

Time Resolved Three-dimensional Automated Segmentation of the Left Ventricle

E Heiberg¹, L Wigström², M Carlsson¹, AF Bolger³, M Karlsson⁴

¹Department of Clinical Physiology, Lund University, Lund, Sweden

²Department of Clinical Physiology, Linköping University, Linköping, Sweden

³Department of Cardiology, University of California, San Francisco, USA

⁴Department of Biomedical Engineering, Linköping University, Linköping, Sweden

Abstract

This paper describes a robust approach for multimodality segmentation of the cardiac left ventricle. The method is based on the concept of deformable models, but extended with an enhanced and fast edge detection scheme that includes temporal information, and anatomical a priori information. The algorithm is implemented with a fast numeric scheme for solving energy minimization, and efficient filter nets for fast edge detection. This allows clinically applicable time for a whole time resolved 3D cardiac data set to be achieved on a standard desktop computer. The algorithm is validated on images acquired using MRI Gradient echo, MRI (SSFP) images, and Cardiac CT. The complete algorithm is implemented into a software package freely available for non commercial research at <http://segment.heiberg.se>.

1. Introduction

Segmentation of the left ventricle (LV) is of great clinical interest since it allows to directly measure important parameters such as end-diastolic volume, ejection fraction, and myocardial mass.

A broad spectrum of LV segmentation techniques for different imaging modalities have been proposed, ranging from simple techniques such as thresholding or region growing, to boundary tracing [1], probabilistic or statistical models [2, 3] level-sets [4], deformable models/active contours [5], and 3D active appearance models [6]. Active appearance models have problems of coping with shape variability outside the learning set, and it is computationally expensive to have a learning set that includes all phases of the cardiac cycle. Level set methods are rather computationally intensive for 3D+T data sets.

The presented method can easily be adopted to specific imaging modalities, by a changing a few carefully selected

parameters. There are in practice two parameters governing the forces on the deformable model, and two parameters that control how the image intensity is treated.

2. Method

The basic idea behind the concept of deformable models is that a geometrical representation of an object is allowed to deform under an internal deformation energy (controlling the allowed shapes of the deforming object), and an external potential energy field (from the input images or user interaction). The deformable object is allowed to deform such as it fits the input data. Minimizing the energy of the deformable object σ , leads to an Euler-Lagrange equation which basically states or express that the internal and external forces must balance at equilibrium [7]. In this paper deforming the object is based solely on solving a modified Euler-Lagrange equation that states that the forces need only to balance along the normal vector \hat{n} at the model surface. The objective for modifying the Euler-Lagrange equation is to avoid that the parametrization of the deformable model σ influence the deforming forces.

2.1. Model Representation

The left ventricular model used in this approach is a time-resolved mesh representation of the LV as an open 'cone', sliced along the cone's long axis with an equal number of points in each slice. The number of points in each slice is 80. The model is positioned with a single mouse click in the approximate center of the LV. The image intensity in that region is taken as an estimate of the blood pool signal intensity. The segmentation is performed first on the endocardial surface, and that surface is subsequently used to initialize the segmentation of the epicardial surface.

2.2. Edge detection

A majority of the existing deformable model segmentation approaches use traditional edge detection techniques such as Sobel filters or Monga-Deriche operators to produce an edge image. The basic idea in this paper is that, by knowing the general shape, we can readily approximate *a priori* the direction in which to anticipate the edge. This allows us to use a direction-sensitive edge filter to find the edges in only that particular direction, thereby avoiding spurious edges in other directions. In Section 2.4 we show how to include temporal information in the edge detection at a very small computational cost to improve on this. The idea of including temporal information is not new. It has been previously been described in [1, 8, 9, 10].

