. Usage of this method might therefore enable rapid analysis of regional myocardial function.

Automatic myocardial speckle-tracking was feasible in all tested recordings, both simulated and in-vivo. Based on visual assesment of the tracking and computed drift values, the tracking was found to behave robustly. The obvious discrepancy between absolute drift and drift ratio in the simulations is believed to stem from the fact that apex moves very little in the simulations, which leads to high drift ratios in the apical region, even though the absolute drift is quite small. Furthermore, the method was able to identify the infarcted regions in two different ultrasound simulations of a left ventricle, although the strain in the infarcted areas appeared underestimated compared to ground truth.

Tracking accuracy is dependent on the resolution of the subdivision surface used. Low-resolution surfaces, like the one used in this paper, will exhibit high robustness, due to the inherent regularization of having fewer parameters to estimate. This does, however, come at the expense of stronger spatial smoothing across the surface, compared to more high-resolution surfaces. This might be some of the reason for the underestimated infarcted strain. Usage of a higher resolution model, consisting of more control vertices, might therefore enable more accurate tracking with less spatial smearing of the deformation field. Usage of

more surface points for block-matching might also increase tracking accuracy at the expense of computational efficiency.

With feasibility of the method demonstrated, the next step will be to compare its accuracy to alternative approaches in a more quantitative way. Approaches to improve its accuracy should also be investigated. Currently, tracking is performed sequentially, from one frame to the next. This can, however, be extended with bidirectional tracking, using both a forward and backward Kalman-filter to improve tracking accuracy and reduce drift. Speckle-tracking measurements might also be combined more directly with edge-detection to reduce the surface-normal component of the inherent drift associated with sequential block-matching.

References

1. Jacobs, L.D., Salgo, I.S., Goonewardena, S., Weinert, L., Coon, P., Bardo, D., Gerard, O., Allain, P., Zamorano, J.L., de Isla, L.P., Mor-Avi, V., Lang, R.M.: Rapid online quantification of left ventricular volume from real-time three-dimensional echocardiographic data. *European Heart Journal* **27** (November 2006) 460–468
2. D’hooge, J., Bijnens, B., Thoen, J., Van de Werf, F., Sutherland, G., Suetens, P.: Echocardiographic strain and strain-rate imaging: a new tool to study regional myocardial function. *Medical Imaging, IEEE Transactions on* **21**(9) (Sep 2002) 1022–1030
3. Meunier, J.: Tissue motion assessment from 3D echographic speckle tracking. *Physics in Medicine and Biology* **43**(5) (1998) 1241–1254
4. Yu, W., Lin, N., Yan, P., Purushothaman, K., Sinusas, A., Thiele, K., Duncan, J.S.: Motion analysis of 3D ultrasound texture patterns. *Functional Imaging and Modeling of the Heart* (2003) 1006–1006
5. Song, X., Myronenko, A., Sahn, D.J.: Speckle tracking in 3D echocardiography with motion coherence. *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on* (17-22 June 2007) 1–7
6. Elen, A., Loeckx, D., Choi, H.F., Gao, H., Claus, P., Maes, F., Suetens, P., D’hooge, J.: P4a-5 3D cardiac strain estimation using spatio-temporal elastic registration: In silico validation. *Ultrasonics Symposium, 2007. IEEE* (28-31 Oct. 2007) 1945–1948
7. Orderud, F., Rabben, S.I.: Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces. *Computer Vision and Pattern Recognition, 2008. CVPR ’08. IEEE Conference on*
8. Doo, D., Sabin, M.: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design* **10**(6) (November 1978) 356–360
9. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *IJCAI81*. (1981) 674–679
10. Behar, V., Adam, D., Lysyansky, P., Friedman, Z.: Improving motion estimation by accounting for local image distortion. *Ultrasonics* **43**(1) (October 2004) 57–65
11. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience (2001)
12. Remme, E., Smiseth, O.: Characteristic strain pattern of moderately ischemic myocardium investigated in a finite element simulation model. In: *Functional Imaging and Modeling of the Heart*. (2007) 330–339
13. Hergum, T., Crosby, J., Langhammer, M., Torp, H.: The effect of including fiber orientation in simulated 3D ultrasound images of the heart. *Ultrasonics Symposium, 2006. IEEE* (2-6 Oct. 2006) 1991–1994

Combining Edge Detection With Speckle-Tracking for Cardiac Strain Assessment in 3D echocardiography

Fredrik Orderud*, Gabriel Kiss*, Stian Langeland[†],

Espen W. Remme[‡], Hans G. Torp* and Stein I. Rabben[†]

*Norwegian University of Science and Technology (NTNU), Norway, [†]GE Vingmed
Ultrasound, Norway,

[‡]Department of Cardiology, Rikshospitalet, Norway

Abstract

In this paper, we extend a computationally efficient framework for tracking of deformable subdivision surfaces in 3D echocardiography with speckle-tracking measurements to track material points. Tracking is performed in a sequential state-estimation fashion, using an extended Kalman filter to update a subdivision surface in a two-step process: Edge-detection is first performed to update the model for changes in shape and position, followed by a second update based on displacement vectors from speckle-tracking with 3D block-matching. The latter speckle-tracking update will only have to correct for residual deformations after edge-detection. This both leads to increased accuracy and computational efficiency compared to usage of speckle-tracking alone.

Automatic tracking is demonstrated in a 3D echocardiography simulation of an infarcted ventricle. The combination of edge-detection and speckle-tracking consistently improved tracking accuracy (RMS 0.483, 0.433, 0.511 mm in X,Y,Z) compared to speckle-tracking alone (RMS 0.663, 0.439, 0.613 mm). It also improved the qualitative agreement for color-coded strain meshes to ground truth, and more clearly identified the infarcted region.

I. INTRODUCTION

The introduction of 3D echocardiography has enabled rapid and low-cost acquisition of volumetric images of the left ventricle (LV). Tools for assessment of global function, based on semi-automatic shape segmentation of the endocardial boundary, have appeared over the last few years [6]. However, in order to evaluate regional function associated with coronary artery disease, methods that also estimate the myocardial deformation field by tracking material points in 3D are required.

The distinctive speckle pattern found in ultrasound images has often been considered an undesirable image artifact, since it reduces the apparent image quality. However, this pattern have the fortunate property that it, despite being gradually decorrelated, moves in the same manner as the underlying tissue being imaged [2]. This property can be exploited to track the LV myocardial deformation field by means of speckle-tracking techniques. Several approaches for speckle-tracking in 3D ultrasound data have been proposed over the last years. Most prominent are the approaches based on optical flow speckle-tracking [7], block-matching, and more recently on elastic volume registration [4]. Most of these approaches are, however, computationally intensive and require manual initialization or endocardial tracing.

This paper describes a extension of a model-based state-estimation framework from [8], [9] with support for combined edge-detection with speckle-tracking. Edge-detection is first used to correct for shape and position changes, followed by a speckle-tracking step to correct for residual deformations. Together, this results in a fully automatic method for rapid assessment of cardiac strain in 3D echocardiography. Experimental validation in a simulation of an infarcted ventricle demonstrates the improved accuracy obtainable with the approach, compared to using speckle-tracking alone.

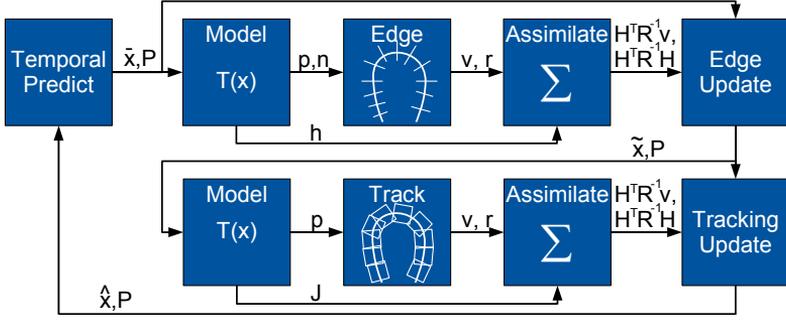


Figure 1: Overview over the processing steps in the Kalman tracking framework.

II. METHODS

The tracking framework is centered around a deformable subdivision surface, which is parametrized by a set of control vertices \mathbf{q}_i for $i \in \{1 \dots N_q\}$ that are allowed to move to alter the shape and parameter-space density of the surface. Unlike in [8], where shape segmentation was the objective, we allow the control vertices to move freely in any direction, and not just in the surface-normal direction.

We denote the local deformations $\mathbf{T}_l(\mathbf{x}_l)$ as the deformations obtained by moving the control vertices of the subdivision model. These local deformations are combined with a global transform $\mathbf{T}_g(\mathbf{x}_g, \mathbf{p}_l)$ to position, scale and orient the model. This leads to a composite state vector $\mathbf{x} = [\mathbf{x}_g^T, \mathbf{x}_l^T]^T$ consisting of N_g global and N_l local deformation parameters.

A manually constructed Doo-Sabin subdivision surface [3] consisting of 20 control vertices is used to represent the LV. A distribution of approximately 500 surface points, spread evenly across the surface, is defined to be used for both edge-detection and speckle-tracking measurements in the tracking framework.

