Engin Dikici

# Ultrasound Cardiac Modeling, Segmentation and Tracking

## Doctoral Thesis

Submitted for the Partial Fulfilment of the Requirements for the Degree of

## philosophiae doctor

Department of Circulation and Medical Imaging
Faculty of Medicine
Norwegian University of Science and Technology

October 10, 2012

# Abstract

Echocardiography plays a key role in assessing cardiac diseases, especially in evaluating left ventricular (LV) function. It enables real-time, non-invasive, and relatively low cost acquisition of cardiac images. Furthermore, it can be performed bedside using portable equipment. The latest generation of echocardiography scanners allows the acquisition of volumetric images of the heart in real time, which could further improve the accuracy of functional analysis. However, accurate and automated analysis of 3D+T recordings is a challenging task. This is due to echocardiography imaging artifacts (e.g. speckle noise, signal dropouts), and the necessity of computationally efficient algorithms to exploit the real-time nature of the modality.

A Kalman filter based tracking framework was previously proposed for automatic and real-time analysis of LV structures in 3D echocardiography recordings. The approach was validated for detection and tracking of the endo- and epicardial borders of the LV, and noteworthy results were reported. The main goal of this thesis has been to extend the existing framework with more advanced algorithms for improving endocardial border tracking accuracy. In this work:

- Advanced edge detection methods including graph-cut based, maximum likelihood, empirical Bayes, and generalized step criterion endocardial edge detectors have been introduced. In addition, a polynomial regression based method has been proposed to filter endocardial edge measurements.
- Biomechanically constrained tracking of multi-resolution Doo-Sabin surface models has been investigated; an isoparametric finite element analysis (FEA) approach for Doo-Sabin surface models, and modification of the tracking framework to use isoparametric FEA have been introduced.

The proposed endocardial edge measurement and biomechanically constrained tracking approaches were evaluated using manually segmented 3D echocardiography recordings provided by medical experts. The comparative analyses showed that:

- The graph-cut based edge detector improves endocardial detection accuracy of the tracking framework at end-diastole.
- The maximum likelihood and empirical Bayes edge detectors improve detection accuracy for the whole cardiac cycle, while introducing additional computational complexity.
- The generalized step criterion edge detectors enable real-time maximum likelihood detectors.
- The polynomial regression based edge filtering provides an intuitive controller for the tradeoff between edge detection bias and variance.
- Biomechanical constraints can significantly improve endocardium tracking accuracy of subdivision surface models with high control node resolutions.

# Preface

This thesis is submitted as partial fulfillment of the degree *Philosophiae Doctor* (PhD) at the Faculty of Medicine of the Norwegian University of Science and Technology (NTNU). The presented work was carried out under the supervisions of MSc, PhD Fredrik Orderud from GE Vingmed Ultrasound, and Professor Hans Torp from the Department of Circulation and Medical Imaging of NTNU, between November 2009 and October 2012.

# Acknowledgements

First of all, I would like to thank to my main supervisor Fredrik Orderud for his great guidance and assistance during my PhD work. Fredrik guided me towards interesting yet valuable research concepts, and provided excellent supervision throughout my doctoral studies. Furthermore, he generously allowed me to use his development and testing environments, and let me have an internship at GE Vingmed Ultrasound to improve my professional experience in the field.

I also want thank to my co-supervisor Hans Torp for his valuable suggestions and ideas in the medical image processing topics. He was always willing to assist and contribute to my research studies, and provided valuable reviews on my presentations and publications.

A special thanks goes to my co-author Sten Roar Snare for his valuable suggestions in discussions regarding isoparametric finite element analysis and Kalman smoothing techniques. Sten Roar and I spent lots of time in front of a white board to derive some equations that made significant contributions to my PhD.

I want to express my gratitude to Bo Henry Lindqvist for co-authoring one of my papers and teaching a great course in *Statistics*, which allowed me to use general statistical methods in my medical imaging research.

I would like to thank all my other co-authors for great cooperation and my fellow workmates at the department for the nice working environment. My appreciation also goes out to all the employees of GE Vingmed Ultrasound in Oslo for sharing their helpful ideas.

Next, I want to thank to Thomas O'Donnell, my mentor at SCR, and Ali Kemal Sinop, my friend whom I met at UPenn, for the valuable discussions we had on mathematics and computer science topics, which inspired my research.

The last but not the least, I would like to thank to my parents Atilla & Nuray Dikici, and my sister Birce Dikici for their love and support throughout my life. During my studies, they were always encouraging and helped keeping my motivation at the highest level.

# Table of Contents

# Abbreviations

| | |
|---|---|
| AGF | Active Geometric Functions. |
| ASPE | Absolute Surface Point Error. |
| AVE | Absolute Volume Error. |
| BBDMM | Baruch and Bar-Itzhack Direct Matrix Modifications. |
| BLUE | Best Linear Unbiased Estimator. |
| CAGD | Computer Aided Geometric Design. |
| CPDM | Control Point Distribution Model. |
| CV | Cross Validation. |
| EB | Empirical Bayes. |
| ED | End Diastole. |
| ES | End Systole. |
| EF | Ejection Fraction. |
| FEA | Finite Element Analysis. |
| LPR | Local Polynomial Regression. |
| LS | Least Squares. |
| LV | Left Ventricle. |
| MG | Maximum Gradient. |
| MFMC | Max Flow / Min Cut. |
| ML | Maximum Likelihood. |
| MRF | Markov Random Field. |
| MSE | Mean Square Error. |
| NURBS | Nonuniform Rational B-Splines. |
| PDF | Probability Distribution Function. |
| SSPE | Squared Surface Point Error. |
| STEP | Step Criterion. |
| SV | Stroke Volume. |

# Chapter 1

# Introduction

Engin Dikici
Norwegian University of Science and Technology, Trondheim, Norway

## 1.1 Background and Motivation

Cardiovascular diseases (CVDs) are the major causes of death for both men and women worldwide, with more than 17 million deaths from CVDs in 2008 [1]. It is estimated that CVDs will remain the main cause of death and will lead to approximately 23 million deaths in 2030. Rapid and portable tools for analyzing cardiac function will most likely be necessary for early detection and diagnosis of several types of CVDs. These tools will also need to be affordable, as 80% of CVD deaths take place in low-middle income countries with limited health care budgets [1].

Cardiac ultrasound, *echocardiography*, is an imaging modality that enables rapid, non-invasive and relatively low cost evaluation of cardiac function. It can be used bedside with portable equipment. Due to these advantages, echocardiography has been the method of choice in most situations requiring cardiac function analysis [2]. The latest generation of echocardiography scanners allows for acquisition of volumetric images of the heart in either real time or over a few heartbeats. Volumetric acquisition allows medical experts to visualize moving cardiac structures from arbitrary planes in real time. It can also improve the accuracy of functional analyses compared to the more traditional motion mode (M-mode) and two dimensional (2D) echocardiography [3–5].

Evaluation of left ventricular (LV) function is the most common reason for conducting an echocardiography examination. This is due to the fact that the structure and deformation pattern of the LV has a particular importance in the diagnosis of CVDs [6]. Segmentation of the LV endocardial borders in echocardiography recordings enables quantitative functional analysis of the heart by providing measurements such as stroke volume (SV) and ejection fraction (EF). Applications requiring a quantitative wall motion analysis, including the detection of myocardial ischemia [7, 8] and assessment of LV dyssynchrony [9, 10], also benefit from endocardial surface segmentation.

Manual segmentation of LV endocardial borders is a time-consuming process that is prone to poor reproducibility [11]. Availability of automated segmentation tools enables more rapid functional analysis and lowers the amount of inter- and intra-examiner variability during the analysis. However, automated segmentation of LV endocardial borders in echocardiography recordings is a challenging task due to reasons including (1) speckle noise, (2) shadowing that can result in missing boundaries, and (3) the existence of intra-cavity structures such as chordae tendineae, papillary muscles and valves [12] (see Figure 1.1). These challenges cause segmentation algorithms that perform well for other modalities, like computed tomography and magnetic resonance, to produce poor results for echocardiographic images. Real-time segmentation of the endocardial borders might also be desirable for invasive procedures and intensive care unit applications with monitoring requirements [13].

Automatic and semi-automatic segmentation algorithms for volumetric and time dependent (3D+T) echocardiography recordings have received considerable attention in recent years. Review articles from Noble & Boukerroui [14], and Leung & Bosch [15] provide excellent overviews of the research efforts in the field. Yet, only a few of the proposed algorithms can perform the segmentation in a rapid fashion, where each frame is processed in means of milliseconds or seconds. In [16], a one-step forward prediction approach using motion manifold learning was introduced for tracking endocardial borders. The method produced less than $1.5mm$ mean surface error, and segmented each 3D echocardiography frame in $1.5s$ (using an undefined system configuration). Yet, it relied on a comprehensive training dataset to capture the endocardial motion patterns of the novel cases. Rajchl et al. treated 3D endocardium segmentation as a binary graph partitioning problem [17]. To formulate an energy function, they used (1) two cost terms based on maximum log-likelihood of two Fisher-Tippett distributions, representing the inside and outside of the LV cavity classes, and (2) an additional cost term to constraint the model geometry. The energy function was minimized using a flow-maximization algorithm described in [18] that can be computed on a parallel computation platform. The approach produced $3.54mm$ mean endocardial surface error, and processed each frame in $100ms$ (using an undefined system configuration). In [19], an Active Geometric Functions (AGF) based

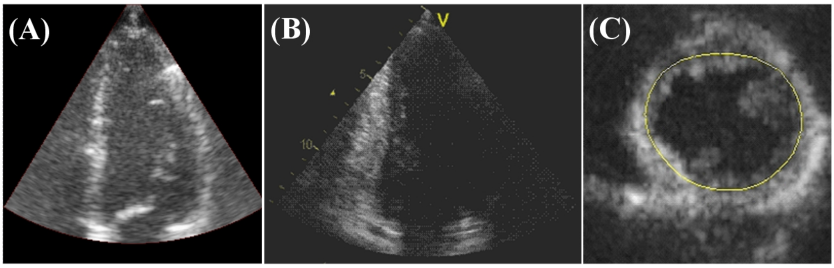


Figure 1.1: Echocardiography imaging challenges. (A) Strong speckle pattern and low contrast, (B) dropout of the anterior wall due to a long shadow, and (C) similar intensity myocardial wall and trabeculations *(the ground-truth endocardial border is shown in yellow)* cases are represented for three sample images.

framework was proposed for real-time segmentation of the endocardium in 3D+T echocardiography. In their study, a specific instance of AGF based on finite element modeling and Hermite endocardial surface descriptors was implemented and evaluated. The mean distance of AGF segmentation to manual tracings was reported as 4.00$mm$, and the processing time per frame was 32.9$ms$ (using a 2.0GHz Intel 32-bit CPU). Duan et al. introduced a semi-automated region-based endocardium tracking approach [20], where the endocardial borders at end-diastole (ED) and end-systole (ES) frames were manually delineated. Their algorithm propagated the segmentation results throughout the cardiac cycle using the estimated myocardial surface displacement fields computed between consecutive frames. The root mean square error for the tracked surfaces was 0.06, and the computation time for tracking a frame was 9$s$ (using a 2.4GHz AMD CPU). In a recent study, Barbosa et al. proposed a pipeline to track endocardial borders in 3D+T echocardiography recordings [21] using B-spline Explicit Active Surfaces [22]. It required six clicks per 3D volume to initialize an ellipsoid-shaped endocardial mesh for the first frame. The pipeline was evaluated for its volumetric measurement accuracy, and the Pearson product-moment correlation coefficients between the extracted and reference LV cavity volumes were reported as 0.963 and 0.947 for the ED and ES frames respectively. The total processing time, including manual initialization and automated tracking, for a 3D+T recording was 30$s$ (using a 2.8 GHz Intel Core i7 CPU).

State-space analysis using Kalman filtering has also been employed for detection and tracking of LV structures in time-dependent recordings. The approach uses a sequential prediction and update strategy, where shape deformations are first predicted by a kinematic model, followed by an update step based on information provided by image measurements. A Kalman filtering framework for tracking B-spline models was first introduced by Blake et al. [23–25]. This framework was adapted for LV tracking in long-axis 2D+T echocardiography in [26–28]. Orderud et al. extended the approach for real-time segmentation in 3D+T echocardiography recordings by tracking rigid ellipsoid [29], flexible spline-based [30] and Doo-Sabin subdivision LV models [31]. The Doo-Sabin subdivision surface tracking framework produced approximately 2.5$mm$ mean surface error for the tracking of endocardial borders in 3D echocardiography recordings, and processed each frame in only 7.5$ms$ (using a 2.80 GHz Intel Core 2 CPU). The approach was later utilized for a variety of tasks including coupled segmentation of endo- and epicardial borders [32], automated alignment of standard views [33], and cardiac strain assessment in 3D echocardiography [34].

The tracking framework proposed in [31] has several advantages: The method (1) provides a true real-time detection and tracking solution, which allows the monitoring of LV function in real time, (2) leads to comparable or better endocardial border tracking accuracy than other state-of-the-art approaches, and (3) does not require manual initialization; the tracked anatomical structure's borders are automatically located at run-time. However, the tracking framework could be further extended with more advanced algorithms to improve the accuracy of 3D+T echocardiography image analysis.

## 1.2 Aims of Study

The main goal of this thesis has been to improve the existing Kalman tracking framework [31] to enhance the accuracy of 3D+T echocardiography image analysis. The framework could benefit from improvements in both the edge detection and state prediction stages.

The tracking framework guides a predicted surface model toward a target object with the assistance of edge detection. A set of simple edge detectors had previously been applied in the framework to detect LV endocardial borders. These approaches were computationally efficient, yet they oversimplified the endocardial edge detection task. Therefore, one of the goals of this study has been to develop advanced endocardial edge detectors that can produce more accurate results while still allowing real-time or nearly real-time tracking.

The state prediction stage of the framework uses a motion model to predict the tracked surface's shape and position in a following frame. The tracking framework had employed an intuitive motion model performing a regularization toward an initial state. The work presented in this thesis has aimed to introduce a more realistic motion model that can factor in some of the biomechanical properties of the LV myocardium. A biomechanically constrained motion model can provide a more advanced regularization; hence, it can enable tracking that scales gracefully with higher resolution subdivision surface models to improve model fitting accuracy.

To summarize, this thesis aims to:

- Introduce edge detection methods that can perform more accurate endocardial edge measurements in either real time or near real time.

- Extend the framework with biomechanically constrained tracking that allows the usage of higher resolution subdivision surface models.

## 1.3 Summary of Presented Work

### 1.3.1 Contribution 1 - Graph Cuts Based Edge Measurements (Chapter 3)

Consistent endocardial border segmentation in 3D echocardiography is a challenging task. One of the major difficulties rises due to the fact that the trabeculated structure of the LV causes endocardial intensity profile characteristics to change over the cardiac cycle. Previously proposed edge measurement methods for the tracking framework process each surface position independently, without considering the other edge measurements performed around a surface model.

We treat endocardial edge detection as a binary graph partitioning task with the inside and outside of the LV cavity classes. The method first generates a graph using all intensity profiles collected around a surface model, then partitions the graph using a max flow/min cut algorithm (MFMC). We observed that for ES, the step criterion edge detector (STEP) [35] produces more accurate segmentation results than MFMC.

Therefore, we introduced a weighted combination of MFMC and STEP methods called a hybrid edge detector (Hybrid). Hybrid combines the responses of MFMC and STEP using weights determined by the size of the tracked endocardial mesh.

A set of 10 apical 3D echocardiography recordings was used to evaluate the proposed approaches. The set included 3 normal cases and 7 cases from patients with heart diseases. The STEP, MFMC and Hybrid edge detectors were each used in connection to the Kalman tracking framework described in [31]. The accuracies of the edge detectors were evaluated by comparing the extracted meshes against reference meshes segmented by a medical expert. The average unsigned distance between the tracked surface points and their estimated correct positions was computed for each method: STEP, MFMC and Hybrid produced 3.06$mm$, 2.62$mm$ and 2.50$mm$ point-to-point errors at ED, and 3.10$mm$, 3.70$mm$ and 3.02$mm$ point-to-point errors at ES respectively. The average signed ejection fraction errors were reported as $-6.2\%$, $-8.3\%$ and $-3.8\%$ for the STEP, MFMC and Hybrid methods respectively. The tracking framework processed each frame in 7.5$ms$ with STEP, 78$ms$ with MFMC, and 80$ms$ with Hybrid when executed on a 2.80 GHz Intel Core 2 Duo CPU. The results suggest that (1) the MFMC method segments the endocardial borders with higher accuracy at ED, while performing poorly at ES, and (2) combining multiple edge detectors could improve edge detection accuracy for all cardiac cycle.

This topic is described in the paper “Graph-Cut Based Edge Detection for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography”, accepted by IEEE Computing in Cardiology 2010.

### 1.3.2 Contribution 2 - Edge Measurement Filtering (Chapter 4)

Edge detection results produced by a rapid but relatively low accuracy edge detector can be filtered to generate more accurate results. In this study, a local polynomial regression based method (LPR) was introduced for filtering the step criterion endocardial edge measurements. The method first computes the STEP edges at evenly distributed positions around an endocardial surface model. Then, the detected STEP edges are filtered by an arbitrary order LPR with an Epanechnikov weighting kernel.

A set of 17 apical 3D echocardiography recordings, which included 10 normal cases and 7 cases from patients with heart diseases, was used for the evaluation. Local polynomial regression based edge filtering method was implemented for the first (LPR-1) and second order (LPR-2) polynomials. STEP, MFMC, LPR-1 and LPR-2 methods were each used in connection to the existing Kalman tracking framework. The outlier edge elimination feature of the tracker was turned off to fully observe the effects of filtering. The accuracies of the edge detectors were evaluated by comparing the extracted surfaces against the verified reference surfaces drawn by a medical expert. The average unsigned distance between the tracked surface points and the reference surfaces was computed for each method: STEP, MFMC, LPR-1 and LPR-2 produced 2.94$mm$, 2.38$mm$, 2.61$mm$ and 2.57$mm$ point-to-surface errors at ED, and 2.20$mm$, 2.46$mm$, 2.16$mm$ and 2.07$mm$ point-to-surface errors at ES respectively. The tracking framework processed each frame in 23.7$ms$ with LPR-1 and 40.8$ms$ with LPR-2 when executed on a 2.80 GHz Intel Core 2 Duo CPU. The results suggest that the LPR method can improve overall edge detection accuracy.

This project is described in the paper "Polynomial Regression Based Edge Filtering for Left Ventricle Tracking in 3D Echocardiography", accepted by Statistical Atlases and Computational Models of the Heart 2011.

### 1.3.3 Contribution 3 - Maximum Likelihood Edge Measurements (Chapters 5, 6 and 7)

Estimation accuracies of edge detection algorithms vary depending on both endocardial surface and cardiac cycle positions. The responses of multiple edge detectors can therefore be combined using a statistical learning approach to generate better endocardial edge estimates [36]. Thus, we proposed three maximum likelihood based approaches that utilize multiple simpler edge detectors.

First, we developed the best linear unbiased estimator (BLUE) that combines the responses of the maximum gradient (MG) [25], STEP and MFMC, which are referred to as the base detectors. The system (1) learns the statistical properties of the base estimators for each endocardial surface point at each cardiac cycle position, and (2) combines the base estimator responses linearly via weights inferred from the statistical properties. It was also shown that BLUE and the maximum likelihood (ML) estimator lead to identical solutions for the investigated edge detection problem if the base estimator error distributions are assumed to be Gaussian. A set of 18 apical 3D echocardiography recordings, including 10 normal cases and 8 cases from patients with heart diseases, was used during the evaluation. As BLUE requires a training process, it was validated using a 3-fold cross validation (CV). The average squared distance between the tracked surface points and the reference surfaces was computed for each method: MG, STEP, MFMC and BLUE produced $15.5mm^2$, $11.1mm^2$, $8.9mm^2$and $7mm^2$ squared point-to-surface errors at ED, $17.2mm^2$, $7.3mm^2$, $10.3mm^2$ and $8.2mm^2$ squared point-to-surface errors at ES respectively. The tracking framework processed each frame in $81ms$ with BLUE when executed on a 2.80 GHz Intel Core 2 Duo CPU. The algorithm is presented in the paper "Best Linear Unbiased Estimator for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography", published in the Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis 2012.

Next, we extended the ML (or BLUE) edge detector by incorporating contextual priors giving the probabilistic distribution of endocardial edges around a surface model. The method first employs a training stage to produce an ML model, giving the optimal base detector weights for each cardiac cycle and endocardial surface position. For a given novel case, the method (1) estimates ML edges using the learned ML model, (2) generates a contextual prior utilizing the estimated ML edges in an empirical fashion, and (3) fuses ML estimates with the contextual prior to find empirical Bayes (EB) endocardial edge estimates. The testing setup from the previously conducted ML study was used to evaluate the EB method (with similar recordings and CV folds). The average squared distance between the tracked surface points and the reference surfaces was computed: EB produced $6.11mm^2$ and $7.44mm^2$ squared point-to-surface errors at ED and ES frames respectively. The tracking framework processed each frame in $83ms$ with the EB method. The approach improved the tracking accuracy of the ML by taking advantage of the *Stein effect*. The Stein effect briefly asserts that the estimation accuracy can be improved using the information coming from the full model, which is the contextual prior in our study [37]. This project is described in the paper "Empirical Bayes Estimators for Endocardial Edge Detection in 3D+T Echocardiography", published in the Proceedings of the International Symposium on Biomedical Imaging 2012.

Both ML and EB improve the endocardial edge detection accuracy compared to sole applications of MG, STEP and MFMC. However, they are computationally expensive solutions processing each frame over $80ms$. Therefore, we developed another ML based approach with a new set of more efficient base detectors; the generalized step detectors. A generalized $k^{th}$ order STEP detector (1) fits multiple piecewise $k^{th}$ order polynomial functions to a given intensity profile, which is extracted on a tracked surface's normal direction, then (2) selects the optimal piecewise function in the least-squares (LS) sense to locate the endocardial edge position. The responses of multiple generalized detectors were combined using a similar approach as the previously developed ML method. The approach was validated using a set of 29 recordings including 10 normal cases and 19 cases from patients with heart diseases. An N-fold CV was applied for the evaluation of the ML method (STEP-ML) using $0^{th}$, $1^{st}$ and $2^{nd}$ order STEP as the base detectors. The average squared distance between the tracked surface points and the reference surfaces was computed for the STEP-ML and classic ML (ML) methods (ML uses MG, STEP and MFMC). STEP-ML and ML produced $8.1mm^2$ and $7.5mm^2$ squared point-to-surface errors at ED, $8.1mm^2$ and $7.5mm^2$squared point-to-surface errors at ES respectively. The tracking framework processed each frame in $46.6ms$ with the STEP-ML edge detector. STEP-ML produced comparable results with the previously defined ML approach while reducing the processing time for each frame over 42%. This topic is described in the paper "Generalized Step Criterion Edge Detectors for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography", accepted by Statistical Atlases and Computational Models of the Heart 2012.

### 1.3.4 Contribution 4 - Improved Physical Modeling for Multi-Resolution Segmentation (Chapters 8 and 9)

The Kalman tracking framework uses a motion model to predict the state of a tracked surface in following frames. An intuitive motion model performing a regularization toward an initial state was used in previous studies. We developed a physically constrained motion model that produces more accurate state estimations and enables tracking of higher resolution surface models.

The motion model definition requires a finite element analysis (FEA) to be performed for the tracked geometric model. The tracking framework described in [31] employs Doo-Sabin subdivision LV models. Therefore, we first introduced an isoparametric FEA method for the Doo-Sabin subdivision surface models equipped with thickness information. The proposed method describes basis functions for the Doo-Sabin limit surfaces using an iterative algorithm. Then, it utilizes the defined basis functions for the isoparametric formulation of shell elements during the physical simulation. A degenerated solids based approach is used to formulate the shell elements. The accuracy of the physical simulation can be adjusted using basis refinements without changing the model geometry or its parameterization. To evaluate the convergence properties of the proposed isoparametric FEA method, surface forces were applied to four distinct models at three resolution levels. Then, the deformed Doo-Sabin model surfaces were compared against corresponding deformed reference model surfaces. For a LV shaped model, the average unsigned distances between the deformed Doo-Sabin and reference models were reported as $2.4mm$, $0.6mm$ and $0.5mm$ for the non-refined, refined and doubly refined models respectively. The experiments show that the physical simulation accuracy of the Doo-Sabin surface models converges rapidly with that of high resolution finite element models, which use classic hexahedron and triangular prism elements. This project is described in the paper ”Isoparametric Finite Element Analysis for Doo-Sabin Subdivision Models”, accepted by Graphics Interface 2012.

Next, we extended the real-time Kalman tracking framework defined in [31] by employing biomechanically constrained state transitions. The proposed method first defines a stiffness matrix for the tracked Doo-Sabin surface model using the isoparametric FEA. Then, it reformulates the state prediction stage of the tracking framework to use the stiffness matrix information for performing physically constrained tracking. In this study, we also introduced an optional statistical model improvement stage for addressing the possible tracking problems due to poorly hypothesized model shape and FEA parameters (e.g. Young's modulus, Poisson's ratio). The approach was validated using a set of 29 recordings including 10 normal cases and 19 cases from patients with heart diseases. A custom LV model with a very low control node resolution was employed during the analyses to assess the effects of biomechanical constraints on multi-resolution models. The average unsigned distances between the tracked surface points and the reference surfaces were reported for the biomechanically constrained and non-constrained trackers. (1) The tracker with no biomechanical constraints produced $3.0mm$, $3.1mm$, and $3.5mm$ point-to-surface errors, and (2) the tracker with the biomechanical constraints produced $3.1mm$, $2.9mm$ and $2.9mm$ point-to-surface errors for the non-refined, refined and doubly refined models respectively. Our analyses show that the biomechanical constraints are necessary, especially when the tracked model has a high control node resolution. The statistical model improvements were also evaluated, and more than 20% point-to-surface error reduction was reported for higher resolution surface models. The extension was described in the paper “Doo-Sabin Surface Models with Biomechanical Constraints for Kalman Filter Based Endocardial Wall Tracking in 3D+T Echocardiography”, accepted by British Machine Vision Conference 2012.

### 1.3.5 List of Publications

**Papers Included in the Thesis**

1. **Engin Dikici**, Fredrik Orderud. Graph-Cut Based Edge Detection for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography. Proceedings of IEEE Computing in Cardiology (CINC), 2010.

2. **Engin Dikici**, Fredrik Orderud. Polynomial Regression Based Edge Filtering for Left Ventricle Tracking in 3D Echocardiography. Proceedings of Statistical Atlases and Computational Models of the Heart (STACOM), 2011.

3. **Engin Dikici**, Fredrik Orderud, Hans Torp. Best Linear Unbiased Estimator for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography. Proceedings of IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA), 2012.

4. **Engin Dikici**, Fredrik Orderud, Bo Henry Lindqvist. Empirical Bayes Estimator for Endocardial Edge Detection in 3D+T Echocardiography. Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI), 2012.

5. **Engin Dikici**, Sten Roar Snare, Fredrik Orderud. Isoparametric Finite Element Analysis for Doo-Sabin Subdivision Models. Proceedings of Graphics Interface (GI), 2012.

6. **Engin Dikici**, Fredrik Orderud. Generalized Step Criterion Edge Detectors for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography. Accepted to Statistical Atlases and Computational Models of the Heart (STACOM), 2012.

7. **Engin Dikici**, Fredrik Orderud, Gabriel Kiss, Anders Thorstensen, Hans Torp. Doo-Sabin Surface Models with Biomechanical Constraints for Kalman Filter Based Endocardial Wall Tracking in 3D+T Echocardiography. Accepted to the British Machine Vision Conference (BMVC), 2012.

**Related Work**

1. **Engin Dikici**, Fredrik Orderud. Maximum Likelihood and James-Stein Edge Estimators for Left Ventricle Tracking in 3D Echocardiography. Proceedings of Machine Learning in Medical Imaging (MLMI), 2011.

## 1.4 Discussion

### 1.4.1 Graph Cuts Based Edge Measurements

The study introduced a graph-cut based endocardial edge detector (MFMC) and a hybrid detector (Hybrid) combining MFMC and STEP. We observed that MFMC performs better than STEP at ED, while STEP produces better results at ES. This might be due to the changing intensity characteristics of trabeculations during a cardiac cycle. At ED, the trabeculations are filled with blood, and therefore cause strong intensity changes across an endocardial surface model. STEP is prone to including trabeculae in the myocardium, since it processes each intensity profile independently. MFMC, on the other hand, leads to more consistent segmentation results due to smoothness criteria constrained by the graph formulation. At ES, the edge detection problem exhibits a different form. Due to the compression and folding, the trabeculae appears as a part of the myocardium, making the intensity profiles more continuous. The smoothness criterion, which is an advantage for MFMC at ED, turns into a burden. The smoother intensity profiles eliminate the need for an additional smoothing term, which also limits the agility of the model. On the other hand, STEP performs well at ES. Accordingly, the hybrid approach produces superior segmentation results during a cardiac cycle. There exists a significant EF estimation improvement from the closest 6.2% error average produced by STEP to 3.8% using Hybrid.

The MFMC detector uses the push-relabel algorithm for finding a minimal cut, which gives the endocardial edge positions. The method processes each frame in $78ms$. The implemented push-relabel algorithm has $O\left(|V|^3\right)$ time complexity, where $V$ gives the total number of graph nodes [38]. A more advanced algorithm for computing a minimal cut, such as the one introduced in [39], might lead to a faster tracking performance.

The reported surface measurement errors in the study [40] were higher compared to the future studies: the average surface error at ED for the MFMC method was $2.62mm$ in [40] and $2.34mm$ in [41]. This is due to a difference between the surface error validation metrics; point-to-point distances were used in [40], while point-to-surface distances were used in the rest of our studies. Significant differences between the number of recordings used during the validation studies might also cause some discrepancies in the results.

### 1.4.2 Edge Measurement Filtering

The study introduced a local polynomial regression based edge filtering method (LPR), which can be applied as a post-processing tool after an arbitrary edge detection process. We chose to filter the step criterion edges as (1) STEP is a computationally efficient approach allowing additional filtering time to still perform in real time, and (2) it processes each intensity profile independently leading to discontinuous edge measurements, which may greatly benefit from an edge filtering process.

The proposed approach was implemented for the first and second order local polynomial regression models. The method description is provided in a degree-independent fashion; hence generalization of the method to higher orders should be an intuitive task. Increasing the order of a regression model lowers the bias component of the mean square error (MSE) while increasing the variance component. Also, the computation time increases with the filtering order; LPR-1 and LPR-2 processed each frame in $23.7ms$ and $40.8ms$ respectively. This is due to the order dependent dimensions of a design matrix, which is computed during the regression. Therefore, a regression order should be decided considering the bias, variance and computation time related trade-offs. In a future study, the optimal kernel radius and regression order might be found via a statistical study.

Comparative analyses showed that both LPR- 1 and LPR-2 lead to improved surface and volumetric measurement accuracies over STEP. At ED, STEP, LPR-1 and LPR-2 produced $2.94mm$, $2.61mm$ (12% improvement) and $2.57mm$ (13% improvement) mean surface errors. LPR-1 and LPR-2 filters also reduced the LV cavity volume error of STEP at ED by 3.73% and 5% respectively. Comparable surface and volumetric measurement improvements were also reported for the ES phase.

The control point resolution of the tracked model is another smoothing factor for the Kalman tracking framework. Higher resolution Doo-Sabin surface models can represent a wider range of surface deformations, which could benefit more from an edge filtering process.

### 1.4.3 Maximum Likelihood Edge Measurements

We introduced ML based statistical edge detectors that combine the responses of multiple detectors to improve endocardial edge detection accuracy. The proposed approaches incorporate prior knowledge into the edge detection process, leading to more accurate surface and volumetric measurements [41]. However, the statistical methods might cause problems when the training dataset is not large enough to cover shape variations of a tracked anatomy. The EB detector partially addresses this limitation by using contextual priors to generate a bias factor on the prior information. Another possible caveat is that employing multiple edge detectors causes higher computation times. The STEP-ML detector exploits the computationally light set of base detectors to avoid this issue.

The EB detector takes advantage of the Stein effect to improve the detection accuracy of the ML detector. The Stein effect asserts that the estimation MSE can be lowered using information from the full model. Accordingly, we computed the average squared point-to-surface distances for both ML and EB during the evaluation. The EB detector outperformed the ML detector at both ED and ES frames. The EB approach is computationally efficient; finding a contextual prior and refining ML estimations take $\sim 2ms$ per frame. Please note that the contextual prior is found using a convex optimization technique.

The generalized STEP study showed that the higher order STEP detectors do not outperform the classic STEP detector; $0^{th}$(the classic STEP detector), $1^{st}$ and $2^{nd}$ order step detectors produced $2.18mm$, $2.17mm$ and $2.6mm$ average unsigned surface errors respectively. The combined usage of these generalized detectors (STEP-ML) lowered the average unsigned surface error to $2.02mm$. STEP-ML also caused significantly lower regional estimation bias than the generalized detectors. This is due to the learned bias information stored in the ML model, which is factored in during the ML estimations. STEP-ML produced comparable results with the ML approach while reducing the processing time for each frame by over 42%.

The EB detector was implemented and tested using MG, STEP and MFMC as the base detectors. In a future study, it might be evaluated with a base detector set including the generalized STEP detectors. As EB causes $\sim 2ms$ additional frame processing time with an ML detector, it should track each frame in $\sim 48.6ms$ when used with STEP-ML (with a 2.80 GHz Intel Core 2 Duo CPU).

The ML, EB and STEP-ML edge detectors calculate the base estimator bias and weight properties at the ED and ES frames. Then, they interpolate these properties to intermediate frames using a linear interpolation. However, these properties could be learned at more than two cardiac cycle positions, and trigonometrically interpolated to the complete cardiac cycle. The trigonometric interpolation would then take advantage of the cyclic motion of the endocardial walls. This improvement might be investigated in a future study.

### 1.4.4 Improved Physical Modeling for Multi-Resolution Segmentation

We introduced an isoparametric FEA approach for Doo-Sabin surface models with thickness information. There had been a variety of isoparametric formulations proposed for other subdivision schemes in the literature; however, these methods could not be modified for our case in intuitive manners. The isoparametric FEA method proposed for Catmull-Clark subdivision solids [42] requires an additional hexagonal meshing procedure prior to FEA for Doo-Sabin surface models. Our approach eliminates the need for intermediate meshing tools by using a unified geometric representation for the design and FEA stages. The isoparametric formulation developed for Loop subdivision surfaces [43–45] inherit thin shell assumptions neglecting the shear deformations, which are necessary for modeling the myocardial deformations. On the other hand, our method factors in Mindlin plate theory to properly simulate shear deformations.

The degenerated solids based A-I-J elements [46] were used during the physical simulation of Doo-Sabin surface patches in our studies. The A-I-J element may experience locking problems, where the element behaves extra stiff, particularly when the shell is very thin [47]. Several methods can be used to avoid locking problems, including reduced [48] and cross-reduced [49] integration techniques. Locking-free shell elements for our framework might therefore be investigated in a future study.

Our analyses showed that the biomechanical constraints are necessary, especially when the tracked model has a high control node resolution. This is due to the fact that a model with higher complexity can benefit more from spatial regularization during tracking, which is provided by biomechanical constraints. Accordingly, the biomechanically constrained tracker allowed the segmentation accuracy to be stabilized over the model resolution levels; it produced $3.5mm$, $3.3mm$ and $3.3mm$ average unsigned point-to-surface errors at ED for the non-refined, refined and doubly-refined models respectively. The proposed statistical model improvement stage takes advantage of higher model resolution levels as (1) the model node updates provide a more realistic model shape to perform tracking, and (2) deformation modes learned from control point distribution model (CPDM) improve the stiffness matrix accuracy. The tracker using the statistical model improvements produced $3.2mm$, $2.8mm$ and $2.6mm$ average unsigned point-to-surface errors at ED for the non-refined, refined and doubly-refined models respectively. The biomechanical constraints do not introduce any noticeable computational burden for the tracker, since the stiffness and state transition matrices are computed offline and only once for each model.

## 1.5 Conclusion

In this study, we have introduced a set of novel endocardial edge detectors for 3D+T echocardiography, using graph cuts, maximum likelihood, empirical Bayes and generalized step criterion based approaches. We also proposed a polynomial regression based filtering technique to further improve the edge detection accuracy. For physical analysis of Doo-Sabin subdivision surface models, we developed an isoparametric FEA method. Later, we used this concept for biomechanically constrained tracking of multi-resolution Doo-Sabin surface models in a Kalman tracking framework.

The introduced methods have been successfully integrated into the state prediction and update stages of a Kalman tracking framework. The accuracies of these methods were evaluated by comparing the extracted meshes against reference meshes, which were generated by a trained medical doctor using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway). The cross validation approach was used for the evaluation of methods that require training process with pre-segmented data. The analyses showed that the graph-cut based edge detector improves endocardial detection accuracy at end-diastole, and the ML based edge detectors improve detection accuracy for the whole cardiac cycle. Furthermore, these approaches process each frame in either real-time (STEP-ML), or nearly real time (MFMC, ML, EB). Polynomial regression based edge filtering was shown to reduce endocardial edge tracking error, while providing a controller for the tradeoff between edge detection bias and variance. The analyses also showed that the biomechanical constraints are necessary for accurate tracking of subdivision surface models with high control node resolutions.

The edge detection and physical modeling concepts have been validated for time dependent segmentation of the endocardium in 3D+T echocardiography. However, these approaches can be simply modified for the segmentation and tracking of a variety of anatomical structures in ultrasound recordings. The introduced isoparametric FEA formulation can be employed in engineering applications that use Doo-Sabin surface models for computer aided geometric design and analysis.

# References

[1] S. Mendis, P. Puska, B. Norrving, W. H. Organization., W. H. Federation, and W. S. Organization., *Global atlas on cardiovascular disease prevention and control / editors, Shanthi Mendis, Pekka Puska and Bo Norrving.* World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization, Geneva :, 2011.

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

[3] R. C. Houck, J. E. Cooke, and E. A. Gill, "Live 3D echocardiography: a replacement for traditional 2D echocardiography?," *American Journal of Roentgenology (AJR)*, vol. 187, pp. 1092–1106, Oct 2006.

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

[5] A. M. Johri, J. J. Passeri, and M. H. Picard, "Three dimensional echocardiography: approaches and clinical utility," *Heart*, vol. 96, no. 5, pp. 390–397, 2010.

[6] J. Oh, J. Seward, and A. Tajik, *The Echo Manual.* Lippincott Williams & Wilkins, 2006.

[7] S. L. Herz, C. M. Ingrassia, S. Homma, K. D. Costa, and J. W. Holmes, "Parameterization of left ventricular wall motion for detection of regional ischemia," *Annals of Biomedical Engineering*, vol. 33, pp. 912–919, 2005. 10.1007/s10439-005-3312-7.

[8] V. Mor-Avi, K. A. Collins, C. E. Korcarz, M. Shah, K. T. Spencer, and R. M. Lang, "Detection of regional temporal abnormalities in left ventricular function during acute myocardial ischemia," *Am. J. Physiol. Heart Circ. Physiol.*, vol. 280, pp. H1770–1781, Apr 2001.

[9] S. Kapetanakis, M. T. Kearney, A. Siva, N. Gall, M. Cooklin, and M. J. 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.

[10] C. M. Ingrassia, T. P. Usyk, R. C. Kerckhoffs, A. D. McCulloch, K. D. Costa, and J. W. Holmes, "Model-based development of four-dimensional wall motion measures," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 3132, pp. 3061 – 3069, 2007.

[11] G. Carneiro, J. C. Nascimento, and A. Freitas, "The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods," *IEEE Transactions on Image Processing*, pp. 968–982, 2012.

[12] S. K. Setarehdan and J. J. Soraghan, *Segmentation in echocardiographic images*, pp. 64–130. New York, NY, USA: Springer-Verlag New York, Inc., 2002.

[13] B. Eklöf, K. Lindström, and S. Persson, *Ultrasound in Clinical Diagnosis: From pioneering developments in Lund to global application in medicine.* OUP Oxford, 2011.

[14] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey.," *IEEE Transactions on Medical Imaging*, vol. 25, no. 8, pp. 987–1010, 2006.

[15] K. E. Leung and J. G. Bosch, "Automated border detection in three-dimensional echocardiography: principles and promises," *European Journal of Echocardiography*, 2010.

[16] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu, "3d ultrasound tracking of the left ventricles using one-step forward prediction and data fusion of collaborative trackers," in *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[17] M. Rajchl, J. Yuan, and T. M. Peters, "Real-time segmentation in 4d ultrasound with continuous max-flow," vol. 8314, p. 83141F, SPIE, 2012.

[18] J. Yuan, E. Bae, and X.-C. Tai, "A study on continuous max-flow and min-cut approaches," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2217 –2224, june 2010.

[19] 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*, pp. 233 –236, may 2008.

[20] Q. Duan, E. D. Angelini, S. L. Herz, C. M. Ingrassia, K. D. Costa, J. W. Holmes, S. Homma, and A. F. Laine, "Region-based endocardium tracking on real-time three-dimensional ultrasound.," *Ultrasound in medicine biology*, vol. 35, no. 2, pp. 256–265, 2009.

[21] D. Barbosa, O. Bernard, T. Dietenbeck, B. Heyde, D. Friboulet, H. Houle, and J. D'hooge, "Fast 3d echocardiographic segmentation using b-spline explicit active surfaces: A validation study in a clinical setting," in *Proceedings of the 2011 IEEE International Ultrasonics Symposium*, p. (In press), 2011.