We further separate the proposed edge detection scheme in two variants, depending on the type of edge being expected. These edge types are denoted as 'concordant edge' and 'discordant edge'. In the case of concordant edge the deformable model is attracted to a black-white transition. In the case of discordant edge the deformable model is attracted to any border/edge regardless of the type of transition (black-white, or white-black). The input image is processed with an edge detection scheme in four different directions ϕ (evenly distributed within half of the unit circle) producing four 'edge images' \mathbf{E}_ϕ as:

$$\begin{aligned} \text{concordant edge: } \mathbf{E}_\phi &= \mathbf{I} * \mathbf{f}_\phi * \mathbf{s}_\phi^a * \mathbf{s}_{\perp\phi}^b \\ \text{discordant edge: } \mathbf{E}_\phi &= \left| \mathbf{I} * \mathbf{d}_\phi * \mathbf{s}_\phi^a * \mathbf{s}_{\perp\phi}^b \right| * \mathbf{d}_\phi \end{aligned} \quad (1)$$

where \mathbf{I} is the input image, $*$ denotes image convolution, \mathbf{f}_ϕ is a small edge-detecting filter (second order differentiation), \mathbf{d}_ϕ is a derivate filter, and \mathbf{s}_ϕ^a is a small directional smoothing filter applied recursively a times. All filters are directionally sensitive, denoted by the index ϕ , and $\perp\phi$ denotes a direction perpendicular to the direction ϕ . The kernels are $\mathbf{f} = [-2 \ 1 \ 2]$, $\mathbf{d} = [-1 \ 0 \ 1]$, $\mathbf{s} = [1 \ 2 \ 1]$. The directional sensitivity is implemented by rotating the kernels. Note that different filters are applied to the two edge types, it is not merely the absolute operator that differs.

2.3. Force Calculation

The forces are calculated on points/nodes on the model $\mathbf{o}(l, t, z)$ surface, where l is along the circumference of the model, t is time, and z is in the long-axis direction.

There are two image dependent forces, an inflating 'balloon force', and one edge force. By making the balloon force image-dependent, we are able to incorporate both regional image intensity information and edge information in the model. The balloon force is calculated as:

$$\text{concordant edge: } \mathbf{F}_B = \hat{\mathbf{n}} (\alpha_0 \mathbf{I} + \lambda \alpha_1) \quad (2)$$

$$\text{discordant: } \mathbf{F}_B = \hat{\mathbf{n}} \left(e^{-\frac{(\mathbf{I}-\lambda)^2}{\alpha_0}} + \alpha_1 \right) \quad (3)$$

where \mathbf{I} is the input image, λ is an estimated image intensity for the object, α_0, α_1 are two image modality-dependent constants. The edge detection scheme in Section 2.2 results in four precalculated edge images \mathbf{E}_i . These edge images are mapped to an edge force by:

$$\text{concordant: } \mathbf{F}_E = \hat{\mathbf{n}} \frac{|\hat{\mathbf{n}}\hat{\mathbf{e}}_l| \mathbf{E}_l + |\hat{\mathbf{n}}\hat{\mathbf{e}}_s| \mathbf{E}_s}{|\hat{\mathbf{n}}\hat{\mathbf{e}}_l| + |\hat{\mathbf{n}}\hat{\mathbf{e}}_s|} \quad (4)$$

$$\text{discordant: } \mathbf{F}_E = \hat{\mathbf{n}} \frac{(\hat{\mathbf{n}}\hat{\mathbf{e}}_l) \mathbf{E}_l + (\hat{\mathbf{n}}\hat{\mathbf{e}}_s) \mathbf{E}_s}{|\hat{\mathbf{n}}\hat{\mathbf{e}}_l| + |\hat{\mathbf{n}}\hat{\mathbf{e}}_s|} \quad (5)$$

where \mathbf{E}_l is the precalculated edge image for the filter where $|\hat{\mathbf{n}}\hat{\mathbf{e}}_l|$ is largest, and respectively \mathbf{E}_s is the filter response for which $|\hat{\mathbf{n}}\hat{\mathbf{e}}_s|$ is second largest. Note that since we are using the model normal $\hat{\mathbf{n}}$ to interpolate differently from the edge images we are only looking for edges in *a priori* estimated directions, and can disregard spurious edges.

