Dynamic Deformable Models for MRI heart segmentation

Leonid Zhukov, Josh Bao, Igor Guskov[†], John Wood [‡], David Breen

Department of Computer Science
California Institute of Technology
Pasadena, CA

[†] Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI

[‡] Childrens Hospital of Los Angeles
Los Angeles, CA

1 Introduction

Cine magnetic resonance imaging has become the gold-standard for measurement of myocardial mass and function. Quantization of ventricular mass and function are important in myocardial diseases, such as coronary artery disease, myocarditis, and cardiomyopathy, as well as systemic diseases such as hypertension, diabetes, and cancer. In this regard, cardiac magnetic resonance imaging (MRI) has consistently demonstrated superior accuracy and reproducibility to 2D-echocardiography without the ionizing radiation needed by computed tomography. Typically, cross-section images at 10-12 anatomic levels are imaged at 16-32 equally spaced points throughout the cardiac cycle. Current state-of-the-art commercial tools use semi-automatic boundary detection to trace endocardial and epicardial boundaries at their maximum (end-diastolic) and minimum (end-systolic) frames. The traced contours are then combined to produce boundary representation of the volumes.

The goal of our work is to develop a clinically useful, semi-automatic method to reduce postprocessing of cardiac cine-MRI images. Given the inherent difficulty that level and gradient techniques encounter near isocontrast boundaries, such as papillary muscle next to the ventricular free-wall, we developed a method that combines a small number of user-defined constraint points with a deformable model using some of the latest advances in mesh processing technology. The method allows a user to quickly extract a 3D model of heart structures by specifying key features in an MRI dataset and controlling a deformable model.

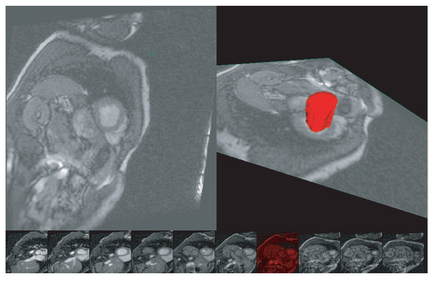

Figure 1 Main window of our software system displaying a loaded heart MRI dataset with several constraint points

2 Method

2.1 Data

Cardiac images were collected using a 4 element phased-array torso coil on a 1.5 T CVi System (General Electric Medical Systems, Milwauke, WI) running CNV 3 system software. Breath-hold cine steady-state free precession gradient echo images (SSFP) were collected at the cardiac short axis plane. SSFP parameters were as follows: 40 x 40 cm field of view (FOV), 128 phase encodes, 256 frequency encodes, 1 excitations (NEX), slice thickness 9 mm (short axis), TE 1.5 ms, TR 3.1 ms, 16 views per segment (VPS), 20 reconstructed cardiac phases, yielding true voxel resolution of 1.6mm x 9mm x 3.1 mm. Images were Fourier interpolated by 2 in the phase dimension. 14 slices were collected in short axis dimension.

2.2 Deformable Model

The dynamic equation for the motion of a deformable model can be written as a set of second order ordinary differential equations for the vertex position:

$$\mathbf{M}\frac{d^2\bar{\mathbf{P}}(\mathbf{r},\mathbf{t})}{dt^2}+\mathbf{C}\frac{d\bar{\mathbf{P}}(\mathbf{r},\mathbf{t})}{dt}+\mathbf{K}\bar{\mathbf{P}}(\mathbf{r},\mathbf{t})=\bar{\mathbf{F}}(\mathbf{r},\mathbf{t}) \quad (1)$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is a damping factor, $\mathbf{K}$ is a stiffness matrix of the mesh structure and $\bar{\mathbf{P}}$ is a vector of all vertices. $\bar{\mathbf{F}}$ is a vector of external forces.

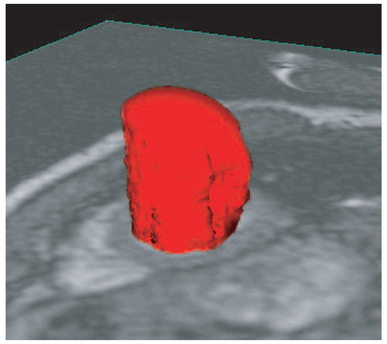

Figure 2 Evolving deformable model overlayed on a slice of MRI data.