[22] D. Barbosa, T. Dietenbeck, J. Schaerer, J. D'hooge, D. Friboulet, and O. Bernard, "B-spline explicit active surfaces: An efficient framework for real-time 3-d region-based segmentation," *Image Processing, IEEE Transactions on*, vol. 21, pp. 241 –251, jan. 2012.

[23] 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, pp. 127–145, 1993. 10.1007/BF01469225.

[24] A. Blake, M. Isard, and D. Reynard, "Learning to track the visual motion of contours," *Artificial Intelligence*, vol. 78, pp. 101–134, 1995.

[25] 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., 1st ed., 1998.

[26] G. Jacob, J. A. 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, 1999.

[27] G. Jacob, J. Noble, A. D. Kelion, and A. P. Banning, "Quantitative regional analysis of myocardial wall motion," *Ultrasound in Medicine &amp; Biology*, vol. 27, no. 6, pp. 773 – 784, 2001.

[28] G. Jacob, J. Noble, C. Behrenbruch, A. Kelion, and A. 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, pp. 226 –238, march 2002.

[29] 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*, pp. 125 –128, sept. 2006.

[30] F. Orderud, J. Hansgard, and S. I. Rabben, "Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach," *Medical Image Computing and Computer Assisted Intervention*, vol. 10, no. Pt 1, pp. 858–865, 2007.

[31] F. Orderud and S. Rabben, "Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[32] F. Orderud, G. Kiss, and H. Torp, "Automatic coupled segmentation of endo- and epicardial borders in 3d echocardiography," in *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, pp. 1749 –1752, nov. 2008.

[33] F. Orderud, H. Torp, and S. I. Rabben, "Automatic alignment of standard views in 3d echocardiograms using real-time tracking," vol. 7265, p. 72650D, SPIE, 2009.

[34] 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*, pp. 1959 –1962, nov. 2008.

[35] S. I. Rabben, A. H. Torp, A. Støylen, S. Slørdahl, K. Bjørnstad, B. O. Haugen, and B. Angelsen, "Semiautomatic contour detection in ultrasound m-mode images," *Ultrasound in Medicine & Biology*, vol. 26, no. 2, pp. 287 – 296, 2000.

[36] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[37] G. Casella, "An introduction to empirical Bayes data analysis," *The American Statistician*, vol. 39, no. 2, pp. 83–87, 1985.

[38] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[39] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, pp. 1124 –1137, sept. 2004.

[40] E. Dikici and F. Orderud, "Graph-cut based edge detection for kalman filter based left ventricle tracking in 3d+t echocardiography," in *Computing in Cardiology, 2010*, pp. 205 –208, sept. 2010.

[41] E. Dikici, F. Orderud, and H. Torp, "Best linear unbiased estimator for kalman filter based left ventricle tracking in 3d+t echocardiography," in *Mathematical Methods in Biomedical Image Analysis (MMBIA), 2012 IEEE Workshop on*, pp. 201 –208, jan. 2012.

[42] D. Burkhart, B. Hamann, and G. Umlauf, "Iso-geometric analysis based on catmull-clark solid subdivision," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1575–1784, 2010.

[43] F. Cirak, M. Ortiz, and P. Schröder, "Subdivision surfaces: A new paradigm for thin-shell finite-element analysis," *International Journal for Numerical Methods in Engineering*, vol. 47, pp. 2039–2072, 2000.

[44] F. Cirak and M. Ortiz, "Fully c1-conforming subdivision elements for finite deformation thin-shell analysis," *International Journal for Numerical Methods in Engineering*, vol. 51, no. 7, pp. 813–833, 2001.

[45] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder, "Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision," *Computer-Aided Design*, vol. 34, no. 2, pp. 137 – 148, 2002.

[46] S. Ahmad, B. M. Irons, and O. C. Zienkiewicz, "Analysis of thick and thin shell structures by curved finite elements," *International Journal for Numerical Methods in Engineering*, vol. 2, no. 3, pp. 419–451, 1970.

[47] S. S. Bhavikatti, *Finite Element Analysis.* Principal, Vijayanagar Engineering College, Bellary, Karnataka: New Age International, 2005.

[48] O. C. Zienkiewicz, R. L. Taylor, and J. M. Too, "Reduced integration technique in general analysis of plates and shells," *International Journal for Numerical Methods in Engineering*, vol. 3, no. 2, pp. 275–290, 1971.

[49] Z. Zhong, "A bilinear shell element with the cross-reduced integration technique," *International Journal for Numerical Methods in Engineering*, vol. 36, no. 4, pp. 611–625, 1993.

# Chapter 2

# Background Material

Engin Dikici
Norwegian University of Science and Technology, Trondheim, Norway

## 2.1 The Human Heart

The heart is the circulatory system's pump that is responsible for maintaining the blood flow throughout the body. The heart consists of (1) four chambers; the left atrium, the left ventricle, the right atrium and the right ventricle, (2) valves to inhibit the backflow of the blood, and (3) joining vessels. The left and right sides of the heart both behave like pumps, and they are not directly connected with each other. Figure 2.1 shows an overview of the anatomy of the cardiac chambers and the direction of the blood flow throughout the heart.

The blood flow for a healthy person can be briefly described as follows. First, oxygenated blood from the lungs travels through the pulmonary veins and enters into the left atrium. Blood then flows into the left ventricle by passing through a one-way valve called the mitral valve. As the left ventricle contracts, it pushes blood to the largest artery in the body, the aorta. The aortic valve between the left ventricle and aorta inhibits backflow into the left ventricle during this stage. Oxygen-rich blood is then carried to the body regions and organs through the arterial system. Next, deoxygenated blood is returned to the right atrium through two large veins called the superior vena cava and inferior vena cana, which drain the upper and lower parts of the body respectively. Blood flows into the right ventricle through a one-way valve called the tricuspid valve. As the right ventricle contracts, it passes blood into the pulmonary artery, which carries deoxygenated blood back to the lungs.

The pumping of the heart is a repeating cycle of relaxation and contraction of the heart muscle. A cardiac cycle consists of two phases called the *systole* and the *diastole*. During the *systole*, the left and right atria first contract at nearly the same time pumping blood into the left and right ventricles. Then, the right and left ventricles contract to pump the blood to the lungs and body. During the *diastole*, blood fills each of the atria and begins filling the relaxed ventricles.

For evaluation of global function of the heart, two metrics are commonly used: stroke volume (SV) and ejection fraction (EF). SV gives the total amount of blood

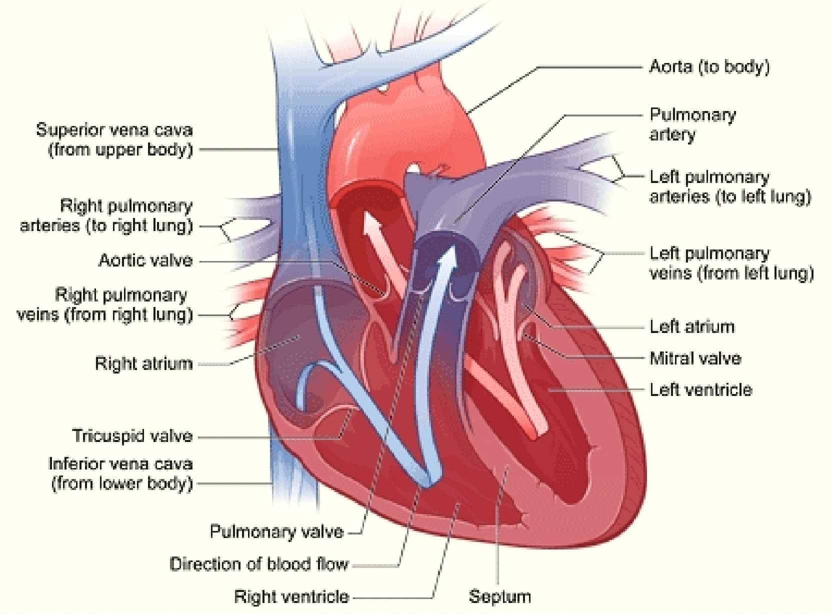

Superior vena cava (from upper body)
Right pulmonary arteries (to right lung)
Aortic valve
Right pulmonary veins (from right lung)
Right atrium
Tricuspid valve
Inferior vena cava (from lower body)
Pulmonary valve
Direction of blood flow
Right ventricle
Septum
Aorta (to body)
Pulmonary artery
Left pulmonary arteries (to left lung)
Left pulmonary veins (from left lung)
Left atrium
Mitral valve
Left ventricle

Figure 2.1: Cross-section of a heart with its internal structures. The blue arrow shows the direction in which deoxygenated blood flows from the body to the lungs. The red arrow shows the direction in which oxygen-rich blood flows from the lungs to the rest of the body. Source: NHLBI.

pumped during a cardiac cycle. It is calculated by finding the difference between the end diastolic (EDV) and end systolic (ESV) volumes of the left ventricle. EF is the ratio between the stroke volume and the end diastolic volume of the left ventricle, and measures the percentage of the ventricle being emptied in each cardiac cycle:

$$SV = EDV - ESV, \tag{2.1}$$

$$EF = \frac{SV}{EDV}. \tag{2.2}$$

## 2.2 Medical Ultrasound

Medical ultrasound is a widely used modality for imaging interior structures of the body. It is portable, considered free of radiation risk, and inexpensive compared to modalities such as magnetic resonance (MR) and computed tomography (CT). Furthermore, it allows for acquisition of real-time images, providing instantaneous visual guidance for many interventional procedures. The modality is based on a pulse-echo approach. This involves transmitting focused ultrasound pulses from a transducer into the body. Due to differences in acoustic impedances along the path of transmission, some of the pulses are then reflected back to the transducer as echos while some continue to penetrate deeper. Received echo signals are then processed and combined using beamformer to generate an image. The frequencies used in medical ultrasound typically lie between 1 and $15MHz$.

Most of the ultrasound transducers use an array of piezoelectric elements to transmit and receive ultrasound pulses. These transducers utilize the fact that piezoelectric crystals can convert electrical signals into mechanical vibrations, and vice versa. During the transmit process, an electrical voltage is applied to the piezoelectric elements, causing them to vibrate and generate ultrasonic waves. Then, during the receive process, acoustical vibrations from the received echos induce electrical signals across the crystals that is sampled and used for image formation.

The distance $r$ between the transducer and the origin of an echo signal is determined using the two-way transit time. By assuming a constant wave propagation speed $c$, the distance is given as function of a two-way transit time $t$:

$$r\left(t\right) = \frac{c \cdot t}{2}, \tag{2.3}$$

where $c \cong 1540m/s$ for soft body tissue.

Transmitted ultrasound waves are partly transmitted to deeper structures, partly reflected back to the transducer as echoes, partly scattered, and partly transformed to heat. The amount of echo returned after hitting a tissue interface is determined by (1) the angle of incidence, (2) dimensions of the reflecting structure, and (3) a tissue property called acoustic impedance $Z$ given by $Z = \rho c$, where $\rho$ is the density of the tissue. Structures that are smooth and larger than the wavelength cause partially reflected echo that travels back to the transducer and a partially transmitted pulse that travels deeper into the tissue body. This type of reflection is called *specular reflection.* The amount of reflected echo therefore depends on the angle of incident and difference in acoustic impedances between two mediums. For non-perpendicular incidence at an interface between tissues with different acoustic impedances, the transmitted pulse is deflected by an angle described by Snell's law. If the ultrasound pulse encounters reflectors whose dimensions are smaller than the ultrasound wavelength, then omnidirectional *scattering* occurs, which does not depend on angle of incidence.

Sector (2D) and volumetric (3D) images can be acquired by transmitting beams that are steered in different directions to cover a region of interest. An array of transducer elements can be focused by controlling the phase of a signal associated with each element. Application of distinct time delays to excite each element focuses the transmitted beam at a particular range. The amount of delay for a given element depends on the distance between the element and the focal point (see Figure 2.2). On reception, the focus can be swept along the beam by dynamically setting the time delays associated with the elements; hence, their position coincide continuously with that of the instantaneous origin of the echoes.

The spatial resolution of the acquired images can be described in the *lateral* and *axial* directions. The lateral resolution is defined as the ability of the system to distinguish two reflectors in the direction perpendicular to the direction of the

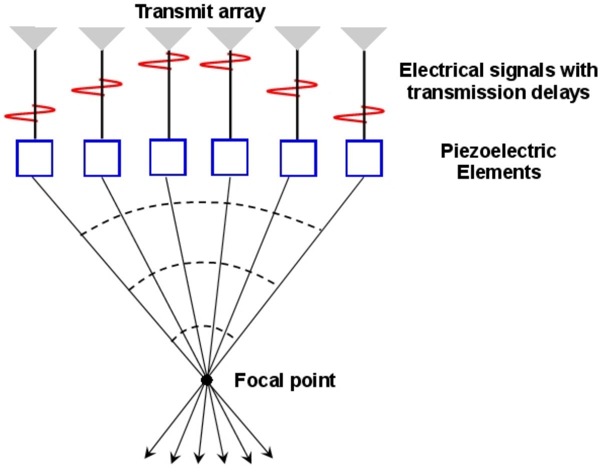


Figure 2.2: Beam focusing by adding time delays across the array of piezoelectric elements.

ultrasound beam. The $3dB$ lateral resolution is given by,

$$Res_l = \frac{\lambda |r|}{D}, \tag{2.4}$$

where $\lambda$ is the wavelength of the ultrasound beam, $D$ is the transducer diameter, and $r$ is the imaging depth. Thus, a larger transducer typically produces a higher lateral resolution. Please note that the lateral resolution is dependent on the depth, which leads to images with a spatially variant resolution for a fixed transducer size. This issue can be partly addressed by using dynamic focusing [1]. The resolution in the direction parallel to the ultrasound beam is given by the axial resolution,

$$Res_a = \frac{\lambda}{2B}, \tag{2.5}$$

where $B$ is the fractional bandwidth of the imaging system. The division by two is due to the pulse-echo response of the system.

The temporal resolution of the images defines the ability to accurately locate structures at a particular time position. The temporal resolution, or frame rate, is limited by the sweep speed of the beam, which is limited by the speed of sound. Frame rates of ultrasound imaging systems typically lie between 30 and $100Hz$.

A more detailed discussion on ultrasound image formation topics can be found in [1].

## 2.3 Graph Cuts and Segmentation

The max-flow/min-cut based endocardial edge detector (MFMC) was introduced in [2], and used to develop more advanced detectors in [3, 4]. This section provides a theoretical background for the graph cut based segmentation concept. Most of the descriptions in this section are based on [5] and [6].

### 2.3.1 Exact Maximum a Posteriori Classification

The endocardial edge detection problem can be formulated as a binary classification task, where $x_i$ denotes the correct class for the $i^{th}$ intensity sample. Each sample either belongs to the inside ($x_i = 0$) or outside ($x_i = 1$) LV cavity class. $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ then gives the correct binary classification for a set of $n$ samples. A Bayesian formulation attains a prior distribution $p(\mathbf{x})$ for all samples, that can be assumed to be a locally dependent Markov random field (MRF) given by,

$$p(\mathbf{x}) \propto \exp\left[\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}\left\{x_i x_j + (1 - x_i)(1 - x_j)\right\}\right], \tag{2.6}$$

where (1) $\beta_{ii} = 0$, (2) $\beta_{ij} = \beta_{ji} > 0$ if $i^{th}$ and $j^{th}$ samples are neighbors, and $\beta_{ij} = \beta_{ji} = 0$ otherwise [5]. This formulation helps providing a smooth classification over the data by penalizing classification differences between the neighboring samples.

The observed intensity value vector for all samples can be represented as $\mathbf{y} = [y_1, y_2, \ldots, y_n]$. Each $y_i$ depends on $\mathbf{x}$ only through $x_i$ with a known conditional density function $g(y_i—x_i)$. Therefore, the likelihood function $l(\mathbf{y}—\mathbf{x})$ of the data can be defined as [5],

$$l(\mathbf{y}—\mathbf{x}) = \prod_{i=1}^{n} g(y_i|x_i) = \prod_{i=1}^{n} g(y_i|1)\, g(y_i|0)^{1-x_i}. \tag{2.7}$$

Combining the likelihood $l(\mathbf{y}—\mathbf{x})$ with the prior distribution $p(\mathbf{x})$, in accordance with Bayes' theorem, gives the posterior distribution,

$$p(\mathbf{x}|\mathbf{y}) \propto l(\mathbf{y}—\mathbf{x})\, p(\mathbf{x}). \tag{2.8}$$

The maximum a posteriori (MAP) estimate seeks $\hat{\mathbf{x}}$ that maximizes $\ln p(\mathbf{x}—\mathbf{y})$ given by,

$$\ln p(\mathbf{x}—\mathbf{y}) = \sum_{i=1}^{n}\lambda_i x_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}\left\{x_i x_j + (1 - x_i)(1 - x_j)\right\}, \tag{2.9}$$

where $\lambda_i = \ln\left\{g(y_i|1)/g(y_i|0)\right\}$ is the log-likelihood ratio at sample $i$. Please note that there are $2^n$ possible values for $\ln p(\mathbf{x}—\mathbf{y})$, making a direct search for $\hat{\mathbf{x}}$ infeasible in most situations. Geman and Geman proposed a simulated annealing based method for this maximization task in [7], but this method is riddled by suboptimal solutions

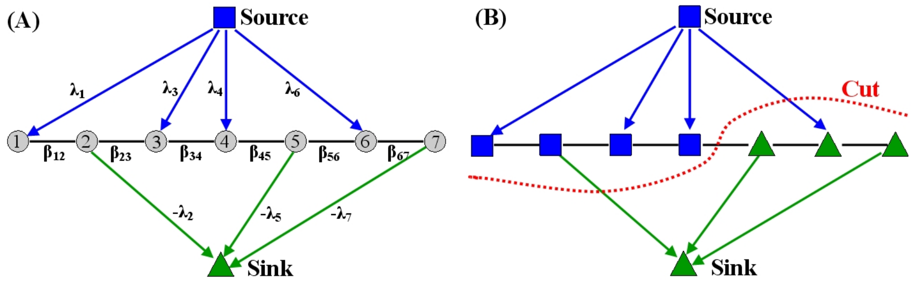


Figure 2.3: (A) Graph representation for a 1-D image with seven intensity samples. Intensity sample nodes are represented by circles, the source node is represented by a square, the sink node is represented by a triangle, and edge capacities are denoted next to the edges. (B) Cut for the graph represented in A. The cut is represented by a dotted red curve. It has the total capacity given by $C(x) = -\lambda_2 + \beta_{45} + \lambda_6$, computed using Equation 2.10.

as it commonly gets trapped in local maximas. An alternative solution introduced by Greig et al. [5] finds the exact maxima via graph cuts in polynomial time complexity.

Greig's solution first formulates a graph with $n+2$ nodes; $n$ nodes corresponding to the intensity samples, and 2 terminal nodes for the abstract source and sink nodes. There is a directed edge from the source to the $i^{th}$ non-terminal node with capacity $\lambda_i$, if $\lambda_i \geq 0$; otherwise, there is a directed edge from the $i^{th}$ non-terminal node to the sink with capacity $-\lambda_i$. There is an undirected edge between the $i^{th}$ and $j^{th}$ non-terminal nodes with capacity $\beta_{ij}$ if $i^{th}$ and $j^{th}$ samples are neighbors (see Figure 2.3-A).

Let $A = \{\textit{source node}\} \cup \{i : x_i = 1\}$ and $B = \{\textit{sink node}\} \cup \{i : x_i = 0\}$, the set of edges with vertices in both $A$ and $B$ is then called a *cut* (see Figure 2.3-B). A cut has a total capacity given by,

$$C(\mathbf{x}) = \sum_{i=1}^{n} x_i \max(0,\, -\lambda_i) + \sum_{i=1}^{n} (1 - x_i) \max(0,\, \lambda_i) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} (x_i - x_j)^2 . \tag{2.10}$$

It can be shown that $C(\mathbf{x}) \propto -\ln p(\mathbf{x}\text{—}\mathbf{y})$. Therefore, the *minimal cut* maximizes $\ln p(\mathbf{x}\text{—}\mathbf{y})$, and produces the MAP estimate for this binary classification task. A node is classified as inside of the LV cavity if it is on the source side; otherwise, it is classified as outside of the LV cavity. An endocardial edge is found at a position where the node classification alters.

The maximum flow and minimal cut problems are dual to each other; hence, the minimal cut can be found by computing the maximum flow for a given graph. There have been a variety of maximum flow algorithms proposed in the literature [8–10]. The push-relabel algorithm introduced by Goldberg and Tarjan [6] is used in the MFMC endocardial edge detector [2] due to its implementation simplicity and decent computational complexity. The details of this approach are provided in the following section.

### 2.3.2 Push-Relabel Algorithm of Goldberg and Tarjan

#### Maximum Flow Problem

Let $G = (V, E)$ be a directed graph with a vertex set $V$ and an edge set $E$. The sizes of $V$ and $E$ are given by $n$ and $m$ respectively. $G$ is a *flow network* if it has (1) the source $s$ and sink $t$ nodes, where a flow is defined between them, and (2) a positive real valued capacity defined for each edge $(v, w) \in E$ as $c(v, w)$. The capacity definition is extended to all vertex pairs by setting $c(v, w) = 0$ if $(v, w) \notin E$. A flow $f$ on $G$ is a real valued function satisfying the following constraints:

1. Capacity: $f(v, w) \leq c(v, w)$ for all $(v, w) \in V \times V$.
2. Anti-symmetry: $f(v, w) = -f(w, v)$ for all $(v, w) \in V \times V$.
3. Flow conservation: $\sum_{u \in V} f(u, v) = 0$ for all $v \in V - \{s, t\}$.

The value of a flow is the net flow into the sink: $|f| = \sum_{u \in V} f(u, t)$. The *maximum flow* is the highest value for a flow on $G$.

#### Generic Push-Relabel Algorithm

The push-relabel algorithm computes the maximum flow for a given graph $G$. The method uses a set of solution specific concepts described below,

- Network preflow: A preflow is a real valued function that satisfies the capacity, anti-symmetry constraints, and a weakened version of the flow conservation constraint given by $\sum_{u \in V} f(u, v) \geq 0$ for all $v \in V - \{s\}$.
- Excessive flow: It gives the net flow into $v$ as $e(v) = \sum_{u \in V} f(u, v)$.
- Residual capacity: The residual capacity of a vertex pair is given by $r(v, w) = c(v, w) - f(v, w)$. A pair $(v, w)$ is called a residual edge if $r(v, w) > 0$.
- Valid labeling: A valid labeling $d$ is a function from the vertices to the non-negative integers and infinity, such that $d(s) = n$, $d(t) = 0$ and $d(v) \leq d(w) + 1$ for every residual edge $(v, w)$.
- Active vertices: A vertex is called active if $v \in V - \{s, t\}$, $d(v) < \infty$, and $e(v) > 0$.

The push-relabel algorithm begins by initializing a preflow $f$ that is equal to the edge capacity on each edge leaving the source, and zero on all other edges. Next, it initializes the labels and excessive flows. After the initialization stage, the algorithm performs the basic operations, push and relabel, (see Figure 2.4) in an iterative fashion until there are no active vertices left. The basic operations modify the preflow $f$ and the labeling $d$. A push from $v$ to $w$ increases $f(v, w)$ and $e(w)$ by $\delta = min(e(v), r(v, w))$, and decreases $f(w, v)$ and $e(v)$ by the same amount. The push is saturating if it causes $r(v, w) = 0$, and it is non-saturating otherwise. A relabeling operation on $v$ sets the label of $v$ to the largest value allowed by the valid labeling constraints. A pseudo code for the algorithm is given in Figure 2.5.

**Push**$(v,\ w)$
*Applicability*: $v$ is active, $r\,(v,\ w) > 0$ and $d\,(v) = d\,(w) + 1$.
*Action*: Send $\delta = min\,(e\,(v)\,,\ r\,(v,\ w))$ units of flow from $v$ to $w$ as follows:

- $f\,(v,\ w) \leftarrow f\,(v,\ w) + \delta,\ \ f\,(w,\ v) \leftarrow f\,(w,\ v) - \delta,$
- $e\,(v) \leftarrow e\,(v) - \delta,\ \ e\,(w) \leftarrow e\,(w) + \delta.$

**Relabel**$(v)$
*Applicability*: $v$ is active, $\forall w \in V,\ r\,(v,\ w) > 0 \Rightarrow d\,(v) \leq d\,(w)$.
*Action*: $d\,(v) \leftarrow min\,\{d\,(w) + 1 — (v,\ w) \in E\}$. (if the minimum is over an empty set, then $d\,(v) \leftarrow \infty$).

Figure 2.4: The basic operations.

```
Max-Flow (V, E, s, t, c)
  ¡Initialize preflow¿
    ∀ (v, w) ∈ (V − {s}) × (V − {s}) do begin
      f (v, w) ← 0; f (w, v) ← 0;
    end
    ∀v ∈ V do begin
      f (s, v) ← c (s, v) ;
      f (v, s) ← −c (s, v) ;
    end
  ¡Initialize labels and excessive flows¿
    d (s) ← n;
    ∀v ∈ V − {s} do begin
      d (v) ← 0;
      e (v) ← f (s, v) ;
    end
  ¡Push and relabel loop¿
    while ∃ a basic operation that applies do
      Perform the applicable basic operation
    end
  return f
```

Figure 2.5: The generic push-relabel algorithm.

```
Discharge
Applicability: $Q \neq empty$.
Action: Remove the vertex $v$ from the front of $Q$.
    Repeat
      Push/relabel($v$)
      if $w$ becomes active during this push/relabel operation then
        add $w$ to the rear of $Q$
      until $e(v) = 0$ or $d(v)$ increases.
      if $v$ is still active then add $v$ to the rear of $Q$.
```

Figure 2.6: The discharge operation.

### First-in First-out Implementation

The selection of the vertices on which the push/relabel is performed has a direct impact on the computational efficiency of the algorithm. The first-in first-out (FIFO) strategy is commonly adopted for this stage. The queue $Q$ of vertices with excessive flows are kept, and the algorithm is executed until $Q$ is empty.

In each iteration, (1) a vertex $v$ from the front of $Q$ is removed, (2) push/relabel operations to $v$ until the excess becomes zero or the label of the vertex increases are performed, and (3) newly active vertices are added to the rear of $Q$ (including $v$ if it is still active). This operation is called the discharge operation (see Figure 2.6).

The computational complexity of the push-relabel algorithm with FIFO implementation is $O\left(n^3\right)$ [6].

## 2.4 Finite Element Analysis using Degenerated Shell Elements

This thesis introduces an isoparametric finite analysis method for Doo-Sabin surface models [11] together with an application of this approach for tracking LV structures in 3D+T echocardiography recordings [12]. The biomechanically constrained tracker introduced in [12] uses FEA to produce a stiffness matrix for a given Doo-Sabin endocardial model, then formulates Kalman filter using this information. The method aims to regulate the motion pattern of tracked objects, without providing a complete finite element model to calculate the stress and strain distribution in the LV wall.

In [11], the formulation of degenerated solids for isoparametric FEA is briefly introduced, but some of the mathematical derivations are referred from other sources due to space limitations. This section provides a more detailed description of the concept based on [13–15].

### 2.4.1 Degenerated Solids

The degenerated solids based FEA approach was first introduced by Ahmad et al. [15]. The method reduces 3D finite elements to 2D by deleting the intermediate vertices in the element's thickness direction, and then by projecting the vertices on each surface to the mid surface (see Figure 2.7). However, the position information for the deleted vertices is utilized to perform the analysis in 3D. The approach is developed with the following assumptions:

1. The normal strains and stresses in the direction of element thickness are always zero. Therefore, the strain energy associated with the stresses perpendicular to the mid surface is ignored.

2. The mid surface normals may not remain as normals after a deformation, however they remain straight as in the Mindlin plate theory [16]. Hence, the shear strain energy needs to be considered during the analysis.

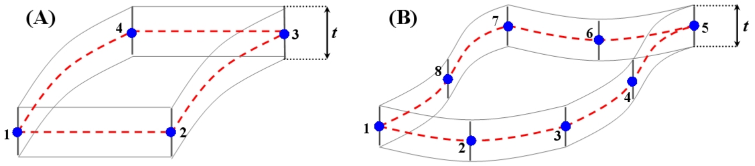


Figure 2.7: Illustrations of (A) 4-noded and (B) 8-noded degenerated shell elements. The projected mid surface vertices are represented by filled blue circles, and the mid surface borders are represented by dashed red curves.

### 2.4.2 Finite Element Formulation of Doo-Sabin Surface Patches

#### Geometric Definition of a Shell Element

The Doo-Sabin surface models are formed using bi-quadric B-spline surface patches [17]. The control vertices representing a surface patch are given by $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m]^T$, where $m$ gives the total number of control vertices for a patch, and $\mathbf{q}_i$ $(i \leq m)$ gives the 3D *physical* coordinates for the $i^{th}$ control vertex. Each physical surface position on a surface patch is uniquely defined using *natural* patch coordinates $(\xi, \eta)$ that vary between $-1$ and 1. The Doo-Sabin basis functions provide a mapping between the natural and physical coordinate systems as,

$$\mathbf{x} = \sum_{i=1}^{m} b_i(\xi, \eta)\, \mathbf{q}_i, \tag{2.11}$$

where $b_i$ is the basis function for the $i^{th}$control vertex, and $\mathbf{x} = [x,\, y,\, z]^T$ is the mapped physical coordinate vector [18].

In degenerated solids approach, elements are described using three surfaces, namely the top, bottom and mid. Top and bottom surfaces are defined on the positive and negative surface normal directions using a similar basis function as the mid surface. These surfaces' control vertex matrices are,

$$\mathbf{Q}^{top} = \left[\mathbf{q}_1^{top}, \mathbf{q}_2^{top}, \ldots, \mathbf{q}_m^{top}\right]^T, \tag{2.12}$$

$$\mathbf{Q}^{bottom} = \left[\mathbf{q}_1^{bottom}, \mathbf{q}_2^{bottom}, \ldots \mathbf{q}_m^{bottom}\right]^T, \tag{2.13}$$

where $\mathbf{Q} = \frac{\left(\mathbf{Q}^{top} + \mathbf{Q}^{bottom}\right)}{2}$.

Assuming that an additional natural parameter $\zeta$ gets values between -1 and 1 on the respecting faces of the element ($\zeta = 0$ gives the mid surface); the relationship between the physical and natural coordinates is given by,

$$\mathbf{x} = \sum_{i=1}^{m} b_i(\xi, \eta) \frac{(1+\zeta)}{2} \mathbf{q}_i^{top} + \sum_{i=1}^{m} b_i(\xi, \eta) \frac{(1-\zeta)}{2} \mathbf{q}_i^{bottom}, \tag{2.14}$$

where $\mathbf{q}_i^{top}$ and $\mathbf{q}_i^{bottom}$ are the $i^{th}$ control vertices for the top and bottom surfaces respectively. Using the mid surface control vertices, Equation 2.14 can alternatively be written as

$$\mathbf{x} = \sum_{i=1}^{m} b_i(\xi, \eta) \left\{ \mathbf{q}_i + t_i \frac{\zeta}{2} \mathbf{v}_{3i} \right\}, \tag{2.15}$$

with

$$t_i = \left|\mathbf{q}_i^{top} - \mathbf{q}_i^{bottom}\right| \quad and \quad \mathbf{v}_{3i} = \frac{\mathbf{q}_i^{top} - \mathbf{q}_i^{bottom}}{\left|\mathbf{q}_i^{top} - \mathbf{q}_i^{bottom}\right|},$$

where $t_i$ and $\mathbf{v}_{3i} = [l_{3i},\, m_{3i},\, n_{3i}]^T$ denote the surface thickness and normal direction for the $i^{th}$ control vertex respectively.

### Local Coordinate System

During the FEA of the shell structures, a *local* coordinate system $(x',\, y',\, z')$ of the element needs to be determined at each mid surface position $(\xi,\, \eta)$. By applying a linear interpolation, an orthogonal set of coordinate axes for any point in the element are given by,

$$\mathbf{v}_3\left(\xi,\eta\right) = \frac{\mathbf{e}_1\left(\xi,\eta\right) \times \mathbf{e}_2\left(\xi,\eta\right)}{\left|\mathbf{e}_1\left(\xi,\eta\right) \times \mathbf{e}_2\left(\xi,\eta\right)\right|}, \tag{2.16}$$

$$\mathbf{v}_1\left(\xi,\eta\right) = \frac{\mathbf{e}_1\left(\xi,\eta\right)}{\left|\mathbf{e}_1\left(\xi,\eta\right)\right|}, \tag{2.17}$$

$$\mathbf{v}_2\left(\xi,\eta\right) = \mathbf{v}_1 \times \mathbf{v}_3, \tag{2.18}$$

where

$$\mathbf{e}_1\left(\xi,\eta\right) = \sum_{i=1}^{m} \frac{\partial b_i\left(\xi,\eta\right)}{\partial \xi}\mathbf{q}_i \quad \mathit{and} \quad \mathbf{e}_2\left(\xi,\eta\right) = \sum_{i=1}^{m} \frac{\partial b_i\left(\xi,\eta\right)}{\partial \eta}\mathbf{q}_i. \tag{2.19}$$

The coordinate transformation matrix, also called the direction cosine matrix, between the physical and local coordinates can be written as,

$$\mathbf{DC} = [\mathbf{v}_1,\, \mathbf{v}_2,\, \mathbf{v}_3] = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix}. \tag{2.20}$$

### Displacement Field

The displacement field definition is derived from Equation 2.15; the displacement field $\mathbf{u} = [u,\, v,\, w]^T$ of any point in the element is given by,

$$\mathbf{u} = \sum_{i=1}^{m} b_i\left(\xi,\eta\right)\left\{\mathbf{u_i} + t_i\frac{\zeta}{2}\begin{bmatrix} n_{3i}\theta_{yi} - m_{3i}\theta_{zi} \\ -n_{3i}\theta_{xi} - l_{3i}\theta_{zi} \\ m_{3i}\theta_{xi} - l_{3i}\theta_{yi} \end{bmatrix}\right\}, \tag{2.21}$$

where (1) $\mathbf{u_i} = [u_i,\, v_i,\, w_i]^T$ is the displacement vector for the $i^{th}$ control vertex, and (2) $\theta_{xi}$, $\theta_{yi}$ and $\theta_{zi}$ are scalar rotations in radians around $x$, $y$ and $z$ axes respectively. The rotation parameters enable the rotation of the axes defined between each $\mathbf{q}_i^{top}$ and $\mathbf{q}_i^{bottom}$ , such that the shearing deformations can be applied on the element.

### Strains and Stresses

*Strain* is a measure of deformation in terms of relative displacement of particles in the element. The strain vector $\varepsilon' = [\varepsilon_{x'},\, \varepsilon_{y'},\, \gamma_{x'y'},\, \gamma_{x'z'},\, \gamma_{y'z'}]^T$ is defined using the first partial derivatives of the local displacement vector $\mathbf{u'} = [u',\, v',\, w']^T$ by,

$$\varepsilon' = \left[\frac{\partial u'}{\partial x'}, \frac{\partial v'}{\partial y'}, \frac{\partial u'}{\partial y'} + \frac{\partial v'}{\partial x'}, \frac{\partial u'}{\partial z'} + \frac{\partial w'}{\partial x'}, \frac{\partial v'}{\partial z'} + \frac{\partial w'}{\partial y'}\right]^T , \tag{2.22}$$

where (1) $\varepsilon_{x'}$ and $\varepsilon_{y'}$ are the normal strains in $x'$ and $y'$ directions, and (2) $\gamma_{x'y'}$, $\gamma_{x'z'}$ and $\gamma_{y'z'}$ are the shear strains in the $x'y'$, $x'z'$ and $y'z'$ planes respectively [13]. Please note that the normal strain in $z'$ direction is ignored due to the first shell assumption. For a clear formulation, the strain vector is split into two vectors as,

$$\varepsilon'_m = [\varepsilon_{x'},\, \varepsilon_{y'},\, \gamma_{x'y'}]^T , \tag{2.23}$$

$$\varepsilon'_s = [\gamma_{x'z'},\, \gamma_{y'z'}]^T . \tag{2.24}$$

*Stress* is a measure of the average force per unit area for the element. The stress components that correspond to the provided strain components are given by,

$$\sigma' = [\sigma_{x'},\, \sigma_{y'},\, \tau_{x'y'},\, \tau_{x'z'},\, \tau_{y'z'}]^T . \tag{2.25}$$

The relation between the strain and stress vectors can be formed using the *Generalized Hooke's Law*, which states that the components of stress are linearly related to the components of strain by $\sigma' = \mathbf{D}\varepsilon'$. The elasticity matrix $\mathbf{D}$ is defined for an isotropic material as,

$$\mathbf{D} = \frac{E}{(1-\mu^2)} \begin{bmatrix} 1 & \mu & 0 & 0 & 0 \\ \mu & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1-\mu}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1-\mu}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} , \tag{2.26}$$

where $E$ is the Young's modulus and $\mu$ is the Poisson's ratio. The elasticity matrix can also be represented as,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_m & 0 \\ 0 & \mathbf{D}_s \end{bmatrix} , \tag{2.27}$$

with

$$\mathbf{D}_m = \frac{E}{(1-\mu^2)} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \quad \textit{and} \quad \mathbf{D}_s = \frac{E}{(1-\mu^2)} \begin{bmatrix} \frac{1-\mu}{2} & 0 \\ 0 & \frac{1-\mu}{2} \end{bmatrix} .$$

### Jacobian Matrix

The computation of a stiffness matrix for an isoparametric element requires the Jacobian of three-dimensional transformations that connect the differentials of $\{x,\, y,\, z\}$ to those of $\{\xi,\, \eta,\, \zeta\}$. The Jacobian matrix $\mathbf{J}$ is given by,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}, \tag{2.28}$$

which is computed using Equation 2.15.