There are four internal forces in the proposed algorithm, curvature \mathbf{F}_C , slice \mathbf{F}_Z , acceleration \mathbf{F}_M , and damping force \mathbf{F}_D all with the purpose of ensuring spatial and temporal smoothness. They are calculated as:

$$\mathbf{F}_C = \hat{\mathbf{n}} \frac{(\hat{\mathbf{x}} \frac{\partial \mathbf{o}}{\partial t}) \left(\hat{\mathbf{y}} \frac{\partial^2 \mathbf{o}}{\partial t^2} \right) - \left(\hat{\mathbf{x}} \frac{\partial^2 \mathbf{o}}{\partial t^2} \right) \left(\hat{\mathbf{y}} \frac{\partial \mathbf{o}}{\partial t} \right)}{\left((\hat{\mathbf{x}} \frac{\partial \mathbf{o}}{\partial t})^2 + (\hat{\mathbf{y}} \frac{\partial \mathbf{o}}{\partial t})^2 \right)^{\frac{3}{2}}} \quad (6)$$

$$\mathbf{F}_Z = \hat{\mathbf{n}} \frac{\partial^2 \mathbf{o}}{\partial z^2} \quad (7)$$

$$\mathbf{F}_D = -\hat{\mathbf{n}} \frac{\partial \mathbf{o}}{\partial t} \quad (8)$$

$$\mathbf{F}_M = \hat{\mathbf{n}} \frac{\partial^2 \mathbf{o}}{\partial t^2} \quad (9)$$

In order to allow user interaction, so-called pin forces were included in the model formulation. When a pin is placed, the closest point on the surface is located within the same slice and time frame. The pin forces are the basis for the three basic types of user interaction with the segmentation program where the user 1) can place pins at specific locations and times, 2) interactively drag the contour while optimizing the contour 3) manually draw a section of the wall with a pen-tool. The user interaction force is given by:

$$\mathbf{F}_P = w_{i,j} \hat{\mathbf{n}} (\mathbf{p} - \mathbf{o}) \quad (10)$$

where \mathbf{p} is the coordinate of the pin, and $w_{i,j}$ is a weight for the different nodes and is lineary decreasing away from the point.

2.4. Temporal edge detection

Temporal information in the edge detection is included by temporally smoothing the edge force at one node-point \mathbf{F}_E over several timeframes.

$$\mathbf{F}_{\bar{E}}(l, t_n, z) = \sum_{i=-N}^N w_i \mathbf{F}_E(l, t_{n+i}, z) \quad (11)$$

where w_i decreases linearly away from i , and $\mathbf{F}_{\bar{E}}$ is the temporally smoothed edge force. Note that we smooth the edge force over spatially different positions $\mathbf{o}(l, t_{n+i}, z)$. The sum in (11) can be calculated by convolution, at a completely negligible cost in processing time compared to the model deformation.

2.5. Deforming the Model

The different kinds of forces are calculated for each node-point and summed to form the deforming forces as:

$$\begin{cases} \mathbf{F}_{ext} &= \alpha_B \mathbf{F}_B + \alpha_E \mathbf{F}_E \\ \mathbf{F}_{int} &= \alpha_C \mathbf{F}_C + \alpha_Z \mathbf{F}_Z + \alpha_M \mathbf{F}_M + \alpha_D \mathbf{F}_D \\ \mathbf{F}_{user} &= \alpha_P \mathbf{F}_P \end{cases} \quad (12)$$

where $\alpha_B \dots \alpha_P$ are image modality dependent. α_B is set to one, α_E and α_C are image modality dependent, whereas $\alpha_Z, \alpha_M, \alpha_D$ can be estimated directly from the image spatial and temporal resolution. α_P is simply set such as $\alpha_P \gg \alpha_B$. The model is deformed as:

$$\mathbf{o}_{n+1} = \mathbf{o}_n + \gamma (\hat{\mathbf{n}} \mathbf{F}_{ext} + \hat{\mathbf{n}} \mathbf{F}_{user} - \hat{\mathbf{n}} \mathbf{F}_{int}) \hat{\mathbf{n}} \quad (13)$$

where γ is a scaling term. The constant γ is set at 0.75 times the minimal distance between two node points divided by the largest force on the model. It is decreased to a minimum value of 0.05 times the normalization. In order to achieve numerical stability, after each iteration the points defining the contour of each slice in every time frame are redistributed to maintain equidistance between points. Care is also taken to avoid rotation between slices by maintaining the correspondance between individual reference points for each slice.