The tracking framework consists of three separate stages, namely temporal prediction, edge-detection update and speckle-tracking update. The latter two can be decomposed into 4 steps each, which results in a total of 9 separate processing steps, as can be seen in Fig. 1. The prediction and edge update steps are similar to [8], and will adapt the model to align it to the endocardial boundary. The second measurement stage then uses speckle-tracking to acquire 3D displacement measurements, and update the model for any residual deformations after the edge-detection. This combined approach is motivated by the inherent problem of drift, due to cumulative buildup of tracking error, associated with speckle-tracking based on sequential block-matching. The proposed approach limits this problem by using edge-detection to ensure that the model remains aligned to the endocardial border at all times during tracking.

The temporal prediction and both state update steps are identical as in [8], and therefore not covered in this paper. Instead, this paper focuses on what is new, namely evaluation of the deformable model, speckle-tracking measurements and the assimilation of 3D displacement vectors:

A. Evaluation of Deformable Model

1) *Calculation of Local Surface Points:* The Kalman filter framework requires the creation of a set of surface points \mathbf{p}_l and Jacobi matrices \mathbf{J}_l , based on a predicted state vector $\bar{\mathbf{x}}_l$. The creation of these objects can be performed efficiently following the steps below:

- 1) Update position of control vertices \mathbf{q}_i based on the state vector: $\mathbf{q}_i = \bar{\mathbf{q}}_i + x_{(3i)}\mathbf{v}_x + x_{(3i+1)}\mathbf{v}_y + x_{(3i+2)}\mathbf{v}_z$, where $\bar{\mathbf{q}}_i$ is the initial position of the control vertex, \mathbf{v}_x , \mathbf{v}_y , \mathbf{v}_z are unit vectors along the x, y and z axis respectively, and $x_{(3i)}$, $x_{(3i+1)}$, $x_{(3i+2)}$ are the parameters in the state vector corresponding to this control vertex. The local state vector for the model then becomes the concatenation of the state parameters for all control vertices $\mathbf{x}_l = [x_0, x_1, \dots, x_{(3N_q-1)}]^T$.

- 2) Calculate surface points \mathbf{p}_l as a sum of control vertices weighted with their respective basis functions within the surface patch of each surface point: $\mathbf{p}_l = \sum_{i \in C(c_l)} \mathbf{b}_i \mathbf{q}_i$.
- 3) Calculate Jacobian matrices for the local deformations \mathbf{J}_l by concatenating the unit vectors multiplied with their respective basis functions: $\mathbf{J}_l = \left[\mathbf{b}_{i_1} \mathbf{v}_x, \mathbf{b}_{i_1} \mathbf{v}_y, \mathbf{b}_{i_1} \mathbf{v}_z, \mathbf{b}_{i_2} \mathbf{v}_x, \dots \right]_{i \in C(c_l)}$. The Jacobian matrix will here be padded with zeros for columns corresponding to control vertices outside the region of support for the surface patch of each surface point.

Basis functions and Jacobians for these points can be precomputed during initialization, as described in [8], since the parametric coordinate distribution remain fixed during tracking.

2) *Global Transform*: We denote \mathbf{p}_l and \mathbf{J}_l as the surface points and Jacobian created from the subdivision surface with local deformations $\mathbf{T}_l(\mathbf{x}_l)$. These points are subsequently transformed by means of a global *pose transform* \mathbf{T}_g , that translates, rotates and scales the model to align it correctly within the image volume:

$$\mathbf{p}_g = \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g) . \quad (1)$$

The Jacobian matrices for the composite deformations then becomes the concatenation of both global and local state-space derivatives. The local part is created by multiplying the 3×3 spatial Jacobian matrix for the global transform with the $3 \times N_l$ local Jacobian matrix for the deformable model, as follows from the chain-rule of multivariate calculus:

$$\mathbf{J}_g = \left[\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{x}_g}, \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \mathbf{J}_l \right] . \quad (2)$$

B. Edge or Speckle-tracking Measurements

Either edge-detection or speckle-tracking is here performed to compute 3D displacement vectors $\mathbf{v} = [v_x, v_y, v_z]^T$ relative to the predicted surface points \mathbf{p} , as well as associated measurement noise values r .

Endocardial edge-detection is performed in search-profiles perpendicular to the surface, as described in [8]. The position of the detected edges are then shifted 1 mm to align the model slightly inside the myocardium where speckle pattern is present.

Speckle-tracking is performed by matching search windows centered around the predicted surface points after the edge update to a smaller kernel window extracted from surface points in previous frame after all update steps were performed. The displacement vectors are computed by first performing 3D block-matching using a sum of absolute differences (SAD) metric to determine integer displacements. This is followed by translative Lucas-Kanade optical flow estimation on the best integer voxel match to correct for sub-sample displacements. The SAD matching operation was implemented using SIMD vector instructions and multi-core parallelization to run efficiently. Tracking is performed directly on the "raw" grayscale data acquired in spherical coordinates. More details on the speckle-tracking implementation and parameter configuration can be found in [9]. Measurement noise values was computed based on the ratio between the best and average SAD matching value for each point.

Simple outlier rejection was performed for both edge-detection and speckle-tracking, based on the measurement noise values and a comparison with neighboring displacement vectors in a local search area. Parameters were adjusted by trial and error.

C. Measurement Assimilation

The measurements can be efficiently assimilated in information space if we assume that they are uncorrelated [1], since uncorrelated measurements lead to a diagonal measurement covariance matrix \mathbf{R} . All measurement information can then be summed into an information vector and matrix of dimensions invariant to the number of measurements:

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{H}_i^T \mathbf{v}_i r_i^{-1} \quad (3)$$

$$\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{H}_i^T \mathbf{H}_i r_i^{-1} . \quad (4)$$

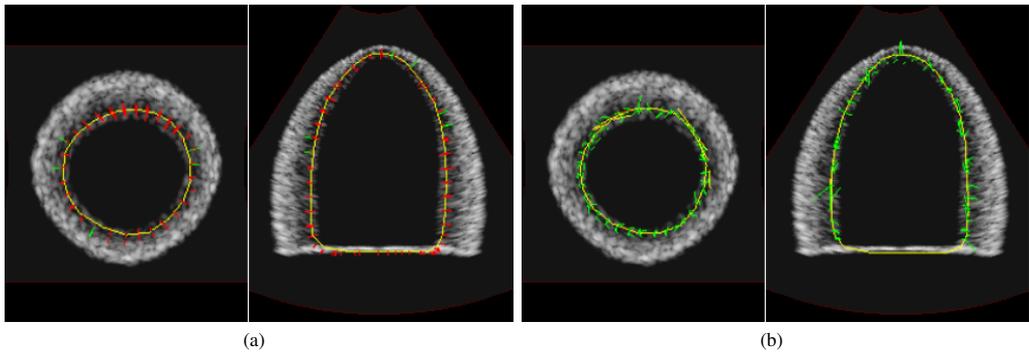


Figure 2: Example screenshots showing typical frame-to-frame displacement vectors from edge-detection (a) and speckle-tracking (b).

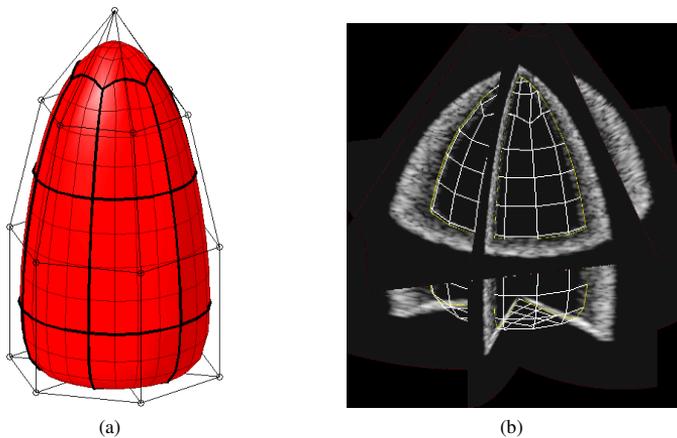


Figure 3: The topology of the subdivision model (a), as well as a rendering of the same model fitted to the endocardium during tracking (b).

Usage of unit vectors in x , y and z -direction for displacing control vertices enables direct usage of the Jacobian matrices as measurement matrices in the Kalman filter, since $\mathbf{H}^T = [\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z]^T \mathbf{J} = \mathbf{J}$. A covariance matrix for the measurement can also be used instead of scalar measurement noise values if one desires to capture any non-isotropy in the spatial uncertainty of the displacement, but this is not done in this paper.

III. RESULTS

In order to evaluate the feasibility and performance of the method, both the proposed method, as well as the speckle-tracking method of [9] was applied to a 3D ultrasound simulation of an infarcted ventricle. The simulation served as ground truth on which to compare both methods against.