### Assembly of Patch Element Stiffness Matrix

Using the standard variational formulation, the stiffness matrix of an element is given by:

$$\mathbf{K}_e = \iiint \mathbf{B}^T \mathbf{D} \mathbf{B} dxdydz, \tag{2.29}$$

where the strain-displacement matrix $\mathbf{B}$ relates the strains to control vertex displacements ($\delta$) by $\varepsilon' = \mathbf{B}\delta$. The components of $\varepsilon'$ can be found by transforming physical displacement derivatives into local displacement derivatives as,

$$\begin{bmatrix} \frac{\partial u'}{\partial x'} & \frac{\partial v'}{\partial x'} & \frac{\partial w'}{\partial x'} \\ \frac{\partial u'}{\partial y'} & \frac{\partial v'}{\partial y'} & \frac{\partial w'}{\partial y'} \\ \frac{\partial u'}{\partial z'} & \frac{\partial v'}{\partial z'} & \frac{\partial w'}{\partial z'} \end{bmatrix} = \mathbf{DC}^T \left( \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix} \right) \mathbf{DC}, \tag{2.30}$$

where the derivatives of the displacements with respect to the global axes are given by a matrix relation,

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial v}{\partial \xi} & \frac{\partial w}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial v}{\partial \eta} & \frac{\partial w}{\partial \eta} \\ \frac{\partial u}{\partial \zeta} & \frac{\partial v}{\partial \zeta} & \frac{\partial w}{\partial \zeta} \end{bmatrix}. \tag{2.31}$$

To compute $\mathbf{B}$, Equations 2.23 and 2.24 are written as,

$$\varepsilon'_m = \sum_{i=1}^{m} \left\{ \mathbf{B}_{1mi}\mathbf{u}_i + [\mathbf{B}_{2mi} + \zeta \mathbf{B}_{3mi}] \begin{bmatrix} \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \end{bmatrix} \right\}, \tag{2.32}$$

$$\varepsilon'_s = \sum_{i=1}^{m} \left\{ \mathbf{B}_{1si}\mathbf{u}_i + [\mathbf{B}_{2si} + \zeta \mathbf{B}_{3si}] \begin{bmatrix} \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \end{bmatrix} \right\}, \tag{2.33}$$

where (1) $\mathbf{B}_{1mi}$ and $\mathbf{B}_{1si}$ are the strain-displacement matrices formed by considering only the local displacements $u'$, $v'$ and $w'$, and (2) $\mathbf{B}_{2mi}$, $\mathbf{B}_{3mi}$, $\mathbf{B}_{2si}$ and $\mathbf{B}_{3si}$ are the strain-displacement matrices formed by considering only the rotations $\theta_{xi}$, $\theta_{yi}$ and $\theta_{zi}$. Due to the first shell assumption, $\mathbf{B}_{2mi}$ is zero. The other terms are given by,

$$\mathbf{B}_{1mi} = \begin{bmatrix} l_1 B'(1, i) & m_1 B'(1, i) & n_1 B'(1, i) \\ l_2 B'(2, i) & m_2 B'(2, i) & n_2 B'(2, i) \\ l_1 B'(2, i) + & m_1 B'(2, i) + & n_1 B'(2, i) + \\ l_2 B'(1, i) & m_2 B'(1, i) & n_2 B'(1, i) \end{bmatrix}, \tag{2.34}$$

$$\mathbf{B}_{3mi} = \frac{t_i}{2}\left[\begin{array}{ccc} B'(1,\,i)\,(m_{3i}n_1 - n_{3i}m_1) & B'(1,\,i)\,(n_{3i}l_1 - l_{3i}n_1) & B'(1,\,i)\,(l_{3i}m_1 - m_{3i}l_1) \\ B'(2,\,i)\,(m_{3i}n_2 - n_{3i}m_2) & B'(2,\,i)\,(n_{3i}l_2 - l_{3i}n_2) & B'(2,\,i)\,(l_{3i}m_2 - m_{3i}l_2) \\ B'(2,\,i)\,(m_{3i}n_1 - n_{3i}m_1) + & B'(2,\,i)\,(n_{3i}l_1 - l_{3i}n_1) + & B'(2,\,i)\,(l_{3i}m_1 - m_{3i}l_1) + \\ B'(1,\,i)\,(m_{3i}n_2 - n_{3i}m_2) & B'(1,\,i)\,(n_{3i}l_2 - l_{3i}n_2) & B'(1,\,i)\,(l_{3i}m_2 - m_{3i}l_2) \end{array}\right], \tag{2.35}$$

$$\mathbf{B}_{1si} = \left[\begin{array}{ccc} l_1B'(3,\,i) + & m_1B'(3,\,i) + & n_1B'(3,\,i) + \\ l_3B'(1,\,i) & m_3B'(1,\,i) & n_3B'(1,\,i) \\ l_2B'(3,\,i) + & m_2B'(3,\,i) + & n_2B'(3,\,i) + \\ l_3B'(2,\,i) & m_3B'(2,\,i) & n_3B'(2,\,i) \end{array}\right], \tag{2.36}$$

$$\mathbf{B}_{2si} = \frac{t_i}{2}b_iB''\left[\begin{array}{ccc} m_{3i}n_i - n_{3i}m_i & n_{3i}l_i - l_{3i}n_i & l_{3i}m_i - m_{3i}l_i \\ m_{3i}n_2 - n_{3i}m_2 & n_{3i}l_2 - l_{3i}n_2 & l_{3i}m_2 - m_{3i}l_2 \end{array}\right], \tag{2.37}$$

$$\mathbf{B}_{3si} = \frac{t_i}{2}\left[\begin{array}{ccc} B'(3,\,i)\,(m_{3i}n_1 - n_{3i}m_1) + & B'(3,\,i)\,(n_{3i}l_1 - l_{3i}n_1) + & B'(3,\,i)\,(l_{3i}m_1 - m_{3i}l_1) + \\ B'(1,\,i)\,(m_{3i}n_3 - n_{3i}m_3) & B'(1,\,i)\,(n_{3i}l_3 - l_{3i}n_3) & B'(1,\,i)\,(l_{3i}m_3 - m_{3i}l_3) \\ B'(3,\,i)\,(m_{3i}n_2 - n_{3i}m_2) + & B'(3,\,i)\,(n_{3i}l_2 - l_{3i}n_2) + & B'(3,\,i)\,(l_{3i}m_2 - m_{3i}l_2) + \\ B'(2,\,i)\,(m_{3i}n_3 - n_{3i}m_3) & B'(2,\,i)\,(n_{3i}l_3 - l_{3i}n_3) & B'(2,\,i)\,(l_{3i}m_3 - m_{3i}l_3) \end{array}\right], \tag{2.38}$$

where

$$B'(1,\,i) = \frac{\partial b_i}{\partial x}l_1 + \frac{\partial b_i}{\partial y}m_1 + \frac{\partial b_i}{\partial z}n_1, \tag{2.39}$$

$$B'(2,\,i) = \frac{\partial b_i}{\partial x}l_2 + \frac{\partial b_i}{\partial y}m_2 + \frac{\partial b_i}{\partial z}n_2, \tag{2.40}$$

$$B'(3,\,i) = \frac{\partial b_i}{\partial x}l_3 + \frac{\partial b_i}{\partial y}m_3 + \frac{\partial b_i}{\partial z}n_3, \tag{2.41}$$

$$B'' = l_3\mathbf{J}_{13} + m_3\mathbf{J}_{23} + n_3\mathbf{J}_{33}. \tag{2.42}$$

Accordingly,

$$\varepsilon' = \left[\begin{array}{c} \varepsilon'_m \\ \varepsilon'_s \end{array}\right] = \sum_{i=1}^{m}\left[\begin{array}{cc} \mathbf{B}_{1mi} & \zeta\mathbf{B}_{3mi} \\ \mathbf{B}_{1si} & \mathbf{B}_{2si} + \zeta\mathbf{B}_{3si} \end{array}\right]\delta, \tag{2.43}$$

with $\delta = [u_i,\, v_i,\, w_i,\, \theta_{xi},\, \theta_{yi},\, \theta_{zi}]^T$.

The infinitesimal volume computed in physical coordinates can be expressed in terms of the volume in natural coordinates as $dxdydz = |\mathbf{J}|\, d\xi d\eta d\zeta$. Finally, $\mathbf{K}_e = \iiint \mathbf{B}^T\mathbf{D}\mathbf{B}\,|\mathbf{J}|\, d\xi d\eta d\zeta$ can be numerically estimated by Gauss Legendre quadrature rules. Usage of two samples in the $\zeta$ direction and a minimum of four samples in both the $\xi$ and $\eta$ directions has been provent to be sufficient for thin shell element stiffness matrix calculations for most purposes [15].

## 2.5 Kalman Filter Based Tracking

This section provides a brief overview on Bayesian tracking, Kalman filter, extended Kalman filtering, information filter and Kalman smoothing concepts. The descriptions in this section are based on [19–21], and serve as the theoretic foundation for the Kalman filter-based approach for the tracking of cardiac chambers in this thesis.

### 2.5.1 State Space Modeling and Bayesian Estimation

A generalized non-linear state space model can be represented using,

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k,\, \mathbf{u}_k,\, \mathbf{w}_k\right), \tag{2.44}$$

$$\mathbf{z}_k = h\left(\mathbf{x}_k,\, \mathbf{v}_k\right), \tag{2.45}$$

where (1) $f$ and $h$ are the governing state and output functions, (2) $\mathbf{x}_k$ is the system state, (2) $\mathbf{u}_k$ is the system input, (3) $\mathbf{z}_k$ is the measurement, and (4) $\mathbf{w}_k$ and $\mathbf{v}_k$ are the stochastic state and measurement noise vectors at the discrete time $k$.

The recursive Bayesian tracking approach estimates $\mathbf{x}_k$ using the posterior probability distribution function (pdf) given by $p\left(\mathbf{x}_k—\mathbf{z}_{1:k}\right)$. For this task, it computes $p\left(\mathbf{x}_k—\mathbf{z}_{1:k}\right)$ via recursive prediction and update stages, where the initial distribution $p\left(\mathbf{x}_0—\mathbf{z}_0\right)$ is assumed to be known.

The prediction stage finds the prior pdf given by,

$$p\left(\mathbf{x}_k—\mathbf{z}_{1:k-1}\right) = \int p\left(\mathbf{x}_k—\mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1}—\mathbf{z}_{1:k-1}\right) d\mathbf{x}_{k-1}, \tag{2.46}$$

which is solved using (1) the posterior pdf $p\left(\mathbf{x}_{k-1}—\mathbf{z}_{1:k-1}\right)$, and (2) the probabilistic model of the state transition $p\left(\mathbf{x}_k—\mathbf{x}_{k-1}\right)$ that is defined using the Equation 2.44. Please note the Equation 2.44 describes a first order Markov process.

The update stage employs the measurement $\mathbf{z}_k$, and computes the posterior pdf given by,

$$p\left(\mathbf{x}_k—\mathbf{z}_{1:k}\right) = \frac{p\left(\mathbf{z}_k \middle| \mathbf{x}_k\right) p\left(\mathbf{x}_k—\mathbf{z}_{1:k-1}\right)}{p\left(\mathbf{z}_k \middle| \mathbf{z}_{1:k-1}\right)}, \tag{2.47}$$

where (1) the likelihood function $p\left(\mathbf{z}_k \middle| \mathbf{x}_k\right)$ is derived using the Equation 2.45, and (2) the normalizing constant $p\left(\mathbf{z}_k \middle| \mathbf{z}_{1:k-1}\right)$ depends on the likelihood function as $p\left(\mathbf{z}_k \middle| \mathbf{z}_{1:k-1}\right) = \int p\left(\mathbf{z}_k \middle| \mathbf{x}_k\right) p\left(\mathbf{x}_k—\mathbf{z}_{1:k-1}\right) d\mathbf{x}_k$.

Optimal Bayesian estimation depends on computation of exact solutions to Equations 2.46 and 2.47 in a recursive fashion. In many situations, the solution unfortunately cannot be determined analytically. However, for a limited set of system definitions with linear functions and/or discrete state, approaches including the Kalman and grid-based filters can produce optimal estimates. If the analytic solution is intractable, extended Kalman, particle and approximate grid-based filters can be used to approximate the optimal Bayesian estimate.

### 2.5.2 Kalman Filter

The Kalman filter was invented in 1960 by Rudolph E. Kalman as a technique for filtering and prediction in linear systems subject to Gaussian noise [20]. It can be shown that the posterior pdf is a Gaussian, which can be parametrized using a mean and a covariance, if the following three properties hold:

1. $f\left(\mathbf{x}_k,\, \mathbf{u}_k,\, \mathbf{w}_k\right)$ is a linear function of $\mathbf{x}_k$, $\mathbf{u}_k$ and $\mathbf{w}_k$,
2. $h\left(\mathbf{x}_k,\, \mathbf{v}_k\right)$ is a linear function of $\mathbf{x}_k$ and $\mathbf{v}_k$,
3. The state and measurement noise vectors $\mathbf{w}$ and $\mathbf{v}$ are uncorrelated, white Gaussian processes with zero mean.

Following these assumptions, the system can be represented as,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, && (2.48)\\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, && (2.49)\\ \mathbf{w}_k \sim N\left(0,\, \mathbf{Q}_k\right), &\quad \mathbf{v}_k \sim N\left(0,\, \mathbf{R}_k\right), && (2.50)\\ E\left(\mathbf{w}_i\mathbf{w}_j^T\right) = \mathbf{Q}_i\delta_{i-j}, &\quad E\left(\mathbf{v}_i\mathbf{v}_j^T\right) = \mathbf{R}_i\delta_{i-j}, && (2.51)\\ E\left(\mathbf{w}_k\mathbf{v}_k^T\right) &= 0, && (2.52) \end{aligned}$$

where (1) $\mathbf{F}_k$, $\mathbf{B}_k$ and $\mathbf{H}_k$ are known matrices that can be time-dependent, and (2) $\mathbf{Q}_k$ and $\mathbf{R}_k$ are covariance matrices corresponding to the second order statistics of $\mathbf{w}_k$ and $\mathbf{v}_k$ respectively. As the input has a Gaussian pdf and the model is linear, the state $\mathbf{x}_k$ and measurement $\mathbf{z}_k$ vectors also have Gaussian pdfs, which are fully described by their means and covariances. Therefore, the recursive process, defined using Equations 2.46 and 2.47, is given by,

$$\begin{aligned} p\left(\mathbf{x}_k \middle|\, \mathbf{z}_{1:k-1}\right) &= N\left(\mathbf{x}_k;\, \bar{\mathbf{x}}_k,\, \bar{\mathbf{P}}_k\right), && (2.53)\\ p\left(\mathbf{x}_k \middle|\, \mathbf{z}_{1:k}\right) &= N\left(\mathbf{x}_k;\, \hat{\mathbf{x}}_k,\, \hat{\mathbf{P}}_k\right). && (2.54) \end{aligned}$$

where

$$\begin{aligned} \bar{\mathbf{x}}_k &= \mathbf{F}_k\hat{\mathbf{x}}_{k-1} + \mathbf{B}_k\mathbf{u}_k, && (2.55)\\ \bar{\mathbf{P}}_k &= \mathbf{F}_k\hat{\mathbf{P}}_{k-1}\mathbf{F}_k^T + \mathbf{Q}_k, && (2.56) \end{aligned}$$

are computed during the prediction stage, and

$$\begin{aligned} \mathbf{K}_k &= \bar{\mathbf{P}}_k\mathbf{H}_k^T\left(\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}, && (2.57)\\ \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\bar{\mathbf{x}}_k\right), && (2.58)\\ \hat{\mathbf{P}}_k &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\bar{\mathbf{P}}_k, && (2.59) \end{aligned}$$

are computed during the update stage. (1) $\bar{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_k$ are called the prior and posterior state estimates, (2) $\bar{\mathbf{P}}_k$ and $\hat{\mathbf{P}}_k$ are called the prior and posterior state error covariances, and (3) $\mathbf{K}_k$ is called the Kalman gain matrix for the time step $k$. These equations are recognized as the Kalman filter [22].

The Kalman filter gives the optimal solution to the linear estimation problem if the given assumptions hold, where the optimality is provided in a least squares (LS) sense. For situations where the posterior pdf is not a Gaussian distribution, the filter is not guaranteed to be optimal [19].

### 2.5.3 Extended Kalman Filter

In many situations, the model cannot be represented using linear functions; hence, the governing state and output functions are nonlinear. The extended Kalman filter (EKF) addresses the problem by linearizing the model definition around the current mean and covariance estimates.

The EKF approximates the prior and posterior pdfs as,

$$p\left(\mathbf{x}_k | \, \mathbf{z}_{1:k-1}\right) \approx N\left(\mathbf{x}_k; \, \bar{\mathbf{x}}_k, \, \bar{\mathbf{P}}_k\right), \tag{2.60}$$

$$p\left(\mathbf{x}_k | \, \mathbf{z}_{1:k}\right) \approx N\left(\mathbf{x}_k; \, \hat{\mathbf{x}}_k, \, \hat{\mathbf{P}}_k\right), \tag{2.61}$$

where

$$\bar{\mathbf{x}}_k = f\left(\hat{\mathbf{x}}_{k-1}, \, \mathbf{u}_k, \, 0\right), \tag{2.62}$$

$$\bar{\mathbf{P}}_k = \hat{\mathbf{F}}_k \hat{\mathbf{P}}_{k-1} \hat{\mathbf{F}}_k^T + \mathbf{Q}_k, \tag{2.63}$$

are computed during the prediction stage, and

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \hat{\mathbf{H}}_k^T \left(\hat{\mathbf{H}}_k \bar{\mathbf{P}}_k \hat{\mathbf{H}}_k^T + \mathbf{R}_k\right)^{-1}, \tag{2.64}$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{z}_k - h\left(\bar{\mathbf{x}}_k, \, 0\right)\right), \tag{2.65}$$

$$\hat{\mathbf{P}}_k = \left(\mathbf{I} - \mathbf{K}_k \hat{\mathbf{H}}_k\right) \bar{\mathbf{P}}_k, \tag{2.66}$$

are computed during the update stage. (1) $f$ and $h$ are nonlinear functions, and (2) $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{H}}_k$ are the Jacobians of $f$ and $h$ given by,

$$\hat{\mathbf{F}}_k = \mathbf{J}_x f\left(\mathbf{x}, \, \mathbf{u}, \, \mathbf{w}\right) = \frac{\partial f\left(\mathbf{x}, \, \mathbf{u}, \, \mathbf{w}\right)}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_k, \mathbf{u}_k, 0}, \tag{2.67}$$

$$\hat{\mathbf{H}}_k = \mathbf{J}_x h\left(\mathbf{x}, \, \mathbf{v}\right) = \frac{\partial h\left(\mathbf{x}, \, \mathbf{v}\right)}{\partial \mathbf{x}} |_{\bar{\mathbf{x}}_k, 0} \, . \tag{2.68}$$

The represented EKF uses a Taylor expansion where only the first term is utilized. It is possible to utilize more terms to perform better approximations for the $f$ and $h$ functions at the cost of additional computational complexity.

### 2.5.4 Information Filter

The information filter is an alternative formulation of the Kalman filter, preferred when either the measurement vector is large compared to the state vector, or the initial system state is unknown. The prediction step of the approach is unchanged from the conventional Kalman filter as,

$$\begin{aligned}\bar{\mathbf{x}}_k &= \mathbf{F}_k\hat{\mathbf{x}}_{k-1} + \mathbf{B}_k\mathbf{u}_k, && (2.69)\\ \bar{\mathbf{P}}_k &= \mathbf{F}_k\hat{\mathbf{P}}_{k-1}\mathbf{F}_k^T + \mathbf{Q}_k. && (2.70)\end{aligned}$$

The update stage is modified to use the terms,

$$\begin{aligned}\hat{\mathbf{P}}_k^{-1} &= \bar{\mathbf{P}}_k^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k, && (2.71)\\ \hat{\mathbf{P}}_k^{-1}\hat{\mathbf{x}}_k &= \bar{\mathbf{P}}_k^{-1}\bar{\mathbf{x}}_k + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k, && (2.72)\end{aligned}$$

where the posterior state estimate and its error covariance are found as,

$$\begin{aligned}\hat{\mathbf{P}}_k &= \left(\bar{\mathbf{P}}_k^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\right)^{-1}, && (2.73)\\ \hat{\mathbf{x}}_k &= \hat{\mathbf{P}}_k\left(\bar{\mathbf{P}}_k^{-1}\bar{\mathbf{x}}_k + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k\right). && (2.74)\end{aligned}$$

The formulation sums the measurements into the *information matrix* $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ and *information vector* $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}$. Using the Equation 2.74, it can be shown that the Kalman gain for the information filter is $\hat{\mathbf{P}}_k\mathbf{H}_k^T\mathbf{R}_k^{-1}$, which is algebraically equivalent to the Kalman gain given in Equation 2.57.

The approach avoids the problem of inverting matrices of the size of the measurement covariance matrix. Instead, matrices with the same dimensions as the state error covariance matrix are inverted. In cases where there is no information about the initial state, the entries of $\hat{\mathbf{P}}_0$ will contain very large values in a conventional Kalman filter, which might lead to numerical instability during initialization. The information filter avoids this problem by using the inverse of the initial state error covariance matrix that can be initialized to zero.

### 2.5.5 Kalman Smoother

The conventional Kalman filter utilizes measurements acquired up until the current time position $k$ to compute state estimates using the posterior distribution given by $p\left(\mathbf{x}_k \middle| \mathbf{z}_{1:k}\right)$. If future measurements are also available, such that $\mathbf{z}_{1:k+q}$ is known where $q > 0$, then a Kalman filter is not able to incorporate these future measurements to improve estimation accuracy. The Kalman smoother [23] addresses this problem by computing state estimates based on all available measurements both before and after $k$. It provides an optimal estimator for $p\left(\mathbf{x}_k \middle| \mathbf{z}_{1:k+q}\right)$ given that the same linearity and Gaussian assumptions as the regular Kalman filter hold for a given system [19].

The fixed-interval and fixed-lag smoothers are two of the commonly used Kalman smoothing approaches. The fixed-interval smoother performs the smoothing in batch mode using all measurements within $0 \leq q \leq k$ in a single run. The fixed-lag smoother performs smoothing for a time position that is prior to the current measurement by a fixed number of time steps, meaning that $q$ is fixed.

Fraser and Potter introduced a fixed-interval smoother formulation that combines two Kalman filters running in forward and backward time directions [24]. This idea is extended in [20] for fixed-lag smoothing with reasonably short fixed lags. This interpretation of the smoothers is called the *forward-backward* smoother.

### Forward-Backward Interpretation of the Fixed-Lag Kalman Smoother

The forward-backward smoother employs two Kalman trackers, the forward and backward filters, for estimating the state vector $\mathbf{x}_k$ at time step $k$. The forward tracker processes the measurements in the forward time direction, giving the processing order of $\mathbf{z}_1,\, \mathbf{z}_2, \ldots,\, \mathbf{z}_k$. The posterior estimate for time step $k$ produced by the forward tracker is shown as,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}\left(k|k\right) = E\left[\mathbf{x}_k|\mathbf{z}_1,\, \mathbf{z}_2, \ldots,\, \mathbf{z}_k\right]. \tag{2.75}$$

The posterior error covariance matrix of $\hat{\mathbf{x}}_k$ is given by,

$$\mathbf{P}_k = E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T\right].$$

The backward tracker processes the measurements in the backward time direction starting from a future time point, giving the processing order of $\mathbf{z}_{k+q},\, \mathbf{z}_{k+q-1}, \ldots,\, \mathbf{z}_{k+1}$ for a predefined lag value of $q$. The backward filter is initialized with no prior information, where the diagonal entries of the initial error covariance matrix are set very large and the off-diagonal entries are set as zeros.

Assuming that the state transition matrix for the forward tracker is $\mathbf{A}$, the backward filter from [24] computes a prior state estimate for step $g \in \{k,\, k+1, \ldots,\, k+q-1\}$ by,

$$\hat{\mathbf{x}}^*_{bg} = \mathbf{A}^{-1}\hat{\mathbf{x}}_{b(g+1)}, \tag{2.76}$$

where $\hat{\mathbf{x}}_{b(g+1)}$ is the backward filter's posterior estimate for time step $g+1$. If the forward filter uses the state transition matrix only for the regularization purposes; pulling state estimations toward an initial state, then the backward tracker should not invert the state transition matrix for computing the prior estimates and it should use the similar state transition matrix as the forward filter.

The prior estimate for time step $k$ produced by the backward tracker is shown as,

$$\hat{\mathbf{x}}^*_{bk} = E\left[\mathbf{x}_k|\mathbf{z}_{k+1},\, \mathbf{z}_{k+2}, \ldots,\, \mathbf{z}_{k+q}\right]. \tag{2.77}$$

Please note that the backward tracker uses the prior estimate at $k$ to avoid assimilating $\mathbf{z}_k$ twice during the smoothing process. The error covariance matrix of $\hat{\mathbf{x}}^*_{bk}$ is given by

$$\mathbf{P}_{bk}^{*} = E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{bk}^{*}\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_{bk}^{*}\right)^T\right]. \tag{2.78}$$

Finally, the forward and backward filter results are fused using the covariance intersection method [20] as,

$$\hat{\mathbf{x}}\left(k|\left(k+q\right)\right) = \mathbf{P}\left(k|\left(k+q\right)\right)\left[\mathbf{P}_k^{-1}\hat{\mathbf{x}}_k + \mathbf{P}_{bk}^{*-1}\hat{\mathbf{x}}_{bk}^{*}\right], \tag{2.79}$$

where

$$\mathbf{P}\left(k|\left(k+q\right)\right) = \left[\mathbf{P}_k^{-1} + \mathbf{P}_{bk}^{*-1}\right]^{-1}. \tag{2.80}$$

The smoothed estimate $\hat{\mathbf{x}}\left(k|\left(k+q\right)\right)$ uses all measurements from the first to the $(k+q)^{th}$ time step,

$$\hat{\mathbf{x}}\left(k|\left(k+q\right)\right) = E\left[\mathbf{x}_k|\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{k+q}\right], \tag{2.81}$$

where $k$ and $q$ measurements are factored in by the forward and backward trackers respectively. The fixed-lag smoother processes $q+1$ frames in each time step: (1) the current frame is processed by the forward tracker, which has already processed the measurements from the first to the current frame, and (2) $q$ future frames are processed by the backward tracker. The estimation process lags for $q$ frames with respect to the most recently acquired frame due to the backward tracking process. Please see Figure 2.8 for the illustration of the method for the endocardial mesh tracking.

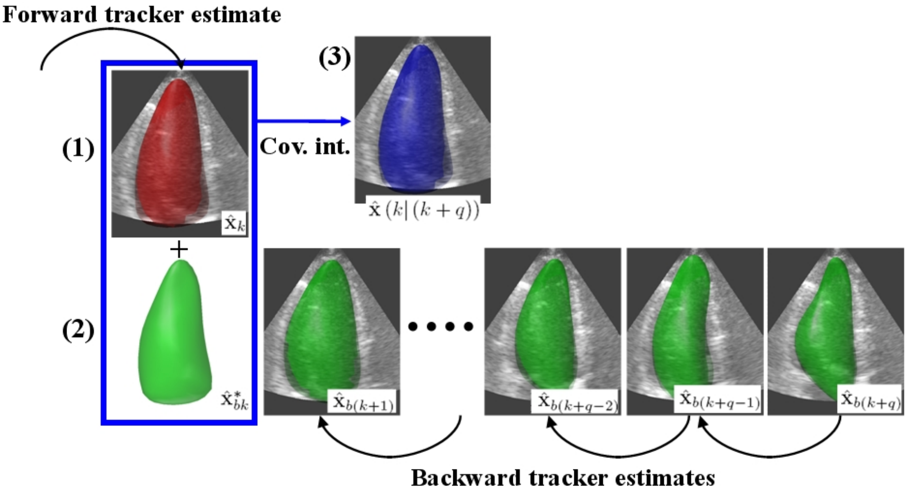

Forward tracker estimate
(1)
(2)
(3)
Cov. int.
$\hat{\mathbf{x}}_k$
+
$\hat{\mathbf{x}}^*_{bk}$
$\hat{\mathbf{x}}\,(k|\,(k+q))$
$\hat{\mathbf{x}}_{b(k+1)}$
$\hat{\mathbf{x}}_{b(k+q-2)}$
$\hat{\mathbf{x}}_{b(k+q-1)}$
$\hat{\mathbf{x}}_{b(k+q)}$
Backward tracker estimates

Figure 2.8: The system state $\mathbf{x}$ represents the endocardial model shape and position. (1) The forward tracker computes the posterior estimate (red) for time step $k$, (2) the backward tracker computes the prior estimate (green) for time step $k$ using $q$ measurements, (3) smoothed estimate (blue) is found using the covariance intersection method.

# References

[1] T. Szabo, *Diagnostic Ultrasound Imaging: Inside Out: Inside Out.* Diagnostic Ultrasound Imaging: Inside Out, Elsevier Science, 2004.

[2] E. Dikici and F. Orderud, "Graph-cut based edge detection for kalman filter based left ventricle tracking in 3d+t echocardiography," in *Computing in Cardiology, 2010*, pp. 205 –208, sept. 2010.

[3] E. Dikici, F. Orderud, and H. Torp, "Best linear unbiased estimator for kalman filter based left ventricle tracking in 3d+t echocardiography," in *Mathematical Methods in Biomedical Image Analysis (MMBIA), 2012 IEEE Workshop on*, pp. 201 –208, jan. 2012.

[4] E. Dikici, F. Orderud, and B. H. Lindqvist, "Empirical Bayes estimator for endocardial edge detection in 3d+t echocardiography," in *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*, pp. 1331 –1334, may 2012.

[5] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 2, pp. 271–279, 1989.

[6] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[7] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, Nov. 1984.

[8] L. R. Ford and D. R. Fulkerson, *Flows in Networks.* Princeton University Press, 1962.

[9] E. A. Dinic, "Algorithm for Solution of a Problem Of Maximum Flow in Networks with Power Estimation," *Soviet Math. Dokl*, vol. 11, pp. 1277–1280, 1970.

[10] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 1124–1137, Sept. 2004.

[11] E. Dikici, S. R. Snare, and F. Orderud, “Isoparametric finite element analysis for doo-sabin subdivision models,” in *Proceedings of the 2012 Graphics Interace Conference*, GI ’12, (Toronto, Ont., Canada, Canada), pp. 19–26, Canadian Information Processing Society, 2012.

[12] E. Dikici, F. Orderud, G. Kiss, A. Thorstensen, and H. Torp, “Doo-Sabin surface models with biomechanical constraints for Kalman filter based endocardial wall tracking in 3d+t echocardiography,” in *Accepted to British Machine Vision Conference*, 2012.

[13] S. S. Bhavikatti, *Finite Element Analysis.* Principal, Vijayanagar Engineering College, Bellary, Karnataka: New Age International, 2005.

[14] G. Buchanan, *Schaum’s Outline of Finite Element Analysis.* Schaum’s Outline Series, McGraw-Hill Companies,Incorporated, 1994.

[15] S. Ahmad, B. M. Irons, and O. C. Zienkiewicz, “Analysis of thick and thin shell structures by curved finite elements,” *International Journal for Numerical Methods in Engineering*, vol. 2, no. 3, pp. 419–451, 1970.

[16] R. D. Mindlin, “Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates,” *Journal of Applied Mechanics*, vol. 18, pp. 31–38, 1951.

[17] D. Doo and M. Sabin, “Behaviour of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978.

[18] F. Orderud and S. I. Rabben, “Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces,” in *CVPR*, 2008.

[19] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *Signal Processing, IEEE Transactions on*, vol. 50, pp. 174 –188, feb 2002.

[20] R. Brown and P. Hwang, *Introduction to random signals and applied Kalman filtering.* No. v. 1, J. Wiley, 1992.

[21] F. Orderud, *Real-time Segmentation of 3D Echocardiograms Using a State Estimation Approach with Deformable Models.* PhD thesis, Norwegian University of Science and Technology (NTNU), 2010.

[22] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME - Journal of Basic Engineering*, no. 82 (Series D), pp. 35–45, 1960.

[23] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation.* New York, NY, USA: John Wiley & Sons, Inc., 2002.

[24] D. Fraser and J. Potter, “The optimum linear smoother as a combination of two optimum linear filters,” *Automatic Control, IEEE Transactions on*, vol. 14, pp. 387 – 390, aug 1969.

## Chapter 3

# Graph-Cut Based Edge Detection for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography

Engin Dikici[1], Fredrik Orderud[2]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

Consistent endocardial border segmentation in 3D echocardiography is a challenging task. One of the major difficulties rises due to the fact that the trabeculated structure of the endocardium causes the endocardial intensity profile characteristics to change over a cardiac cycle. In this paper, we present a hybrid edge detection approach using both max flow/min cut (MFMC) and step criterion (STEP) edge detectors, and its integration into a Kalman filter based left ventricle (LV) tracking framework.

We treat the endocardial edge detection problem as a graph partitioning problem where the graph is defined by using the intensity profiles, and propose a max flow/min cut based solution. For the end-systole, the step criterion edge detector is a more suitable option. Accordingly, we introduce the weighted combination of these techniques called the hybrid edge detector (Hybrid) where the weight factor is determined by the size of the tracked endocardial mesh.

Surface and volumetric measurement comparisons between the STEP, MFMC and Hybrid shows that the Hybrid handles the specific problem of time-dependent intensity profiles better than the other approaches.

## 3.1 Introduction

The introduction of 3D echocardiography has enabled rapid and low-cost acquisition of volumetric images of the left ventricle (LV). Numerous techniques for segmenting these images have appeared in the literature. However, accurate and consistent detection of the endocardial border still remains a challenging task. Part of the reason for this is that the trabeculated structure of the endocardial boundary leads to alternating edge characteristics over a cardiac cycle.

Kalman filter-based 3D segmentation adopts a sequential prediction and update strategy; the surface deformations are predicted by using a kinematic model, then the prediction is updated based on the information provided by image measurements. Normal displacement measurements, where the object border search is performed on a set of normal lines defined on the predicted surface, are commonly preferred during the measurement process. In an early work by Blake et al., Kalman filtering was used for tracking B-spline models deformed in an affine shape space [1]. In their study, the normal displacements were determined by selecting the gradient maxima of the image intensity profiles. Later, this framework was utilized with a principal component analysis based shape space for LV tracking in 2D ultrasound [2]. A local-phase edge detector [3] was applied for the measurements, and improved results were reported. More recently, a framework that uses extended Kalman filtering for tracking subdivision surfaces in 3D image data sets was introduced by Orderud et al. [4]. In their study, a step criterion was applied for the detection of the edges [5]. Common in all these studies, for a given contour/mesh, the normal displacement for each edge position is found independently from the other edge positions, which does not exploit the relationship between the neighboring intensity profiles.

Given a shape model positioned in a close proximity of the target object border, max flow/min-cut algorithms from the combinatorial optimization can be used for updating the model points closer to their target locations. This may be achieved by processing the graph consisting of the information gathered from a narrow-band around the target object contour. In the graph-cut based active contours study [6], the optimal object contour was located by calculating the minimal cut for a narrow-band graph iteratively. Later, a similar approach was used for the segmentation of elliptical objects [7]. The remainder of this paper illustrates how the max-flow / min-cut algorithm in narrow-band graphs may be utilized for the endocardial border detection.

## 3.2 Methods

### 3.2.1 Tracking framework

The framework is built around a deformable subdivision model parametrized by a set of control vertices and their associated displacement direction vectors. Model deformations are handled by a composite transform, where local shape deformations are obtained by moving control vertices in the subdivision model together with a global transformation that translates, rotates and scales the whole model.

A manually constructed Doo-Sabin surface is used to represent the endocardial borders. This model consists of 20 control vertices that are allowed to move in the surface normal direction to alter the shape. The edge detection is conducted from a set of approximately 500 surface points, spread evenly across the endocardial surface.

The tracking framework consists of five separate stages, namely the (1) state prediction, (2) evaluation of tracking model, (3) edge measurements, (4) measurement assimilation, and (5) measurement update. In this study, the stages are identical as in [4], and therefore not covered. The endocardial edge detection performed at the *edge measurements* stage is further investigated.

### 3.2.2 Edge detection methods

The edge detection process is performed by first extracting $N$ intensity profiles $I = \{I_i | i \in \{1, ..., N\}\}$ , where each profile is centered around a surface point $p_i$ and oriented in a surface normal direction $n_i$. The total number of samples in each profile, $K$, and the distance between consecutive samples are determined empirically. $I_{i,k}$ is used for referring to the intensity value of the $i^{\text{th}}$ intensity profile's $k^{\text{th}}$ sample ($I_{i,\frac{K}{2}}$ gives the intensity value at $p_i$). The function $L$ gives the index of the most probable edge in each intensity profile, and is described for different edge detection methods in the following subsections.

#### Step criterion edge detector (STEP)

STEP assumes that the intensity profile $I_i$ forms a transition from one intensity plateau to another. It calculates the average intensity heights of the two plateaus for each index value, and selects the index with the lowest model-data disagreement. The measurement noise can be set as the inverse of the height difference between the plateaus. For each profile, the edge index is determined as:

$$L_i = \arg \min_{k \in \{0 \ldots K-1\}} \left( \sum_{t=0}^{k} \left| \left( \frac{1}{k+1} \sum_{j=0}^{k} I_{i,j} \right) - I_{i,t} \right| + \sum_{t=k+1}^{K-1} \left| \left( \frac{1}{K-k-1} \sum_{j=k+1}^{K} I_{i,j} \right) - I_{i,t} \right| \right). \quad (3.1)$$

#### Max flow / min cut edge detector (MFMC)

Max flow/min cut algorithms from combinatorial optimization can be used for finding the global optima of a set of important energy functions [8]. A very common energy function that is addressed by Greig et al. can be expressed as:

$$E(f) = \sum_{v \in V} D_v(f_v) + \sum_{(v,y) \in Edges} Q_{v,y}(f_v, f_y). \quad (3.2)$$

The optimization process seeks a labeling function $f$ that assigns binary values to the nodes that are defined under a set $V$, distinguishing the inside of the LV cavity ($f = 1$) from the outside ($f = 0$). The classification is constrained by *data penalty* $D_v$, and *interaction potential* $Q_{v,y}$ functions. In this setting, (1) $D_v$ penalizes the labeling

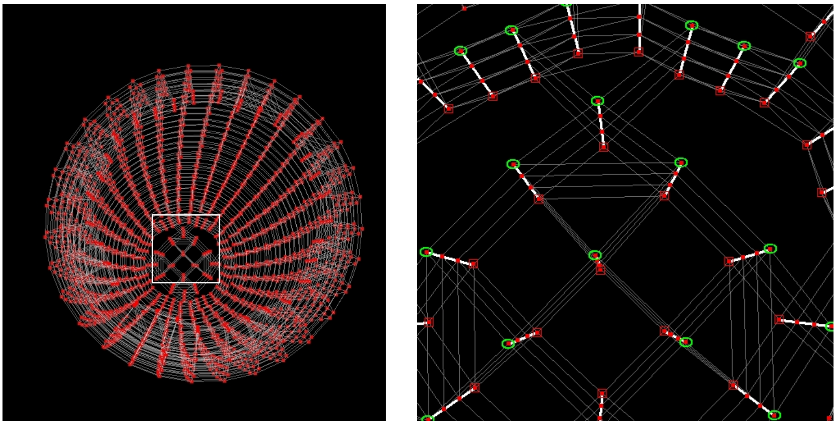


Figure 3.1: Left: The graph nodes are shown with red, and the n-links with gray. Right: (A closer look at the graph) the graph nodes are shown with red, the nodes having t-links with the source are shown in red boxes, the nodes having t-links with the sink are shown in ellipsoids, the inter-profile n-links are shown with gray, and the in-profile n-links with white.

of $v$ based on the predefined likelihood function, and (2) $Q_{v,y}$ penalizes the labeling discontinues between the neighboring nodes $v$ and $y$.

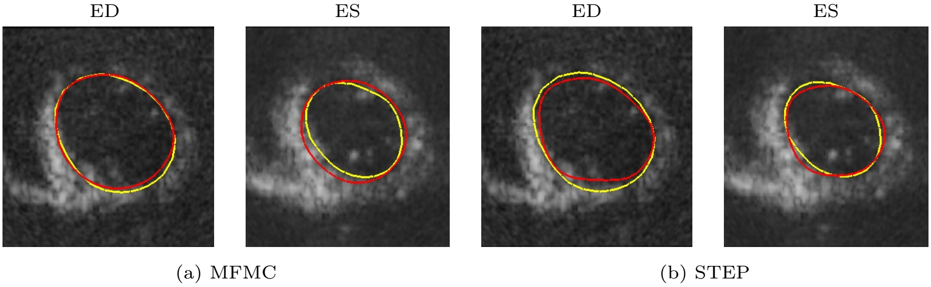
The problem of finding the optimal edges for a set of intensity profiles is formulated as in *eqn-5.11* by the MFMC method. Initially, a graph with nodes corresponding to the each profile sample is created. Two additional terminal nodes, the source and the sink, corresponding to the inside and the outside of the endocardium are appended to the node-set. The source and the sink are connected by edges to the nodes corresponding to the first and the last members of the intensity profiles respectively. The edges connecting the terminals to the other nodes are referred as the *t-links*, and they are set with infinite weights, which guarantees null data penalties in *eqn-5.11.*

The nodes corresponding to (1) the consecutive samples of the same profile, and (2) the same index samples of the neighboring profiles, are connected by undirected weighted edges called the *n-links.* For the first case, the weight of an n-link that connects nodes $v$ and $y$ can be calculated as:

$$weight(v, y) = C \times \exp\left(\frac{-\left(I_v - I_y\right)^2}{2\sigma^2}\right), \tag{3.3}$$

where $I_v$ and $I_y$ refers to the intensity values at the associated profile samples and $C$ is a constant.

For the second case, when the inter-profile connections are formed, the weight function from *eqn-5.12* is multiplied with a *smoothness constant.* The smoothness constant ensures that the final cut finds the proximate edge indexes for the neighboring profiles (see figure 1).



(a) MFMC

(b) STEP

Figure 3.2: The reference contours are shown with yellow, and the tracker result with red. (a) MFMC edges are fitting well to the reference at ED, there is an over-estimation happening at ES. (b) STEP edges are fitting well to the reference at ES, there is an under-estimation happening at ED.

After the graph is created, the maximum flow / minimum cut between the source and the sink nodes are found by the push-relabel algorithm [9] in a polynomial time $O\left(|V|^3\right)$. The resulting cut defines the edge positions for all intensity profiles simultaneously. Furthermore, the reverse of the flow is proportional with the quality of the cut, and therefore can be utilized as the measurement noise in the Kalman filter.

### Tradeoffs for the sole applications of the edge detectors

Including the trabeculae in the LV volume is a commonly accepted practice for enhancing the reproducibility of the LV measurements [10]. At the end-diastole (ED), the trabeculations are filled with blood, and therefore cause strong intensity changes across the endocardial surface. STEP criteria is prone to including trabeculae with the myocardium, since it processes each intensity profile independently. MFMC, on the other hand, is observed to behave better in this scenario, since the smoothness criteria constrained by the inter-profile n-links leads to more consistent edge indexes. The trabeculations are thus included with the cavity (see figure 2).

At the end-systole (ES), the edge detection problem attains a different form. Due to the compression and folding, the trabeculae appears as a part of the myocardium, making the intensity profiles more continuous. The inter-profile n-links that previously was an advantage for MFMC at ED then turns into a burden. The smoother profiles eliminate the need for them, and they limit the agility of the model when it is most needed. On the other hand, STEP performs well at ES (see figure 2).

### Hybrid edge detector

The weighted average of MFMC and STEP edge indexes, a *hybrid* edge index, is computed for each intensity profile during the cardiac cycle. The weight factor is determined by the size of the endocardial mesh. As the mesh size converges to its maxima, the relative weight for the MFMC increases, and as it converges to its

| Surface | ED [mm] | ES [mm] |
|---|---|---|
| STEP | $3.06 \pm 2.03$ | $3.10 \pm 2.21$ |
| MFMC | $2.62 \pm 1.74$ | $3.70 \pm 2.24$ |
| Hybrid | $2.50 \pm 1.49$ | $3.02 \pm 1.88$ |

Table 3.1: Bland-Altman analysis of the surface measurements: mean error±1.96SD.

| Volumetric | EDV [%] | ESV [%] | EF [%] |
|---|---|---|---|
| STEP | $-11.4 \pm 9.0$ | $2.5 \pm 35.2$ | $-6.2 \pm 14.0$ |
| MFMC | $2.9 \pm 21.0$ | $23.0 \pm 49.1$ | $-8.3 \pm 11.7$ |
| Hybrid | $-2.8 \pm 13.9$ | $6.7 \pm 36.8$ | $-3.8 \pm 11.5$ |

Table 3.2: Bland-Altman analysis of the volumetric measurements: mean error±1.96SD.

minima, the relative weight for the STEP increases. For each profile, the edge index is determined by finding

$$L_i^{hybrid} = \beta L_i^{MFMC} + (1 - \beta) L_i^{STEP}, \tag{3.4}$$

where $\beta = (meshsize(t) - meshsize(ES))/(meshsize(ED) - meshsize(ES))$.

