LV SURFACE RECONSTRUCTION FROM SPARSE TMRI USING LAPLACIAN SURFACE DEFORMATION AND OPTIMIZATION

Shaoting Zhang, Xiaoxu Wang, Dimitris Metaxas

Rutgers, the State University of New Jersey
Computer Science Department

Ting Chen, Leon Axel

New York University
Radiology Department

ABSTRACT

We propose a novel framework to reconstruct the left ventricle (LV)'s 3D surface from sparse tagged-MRI (tMRI). First we acquire an initial surface mesh from a dense tMRI. Then landmarks are calculated both on contours of a specific new tMRI data and on corresponding slices of the initial mesh. Next, we employ several filters including global deformation, local deformation and remeshing to deform the initial surface mesh to the image data. This step integrates Polar Decomposition, Laplacian Surface Optimization (LSO) and Deformation (LSD) algorithms. The resulting mesh represents the reconstructed surface of the image data. Further more, this high quality surface mesh can be adopted by most deformable models. Using tagging line information, these models can reconstruct LV motion. The experimental results show that compared to Thin Plate Spline (TPS) our algorithm is relatively fast, the shape represents image data better and the triangle quality is more suitable for deformable model.

Index Terms— Reconstruction, deformable model, laplacian surface, optimization, triangle quality, remeshing

1. INTRODUCTION

tMRI is a non-invasive way to track the in vivo myocardial motion during cardiac cycles. Reconstructing the 3D LV motion from tMRI can assist doctors to diagnose cardiac diseases earlier, and can also be used for 3D strain analysis of the myocardium. A typical approach of reconstruction is to calculate the motion of material markers from tags. Employing these material markers as control points, deformable models can reconstruct myocardial motion. Mesh based models approaches such as Finite Element Methods (FEM) and quadric models have been widely used for LV motion reconstruction. An accurate and high quality mesh is crucial to initialize these models. If the tMRI data is dense enough, the resulting mesh can be acquired by segmentation and triangulation. However, in many cases dense data is unavailable. An alternative approach is to obtain a high quality mesh from a generic dense tMRI, and then deform this mesh to any sparse data for surface reconstruction.

In our previous research [1], a plausible mesh is built by manual segmentation with validation by an expert, and Delaunay triangulation using geodesic distances. Next landmarks are calculated from this mesh and any sparse LV tMRI. Finally TPS [2] are employed to deform this mesh to LV tMRI. Due to TPS, the whole procedure is slow and the resulting mesh is not good enough measured by radius ratio.

This paper presents an effective deformation method, which can replace TPS. The algorithm consists of global deformation, local deformation and remeshing. LSO [3] and LSD [4] are employed for remeshing and local deformation respectively. The output is the reconstructed surface mesh and it can be employed as the initial mesh for deformable models. For modeling purpose, this mesh should be accurate and the triangle quality should be good [3]. Our output mesh is tested with different deformable models, such as FEM [1] [5], Laplacian Volume Editing [6], volume registration based tracking [7] and mass-spring based method [8]. The results show that it is suitable.

Our paper is organized as follows: section 2 introduces the detail of the algorithm. Section 3 applies our algorithm on sparse LV tMRI and compares it to TPS in running time, deformed shape and triangle quality. In section 4 we draw the conclusions.

2. METHODS

2.1. Framework

Figure 1 shows the framework of our algorithm. The input is the surface mesh of a generic model, model landmarks and image landmarks. In our specific case, the generic model is segmented from dense tMRI obtained from a healthy volunteer and the surface mesh was build by a Delaunay triangulation using geodesic distances. The landmarks are calculated both on the image contours and on the corresponding slices of the model. The output is the reconstructed surface mesh.