The 3D ultrasound simulation was based on a finite element simulation of a left ventricle with an antero-apical infarction [10]. Material points from the finite element simulation were fed as scatter points into a k -space ultrasound simulator [5] to create 3D ultrasound images with a realistic speckle pattern. Tracking was initialized automatically by using edge-detection to align the model to the endocardial boundary. After initialization, the model was moved 1 mm inside the myocardium, and edge-detection

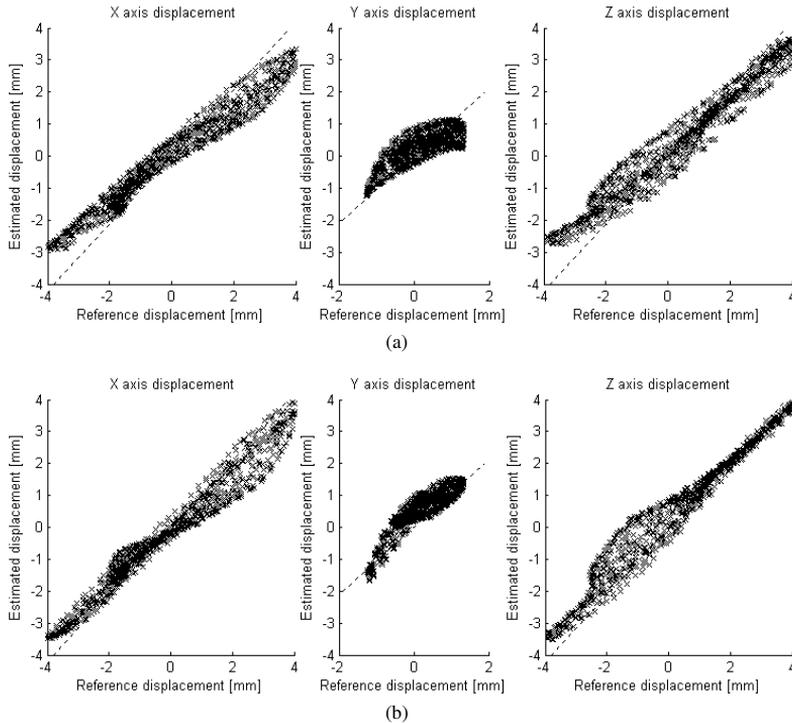


Figure 4: Scatter plot for estimated displacement vectors relative to ground-truth for tracking using speckle-tracking alone (a), and for edge-detection combined with speckle-tracking (b).

	Speckle-tracking alone	Edge-detection & speckle-tracking
X axis	0.663 mm (30.3%)	0.483 mm (22.1%)
Y axis	0.439 mm (60.2%)	0.433 mm (60.1%)
Z axis	0.613 mm (26.4%)	0.511 mm (22.3%)

Table I: Root-mean-square (RMS) errors in ED to ES displacement vectors on the tracked mesh, compared to ground truth. The relative errors are computed relative to the RMS of ground-truth displacement vectors.

was combined with speckle-tracking to track the speckle pattern from frame to frame, as can be seen in fig. 2.

Figure 4 shows scatter plots of the correlation of displacement vectors from end diastole (ED) to end systole (ES) on the tracked surface. The displacements are divided into their X, Y and Z components¹, and compared to ground truth values from the simulation. Both absolute and relative root-mean-square (RMS) error analysis on the scatter plots was conducted as shown in table I. The analysis shows improved agreement for the combined approach, with the scatter points closer to the unit line than with speckle-tracking alone. This improvement was mainly caused by less underestimation of large displacements.

After tracking, the subdivision surface were re-meshed into a quadrilateral mesh, and color-coded based on area strain ($\epsilon = (a - a_0)/a_0$), which is a measure of relative area change in each quadrilateral that combines the effect of longitudinal and circumferential strain. Figure 5 shows ES area strain values across the tracked surface using speckle-tracking alone, combined edge-detection and speckle-tracking, as well

¹The X axis corresponds to the *azimuth* axis of the probe, the Y axis image *depth* and the Z axis the *elevation* axis of the probe.

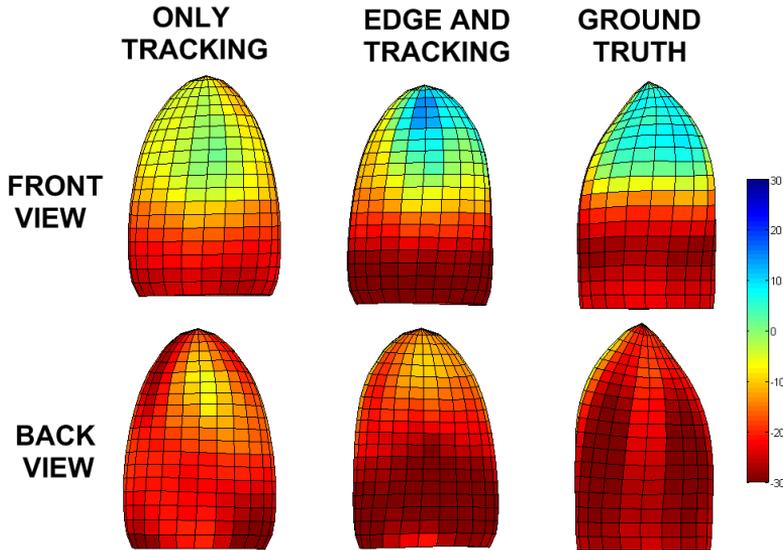


Figure 5: Front and back views of color-coded area strain meshes at ES based on speckle-tracking alone (left), edge-detection and speckle-tracking (center), as well as the ground truth (right).

as ground truth strain values. The infarcted regions are correctly identified by both methods, although speckle-tracking alone leads to underestimated and smeared out strain values compared to the combined approach. A slight misfit between the shape of the model and the ground truth, caused by the smooth subdivision model's inability to represent the sharp apex shape, was also observed.

Usage of the tracking sequence consumed approximately 68 ms of processing time per frame on a 2.2 GHz Intel Core 2 duo processor. Speckle-tracking alone required larger search windows to compensate for the lack of edge-detection updates, and therefore consumed 130 ms processing time per frame.

IV. DISCUSSION AND CONCLUSIONS

We have presented a new approach for model-based LV tracking based on [8] that combines edge-detection with speckle-tracking measurements to track material points over time in 3D echocardiography. The proposed method automatically initializes the model to the endocardium using edge-detection, and then uses a combination of speckle-tracking with edge-detection to track material deformations over time. Usage of this method might therefore enable rapid analysis of regional myocardial function.

The robustness of the Kalman filter framework enables fully automatic behavior without manual initialization, as shown in previous papers. The non-iterative formulation also makes the approach computationally efficient, although not as fast as [9]. This increase in running-time is caused by the extra edge-detection step, as well as usage of more surface points for speckle-tracking. Larger search windows was also used in the speckle-tracking alone approach to improve tracking accuracy at the expense of computational efficiency.

Results from the simulated infarcted heart clearly show that the combination of speckle-tracking with edge-detection leads to improved tracking accuracy over speckle-tracking alone. The agreement between ES displacement vectors and ground truth improves consistently with the proposed method, and the problem with underestimated strain values from [9] is no longer present. Instead, the center of the infarction exhibits magnified strain. The authors are not sure about the reason for this magnification, but suspects that it might be related to how the Kalman filter compensates for the interpolative effect of the subdivision surface in the model update step.

The combination of speckle-tracking with edge-detection offers several advantages compared to using speckle-tracking alone. Usage of edge-detection will correct for shape changes as well as gross global deformations, such as the movement of the basal plane, prior to the speckle-tracking step. This helps reduce the surface-normal component of the inherent drift associated with sequential block-matching over time, making tracking more robust. It also enables smaller search windows to be used, since speckle-tracking is only used to correct for residual deformations, after edge-detection is performed. Smaller search windows both increases computational efficiency, and offers more robust tracking due to fewer local minimums in the block-matching.

With feasibility of the method demonstrated, the next step will be to test it in more simulations, as well as in in-vivo data. The latter will pose additional challenges, due to the noisy nature of echocardiography, as well as the range of artifacts degrading image quality. Approaches to further reduce drift and improve accuracy should therefore be investigated. Currently, tracking is performed sequentially, from one frame to the next. This can be extended with bidirectional tracking, using both a forward and backward Kalman filter to improve tracking accuracy and reduce drift.

REFERENCES

- [1] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.
- [2] J. D'hooge, B. Bijnens, J. Thoen, F. Van de Werf, G.R. Sutherland, and P. Suetens. Echocardiographic strain and strain-rate imaging: a new tool to study regional myocardial function. *Medical Imaging, IEEE Transactions on*, 21(9):1022–1030, Sep 2002.
- [3] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, November 1978.
- [4] A. Elen, D. Loeckx, H. F. Choi, H. Gao, P. Claus, F. Maes, P. Suetens, and J. D'hooge. P4a-5 3D cardiac strain estimation using spatio-temporal elastic registration: In silico validation. *Ultrasonics Symposium, 2007. IEEE*, pages 1945–1948, 28-31 Oct. 2007.
- [5] T. Hergum, J. Crosby, M.J. Langhammer, and H. Torp. The effect of including fiber orientation in simulated 3D ultrasound images of the heart. *Ultrasonics Symposium, 2006. IEEE*, pages 1991–1994, 2-6 Oct. 2006.
- [6] L. D. Jacobs, I. S. Salgo, S. Goonewardena, L. Weinert, P. Coon, D. Bardo, O. Gerard, P. Allain, J. L. Zamorano, L. P. de Isla, V. Mor-Avi, and R. M. Lang. Rapid online quantification of left ventricular volume from real-time three-dimensional echocardiographic data. *European Heart Journal*, 27:460–468, November 2006.
- [7] J Meunier. Tissue motion assessment from 3D echographic speckle tracking. *Physics in Medicine and Biology*, 43(5):1241–1254, 1998.
- [8] F. Orderud and S. I. Rabben. Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces. *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, 2008.
- [9] Fredrik Orderud, Gabriel Kiss, Stian Langeland, Espen W. Remme, Hans G. Torp, and Stein I. Rabben. Real-time left ventricular speckle-tracking in 3d echocardiography with deformable subdivision surfaces. In *MICCAI 2008 Workshop on Analysis of Functional Medical Images*, pages 41–48, 2008.
- [10] Espen Remme and Otto Smiseth. Characteristic strain pattern of moderately ischemic myocardium investigated in a finite element simulation model. In *Functional Imaging and Modeling of the Heart*, pages 330–339, 2007.

