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

Fredrik Orderud
Norwegian University of Science and Technology
Trondheim, Norway
fredrik.orderud@idi.ntnu.no

Stein Inge Rabben
GE Vingmed Ultrasound
Oslo, Norway
stein.rabben@med.ge.com

Abstract

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

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

1. Introduction

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

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

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

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

1.1. Contributions

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

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

1.2. Nomenclature

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

2. Evaluation of Doo-Sabin Surfaces

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

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

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

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

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

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

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

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

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

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

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

Note that this approach is not dependent on diagonalization of the subdivision matrix, as in [10]. Instead, repeated

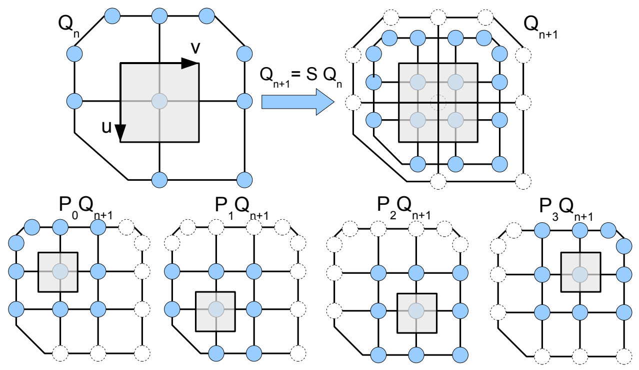


Figure 1. Illustration of the Doo-Sabin subdivision process. The control vertices $\mathbf{Q}_n$ that define the initial surface patch (upper left) are subdivided into new control vertices $\mathbf{Q}_{n+1}$ (upper right) by multiplying $\mathbf{Q}_n$ with the subdivision matrix $\mathbf{S}$. Application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

matrix multiplication performed n times will result in exactly the same result. The associated increase in computational complexity associated with this repeated multiplication will not be a burden if evaluation of basis functions is performed only once, and later re-used to compute surface points regardless of movement of the associated control vertices.

3. Deformable Subdivision Model

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

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

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

3.1. Calculation of Surface Points

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

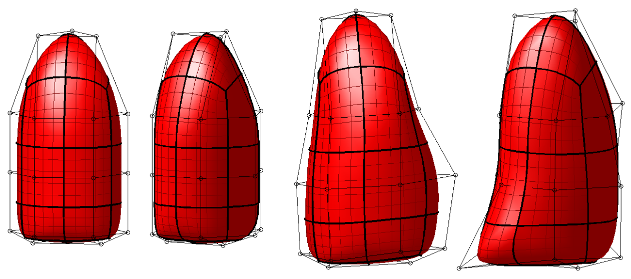
1. Update position of control vertices $\mathbf{q}_i$ based on the state vector: $\mathbf{q}_i = \bar{\mathbf{q}}_i + x_i \mathbf{d}_i$, where $\bar{\mathbf{q}}_i$ is the mean position of the control vertex and x_i is the parameter corresponding to this control vertex in the state vector for each control vertex. $\mathbf{d}_i$ is the displacement direction for control vertex $\mathbf{q}_i$. The full state vector for the model then becomes the concatenation of the state parameters for all control vertices $\mathbf{x}_l = [x_1,\ x_2,\ \dots,\ x_{N_l}]^T$. One can here chose to force certain vertices to remain stationary during tracking without altering the overall approach. This would both reduce the deformation space, as well as the number of parameters to estimate.

2. Calculate surface points $\mathbf{p}_l$ as a sums of control vertices weighted with their respective basis functions within the surface patch of each surface point: $\mathbf{p}_l = \sum_{i \in C(c_l)} \mathbf{b}_i \mathbf{q}_i$.

3. Calculate surface normals $\mathbf{n}_l$ as the cross product between the partial derivatives of the basis functions with regards to parametric values *u* and *v* within the surface patch of each surface point: $\mathbf{n}_l = \sum_{i \in C(c_l)} (\mathbf{b}_u)_i \mathbf{q}_i \times \sum_{i \in C(c_l)} (\mathbf{b}_v)_i \mathbf{q}_i$.

4. Calculate Jacobian matrices for the local deformations $\mathbf{J}_l$ by concatenating the displacement vectors multiplied with their respective basis functions: $\mathbf{J}_l = \left[\ \mathbf{b}_{i_1}\mathbf{d}_{i_1},\ \ \mathbf{b}_{i_2}\mathbf{d}_{i_2},\ \ \dots\ \right]_{i \in C(c_l)}$. The Jacobian matrix will here be padded with zeros for columns corresponding to control vertices outside the region of support for the surface patch of each surface point.