In between there are several transformation filters. Global deformations are used to roughly place the model on the image data location based on affine transformation including translation, rotation and isotropic scaling. LSO is employed twice. The first pass smooths the initial surface mesh. The

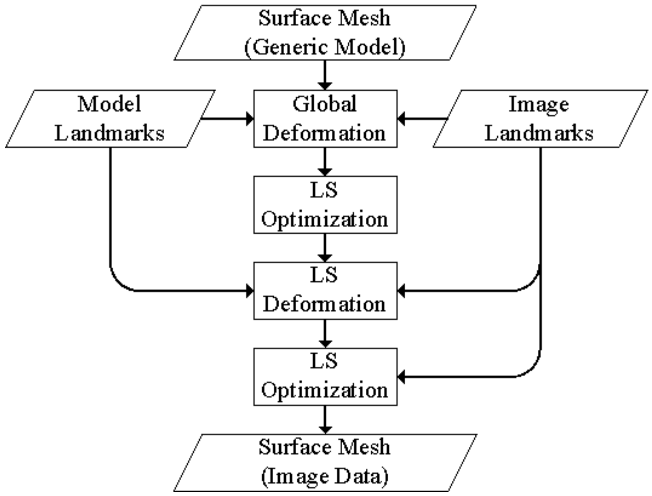

Fig. 1. Algorithm framework. Inputs are surface mesh of a generic model, model landmarks and image landmarks. Output is the deformed surface mesh of an image data. In between there are several filters such as global deformation, LSO and LSD.

second pass improves the triangle quality of output. Here triangle quality is measured by radius ratio (Section 3). LSD is employed to locally deform the model mesh to image data according to model and image landmarks. Section 2.3 and 2.4 give details on these filters.

2.2. Landmarks and Global Deformation

Our previous research [1] describes the detail of obtaining landmarks and material markers, as well as the data information. First LV boundaries and tagging line information are obtained from tMRI using Gabor Filters [9] and Metamorphs [10]. Secondly, landmarks are evenly selected from the boundary lines in short axis (SA). Since there are less then 5 slices in SA for sparse data, the landmarks are relatively sparse. Finally, the intersections of the three tagging planes are calculated, as well as the intersections of the LV boundary and the tagging planes. These intersections can be used as the material markers in LV motion tracking.

After obtaining corresponding landmarks from the surface mesh and tMRI, global deformation is applied by approximately finding the translation, scaling and rotation matrix. The global translation is defined as a vector pointing from the center of source points to the center of target points. The global scaling after translation is defined as the ratio between the radius of the source points and the radius of the target points. The global rotation after translation and scaling is obtained by Polar decomposition. After applying global translation, scaling and rotation, non-rigid local deformation is required to ensure the accuracy. LSO and LSD are employed here.

2.3. Laplacian Surface Optimization

Laplacian surface is also called differential coordinates. It represents each point as the difference between such point and its neighborhoods. LSO is an algorithm to improve triangle quality of a surface mesh. The inputs are anchor points and an initial surface mesh. The output is an optimized surface mesh.

Here we introduce notations and this algorithm. Let the mesh $\mathbb{M}$ be described by a pair $(\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{v_1, ..., v_n\}$ describes the geometric positions of the vertices in $\mathbb{R}^3$ and $\mathbb{E}$ describes the connectivity. The neighborhood ring of a vertex i is the set of adjacent vertices $\mathbb{N}_i = \{j|(i,j) \in \mathbb{E}\}$ and the degree d_i of this vertex is the number of elements in $\mathbb{N}_i$. Instead of using absolute coordinates $\mathbb{V}$, the mesh geometry is described as a set of differentials $\Delta = \{\delta_i\}$. Specifically, coordinate i will be represented by the difference between v_i and the average of its neighbors:

$$\delta_i = v_i - \frac{1}{d_i} \sum_{j \in \mathbb{N}_i} v_j \tag{1}$$

Assume V is the matrix representation of $\mathbb{V}$. The transformation between vertex coordinates V and Laplacian coordinates Δ can be described in matrix algebra. Let N be the mesh adjacency (neighborhood) matrix and $D = diag(d_1, ..., d_n)$ be the degree matrix. Then $\Delta = LV$, where $L = I - D^{-1}N$ for the uniform weights.