Automatic Coupled Segmentation of Endo- and Epicardial Borders in 3D Echocardiography

Fredrik Orderud*, Gabriel Kiss* and Hans G. Torp*

*Norwegian University of Science and Technology (NTNU)

Abstract

In this paper, we present an extension of a computationally efficient Kalman filter based tracking framework to allow simultaneous tracking of several deformable models in volumetric data. The models are coupled through shared transform nodes in a hierarchical structure to enforce common position, orientation and scaling, but are allowed to alter shape independently. Automatic tracking of the endo- and epicardial border using two Doo-Sabin subdivision surfaces in 3D echocardiography serves as exemplary application.

Fully automatic endo- and epicardial surface is demonstrated in a simulation and 5 in-vivo recordings of high image quality. The estimated myocardial volumes is overestimated by 4.2 ml (7.0%) compared to ground truth in the simulation, whereas the volumes in the in-vivo recordings are on average underestimated by 10.7 ml (8.0%) compared to independent reference segmentations.

I. INTRODUCTION

Left ventricular (LV) mass and volume has been proven to be an important precursor for a variety of conditions such as cardiomyopathy, hypertension, valvular disease [8], as well as a predictor of prognosis.

The emergence of 3D ultrasound has enabled real-time volumetric imaging of the heart. Several methods for 3D segmentation have been proposed, most intended for the detection of the endocardium for volume measurement. However, simultaneous detection of endo- and epicardium also allows for estimation of myocardial volume, and automatic wall thickening analysis.

The feasibility of measuring LV thickness in 3D echocardiography was proven by Hubka et al. [5], which employ manually traced contours, on data to reconstruct smooth subdivision surface of the endo- and epicardium separately. Walimbe et. al [11] presented a fully automatic method, which uses deformable models coupled with an iterative mesh refinement scheme that incorporates a-priori knowledge of endo- and epicardial surfaces. Results on MRI data have been presented in [6][1]. More recently, [10], [9] validated myocardial volumes from semi-automatic endo- and epicardial segmentation against MRI. However to our knowledge, most of these approach require extensive user interaction to achieve sufficient accuracy. They are also computationally intensive, which prevents them from operating in real-time.

In this paper, we extend previous work on 3D endocardial segmentation [7] with support for simultaneous tracking of several deformable models, arranged in a hierarchy of geometric transforms. Two subdivision surfaces are used to represent the endo- and epicardium, and tracking is evaluated in a simulated phantom image, as well as in five 3D echocardiography recordings of high quality. The obtained myocardial volumes are compared against the ground-truth and volumes from a commercial segmentation tool.

II. METHODS

Simultaneous multi-model tracking poses additional challenges, compared to single-model tracking. This especially applies to the relative position and shape of the models, which has to be constrained to prevent intersections and other form of unrealistic results. In this paper, we employ a hierarchical approach

The authors want to thank Brage Amundsen (NTNU) for acquiring the in-vivo recordings, and Olivier Gérard (GE Vingmed Ultrasound) for conducting the reference volume measurements.

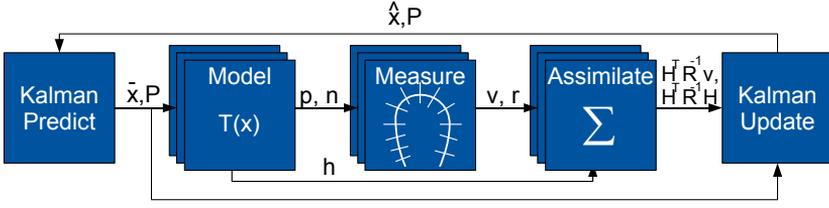


Figure 1: Flowchart over the 5 processing stages in the Kalman tracking framework.

to multi-model tracking, that constrains the position, size and orientation of the models relative to each other in a simple and intuitive way.

Deformable models are arranged in a hierarchical structure, much like the scene-graphs known from computer graphics[3]. The models becomes leaf nodes in this hierarchy, which are connected by geometric transforms as internal nodes. Geometric transforms are typically used for positioning, scaling and orientation, and the models beneath each transform automatically shares these properties. Geometric transforms may also be arranged beneath other transforms recursively, to form multi-level hierarchical structures. Fig. 2(a) shows an example of the simple tracking hierarchy used in this paper, that consists of an endocardium and an epicardium model, coupled together by a global pose transform, as well as a scale transform to enlarge the epicardium model relative to the endocardium model.

Both model and transform nodes will typically have parameters that affects either shape or pose. For tracking in a state-estimation framework, these parameters have to be concatenated into a composite state vector. For doing so, we follow a depth-first traversal convention to construct the state vector. This convention works recursively from the root transform, by first appending the transform parameters with parameters from each child model, and then appending recursively with concatenated parameters from each child transform. The example hierarchy in Fig. 2(a) will e.g. result in a state vector on the following form $\mathbf{x} = [\mathbf{x}_g^T \mathbf{x}_a^T \mathbf{x}_s^T \mathbf{x}_b^T]^T$.

Tracking is performed by following approximately the same 5-step procedure as in [7], but model construction and edge measurements will have to be performed independently for each model as shown in Fig. 1.

A. Deformable Subdivision Models

Both the endocardial and epicardial surface are modeled as smooth subdivision surfaces. In this paper, we use two Doo-Sabin [4] subdivision surface of identical topology, as shown in fig. 2. Both models are parametrized by 20 control vertices \mathbf{q}_i , that are allowed to move inwards and outwards along a displacement vector \mathbf{d}_i approximately equal to the surface normal to alter the shape. Fig. 2(a) shows the initial shape and topology for the subdivision models. In building the tracking hierarchy, the epicardial model is connected to a fixed scaling transform to enlarge it relative to the endocardial model. Both models as then connected to a global transform with 7 parameters for position, scaling and rotation. In total, this yields 47 parameters to be estimated based on edge measurements during tracking.

Edge-detection is performed relative to fixed parametric coordinates on the models. The distribution of these parametric coordinates (including patch number) $(u, v, c)_l$ are typically distributed evenly across the model surface as shown in Fig. 2(b), and remains fixed during tracking. This enables efficient precalculation of their basis functions during initialization as described in [7]. These basis functions are independent of the position of the control vertices, and can therefore be re-used during tracking to efficiently generate surface points regardless of shape deformations.

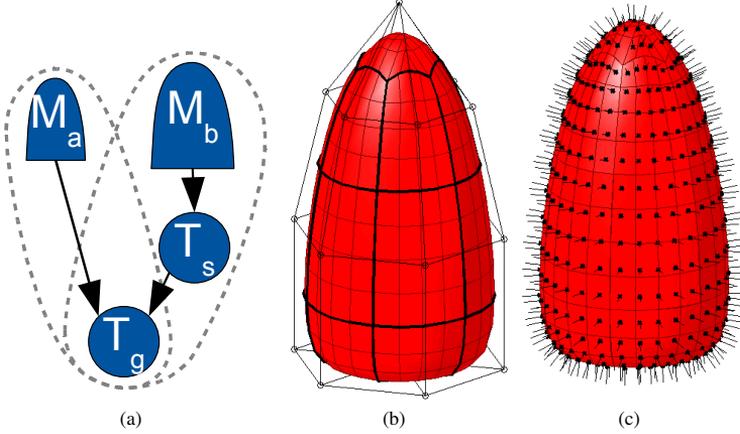


Figure 2: (a) The tracking hierarchy, which consists of two subdivision models (M_a and M_b), connected by a scale transform and a global transform (T_s and T_g). (b-c) Renderings of the subdivision model used for both endo- and epicardium: (b) shows the individual surface patches, enclosed in a wireframe grid of control vertices, while (c) shows the distribution of search profiles used for edge-detection.

