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

Fredrik Orderud*, Gabriel Kiss*, Stian Langeland†,
Espen W. Remme‡, Hans G. Torp* and Stein I. Rabben†

*Norwegian University of Science and Technology (NTNU), Norway, †GE Vingmed Ultrasound, Norway,
‡Department of Cardiology, Rikshospitalet, Norway

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

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

I. Introduction

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

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

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

II. Methods

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

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

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

The tracking framework consists of three separate stages, namely temporal prediction, edge-detection update and speckle-tracking update. The latter two can be decomposed into 4 steps each, which results in a total of 9 separate processing steps, as can be seen in Fig. 1. The prediction and edge update steps are similar to [8], and will adapt the model to align it to the endocardial boundary. The second measurement stage then uses speckle-tracking to acquire 3D

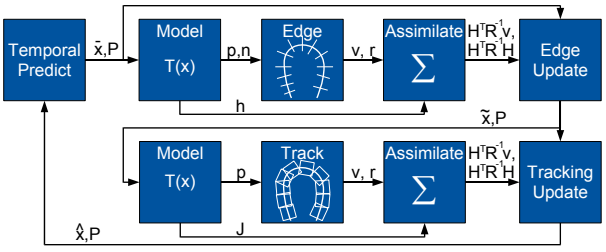


Figure 1. Overview over the processing steps in the Kalman tracking framework.

displacement measurements, and update the model for any residual deformations after the edge-detection. This combined approach is motivated by the inherent problem of drift, due to cumulative buildup of tracking error, associated with speckle-tracking based on sequential block-matching. The proposed approach limits this problem by using edge-detection to ensure that the model remains aligned to the endocardial border at all times during tracking.

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

A. Evaluation of Deformable Model

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

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

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

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

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

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

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

B. Edge or Speckle-tracking Measurements

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

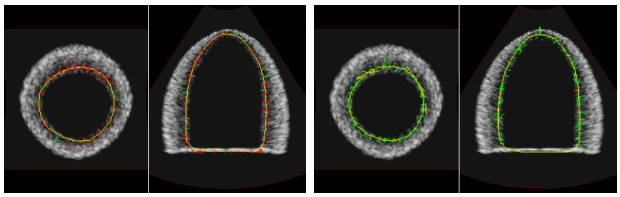
Endocardial edge-detection is performed in search-profiles perpendicular to the surface, as described in [8]. The position of the detected edges are then shifted 1 mm to align the model slightly inside the myocardium where speckle pattern is present.

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

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

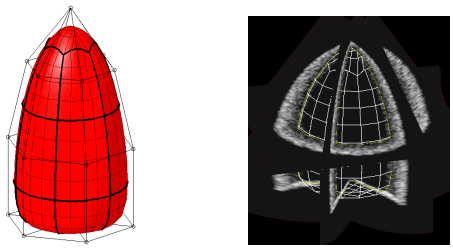
C. Measurement Assimilation

The measurements can be efficiently assimilated in information space if we assume that they are uncorrelated [1], since uncorrelated measurements lead to a diagonal measurement covariance matrix $\mathbf{R}$. All measurement information can then



(a) (b)

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



(a) (b)

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

be summed into an information vector and matrix of dimensions invariant to the number of measurements:

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

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

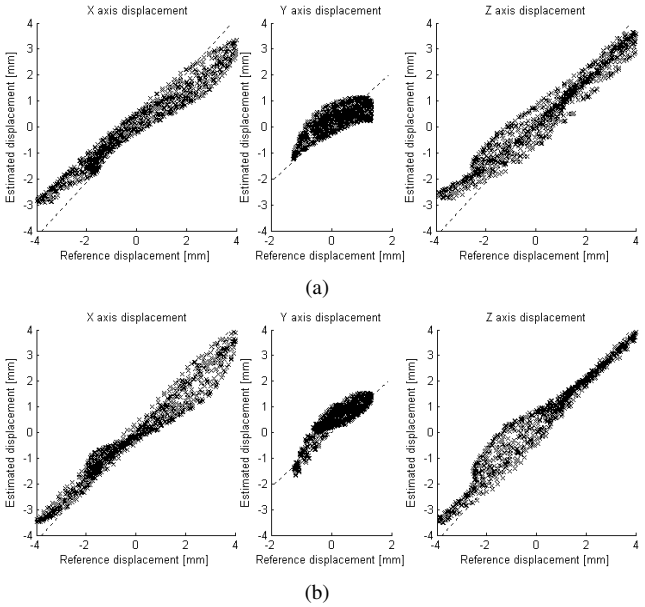
Usage of unit vectors in x, y and z-direction for displacing control vertices enables direct usage of the Jacobian matrices as measurement matrices in the Kalman filter, since $\mathbf{H}^T = [\mathbf{v}_x,\ \mathbf{v}_y,\ \mathbf{v}_z]^T\mathbf{J} = \mathbf{J}$. A covariance matrix for the measurement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

III. Results

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

The 3D ultrasound simulation was based on a finite element simulation of a left ventricle with an anterio-apical infarction [10]. Material points from the finite element simulation were fed as scatter points into a *k-space* ultrasound simulator [5] to create 3D ultrasound images with a realistic speckle pattern. Tracking was initialized automatically by using edge-detection to align the model to the endocardial boundary. After initialization, the model was moved 1 mm inside the myocardium, and edge-detection was combined with speckle-tracking to track the speckle pattern from frame to frame, as can be seen in fig. 2.