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



(a)

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

3.2. Global Transform

The composite object deformation $\mathbf{T}(\mathbf{x}) = \mathbf{T}_g(\mathbf{T}_l(\mathbf{x}_l),\ \mathbf{x}_g)$ is obtained by combining the local deformations of the subdivision model with a global transform to create a joint model. This leads to a composite state vector $\mathbf{x} = [\mathbf{x}_g^T,\ \mathbf{x}_l^T]^T$ consisting of N_g global and N_l local deformation parameters. This separation between local and global deformations is intended to ease modeling, since changes in shape are often parametrized differently compared to deformations associated with global position, size and orientation.

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

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

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

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

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

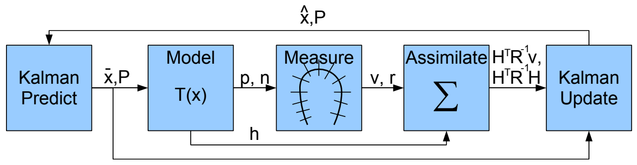


Figure 3. Overview over the processing chain in the Kalman filter based tracking framework. All five steps are performed only once for each new frame.

4. Kalman Tracking Framework

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

4.1. State Prediction

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

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

where $\hat{\mathbf{x}}_k$ is the estimated state from time step k. Tuning of properties like damping and regularization towards the mean state $\mathbf{x}_0$ for all deformation parameters can then be accomplished by adjusting the coefficients in matrices $\mathbf{A}_1$ and $\mathbf{A}_2$. Prediction uncertainty can similarly be adjusted by manipulating the process noise covariance matrix $\mathbf{B}_0$ that is used in the associated covariance update equation for $\bar{\mathbf{P}}_{k+1}$. The latter will then restrict the rate of which parameter values are allowed to vary.

4.2. Evaluation of Deformable Model

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

4.3. Edge Measurements

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

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

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

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

$$\mathbf{h}^T = \mathbf{n}^T \mathbf{J} \ . \tag{11}$$

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

4.4. Measurement Assimilation

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

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

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{v} = \textstyle\sum_i \mathbf{h}_i r_i^{-1} v_i \tag{12}$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} = \textstyle\sum_i \mathbf{h}_i r_i^{-1} \mathbf{h}_i^T \ . \tag{13}$$

4.5. Measurement Update

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

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

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

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

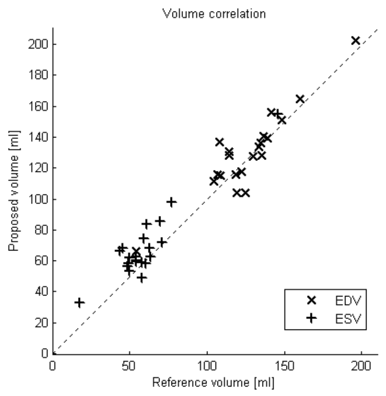


Figure 4. Volume correlation plot for the proposed segmentation against the reference method at end-diastole (EDV) and end-systole (ESV) in each of the 21 recording.

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

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

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

5. Results

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

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

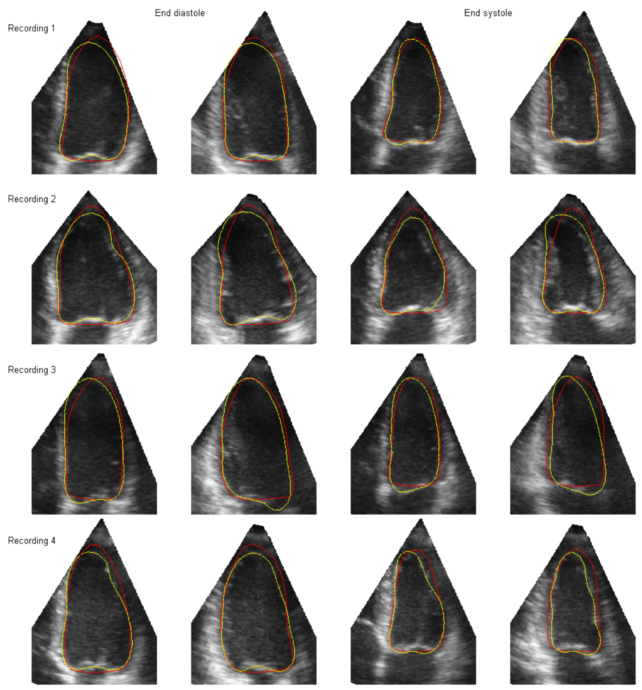


Figure 5. Example segmentation results from four of the recordings (rows). Orthogonal intersection slices through each volume shows the segmentation result at end-diastole (left) and end systole (right). The red intersection lines show the proposed Kalman segmentation, and the yellow lines the reference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