B. Evaluation of Surface Points

The Kalman filter framework requires creation of a set of surface points \mathbf{p}_l with associated normal vectors \mathbf{n}_l and Jacobi matrices \mathbf{J}_l , based on a predicted state vector $\bar{\mathbf{x}}_l$. The creation of these objects can be performed efficiently following the steps outlined below:

- 1) Update position of control vertices \mathbf{q}_i based on the state vector: $\mathbf{q}_i = \bar{\mathbf{q}}_i + x_i \mathbf{d}_i$, where $\bar{\mathbf{q}}_i$ is the mean position of the control vertex and x_i is the parameter corresponding to this control vertex in the state vector for each control vertex. \mathbf{d}_i is the displacement direction for control vertex \mathbf{q}_i . The local state vector for the model is the concatenation of the state parameters for all control vertices $\mathbf{x}_l = [x_1, x_2, \dots, x_{N_l}]^T$. One can here choose to force certain vertices to remain stationary during tracking without altering the overall approach. This would both reduce the deformation space, as well as the number of parameters to estimate.
- 2) Calculate surface points \mathbf{p}_l as a sums of control vertices weighted with their respective basis functions within the surface patch of each surface point: $\mathbf{p}_l = \sum_{i \in C(c_l)} \mathbf{b}_i \mathbf{q}_i$.
- 3) Calculate surface normals \mathbf{n}_l as the cross product between the partial derivatives of the basis functions with regards to parametric values u and v within the surface patch of each surface point: $\mathbf{n}_l = \sum_{i \in C(c_l)} (\mathbf{b}_u)_i \mathbf{q}_i \times \sum_{i \in C(c_l)} (\mathbf{b}_v)_i \mathbf{q}_i$.
- 4) Calculate Jacobian matrices for the local deformations \mathbf{J}_l by concatenating the displacement vectors multiplied with their respective basis functions: $\mathbf{J}_l = [\mathbf{b}_{i_1} \mathbf{d}_{i_1}, \mathbf{b}_{i_2} \mathbf{d}_{i_2}, \dots]_{i \in C(c_l)}$. The Jacobian matrix will here be padded with zeros for columns corresponding to control vertices outside the region of support for the surface patch of each surface point.

Precomputation of basis functions enables the operations above to be performed very quickly, which is crucial for enabling real-time implementations.

C. Surface Point Transformation

Surface points originating from child models or transforms are transformed by means of a geometric transform \mathbf{T}_g to position, orient and scale the models.

We denote \mathbf{p}_l , \mathbf{n}_l and \mathbf{J}_l for the surface points created from the subdivision surface with local deformations $\mathbf{T}_l(\mathbf{x}_l)$. These points are subsequently transformed by means of a transform \mathbf{T}_g , that translates,

rotates and scales the model to align it correctly within the image volume. Surface points are trivially transformed using \mathbf{T}_g , whereas normal vectors must be transformed by multiplying with the normalized inverse spatial derivative of \mathbf{T}_g to remain surface normals after the global transform [2]:

$$\mathbf{p}_g = \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g) \quad (1)$$

$$\mathbf{n}_g = \left| \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \right| \left(\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \right)^{-T} \mathbf{n}_l \quad (2)$$

The Jacobian matrices for the composite deformations then becomes the concatenation of both global and local state-space derivatives. The local part is created by multiplying the 3×3 spatial Jacobian matrix for the global transform with the $3 \times N_l$ local Jacobian matrix for the deformable model, as follows from the chain-rule of multivariate calculus:

$$\mathbf{J}_g = \left[\frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{x}_g}, \mathbf{0}, \frac{\partial \mathbf{T}_g(\mathbf{p}_l, \mathbf{x}_g)}{\partial \mathbf{p}_l} \mathbf{J}_l, \mathbf{0} \right]. \quad (3)$$

The zero padding of the Jacobian matrices depends of the relative position of the model's parameters in the tracking hierarchy. The Jacobians for the models in Fig. 2(a) therefore becomes $\mathbf{J}_a = [\mathbf{J}_g \ \mathbf{J}_a \ \mathbf{0} \ \mathbf{0}]^T$ and $\mathbf{J}_b = [\mathbf{J}_g \ \mathbf{0} \ \mathbf{J}_s \ \mathbf{J}_b]^T$.

D. Kalman Tracking Framework

The overall tracking framework is based on the framework introduced in [7], with most steps very similar and therefore only briefly presented in this paper. The primary difference is that steps 2 through 4 are performed independently for each model in the tracking hierarchy as shown in Fig. 1, instead of only for a single model. The 5 steps can be summarized as:

- 1) Temporal prediction of the composite state vector $\bar{\mathbf{x}}_{k+1} = f(\hat{\mathbf{x}}_k, \mathbf{x}_0)$ based on the updated state from previous frame and a prediction function f , with associated increase in the covariance matrix. The temporal function is typically a linear autoregressive model.
- 2) Evaluation of tracking points \mathbf{p} , normal vectors \mathbf{n} and Jacobian matrices \mathbf{J} for all models in the tracking hierarchy, based on the predicted state $\bar{\mathbf{x}}_k$ as described in sections II-B and II-C.
- 3) Detection of normal displacement measurements v , measurement noise r and measurement vectors $\mathbf{h} = \mathbf{n}^T \mathbf{J}$, based on edge detection in the image volume, relative to surface points from the predicted models.
- 4) Assimilate measurement results from each model by summing the results in information space: $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{h}_i r_i^{-1} v_i$, $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T$.
- 5) Compute an updated state estimate, based on the prediction and measurement information: $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1} \mathbf{v}_k$, $\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$.

The net result of this approach is that edge measurement performed on each model will primarily affect local shape parameters, as well as parameters for each transform above the model in the hierarchy. Changes in parameters to unconnected nodes or non-descendant nodes in the hierarchy, such as e.g. sibling models, will not be observable due to the sparse nature of the Jacobian matrices.

III. RESULTS

Tracking was validated in one 3D echocardiography simulation of an infarcted ventricle, as well as in 5 in-vivo recordings of high image quality (Vivid 7, GE Vingmed Ultrasound, Norway). Myocardial volume was computed for all frames in each recording, based on the volume difference between the two models. Semiautomatic segmentation of the endo- an epicardium was performed by an experienced independent operator in both end diastole (ED) and end systole (ES) for the in-vivo recordings using a modified version of the AutoLVQ tool (GE Vingmed Ultrasound, Norway). The average of the ED and

Recording	Estimated volume	Reference volume
1	95.1 \pm 7.9 ml	102.5 \pm 0.7 ml
2	136.0 \pm 9.9 ml	167.5 \pm 24.7 ml
3	114.2 \pm 5.1 ml	108.5 \pm 0.7 ml
4	134.9 \pm 7.0 ml	149.5 \pm 9.2 ml
5	135.7 \pm 5.4 ml	141.5 \pm 9.2 ml
simulation	64.2 \pm 1.7 ml	60.0 ml

Table I: The center column shows estimated myocardial volume (mean \pm STD) for each recording, based on endo- and epicardial tracking in one cardiac cycle. The right column shows the reference volumes (mean \pm STD), based on manual tracing in ED and ES.

ES myocardial volume served as reference for the experiment. The simulation had a known myocardial volume, that was used as reference.

Tracking was fully automatic, with the exact same initialization in all of the recordings, and consumed an average of 9.5 ms processing time per frame on a 2.2 GHz Intel core 2 duo processor. This enables it able to operate in real-time, potentially during image acquisition.

Fig. 3 shows screenshots of tracking in the simulation, one of the in-vivo recordings, as well as a normalized myocardial volume-plot for each of the in-vivo recordings. For the simulated data, the proposed method overestimated the myocardial volume on average with 4.2 ml (7.0%) compared to the simulation, with a mean computed myocardial volume 64.2 \pm 1.7 ml. On the in-vivo data the proposed method underestimated the myocardial volume on average with 10.7 ml (8.0%), with a mean myocardial volume of 123.2 \pm 18.2 ml when averaging across all recordings, whereas the average volume from the references was 133.9 ml. Table I shows the individual results within each recording. The 68% confidence interval (mean \pm STD) for all in-vivo volumes lie within the corresponding 68% confidence intervals from the reference tool.

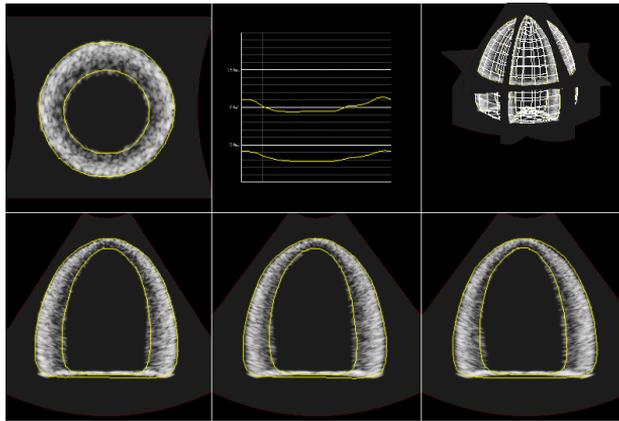
IV. DISCUSSION AND CONCLUSIONS

In this paper, we have presented an extension of an existing tracking framework to support simultaneous segmentation and tracking of several deformable models by means of hierarchical modeling. Coupled endo- and epicardial segmentation in 3D echocardiography served as exemplary application. Results from fully automatic tracking in one simulation as well as 5 recordings of high image quality demonstrates the feasibility of the approach for myocardial volume estimation. Results from the simulation showed low volume bias, while the in-vivo results are within the interval of myocardial masses reported by Takeuchi and Pouleur [10], [9].