We are looking for a static solution and following [1] we set $\mathbf{M}$ to zero, thus removing inertia. Then the model will come to rest when forces equilibrate. We can also substitute the stiffness matrix term with a force due to the curvature of the mesh (see section 2.4). This force should oppose the motion of the regions with high curvature, thus keeping the model smooth. After the renormalization the system of equations becomes

$$\frac{d\mathbf{P}(\mathbf{r},t)}{dt}-\beta\cdot\bar{\mathcal{K}}\mathbf{n}(\mathbf{P})=\mathbf{F}(\mathbf{P}). \quad (2)$$

The object grows due the to expansion force until it reaches the boundary, where the expansion force equals the curvature term and the gradient image force. Those elements that reach the vicinity of constraint points are given infinite masses and no longer move. Due to the stiffness of the mesh (guaranteed through the curvature term) the neighbors of the constrained elements have reduced mobility.

2.3 Forces

The external forces on the RHS of Eq.(2) consist of two terms: an image force and an expansion (balloon) force:

$$\mathbf{F}=\mathbf{F}_{image}+\mathbf{F}_{exp} \quad (3)$$

The image force is used to attract the model to significant 3D intensity edges (gradients). It is constructed from the image intensity as described in Gupta [2]. G_σ denotes Gaussian smoothing filter with width σ.

$$\mathbf{F}_{image}=-\alpha\cdot((\nabla|\nabla(G_\sigma * I)|)\cdot\mathbf{n})\cdot\mathbf{n} \quad (4)$$

To compute the force at any point on the model from a discrete $I(i,j,k)$ image dataset, we use linear interpolation.

When the shape of the model is not close enough to the surface of the object, the image forces may not be able to attract it. To compensate for this, an inflating force is used to expand the surface towards the boundary. This force is oriented normal to the evolving surface and has a constant value.

$$\mathbf{F}_{exp}=\gamma\cdot\mathbf{n} \quad (5)$$

2.4 Curvature Computations

We use a differential geometry definition of the mean curvature

$$\bar{\mathcal{K}}\mathbf{n}=1/2\lim_{diam(\mathcal{A})\to 0}\frac{\nabla\mathcal{A}}{\mathcal{A}} \quad (6)$$

where $\mathcal{A}$ is an infinitesimal area around a point on the surface, where we want to calculate the surface curvature. We find the curvature directly from the surface using the operator introduced in Desbrun [3], since it guarantees a zero tangential component. The discrete formulation of the above definition is

$$\bar{\mathcal{K}}\mathbf{n}(\mathbf{P}_i)=-\frac{1}{4A}\sum_{j\in N(i)}(cot\,\alpha_j+cot\,\beta_j)(\mathbf{P}_j-\mathbf{P}_i) \quad (7)$$

where α_j and β_j are the two angles opposite to the edge in the two triangles having the edge e_{ij} in common. A is the sum of the areas of the triangles having $\mathbf{P}_i$ as a common vertex. After normalization the equation (7) becomes

$$\bar{\mathcal{K}}\mathbf{n}(\mathbf{P}_i)=-\frac{\sum_{j\in N(i)}(cot\,\alpha_j+cot\,\beta_j)(\mathbf{P}_j-\mathbf{P}_i)}{\sum_{j\in N(i)}(cot\,\alpha_j+cot\,\beta_j)}. \quad (8)$$

2.5 Dynamic Meshes

During the deformation, adaptive subdivision is invoked to ensure that adequate resolution is maintained for the model. Edges longer than predefined ϵ are removed by midpoint insertion. An edge collapse is performed when an edge is too short, less than $\epsilon/2$ in this case. The endpoint with lower valence is collapsed into the end with higher valence and the neighbors from the two original vertices are combined. In order to prevent deformities, special care is taken to not collapse valence 3 vertices. Finally, edge swapping is performed periodically to optimize geometric properties and force the average valence to be as close to 6 as possible. For two neighboring triangles ABC and CBD, we perform valence optimization by flipping the diagonal BC if and only if the quantity

$$\sum_{p\in\triangle}(valence(p)-6)^2 \quad (9)$$

is minimized after the flip. Since edge swaps can change the local topological properties of the mesh significantly when the mesh is coarse, this is only performed periodically after the number of vertices exceeds 500, or some other sufficiently large number. Furthermore, edge swapping can be very expensive; hence it is performed only when the average valence deviates too far from 6. This is controlled by a parameter that can be altered dynamically by the user. Combining the three operations, we can achieve a provably good mesh. Since the edge ratio is at most 1:2, triangle sizes vary by at most a factor of 4.

2.6 Hard Constraints

User specified constraint points can be added near ill-defined boundaries to help guide the segmentation. A moving triangle (all its vertices) will stop when it reaches the predefined neighborhood of the constraint point.