Figure 4 shows scatter plots of the correlation of displacement vectors from end diastole (ED) to end systole (ES)



(a)

(b)

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

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

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

on the tracked surface. The displacements are divided into their X, Y and Z components[1], and compared to ground truth values from the simulation. Both absolute and relative root-mean-square (RMS) error analysis on the scatter plots was conducted as shown in table I. The analysis shows improved agreement for the combined approach, with the scatter points closer to the unit line than with speckle-tracking alone. This improvement was mainly caused by less underestimation of large displacements.

After tracking, the subdivision surface were re-meshed into a quadrilateral mesh, and color-coded based on area strain ($\epsilon = (a - a_0)/a_0$), which is a measure of relative area change in each quadrilateral that combines the effect of longitudinal and circumferential strain. Figure 5 shows ES area strain values across the tracked surface using speckle-tracking alone, combined edge-detection and speckle-tracking, as well as ground truth strain values. The infarcted regions are correctly identified by both methods, although speckle-tracking alone leads to underestimated and smeared out strain values compared to the combined approach. A slight misfit between the shape of the model and the ground truth, caused

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

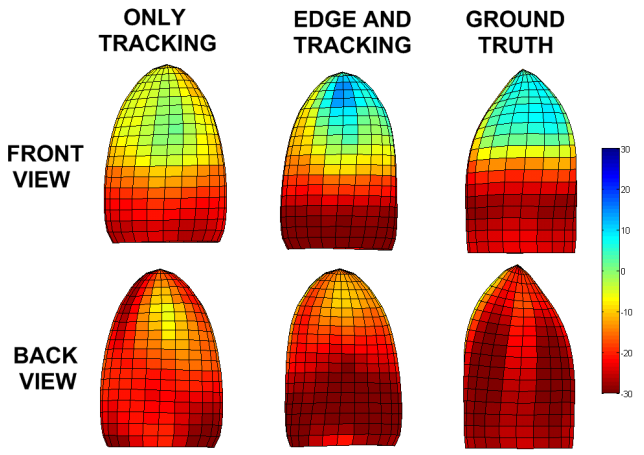


Figure 5. Front and back views of color-coded area strain meshes at ES based on speckle-tracking alone (left), edge-detection and speckle-tracking (center), as well as the ground truth (right).

by the smooth subdivision model's inability to represent the sharp apex shape, was also observed.

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

IV. Discussion and Conclusions

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

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

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

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

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

References

[1] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.

[2] J. D'hooge, B. Bijnens, J. Thoen, F. Van de Werf, G.R. Sutherland, and P. Suetens. Echocardiographic strain and strain-rate imaging: a new tool to study regional myocardial function. *Medical Imaging, IEEE Transactions on*, 21(9):1022–1030, Sep 2002.

[3] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, November 1978.

[4] A. Elen, D. Loeckx, H. F. Choi, H. Gao, P. Claus, F. Maes, P. Suetens, and J. D'hooge. P4a-5 3D cardiac strain estimation using spatio-temporal elastic registration: In silico validation. *Ultrasonics Symposium, 2007. IEEE*, pages 1945–1948, 28-31 Oct. 2007.

[5] T. Hergum, J. Crosby, M.J. Langhammer, and H. Torp. The effect of including fiber orientation in simulated 3D ultrasound images of the heart. *Ultrasonics Symposium, 2006. IEEE*, pages 1991–1994, 2-6 Oct. 2006.

[6] L. D. Jacobs, I. S. Salgo, S. Goonewardena, L. Weinert, P. Coon, D. Bardo, O. Gerard, P. Allain, J. L. Zamorano, L. P. de Isla, V. Mor-Avi, and R. M. Lang. Rapid online quantification of left ventricular volume from real-time three-dimensional echocardiographic data. *European Heart Journal*, 27:460–468, November 2006.

[7] J Meunier. Tissue motion assessment from 3D echographic speckle tracking. *Physics in Medicine and Biology*, 43(5):1241–1254, 1998.

[8] F. Orderud and S. I. Rabben. Real-time 3D segmentation of the left ventricle using deformable subdivision surfaces. *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, 2008.

[9] Fredrik Orderud, Gabriel Kiss, Stian Langeland, Espen W. Remme, Hans G. Torp, and Stein I. Rabben. Real-time left ventricular speckle-tracking in 3d echocardiography with deformable subdivision surfaces. In *MICCAI 2008 Workshop on Analysis of Functional Medical Images*, pages 41–48, 2008.

[10] Espen Remme and Otto Smiseth. Characteristic strain pattern of moderately ischemic myocardium investigated in a finite element simulation model. In *Functional Imaging and Modeling of the Heart*, pages 330–339, 2007.