## 3.3 Results

A set of 10 apical 3D echocardiography recordings, which includes 3 normal cases and 7 cases from patients with heart diseases, was used for the evaluation. The recordings were acquired using a Vivid 7 ultrasound scanner (GE Vingmed Ultrasound, Norway) using a matrix array transducer. The STEP, MFMC and Hybrid edge detectors were each used in connection to the existing contour tracking framework. Tracked 3D meshes were extracted after running the tracker through 3 cardiac cycles for a convergence. The accuracy of the edge detectors were evaluated by comparing the extracted meshes against verified reference meshes by a medical expert using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway).

A Doo-Sabin endocardial model controlled by 20 control points was used in the tracking framework. The edge measurements were performed on 528 intensity profiles evenly distributed around the endocardial model. Each profile consisted of 30 samples spaced $1mm$ apart. For the STEP, the normal displacement measurements that were significantly different from their neighbors were discarded as outliers.

In table 1, a Bland-Altman analysis for the surface error measurements is provided. The distribution of the average distance between the tracked and the reference surface points for each edge detector is given. In figure 3, color coded surface error maps are represented for a sample case. In table 2, LV cavity volume errors for ED and ES, and the ejection fraction (EF) errors, all in percentages, are reported.

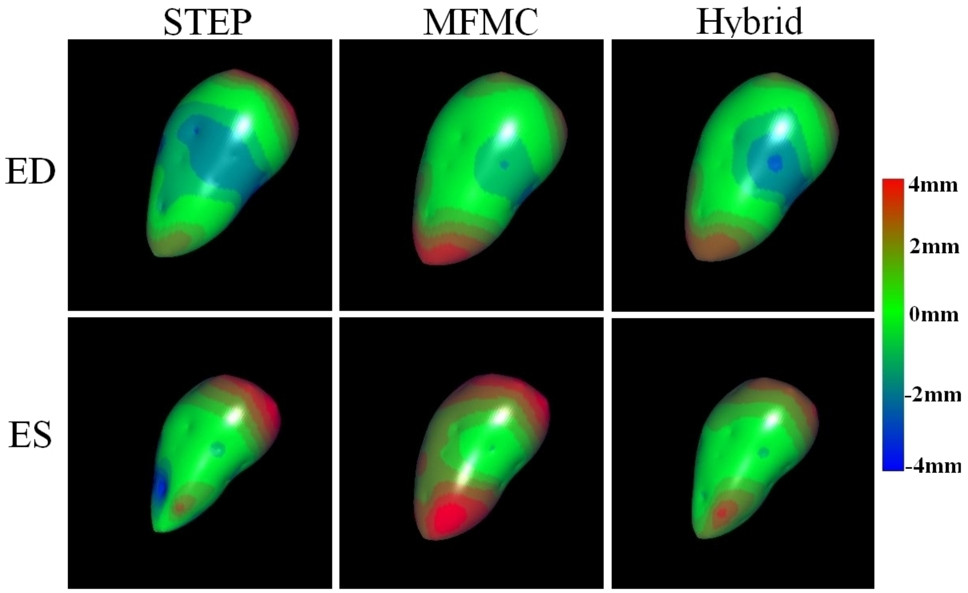


Figure 3.3: The signed surface errors are represented by using a color coding: 4*mm* over-estimation is *red,* 4*mm* under-estimation is *blue,* 0*mm* no-error is light green. ED phase signed errors (the upper row), and ES phase signed errors (the lower row) are shown for a case.

The tracking framework is implemented in C++, and processed each frame in 7.5ms with STEP, 78ms with MFMC, and 80ms with Hybrid when executed on a 2.80 GHz Intel Core 2 Duo CPU.

## 3.4 Discussion and conclusions

In this paper, a graph-cut based edge detection approach MFMC, a hybrid approach combining MFMC and STEP, and their integration into a Kalman filter based tracking framework have been proposed.

Comparative evaluation of the STEP, MFMC and Hybrid showed that the Hybrid leads to improved endocardial surface segmentation results, and hence volumetric measurements. As the accuracy of EF estimation is determined by both ED and ES surface segmentation results, there exists a significant estimation improvement from the closest 6.2% error average produced by STEP to 3.8% using Hybrid.

The tracking framework utilizes a Doo-Sabin surface, which generates the model points evenly distributed around a model. This is ideal for the MFMC implementation proposed in this work. Otherwise, the design of the inter-profile n-link weights should also factor in the varying distances between the profiles.

# References

[1] 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.

[2] G. Jacob, J. A. 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, 1999.

[3] S. Venkatesh and R. Owens, "On the classification of image features," *Pattern Recogn. Lett.*, vol. 11, no. 5, pp. 339–349, 1990.

[4] F. Orderud and S. I. Rabben, "Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces," in *CVPR*, 2008.

[5] S. I. Rabben, A. H. Torp, A. Støylen, S. Slørdahl, K. Bjørnstad, B. O. Haugen, and B. Angelsen, "Semiautomatic contour detection in ultrasound m-mode images," *Ultrasound in Medicine & Biology*, vol. 26, no. 2, pp. 287 – 296, 2000.

[6] N. Xu, N. Ahuja, and R. Bansal, "Object segmentation using graph cuts based active contours," *Comput. Vis. Image Underst.*, vol. 107, no. 3, pp. 210–224, 2007.

[7] G. G. Slabaugh and G. B. Unal, "Graph cuts segmentation using an elliptical shape prior," in *ICIP (2)*, pp. 1222–1225, 2005.

[8] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 2, pp. 271–279, 1989.

[9] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[10] T. Papavassiliu, H. P. Kuhl, M. Schroder, T. Suselbeck, O. Bondarenko, C. K. Bohm, 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 Imaging1," *Radiology*, vol. 236, no. 1, pp. 57–64, 2005.

## Chapter 4

# Polynomial Regression Based Edge Filtering for Left Ventricle Tracking in 3D Echocardiography

Engin Dikici[1], Fredrik Orderud[2]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

Automated detection of endocardial borders in 3D echocardiography is a challenging task. Part of the reason for this is the endocardial boundary leads to alternating edge characteristics that vary over a cardiac cycle. The maximum gradient (MG), step criterion (STEP) and max flow/min cut (MFMC) edge detectors have been previously applied for the endocardial edge detection problem. In this paper, a local polynomial regression based method (LPR) is introduced for filtering the STEP results. For each endocardial model point, (1) the surface is parametrized locally around the point, (2) a polynomial regression is applied on the STEP edges in the parametric domain, and (3) the fitted polynomial is evaluated at the origin of the parametric domain to determine the endocardial edge position. The effectiveness of the introduced method is validated via comparative analyses among the MFMC, STEP, and first & second degree LPR methods.

## 4.1 Introduction

3D echocardiography has enabled real-time, non-invasive and low cost acquisition of volumetric images of the LV. The problem of automatic detection and tracking of heart chambers in ultrasound images has received considerable attention lately [1, 2]. However, the accurate detection of the endocardial borders remains a challenging task. This is partially due to the trabeculated structure of the endocardial borders, which leads to alternating edge characteristics over a cardiac cycle. Furthermore, the real-time imaging capability of 3D ultrasound requires highly time-efficient algorithms.

One approach for the LV detection is to use a Kalman filter based tracking framework to update a deformable model based on the edge measurements. In an early work by Blake et al., Kalman filtering was used for tracking B-spline models deformed in an affine shape space [3]. In their study, object boundaries were determined by selecting the gradient maxima (MG) of image intensity profiles. Later, this framework was utilized with a principal component analysis based shape space for the LV tracking in 2D ultrasound by Jacob et al.[4, 5]. This study employed a local-phase edge detector [6] for the edge measurements, and reported visually enhanced results compared to the maximum gradient method. Orderud et al. utilized an extended Kalman filter to track deformable subdivision surfaces in 3D image data sets [2]. The latter work used a step criterion (STEP) [7] for the detection of endocardial edges. More recently, Dikici et al. applied the max flow / min cut algorithm (MFMC) for the detection of endocardial edges in a Kalman tracking framework [8].

Local polynomial regression is a simple and effective method for nonparametric regression. It has been applied for many tasks including the multivariate prediction [9], image filtering [10] and image reconstruction [11]. In this paper, we introduce a novel local polynomial regression based edge filtering approach (LPR) for smoothing the STEP results in a parametric domain. First, the STEP edges are calculated at evenly distributed positions around an endocardial model. Then, the detected STEP edges are filtered by a local polynomial regression using a kernel weighting scheme. The major motivation for this work is to improve the edge detection quality offered by STEP method while still providing a real-time solution. The effectiveness of the introduced method is validated via comparative analyses among the MFMC, STEP, and first & second degree LPR methods.

## 4.2 Tracking Framework

The tracking framework is built around a deformable subdivision model parametrized by a set of control vertices with associated displacement direction vectors. Shape and pose deformations are handled by a composite transform $T = T_g(T_l(x_l), x_g)$, where local shape deformations $T_l(x_l)$ are obtained by moving control vertices in the subdivision model together with a global transformation $T_g(p_l, x_g)$ that translates, rotates and scales the whole model. This leads to a composite state vector $x$, consisting of $N_g$ global and $N_l$ local parameters.

A manually constructed Doo-Sabin surface is used as a template for representing the endocardial borders. The control vertices are allowed to move in the surface normal direction to alter the shape. The edge detection is conducted from a set of evenly distributed endocardial surface points.

The tracking framework consists of five separate stages, which will be described briefly in the following subsections (please refer to [2] for further details).

### 4.2.1 State Prediction

A *motion model* for predicting the state vector $\overline{x}$ at time time $k+1$ is formulated as:

$$\overline{x}_{k+1} - x_0 = A_1(\hat{x}_k - x_0) + A_2(\hat{x}_{k-1} - x_0), \tag{4.1}$$

where $\hat{x}_k$ is the estimated state from time-step $k$, and $x_0$ is the initial state. Temporal properties like damping and regularization towards $x_0$ can be adjusted using coefficients in the matrices $A_1$ and $A_2$. Prediction uncertainty can similarly be changed by manipulating the process noise covariance matrix used in the associated covariance update equation.

### 4.2.2 Evaluation of Tracking Model

A set of surface points $p$ with associated normal vectors $n$ are calculated from the predicted state. Then, the state-space Jacobi matrices relating surface point position changes to state changes are found. The composite deformation model leads to Jacobi matrices including both the global and local derivatives:

$$J_g = \left[\frac{\partial T_g(p_l,x_g)}{\partial x_g}, \frac{\partial T_g(p_l,x_g)}{\partial p_l} J_l\right]. \tag{4.2}$$

### 4.2.3 Edge Measurements

The predicted model is guided towards the target object using edge measurements. Edge detection is conducted in the surface normal direction $n_i$ from each point $p_i$ on the predicted surface (different methods for this part are elaborated in Section-3). The end result is a *normal displacement* value $v_i$ that gives the signed distance between the detected edge $p_{obs,i}$ and the surface point:

$$v_i = n_i^T(p_{obs,i} - p_i). \tag{4.3}$$

Each normal displacement measurement is coupled with a measurement noise $r_i$ that specifies the spatial uncertainty of the detected edge. Associated measurement vectors $h_i$ for each edge are computed by taking the normal vector projection of the state-space Jacobi matrices:

$$h_i^T = n_i^T J. \tag{4.4}$$

### 4.2.4 Measurement Assimilation

All measurement results are assimilated in an information space with the assumption of uncorrelated measurements. This allows for efficient weighted summation of all measurement results into information vector and matrix with dimensions invariant to the number of measurements:

$$H^T R^{-1} v = \sum_i h_i r_i^{-1} v_i, \tag{4.5}$$

$$H^T R^{-1} H = \sum_i h_i r_i^{-1} h_i^T. \tag{4.6}$$

### 4.2.5 Measurement Update

The measurement information is combined with the predicted state to compute an updated state estimate. By using the information filter formulation of the Kalman filter, the updated state estimate $\hat{x}$ for a time step $k$ becomes:

$$\hat{x}_k = \bar{x}_k + \hat{P}_k H^T R^{-1} v_k, \tag{4.7}$$

where an updated error covariance matrix $\hat{P}_k$ can also be calculated in the information space to avoid inverting large matrices:

$$\hat{P_k}^{-1} = \bar{P}_k^{-1} + H^T R^{-1} H. \tag{4.8}$$

## 4.3 Edge Detection

The edge detection process is performed by first extracting $N$ intensity profiles $(I_1, I_2 \cdots I_N)$ centered around the surface points $p_i$ and oriented in the surface normal directions $n_i$. The total number of samples in each profile, $M$, and the distance between consecutive samples are determined empirically. $I_{i,m}$ is used for referring to the intensity value of the $i^{\text{th}}$ intensity profile's $m^{\text{th}}$ sample. The function $L$ gives the index of the most probable edge in each intensity profile, and is described for the STEP and LPR methods in the following subsections.

### 4.3.1 Step Criterion Edge Detector (STEP)

STEP assumes that the intensity profile $I_i$ forms a transition from one intensity plateau to another. It calculates the heights of the two plateaus for each index value, and selects the index with the lowest sum of squared differences between the criteria and the image data. For each profile, the edge index is determined as:

$$L_i = argmin_m \sum_{t=1}^{m} \left( \left( \frac{1}{m} \sum_{j=1}^{m} I_{i,j} \right) - I_{i,t} \right)^2 + \sum_{t=m+1}^{M} \left( \left( \frac{1}{M-m} \sum_{j=m+1}^{M} I_{i,j} \right) - I_{i,t} \right)^2 . \tag{4.9}$$

If the plateau heights for the determined edge index are similar $(L_i = m \; and \; \frac{1}{m} \sum_{j=1}^{m} I_{i,j} = \frac{1}{M-m} \sum_{j=m+1}^{M} I_{i,j})$, then the edge index is reset to the profile center by $L_i = \frac{M}{2}$. The measurement noise is defined inversely proportional with the height difference between the plateaus.

### 4.3.2 Local Polynomial Regression Edge Detector (LPR)

STEP method processes each intensity profile independently, which may cause discontinuous edge measurement over an endocardial model. The discontinuity problem can be resolved by filtering the measurements via local polynomial regression. For applying a local polynomial regression, (1) the local neighborhood for each

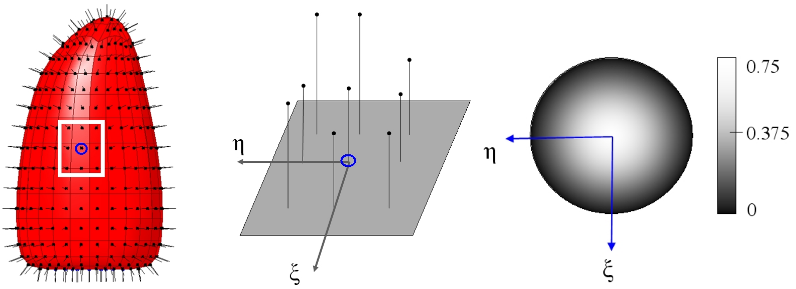


Figure 4.1: (Left) An intensity profile and its local neighborhood borders are shown, (middle) the local coordinate system for the selected intensity profile is represented, (right) Epanechnikov quadratic kernel weights are shown.

intensity profile, (2) a weighting function or a *kernel,* and (3) the model degree are needed to be defined.

The distance between the intensity profiles $I_i$ and $I_j$ is defined as the Cartesian distance between their intensity profile centers by $\Gamma_{i,j} = |p_i - p_j|$. The local neighborhood of the $i^{th}$ intensity profile is called $K_i$, and it includes $I_j$ iff $\Gamma_{i,j} < \lambda$: *kernel radius.* For a notational simplicity, $K_{i,j}$, $K_{i,j}^{(p)}$, and $K_{i,j}^{(l)}$ are used for referring to the $i^{th}$ neighborhood's $j^{th}$ member *($K_{i,j}$)*, the member's intensity profile center *($K_{i,j}^{(p)}$)*, and the member's measured STEP edge position *($K_{i,j}^{(l)}$)* respectively.

The local coordinate system for $K_i$ can be defined as,

$$e_i = \begin{cases} [1,\, 0,\, 0]^T & if\ n_i \neq [1,\, 0,\, 0]^T \\ [0,\, 1,\, 0]^T & else. \end{cases} \tag{4.10}$$

$$\vec{V}_1 = n_i, \quad V_2 = \vec{V}_1 \times e_i, \quad \vec{V}_3 = \vec{V}_1 \times \vec{V}_2. \tag{4.11}$$

Each member of $K_i$ can be parametrized using $\xi$ and $\eta$ parameters that can be found by,

$$K_{i,j}^{(\xi)} = \left(K_{i,j}^{(p)} - p_i\right) \cdot \vec{V}_2, \tag{4.12}$$

$$K_{i,j}^{(\eta)} = \left(K_{i,j}^{(p)} - p_i\right) \cdot \vec{V}_3, \tag{4.13}$$

where $K_{i,j}^{(\xi)}$ and $K_{i,j}^{(\eta)}$ refer to $\xi$ and $\eta$ parameters of the $i^{th}$ neighborhood's $j^{th}$ member respectively.

The STEP edges can be averaged locally for generating smoother results using $\hat{L}_i = Ave\left(K_{i,j}^{(l)} | j \in \{1, 2 \ldots |K_i|\}\right)$. However, this method might still lead to abrupt discontinuities due to constant weight function. Rather than giving all the points equal weights, we can assign weights that die off smoothly with the distance from the neighborhood center [12]. *Nadaraya–Watson* kernel-weighted average,

$$\hat{L}_i = \frac{\sum_{j=1}^{|K_i|} w_\lambda \left(p_i,\, K_{i,j}^{(p)}\right) K_{i,j}^{(l)}}{\sum_{j=1}^{|K_i|} w_\lambda \left(p_i,\, K_{i,j}^{(p)}\right)}, \tag{4.14}$$

with the *Epanechnikov* quadratic kernel,

$$w_\lambda \left(p,\, q\right) = 0.75 \left(1 - \left(\frac{|p-q|}{\lambda}\right)^2\right), \tag{4.15}$$

can be used for this weighted filtering task *(see Figure 4.1)*. It can be shown that the Nadara-Watson method solves a weighted least squares problem at each intensity profile by,

$$min_{\beta_0} \sum_{j=1}^{|K_i|} w_\lambda \left(p_i,\, K_{i,j}^{(p)}\right) \left[K_{i,j}^{(l)} - \beta_0\right]^2, \tag{4.16}$$

where the estimate is $\hat{L}_i = \beta_0$. Since $\beta_0$ is a $0^{th}$ degree polynomial, the introduced regression is a $0^{th}$ degree local polynomial regression. This filter might produce high estimation bias due to the fact that a local polynomial regression of degree $D$ only has the bias terms of degree $(D+1)$ and higher (see Appendix). Therefore, a higher degree polynomial regression model should lead to a lower estimation bias, while producing higher estimation variance and computational cost. Accordingly, the model degree should be set considering this tradeoff.

$D^{th}$ degree local polynomial regression plane defined in the parametric coordinates *($\xi$, $\eta$)* solves,

$$min_{\beta_0 \ldots \beta_M} \sum_{j=1}^{|K_i|} w_\lambda \left(p_i,\, K_{i,j}^{(p)}\right) \left[K_{i,j}^{(l)} - \left(\beta_0 + \beta_1 K_{i,j}^{(\xi)} + \beta_2 K_{i,j}^{(\eta)} \ldots \beta_M \left(K_{i,j}^{(\eta)}\right)^D\right)\right]^2. \tag{4.17}$$

The regression plane needs to be evaluated at the parametric domain's center *($\xi = 0$, $\eta = 0$)* to determine the filtered edge position. This calculation can be performed using a matrix notation as $\hat{L} = b\left(0,0\right)\left(B^T W B\right)^{-1} B^T W y$, where

$$b\left(\xi,\eta\right) = \left[1,\, \xi,\, \eta,\, \xi^2,\, \eta^2,\, \xi\eta \ldots \eta^D\right], \tag{4.18}$$

$$B = \left[b\left(K_{i,1}^{(\xi)},\, K_{i,1}^{(\eta)}\right)^T,\, b\left(K_{i,2}^{(\xi)},\, K_{i,2}^{(\eta)}\right)^T \ldots b\left(K_{i,|K_i|}^{(\xi)},\, K_{i,|K_i|}^{(\eta)}\right)^T\right]^T, \tag{4.19}$$

$$y = \left[K_{i,1}^{(l)},\, K_{i,2}^{(l)} \ldots K_{i,|K_i|}^{(l)}\right]^T, \tag{4.20}$$

and $W$ is a $|K_i| \times |K_i|$ diagonal matrix with $j^{th}$ diagonal element $w_\lambda\left(p_i,\, K_{i,j}^{(p)}\right)$. In Figure 4.2, $0^{th}$, $1^{st}$ and $2^{nd}$ degree regression planes are represented for a given STEP data.

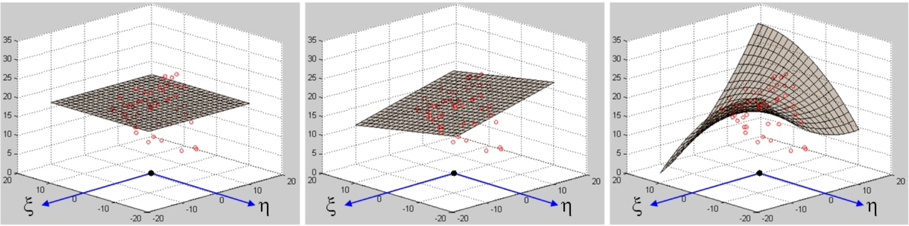


Figure 4.2: (Left) $0^{th}$ degree, (middle) $1^{st}$ degree - linear, (right) $2^{nd}$ degree - quadratic planes are fit on the STEP results in the parametric domain.

| | **Cycle-1** | **Cycle-2** | **Cycle-3** | **Cycle-4** | **Cycle-5** |
|---|---|---|---|---|---|
| **STEP** | 7.15 – 2.44 | 2.90 – 2.18 | 2.95 – 2.19 | 2.94 – 2.20 | 2.97 – 2.20 |
| **MFMC** | 5.27 – 2.48 | 2.43 – 2.46 | 2.43 – 2.47 | 2.38 – 2.46 | 2.42 – 2.45 |
| **LPR-1** | 6.87 – 2.29 | 2.70 – 2.16 | 2.61 – 2.17 | 2.61 – 2.16 | 2.60 – 2.18 |
| **LPR-2** | 6.86 – 2.30 | 2.62 – 2.08 | 2.56 – 2.11 | 2.57 – 2.07 | 2.58 – 2.08 |

Table 4.1: Mean surface error *(in mm)* for the ED and ES frames for the first 5 cardiac cycles [ED error - ES error]. The tracker converged after the first cycle; surface error measurements deviated in small amounts in the following cycles.

## 4.4 Results

A set of 17 apical 3D echocardiography recordings, which includes 10 normal cases and 7 cases from patients with heart diseases, was used for the evaluation. The recordings were acquired using a Vivid 7 ultrasound scanner (GE Vingmed Ultrasound, Norway) and a matrix array transducer. Local polynomial regression based edge filtering method was implemented for the $1^{st}$ (LPR-1) and $2^{nd}$ degree (LPR-2) polynomials both with $1cm$ kernel radius ($\lambda$). MFMC (as introduced in [8]), STEP, LPR-1 and LPR-2 methods were each employed in connection to the existing Kalman tracking framework. 3D meshes were extracted after running the tracker through 3 cardiac cycles for a convergence (see Table 4.1 for the surface error convergences). The accuracy of the edge detectors were evaluated by comparing the extracted meshes against the verified reference meshes drawn by a medical expert using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway).

A handcrafted Doo-Sabin endocardial model consisting of 20 control points was used as the LV model. Edge measurements were performed in 528 intensity profiles evenly distributed across the endocardial model. Each profile consisted of 30 samples, spaced 1 $mm$ apart.

Table 4.2 shows Bland-Altman analyses for the LV surface, LV cavity volume, and the associated ejection fraction (EF) agreement. The color coded surface error maps of a sample case are represented in Figure 4.3 rows (A) and (C).

The tracking framework is implemented in C++, and processed each frame in

| | ED [mm] | ES [mm] | EDV [%] | ESV [%] | EF [%] |
|---|---|---|---|---|---|
| **STEP** | $2.94 \pm 1.56$ | $2.20 \pm 1.39$ | $-23.01 \pm 16.05$ | $-13.03 \pm 24.48$ | $-6.05 \pm 10.09$ |
| **MFMC** | $2.38 \pm 1.43$ | $2.46 \pm 1.51$ | $-7.44 \pm 24.53$ | $12.76 \pm 42.92$ | $-8.14 \pm 10.50$ |
| **LPR-1** | $2.61 \pm 1.92$ | $2.16 \pm 1.47$ | $-19.28 \pm 17.98$ | $-10.54 \pm 24.42$ | $-4.98 \pm 9.76$ |
| **LPR-2** | $2.57 \pm 1.95$ | $2.07 \pm 1.52$ | $-18.02 \pm 19.02$ | $-9.03 \pm 24.01$ | $-5.02 \pm 8.67$ |

Table 4.2: Columns 1-2: Mean surface error±1.96SD for the ED and ES frames. Columns 3-4: Mean LV cavity volume error±1.96SD for the ED and ES frames. Column 5: Mean EF error±1.96SD.

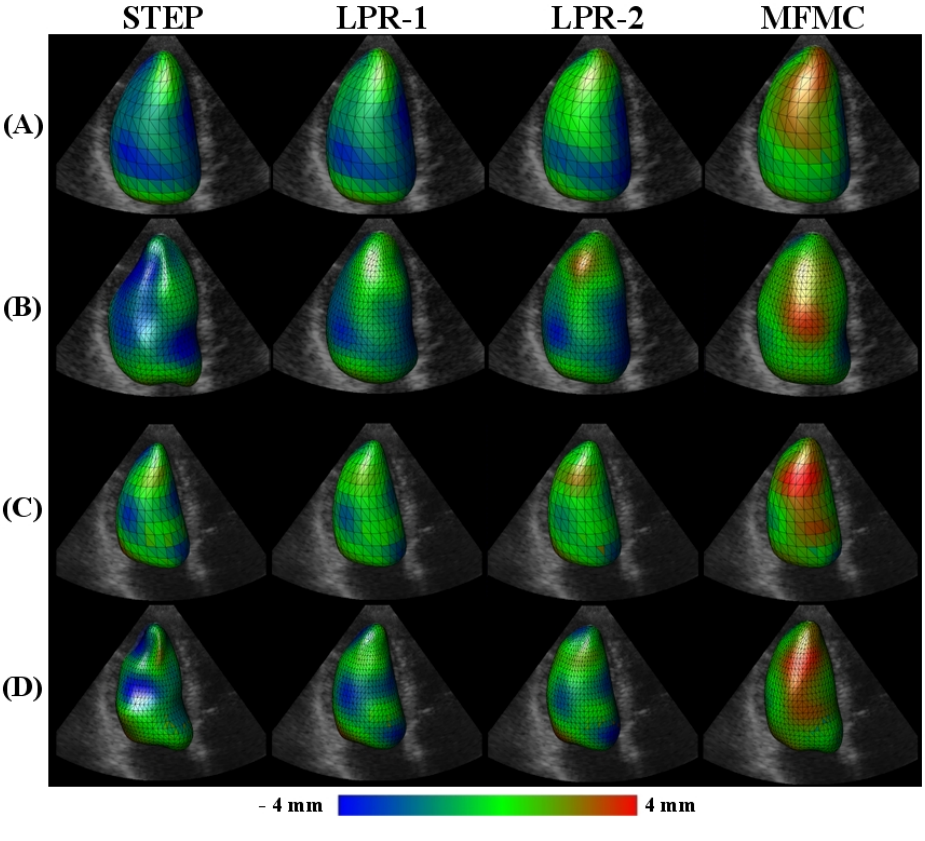


Figure 4.3: For a sample case, the signed surface errors for the (A) ED phase of the original model, (B) ED phase of the refined model, (C) ES phase of the original model, and (D) ES phase of the refined model. The original and refined models consist of 20 and 84 control points respectively. *(The original model was used for the surface and volumetric analyses provided in this paper).*

7.5 $ms$ with STEP, 78 $ms$ with MFMC, 23.7 $ms$ with LPR-1, and 40.8 $ms$ with LPR-2 when executed on a 2.80 GHz Intel Core 2 Duo CPU.

## 4.5 Discussion and Conclusion

We have introduced a local polynomial regression based filtering for the STEP edges. The proposed approach was implemented for the first and second degree polynomial regression models. The method description is provided in a degree-independent fashion; hence the generalization of the method for higher degrees should be an intuitive task. Increasing the degree of regression model lowers the bias component of the mean square error (MSE), while increasing the variance component. Therefore, a proper degree should be selected by considering the bias-variance tradeoff. In a future study, optimal kernel radius and the regression order can be learned from a training data statistically.

A comparative evaluation between the edge detectors showed that both LPR-1 and LPR-2 lead to improved surface and volumetric measurement accuracies over the STEP method. For the ED phase, STEP, LPR-1 and LPR-2 produced 2.94 $mm$, 2.61 $mm$ (12% improvement) and 2.57 $mm$ (13% improvement) mean surface errors. LPR-1 and LPR-2 filters also reduced the LV cavity volume error of the STEP method at the ED phase by 3.73% and 5% respectively. Comparable surface and volumetric measurement improvements were reported for the ES phase (see Table 4.2).

The control point resolution of the endocardial model is another smoothing factor for the Kalman tracking framework. A higher resolution endocardial model, generated by refining the original model via Doo-Sabin subdivision rules [13], can represent a wider range of deformations. Hence, the effects of edge filtering becomes visually more assessable for the refined model (see Figure 4.3 rows (B) and (D)). Multiresolution Doo-Sabin surface models with the measurement filtering might also be investigated in a future study.

## Appendix: Local Polynomial Regression Bias

$D^{th}$ degree local polynomial regression curve for 1D data defined at $x_0$ as,

$$\hat{f}\left(x_0\right) = \left[1,\, x_0 \ldots x_0^D\right] \left(B^T W B\right)^{-1} B^T W y = \sum_{i=1}^{N} l_i\left(x_0\right) y_i \tag{4.21}$$

$$E\left[\hat{f}\left(x_0\right)\right] = f\left(x_0\right) \sum_{i=1}^{N} l_i\left(x_0\right) + f^{'}\left(x_0\right) \sum_{i=1}^{N} \left(x_i - x_0\right) l_i\left(x_0\right) \cdots \tag{4.22}$$

*Lemma 1:* $\sum_{i=1}^{N} l_i\left(x_0\right) = 1$.

*Proof:* Assume that all $y_i = 1$. Since $l_{i\in\{1,2\ldots N\}}\left(x_0\right)$ do not depend on $y_i$, $\hat{f}\left(x_0\right) = \sum_{i=1}^{N} l_i\left(x_0\right) y_i = \sum_{i=1}^{N} l_i\left(x_0\right) = 1$.

*Lemma 2:* Define $b_j\left(x_0\right) = \sum_{i=1}^{N}\left(x_i - x_0\right)^j l_i\left(x_0\right)$. Then, $b_j\left(x_0\right) = 0$ for all $j \in \{1, 2 \ldots D\}$.

*Proof:* Assume that $y_i = \left(x_i - x_0\right)^D$. LPR solves

$$min_\beta \left(\sum_{m=0}^{D} C_m x_i^m \left(-x_0\right)^{D-m} - \sum_{m=0}^{D} \beta_m\left(x_0\right) x_i^m\right)^2 . \tag{4.23}$$

where $\beta_m\left(x_0\right) = C_m\left(-x_0\right)^{D-m}$ minimizes the term. Therefore,

$$\hat{f}\left(x_0\right) = \sum_{i=1}^{N}\left(x_i - x_0\right)^j l_i\left(x_0\right) = \sum_{m=0}^{D} C_m\left(-x_0\right)^{D-m} x_0^m = \left(x_0 - x_0\right)^D = 0. \tag{4.24}$$

Due to Lemma-1 and Lemma-2, a local polynomial regression of degree $D$ only has the bias terms of degree $(D+1)$ and higher.

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

# References

[1] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu, “3d ultrasound tracking of the left ventricles using one-step forward prediction and data fusion of collaborative trackers,” in *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[2] F. Orderud and S. I. Rabben, “Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces,” in *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[3] 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.

[4] G. Jacob, J. A. 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, 1999.

[5] 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, 2001.

[6] S. Venkatesh and R. A. Owens, “On the classification of image features,” *Pattern Recognition Letters*, vol. 11, no. 5, pp. 339–349, 1990.

[7] S. I. Rabben, A. H. Torp, A. Støylen, S. Slørdahl, K. Bjørnstad, B. O. Haugen, and B. Angelsen, “Semiautomatic contour detection in ultrasound m-mode images,” *Ultrasound in Medicine & Biology*, vol. 26, no. 2, pp. 287 – 296, 2000.

[8] E. Dikici and F. Orderud, “Graph-cut based edge detection for kalman filter based left ventricle tracking in 3d+ t echocardiography,” in *Proceedings of Computing in Cardiology*, 2010.

[9] L. Su, “Prediction of multivariate chaotic time series with local polynomial fitting,” *Computers & Mathematics with Applications*, vol. 59, no. 2, pp. 737 – 744, 2010.

[10] R. M. Palenichka and P. Zinterhof, “Structure-adaptive filtering based on polynomial regression modeling of image intensity,” *Journal of Electronic Imaging*, vol. 10, pp. 521–534, 2001.

[11] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Transactions on Image Processing*, vol. 16, pp. 349–366, feb. 2007.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, corrected ed., Aug. 2003.

[13] D. Doo and M. Sabin, “Behaviour of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978.

# Chapter 5

# Best Linear Unbiased Estimator for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography

Engin Dikici[1], Fredrik Orderud[2] and Hans Torp[1]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

In this paper, we introduce the best linear unbiased estimator (BLUE) for the detection of endocardial edges in 3D+T echocardiography recordings.
The maximum gradient (MG), step criterion (STEP) and max flow/min cut (MFMC) edge detectors have been previously applied for the detection of the endocardial edges. BLUE combines the responses of these 3 base estimators using statistical inferences. First, the base estimator bias and covariance properties are learned for each endocardial surface point at each cardiac cycle position. Then, these statistical properties are utilized to compute an optimal linear combination of the base detectors by BLUE.
For the validation, MG, STEP, MFMC and BLUE were each employed in connection to a Kalman tracking framework. Comparative analyses showed that BLUE outperform the other estimators in surface and volumetric measurement accuracy.

## 5.1 Introduction

3D echocardiography is a widely used clinical diagnosis tool as it provides real-time, non-invasive and low cost acquisition of volumetric images of the heart. Automated segmentation and tracking of heart chambers in echocardiography images have received considerable attention in the recent years [1–3]. However, accurate detection of endocardial borders remains a challenging task due to reasons including (1) speckle noise, (2) shadowing that can result in missing boundaries, and (3) the existence of intra-cavity structures such as chordae tendineae, papillary muscles and valves [4].

A Kalman filter based framework can be employed for the detection of the LV structures in time-dependent recordings. The approach adopts a sequential prediction and update strategy, where surface deformations are first predicted by using a kinematic model, then the prediction is updated based on information provided by image measurements. In an early work by Blake et al., Kalman filtering was used for tracking B-spline models deformed in an affine shape space [5]. In their study, object boundaries were determined by selecting the gradient maxima (MG) of image intensity profiles. Later, this framework was utilized with a principal component analysis based shape space for the LV tracking in 2D ultrasound by Jacob et al. [6, 7]. Their study employed a local-phase edge detector [8] for the edge measurements, and reported visually enhanced results compared to the maximum gradient method. Orderud et al. utilized an extended Kalman filter to track deformable subdivision surfaces in 3D recordings [9]. The latter work used a step criterion (STEP) [10] for the detection of endocardial edges. More recently, Dikici et al. applied the max flow / min cut algorithm (MFMC) for the detection of endocardial edges in a Kalman tracking framework [11]. Their study provided comparative analyses representing the shortcomings of STEP and MFMC methods, and accordingly proposed a hybrid edge detector.

Ensemble methods combine the responses of multiple base estimators/classifiers in some principled manner to answer a query [12]. They have been shown to be effective, since the resulting classifiers are often more accurate than the individual classifiers making up the ensemble [13]. Konishi et al. used statistical inferences to combine responses of multiple edge detectors (e.g. intensity gradient, the Laplacian of a Gaussian, filterbanks of oriented filter pairs) [14, 15]. In [16], the maximum likelihood and James-Stein estimators were employed to combine multiple endocardial edge detectors in a Kalman tracking framework. The approach had 2 major limitations: (1) the edge detectors were assumed to be uncorrelated, and (2) the statistical properties of the edge detectors were marginalized to the endocardial surface; the statistical properties did not vary over the endocardial model.

In this study, we introduce the best linear unbiased estimator (BLUE) that combines the responses of MG, STEP and MFMC detectors, which are also referred as the base estimators. The proposed system (1) learns the statistical properties of the base estimators for each endocardial surface point at each cardiac cycle position, and (2) combines the base estimator responses linearly via weights inferred from the statistical properties. It can also be shown that BLUE and the maximum likelihood estimator (MLE) lead to identical solutions for the investigated edge detection problem if the base estimator error distributions are assumed to be Gaussian. Hence, the results of our study can be used for the derivation of a Bayes estimator in a future study. The proposed method avoids the limitations of [16] by (1) factoring in the base estimator correlations and (2) calculating the base estimator properties for each surface point independently. The major motivation of our study is to improve the endocardial edge detection accuracy for the Kalman filter based tracking framework described in [9]. The effectiveness of the introduced method is represented via comparative analyses among MG, STEP, MFMC, and BLUE.

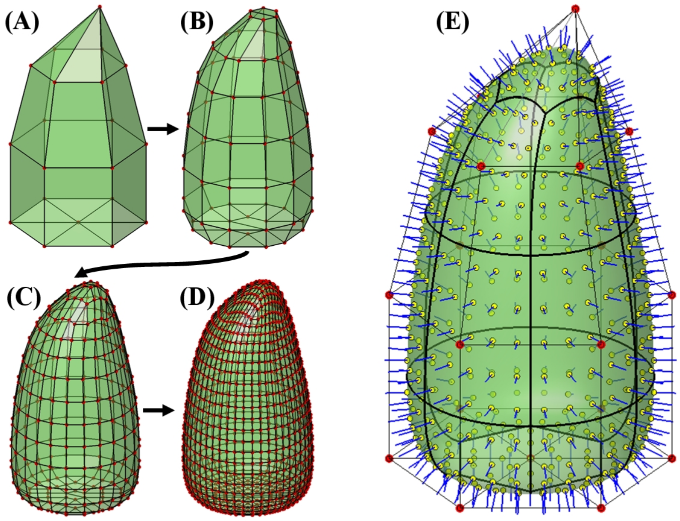


Figure 5.1: Doo-Sabin surfaces generalize bi-quadric B-spline surfaces to arbitrary topologies [17]. Three steps of Doo-Sabin refinements for a model (A) are shown in (B), (C) and (D). In (E), the endocardial model used in our study (green), its control points (red), a set of evenly distributed endocardial surface points (yellow) and the edge detection directions (blue) are represented.

## 5.2 Tracking framework

The tracking framework is built around a deformable subdivision model parametrized by a set of control vertices with associated displacement direction vectors. Shape and pose deformations are handled by a composite transform $T = T_g\left(T_l(\mathbf{x}_l), \mathbf{x}_g\right)$, where local shape deformations $T_l(\mathbf{x}_l)$ are obtained by moving control vertices in the subdivision model together with a global transformation $T_g(\mathbf{p}_l, \mathbf{x}_g)$ that translates, rotates and scales the whole model. This leads to a composite state vector $\mathbf{x} = \left[\mathbf{x}_g^T, \mathbf{x}_l^T\right]^T$, consisting of $N_g$ global and $N_l$ local parameters.

A manually constructed Doo-Sabin surface is used as a template for representing the endocardial borders. The control vertices are allowed to move in the surface normal direction to alter the shape. Edge detection is conducted from a set of evenly distributed endocardial surface points (see Figure 5.1).

The tracking framework consists of five separate stages, which will be described briefly in the following subsections (please refer to [9] for further details).

### 5.2.1 State prediction

A *motion model* for predicting the state vector $\overline{\mathbf{x}}$ at time $k+1$ is formulated as:

$$\overline{\mathbf{x}}_{k+1} - \mathbf{x}_0 = A_1\left(\hat{\mathbf{x}}_k - \mathbf{x}_0\right) + A_2\left(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_0\right), \tag{5.1}$$

where $\hat{\mathbf{x}}_k$ is the estimated state from time-step $k$, and $\mathbf{x}_0$ is the initial state. Temporal properties like damping and regularization towards $\mathbf{x}_0$ can be adjusted using coefficients in the matrices $A_1$ and $A_2$. Prediction uncertainty can similarly be changed by manipulating the process noise covariance matrix used in the associated covariance update equation.