2.6. Papillary muscles

The segmentation is performed such as the papillary muscles are excluded using an ad-hoc algorithm. After a coarse segmentation of the endocardium (fewer iterations), a shape given on a polar form was fitted based on a $L^{0.5}$ norm in each slice and timeframe. Thereafter the shape is refined by another 20 iterations. The volume of papillary

muscle and trabeculation within the 'blood pool' is estimated by calculating an automatic threshold. This hybrid approach allows both accurate regional wall motion analysis and correct LV mass estimates.

2.7. Long Axis Motion

Long-axis motion is an important component of LV function. It is implemented by removing fractions of the basal slice(s) by the LV volume curve. The amount of long axis motion is manually measured from a separately long axis acquisition or can be estimated using an assumption that the LV mass is constant throughout the cardiac cycle.

3. Materials

For the validation the following set of image stacks were used, 14 MRI-SSFP (patients), 9 gradient echo MRI (healthy normals) and finally 7 3D Cardiac CT image stacks. The proposed method and a graphical user interface was implemented in Matlab (Mathworks, Natwick, USA) and compiled to a stand-alone application.

4. Results

For validation the data sets were analysed manually and automatically with no user interaction. The volume of the papillary muscles were included in the volume calculation. End-diastolic volumes were compared, and the results are shown using a Bland-Altman plot in Fig. 1, panel A-B).

The results of the epicardium detection was not satisfactory without manual corrections for gradient echo images. The largest difference between manual and automatic segmentation was in the two most basal slices. The total computational time for segmentation of the whole left ventricle in a typical MRI dataset with 12 slices and 30 timeframes was approximately 35 seconds on a 1.6GHz Intel Pentium 4 PC where of time for the edge detection was less than 5 seconds. Time to do the manual corrections in all time frames is about 2-3 minutes. This should be compared to 20 ± 5 minutes for manual delineation of two timeframes in the SSFP data sets. To investigate the importance of the temporal smoothing the 9 gradient echo data sets were segmented with and without application of temporal smoothing described in Section 2.4. Visual comparison of these paired segmentation results demonstrated a noticeable difference in 3 of the 9 data sets and with temporal edge detection the true endocardial border was identified.

The software is available free for non commercial applications at: <http://segment.heiberg.se>. An example of the graphical user interface is shown in Fig.1 C).