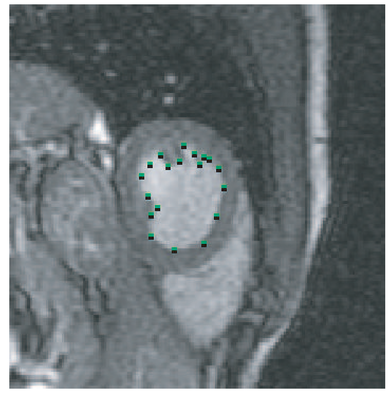

Figure 3 A close-up of a single slice of MRI data, showing user-defined constraint points.

2.7 Numerical Method

We convert time derivatives into finite difference operators (in time) and solve the system by simple forward Euler integration.

$$\mathbf{P}_i(\mathbf{r},t+\triangle t)=\mathbf{P}_i(\mathbf{r},t)+(\mathbf{F}(\mathbf{P_i})-\beta\cdot\bar{\mathcal{K}}\mathbf{n}(\mathbf{P})_i)\triangle t \quad (10)$$

Time step $\triangle t$ and coefficients α, β and γ are chosen empirically to provide convergence. Since tension and ballooning forces are defined directly on the surface, we do not need to discretize the volume to find the solution. The image force is also only computed at the 3D location where it acts on the vertices of the mesh.

After every time step, the mesh is updated using dynamic meshing (2.5), thus the number of vertices and connectivities are changing every iteration step. This step guarantees smooth and consistent properties of the mesh.

2.8 Software System

Our dynamic deformable model has been implemented in a software system written in C++ using the OpenGL, GLUT and GLUI libraries. It provides an interface for a user to tag key features in the data with constraint points and to control the model's deformation by interactively adjusting parameters. The data is presented on three orthogonal cutting planes, which may be individually swepted through the data. The constraint points may be placed directly on any of the three planes, completely specifying their location in Cartesian space.

A control panel is used to adjust all parameters dynamically, while the program is running. For instance, the image force can be increased, once the dynamic model approaches a weak boundary.

A GLUI interface and GLUT menus are available for easy navigation of the software system. Directional arrows can be dragged to move and resize the mesh as well as the orthogonal planes. Options for hiding the mesh, orthogonal planes, or constraint points are also readily accessible in the form of checkboxes. Different modes of rendering are kept in radio boxes, while parameters for expanding and subdividing the mesh can be dynamically changed in text boxes during the expansion phase. More advanced options for saving and loading constraint points and the mesh are available through GLUT menus.

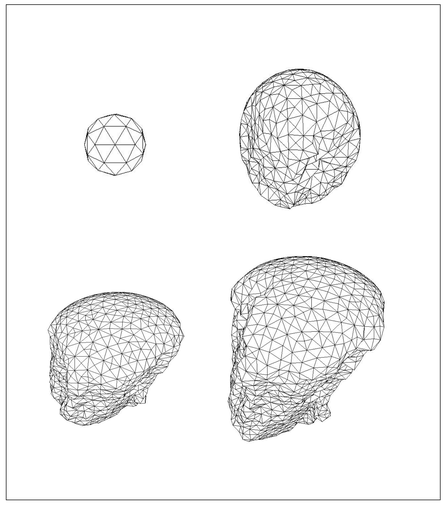

Figure 4 A sequence of dynamic models produced from a segmentation of the left ventricle.

3 Results

Figures 1,2 and 3 present segmentation results produced by our software system. Figure 4 contains a close-up of one of the cutting planes after a user has placed constraint points. In Figure 2 the initial model has begun to fit to the wall of the left ventricle. The mesh model is overlayed on the cutting plane. Figure 3 contains four models produced during the dynamic deformation process. The top left is the initial model. The top right and bottom left are intermediate shapes. The final model is presented in the bottom right.

4 Conclusions

We have presented a new dynamic deformable modeling approach to 3D segmentation. It utilizes recently developed dynamic remeshing techniques and curvature estimation methods to produce high-quality meshes. The approach has been implemented in an interactive environment that allows a user to specify an initial model and identify key features in the data. These features act as hard constraints that the model must not pass through as it deforms. We have employed the method to perform semi-automatic segmentation of heart structures from cine MRI data.

References

[1] T. McInerney and D. Terzopoulos, "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* **19**(1), pp. 69–83, 1995.

[2] A. Gupta, T. O'Donnell, and A. Singh, "Segmentation and tracking of cine cardiac MR and CT images using a 3-D deformable model," in *Proc. IEEE Conference on Computers in Cardiology*, pp. 661–664, 1994.

[3] M. Desbrun, M. Meyer, P. Schröder, and A. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. SIGGRAPH '99*, pp. 317–324.