### 5.2.2 Evaluation of tracking model

A set of surface points $\mathbf{p}$ with associated normal vectors $\mathbf{n}$ are calculated from the predicted state. Then, state-space Jacobi matrices that relate surface point position changes to state changes are computed as follows. The composite deformation model leads to Jacobi matrices including both the global and local derivatives:

$$J_g = \left[\frac{\partial T_g(\mathbf{p}_l,\mathbf{x}_g)}{\partial \mathbf{x}_g}, \frac{\partial T_g(\mathbf{p}_l,\mathbf{x}_g)}{\partial \mathbf{p}_l} J_l\right]. \tag{5.2}$$

### 5.2.3 Edge measurements

The predicted model is guided towards the target object using edge measurements. Edge detection is conducted in the surface normal direction $\mathbf{n}_i$ from each point $\mathbf{p}_i$ on the predicted surface (different methods for this part are elaborated in Section-3). The end result is a *normal displacement* value $v_i$ that gives the signed distance between the detected edge $\mathbf{p}_{obs,i}$ and the surface point:

$$v_i = \mathbf{n}_i^T \left(\mathbf{p}_{obs,i} - \mathbf{p}_i\right). \tag{5.3}$$

Each normal displacement measurement is coupled with a measurement noise $r_i$ that specifies the spatial uncertainty of the detected edge. Associated measurement vectors $\mathbf{h}_i$ for each edge are computed by taking the normal vector projection of the state-space Jacobi matrices:

$$\mathbf{h}_i^T = \mathbf{n}_i^T J. \tag{5.4}$$

### 5.2.4 Measurement assimilation

All measurement results are assimilated in an information space with the assumption of uncorrelated measurements. This allows for efficient weighted summation of all measurement results into information vector and matrix with dimensions invariant to the number of measurements:

$$H^T R^{-1} \mathbf{v} = \textstyle\sum_i h_i r_i^{-1} v_i, \tag{5.5}$$

$$H^T R^{-1} H = \textstyle\sum_i h_i r_i^{-1} h_i^T. \tag{5.6}$$

### 5.2.5 Measurement update

The measurement information is combined with the predicted state to compute an updated state estimate. By using the information filter formulation of the Kalman filter, the updated state estimate $\hat{\mathbf{x}}$ for a time step $k$ becomes:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \hat{P}_k H^T R^{-1} \mathbf{v}_k, \tag{5.7}$$

where an updated error covariance matrix $\hat{P}_k$ can also be calculated in the information space to avoid inverting large matrices:

$$\hat{P_k}^{-1} = \bar{P}_k^{-1} + H^T R^{-1} H. \tag{5.8}$$

## 5.3 Edge detection methods

The edge detection process is performed by first extracting $N$ 1D intensity profiles $(I_1, I_2, \ldots I_N)$ centered around the surface points $\mathbf{p}_i$ and oriented in the surface normal directions $\mathbf{n}_i$. The total number of samples in each profile, $K$, and the distance between consecutive samples are determined empirically. $I_{i,k}$ is used for referring to the intensity value of the $i^{\text{th}}$ intensity profile's $k^{\text{th}}$ sample. Edge detection methods, processing intensity profiles to estimate endocardial border positions, are described in the following subsections.

### 5.3.1 Maximum gradient edge detector (MG)

The intensity profile $I_i$ is convolved with a Gaussian kernel $G$ to create a smoother intensity profile. Then, a gradient profile for the smoothed profile is computed by using the forward-difference approximation. The position of the maximum of the gradient profile is selected as the edge index. The measurement noise is set inversely proportional with the maximum gradient. For each profile, the edge index is determined as:

$$s_i = argmax_k \left( \left| [I * G]_{i,k} - [I * G]_{i,k+1} \right| \right). \tag{5.9}$$

### 5.3.2 Step criterion edge detector (STEP)

STEP assumes that the intensity profile $I_i$ forms a transition from one intensity plateau to another. It calculates the heights of the two plateaus for each index value, and selects the index with the lowest sum of squared differences between the criteria and the image data. For each profile, the edge index is determined as:

$$s_i = argmin_k \sum_{t=0}^{k-1} \left( \left( \frac{1}{k} \sum_{j=0}^{k-1} I_{i,j} \right) - I_{i,t} \right)^2 + \sum_{t=k}^{K-1} \left( \left( \frac{1}{K-k} \sum_{j=k}^{K-1} I_{i,j} \right) - I_{i,t} \right)^2 . \quad (5.10)$$

If the plateau heights for the determined edge index are similar *($s_i = m$ and $\frac{1}{m} \sum_{j=1}^{m} I_{i,j} = \frac{1}{M-m} \sum_{j=m+1}^{M} I_{i,j}$)*, then the edge index is reset to the profile center by $s_i = \frac{M}{2}$. The measurement noise is defined inversely proportional with the height difference between the plateaus.

### 5.3.3 Max flow/min cut edge detector (MFMC)

The max flow/min cut algorithm can be used for finding the global optima of a set of energy functions including [18],

$$E(f) = \sum_{v \in V} D_v(f_v) + \sum_{(v,y) \in Edges} Q_{v,y}(f_v, f_y). \quad (5.11)$$

The optimization process seeks a labeling function $f$ that assigns binary values to the nodes that are defined under a set $V$, distinguishing inside of the LV cavity ($f = 1$) from the outside ($f = 0$). The classification is constrained by *data penalty* $D_v$, and *interaction potential* $Q_{v,y}$ functions. $D_v$ penalizes the labeling of $v$ based on the predefined likelihood function, whereas $Q_{v,y}$ penalizes the labeling discontinues between the neighboring nodes $v$ and $y$.

The problem of finding the optimal edges is formulated as Equation 5.11 in MFMC edge detector [11]. Initially, a graph with nodes representing the profile samples is created. The source and sink terminal nodes are appended to the node-set, and connected with the nodes corresponding to the first and the last samples of the intensity profiles respectively. These connections are called the *t-links* and have infinite weights. The nodes corresponding to (1) the consecutive samples of the same profile, and (2) the same index samples of the neighboring profiles, are connected by undirected edges called the *n-links*. The weight of an n-link connecting the nodes $v$ and $y$ is calculated as:

$$weight(v, y) = C \times \exp\left( \frac{-(I_v - I_y)^2}{2\sigma^2} \right), \quad (5.12)$$

where $I_v$ and $I_y$ refers to the intensity values at the associated profile samples and $C$ is a constant. After the graph is created, the maximum flow / minimum cut between the source and sink nodes are found using the push-relabel algorithm. The resulting cut defines the edge positions for all intensity profiles simultaneously. The reverse of the flow amount is utilized as the measurement noise in the Kalman filter.

### 5.3.4 Best linear unbiased edge estimator (BLUE)

The estimation accuracies of MG, STEP and MFMC vary depending on both (1) the endocardial surface position (spatial dependency), and (2) the cardiac cycle position (temporal dependency). As an example, MFMC might be the most successful method for the apical inferior region, while performing poor for the mid inferior region at ED. On the other hand, STEP might out perform the other methods for both the apical inferior and mid inferior regions at ES. The responses of multiple edge detectors can be combined using a statistical learning approach for generating better edge estimates; the confidences of the detectors for a given spatial and temporal position determine their weights.

For a given edge detector, estimated edge indices for a model can be represented as,

$$\mathbf{s}_\zeta = \left[s_{1,\zeta},\, s_{2,\zeta}, \ldots s_{N,\zeta}\right], \tag{5.13}$$

where $s_{i,\zeta}$ is the estimated edge index for the $i^{th}$ intensity profile, and $\zeta \in [0 : ES,\, 1 : ED]$ gives the cardiac cycle position. Using a similar notation, the correct edge indices for the model can be defined by,

$$\theta_\zeta = \left[\theta_{1,\zeta},\, \theta_{2,\zeta}, \ldots \theta_{N,\zeta}\right]. \tag{5.14}$$

Please note that $\theta_\zeta$ is an unknown vector variable that needs to be estimated. The error bias for the $i^{th}$ intensity profile at time $\zeta$ can be learned from a training dataset of size $B$ using,

$$Bias_{i,\zeta} = E\left[s_{i,\zeta} - \theta_{i,\zeta}\right] \cong \frac{1}{B}\sum_{b=1}^{B}\left(s_{i,\zeta}^{(b)} - \theta_{i,\zeta}^{(b)}\right). \tag{5.15}$$

For a given edge detector, the learned bias for the cardiac cycle position $\zeta$ can be written in a vector form as:

$$\mathbf{Bias}_\zeta = \left[Bias_{1,\zeta},\, Bias_{2,\zeta}, \ldots Bias_{N,\zeta}\right]. \tag{5.16}$$

The bias eliminated edge detector would estimate the edge indeces by $\tilde{\mathbf{s}}_\zeta = \mathbf{s}_\zeta - \mathbf{Bias}_\zeta$. The bias eliminated estimates of MG, STEP and MFMC methods for the $i^{th}$ intensity profile at $\zeta$ are given by,

$$\mathbf{y_{i}}_{,\zeta} = \left[\tilde{s}_{i,\zeta}^{MG},\, \tilde{s}_{i,\zeta}^{STEP},\, \tilde{s}_{i,\zeta}^{MFMC}\right]^T. \tag{5.17}$$

The probability distribution function (PDF) $p\left(\mathbf{y_{i}}_{,\zeta};\, \theta_{i,\zeta}\right)$ depends on the unknown parameter $\theta_{i,\zeta}$. BLUE restricts the estimator to be linear in the data as,

$$\hat{\theta}_{i,\zeta} = \sum_{n=1}^{3} a_n \mathbf{y_{i}}_{,\zeta}\left[n\right] = \mathbf{a^T}\mathbf{y_{i}}_{,\zeta}, \tag{5.18}$$

where the constant $a_n$ values are yet to be determined. One may consider $a_n$'s as the weights of the unbiased base estimators: $a_1$, $a_2$, $a_3$ give the weights of MG, STEP and

MFMC estimates for the $i^{th}$ intensity profile at time $\zeta$. Depending on the selected $a_n$ values, a large number of different estimators for $\theta_{i,\zeta}$ can be generated. However, BLUE is defined as the one with the no bias and minimum variance. As the class of estimators are restricted to be linear, BLUE will be optimal only when the minimum variance unbiased (MVU) estimator is linear.

The unbiased constrain of BLUE requires that,

$$E\left[\hat{\theta}_{i,\zeta}\right] = \sum_{n=1}^{3} a_n E\left[\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]\right] = \theta_{i,\zeta}. \tag{5.19}$$

In order the satisfy this constraint, $E\left[\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]\right]$ must be linear in $\theta_{i,\zeta}$:

$$E\left[\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]\right] = h\left[n\right]\theta_{i,\zeta}. \tag{5.20}$$

$\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]$ can also be written as,

$$\begin{aligned}\mathbf{y}_{\mathbf{i},\zeta}\left[n\right] &= E\left[\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]\right] + \left(\mathbf{y}_{\mathbf{i},\zeta}\left[n\right] - E\left[\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]\right]\right) && (5.21)\\ &= h\left[n\right]\theta_{i,\zeta} + w\left[n\right], && (5.22)\end{aligned}$$

where $w\left[n\right]$ is the error term, and $h\left[n\right] = 1$ as the bias eliminated estimators are employed in Equation 5.18. Equation 5.19 can be further expanded as,

$$\sum_{n=1}^{3} a_n E\left[\mathbf{y}_{\mathbf{i},\zeta}\left[n\right]\right] = \sum_{n=1}^{3} a_n h\left[n\right]\cdot\theta_{i,\zeta} = \theta_{i,\zeta}, \tag{5.23}$$

showing that,

$$\sum_{n=1}^{3} a_n h\left[n\right] = 1 \quad or \quad \mathbf{a^T 1} = 1, \tag{5.24}$$

where $\mathbf{1} = [1,\ 1,\ 1]^T$. Equation 5.24 shows that the sum of the base estimator weights should be 1. The variance of the $\hat{\theta}_{i,\zeta}$ can be computed as,

$$\begin{aligned} var\left(\hat{\theta}_{i,\zeta}\right) &= E\left[\left(\mathbf{a^T}\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{a^T}E\left[\mathbf{y}_{\mathbf{i},\zeta}\right]\right)^2\right],\\ &= E\left[\left(\mathbf{a^T}\left(\mathbf{y}_{\mathbf{i},\zeta} - E\left[\mathbf{y}_{\mathbf{i},\zeta}\right]\right)\right)^2\right] = \mathbf{a^T}C_{i,\zeta}\mathbf{a}. && (5.25)\end{aligned}$$

Due to the eliminated bias terms, (1) $E\left[\mathbf{y}_{\mathbf{i},\zeta}\right] = \theta_{i,\zeta}\mathbf{1}$, and (2) $C_{i,\zeta}$ is a $3\times 3$ symmetric matrix holding the error covariances between the base detectors (e.g. $C_{i,\zeta}\left(1,1\right)$ holds the error variance of MG detector's edge estimations, and $C_{i,\zeta}\left(2,3\right)$ holds the error covariance between STEP and MFMC detectors' edge estimations for the $i^{th}$ intensity profile at $\zeta$). Like the bias terms, error covariances can also be learned from a training dataset.

BLUE minimizes $var\left(\hat{\theta}_{i,\zeta}\right) = \mathbf{a^T}C_{i,\zeta}\mathbf{a}$ subject to the unbiased constraint $\mathbf{a^T 1} = 1$. The optimal $\mathbf{a_{opt}}$ vector is found as (see [19] for the derivation),

$$\mathbf{a_{opt}} = \frac{\mathbf{C_{i,\zeta}^{-1}1}}{\mathbf{1^T C_{i,\zeta}^{-1}1}}. \tag{5.26}$$

Therefore, BLUE for $\hat{\theta}_{i,\zeta}$ is,

$$\hat{\theta}_{i,\zeta}^{BLUE} = \frac{\mathbf{1}^T C_{i,\zeta}^{-1}}{\mathbf{1}^T C_{i,\zeta}^{-1}\mathbf{1}} \mathbf{y_{i,\zeta}}. \tag{5.27}$$

The variance of the proposed BLUE can be simply computed by,

$$var\left(\hat{\theta}_{i,\zeta}\right) = \mathbf{a_{opt}^T} C_{i,\zeta} \mathbf{a_{opt}} = \frac{1}{\mathbf{1^T} C_{i,\zeta}^{-1}\mathbf{1}}, \tag{5.28}$$

where the estimator variance can be utilized as the measurement noise in the Kalman filter.

It can also be shown that BLUE is identical to MLE for the investigated edge detection problem if the base estimator error distributions are assumed to be Gaussian (see Appendix A for the derivation). This is due to the facts that (1) BLUE produces the optimal estimator for the problems with a linear model and a Gaussian noise, and (2) MLE estimator for a linear model with a Gaussian noise is optimal. Therefore, the results of our study can be used for the derivation of a Bayes estimator for the endocardial edge detection in a future study.

## 5.4 Results

A set of 18 apical 3D echocardiography recordings were acquired using a Vivid 7 ultrasound scanner (GE Vingmed Ultrasound, Norway) and a matrix array transducer. The endocardial border segmentation of the recordings were performed by a medical expert using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway).

A 3-fold cross validation (CV) was applied for the validation of BLUE method. First, the recordings were randomly divided into 3 datasets, each including 6 recordings. Then, for each of these 3 datasets:

1. The other 2 datasets were used for the training: (a) bias, (b) covariance and (c) weight properties of the base detectors were estimated for the ED and ES frames. Figure 5.2 shows the estimated covariances between the base detectors, together with the corresponding detector weights from the first dataset.

2. The estimated bias and weight properties, which vary over the endocardial surface model, were interpolated between ED and ES through the cardiac cycle using linear interpolation. Figure 6.2 shows edge-detector weight interpolation from ED to ES for 4 random surface points for the first dataset.

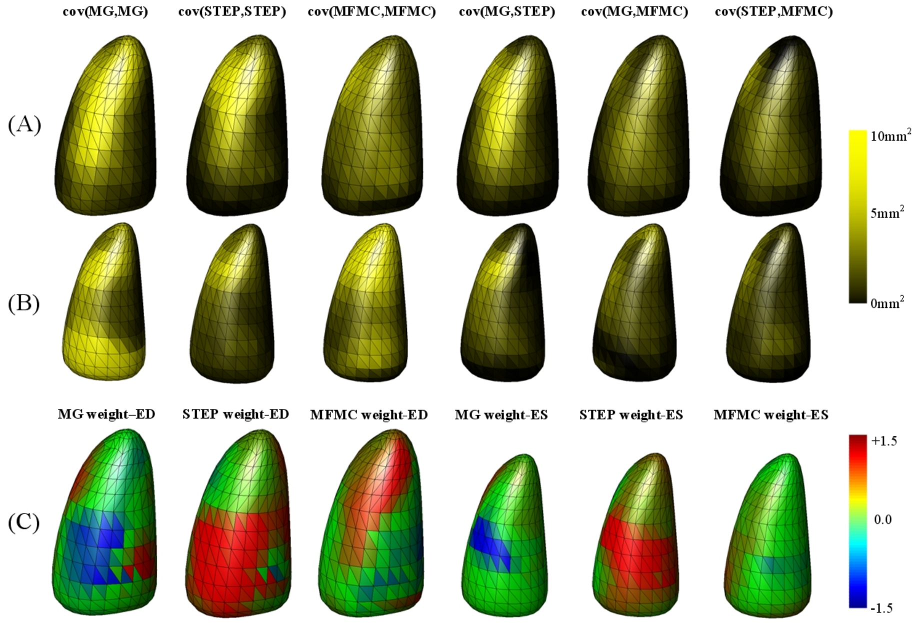


Figure 5.2: Covariances between the base estimators and the corresponding base estimator weights. Row (A) and (B) show covariances between the base estimators for a dataset at ED and ES frames respectively. Row(C) shows the weights of the base estimators with color coded images, where columns 1-3 show the weights for ED and columns 4-6 show the weights for ES.

3. The BLUE method with the learned bias and weight properties was employed in connection to the existing Kalman tracking framework for tracking the recordings in the testing dataset. The error measurements including the (a) *absolute surface point error* giving the average absolute distance of each predicted surface point to ground truth surface, (b) *squared surface point error* giving the average squared distance of each predicted surface point to ground truth surface, and (3) *absolute volume error* giving the average of predicted surface's absolute volume errors were computed.

Finally, the measurements of each fold were averaged to find the final statistics for BLUE as described in [20]. Similar error measurements were also performed for MG, STEP and MFMC using all 18 recordings. The final results for each method are reported in Figure 5.4.

In the Kalman tracking framework, a handcrafted Doo-Sabin endocardial model consisting of 20 control points was used as LV model [9]. Edge measurements were performed in 528 intensity profiles distributed evenly across the endocardial surface. Each profile consisted of 30 samples, spaced 1 *mm* apart. For MG and STEP, normal displacement measurements that were significantly different from their neighbors were

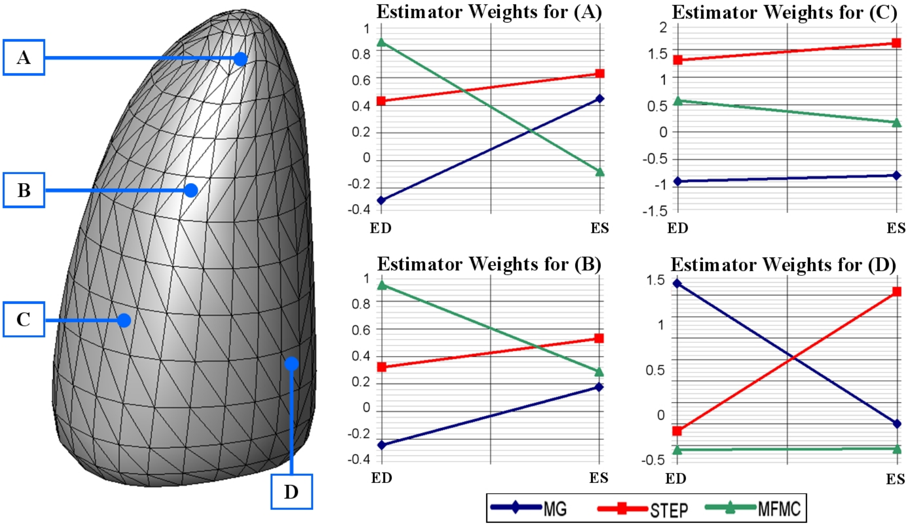


Figure 5.3: Base estimator weights for 4 random surface points from ED to ES. Base estimator weights vary over the endocardial surface and cardiac cycle as represented for 4 points. For a given endocardial surface and cardiac cycle position, sum of the weights is always 1, where the negative weights are allowed.

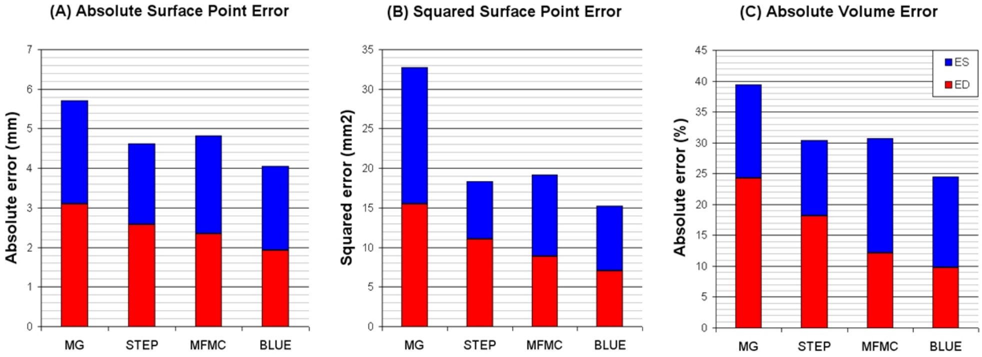


Figure 5.4: (A) Absolute surface point error (in $mm$) , (B) squared surface point error (in $mm^2$), and (C) absolute volume error (in percentages) for the Kalman tracking framework using MG, STEP, MFMC and BLUE methods. Red and blue bars represent the average error for the ED and ES frames respectively.

discarded as outliers. The tracking framework is implemented in C++, and processed each frame in 6.8ms with MG, 7.5ms with STEP, 78ms with MFMC and 81ms with BLUE when executed on a 2.80 GHz Intel Core 2 Duo CPU.

## 5.5 Discussion and conclusion

We have introduced BLUE for the endocardial edge detection, which leads to identical solutions as MLE under the assumption of Gaussian distributed base estimator errors. Introduced novel approach utilizes statistical inferences depending on both the endocardial surface and cardiac cycle positions to combine multiple edge detectors. Comparative analyses have showed that the proposed method outperforms the MG, STEP and MFMC methods both in surface and volumetric measurement accuracies (see Figure 5.4).

Our study significantly differs from [16], as (1) the statistical properties of the base estimators are defined for each endocardial surface point independently, and (2) the base estimator covariances are utilized. The *probability-variance estimator* represented in [16] estimates the endocardial edge positions as,

$$\hat{\theta}_{i,\zeta}^{Mrg\,MLE} = \frac{\frac{\tilde{s}_{i,\zeta}^{MG}}{\sigma^2_{\zeta,(MG)}} + \frac{\tilde{s}_{i,\zeta}^{STEP}}{\sigma^2_{\zeta,(STEP)}} + \frac{\tilde{s}_{i,\zeta}^{MFMC}}{\sigma^2_{\zeta,(MFMC)}}}{\frac{1}{\sigma^2_{\zeta,(MG)}} + \frac{1}{\sigma^2_{\zeta,(STEP)}} + \frac{1}{\sigma^2_{\zeta,(MFMC)}}}, \tag{5.29}$$

where the bias and variance ($\sigma^2_\zeta$) properties of the base predictors do *not* depend on a cardiac surface position. Equations 5.27 and 5.29 clearly show that the $\hat{\theta}^{Mrg\,MLE}$ could be considered as the surface marginalized version of the proposed $\hat{\theta}_{i,\zeta}^{BLUE}$ (or $\hat{\theta}_{i,\zeta}^{MLE}$) only if the base estimators were *not* correlated. As represented in Figure 5.2, the covariances between the base estimators get high values depending on both the cardiac surface and cardiac cycle positions. Hence, the zero correlation assumption in [16] would not be valid for the endocardial surface position dependent statistical inferences utilized in this study.

The proposed approach can be further improved in the following areas:

1. The introduced method does not make any assumptions about the prior distribution of $\theta_{\mathbf{i},\zeta}$, as the estimations are directly derived from $p\left(\mathbf{y}_{\mathbf{i},\zeta};\, \theta_{i,\zeta}\right)$. If the prior distribution of the $\theta_{i,\zeta}$ were learned from a training dataset, then the empirical Bayes method could have been employed. On the average, empirical Bayes estimations are always closer to $\theta_{i,\zeta}$ than MLE's [21]. The application of the empirical Bayes method for the endocardial edge detection problem might be investigated in a future study.

2. The proposed method calculates the base estimator bias and weight properties at the ED and ES frames. Then, it interpolates these properties to intermediate frames using a linear interpolation. However, these properties could be learned at more than 2 cardiac cycle positions, and *trigonometrically* interpolated to all

cardiac cycle. The trigonometric interpolation would then take advantage of the cyclic motion of the endocardial walls.

# A. Maximum likelihood edge estimator (MLE)

The system can be described for the $i^{th}$ intensity profile at time $\zeta$ using a general linear model (mostly similar with the Equation 5.22) as:

$$\mathbf{s}_{\mathbf{i},\zeta} = \mathbf{1}\theta_{\mathbf{i},\zeta} + \mathbf{b}_{\mathbf{i},\zeta} + \mathbf{w}_{\mathbf{i},\zeta}, \tag{5.30}$$

where $\theta_{\mathbf{i},\zeta}$ is the unknown edge position (scalar valued), $\mathbf{s}_{\mathbf{i},\zeta}$ is the measurement vector of the edge detectors, $\mathbf{b}_{\mathbf{i},\zeta} = \left[Bias^{MG}_{i,\zeta},\ Bias^{STEP}_{i,\zeta},\ Bias^{MFMC}_{i,\zeta}\right]^T$ is the learned bias vector, and $\mathbf{w}_{\mathbf{i},\zeta} = \left[w^{MG}_{ij},\ w^{STEP}_{ij},\ w^{MFMC}_{ij}\right]^T$ is the noise vector with a learned Gaussian PDF $N\left(0, C_{i,\zeta}\right)$. The bias vector can be merged with the measurement vector as in Equation 5.17 giving,

$$\mathbf{y}_{\mathbf{i},\zeta} = \mathbf{1}\theta_{\mathbf{i},\zeta} + \mathbf{w}_{i,\zeta}. \tag{5.31}$$

Under these conditions $p\left(\mathbf{y}_{\mathbf{i},\zeta};\, \theta_{i,\zeta}\right)$ can be defined by,

$$p\left(\mathbf{y}_{\mathbf{i},\zeta};\, \theta_{i,\zeta}\right) = K \cdot e^{-\frac{1}{2}\left(\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{1}\theta_{\mathbf{i},\zeta}\right)^T C^{-1}_{i,\zeta}\left(\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{1}\theta_{\mathbf{i},\zeta}\right)}, \tag{5.32}$$

where $K$ normalizes the PDF. MLE of $\theta_{i,\zeta}$ is found by minimizing,

$$J\left(\theta_{i,\zeta}\right) = \left(\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{1}\theta_{\mathbf{i},\zeta}\right)^T C^{-1}_{i,\zeta}\left(\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{1}\theta_{\mathbf{i},\zeta}\right). \tag{5.33}$$

Since (1) Equation 5.33 is a quadratic function of the elements of $\theta_{\mathbf{i},\zeta}$, and (2) $C^{-1}_{i,\zeta}$ is a positive definite matrix, differentiation of $J\left(\theta_{i,\zeta}\right)$ produces the global minimum. Differentiation of the natural logarithm of $p\left(\mathbf{y}_{\mathbf{i},\zeta};\, \theta_{i,\zeta}\right)$ gives (see [19] for the derivation),

$$\frac{\partial \ln p\left(\mathbf{y}_{\mathbf{i},\zeta};\, \theta_{i,\zeta}\right)}{\partial \theta_{i,\zeta}} = \frac{\partial\left(\mathbf{1}\theta_{\mathbf{i},\zeta}\right)^T}{\partial \theta_{i,\zeta}} C^{-1}_{i,\zeta}\left(\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{1}\theta_{\mathbf{i},\zeta}\right). \tag{5.34}$$

Setting this gradient equal to zero yields,

$$\mathbf{1}^T C^{-1}_{i,\zeta}\left(\mathbf{y}_{\mathbf{i},\zeta} - \mathbf{1}\hat{\theta}_{\mathbf{i},\zeta}\right) = 0 \tag{5.35}$$

Solving Equation 5.35 for $\hat{\theta}_{i,\zeta}$ produces MLE:

$$\hat{\theta}^{MLE}_{i,\zeta} = \frac{\mathbf{1}^T C^{-1}_{i,\zeta}}{\mathbf{1}^T C^{-1}_{i,\zeta}\mathbf{1}}\mathbf{y}_{\mathbf{i},\zeta}. \tag{5.36}$$

$\hat{\theta}_{i,\zeta}$ is an unbiased estimator with the variance of $var\left(\hat{\theta}_{i,\zeta}\right) = \left(\mathbf{1^T} C^{-1}_{i,\zeta}\mathbf{1}\right)^{-1}$.

As represented in Equations 5.27 and 5.36, BLUE and MLE estimators are identical for the application defined in this paper.

# References

[1] N. Lin, W. Yu, and J. S. Duncan, "Combinative multi-scale level set framework for echocardiographic image segmentation," *Medical Image Analysis*, vol. 7, pp. 529–537, 2002.

[2] R. Juang, E. McVeigh, B. Hoffmann, D. Yuh, and P. Burlina, "Automatic segmentation of the left-ventricular cavity and atrium in 3d ultrasound using graph cuts and the radial symmetry transform," in *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pp. 606 –609, 30 2011-april 2 2011.

[3] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu, "3d ultrasound tracking of the left ventricles using one-step forward prediction and data fusion of collaborative trackers," in *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[4] S. K. Setarehdan and J. J. Soraghan, *Segmentation in echocardiographic images*, pp. 64–130. New York, NY, USA: Springer-Verlag New York, Inc., 2002.

[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.

[6] G. Jacob, J. A. 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, 1999.

[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, 2001.

[8] S. Venkatesh and R. Owens, "On the classification of image features," *Pattern Recogn. Lett.*, vol. 11, no. 5, pp. 339–349, 1990.

[9] F. Orderud and S. I. Rabben, "Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces," in *CVPR*, 2008.

[10] S. I. Rabben, A. H. Torp, A. Støylen, S. Slørdahl, K. Bjørnstad, B. O. Haugen, and B. Angelsen, "Semiautomatic contour detection in ultrasound m-mode images," *Ultrasound in Medicine & Biology*, vol. 26, no. 2, pp. 287 – 296, 2000.

[11] E. Dikici and F. Orderud, "Graph-cut based edge detection for kalman filter based left ventricle tracking in 3d+ t echocardiography," in *Proceedings of Computing in Cardiology*, 2010.

[12] C.-H. Lee, R. Greiner, and S. Wang, "Using query-specific variance estimates to combine bayesian classifiers," in *Proceedings of the 23rd international conference on Machine learning*, ICML '06, (New York, NY, USA), pp. 529–536, ACM, 2006.

[13] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[14] S. Konishi, A. Yuille, and J. Coughlan, "A statistical approach to multi-scale edge detection," in *Proc. Workshop Generative-Model-Based Vision: GMBV 2002*, 2002.

[15] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: Learning and evaluating edge cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 57–74, 2003.

[16] E. Dikici and F. Orderud, "Maximum likelihood and james-stein edge estimators for left ventricle tracking in 3d echocardiography," in *2nd International Workshop on Machine Learning in Medical Imaging (MLMI) in conjunction with MICCAI*, 2011.

[17] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978.

[18] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 2, pp. 271–279, 1989.

[19] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer series in statistics, Springer, 2009.

[21] G. Casella, "An Introduction to Empirical Bayes Data Analysis," *The American Statistician*, vol. 39, no. 2, pp. 83–87, 1985.

# Chapter 6

# Empirical Bayes Estimator for Endocardial Edge Detection in 3D+T Echocardiography

Engin Dikici[1], Fredrik Orderud[2] and Bo Henry Lindqvist[1]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

This paper presents an empirical Bayes (EB) estimator for detection of endocardial edges in 3D+T echocardiography recordings. A maximum likelihood (ML) edge detector, proposed in a previous study, combines the responses of multiple edge detectors to improve the detection accuracy. We aim to further extend this approach with the use of contextual priors, that gives the probabilistic distribution of correct (yet unknown) endocardial edge positions. For training, a ML model that gives an optimal linear combination of multiple endocardial edge detectors is learned from a pre-segmented dataset. For a given test data, (1) ML edges are estimated using the learned ML model, (2) a contextual prior is derived using the ML edge estimations in an empirical fashion, and (3) ML estimates and the contextual prior are fused to produce empirical Bayes endocardial edge estimates. Comparative analyses show that EB reduces the mean square endocardial surface error with respect to ML estimations. This is due to the Stein effect that briefly asserts that the expected mean square error of the ML estimations should be reduced with the use of empirically-derived prior information.

## 6.1 Introduction

This paper considers the problem of endocardial border detection in 3D+T echocardigraphy recordings. This is a challenging task due to problems including (1) speckle noise, (2) shadowing that can result in missing boundaries, and (3) the existence of intra-cavity structures such as chordae tendineae, papillary muscles and valves [1]. Furthermore, real time, or nearly real time, detection of endocardial borders might be necessary during invasive procedures and intensive care.

A Kalman tracking framework can be employed for rapid and model based detection of endocardial borders in 3D+T recordings. The approach utilizes a sequential prediction and update strategy, where surface deformations are first predicted by a kinematic model, followed by an update step based on information provided by image measurements. Maximum gradient (MG) [2], step criterion (STEP) [3], local-phase [4] and max flow/min cut (MFMC) [5] edge detectors were previously employed for the image measurement stage of the Kalman tracking framework. In [6], a maximum likelihood (ML) edge detector that combines the responses of multiple base detectors (e.g. MG, STEP and MFMC) via learned statistical inferences was introduced. The method was shown to be more accurate than the base detectors, however it had 2 major limitations: (1) the edge detectors were assumed to be uncorrelated, and (2) statistical properties of the edge detectors were marginalized to the endocardial surface. More recently, the shortcomings of [6] were addressed by an improved ML method employing both cardiac cycle and endocardial surface position dependent statistics [7].

In this paper, we further extend the ML endocardial edge detector, proposed in [7], by incorporating contextual priors giving the probabilistic distribution of endocardial edges around a surface model. A pre-segmented training dataset is used for learning a ML model, giving the optimal base detector weights for each cardiac cycle and endocardial surface position. For testing, ML edges are first estimated using the ML model. Next, the ML estimates are utilized to empirically compute Gaussian contextual prior parameters. Finally, the ML estimates are combined with the contextual prior to produce empirical Bayes (EB) estimations for the endocardial edge positions (see Figure 6.1). The effectiveness of the introduced method is represented via comparative analyses among MG, STEP, MFMC, ML and EB.

## 6.2 Methods

### 6.2.1 Kalman Tracking Framework

The tracking framework is built around a deformable subdivision model parametrized by a set of control vertices with associated displacement direction vectors. Model deformations are handled by a composite transform, where local shape deformations are obtained by moving control vertices in the subdivision model together with a global transformation that translates, rotates and scales the model as a whole. A manually constructed Doo-Sabin surface is used as a template for representing the endocardial borders, where the model control vertices are allowed to move in the surface normal direction to alter the shape.

The tracking framework consists of five separate stages, namely (1) state prediction, (2) evaluation of tracking model, (3) edge measurements, (4) measurement assimilation, and (5) measurement update [8].

Endocardial edge detection methods are employed during the *edge measurements* stage of the framework. The edge detection is performed by first extracting $N$ 1D

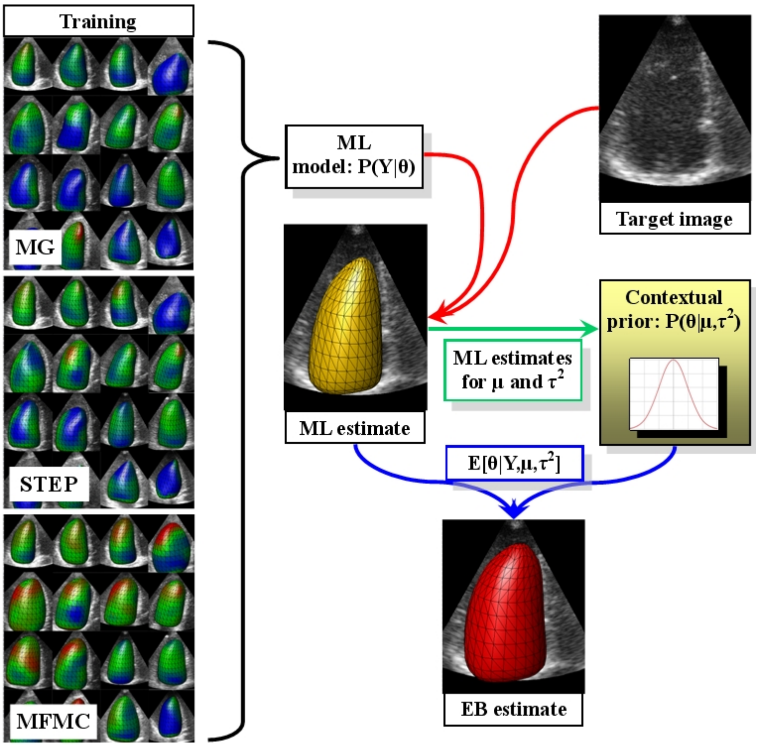


Figure 6.1: Overview of the proposed approach. Training stage produces a ML model. Next, the ML model is used for finding a ML estimate that is also utilized to derive a contextual prior. Finally, the ML estimate and the contextual model are combined to produce an EB estimate.

intensity profiles centered around the surface points, which are spread evenly across the endocardial surface model. The intensity profiles are oriented in the surface normal directions. MG, STEP and MFMC edge detectors, also referred as the base detectors, define different intensity profile processing schemes to estimate endocardial edge positions [7]. ML and EB edge detectors, combining the responses of these base detectors, are described in the following subsections.

### 6.2.2 Maximum Likelihood Edge Detector (ML)

The estimation accuracies of MG, STEP and MFMC vary depending on both endocardial surface and cardiac cycle positions [5, 7]. As an example, MFMC might be the most successful method for the apical inferior region, while performing poor for the mid inferior region at end diastole (ED). On the other hand, STEP might outperform the other methods for both the apical inferior and mid inferior regions at end systole (ES). The responses of multiple edge detectors can be combined using a statistical learning approach for generating better edge estimates; the confidences of the detectors for a given spatial and temporal position determine their weights.

The system can be described for the $i^{th}$ intensity profile at cardiac cycle position $\zeta \in [0 : ES,\, 1 : ED]$ using a general linear model as:

$$\mathbf{s}_{\mathbf{i},\zeta} = \mathbf{1}\theta_{\mathbf{i},\zeta} + \mathbf{b}_{\mathbf{i},\zeta} + \mathbf{w}_{\mathbf{i},\zeta}, \tag{6.1}$$

where $\theta_{\mathbf{i},\zeta}$ is the unknown edge position (scalar valued), $\mathbf{s}_{\mathbf{i},\zeta}$ is the measurement vector of the edge detectors, $\mathbf{b}_{\mathbf{i},\zeta} = \left[Bias_{i,\zeta}^{MG},\, Bias_{i,\zeta}^{STEP},\, Bias_{i,\zeta}^{MFMC}\right]^T$ is the learned bias vector for the $i^{th}$ intensity profile at time $\zeta$, and $\mathbf{w}_{\mathbf{i},\zeta} = \left[w_{ij}^{MG},\, w_{ij}^{STEP},\, w_{ij}^{MFMC}\right]^T$ is the noise vector with a learned Gaussian probability distribution function (PDF) $N\left(0,\, C_{i,\zeta}\right)$. Note that $C_{i,\zeta}$ is a $3 \times 3$ symmetric matrix holding the error covariances between the base detectors (e.g. $C_{i,\zeta}\left(1,1\right)$ holds the error variance of MG detector's edge estimations, and $C_{i,\zeta}\left(2,3\right)$ holds the error covariance between STEP and MFMC detectors' edge estimations for the $i^{th}$ intensity profile at $\zeta$). The bias vector can be merged with the measurement vector giving,

$$\mathbf{y}_{\mathbf{i},\zeta} = \mathbf{1}\theta_{\mathbf{i},\zeta} + \mathbf{w}_{i,\zeta}. \tag{6.2}$$

Maximum likelihood estimator of $\theta_{i,\zeta}$ can be derived as [7]:

$$\hat{\theta}_{i,\zeta}^{ML} = \frac{\mathbf{1}^T C_{i,\zeta}^{-1}}{\mathbf{1}^T C_{i,\zeta}^{-1}\mathbf{1}} \mathbf{y}_{\mathbf{i},\zeta}. \tag{6.3}$$

$\hat{\theta}_{i,\zeta}^{ML}$ is an unbiased estimator with the variance of $\sigma_{i,\zeta}^2 = \left(\mathbf{1^T} C_{i,\zeta}^{-1}\mathbf{1}\right)^{-1}$. The estimator variance can be utilized as the measurement noise in the Kalman filter. ML endocardial edge estimations for the model at $\zeta$ are given as,

$$\hat{\theta}_{i,\zeta}^{ML} \sim N\left(\theta_{\mathbf{i},\zeta},\, \sigma_{i,\zeta}^2\right), \qquad i = 1, \ldots, N. \tag{6.4}$$

### 6.2.3 Empirical Bayes Edge Detector (EB)

$\hat{\theta}_{i,\zeta}^{ML}$ is both the best linear unbiased and maximum likelihood estimator for $\theta_{i,\zeta}$, which might seem enough to declare $\hat{\theta}_{i,\zeta}^{ML}$ as an optimal detector for a given base detector set. However, if all the base detectors are producing highly correlated and wrong results for some parts of the endocardium, it becomes obvious that a better estimator should benefit from an additional bias term occasionally. Empirical Bayes estimator (1) uses a contextual prior model giving the PDF of the endocardial edge displacements, (2) and refines the ML estimations with this prior model to produce biased estimations. The prior model helps pulling low-confidence-inaccurate endocardial edge estimations towards a prior distribution mean, which eventually lowers the estimation mean square error.