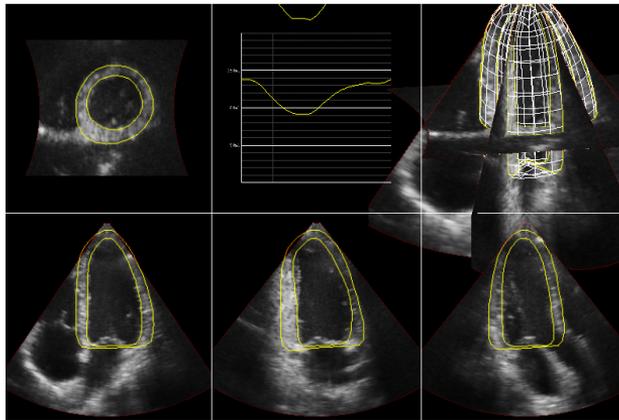
3D echo recordings of less than optimal image quality often suffers from areas of drop-out and other forms of image degradation that makes accurate edge-detection of especially the epicardium difficult. These difficulties can at least partly be alleviated by incorporating surface attractors, that are manually placed by the user. Such attractors can be implemented as virtual edge measurements towards specific spatial positions, indicated by the user in e.g. ED and ES, and would then assist the segmentation in areas of missing or degraded image data.

Currently, the shape of each model are updated independently. The global transform is the only parameters shared between the models. This independence makes it difficult to impose efficient regularization of inter-model thickness or to prevent intersection between the models. An alternative would be to extend the surface models with inherent thickness parameters, instead of using two coupled models. This could enable more robust and well behaved tracking in recordings of lower image quality.

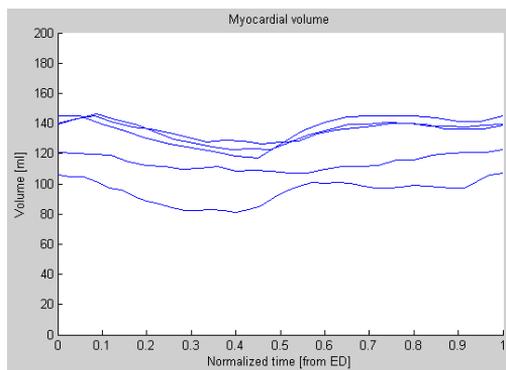
Multi-model tracking leads to long state-vectors and covariance matrices, since parameters from all models are concatenated together. Matrix operations in the Kalman filter generally scale in $O(N^3)$ for N states, which leads to poor computational efficiency when many models are combined. Possibilities for decomposing the Kalman filter into several smaller filters, with responsibility for smaller parts of the hierarchy should therefore be investigated.



(a)



(b)



(c)

Figure 3: Screenshots of tracking in the simulation (a), as well as one of the in-vivo recordings (b). The screenshots shows several slices through the volume, with intersection lines for the endo- and epicardium models, as well as a volume plot for each model. (c) Shows the computed myocardial volume in normalized time (from ED to ED) for each in-vivo recording.

REFERENCES

- [1] C. Alberola-Lopez, P. Casaseca-de-la Higuera, L. Cordero-Grande, and M. Martin-Fernandez. Endocardium and epicardium contour modeling based on markov random fields and active contours. In *Conf Proc IEEE Eng Med Biol Soc. 2006 ;1 (1):928-931*, 2006.
- [2] Alan H. Barr. Global and local deformations of solid primitives. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 21–30, New York, NY, USA, 1984. ACM Press.
- [3] H.G. Barrow and J.M. Tenenbaum. Computational vision. *Proceedings of the IEEE*, 69(5):572–595, May 1981.
- [4] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, November 1978.
- [5] M. Hubka, J. Lipiecki, E.L. Bolson, R.W. Martin, B. Munt, S.R. Maza, and F.H. Sheehan. Three-dimensional echocardiographic measurement of left ventricular wall thickness: In vitro and in vivo validation. *Journal of the American Society of Echocardiography*, 15:129–135, 2002.
- [6] N.M.I. Noble, D.L.G. Hill, M. Breeuwer, J.A. Schnabel, D.J. Hawkes, F.A. Gerritsen, and R. Razavi. Myocardial delineation via registration in a polar coordinate system. In *Lecture Notes in Computer Science, Medical Image Computing and Computer-Assisted Intervention, MICCAI*, 2002.
- [7] F. Orderud and S. I. Rabben. Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces. *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, 2008.
- [8] J.K. Perloff. Development and regression of increased ventricular mass. *Am J Cardiol.*, 50(3):605–11, 1982.
- [9] Anne-Catherine Pouleur, Jean-Benoit le Polain de Waroux, Agnes Pasquet, Bernhard L Gerber, Olivier Gerard, Pascal Allain, and Jean-Louis Vanoverschelde. Assessment of Left Ventricular Mass and Volumes by Three-Dimensional Echocardiography in Patients with or without Wall Motion Abnormalities: Comparison against Cine Magnetic Resonance Imaging. *Heart*, page hrt.2007.123711, 2007.
- [10] Masaaki Takeuchi, Tomoko Nishikage, Victor Mor-Avi, Lissa Sugeng, Lynn Weinert, Hironi Nakai, Ivan S. Salgo, Olivier Gerard, and Roberto M. Lang. Measurement of left ventricular mass by real-time three-dimensional echocardiography: Validation against magnetic resonance and comparison with two-dimensional and m-mode measurements. *Journal of the American Society of Echocardiography*, 21(9):1001–1005, September 2008.
- [11] V. Walimbe, V. Zagrodsky, and R. Shekhar. Fully automatic segmentation of left ventricular myocardium in real-time three-dimensional echocardiography. In *Medical Imaging 2006: Image Processing, Proc. of SPIE Vol. 6144, 61444H*, 2006.

Automatic Alignment of Standard Views in 3D Echocardiograms Using Real-time Tracking

Fredrik Orderud^a, Hans Torp^a and Stein Inge Rabben^b

^aNorwegian University of Science and Technology (NTNU), Trondheim, Norway;

^bGE Vingmed Ultrasound, Oslo, Norway

ABSTRACT

In this paper, we present an automatic approach for alignment of standard apical and short-axis slices, and correcting them for out-of-plane motion in 3D echocardiography. This is enabled by using real-time Kalman tracking to perform automatic left ventricle segmentation using a coupled deformable model, consisting of a left ventricle model, as well as structures for the right ventricle and left ventricle outflow tract. Landmark points from the segmented model are then used to generate standard apical and short-axis slices. The slices are automatically updated after tracking in each frame to correct for out-of-plane motion caused by longitudinal shortening of the left ventricle.

Results from a dataset of 35 recordings demonstrate the potential for automating apical slice initialization and dynamic short-axis slices. Apical 4-chamber, 2-chamber and long-axis slices are generated based on an assumption of fixed angle between the slices, and short-axis slices are generated so that they follow the same myocardial tissue over the entire cardiac cycle. The error compared to manual annotation was 8.4 ± 3.5 mm for apex, 3.6 ± 1.8 mm for mitral valve and $8.4 \pm 7.4^\circ$ for apical 4-chamber view. The high computational efficiency and automatic behavior of the method enables it to operate in real-time, potentially during image acquisition.

1. INTRODUCTION

Numerous tools for quantitative analysis of 3D echocardiograms have appeared over the last years, especially for left ventricular assessment. But despite all this innovation, visual wall-motion analysis still remains one of the most important clinical procedures in assessment of cardiac function. Traditionally, assessment of cardiac segments were time consuming, since several apical and parasternal 2D recordings had to be acquired independently, and examined successively to assess regional cardiac function.

With 3D echocardiography, arbitrary image slices can be extracted from acquired volumes. 3D echocardiography scanners incorporate tools to align and extract non-foreshortened apical 4-chamber (A4C), 2-chamber (A2C) and long-axis slices (LAX), as well as several short-axis slices. All cardiac segments can then be visualized simultaneously, based on a single 3D recording with sector wide enough to cover the entire ventricle.¹

Existing tools require manual alignment of the left ventricular long-axis, which increases the examination time. More seriously, the resulting slices remain at fixed spatial positions within

the image volume throughout the cardiac cycle. This means that the myocardial tissue being displayed differs during the cardiac cycle. This is especially a problem for basal short-axis slices, where the longitudinal shortening may be up to 1.2 cm during the cardiac cycle. The resulting out-of-plane motion can give rise to artificial wall thickening unrelated to cardiac contraction, and basal slices can end up in the atria at systole.

Several approaches for automatic identification of the left ventricular orientation have previously been published. Veronesi have published an optical-flow approach for measurement of the long-axis,² and Stralen have published a similar paper,³ based on a combination of Hough-transform with dynamic programming. In addition, some approaches that also identify apical view orientation in addition to the long-axis have also been published. Lu has presented a database-driven approach for the detection of standard view planes,⁴ and Leung have presented an approach for registration of stress echocardiography that also identify left ventricular orientation.⁵ All of these studies did, however, report computational costs in the order of minutes per cardiac cycle, which makes them unsuitable for real-time operation. The results published by Veronesi² were also limited to analyzing the length of the long-axis, whereas Stralen also compared the long-axis angle.³ None of the latter papers reported any position error for apex or base. Instead, they reported aggregate results based on comparison in every frame in each recording in the dataset, which makes comparison difficult.

In this paper, we present a method for automatically generating anatomically corrected short-axis slices, as well as automatically aligned standard apical views. The short-axis slices are evenly spaced between the apex and base of a fitted deformable model, whereas the apical view are aligned based on the orientation of the model, with a fixed angle between the slices. The slices are also automatically corrected for out-of-plane motion to ensure that the same myocardial tissue is imaged throughout the cardiac cycle. Due to the high computational efficiency, the method is able to operate in real-time, potentially during image acquisition.