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

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

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

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

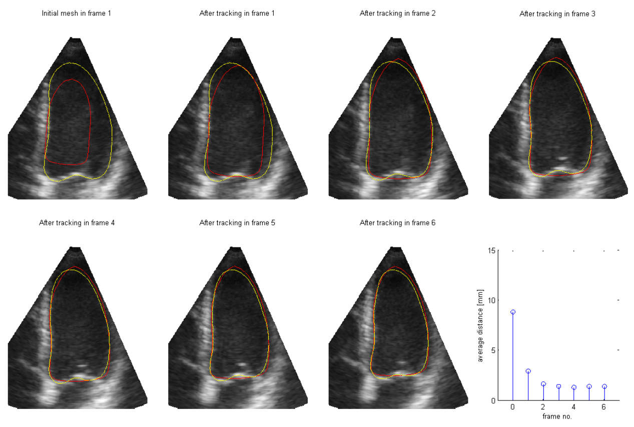

Initial mesh in frame 1
After tracking in frame 1
After tracking in frame 2
After tracking in frame 3
After tracking in frame 4
After tracking in frame 5
After tracking in frame 6
average distance [mm]
frame no.
15
10
5
0
0
2
4
6

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

6. Discussion

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

6.1. Subdivision Model

Usage of subdivision models for segmentation has some desirable properties that makes them suitable for modeling of cardiac structures, in that they combine the inherent smoothness and continuity of surface derivatives of spline surfaces with the support of arbitrary topology from flat polygonal meshes. This enables more flexible modeling, since control vertices of the surface models are no longer restricted to the quadrilateral structure known from spline surfaces. Subdivision models that form closed surfaces and surfaces with complex geometries can therefore be constructed in a simple and intuitive fashion.

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

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

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

6.2. Segmentation Results

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

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

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

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

6.3. Kalman Filter Approach

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

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

Firstly, the framework assumes that the normal displacement values are independent and follow a Gaussian distribution. This, however, only accounts for the normal displacement values only, and not to the shape of the underlying edge profiles or details in the edge detector, in which are not assumed to follow any given distribution in the Kalman filter. Secondly, the extended Kalman filter assumes all deformations to be linear. Except for global rotation, which is inherently non-linear, every other mode of deformation is linear. This includes global translation, scaling and the local shape deformations, which due to the tensor product formulation of the polynomial basis functions are linear in the position of the control vertices. There is very little change in global rotation of the heart between successive frames, so the linearization approximation penalty is believed to be small.

7. Conclusion

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

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

A. Doo-Sabin Subdivision Matrix

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

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

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

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

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

B. Basis functions for Quadratic B-splines

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

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

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

$$2P_0(t) = 1 - 2t + t^2 \tag{19}$$

$$2P_1(t) = 1 + 2t - 2t^2 \tag{20}$$

$$2P_2(t) = t^2 \tag{21}$$

References

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

[2] F. Orderud, J. Hansegård, and S. I. Rabben, "Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, vol. 4791 of *LNCS*, pp. 858–865, Springer, 2007.

[3] J. Hansegård, F. Orderud, and S. Rabben, "Real-time active shape models for segmentation of 3D cardiac ultrasound," in *Computer Analysis of Images and Patterns - CAIP*, pp. 157–164, 2007.

[4] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

[5] G. Jacob, J. A. Noble, C. Behrenbruch, A. D. Kelion, and A. P. Banning, "A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography," *Medical Imaging, IEEE Transactions on*, vol. 21, no. 3, pp. 226–238, 2002.

[6] D. Comaniciu, X. S. Zhou, and S. Krishnan, "Robust real-time myocardial border tracking for echocardiography: An information fusion approach," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 7, pp. 849–860, 2004.

[7] H. Liu and P. Shi, "State-space analysis of cardiac motion with biomechanical constraints," *Image Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 901–917, April 2007.

[8] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, pp. 356–360, Nov. 1978.

[9] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological meshes," *Computer-Aided Design*, vol. 10, pp. 350–355, Nov. 1978.

[10] J. Stam, "Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 395–404, ACM Press, 1998.

[11] A. H. Barr, "Global and local deformations of solid primitives," in *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 21–30, ACM Press, 1984.

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation.* Wiley-Interscience, 2001.

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