Let us make the Bayesian assumption that the contextual prior has a Gaussian PDF given as,

$$\theta_{\mathbf{i},\zeta} \sim N\left(\mu_\zeta, \tau_\zeta^2\right), \qquad i = 1, \ldots, N. \tag{6.5}$$

The Bayes estimate for $\theta_{\mathbf{i},\zeta}$, which is the expected value of the posterior distribution $p\left(\theta_{i,\zeta} | \, \hat{\theta}_{1,\zeta}^{ML} \ldots \hat{\theta}_{N,\zeta}^{ML}\right)$, is given by,

$$\hat{\theta}_{i,\zeta}^{Bayes} = \left(\frac{\sigma_{i,\zeta}^2}{\sigma_{i,\zeta}^2 + \tau_\zeta^2}\right)\mu_\zeta + \left(\frac{\tau_\zeta^2}{\sigma_{i,\zeta}^2 + \tau_\zeta^2}\right)\hat{\theta}_{i,\zeta}^{ML}. \tag{6.6}$$

$\hat{\theta}_{i,\zeta}^{Bayes}$is the weighted average of $\mu_\zeta$ and $\hat{\theta}_{i,\zeta}^{ML}$; the weights used in the weighted average depend on the relative sizes of $\sigma_{i,\zeta}^2$ and $\tau_\zeta^2$. Hence the confidences of $\hat{\theta}_{i,\zeta}^{ML}$ and $\mu_\zeta$ determine their weights. More explicitly, the ML estimations are pulled towards a contextual prior mean $\mu_\zeta$ with rates determined by their confidences.

The empirical Bayes approach agrees with the Bayes model, and estimates the unknown prior distribution parameters, $\mu_\zeta$ and $\tau_\zeta^2$, from the data [9]. For this estimation task, it uses the marginal distributions of $\hat{\theta}_{i,\zeta}^{ML}$, represented as $m\left(\hat{\theta}_{i,\zeta}^{ML}\right)$ . A standard calculation shows that,

$$m\left(\hat{\theta}_{i,\zeta}^{ML}\right) \sim N\left(\mu_\zeta, \sigma_{i,\zeta}^2 + \tau_\zeta^2\right), \qquad i = 1, \ldots, N. \tag{6.7}$$

The joint marginal distribution of all $\hat{\theta}_{i,\zeta}^{ML}$ is hence given by,

$$m\left(\hat{\theta}_{\zeta}^{ML}\right) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\left(\sigma_{i,\zeta}^2 + \tau_\zeta^2\right)}} e^{\frac{-\left(\hat{\theta}_{i,\zeta}^{ML} - \mu_\zeta\right)^2}{2\left(\sigma_{i,\zeta}^2 + \tau_\zeta^2\right)}}. \tag{6.8}$$

Maximum likelihood estimators for $\mu_\zeta$ and $\tau_\zeta^2$ can be found by maximizing the log of this function, solving,

$$\frac{\partial \log\left(m\left(\hat{\theta}_{\zeta}^{ML}\right)\right)}{\partial \mu_\zeta} = \frac{\partial \log\left(m\left(\hat{\theta}_{\zeta}^{ML}\right)\right)}{\partial \tau_\zeta^2} = 0, \tag{6.9}$$

to give the estimates $\hat{\mu}_\zeta$ and $\hat{\tau}_\zeta^2$. Note that $\hat{\mu}_\zeta$ and $\hat{\tau}_\zeta^2$ are asymptotically efficient estimations for $\mu_\zeta$ and $\tau_\zeta^2$ [10]. The Empirical Bayes estimate is found by placing these estimated prior distribution parameters into Equation 6.6 by,

$$\hat{\theta}_{i,\zeta}^{EB} = \left(\frac{\sigma_{i,\zeta}^2}{\sigma_{i,\zeta}^2 + \hat{\tau}_\zeta^2}\right)\hat{\mu}_\zeta + \left(\frac{\hat{\tau}_\zeta^2}{\sigma_{i,\zeta}^2 + \hat{\tau}_\zeta^2}\right)\hat{\theta}_{i,\zeta}^{ML}. \tag{6.10}$$

$\hat{\theta}_{i,\zeta}^{EB}$ uses all $\hat{\theta}_{i,\zeta}^{ML}$ for detecting the endocardial edges taking the advantage of the *Stein effect*. The Stein effect briefly asserts that the estimation accuracy can be improved by using the information coming from the full model. It can be shown that for $N \geq 4$,

| | MG | STEP | MFMC | ML | EB |
|---|---|---|---|---|---|
| ED | 15.53 | 11.05 | 8.90 | 7.00 | 6.11 |
| ES | 17.22 | 7.27 | 10.31 | 8.21 | 7.44 |

Table 6.1: Squared surface point errors (in $mm^2$) for the Kalman tracking framework using MG, STEP, MFMC, ML, EB methods for ED and ES frames.

$$E\left[\sum_{i=1}^{N}\left(\theta_{i,\zeta}-\hat{\theta}_{i,\zeta}^{EB}\right)^2\right] < E\left[\sum_{i=1}^{N}\left(\theta_{i,\zeta}-\hat{\theta}_{i,\zeta}^{ML}\right)^2\right], \tag{6.11}$$

where the expectation is over the distribution of $\hat{\theta}_{i,\zeta}^{ML}$ given $\theta_{i,\zeta}$ [9]. Therefore, a lower mean square surface error is expected for the empirical Bayes estimator compared with the maximum likelihood estimator theoretically.

## 6.3 Results

18 apical 3D echocardiography recordings were acquired using a Vivid 7 ultrasound scanner (GE Vingmed Ultrasound, Norway) and a matrix array transducer. The endocardial border segmentation of the recordings were performed by a medical expert using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway).

A 3-fold cross validation (CV) was applied for validation of the ML and EB edge detectors. For each testing fold, the other 2 folds were used for learning a ML model at the ED and ES frames. The ML model included bias, covariance and corresponding ML weights at 528 evenly distributed endocardial surface positions. The learned model was later interpolated between ED and ES through the cardiac cycle using linear interpolation (see Figure 6.2). For testing, (1) ML and EB detectors were used in connection to the Kalman tracking framework with the learned ML model, (2) *squared surface point errors (SSPE)* giving the average squared distances between the estimated and ground truth surface points were computed, and (3) SSPE measurements of the folds were averaged to find final statistics. SSPE based evaluation was performed, since the objective of the study is to lower the mean square estimation error by taking advantage of the Stein effect. SSPE of STEP, MFMC and MG methods were also computed using all 18 recordings (see Table 6.1).

In the Kalman tracking framework, a handcrafted Doo-Sabin endocardial model consisting of 20 control points was used as LV model [8]. Edge measurements were performed in 528 intensity profiles distributed evenly across the endocardial surface. Each profile consisted of 30 samples, spaced 1 $mm$ apart. The tracking framework is implemented in C++, and processed each frame in 6.8ms with MG, 7.5ms with STEP, 78ms with MFMC, 81ms with ML and 83ms with EB when executed on a 2.80 GHz Intel Core 2 Duo CPU.

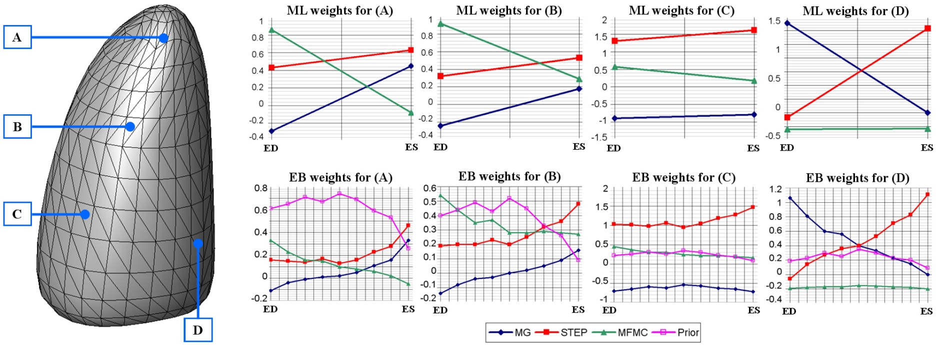


Figure 6.2: Base estimator and contextual prior weights for four random surface points for the ML and EB detectors. Upper row: ML computes the optimal base detector weights for ED and ES frames, and interpolate these to all cardiac cycle. Lower row: EB multiples the ML weights with $\left(\hat{\tau}_\zeta^2 / \left(\sigma_{i,\zeta}^2 + \hat{\tau}_\zeta^2\right)\right)$ and sets the prior distribution weight as $\left(\sigma_{i,\zeta}^2 / \left(\sigma_{i,\zeta}^2 + \hat{\tau}_\zeta^2\right)\right)$ (see Equation 6.10). For a given endocardial surface and cardiac cycle position, sum of the weights is always 1.

## 6.4 Conclusion

We have introduced the EB approach, which finds a contextual prior capturing the probabilistic distribution of endocardial edges around a surface model, and refines ML estimations with this empirically estimated prior. The introduced method is (1) simple and efficient; computation of contextual priors is an intuitive and computationally non-expensive task, (2) yet novel as the empirical Bayes approach has not been used for the Kalman measurements prior to our study. Results on 18 datasets demonstrate that the use of contextual information can lower the mean square estimation error as theoretically asserted by the Stein effect.

# References

[1] S. K. Setarehdan and J. J. Soraghan, *Segmentation in echocardiographic images*, pp. 64–130. New York, NY, USA: Springer-Verlag New York, Inc., 2002.

[2] 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.

[3] S. I. Rabben, A. H. Torp, A. Støylen, S. Slørdahl, K. Bjørnstad, B. O. Haugen, and B. Angelsen, "Semiautomatic contour detection in ultrasound m-mode images," *Ultrasound in Medicine & Biology*, vol. 26, no. 2, pp. 287 – 296, 2000.

[4] S. Venkatesh and R. Owens, "On the classification of image features," *Pattern Recogn. Lett.*, vol. 11, no. 5, pp. 339–349, 1990.

[5] E. Dikici and F. Orderud, "Graph-cut based edge detection for Kalman filter based left ventricle tracking in 3d+ t echocardiography," in *Proceedings of Computing in Cardiology*, 2010.

[6] E. Dikici and F. Orderud, "Maximum likelihood and James-Stein edge estimators for left ventricle tracking in 3d echocardiography," in *2nd International Workshop on Machine Learning in Medical Imaging (MLMI) in conjunction with MICCAI*, 2011.

[7] E. Dikici, F. Orderud, and H. Torp, "Best linear unbiased estimator for Kalman filter based left ventricle tracking in 3d+t echocardiography," *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*, 2012.

[8] F. Orderud and S. I. Rabben, "Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces," in *CVPR*, 2008.

[9] G. Casella, "An introduction to empirical Bayes data analysis," *The American Statistician*, vol. 39, no. 2, pp. 83–87, 1985.

[10] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

# Chapter 7

# Generalized Step Criterion Edge Detectors for Kalman Filter Based Left Ventricle Tracking in 3D+T Echocardiography

Engin Dikici[1], Fredrik Orderud[2]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

Step criterion edge detector (STEP) has been employed for the detection of endocardial edges in a Kalman filter based left ventricle tracking framework in previous studies. STEP determines the endocardial edge positions by fitting piecewise constant functions to intensity profiles, which are extracted on a tracked surface's normal directions. In this study, we generalize STEP using higher order piecewise polynomial functions. The generalized STEP detectors make different assumptions about the endocardial edge representations, and their accuracies vary over the endocardial surface and cardiac cycle positions. Accordingly, we combine the responses of the generalized detectors using a maximum likelihood (ML) approach. Unlike previously proposed ML approaches, our combined edge detector provides a real-time tracking solution as the majority of regressive functions for the polynomial fitting can be computed offline. Comparative analyses showed that the combined detector (1) outperforms each of the generalized STEP detectors, and (2) provides a comparable accuracy with the previously defined slower ML approach.

## 7.1 Introduction

3D+T echocardiography is a valuable tool for assessing cardiac function, as it enables real-time, non-invasive and low cost acquisition of volumetric images of the heart. The automated analysis methods for the echocardiography recordings have received considerable attention over the recent years [1, 2]. However, the automated segmentation and tracking of heart chambers remain challenging tasks due to imaging artifacts; including speckle noise, shadows and signal dropouts [3]. Furthermore, the *real-time* detection of endocardial borders might be desirable for the invasive procedures and intensive care applications.

State-space analysis using Kalman filtering can be utilized for the tracking of heart chambers in time dependent recordings. The approach uses a sequential prediction and update strategy, where surface deformations are first predicted by a kinematic model, followed by an update step based on information provided by image measurements. Maximum gradient (MG) [4], step criterion (STEP) [5], local-phase [6] and max flow/min cut (MFMC) [7] edge detectors were previously employed for the detection of the left ventricle's (LV) endocardial edges in a Kalman tracking framework. In [8], a maximum likelihood (ML) edge detector combining the responses of multiple base detectors via learned statistical inferences was introduced. The ML detector was shown to be more accurate than the base detectors, however it offered a slower tracking solution due to the high computational complexity of a utilized base detector, MFMC.

The motivation for our study is to define an accurate and real-time ML endocardial edge detector. Accordingly, we first define a new set of base detectors that generalize STEP to higher order piecewise polynomials. The $k^{th}$ order STEP detector (1) fits multiple piecewise $k^{th}$ order polynomial functions to a given intensity profile, which is extracted on a tracked surface's normal direction, then (2) selects the optimal piecewise function in the least-squares (LS) sense. Each generalized STEP detector makes different assumptions about the intensity distribution characteristics of the myocardium and blood-pool; the accuracies of these detectors vary depending on the endocardial surface and cardiac cycle positions. Accordingly, we combine the responses of the generalized STEP detectors utilizing a space-time position dependent ML method described in [8]. The combined detector provides a real-time tracking solution as the majority of regressive functions for the polynomial fitting can be computed offline. The effectiveness of the introduced method is represented via comparative analyses among the $0^{th}$, $1^{st}$, $2^{nd}$ order generalized STEP detectors, proposed ML and previously defined ML approaches.

## 7.2 Tracking Framework

The tracking framework is built around a deformable subdivision model parametrized by a set of control vertices with associated displacement direction vectors. Model deformations are handled by a composite transform, where local shape deformations are obtained by moving control vertices in the subdivision model together with a global transformation that translates, rotates and scales the model as a whole. A manually constructed Doo-Sabin surface is used as a template for representing the endocardial borders, where the model control vertices are allowed to move in the surface normal direction to alter the shape.

The tracking framework consists of five separate stages, namely (1) state prediction, (2) evaluation of tracking model, (3) edge measurements, (4) measurement assimilation, and (5) measurement update [9].

Endocardial edge detection methods are employed during the *edge measurements* stage of the framework. First, $N$ 1D intensity profiles *(*$I_1, I_2, \ldots I_N$*)* are extracted, where each profile is centered by an endocardial surface point and oriented in a surface normal direction. $I_{i,k}$ is used for referring to the intensity value of the $i^{\text{th}}$ profile's $k^{\text{th}}$ sample, and $M$ gives the total number of samples in each profile. Next, an edge detection method is employed for estimating the endocardial edge positions by processing the intensity profiles. Generalized STEP detectors, and a ML detector combining the responses of multiple generalized detectors are described in the following subsections.

### 7.2.1 Generalized Step Criterion Edge Detectors

The classical step criterion edge detector assumes that the intensity profile $I_i$ forms a transition from one intensity plateau to another. It calculates the heights of the two plateaus for each index value, and selects the index with the lowest sum of squared differences between the criteria and the image data. For each profile, the edge index is estimated as:

$$\hat{\theta}_i^{STEP} = argmin_\alpha \sum_{t=1}^{\alpha}\left(\left(\frac{1}{\alpha}\sum_{j=1}^{\alpha} I_{i,j}\right) - I_{i,t}\right)^2 + \sum_{t=\alpha+1}^{M}\left(\left(\frac{1}{M-\alpha}\sum_{j=\alpha+1}^{M} I_{i,j}\right) - I_{i,t}\right)^2. \tag{7.1}$$

If the plateau heights for the determined edge index are similar *(*$\hat{\theta}_i^{STEP} = \alpha$ *and* $\frac{1}{\alpha}\sum_{j=1}^{\alpha} I_{i,j} = \frac{1}{M-\alpha}\sum_{j=\alpha+1}^{M} I_{i,j}$*)*, then the edge index is reset to the profile center by $\hat{\theta}_i^{STEP} = \frac{M}{2}$.

The minimization task in Equation 7.1 can alternatively be represented as,

$$\hat{\theta}_i^{STEP} = argmin_{\alpha \in \{1,2 \ldots M\}}\left(\sum_{j=1}^{M}\left[I_{i,j} - f_\alpha\left(j\right)\right]^2\right), \tag{7.2}$$

with a piecewise constant function $f_\alpha$ minimizing,

$$min_{\beta_0^\omega, \beta_0^\varpi}\left(\sum_{j=1}^{\alpha}\left[I_{i,j} - \beta_0^\omega\right]^2 + \sum_{j=\alpha+1}^{M}\left[I_{i,j} - \beta_0^\varpi\right]^2\right). \tag{7.3}$$

The STEP detector optimizes three unknown variables: (1) $\beta_0^\omega$ giving the estimated intensity value for the blood pool, (2) $\beta_0^\varpi$ giving the estimated intensity value for the myocardium, and (3) $\alpha$ giving the expected endocardial position. More explicitly, the STEP detector (1) fits an optimal single knot piecewise constant function to an intensity profile for each knot position (Equation 7.3), (2) then selects the optimal knot position (Equation 7.2). Both the function fitting and knot selection procedures are optimal in the LS sense.

The idea can be generalized to $k^{th}$ order, such that STEP$_k$ fits an optimal single knot piecewise $k^{th}$ order polynomial function for each knot position instead. Using the notations from Equations 7.2 and 7.3, the generalized $k^{th}$ order detector computes,

$$\hat{\theta}_i^{STEP_k} = argmin_{\alpha \in \{1,2 \ldots M\}} \left( \sum_{j=1}^{M} \left[ I_{i,j} - f_\alpha^k \left( j \right) \right]^2 \right), \tag{7.4}$$

with a piecewise $k^{th}$ order polynomial function $f_\alpha^k$ minimizing,

$$\begin{aligned} & min_{\beta_0^\omega, \beta_1^\omega, \ldots \beta_k^\omega} \left( \sum_{j=1}^{\alpha} \left[ I_{i,j} - \left( \beta_0^\omega + j \cdot \beta_1^\omega + j^2 \cdot \beta_2^\omega + \ldots j^k \cdot \beta_k^\omega \right) \right]^2 \right) + \\ & min_{\beta_0^\varpi, \beta_1^\varpi, \ldots \beta_k^\varpi} \left( \sum_{j=\alpha+1}^{M} \left[ I_{i,j} - \left( \beta_0^\varpi + j \cdot \beta_1^\varpi + j^2 \cdot \beta_2^\varpi + \ldots j^k \cdot \beta_k^\varpi \right) \right]^2 \right). \end{aligned} \tag{7.5}$$

The regression task of finding the optimal parameters can be performed solving $\beta^\omega = \left( \mathbf{D}_\alpha^T \mathbf{D}_\alpha \right)^{-1} \mathbf{D}_\alpha^T \mathbf{y}$, where

$$\beta^\omega = \left[ \beta_0^\omega, \, \beta_1^\omega, \ldots, \, \beta_k^\omega \right]^T, \, \mathbf{D}_\alpha = \begin{bmatrix} 1 & j & \cdots & j^k \\ 1 & j & \cdots & j^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & j & \cdots & j^k \end{bmatrix}, \, \mathbf{y} = \left[ I_{i,1}, \, I_{i,2}, \ldots, \, I_{i,\alpha} \right]^T. \tag{7.6}$$

$\mathbf{D}_\alpha$ is a $\alpha \times (k+1)$ constant design matrix that can be computed offline for all possible $\alpha$ values ($\leq M$). Please note that the computation of $\beta^\varpi = \left[ \beta_0^\varpi, \, \beta_1^\varpi, \ldots, \, \beta_k^\varpi \right]^T$ follows a similar method as the $\beta^\omega$'s (see Figure 7.1).

### 7.2.2 Maximum Likelihood Edge Detector

STEP$_k$ makes an assumption that the myocardium and blood pool sections of a given intensity profile can be represented using $k^{th}$ order polynomial functions. However, this assumption might only be valid for parts of the endocardial surface and cardiac cycle. As an example, STEP$_0$ might be a proper detector for the apical region, while performing poor for the basal anterior region at the end systole (ES). On the other hand, STEP$_2$ might outperform other generalized detectors for the apical region at ES, and produce highly biased detection results at the end diastole (ED). The responses of the generalized STEP detectors with orders $\leq p$ can be combined using a statistical

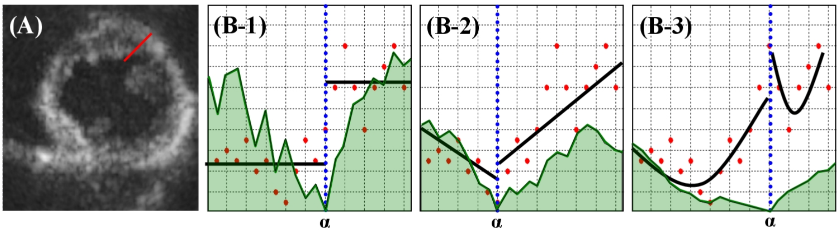


Figure 7.1: The endocardial edge detection with generalized STEP detectors. (A) The target image and the intensity profile (red) are represented. (B-1) $0^{th}$ , (B-2) $1^{st}$ and (B-3) $2^{nd}$ order STEP edge detections are shown; $x$ axis gives the spatial position, $y$ axis gives the intensity, red dots show the image intensity values on the intensity profile, black lines (curves) are the fitted piecewise polynomial functions, green regions show the minimized energy functions from Equation 7.4, and $\alpha$ shows the detected edge position.

learning approach, where the *learned* confidences of the base detectors for a given spatial and temporal position determine their weights.

The system can be described for the $i^{th}$ intensity profile at cardiac cycle position $\zeta \in [0:ES,\, 1:ED]$ using a general linear model as:

$$\mathbf{s}_{\mathbf{i},\zeta} = \mathbf{1}\theta_{\mathbf{i},\zeta} + \mathbf{b}_{\mathbf{i},\zeta} + \mathbf{w}_{\mathbf{i},\zeta}, \tag{7.7}$$

where

1. $\theta_{\mathbf{i},\zeta}$ is the unknown endocardial edge position (scalar valued),
2. $\mathbf{s}_{\mathbf{i},\zeta} = \left[\hat{\theta}_{i,\zeta}^{STEP_0},\, \hat{\theta}_{i,\zeta}^{STEP_1}, \ldots \hat{\theta}_{i,\zeta}^{STEP_p}\right]^T$ is the measurement vector for the base edge detectors,
3. $\mathbf{b}_{\mathbf{i},\zeta} = \left[Bias_{i,\zeta}^{STEP_0},\, Bias_{i,\zeta}^{STEP_1}, \ldots Bias_{i,\zeta}^{STEP_p}\right]^T$ is the learned bias vector,
4. $\mathbf{w}_{\mathbf{i},\zeta}$ is the noise vector with a learned Gaussian probability distribution function $N\left(0,\, C_{i,\zeta}\right)$ for the $i^{th}$ intensity profile at time $\zeta$.

Note that $C_{i,\zeta}$ is a $p \times p$ symmetric matrix holding the error covariances between the base detectors (e.g. $C_{i,\zeta}\left(q,\, q\right)$ holds the error variance of STEP$_q$ edge estimations, and $C_{i,\zeta}\left(q,\, r\right)$ holds the error covariance between STEP$_q$ and STEP$_r$ edge estimations for the $i^{th}$ intensity profile at $\zeta$). The bias vector can be merged with the measurement vector giving,

$$\mathbf{y}_{\mathbf{i},\zeta} = \mathbf{1}\theta_{\mathbf{i},\zeta} + \mathbf{w}_{i,\zeta}. \tag{7.8}$$

Maximum likelihood estimator of $\theta_{i,\zeta}$ can be derived as [8]:

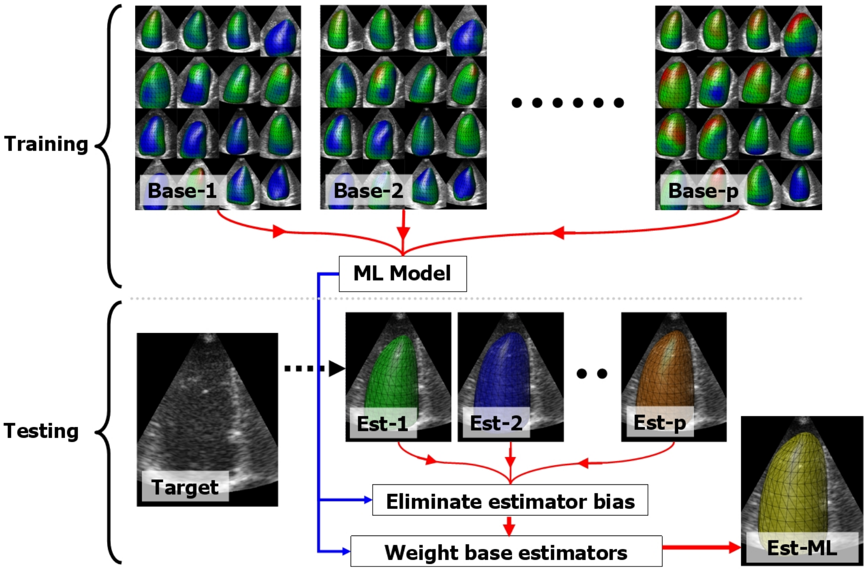


Figure 7.2: Overview of the ML estimator. Training stage produces a ML model. For a given test image, (1) the base detectors produces their estimates (green, blue and orange meshes), (2) the bias of the estimates are eliminated using the ML model, and (3) the base estimators are weighted and fused using the ML model (yellow mesh).

$$\hat{\theta}_{i,\zeta}^{ML} = \frac{\mathbf{1}^T C_{i,\zeta}^{-1}}{\mathbf{1}^T C_{i,\zeta}^{-1}\mathbf{1}} \mathbf{y}_{\mathbf{i},\zeta}. \tag{7.9}$$

$\hat{\theta}_{i,\zeta}^{ML}$ is an unbiased estimator with the variance of $\sigma_{i,\zeta}^2 = \left(\mathbf{1}^{\mathbf{T}} C_{i,\zeta}^{-1}\mathbf{1}\right)^{-1}$. The estimator variance can be utilized as the measurement noise in the Kalman filter. ML endocardial edge estimations for the model at $\zeta$ are given as,

$$\hat{\theta}_{i,\zeta}^{ML} \sim N\left(\theta_{\mathbf{i},\zeta}, \sigma_{i,\zeta}^2\right), \qquad i = 1, \dots, N. \tag{7.10}$$

Please see Figure 7.2 for an overview of the ML training and testing processes.

## 7.3 Results

3D echocardiography was performed on 10 healthy subjects and 19 subjects with recent first time myocardial infarction, using a Vivid 7 (26 recordings) or a Vivid E9 (3 recordings) ultrasound scanner (GE Vingmed Ultrasound, Norway) with a matrix array (3V) transducer. The endocardial border segmentation of the recordings was performed by a trained medical doctor using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway).

An $N$-fold cross validation (CV) was applied for the evaluation of the ML method (*STEP-ML*) using the STEP$_0$, STEP$_1$ and STEP$_2$ as the base detectors. For each testing fold, the other folds were used for learning a ML model at the ED and ES frames. The ML model included bias, covariance and corresponding ML weights for the base detectors at 528 evenly distributed endocardial surface positions (see Figure 7.3). The learned model was later interpolated between ED and ES through the cardiac cycle using a linear interpolation. For the testing, (1) STEP-ML detector was used in connection to the Kalman tracking framework with the learned ML model. The error measurements including the (a) *absolute surface point error (ASPE)* giving the average absolute distance of predicted surface points to a ground truth surface, (b) *squared surface point error (SSPE)* giving the average squared distance of predicted surface points to a ground truth surface, and (c) *absolute volume error (AVE)* giving the average absolute volume error of predicted surfaces were computed. Similar error measurements were also computed for the STEP$_0$, STEP$_1$ and STEP$_2$ detectors using all 29 recordings directly without a CV *(as these base detectors do not require a training)*. For a comparison, a previously defined ML method (*CLS-ML*) [8], which uses MG, STEP and MFMC as the base detectors, was also evaluated via $N$-fold CV (see Table 7.1).

*Signed surface error polar plots*, showing the average signed distances between the predicted and ground-truth surfaces using 17-segment model of the American Heart Association [10], for the STEP$_0$, STEP$_1$, STEP$_2$ and STEP-ML detectors are represented in Figure 7.4.

In the Kalman tracking framework, a handcrafted Doo-Sabin endocardial model consisting of 20 control points was used as LV model [9]. Edge measurements were performed in 528 intensity profiles distributed evenly across the endocardial surface. Each profile consisted of 30 samples, spaced 1 $mm$ apart. During the edge detection, normal displacement measurements that were significantly different from their neighbors were discarded as outliers. The tracking framework is implemented in C++, and processed each frame in 15.5$ms$ with STEP$_0$, 16.3$ms$ with STEP$_1$, 17.7$ms$ with STEP$_2$, 46.6$ms$ with STEP-ML, and 81$ms$ with CLS-ML when executed on a 2.80 GHz Intel Core 2 Duo CPU.

| | ASPE [$mm$] | SSPE [$mm^2$] | AVE [%] |
|---|---|---|---|
| STEP$_0$ | 2.211 – 2.153 | 8.862 – 8.274 | 15.718 – 13.674 |
| STEP$_1$ | 2.019 – 2.326 | 7.467 – 9.949 | 10.166 – 19.421 |
| STEP$_2$ | 2.268 – 2.941 | 10.831 – 15.428 | 13.137 – 30.547 |
| STEP-ML | 1.982 – 2.057 | 8.164 – 8.129 | 12.346 – 13.562 |
| CLS-ML | 1.999 – 2.043 | 7.542 – 7.545 | 12.592 – 14.298 |

Table 7.1: Absolute surface point (in $mm$) , squared surface point (in $mm^2$), and absolute volume (in percentages) errors at ED and ES frames, [ED error - ES error], for the Kalman tracking framework using STEP$_0$, STEP$_1$ and STEP$_2$, STEP-ML and CLS-ML edge detectors.

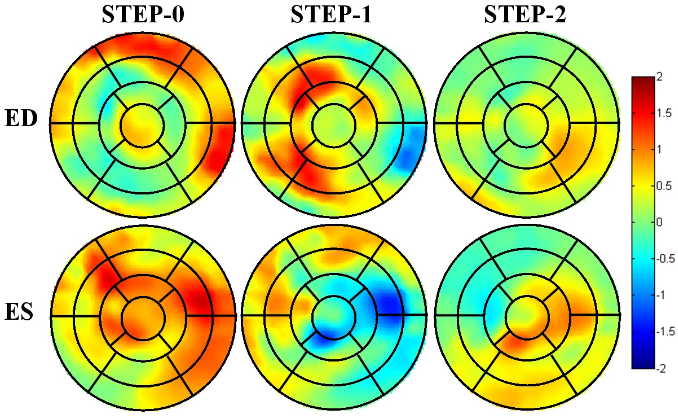


Figure 7.3: 17-segment model representations for the learned ML weights for the base detectors: +2 weight is red, -2 weight is blue, 0 weight is green. Please note that the base detector weights vary spatially and temporarily. For a given endocardial surface and cardiac cycle position, the sum of the weights is always 1, where the negative weights are allowed.

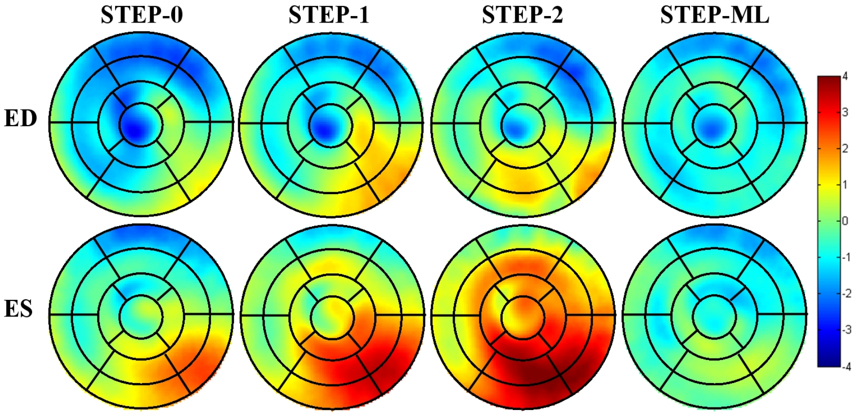


Figure 7.4: 17-segment model representations for the signed surface error: 4mm overestimation is red, 4mm under-estimation is blue, 0mm no-error is green.

## 7.4 Discussion and Conclusion

In this paper, we first introduced the generalized step criterion edge detectors, then combined the responses of these generalized detectors using a space-time dependent ML approach. To our knowledge, step criterion edge detector has not been generalized to higher order polynomial functions prior to this study.

Our analyses showed that STEP$_1$ and STEP$_0$ outperform the other generalized

STEP detectors at ED and ES frames respectively (see Table 7.1, rows 1,2 and 3). These results suggest that the intensity distribution of myocardium and blood pool can be modeled better by constant functions at ES, and linear functions at ED. The sole application of $\text{STEP}_2$ leads to worse segmentation results (see Table 1, row 3). However, the relatively high ML weight of $\text{STEP}_2$ at mid inferolateral segment during ED shows the positive contribution of this base detector (see Figure 7.3, row 1). Accordingly, the combined usage of generalized STEP detectors, STEP-ML, produces better tracking results compared with the individual applications of the generalized detectors (see Table 7.1, row 4). Furthermore, STEP-ML introduces a significantly lower regional estimation bias (see Figure 7.4). This is due to the learned bias information stored in the ML model, which is factored in during the ML estimation.

STEP-ML produced comparable results with the previously defined ML detector, CLS-ML, [8] (see Table 7.1, rows 4 and 5). This shows that the linear combination of generalized STEP detectors can generate close results to MG and MFMC detectors, which are utilized in CLS-ML. Furthermore, STEP-ML reduces the processing time for each frame over 42% with respect to CLS-ML.

The current and previously proposed ML based approaches seek for an optimal linear combination of multiple base detectors. The non-linear detector fusion approaches might be investigated in a future study. Utilizing base detectors with well defined energy functions, such as the generalized STEP and MFMC, allows the definition of unified energy functions for the endocardial edge detection process. The unified energy functions might help us understand the visual perception process for this specific task; the derivation and usage of these functions might also be investigated in a future study.

# References

[1] Z. Zhang, D. J. Sahn, and X. Song, "Temporal diffeomorphic motion analysis from echocardiographic sequences by using intensity transitivity consistency," in *Proceedings of the Second International Conference on Statistical Atlases and Computational Models of the Heart*, STACOM'11, (Berlin, Heidelberg), pp. 274–284, Springer-Verlag, 2012.

[2] P. Pearlman, H. Tagare, B. Lin, A. Sinusas, and J. Duncan, "Segmentation of 3d radio frequency echocardiography using a spatio-temporal predictor," *Medical Image Analysis*, vol. 16, no. 2, pp. 351 – 360, 2012.

[3] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 987–1010, Aug. 2006.

[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] S. I. Rabben, A. H. Torp, A. Støylen, S. Slørdahl, K. Bjørnstad, B. O. Haugen, and B. Angelsen, "Semiautomatic contour detection in ultrasound m-mode images," *Ultrasound in Medicine & Biology*, vol. 26, no. 2, pp. 287 – 296, 2000.

[6] S. Venkatesh and R. Owens, "On the classification of image features," *Pattern Recogn. Lett.*, vol. 11, no. 5, pp. 339–349, 1990.

[7] E. Dikici and F. Orderud, "Graph-cut based edge detection for Kalman filter based left ventricle tracking in 3d+t echocardiography," in *Computing in Cardiology, 2010*, pp. 205 –208, sept. 2010.

[8] E. Dikici, F. Orderud, and H. Torp, "Best linear unbiased estimator for Kalman filter based left ventricle tracking in 3d+t echocardiography," in *Mathematical Methods in Biomedical Image Analysis (MMBIA), 2012. IEEE Workshop on*, pp. 201 –208, jan. 2012.

[9] F. Orderud and S. Rabben, "Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces," in *Computer Vision and Pattern Recognition (CVPR), 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[10] M. D. Cerqueira, N. J. Weissman, V. Dilsizian, *et al.*, “Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association. ,” *Circulation*, vol. 105, pp. 539–542, Jan. 2002.

# Chapter 8

# Isoparametric Finite Element Analysis for Doo-Sabin Subdivision Models

Engin Dikici[1], Sten Roar Snare[2] and Fredrik Orderud[2]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

We introduce an isoparametric finite element analysis method for models generated using Doo-Sabin subdivision surfaces. Our approach aims to narrow the gap between geometric modeling and physical simulation that have traditionally been treated as separate modules. This separation is due to the substantial geometric representation differences between these two processes. Accordingly, a unified representation is investigated in this study. Our proposed method performs the geometric modeling via Doo-Sabin subdivision surfaces, which are defined as the limit surface of a recursive Doo-Sabin refinement process. The same basis functions are later utilized to define isoparametric shell elements for physical simulation. Furthermore, the accuracy of the simulation can be adjusted by the basis refinements, without changing the geometry or its parametrization. The unified representation allows rapid data transfer between geometric design and finite-element analysis, eliminating the need for inconvenient remodeling/meshing procedures commonly deployed. Experiments show that the physical simulation accuracy of the introduced models quickly converges to high resolution finite element models, using classical hexahedron and triangular prism elements.

## 8.1 Introduction

Finite element analysis (FEA) tools have dramatically enhanced their functionalities over the recent years by (1) incorporating tools for more realistic physical simulations, (2) increasing the dimensional limits of mechanical models, and (3) developing computational means enabling engineers to create and evaluate alternative analysis models [1]. Computer aided geometric design (CAGD) systems have also gone through evolutionary improvements by (1) developing new mathematical representations of the designed objects, and (2) factoring in “physics-like” considerations into the design process [2]. Despite the vast developments in the field, severe integration problems are encountered when combining CAGD and FEA systems, mainly due to geometric representation differences [3].

Models generated by CAGD systems represent exact geometries, and are typically not suitable as direct inputs for finite-element modelers. This gap is often filled by intermediate tools that convert high-order CAGD models into low-order finer meshes. However, the meshing process can be slow, and cause approximation errors [4]. Even if the meshing is completed successfully, the outputs of an FEA system cannot be directly applied to the original geometric model since there is no intuitive mapping back to the original design degrees of freedom [5]. These limitations can become a major problem when design and analysis are performed back and forth over several iterations. The design-analysis loop is built around a trade-off between simulation speed and precision. Rapid analyses are preferred in the preliminary stages of the process. On the other hand, highly accurate analyses are needed in the later stages. Setting up a proper trade-off is also a challenging task when the different geometric representations are utilized in different tools. Unified representation methods, allowing CAGD and FEA to utilize similar model definitions, have been introduced for several geometric representation schemes to overcome these limitations.

Doo-Sabin subdivision method is conceptually simple and provides compact representations for the surface models, when the $\mathrm{C}^1$ continuity is adequate. It has been used for the geometric modeling of variety of curved objects, such as internal organs, over the recent years [6–8]. These studies involve physical simulations, and might greatly benefit from rapid design-analysis loops to produce mechanically constrained models. However, there has been no publications defining a unified representation scheme for the models generated using Doo-Sabin subdivision surfaces prior to this study. The major motivation for our study is to eliminate this gap by introducing a unified representation for CAGD and FEA when Doo-Sabin surfaces are deployed during the design process. First, the basis functions for the Doo-Sabin limit surfaces are described using an iterative algorithm. Then, the defined basis functions are utilized for the isoparametric formulation of the shell elements during the physical simulation. Degenerated solids based approach, which factors in shear deformations via Mindlin plate theory, is employed to formulate shell elements. The accuracy of the simulation can be adjusted by basis refinements without changing the geometry or its parametrization. The proposed method avoids the need for intermediate meshing tools, and offers an intuitive control for the trade-off between the simulation-speed and precision. The design and analysis process can be performed as:

1. The geometrical modeling is performed using Doo-Sabin limit surfaces, which are bi-quadratic B-spline surfaces with $\mathrm{C}^1$ continuity [9].

2. The generated Doo-Sabin surface’s thickness information is defined to form a solid model.

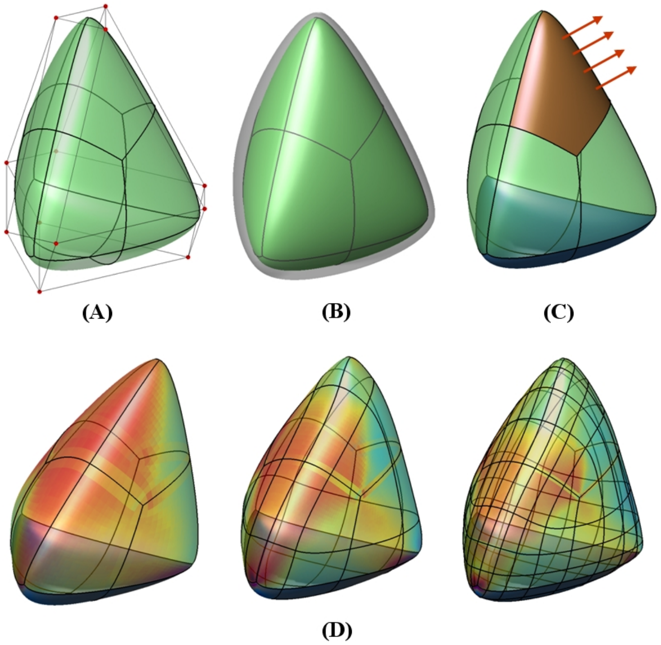


Figure 8.1: Design and analysis of a tetrahedral model. (A) Control point wireframe mesh and the corresponding Doo-Sabin limit surface are shown. (B) Model thickness is set. (C) Boundary and force surfaces are represented with blue and bronze colors respectively. (D) Analysis is performed with the desired accuracy. Interpolated principal stress fields are shown for not refined (left), refined (middle) and doubly refined (right) basis functions.