2. METHODS

Automatic alignment of standard views are achieved by fitting several coupled deformable models to cardiac structures using a computationally efficient tracking framework previously described.^{6,7} This framework used an extended Kalman filter⁸ to perform temporal predictions, and assimilate edge-detection measurements from each model to compute a Bayesian least squares fitting of the models in a non-iterative fashion. Landmarks are then extracted from the fitted models, and subsequently used as basis for the extraction of aligned standard views.

2.1 Coupled deformable models

Automatic alignment of cardiac views requires information about both the ventricular long-axis, as well as the circumferential orientation of the heart. Usage of a left ventricle (LV) model alone is not believed to suffice, since circumferential information extracted from such a model would have to be based solely on the asymmetrical properties of the shape, which can vary greatly between subjects and depending on pathology. An alternative approach is to use coupled models to simultaneously track several cardiac structures. This allows for more reliable assessment of orientation, by computing the angle between the LV model and the different structures.

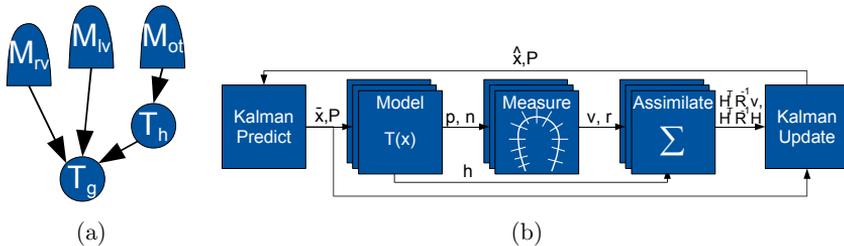


Figure 1: (a) Tracking hierarchy for the deformable models, showing the global transform T_g , LV model M_{lv} , RV-sail M_{rv} , hinge transform T_h , as well as the LV outlet tract M_{ot} . (b) Flowchart over the Kalman tracking framework.

To ensure clear and unambiguous detection of both the long-axis and circumferential orientation, we therefore combine a LV model with a sail-like structure for the inferior right ventricular wall (RV), as well as a tube-like structure for the left ventricular outflow tract (OT). A deformable Doo-Sabin subdivision surface⁹ is used as LV model, as described in.⁶ For the RV, the inferior RV wall is selected since this is the part of the right ventricle that is usually most visible, whereas the anterior wall often suffers from drop-out. All models share a global transform (G) for translation, rotation and scaling. The OT model is also addition connected to a hinge transform (H), which allows the model to rotate to adapt to inter-subject differences in anatomy for the outlet tract.

Fig. 1(a) illustrates how the models are arranged in relationship to one another in a tracking hierarchy, whereas Fig. 2(b) illustrates the relative geometric configuration between the models after fitting in a typical recording.

A state-space representation of the tracking hierarchy can be constructed by concatenating the parameters from all transforms and models into a state vector, as described in.⁷ The RV sail and OT cylinder does not have any shape parameters and are only affected by their associated transforms, so the concatenated state vector therefore becomes $\mathbf{x} = [\mathbf{x}_g^T \quad \mathbf{x}_{lv}^T \quad \mathbf{x}_h^T]^T$.

2.2 Tracking framework

The overall tracking framework for coupled models is based on the framework introduced in,^{6,7} with most steps very similar and therefore only briefly presented in this paper. The primary difference is that steps 2 through 4 are performed independently for each model in the tracking hierarchy as shown in Fig. 1(b), instead of only for a single model. The 5 steps can be summarized as:

1. Temporal prediction of the composite state vector $\bar{\mathbf{x}}_{k+1} = f(\hat{\mathbf{x}}_k, \mathbf{x}_0)$ based on the updated state from previous frame and a prediction function f , with associated increase in the covariance matrix. The temporal function is typically a linear auto-regressive model.
2. Evaluation of surface points \mathbf{p} , normal vectors \mathbf{n} and Jacobian matrices \mathbf{J} for all models in the tracking hierarchy, based on the predicted state $\bar{\mathbf{x}}_k$ as described in.⁶

3. Detection of normal displacement measurements v , measurement noise r and measurement vectors $\mathbf{h} = \mathbf{n}^T \mathbf{J}$, based on edge detection in the image volume, relative to surface points from the predicted models.
4. Assimilate measurement results from each model by summing the results in information space: $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{v} = \sum_i \mathbf{h}_i r_i^{-1} v_i$, $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} = \sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T$.
5. Compute an updated state estimate, based on the prediction and measurement information: $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \hat{\mathbf{P}}_k \mathbf{H}^T \mathbf{R}^{-1} \mathbf{v}_k$, $\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$.

Tracking can be performed fully automatic, and initialized by placing a model with average shape in the center of the image sector. Edge-detection measurement are performed in each frame to detect the endocardial wall in search normals distributed evenly across the surface. Parameters for the shape of the model is combined with parameters for global translation, rotation and scaling to form a state-space representation of the segmentation problem.

2.3 Model-based alignment

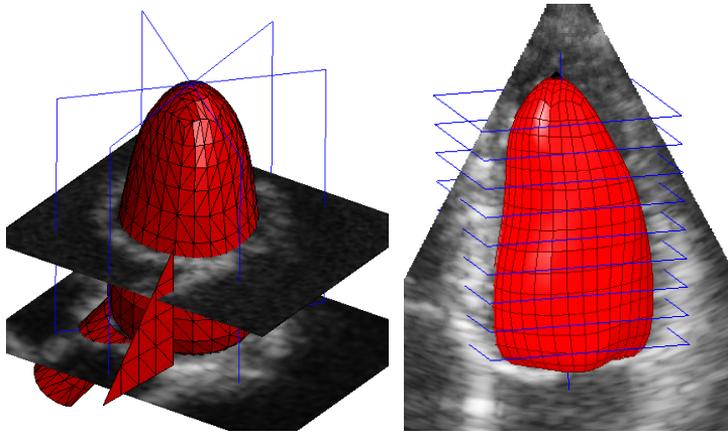
Landmark points from the fitted models are used to generate standard apical and short-axis slices. The slices are automatically updated after tracking in each frame to correct for out-of-plane motion caused by longitudinal shortening of the left ventricle.

During tracking, landmark points from apex and base of the LV model are extracted from the segmented model after fitting in each frame. The angles between the LV model, the RV sail and OT cylinder is also computed to infer circumferential orientation of the heart. This orientation is then used to automatically generate standard apical 4-chamber, 2-chamber and long-axis views centered through the apex-base long-axis vector as seen in figure 2(a). The views are aligned circumferentially oriented based on an assumption of a 60 degree angle between the slices. In addition, evenly distributed short-axis slices orthogonal to the apex-base long-axis are generated, as seen in figure 2(b). The position of these slices is updated after tracking in each frame to correct the slices for out-of-plane motion caused by longitudinal shortening of the left ventricle.

3. RESULTS

The real-time tracking framework was used to perform left ventricular segmentation in a collection of 35 3D echocardiography recordings, preselected so that over 70% of the myocardium was visible (38% exclusion rate). Furthermore, they were all acquired with the convention that the azimuth view shown on the screen during volume acquisition should approximately resemble a A4C view to limit the inter-recording variability in probe orientation.

Tracking was implemented to process the acquired spherical grayscale data directly, and consumed approximately 7 ms processing time per frame (2.16GHz Intel core 2 duo processor). This makes the framework capable of operating in real-time. Automatically generated views and landmarks from the fitted models in all 35 recordings were compared to landmarks from manual image annotation at end-diastole conducted by an independent operator.



(a) Apical alignment

(b) Short-axis alignment

Figure 2: Example illustrations of automatic model-based alignment: (a) Extraction of standard apical views based landmarks from the coupled LV-models. (b) Extraction of short-axis slices, based on the long-axis of the LV model.

3.1 Alignment examples

Landmarks from apex and base was then retrieved from the fitted models in all frames to align the short-axis slices. Figure 3 compares the results of using fixed short-axis slices with dynamic slices that compensate for out-of-plane motion between end-diastole (ED) and end-systole (ES) in an example recording. Notice the distinctive difference in the basal slices, which when not corrected images the atria during systole. For mid-wall slices, the same part of the papillary muscles appear to be tracked in the dynamic slices. This is contrasted by the fixed mid-wall slices, where the papillary muscles move in and out of the slice.

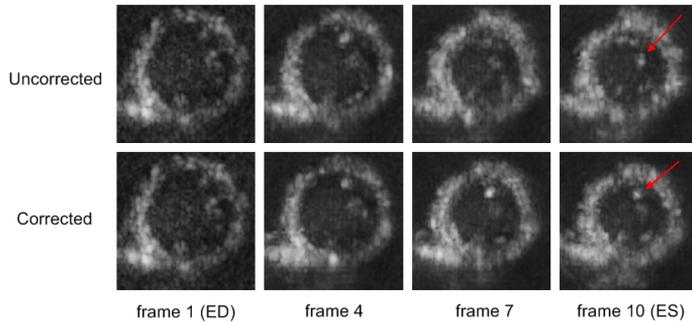
Figure 4 shows an example of automatically extracted standard 4-chamber, 2-chamber and long-axis apical views, based on landmarks from the fitted models. The figure also shows detected long-axis and circumferential orientation (yellow), compared to the long-axis and orientation from manual image annotation (white).