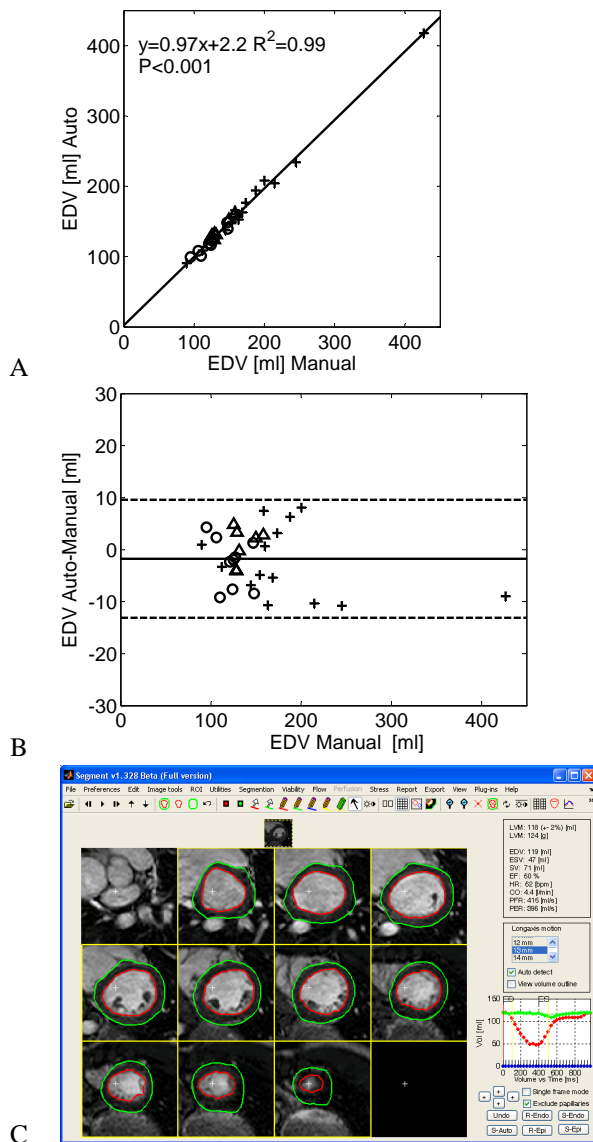


Figure 1. Correlation A) and Bland-Altman plot B) of EDV (end-diastolic volume). Different modalities are encoded as gradient echo (\circ), MRI SSFP ($+$), and contrast enhanced Cardiac CT (\triangle). C) Example of the graphical user interface.

5. Discussion and Conclusions

The multimodality segmentation approach that we present can accomplish automatic LV segmentation with data from MRI and Cardiac CT. The key to its success is that the algorithm uses all available data in a truly 3D+T manner. The used temporal edge detection is an important contribution that allows edge detection improvement at a very small computational cost. The proposed method is a huge time saver compared to manual delineation and provides a time resolved segmentation.

Acknowledgements. Grants for this project was provided by Swedish Research Council, and Swedish Heart and Lung foundation.

References

- [1] Maes L, Bijnens B, Suetens P, Werf Fvd. Automated contour detection of the left ventricle in short axis view in 2d echocardiograms. *Machine Vision and Applications* 1993; 6:1–9.
- [2] Dias JMB, Leitão JMN. Wall position and thickness estimation from sequences of echocardiographic images. *IEEE Transactions on Medical Imaging* 1996;15(1):25–38.
- [3] Lorenzo-Valdes M, Sanchez-Ortiz GI, Elkington AG, Mohiaddin RH, Rueckert D. Segmentation of 4d cardiac mr images using a probabilistic atlas and the em algorithm. *Medical Image Analysis* 2004;8(3):255–65.
- [4] Paragios N. A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE Transactions on Medical Imaging* 2003;22(6):773–6.
- [5] McInerney T, Terzopoulos D. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Computerized Medical Imaging and Graphics* 1995;19(1):69–83.
- [6] Mitchell SC, Bosch JG, Lelieveldt BP, van der Geest RJ, Reiber JH, Sonka M. 3-d active appearance models: segmentation of cardiac mr and ultrasound images. *IEEE Transactions on Medical Imaging* 2002;21(9):1167–78.
- [7] McInerney T, Terzopoulos D. Deformable models in medical analysis: a survey. *Medical Image Analysis* 1996; 1(2):91–108.
- [8] Brandt E, Wigström L, Wranne B. Segmentation of echocardiographic image sequences using spatio-temporal information. In Taylor C, Colchester A (eds.), *MICCAI 99*, volume 1679 of *Lecture Notes in Computer Science*. Cambridge, UK: Springer Verlag, ISBN 3-540-66503-X, 1999; 410–419.
- [9] Friedland N, Adam D. Automated ventricular boundary detection from sequential ultrasound images using simulated annealing. *IEEE Transactions on Medical Imaging* 1989; 4:344–353.
- [10] Mulet-Parada M, Noble JA. 2d+t acoustic boundary detection in echocardiography. In Wells WM, Colchester A, Delp S (eds.), *MICCAI-98, Lecture Notes in Computer Science*. Cambridge: Springer Verlag, 1998; 806–813.

Address for correspondence:

Einar Heiberg
 Dept. of Clinical Physiology / Lund University Hospital
 221 85 Lund / Sweden tel.: +46-46-173137
 einar.heiberg@med.lu.se