3. The stiffness matrix of the model is assembled using shell elements with varying thicknesses. A degenerated solids based approach is adopted for modeling the shells, where the element shape functions are determined by the basis functions of the limit surface. FEA precision may be increased by refining the model basis function multiple times prior to stiffness matrix assembly.

4. Boundary conditions are set for delineating the fixed surface components.

5. Surface forces are applied, and the model strains and stresses are measured.

6. Based on the analysis results, the design-analysis loop may be ended, or continued by going back to the modeling stage.

Figure 8.1 illustrates the major steps of the process for a tetrahedral model.

The outline of the paper is as follows: Section 8.2 goes through the recent work on subdivision surfaces, and discusses the unified representation methods for design and analysis. The Doo-Sabin subdivision surface scheme is described, and the limit surface basis functions are derived in Section 8.3. Section 8.4 is dedicated to the finite element analysis. Subsection 8.4.1 provides a brief overview of the isoparametric analysis, whereas Subsection 8.4.2 describes the degenerated solid element concept and shell elements. Subsection 8.4.3 gives the strain-stress formulation for the proposed method, and Subsection 8.4.4 provides a description of the stiffness matrix assembly. In Section 8.5, the proposed method's convergence properties are represented using four distinct models. We conclude, in Section 8.6, with a discussion of directions of current and future work.

## 8.2 Previous work

### 8.2.1 Subdivision Surfaces

Subdivision surfaces are popular for computer animation, computer aided design and geometric modeling tasks due to their conceptual simplicity and efficiency [10]. The subdivision process constructs smooth surfaces through a converging procedure of repeated refinements starting from an initial control mesh (see Figure 8.2). Each subdivision iteration produces a finer mesh from a coarser mesh by using a set of refinement rules that define a *subdivision scheme* [11, 12]. The first subdivision schemes were introduced in 1978 by Catmull and Clark [13] and Doo and Sabin [9] to address some of the drawbacks of the conventional spline patches when modeling arbitrary topology surfaces. Since then, many other schemes have been proposed and studied [14].

Subdivision schemes are categorized as being either approximating or interpolating. The limit surface approximates the initial polygonal mesh for approximating methods, meaning that the control vertices do not necessarily lie on the limit surface. Catmull-Clark, Doo-Sabin, $\sqrt{3}$ subdivision [15], Mid-Edge subdivision [16], and Loop [17] are some of the most widely used approximating schemes. In the interpolating methods, the control vertices always lie on the limit surface as the subdivision process adds new control points interpolating the existing ones. Some examples for the interpolating methods are Kobbelt [18] and Butterfly [19] schemes.

### 8.2.2 Unified Representation Methods

The unified representation paradigm denotes a design and analysis environment in which the geometric model utilizes the same underlying representation as the appropriate finite-element simulation [5]. The concept was first applied for adaptively refined quadrilateral meshes that approximate smooth surfaces with arbitrary topologies in [18]. This study introduced a simple interpolating subdivision scheme that generated almost planar faces, which were modeled by quadrilateral elements during the analysis.

The unified representation has been shown to be applicable for spline based models. In [1], B-spline basis functions were utilized for geometrical modeling of surface patches and the formulation of isoparametric finite elements. Wang et. al. [20] extended the method for arbitrary manifold spline surfaces [21]. Their approach represented the

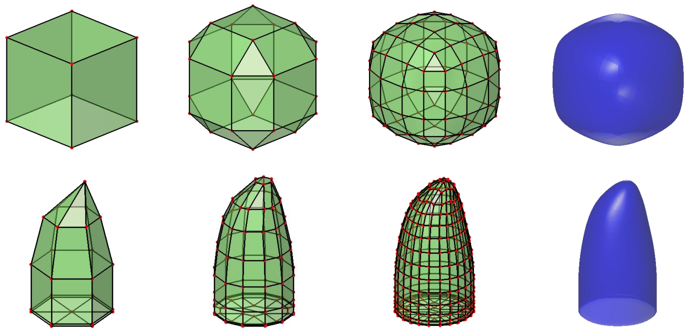


Figure 8.2: Two steps of Doo-Sabin refinements and the corresponding limit surfaces for cube and and heart's left ventricle surface models.

deformations as a linear combination of triangular B-splines over shell elements, then the thin-shell simulation was computed through the minimization of Kirchhoff-Love energy. In [4], basis functions generated by nonuniform rational B-splines *(NURBS)* were employed to construct exact geometrical models, and to model NURBS solids for the analysis. More recently, Benson et. al. [22] introduced a NURBS based isogeometric analysis method formulating the shells structures by Reissner – Mindlin theory, based on the degenerated solid concept. In [23], Kirchhoff – Love shell elements were used for the isogeometric analysis of the NURBS surface models.

The unified approach has also been applied to subdivision surface based models over the last decade. In [5, 24, 25], Loop subdivision surfaces were utilized as a common foundation for modeling, simulation, and design. Thin shell elements were defined using Loop basis functions, and their interpolated displacement fields were found using Kirchhoff-Love theory of thin shells. Burkhart et. al. [26] represented a unified approach for three-manifold hexahedral meshes, where Catmull-Clark solids were used for the geometric modeling and the physical simulation.

Previously defined unified methods for other subdivision schemes can not be modified for Doo-Sabin surface models with thickness information in intuitive manners. [26] requires an additional hexagonal meshing procedure prior to FEA for Doo-Sabin surface models; it does not provide a unified representation for CAGD and FEA stages in our case. [5, 24, 25] consider surface models that can be represented using shell elements as our study. However, these methods inherit thin shell assumptions, which neglect the shear deformations. This would limit the FEA accuracy of the geometric models represented using thick shells. We introduce an approach employing degenerated solids based shell elements to model Doo-Sabin surface patches. Our method factors in Mindlin plate theory to properly simulate shear deformations. Hence, it provides a useful and novel unified solution for subdivision surface models with thickness information.

## 8.3 Evaluation of Doo-Sabin subdivision surfaces

The Doo-Sabin surface [9] is a type of subdivision surface that generalizes bi-quadric B-spline surfaces to an arbitrary topology. Doo-Sabin subdivision process can be described using a matrix notation, as utilized by Stam for Catmull-Clark surfaces in [27].

For a given polyhedron, there is a surface patch for each control vertex $\mathbf{q}$ that is defined using (1) the surface faces including $\mathbf{q}$, and (2) all neighboring control vertices of $\mathbf{q}$ describing these faces. Control vertices of a surface patch with $m$ control vertices can be represented using a matrix notation as,

$$\mathbf{Q}_n = \left[\mathbf{q}_1, \, \mathbf{q}_2 \, , \ldots , \, \mathbf{q}_m\right]^T , \tag{8.1}$$

where (1) $n \geq 0$ gives the number of performed subdivisions, and (2) $\mathbf{q}_i$ is a column vector of length three giving the Cartesian coordinates of the control vertex $i \in \{1, \, 2, \, \ldots, \, m\}$. The surface patch can be subdivided into four new sub-patches by multiplying subdivision matrix $\mathbf{S}$ with $\mathbf{Q}_n$, as is shown in Figure 8.3. The content of $\mathbf{S}$ originates from the regular Doo-Sabin subdivision rules, which are outlined in Appendix A. Control vertices for each sub-patch $k \in \{0, \, 1, \, 2, \, 3\}$ can be extracted from the subdivided control vertices using a picking matrix $\mathbf{P}_k$, such that $\mathbf{Q}_{n+1,k} = \mathbf{P}_k\mathbf{S}\mathbf{Q}_n$.

The sub-patch having the control vertices $\mathbf{Q}_{n+1,k}$ can either be (1) a regular patch including four quadrilateral faces, or (2) an irregular patch including three quadrilateral faces and an irregular face (valence $\neq 4$). If the sub-patch is a regular patch, then its limit surface can be evaluated directly since the sub-patch is a bi-quadric spline patch. Otherwise, if the sub-patch is an irregular patch, a successive subdivision operation on it yields a single irregular patch and three regular patches. By assuming, without loss of generality, that the irregular face in a sub-patch is located top-left, then the corresponding picking matrix $\mathbf{P}_k$ gives a regular $3 \times 3$ bi-quadric control vertex mesh when $k \neq 0$, and an irregular mesh when $k = 0$. This relation can be exploited by performing repeated subdivisions $\alpha$ times until the desired surface point is no longer within an irregular patch ($k \neq 0$). Denoting $\mathbf{S}_0 = \mathbf{P}_0\mathbf{S}$, we can express this as $\mathbf{Q}_{\alpha,k} = \mathbf{P}_k\mathbf{S}\mathbf{S}_0^{\alpha-1}\mathbf{Q}_0$.

Each limit surface position of a surface patch is uniquely defined using *parametric* patch coordinates $(\xi, \eta)$ that vary between 0 and 1. The number of the required subdivision steps $\alpha$ depends on the logarithm of $(\xi, \eta)$ as $\alpha = \lfloor -log_2\left(max\left\{\xi, \eta\right\}\right)\rfloor$. The sub-patch to pick after the final subdivision is determined using the following criteria:

$$k = \begin{cases} 1 & if \ \left(2^\alpha\xi > 1/2\right) \ and \ \left(2^\alpha\eta < 1/2\right) \\ 2 & if \ \left(2^\alpha\xi > 1/2\right) \ and \ \left(2^\alpha\eta > 1/2\right) \\ 3 & if \ \left(2^\alpha\xi < 1/2\right) \ and \ \left(2^\alpha\eta > 1/2\right) \end{cases} \tag{8.2}$$

Accordingly, the subdivision mapping function $\Omega$ can be defined as $\Omega\left(\xi, \eta\right) \rightarrow \left(k, \alpha\right)$ [27].

Direct evaluation of surface points can be performed for any patch location $(\xi, \eta)$ except $(0, 0)$, by subdividing a sufficient number of times, until the new subdivided

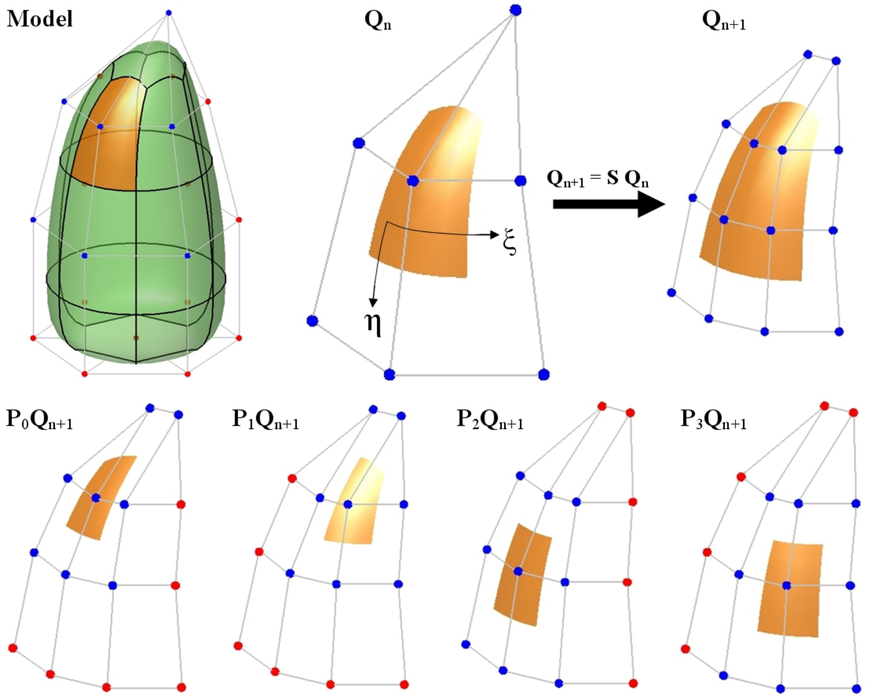


Figure 8.3: Illustration of the Doo-Sabin subdivision process. The control vertices $\mathbf{Q}_n$ that define the initial surface patch (top-middle) are subdivided into new control vertices $\mathbf{Q}_{n+1}$ (top-right) by multiplying $\mathbf{Q}_n$ with the subdivision matrix $\mathbf{S}$. Application 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 limit surface area as the original patch.

patch below $(\xi, \eta)$ no longer contains an extraordinary face, and treating the resulting sub-patch as an ordinary bi-quadric spline surface. For locations near $(0, 0)$, an approximate surface evaluation can be obtained by perturbing $(\xi, \eta)$ slightly to prevent $\alpha$ from growing beyond a predefined upper limit. However, this approximation might only be necessary for the visualization purposes, and can be completely avoided during the finite element analysis *(elaborated in Subsection 8.4.4)*.

Basis functions with regards to the original non-subdivided control vertices can similarly be computed by:

$$b\,(\xi, \eta)\,|_{\Omega(\xi,\eta)\to(k,\alpha)} = \left(\mathbf{P}_k \mathbf{S}\mathbf{S}_0^{\alpha-1}\right)^T \tilde{b}\,(t_{k,\alpha}\,(\xi, \eta))\,, \tag{8.3}$$

where (1) $\tilde{b}$ is the regular bi-quadric B-spline basis function defined in Appendix B, and (2) $t_{k,\alpha}$ is a domain mapping function used to map $(\xi, \eta)$ to the parametric patch coordinates within the desired sub-patch:

$$t_{k,\alpha}\left(\xi,\eta\right)=\begin{cases}\left(2^{\alpha+1}\xi-1,\;2^{\alpha+1}\eta\right) & if\;k=1\\ \left(2^{\alpha+1}\xi-1,\;2^{\alpha+1}\eta-1\right) & if\;k=2\\ \left(2^{\alpha+1}\xi,\;2^{\alpha+1}\eta-1\right) & if\;k=3\end{cases} \tag{8.4}$$

Partial derivatives of the basis functions, $b_\xi$ and $b_\eta$, are similarly computed by replacing $\tilde{b}\left(\xi,\eta\right)$ 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

$$\mathbf{x}\left(\xi,\eta\right)=\mathbf{Q}_0^T b\left(\xi,\eta\right). \tag{8.5}$$

Note that this approach is not dependent on diagonalization of the subdivision matrix. Doo–Sabin subdivision matrices are not in general diagonalizable, hence the direct evaluation approach produced in [27] is not computationally efficient for Doo-Sabin surfaces. However, repeated matrix multiplications performed $\alpha$ times will result in the same result. The associated increase in computational complexity due to the repeated matrix multiplications will not be a burden if the evaluation of basis functions is performed only once, and later re-used to compute surface points regardless of movement of the associated control vertices.

## 8.4 Finite element analysis

FEA is a numerical technique for approximating the field variables, also called the unknowns, encountered in the engineering problems. In a continuum, there is an infinite amount of field variables. The finite element procedure reduces these unknowns to a finite number by dividing the solution domain into small parts called the elements. The field variable distribution in each element is described using the shape functions, defined between vertices. After selecting the vertices and the appropriate shape functions, the element properties can be expressed by an element stiffness matrix $\mathbf{K}_e$. In solid mechanics, $\mathbf{K}_e\mathbf{u}_e=\mathbf{F}_e$ gives the relationship between the element vertex displacement vector $\mathbf{u}_e$ and force vector $\mathbf{F}_e$ using $\mathbf{K}_e$. The contribution of each element is finally assembled into the model stiffness matrix $\mathbf{K}_m$, completing the physical model definition.

A Doo-Sabin surface patch with thickness can be modeled using a shell element. The following sections represent an isoparametric definition for shell elements and the corresponding element stiffness matrix formulation. The assembly of element stiffness matrices into a model stiffness matrix is an intuitive task, hence it is not covered in this paper. However, the interested reader is referred to [28] for further information on the topic.

### 8.4.1 Isoparametric Formulation

The isoparametric formulation [29] uses a mapping function between *natural* and *physical* coordinate systems to define arbitrary object boundaries. In this concept, the problem domain is provided in the physical coordinate system, and the element shape functions are defined in terms of the natural coordinate system. The Doo-Sabin surface patches are inherently represented with an isoparametric formulation, where the basis functions induced by subdivision are utilized as the shape functions. Assuming that the natural coordinates $\xi$ and $\eta$ are scaled between -1 and 1, Doo-Sabin basis functions provide a proper mapping between the natural and physical coordinate system as,

$$\mathbf{x} = \sum_{i=1}^{m} b_i \left(\xi, \eta\right) \mathbf{q}_i, \tag{8.6}$$

where $\mathbf{q}_i$ is the physical coordinate of the $i^{th}$control vertex, $b_i$ is the corresponding basis function, and $\mathbf{x} = \left[x,\, y,\, z\right]^T$ is the mapped physical coordinate. Equation 8.6 needs to be further modified to factor in the element thickness, and then used for modeling an isoparametric shell element during the analysis.

### 8.4.2 A-I-J Element

The degenerated solids is an isoparametric formulation method for the shell elements. Many variations of the concept have been proposed and used [30], where the original definition by Ahmad et. al. [31] (A-I-J element) is utilized in this study. The approach converts 3D elements into 2D elements for the finite element analysis using 3D elastic theory. As an example, a 3D brick element is reduced to a shell element by deleting the intermediate vertices in the thickness direction, and then by projecting the vertices on each surface to the mid surface. However, the vertices on the 2 outer surfaces are not removed to keep the analysis in 3D, and to model shearing deformations. The theory is developed with the following assumptions:

1. The normal strains and stresses in the direction of element thickness are always zero. Therefore, the strain energy associated with the stresses perpendicular to the mid surface are neglected.

2. The mid surface normals may not remain as normals after a deformation, however they remain straight as used in Mindlin plate theory [32]. Hence, the shear strain energy needs to be considered during the analysis.

In the A-I-J method, solid elements are described using three surfaces, namely the top, bottom and mid (see Figure 8.4). Top and bottom surfaces are defined on the positive and negative surface normal directions using a similar basis function as the mid surface. These surfaces' control vertex matrices are

$$\mathbf{Q}^{top} = \left[\mathbf{q}_1^{top},\, \mathbf{q}_2^{top}, \ldots,\, \mathbf{q}_m^{top}\right]^T, \tag{8.7}$$

$$\mathbf{Q}^{bottom} = \left[\mathbf{q}_1^{bottom},\, \mathbf{q}_2^{bottom}, \ldots \mathbf{q}_m^{bottom}\right]^T, \tag{8.8}$$

where $\mathbf{Q} = \frac{\left(\mathbf{Q}^{top} + \mathbf{Q}^{bottom}\right)}{2}$.

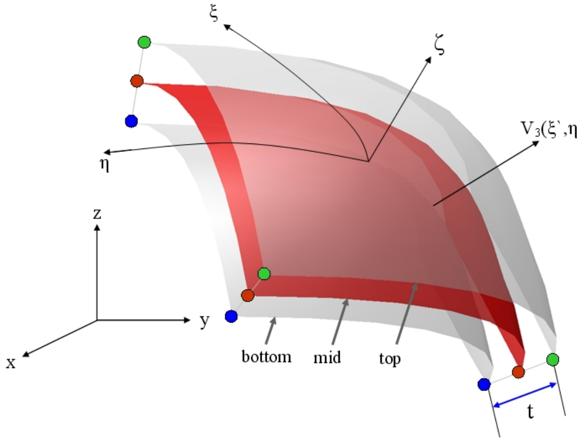


Figure 8.4: Geometry of the shell element. Control vertices of the bottom, mid and top surfaces are shown with blue, red, and green spheres respectively.

Assuming that an additional natural parameter $\zeta$ gets values between -1 and 1 on the respecting faces of the element; A-I-J defines the relationship between the physical and natural coordinates as

$$\mathbf{x} = \sum_{i=1}^{m} b_i\left(\xi, \eta\right) \frac{(1+\zeta)}{2} \mathbf{q}_i^{top} + \sum_{i=1}^{m} b_i\left(\xi, \eta\right) \frac{(1-\zeta)}{2} \mathbf{q}_i^{bottom}, \tag{8.9}$$

where $\mathbf{q}_i^{top}$ and $\mathbf{q}_i^{bottom}$ are the $i^{th}$ control vertices for the top and bottom surfaces respectively. Using the mid surface control vertices, Equation 8.9 can alternatively be written as

$$\mathbf{x} = \sum_{i=1}^{m} b_i\left(\xi, \eta\right) \left\{ \mathbf{q}_i + t_i \frac{\zeta}{2} \mathbf{v}_{3i} \right\}, \tag{8.10}$$

with

$$t_i = \left|\mathbf{q}_i^{top} - \mathbf{q}_i^{bottom}\right| \quad and \quad \mathbf{v}_{3i} = \frac{\mathbf{q}_i^{top} - \mathbf{q}_i^{bottom}}{\left|\mathbf{q}_i^{top} - \mathbf{q}_i^{bottom}\right|},$$

where $t_i$ and $\mathbf{v}_{3i}$ give the surface thickness and the surface normal direction for the $i^{th}$ control vertex respectively.

During the FEA of the shell structures, a *local* coordinate system ($x'$, $y'$, $z'$) of the element needs to be determined. By applying a linear interpolation, an orthogonal set of local coordinate axes for any point in the element are given by,

$$\mathbf{v}_3(\xi,\eta) = \frac{\sum_{i=1}^{m} b_i(\xi,\eta)\,\mathbf{v}_{3i}}{\left|\sum_{i=1}^{m} b_i(\xi,\eta)\,\mathbf{v}_{3i}\right|}, \tag{8.11}$$

$$\mathbf{v}_1(\xi,\eta) = \frac{\mathbf{i}\times\mathbf{v}_3}{|\mathbf{i}\times\mathbf{v}_3|}, \tag{8.12}$$

$$\mathbf{v}_2(\xi,\eta) = \mathbf{v}_3\times\mathbf{v}_1, \tag{8.13}$$

with $\mathbf{i} = [1,\,0,\,0]^T$. Therefore, the coordinate transformation matrix between the physical and local coordinates can be written as $\theta = [\mathbf{v}_1,\,\mathbf{v}_2,\,\mathbf{v}_3]$.

### 8.4.3 Strains and Stresses

The displacement field definition is derived intuitively from Equation 8.10. First, for each control vertex, two other axis vectors that are perpendicular to $\vec{V}_{3i}$ are defined as,

$$\mathbf{v}_{1i} = \frac{\mathbf{i}\times\mathbf{v}_{3i}}{|\mathbf{i}\times\mathbf{v}_{3i}|} \quad \textit{and} \quad \mathbf{v}_{2i} = \mathbf{v}_{3i}\times\mathbf{v}_{1i}.$$

Then, the displacement field $\mathbf{u} = [u,\,v,\,w]^T$ of any point in the element is given by,

$$\mathbf{u} = \sum_{i=1}^{m} b_i(\xi,\eta)\left\{\mathbf{u_i} + t_i\frac{\zeta}{2}\left[\mathbf{v}_{1i}, -\mathbf{v}_{2i}\right]\left[\begin{array}{c}\alpha_i\\ \beta_i\end{array}\right]\right\}, \tag{8.14}$$

where (1) $\mathbf{u_i} = [u_i,\,v_i,\,w_i]^T$ is the displacement vector for the $i^{th}$ control vertex, and (2) $\alpha_i$ and $\beta_i$ are scalar rotations in radians around $\mathbf{v}_{1i}$ and $\mathbf{v}_{2i}$ axes respectively. Equation 8.14 expresses the displacements in the physical coordinates, and assumes zero strains in the $\zeta$ direction. This formulation has five degrees of freedom at each control vertex; three parameters for the translations in $x$, $y$ and $z$ axes, and two parameters for the nodal rotations.

The strain vector $\varepsilon = [\varepsilon_{x'},\,\varepsilon_{y'},\,\gamma_{x'y'},\,\gamma_{x'z'},\,\gamma_{y'z'}]^T$ is defined by the first partial derivatives of the local displacement vector $\mathbf{u'} = [u',\,v',\,w']^T$ by,

$$\varepsilon = \left[\frac{\partial u'}{\partial x'}, \frac{\partial v'}{\partial y'}, \frac{\partial u'}{\partial y'} + \frac{\partial v'}{\partial x'}, \frac{\partial u'}{\partial z'} + \frac{\partial w'}{\partial x'}, \frac{\partial v'}{\partial z'} + \frac{\partial w'}{\partial y'}\right]^T, \tag{8.15}$$

where (1) $\varepsilon_{x'}$ and $\varepsilon_{y'}$ are the normal strains in $x'$ and $y'$ directions, and (2) $\gamma_{x'y'}$, $\gamma_{x'z'}$ and $\gamma_{y'z'}$ are the shear strains in the $x'y'$, $x'z'$ and $y'z'$ planes respectively [33]. Please note that the normal strain in $z'$ direction is neglected due to the first shell assumption.

The relation between the strain and stress *($\sigma$)* vectors can be formed using the *Generalized Hooke's Law*, which states that the components of stress are linearly related to the components of strain by $\sigma = \mathbf{D}\varepsilon$. In this study, $\mathbf{D}$ is defined for an isotropic material as,

$$\mathbf{D} = \frac{E}{(1-\mu^2)} \begin{bmatrix} 1 & \mu & 0 & 0 & 0 \\ \mu & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1-\mu}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1-\mu}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1-\mu}{2} \end{bmatrix}, \tag{8.16}$$

where $E$ is the Young's modulus and $\mu$ is the Poisson's ratio.

### 8.4.4 Assembly of Element Stiffness Matrix

Using the standard variational formulation, the stiffness matrix of an element is given by:

$$\mathbf{K}_e = \iiint \mathbf{B}^T \mathbf{D} \mathbf{B} dxdydz, \tag{8.17}$$

where the strain-displacement matrix $\mathbf{B}$ relates strains to control vertex displacements ($\delta$) by $\varepsilon = \mathbf{B}\delta$. To compute $\mathbf{B}$, firstly the components of $\varepsilon$ are found using,

$$\begin{bmatrix} \frac{\partial u'}{\partial x'} & \frac{\partial v'}{\partial x'} & \frac{\partial w'}{\partial x'} \\ \frac{\partial u'}{\partial y'} & \frac{\partial v'}{\partial y'} & \frac{\partial w'}{\partial y'} \\ \frac{\partial u'}{\partial z'} & \frac{\partial v'}{\partial z'} & \frac{\partial w'}{\partial z'} \end{bmatrix} = \theta^T \left( \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial v}{\partial \xi} & \frac{\partial w}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial v}{\partial \eta} & \frac{\partial w}{\partial \eta} \\ \frac{\partial u}{\partial \zeta} & \frac{\partial v}{\partial \zeta} & \frac{\partial w}{\partial \zeta} \end{bmatrix} \right) \theta \tag{8.18}$$

with

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}, \tag{8.19}$$

where (1) $\mathbf{J}$ is the Jacobian matrix derived using Equation 8.10, and (2) the derivatives $\frac{\partial u}{\partial \xi}, \frac{\partial v}{\partial \eta} \ldots etc.$ are obtained using Equation 8.14. After assembling $\varepsilon$, $\mathbf{B}$ can be found solving $\varepsilon = \mathbf{B}\delta = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_m] [\delta_1, \delta_2, \ldots, \delta_m]^T$ where $\delta_i = [u_i, v_i, w_i, \alpha_i, \beta_i]^T$ [33]. The infinitesimal volume computed in physical coordinates can be expressed in terms of the volume in natural coordinates as $dxdydz = |\mathbf{J}|\, d\xi d\eta d\zeta$.

Finally, $\mathbf{K}_e = \iiint \mathbf{B}^T \mathbf{D} \mathbf{B} \, |\mathbf{J}| \, d\xi d\eta d\zeta$ can be numerically estimated by Gauss Legendre quadrature rules. Using two samples in $\zeta$ direction and minimum four samples in both $\xi$ and $\eta$ directions is sufficient for thin shell element stiffness matrix calculations [31]. The abscissas of the selected Gaussian quadrature rule should exclude parametric coordinates ($\xi = 0$, $\eta = 0$, $\zeta$) to avoid approximate basis evaluations.

## 8.5 Results

The accuracies of the physical models using Doo-Sabin limit surface patch elements were evaluated by comparing them against the high resolution classical finite element models. The analyses were performed on four models: the smoothed cube *(S-cube)*,

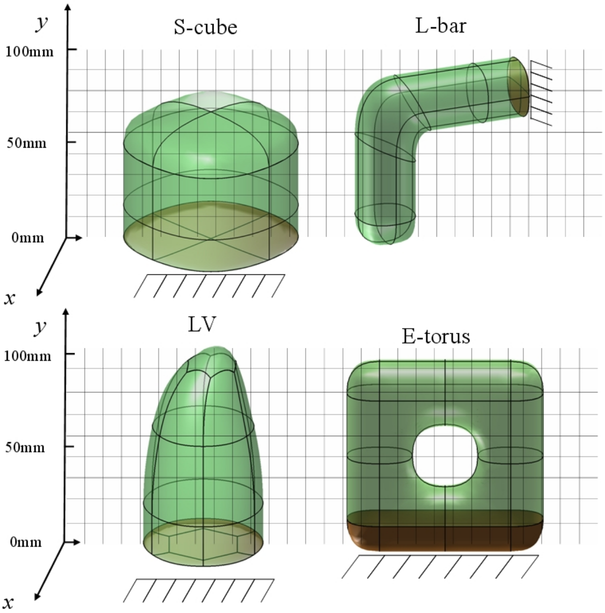


Figure 8.5: Geometry of the testing models.

left ventricle *(LV)*, l-shaped bar *(L-bar)* and edgy torus *(E-torus)*. These models were generated using Doo-Sabin subdivision algorithm, where the limit surfaces define the exact model geometries. Dimensional properties and boundary conditions of the models are represented in Figure 8.5. The model thicknesses were set to $2mm$. Young's modulus and Poisson's ratio values were defined as 3.0 GPa and 0.48 respectively.

For each of the Doo-Sabin models, a high resolution reference model was computed as follows:

1. A polygonal model was generated: a low resolution Doo-Sabin model was refined multiple times till the average distance between the limit and refined model surfaces was below a predefined threshold value $\varsigma = 0.05mm$.

2. Valence $\geq 5$ faces were eliminated: the refined model's surface faces with valences greater than 4 were further divided geometrically.

3. A solid finite element model was computed: (a) the refined surface faces with valence$=$ 4 were used as the middle surfaces of 8-node hexahedron *(brick)* elements with 24 degrees of freedom, (b) the refined surface faces with valence$=$ 3 were used as the middle surfaces of 6-node triangular prism *(wedge)* elements with 18 degrees of freedom. Interested reader is referred to [34] for the derivations of these classical elements.

4. The element thicknesses were set: hexahedron and wedge elements were

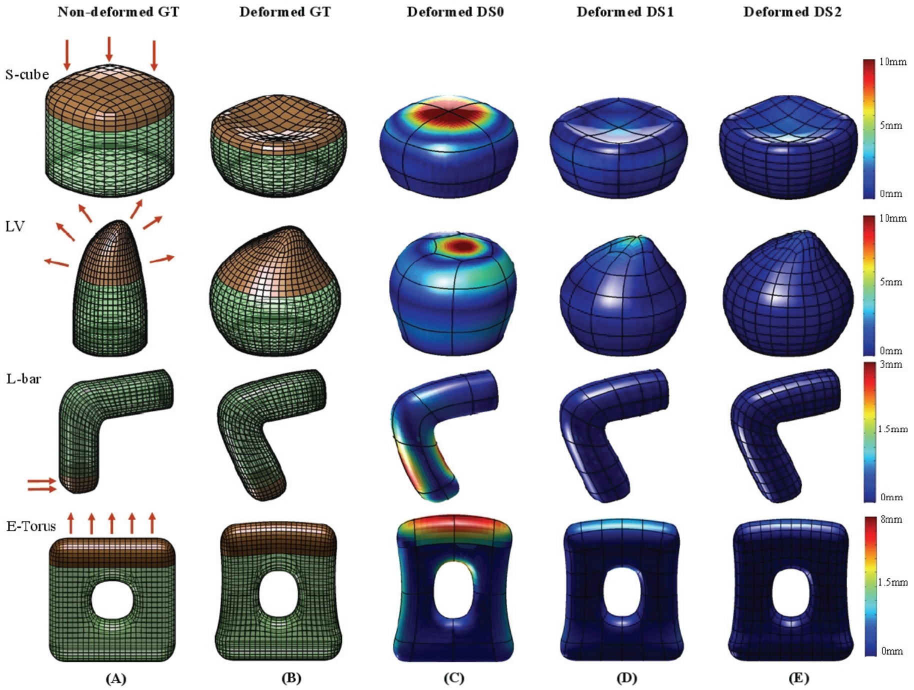


Figure 8.6: Surface displacement error convergences for the S-cube (1st row), LV (2nd row), L-bar (3rd row), E-torus (4th row) models. Column (A and B): Ground truth polygonal meshes before and after the surface forces are applied on the bronze colored elements in the directions shown in red. Columns (C, D and E): non-refined, refined, and doubly refined Doo-Sabin models are colored based on the displacement errors.

assigned similar thicknesses as their corresponding Doo-Sabin limit surface patch elements.

After generating the reference model, stiffness matrices for the reference model (GT) and the Doo-Sabin models with non refined (DS0), refined (DS1) and doubly refined (DS2) basis functions were computed.

For testing the convergence properties of the proposed method, the surface forces were applied on the GT, DS0, DS1 and DS2 models as shown in Figure 8.6. Then, the deformed Doo-Sabin model surfaces were compared against the corresponding deformed reference model surfaces. The total number of elements in each model, stiffness matrix computation times, absolute and percent surface displacement errors are reported in Table 8.1. The mean and the standard deviation of the absolute surface displacement errors are plotted in Figure 8.7 for each model.

The simulation accuracies of the Doo-Sabin models converge to the reference models fairly fast. As represented in Figure 8.7, non-refined S-cube model consisted of 16 limit surface patch elements, and generated 2.4 mm mean absolute error. On

Table 8.1: Total number of elements, mean absolute error, percent error and the stiffness matrix computation times for each model.

| | S-cube | | | | LV | | | |
|---|---|---|---|---|---|---|---|---|
| Model | GT | DS0 | DS1 | DS2 | GT | DS0 | DS1 | DS2 |
| Elements | 898 | 16 | 56 | 224 | 1732 | 26 | 108 | 432 |
| Computation time (sec)[a] | 55.2 | 1.3 | 10.2 | 20.3 | 369.4 | 2.3 | 27.3 | 105.1 |
| Mean absolute error (mm)[b] | 0 | 2.4 | 1.1 | 0.8 | 0 | 2.4 | 0.6 | 0.5 |
| Percent error (%)[c] | 0 | 13.3 | 5.9 | 4.6 | 0 | 6.7 | 1.7 | 1.4 |

| | L-bar | | | | E-torus | | | |
|---|---|---|---|---|---|---|---|---|
| Model | GT | DS0 | DS1 | DS2 | GT | DS0 | DS1 | DS2 |
| Elements | 1410 | 24 | 88 | 352 | 2056 | 32 | 128 | 512 |
| Computation time (sec)[a] | 209.9 | 2.3 | 20.8 | 80.5 | 452.1 | 2.4 | 30.1 | 153.8 |
| Mean absolute error (mm)[b] | 0 | 1.1 | 0.4 | 0.3 | 0 | 1.9 | 0.8 | 0.5 |
| Percent error (%)[c] | 0 | 5.1 | 1.9 | 1.3 | 0 | 10.5 | 4.4 | 2.8 |

[a] The model implementations and analyses were performed using Matlab on a system with 2.80 GHz Intel Core 2 Duo CPU.

[b] The mean absolute error (MAE) is calculated by finding the average of the absolute surface displacement errors over the model surface as:

$$MAE = Average_{model}(\text{—}(Model\ Displacement)\ \text{-}\ (GT\ Displacement)\text{—}).$$

[c] The percent error (PE) is calculated by finding the ratio of MAE to the average of the absolute GT displacements as:

$$PE = 100 \times MAE\ /\ (Average_{model}(\text{—}GT\ Displacement\text{—})).$$

the other hand, S-cube model (with 56 elements) and doubly refined S-cube model (with 224 elements) generated 1.1 mm and 0.8 mm mean absolute errors respectively. Similar error reduction rates were also observed for the LV, L-bar and E-torus models.

The proposed scheme offers a convenient trade-off between the simulation precision and speed. As an example, let's look into the analysis of L-bar model. The high resolution reference model for L-bar consisted of 1410 elements, and its stiffness matrix was computed in 209.9 seconds (see Table 8.1, L-bar and GT column). The corresponding refined Doo-Sabin model (DS1) included only 88 elements, and its stiffness matrix was computed in 20.8 seconds (see Table 8.1, L-bar and DS1 column). During the simulation scenario, in which the surface forces were introduced on the model (see Figure 8.6, L-bar row, column-D), DS1 model generated 0.4 mm mean absolute error and 1.9 percent error (See Figure 8.7). This precision level might be adequate for early and intermediate stages of the design / analysis loop. Otherwise, the Doo-Sabin basis function can be further refined to increase the simulation accuracy with the cost of higher computation times.

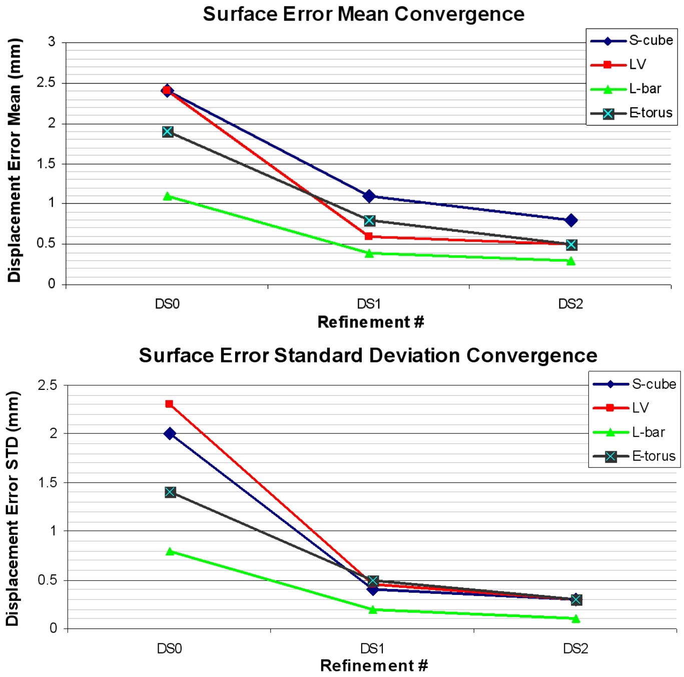


Figure 8.7: The mean (top) and the standard deviation (bottom) of the absolute surface displacement error for the non-refined, refined and doubly refined Doo-Sabin models.

## 8.6 Conclusions and future work

We have introduced a unified representation approach for the models generated using Doo-Sabin subdivision surfaces. The proposed method (1) avoids the need for intermediate meshing tools, and (2) offers an intuitive control for the trade-off between the simulation-speed and precision. The experiments show that the physical simulation accuracy of the introduced models converges rapidly to high resolution finite element models, which use classical hexahedron and triangular prism elements.

A-I-J element may experience *locking* problems, where the element behave extra stiff, particularly when the shell is very thin [33]. Several methods including the reduced [35] or cross-reduced [36] integration can be deployed to avoid locking problems. Locking-free shell elements for this framework might be investigated in a future study.

# Appendix A: Doo-Sabin subdivision matrix

The subdivision weights used for faces consisting of $c$ vertices are used as defined by Doo & Sabin [9]:

$$w_{i,j}^{(c)} = \begin{cases} (c+5)/4c & if\, i = j \\ (3 + 2\cos(2\pi(i-j)/c))/4c & if\, i \neq j \end{cases} \tag{8.20}$$

Subdivision of the control vertices within a single face can be expressed as a linear operation using a subdivision matrix $\mathbf{S}_c$:

$$\mathbf{S}_c = \begin{bmatrix} w_{1,1}^{(c)} & w_{1,2}^{(c)} & \cdots & w_{1,c}^{(c)} \\ w_{2,1}^{(c)} & w_{2,2}^{(c)} & \cdots & w_{2,c}^{(c)} \\ \cdots & \cdots & \cdots & \cdots \\ w_{c,1}^{(c)} & w_{c,2}^{(c)} & \cdots & w_{c,c}^{(c)} \end{bmatrix}. \tag{8.21}$$

Subdivision of whole patches is accomplished by combining $\mathbf{S}_c$ 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.

# Appendix 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 $\xi$ and $\eta$ $(i = 0, \ldots, 8)$:

$$\tilde{b}_i(\xi, \eta) = p_{i\%3}(\xi)\, p_{i/3}(\eta), \tag{8.22}$$

where "%" and "/" denotes the division remainder and division operators respectively. $p_i(y)$ are the basis polynomials for quadratic B-splines with uniform knot vectors:

$$2p_0(y) = 1 - 2y + y^2 \tag{8.23}$$

$$2p_1(y) = 1 + 2y - 2y^2 \tag{8.24}$$

$$2p_2(y) = y^2 \tag{8.25}$$

# References

[1] P. Kagan and A. Fischer, "Integrated mechanically based cae system using b-spline finite elements," *Computer-Aided Design*, vol. 32, no. 8-9, pp. 539 – 552, 2000.