3.2 Alignment errors

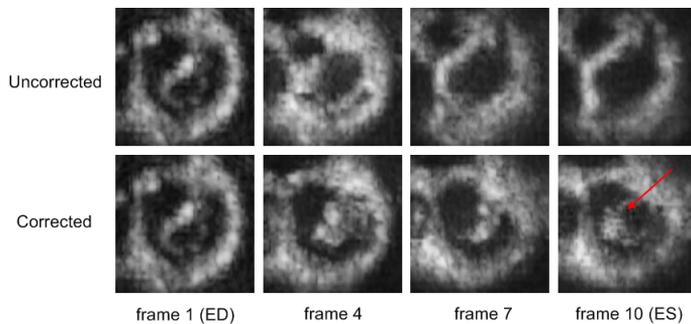
Figure 1 and 2 shows the results of quantitative comparison of the automatically extracted views and landmark points to the manually annotated reference. Position errors are given in absolute Cartesian distances, and angle errors are given as absolute rotation angle differences for a plane through the long-axis. Bias values are also reported for the average signed angle errors, and say something about consistent deviation in one direction.

4. DISCUSSION

A novel approach for automating and improving the generation of standard apical and anatomic short-axis slices from 3D echocardiograms has been presented. Usage of the approach is believed



(a) Mid-wall slices



(b) Basal slices

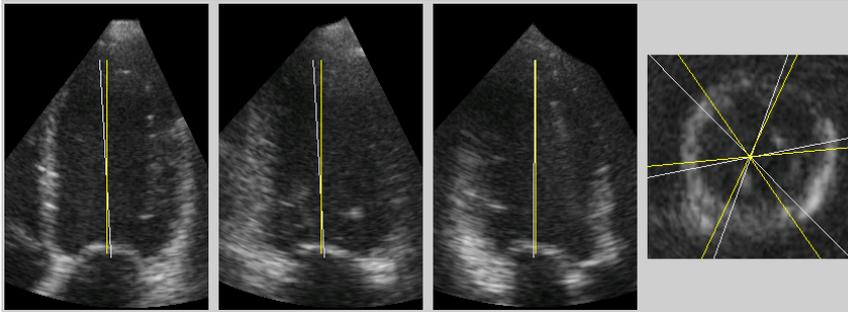
Figure 3: Example uncorrected and corrected short-axis slices from the mid-wall (a) and base (b) of the left ventricle. Notice how the papillary muscles (arrows) appear more similar in the corrected mid-wall slices, and how the mitral valve (red arrow) is followed in the corrected basal slices.

	Apex	Base/Mitral valve
Position error	8.4 ± 3.5 mm	3.6 ± 1.8 mm

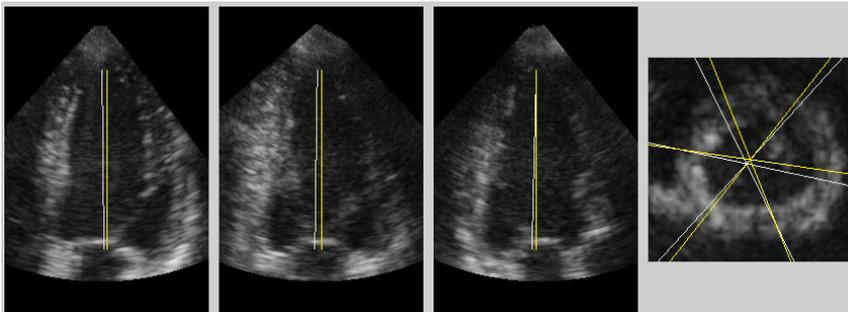
Table 1: Position errors for the apex and base compared to manual landmarks. Values are expressed as mean absolute distance \pm SD in millimeters.

	A4C	A2C	LAX
Rotation error	$8.4 \pm 7.4^\circ$	$8.0 \pm 7.6^\circ$	$11.8 \pm 8.5^\circ$
Bias	0.3°	1.1°	-7.9°

Table 2: Rotation errors for the standard apical views compared to views inferred from manual landmarks. Values are expressed as mean absolute error \pm SD in degrees. Bias is the average signed angle error.



(a)



(b)

Figure 4: Two examples of extraction of standard 4CH, 2CH and LAX apical views, based on landmarks from the fitted model. Projected landmark lines from apex to base are superimposed on the apical views, and slice-direction lines are shown in the short-axis view. Yellow lines are from the proposed method, whereas reference lines are show in white.

Alignment error	Apex	Mitral valve	A4C	A2C	LAX
Proposed	8.4 ± 3.5 mm	3.6 ± 1.8 mm	$8.4 \pm 7.4^\circ$	$8.0 \pm 7.6^\circ$	$11.8 \pm 8.5^\circ$
Lu 2008 ⁴	4.5 ± 3.5 mm	3.6 ± 3.1 mm	$13.2 \pm 12.5^\circ$	$15.2 \pm 13.0^\circ$	$14.5 \pm 13.2^\circ$
Leung 2008 ⁵	7.6 ± 4.8 mm	4.5 ± 2.9 mm	$6.3 \pm 4.6^\circ$	N/A	N/A

Table 3: Comparison of results to known prior art on detection of standard views.

to both automate positioning of standard apical and short-axis slices, as well as reducing the problem of out-of-plane motion experienced in short-axis slices.

Results from a dataset of 35 recordings demonstrates the feasibility of this approach, both for aligning apical and short-axis slices. For short-axis alignment, slices from an example recording clearly shows the advantage of correcting the short-axis slices for out-of-plane motion in imaging the same myocardial tissue throughout the cycle. This effectively prevents basal short-axis slices from dropping down into the atria during systole, and papillary muscles appear more similar from frame to frame in corrected mid-wall slices. Example slices from two example recordings also shows how alignment of apical A4C, A2C and LAX can be performed based on an assumption of fixed angle between the slices, with comparison to manually annotated landmarks. Furthermore, a previous study⁶ has shown that the Kalman tracking approach successfully tracked and segmented the left ventricle in 21 out of tested 21 3D echocardiograms. Hence, high robustness has previously been demonstrated.

Based on the results in table 1, one can see that basal/mitral valve detection is more accurate than apex detection. This corresponds well with the fact that manual apex identification is considered more difficult than manual mitral valve identification. As for rotations, table 2 shows that the assumption of 60 degree angle between the apical slices does not seem to hold for the LAX slice, which has much higher angle bias than the other apical slices. Adjusting the angle of the LAX slice relative to the other slices would reduce the alignment error, and should therefore be considered.

Compared to the results reported by Lu,⁴ the rotation errors for the apical views in this paper are smaller. The error for the basal landmark is similar, while the error for the apical error is larger compared to.⁴ It should be noted that Lu used a much larger dataset (N=326) and undisclosed exclusion criteria, which makes a fair comparison difficult. Compared to the results reported by Leung⁵ (N=20), the apex and base position errors are smaller, whereas Leung report better A4C angle agreement. It should be noted that the aim of Leung’s paper was volume registration of stress 3d echocardiography, instead of detection of standard views, which might have influenced the results. Table 3 provides a comparison to the results to Lu and Leung.

REFERENCES

1. Sugeng, L., Weinert, L., and Lang, R. M., “Left ventricular assessment using real time three dimensional echocardiography,” *Heart* **89**(90003), 29iii–36 (2003).
2. Veronesi, F., Corsi, C., Caiani, E., Sarti, A., and Lamberti, C., “Tracking of left ventricular long axis from real-time three-dimensional echocardiography using optical flow techniques,” *Information Technology in Biomedicine, IEEE Transactions on* **10**, 174–181 (Jan. 2006).

3. van Stralen, M., Leung, K., Voormolen, M., de Jong, N., van der Steen, A., Reiber, J., and Bosch, J., "Time continuous detection of the left ventricular long axis and the mitral valve plane in 3-d echocardiography," *Ultrasound in Medicine & Biology* **34**, 196–207 (Feb. 2008).
4. Lu, X., Georgescu, B., Zheng, Y., Otsuki, J., and Comaniciu, D., "Autompr: Automatic detection of standard planes in 3D echocardiography," *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, 1279–1282 (May 2008).
5. Leung, K. Y. E., van Stralen, M., Nemes, A., Voormolen, M. M., van Burken, G., Geleijnse, M. L., ten Cate, F. J., Reiber, J. H. C., de Jong, N., van der Steen, A. F. W., and Bosch, J. G., "Sparse registration for three-dimensional stress echocardiography," *Medical Imaging, IEEE Transactions on* **27**, 1568–1579 (Nov. 2008).
6. Orderud, F. and Rabben, S. I., "Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces," *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on* (2008).
7. Orderud, F., Kiss, G., and Torp, H. G., "Automatic coupled segmentation of endo- and epicardial borders in 3D echocardiography," *to appear in Ultrasonics Symposium, 2008. IEEE* (2-5 Nov. 2008).
8. Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., [*Estimation with Applications to Tracking and Navigation*], Wiley-Interscience (2001).
9. Doo, D. and Sabin, M., "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design* **10**, 356–360 (Nov. 1978).