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

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

*Norwegian University of Science and Technology (NTNU)

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

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

I. INTRODUCTION

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

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

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

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

II. METHODS

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

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

Both model and transform nodes will typically have parameters that affects either shape or pose. For tracking in a state-estimation framework, these parameters have to be concatenated into a composite state vector. For doing so, we follow a depth-first traversal convention to construct the state vector. This convention works recursively from the root transform, by first appending the transform parameters with parameters from each child model, and then appending recursively with concatenated parameters from each child transform. The example hierarchy in Fig. 2(a) will e.g. result in a state vector on the following form $\mathbf{x} = \begin{bmatrix} \mathbf{x}_g^T & \mathbf{x}_a^T & \mathbf{x}_s^T & \mathbf{x}_b^T \end{bmatrix}^T$.
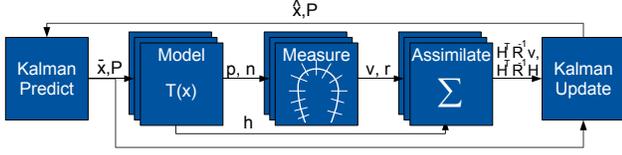
Figure 1. Flowchart over the 5 processing stages in the Kalman tracking framework.

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

A. Deformable Subdivision Models

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

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

B. Evaluation of Surface Points

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

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

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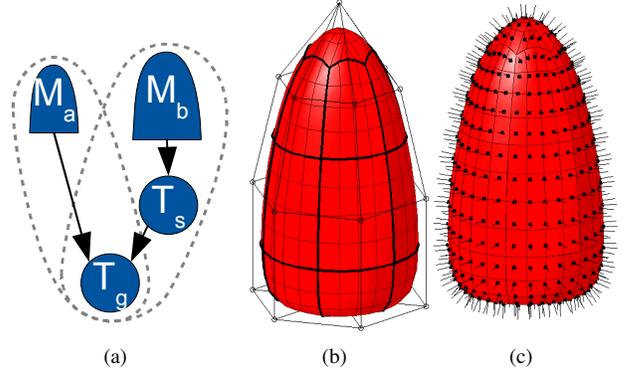
Figure 2. (a) The tracking hierarchy, which consists of two subdivision models (M_a and M_b), connected by a scale transform and a global transform (T_s and T_g). (b-c) Renderings of the subdivision model used for both endo- and epicardium: (b) shows the individual surface patches, enclosed in a wireframe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

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: $\mathbf{J}_l = \left[\begin{array}{ccc} \mathbf{b}_{i_1} \mathbf{d}_{i_1}, & \mathbf{b}_{i_2} \mathbf{d}_{i_2}, & \ldots \end{array} \right]_{i \in C(c_l)}$. 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.

C. Surface Point Transformation

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

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

global transform [2]:

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

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

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

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

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

D. Kalman Tracking Framework

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

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

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

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

Table I
THE CENTER COLUMN SHOWS ESTIMATED MYOCARDIAL VOLUME (MEAN ± STD) FOR EACH RECORDING, BASED ON ENDO- AND EPICARDIAL TRACKING IN ONE CARDIAC CYCLE. THE RIGHT COLUMN SHOWS THE REFERENCE VOLUMES (MEAN ± STD), BASED ON MANUAL TRACING IN ED AND ES.

III. RESULTS

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

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

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

IV. DISCUSSION AND CONCLUSIONS

In this paper, we have presented an extension of an existing tracking framework to support simultaneous segmentation and tracking of several deformable models by means of hierarchical modeling. Coupled endo- and epicardial segmentation in 3D echocardiography served as exemplary application. Results from fully automatic tracking in one simulation as well as 5 recordings of high image quality demonstrates the feasibility

placed by the user. Such attractors can be implemented as virtual edge measurements towards specific spatial positions, indicated by the user in e.g. ED and ES, and would then assist the segmentation in areas of missing or degraded image data.

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

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

REFERENCES

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

(a)

(b)

(c)

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

of the approach for myocardial volume estimation. Results from the simulation showed low volume bias, while the in-vivo results are within the interval of myocardial masses reported by Takeuchi and Pouleur [10], [9].

3D echo recordings of less than optimal image quality often suffers from areas of drop-out and other forms of image degradation that makes accurate edge-detection of especially the epicardium difficult. These difficulties can at least partly be alleviated by incorporating surface attractors, that are manually