[2] H. Qin, "Physics-based modeling framework for graphics, computer-aided design, and visualization," *Framework*, 2008.

[3] E. Cohen, T. Martin, R. M. Kirby, T. Lyche, and R. F. Riesenfeld, "Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5-8, pp. 334 – 356, 2010. Computational Geometry and Analysis.

[4] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39-41, pp. 4135 – 4195, 2005.

[5] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder, "Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision," *Computer-Aided Design*, vol. 34, no. 2, pp. 137 – 148, 2002.

[6] E. Dikici and F. Orderud, "Polynomial regression based edge filtering for left ventricle tracking in 3d echocardiography," in *STACOM*, pp. 168–177, 2011.

[7] F. Orderud, G. Kiss, and H. Torp, "Automatic coupled segmentation of endo- and epicardial borders in 3d echocardiography," in *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, pp. 1749 –1752, nov. 2008.

[8] A. Wakita, T. Hayashi, T. Kanai, and H. Chiyokura, "Using lattice for web-based medical applications.," in *Web3D*, pp. 29–34, 2001.

[9] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978.

[10] F. Cheng, F. Fan, C. Huang, J. Wang, S. Lai, and K. T. Miura, "Chapter 5: Smooth surface reconstruction using doo-sabin subdivision surfaces," in *Geometric Modeling and Imaging, 2008. GMAI 2008. 3rd International Conference on*, pp. 27 –33, 2008.

[11] I. Boier-Martin, D. Zorin, and F. Bernardini, “A survey of subdivision-based tools for surface modeling,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 0000, pp. 1–27, 2005.

[12] J. Warren and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st ed., 2002.

[13] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350 – 355, 1978.

[14] D. Zorin, P. Schröder, T. Derose, L. Kobbelt, A. Levin, and W. Sweldens, “Subdivision for Modeling and Animation,” in *SIGGRAPH Course Notes*, (New York), ACM, 2000.

[15] L. Kobbelt, “sqrt(3)-subdivision,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’00, (New York, NY, USA), pp. 103–112, ACM Press/Addison-Wesley Publishing Co., 2000.

[16] J. Peters and U. Reif, “The simplest subdivision scheme for smoothing polyhedra,” *ACM Trans. Graph.*, vol. 16, pp. 420–431, October 1997.

[17] C. Loop, “Smooth Subdivision Surfaces Based on Triangles,” department of mathematics, University of Utah, Utah, USA, Aug. 1987.

[18] L. Kobbelt, “Interpolatory subdivision on open quadrilateral nets with arbitrary topology,” *Computer Graphics Forum*, vol. 15, no. 3, pp. 409–420, 1996.

[19] N. Dyn, D. Levin, and J. A. Gregory, “A butterfly subdivision scheme for surface interpolation with tension control,” *ACM Transactions on Graphics*, vol. 9, pp. 160–169, 1990.

[20] K. Wang, Y. He, X. Guo, and H. Qin, “Spline thin-shell simulation of manifold surfaces.,” in *Computer Graphics International’06*, pp. 570–577, 2006.

[21] X. Gu, Y. He, and H. Qin, “Manifold splines,” in *Graphical Models*, pp. 27–38, 2005.

[22] D. J. Benson, Y. Bazilevs, M. Hsu, and T. J. R. Hughes, “Isogeometric shell analysis: The reissner-mindlin shell,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5-8, pp. 276 – 289, 2010. Computational Geometry and Analysis.

[23] J. Kiendl, K. U. Bletzinger, J. Linhard, and R. Wüchner, “Isogeometric shell analysis with kirchhoff-love elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 49-52, pp. 3902 – 3914, 2009.

[24] F. Cirak and M. Ortiz, “Fully c1-conforming subdivision elements for finite deformation thin-shell analysis,” *International Journal for Numerical Methods in Engineering*, vol. 51, no. 7, pp. 813–833, 2001.

[25] F. Cirak, M. Ortiz, and P. Schröder, "Subdivision surfaces: A new paradigm for thin-shell finite-element analysis," *International Journal for Numerical Methods in Engineering*, vol. 47, pp. 2039–2072, 2000.

[26] D. Burkhart, B. Hamann, and G. Umlauf, "Iso-geometric analysis based on catmull-clark solid subdivision," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1575–1784, 2010.

[27] J. Stam, "Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, (New York, NY, USA), pp. 395–404, ACM, 1998.

[28] E. Hinton and D. R. J. Owen, *An introduction to finite element computations / E. Hinton, D. R. J. Owen.* Pineridge Press, Swansea [Wales] :, 1979.

[29] I. C. Taig, "Structural analysis by the matrix displacement method," tech. rep., English Electric Aviation No. 5017, 1961.

[30] H. T. Y. Yang, S. Saigal, A. Masud, and R. K. Kapania, "A survey of recent shell finite elements," *International Journal for Numerical Methods in Engineering*, vol. 47, no. 1-3, pp. 101–127, 2000.

[31] S. Ahmad, B. M. Irons, and O. C. Zienkiewicz, "Analysis of thick and thin shell structures by curved finite elements," *International Journal for Numerical Methods in Engineering*, vol. 2, no. 3, pp. 419–451, 1970.

[32] R. D. Mindlin, "Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates," *Journal of Applied Mechanics*, vol. 18, pp. 31–38, 1951.

[33] S. S. Bhavikatti, *Finite Element Analysis.* Principal, Vijayanagar Engineering College, Bellary, Karnataka: New Age International, 2005.

[34] G. Dhond, *The Finite Element Method for Three-Dimensional Thermomechanical Applications.* John Wiley & Sons Ltd., 2004.

[35] O. C. Zienkiewicz, R. L. Taylor, and J. M. Too, "Reduced integration technique in general analysis of plates and shells," *International Journal for Numerical Methods in Engineering*, vol. 3, no. 2, pp. 275–290, 1971.

[36] Z. Zhong, "A bilinear shell element with the cross-reduced integration technique," *International Journal for Numerical Methods in Engineering*, vol. 36, no. 4, pp. 611–625, 1993.

# Chapter 9

# Doo-Sabin Surface Models with Biomechanical Constraints for Kalman Filter Based Endocardial Wall Tracking in 3D+T Echocardiography

Engin Dikici[1], Fredrik Orderud[2], Gabriel Kiss[1], Anders Thorstensen[1] and Hans Torp[1]
[1]Norwegian University of Science and Technology, Trondheim, Norway
[2]GE Vingmed Ultrasound, Oslo, Norway

In this paper, a 3D left ventricle (LV) tracking framework utilizing Doo-Sabin subdivision surface models is extended with biomechanically constrained state transitions. First, an isoparametric finite element analysis (FEA) method for Doo-Sabin surface models is provided. The isoparametric FEA produces a stiffness matrix for a given endocardial model directly, eliminating inconvenient remodeling/meshing procedures commonly conducted prior to FEA. The computed model might lead to inaccurate deformation modes during the tracking due to hypothesized model shape and FEA parameters. Accordingly, we introduce a statistical model improvement approach for modifying the model shape and its stiffness matrix using experimentally observed endocardial surface variations. Finally, the state prediction stage of the Kalman tracking framework is formulated to perform constrained tracking. Comparative analyses show that the biomechanical constraints can significantly improve the endocardium tracking accuracy of the models with high control node resolutions.

## 9.1 Introduction

This paper considers the problem of endocardial border tracking in 3D+T echocardiography recordings. This is a challenging task due to reasons including speckle noise, shadowing, and the existence of intra-cavity structures (e.g. chordae tendineae, papillary muscles and valves) [1]. Furthermore, real-time detection of endocardial borders might be desirable during invasive procedures and intensive care applications.

State-space analysis using Kalman filtering can be employed for the detection of left ventricle (LV) structures in time-dependent recordings. A Kalman filtering framework for tracking B-spline models was first introduced by Blake et al. [2]. This framework was later utilized for rapid tracking of LV in long-axis 2D echocardiography in [3, 4]. Orderud et al. extended the approach with the use of Doo-Sabin subdivision models for real-time tracking of 3D echocardiography recordings [5]. These studies took advantage of compact model representations for rapid tracking, but did not utilize physical properties to constrain model deformations. Liu et al. introduced a biomechanical-model constrained state-space analysis framework for the tracking of short-axis 2D echocardiography recordings [6]. Their study used *dense* Delaunay triangulated models and employed basic tri-nodal linear elements during the finite element analysis (FEA). Due to the triangulated high resolution model representations, it offered a computationally expensive solution.

The motivation for our study is to combine the compact model representations with biomechanical constraints for rapid and accurate tracking. To our knowledge, no work has been published on the biomechanically constrained tracking of subdivision surfaces using a Kalman filter. Accordingly, we extend the real-time Kalman tracking framework defined in [5] by employing biomechanically constrained state transitions. First, the isoparametric FEA method for Doo-Sabin surface models [7] is briefly described. This step enables the computation of a stiffness matrix for a given Doo-Sabin endocardial model using shell elements without changing the model geometry. It also eliminates the inconvenient remodeling/meshing procedures commonly conducted prior to FEA. However, the computed model might lead to inaccurate deformation modes due to hypothesized model shape and FEA parameters (e.g. Young's modulus, Poisson's ratio). The statistical model improvement stage addresses this problem by employing Control Point Distribution Models (CPDM) [8] and Baruch and Bar-Itzhack direct matrix modifications (BBDMM) [9]. It generates a more accurate model shape, and restricts model's deformation modes using experimentally observed endocardial surface variations. To compute CPDM, we introduce a regressive conversion method from ground-truth endocardial surface meshes to Doo-Sabin surface representations. The mean endocardial surface model and its modes of deformation are found using Doo-Sabin surface representations that are converted from a training dataset. The mean shape is utilized as the new model shape, and the modes of deformation are used for modifying the model stiffness matrix in BBDMM method. Finally, the state prediction stage of the Kalman tracking framework is formulated to perform biomechanically constrained tracking. In the results section, endocardial surface tracking quality is compared among (1) Doo-Sabin surface models with different control node resolutions, (2) biomechanically constrained and non-constrained state transitions, and (3) the systems employing statistically improved and not improved Doo-Sabin models.

## 9.2 Isoparametric Formulation of Doo-Sabin Surface Models

The tracking framework introduced in [5] is built around a deformable Doo-Sabin subdivision model parametrized by a set of control nodes with associated displacement direction vectors. For the FEA of Doo-Sabin endocardial models, we propose to use an isoparametric method from [7] as it (1) eliminates the need for meshing tools by employing a unified geometric representation for the design and analysis, and (2) generates a stiffness matrix that is directly defined for the control node displacements of the model.

A Doo-Sabin surface is a type of a subdivision surface that generalizes bi-quadric B-spline patches to an arbitrary topology [10]. It is defined as the limit surface of a recursive subdivision process. Each limit surface position of a patch is uniquely described using *parametric* patch coordinates $(\xi, \eta)$ that vary between 0 and 1. The basis functions map patch coordinates to *physical* coordinates by,

$$\mathbf{y} = \sum_{i=1}^{n} b_i\left(\xi, \eta\right) \mathbf{q_i}, \tag{9.1}$$

where (1) $n$ gives the total number of control nodes for a given patch, (2) $b_i$ is the basis function for the $i^{th}$ control node of the patch, (3) $\mathbf{q}_i$ gives the physical coordinates of the $i^{th}$ control node, and (4) $\mathbf{y}$ holds the mapped physical coordinates. Please refer to [5] for the derivation of the basis functions.

The basis functions can also be utilized for defining isoparametric solid shell elements during the FEA [7]. The degenerated solids based approach, proposed in [11], can be employed for the shell elements with shape functions:

$$\mathbf{y} = \sum_{i=1}^{n} b_i\left(\xi, \eta\right) \left\{ \mathbf{q}_i + t\frac{\zeta}{2}\mathbf{v}_{3i} \right\}, \tag{9.2}$$

where (1) $[\mathbf{v}_{1i}, \mathbf{v}_{2i}, \mathbf{v}_{3i}]$ defines orthogonal axes at the $i^{th}$ control node of the patch, in which $\mathbf{v}_{3i}$ gives the surface normal direction, (2) $t$ gives the shell thickness, and (3) a parametric patch coordinate $\zeta$ traverses on the surface normal direction. Hence, each surface patch is represented by a shell element with a thickness. The displacement vector $\mathbf{u}$ of any point in the element is given by,

$$\mathbf{u} = \sum_{i=1}^{n} b_i\left(\xi, \eta\right) \left\{ \mathbf{u_i} + t\frac{\zeta}{2}\left[\mathbf{v}_{1i}, -\mathbf{v}_{2i}\right] \left[ \begin{array}{c} \alpha_i \\ \beta_i \end{array} \right] \right\}, \tag{9.3}$$

where (1) $\mathbf{u_i} = \left[u_i,\, v_i,\, w_i\right]^T$ is the displacement vector for the $i^{th}$ control node, and (2) $\alpha_i$ and $\beta_i$ are the scalar rotations in radians around $\mathbf{v}_{1i}$ and $\mathbf{v}_{2i}$ axes respectively.

During the FEA of the shell structures, a *local* coordinate system ($x'$, $y'$, $z'$) of the element needs to be determined. By applying a linear interpolation, an orthogonal set of local coordinate axes for any point in the element are given by,

$$\mathbf{v}_3(\xi,\eta) = \frac{\sum_{i=1}^{n} b_i(\xi,\eta)\,\mathbf{v}_{3i}}{\left|\sum_{i=1}^{n} b_i(\xi,\eta)\,\mathbf{v}_{3i}\right|}, \quad \mathbf{v}_1(\xi,\eta) = \frac{\mathbf{h}\times\mathbf{v}_3}{|\mathbf{h}\times\mathbf{v}_3|}, \quad \mathbf{v}_2(\xi,\eta) = \mathbf{v}_3\times\mathbf{v}_1, \tag{9.4}$$

with $\mathbf{h}$ giving an arbitrary vector that satisfies $\mathbf{h}\neq\mathbf{v}_3(\xi,\eta)$.

The strain vector $\varepsilon = [\varepsilon_{x'},\ \varepsilon_{y'},\ \gamma_{x'y'},\ \gamma_{x'z'},\ \gamma_{y'z'}]^T$ is described by the first partial derivatives of the local displacement vector $\mathbf{u'} = [u',\ v',\ w']^T$ as

$$\varepsilon = \left[\ \frac{\partial u'}{\partial x'},\quad \frac{\partial v'}{\partial y'},\quad \frac{\partial u'}{\partial y'}+\frac{\partial v'}{\partial x'},\quad \frac{\partial u'}{\partial z'}+\frac{\partial w'}{\partial x'},\quad \frac{\partial v'}{\partial z'}+\frac{\partial w'}{\partial y'}\ \right]^T, \tag{9.5}$$

where (1) $\varepsilon_{x'}$ and $\varepsilon_{y'}$ are the normal strains in $x'$ and $y'$ directions, and (2) $\gamma_{x'y'}$, $\gamma_{x'z'}$ and $\gamma_{y'z'}$ are the shear strains in the $x'y'$, $x'z'$ and $y'z'$ planes respectively. The relation between the strain and stress *($\sigma$)* vectors can be formed using *Generalized Hooke's Law* as $\sigma = \mathbf{D}\varepsilon$, in which the material matrix $\mathbf{D}$ is defined using Young's modulus and Poisson's ratio [11].

Using a variational formulation, the stiffness matrix of an element is given by:

$$\mathbf{K}_e = \iiint \mathbf{B}^T\mathbf{D}\mathbf{B}\,dxdydz, \tag{9.6}$$

where $\mathbf{B}$ *(strain-displacement matrix)* relates the strains to the control node displacements *($\delta$)* using $\varepsilon = \mathbf{B}\delta$. To compute $\mathbf{B}$, firstly the components of $\varepsilon$ are found as described in [7]. Next, $\mathbf{B}$ can be found solving $\varepsilon = \mathbf{B}[\delta_1,\delta_2,\ldots\delta_n]^T$ where $\delta_i = [u_i,\ v_i,\ w_i,\ \alpha_i,\ \beta_i]^T$. The infinitesimal volume computed in physical coordinates can be expressed in terms of parametric coordinates as $dxdydz = |\mathbf{J}|\,d\xi d\eta d\zeta$, in which $3\times 3$ Jacobian matrix $\mathbf{J}$ gives the first-order partial derivatives of physical coordinates with respect to parametric coordinates. Finally, $\iiint \mathbf{B}^T\mathbf{D}\mathbf{B}\,|\mathbf{J}|\,d\xi d\eta d\zeta$ can be numerically estimated using Gauss Legendre quadrature rules. Using two samples in $\zeta$ direction and minimum four samples in both $\xi$ and $\eta$ directions is sufficient for thin shell element stiffness matrix calculations [11]. Computed element stiffness matrices can be assembled into a model stiffness matrix $\mathbf{K}$ following a standard procedure.

The resolution of a given Doo-Sabin surface model can be adjusted by basis refinements, without changing the model geometry or its parametrization (See Figure 9.1). A Doo-Sabin surface refinement produces a model with a higher number of elements, which provides an increased physical simulation accuracy during FEA. The convergence properties of Doo-Sabin shell elements are reported in [7].

## 9.3 Statistical Model Improvements

An endocardial model designed using hypothetical shape and material properties (e.g. Young's modulus and Poisson's ratio) might lead to poor tracking accuracy. Yet, the

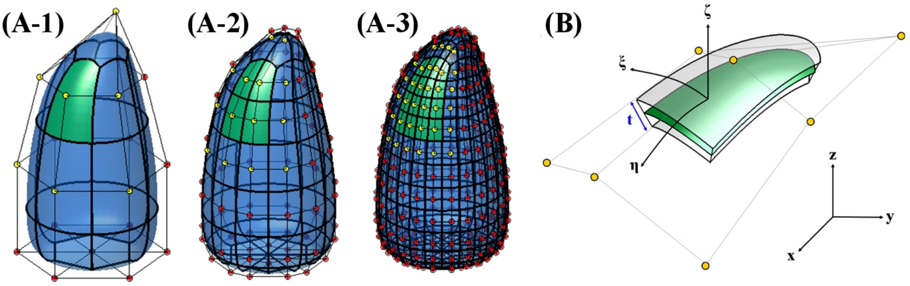


Figure 9.1: (A) The limit surface of an arbitrary patch and its subdivisions are shown in green, the control nodes for the green patches are shown in yellow. Subdivided patches define the exact same limit surface as the original patch. (B) The geometry of a shell element for the green patch from (A-1) is shown; each patch is modeled by a shell element.

identification of the optimal properties requires a considerable amount of user input and time, which might not be manageable for a complex structure such as the LV [12]. We propose to use a statistical model improvement method that (1) learns the mean model shape and its deformations using a training dataset, and (2) updates the model shape and the stiffness matrix directly using the learned information.

For producing a statistical endocardial Doo-Sabin surface model, firstly a conversion procedure between 3D ground-truth endocardial surface segmentations and Doo-Sabin surface representations needs to be described. We propose a regressive approach for this task.

A dense set of landmark points ($m \gg n$), which are evenly spread around a patch, is given by $L = \{(\xi_1, \eta_1), (\xi_2, \eta_2), \ldots, (\xi_m, \eta_m)\}$. The tracker defined in [5], referred as the *classical tracker*, maps $L$ on to a set of Cartesian coordinates as $(\xi_i, \eta_i) \rightarrow \mathbf{y_i}^{(\rho,\tau)}$, where (1) $\rho \in \{1, 2, \ldots r\}$ identifies a recording, and (2) $\tau \in [0 : \text{end-systole}, 1 : \text{end-diastole}]$ gives the temporal cardiac cycle position. Each $\mathbf{y_i}^{(\rho,\tau)}$ can be updated using a normal displacement *(in* $\mathbf{v}_3$ *direction from Equations 9.4)* to a ground-truth surface point $\tilde{\mathbf{y}}_i^{(b,\tau)}$. Next, $m$ equations that relate unknown patch control node positions, $\mathbf{q}^{(\rho,\tau)}$, and the ground-truth surface points can be defined as,

$$\tilde{\mathbf{y}}_i^{(\rho,\tau)} = \sum_{j=1}^{n} b_j (\xi_i, \eta_i)\, \mathbf{q}_j^{(\rho,\tau)}, \quad 1 \leq i \leq m. \tag{9.7}$$

The equations collected from all surface patches can be put into a matrix form:

$$\tilde{\mathbf{Y}}^{(\rho,\tau)} = \mathbf{F}\mathbf{Q}^{(\rho,\tau)} + \mathbf{W}, \tag{9.8}$$

where (1) $\mathbf{F}$ is a design matrix holding the basis function values, (2) $\mathbf{Q}^{(\rho,\tau)}$ is a model control node matrix, and (3) $\mathbf{W}$ is a mapping error matrix. Assuming white mapping errors, the maximum likelihood estimator (MLE) for $\mathbf{Q}^{(\rho,\tau)}$ is found by $\hat{\mathbf{Q}}^{(\rho,\tau)} =$

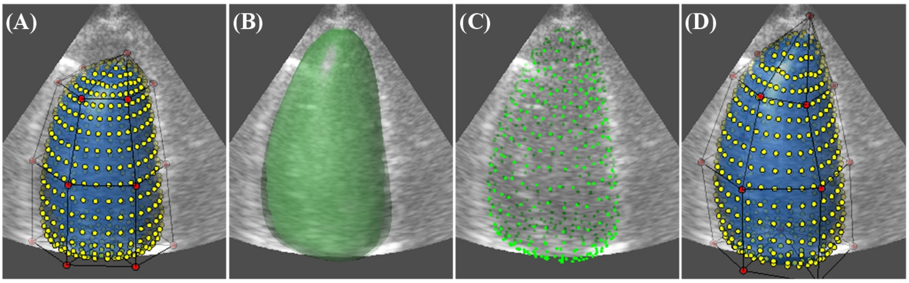


Figure 9.2: (A) Doo-Sabin surface representation produced by the classical tracker with landmark (yellow) and control (red) points, which does not fit at apical region, (B) ground-truth surface segmentation, (C) ground-truth landmark points, and (D) updated Doo-Sabin surface representation using MLE based conversion are represented.

$\left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\tilde{\mathbf{Y}}^{(\rho,\tau)}$ [13]. A Doo-Sabin surface representation with the control nodes given by $\hat{\mathbf{Q}}^{(\rho,\tau)}$ fits onto the ground-truth endocardial surface segmentation for the recording $\rho$ at $\tau$ (see Figure 9.2).

After estimating Doo-Sabin surface representations for all training recordings at each cardiac cycle position, CPDM can be computed. First, the surface representations are aligned utilizing the global state information of the classical tracker, which keeps the model translation and rotation with respect to an initial model state [5]. Next, the mean model representation is computed using the aligned models; $\bar{\mathbf{Q}}$ gives the mean model's control nodes. Finally, the control node covariance matrix $\mathbf{V}$ about $\bar{\mathbf{Q}}$ is found. The $t$ eigenvectors of $\mathbf{V}$, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t]$, corresponding to the largest $t$ eigenvalues $\Omega^2_{pdm} = diag\left(\omega^2_1, \omega^2_2, \ldots, \omega^2_t\right)$, give the observed modes of variation (or deformation) for the model [8].

The Doo-Sabin model employed in the classical tracker can be improved using (1) the mean model representation, and (2) the observed modes of deformation. First, the model control nodes are modified as the mean model control nodes $\bar{\mathbf{Q}}$, hence the modes of deformation of the observed and computed models are comparable. Then, the model stiffness matrix for the modified model is found using the isoparametric formulation introduced in the previous section as $\mathbf{K}^{'}$. *Modal Analysis* can be performed to find deformation modes of the modified model by solving,

$$\mathbf{K}^{'}\Phi_{fem} = \mathbf{M}\Phi_{fem}\Omega^2_{fem}, \tag{9.9}$$

where (1) $\mathbf{M}$ is a mass matrix assumed as an identity matrix in our study, (2) $\Phi_{fem}$ is the eigenvector matrix of $\mathbf{M}^{-1}\mathbf{K}^{'}$ and represents the deformation modes, and (3) $\Omega^2_{fem}$ is a diagonal matrix of the associated eigenvalues. It is desirable to have similar deformation modes as the observed ones given by CPDM.

The BBDMM is a direct matrix modification method [9] that makes minimal amount of modifications on the original stiffness matrix to produce desired deformation modes. In this study, it is utilized to make minimal modifications to $\mathbf{K}'$ by solving,

$$\begin{aligned} &min_{\mathbf{K}_{opt}} \left\| \mathbf{M}^{-1/2} \left( \mathbf{K}_{opt} - \mathbf{K}^{'} \right) \mathbf{M}^{-1/2} \right\|_F , \\ &\qquad\qquad s.t. \quad \mathbf{K}_{opt}\mathbf{S} = \mathbf{M}\mathbf{S}\Omega^2_{pdm}, \qquad \mathbf{K}_{opt} = \mathbf{K}^T_{opt}, \end{aligned} \tag{9.10}$$

where $\|\cdot\|_F$ is the Frobenius norm. With the assumptions of (1) $\mathbf{K}^{'}$ is symmetric, (2) $\mathbf{S}$ and $\mathbf{M}\mathbf{S}\Omega^2_{pdm}$ are both full rank, (3) $\mathbf{S}^T\mathbf{M}\mathbf{S}\Omega^2_{pdm}$ is symmetric and non-singular, and (4) $\mathbf{M}$ is non singular such that $Rank\left(\mathbf{M}^T\mathbf{M}\mathbf{S}\right) = Rank\left(\mathbf{M}\mathbf{S}\Omega^2_{pdm}\right)$, the unique solution to $\mathbf{K}_{opt}$ can be found as,

$$\begin{aligned} \mathbf{K}_{opt} = \mathbf{K}^{'} + \left(\mathbf{Y} - \mathbf{K}^{'}\mathbf{S}\right)\left(\mathbf{Y}^T\mathbf{S}\right)^{-1}\mathbf{Y}^T + \mathbf{Y}\left(\mathbf{S}^T\mathbf{Y}\right)^{-1}\left(\mathbf{Y} - \mathbf{K}^{'}\mathbf{S}\right)^T \\ - \mathbf{Y}\left(\mathbf{Y}^T\mathbf{S}\right)^{-1}\left(\mathbf{Y} - \mathbf{K}^{'}\mathbf{S}\right)^T\mathbf{S}\left(\mathbf{Y}^T\mathbf{S}\right)^{-1}\mathbf{Y}^T, \end{aligned} \tag{9.11}$$

where $\mathbf{Y} = \mathbf{M}\mathbf{S}\Omega^2_{pdm}$ [12], and $\mathbf{K}_{opt}$ is the optimally modified stiffness matrix for the endocardial model. Please note that $\mathbf{S}$ gives the $t$ most prominent deformation modes for $\mathbf{K}_{opt}$: the major deformation modes produced by $\mathbf{K}_{opt}$ are similar with the observed modes learned from a training dataset.

## 9.4 Tracking Framework

The tracking framework represents the shape and pose deformations by a composite transform, where local shape deformations are obtained by moving control nodes in the subdivision model together with a global transformation that translates, rotates and scales the whole model. This leads to a composite state vector $\mathbf{x} = \left[\mathbf{x}_g^T, \mathbf{x}_l^T\right]^T$, consisting of $n_g$ global and $n_l$ local parameters.

The tracking framework consists of five separate stages, namely the (1) state prediction, (2) evaluation of tracking model, (3) edge measurements, (4) measurement assimilation, and (5) measurement update. The biomechanical constraints can be enforced in the state prediction stage [6], which is further elaborated in this section. A more detailed discussion on the other stages of the framework can be found in [5].

The control node displacements of a Doo-Sabin surface model can be formulated under the principle of minimal energy using ordinary differential equation as,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{0}, \tag{9.12}$$

where (1) $\mathbf{u} = \left[u_1, u_2, \ldots, u_p\right]^T$ gives the normal displacements for a model with $p$ control nodes, and (2) $\mathbf{C}$ is a *Rayleigh* damping matrix, which can be found by $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$ with small weighting constants $\alpha$ and $\beta$. Accordingly, a continuous-time linear time-invariant stochastic system can be derived as $\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t)$, where

$$\mathbf{x}[t] = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix}, \quad \mathbf{A}_c = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}. \tag{9.13}$$

This system can be discretized as $\mathbf{x}[k+1] = \mathbf{A}_d \mathbf{x}[k]$, where (1) $\mathbf{x}[k] = \left[\mathbf{u}[k]^T \ \mathbf{v}[k]^T\right]^T$ with $\mathbf{v}[k]$ denoting the velocities of the control nodes at time step $k$, and (2) $\mathbf{A}_d = e^{\mathbf{A}_c \Gamma}$ where $\Gamma = 1$ gives the unit sampling interval used in our study. $\mathbf{A}_d$ can be decomposed as,

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{A}_{00} & \mathbf{A}_{01} \\ \mathbf{A}_{10} & \mathbf{A}_{11} \end{bmatrix}, \tag{9.14}$$

where $\mathbf{A}_{00}$, $\mathbf{A}_{01}$, $\mathbf{A}_{10}$ and $\mathbf{A}_{11}$ are $p \times p$ sub-matrices. Assuming $\mathbf{v}[k] \approx \frac{\mathbf{u}[k]-\mathbf{u}[k-1]}{\Gamma}$, the motion model can be written in terms of the last two successive states as,

$$\mathbf{u}[k+1] = \left[\mathbf{A}_{00} + \Gamma^{-1}\mathbf{A}_{01}\right]\mathbf{u}[k] + \left[-\Gamma^{-1}\mathbf{A}_{01}\right]\mathbf{u}[k-1]. \tag{9.15}$$

In [5], control nodes' normal displacements give the system local states; $\mathbf{x}_l = \mathbf{u}$. Hence, Equation 9.15 can be directly plugged into the joint motion model proposed in [5] by,

$$\bar{\mathbf{x}}[k+1] = \begin{bmatrix} \mathbf{R}_g & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_l\left(\mathbf{A}_{00} + \Gamma^{-1}\mathbf{A}_{01}\right) \end{bmatrix} \hat{\mathbf{x}}[k] + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_l\left(-\Gamma^{-1}\mathbf{A}_{01}\right) \end{bmatrix} \hat{\mathbf{x}}[k-1], \tag{9.16}$$

where (1) $\bar{\mathbf{x}}[k+1]$ is the predicted state for the time step $k$+1, (2) $\hat{\mathbf{x}}[k]$ is the estimated state from the time step $k$, and (3) $\mathbf{R}_g$ and $\mathbf{R}_l$ are the regularization matrices for the global and local state parameters respectively.

## 9.5 Results

3D echocardiography was performed on 10 healthy subjects and 19 subjects with recent first time myocardial infarction, using a Vivid 7 (26 recordings) or a Vivid E9 (3 recordings) ultrasound scanner (GE Vingmed Ultrasound, Norway) with a matrix array (3V) transducer. The endocardial border segmentation of the recordings was performed by a trained medical doctor using a semi-automatic segmentation tool (4D AutoLVQ, GE Vingmed Ultrasound, Norway).

For the initial FEA, the shell thickness, Young's modulus and Poisson's ratio were set as $8mm$, $75000Pa$ and 0.47 as proposed in [6]. For the CPDM based model updates, a set of 441 landmark points evenly spread around the model was used. The observed modes of deformation were represented using five eigenvectors ($t = 5$) corresponding to the five largest eigenvalues, which described $\geq 90\%$ of the experimentally observed endocardial shape variations. The Rayleigh damping constants of the tracker were set as $\alpha = \beta = 0.1$.

The Kalman tracking framework was evaluated for three different configurations:

1. The system with no biomechanical constraints *(noBC)*: it tracks a Doo-Sabin surface model using a tracker with no biomechanical constraints. This setup is identical with the classical tracker [5].

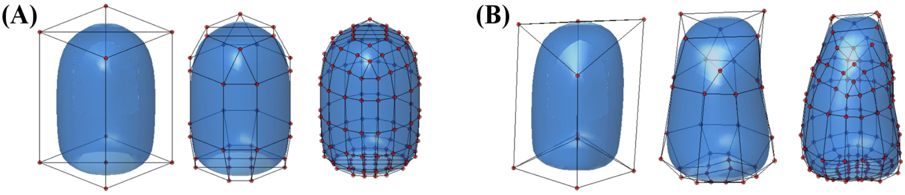


Figure 9.3: Not-refined, refined and double-refined Doo-Sabin surface models (A) for the *noBC* and *BCnoI* , and (B) *BCwithI* setups are represented. For the *BCwithI*, control node positions are updated using the mean model computed by CPDM.

2. The system with biomechanical constraints and no statistical model improvements *(BCnoI)*: it tracks a Doo-Sabin surface model using a tracker with biomechanically constrained state transitions, as described in Section 9.4. The model's stiffness matrix was computed using the isoparametric FEA method from Section 9.2.

3. The system with biomechanical constraints and statistical model improvements *(BCwithI)*: it tracks an improved Doo-Sabin surface model using a tracker with biomechanically constrained state transitions. The improved model has a modified shape and a stiffness matrix as described in Section 9.3.

Each configuration was executed with not-refined (9 nodes), refined (34 nodes) and double-refined (136 nodes) Doo-Sabin surface models represented in Figure 9.3. *noBC* and *BCnoI* setups were tested using all 29 recordings directly. As *BCwithI* requires a training with a pre-segmented dataset, it was tested via leave-one-out cross-validation [14]. The error measurements including the (a) *absolute surface point error* giving the average absolute distance of each predicted surface point to ground-truth surface, (b) *squared surface point error* giving the average squared distance of each predicted surface point to ground-truth surface, and (c) *absolute volume error* giving the average of predicted surface's absolute volume errors are given in Figure 9.4. *Signed surface error polar plots*, showing the average signed distances between the predicted and ground-truth surfaces using 17-segment model of the American Heart Association [15], are represented in Figure 9.5. There were no significant differences between the segmentation performances for the subjects with and without infarction; therefore, the evaluation results are reported for all recordings without any further classification.

The tracking framework is implemented in C++, and processes each frame in $2ms$ with not-refined, $3.4ms$ with refined and $30.6ms$ with double-refined models when executed on a 2.80 GHz Intel Core 2 Duo CPU. There are no execution time differences between the *noBC*, *BCnoI* and *BCwithI* setups as the stiffness and state transition matrices are computed only once for each model.

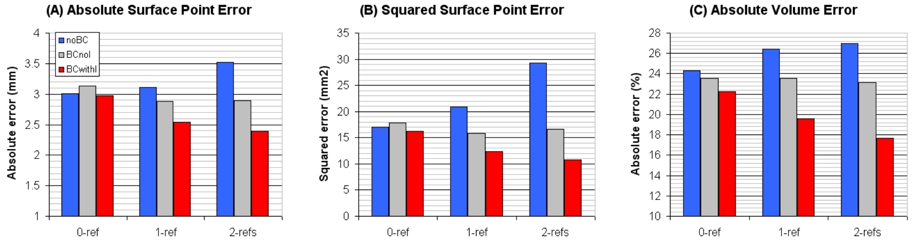


Figure 9.4: (A) Absolute surface point error (in $mm$) , (B) squared surface point error (in $mm^2$), and (C) absolute volume error (in percentages) for the Kalman tracking framework with *noBC*, *BCnoI*, and *BCwithI* setups for non-refined, refined and double-refined Doo-Sabin model tracking.

## 9.6 Discussion and Conclusion

In this study, we extended the real-time Doo-Sabin surface models based Kalman tracking framework with biomechanical constraints. The introduced method is (1) practical; the computed models can be directly used in a Kalman tracking framework by implementing a few modifications in the state prediction stage, (2) useful since it improves the tracking accuracy without introducing additional run-time complexity, (3) yet novel as the biomechanically constrained subdivision surfaces have not been employed in a Kalman tracker prior to our study.

Our analyses showed that the biomechanical constraints are necessary especially when the tracked model has a high control node resolution. This is due to the fact that as the model complexity increases the tracker can benefit more from a spatial regularization, which is provided by biomechanical constraints. Hence, we can observe that the *BCnoI* setup allows tracking quality to be stabilized over model resolution levels (see Figure 9.4 and Figure 9.5 row-2). The statistical model improvements take advantage of higher model resolution levels as (1) the model node updates provide a more realistic model shape to perform tracking, and (2) deformation modes learned from CPDM improve the stiffness matrix accuracy (see Figure 9.4 and Figure 9.5 row-3).

The 17-segment model representations show that the initial tracker produces significant under-estimation error (estimated borders are closer to the object center than the ground-truth borders) at the apex region, the center of the polar plot (see Figure 9.5 row-1). Introducing biomechanical constraints or increasing the model resolution without changing the model shape does not significantly improve the tracking accuracy for the apex region (see Figure 9.5 row-2). The model improvement stage alleviates the problem by modifying the model shape to have a curvier apical part, which is closer to the natural appearance of the LV (see Figure 9.3 (B) and Figure 9.5 row-3).

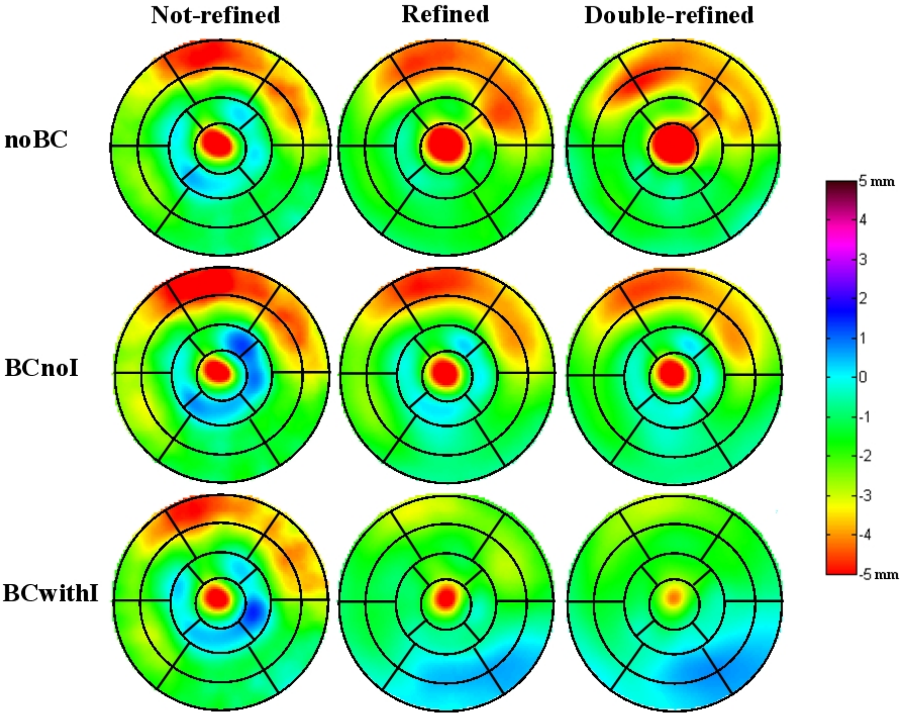


Figure 9.5: 17-segment model representations for the signed surface error: $5mm$ over-estimation is purple, $5mm$ under-estimation is red, $0mm$ no-error is light blue. Rows 1, 2 and 3 show the error plots for the *noBC*, *BCnoI*, and *BCwithI* setups respectively. Columns 1, 2 and 3 show the error plots for the non-refined, refined and double-refined Doo-Sabin model trackers respectively.

# References

[1] S. K. Setarehdan and J. J. Soraghan, *Segmentation in echocardiographic images*, pp. 64–130. New York, NY, USA: Springer-Verlag New York, Inc., 2002.

[2] 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.

[3] G. Jacob, J. A. Noble, M. M. Parada, and A. Blake, "Evaluating a robust contour tracker on echocardiographic sequences," *Medical Image Analysis*, vol. 3, no. 1, pp. 63 – 75, 1999.

[4] G. Jacob, J. Noble, C. Behrenbruch, A. Kelion, and A. 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, pp. 226 –238, march 2002.

[5] F. Orderud and S. I. Rabben, "Real-time 3d segmentation of the left ventricle using deformable subdivision surfaces," in *CVPR*, 2008.

[6] H. Liu and P. Shi, "State-space analysis of cardiac motion with biomechanical constraints," *Image Processing, IEEE Transactions on*, vol. 16, pp. 901–917, Mar. 2007.

[7] E. Dikici, S. R. Snare, and F. Orderud, "Isoparametric finite element analysis for doo-sabin subdivision models," in *Proceedings of Graphics Interface 2012*, GI '12, (Toronto, Ont., Canada, Canada), Canadian Information Processing Society, 2012.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Training models of shape from sets of examples," in *In Proc. British Machine Vision Conference*, pp. 9–18, 1992.

[9] M. Baruch and I. Y. Bar Itzhack, "Optimal weighted orthogonalization of measured modes," *AIAA Journal*, vol. 16, no. 4, pp. 346–351, 1978.

[10] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978.

[11] S. Ahmad, B. M. Irons, and O. C. Zienkiewicz, "Analysis of thick and thin shell structures by curved finite elements," *International Journal for Numerical Methods in Engineering*, vol. 2, no. 3, pp. 419–451, 1970.

[12] C. A. Beattie and S. W. Smith, "Optimal matrix approximants in structural identification," *Journal of Optimization Theory and Applications*, vol. 74, pp. 23–56, 1992. 10.1007/BF00939891.

[13] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer series in statistics, Springer, 2009.

[15] M. D. Cerqueira, N. J. Weissman, V. Dilsizian, *et al.*, "Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association. ," *Circulation*, vol. 105, pp. 539–542, Jan. 2002.