Using a small subset $\mathbb{A} \subset \mathbb{V}$ of m anchor points, a mesh can be reconstructed from connectivity information alone [4]. x, y and z positions of the reconstructed object ($V'_p = [v'_{1p}, ..., v'_{np}]^T, p \in \{x, y, z\}$) can be solved separately by minimizing the quadratic energy

$$\|LV'_p\|^2 + \sum_{a \in \mathbb{A}} \|v'_{ap} - v_{ap}\|^2 \tag{2}$$

where the v_{ap} are anchor (landmark) points. $\|LV'_p\|^2$ tries to smooth the object by minimizing the difference, and $\sum_{a \in \mathbb{A}} \|v'_{ap} - v_{ap}\|^2$ keeps anchor points unchanged. In practice, with m anchors, the $(n+m) \times n$ overdetermined linear system $AV'_p = b$

$$\begin{bmatrix} L \\ I_{ap} \end{bmatrix} V'_p = \begin{bmatrix} 0 \\ V_{ap} \end{bmatrix} \tag{3}$$

is solved in the least squares sense using the method of normal equations $V'_p = (A^TA)^{-1}A^Tb$. The first n rows of $AV'_p = b$ are the Laplacian constraints, corresponding to $\|LV'_p\|^2$, while the last m rows are the positional constraints, corresponding to $\sum_{a \in A} \|v'_{ap} - v_{ap}\|^2$. I_{ap} is the index matrix of V_{ap}, which maps each V'_{ap} to V_{ap}. The reconstructed shape is generally smooth, with the possible exception of small areas around anchor vertices. The minimization procedure moves each vertex to the centroid of its 1-ring, since the uniform Laplacian L is used, resulting in good inner fairness.

The main computation cost of this algorithm is big matrix multiplication and inverse. Since A is sparse matrix, A^TA is sparse symmetric definite matrix. The Conjugate Gradient algorithm can be employed to solve the system.

In our algorithm, LSO is employed twice (Figure 1). At the first time the anchor points are selected evenly and sparsely from the whole object, therefore the shape of surface mesh of generic model can be roughly conserved as well as smoothed. At the second time image landmarks are used as anchor points, therefore truth points can be fixed and triangle qualities can be improved. The output data are generated after the second pass.

2.4. Laplacian Surface Deformation

LSD is also called Laplacian Surface Editing, which is an algorithm for local deformation. The inputs are deformed control points and an initial mesh. In our specific case, control points are model and image landmarks. Model landmarks are moved to image landmarks directly. The deformation of rest points can be calculated by LSD. Note that after global deformation process, the displacements of control points are restricted in a local range. The output is the deformed mesh.

Using the same notation as Section 2.3, this time we need to minimize this quadratic energy function:

$$E(V') = \sum_{i=1}^{n} \|\delta_i - \delta'_i\|^2 + \sum_{i \in \mathbb{C}} \|v_i - v'_i\|^2 \qquad (4)$$

where δ' and v' is Laplacian and Cartesian coordinates after deformation. $\mathbb{C}$ is the set of control points. The first half try to keep the shape according to previous time step, which is δ. The second half can move control points v to deformed positions v'. However, using formula 4, no point will be moved except the control points $\mathbb{C}$. The main idea of LSD is to compute an appropriate transformation T_i for each vertex i which can be plugged into energy formula 4:

$$E(V') = \sum_{i=1}^{n} \|T_i\delta_i - \delta'_i\|^2 + \sum_{i \in \mathbb{C}} \|v_i - v'_i\|^2 \qquad (5)$$

T_i should be constrained to avoid a membrane solution, which will lose all geometric detail. Thus, T_i should include rotations, isotropic scales, and translations. In particular, anisotropic scales should not be allowed, as they allow removing the normal component from Laplacian coordinates. The class of matrices representing isotropic scales and rotation can be written in a specific format. Employing such format, formula 5 can be simplified. The detailed formula can be found in [6]. Formula 5 can be minimized iteratively by finding T_i and applying it on each vertex coordinates. When v converges, this minimization problem is solved. The transformation T_i is an approximation of the isotropic scaling and rotations when the rotation angle is small. In our framework, the major rotation is handled in the global deformation part. The local rotation fits the small angle assumption.

After reconstructing the surface of the LV and initializing the deformation models, [1] [6] [7] [8]can be employed to reconstruct the LV motion.

3. EXPERIMENTS

LSE, LSO and a sparse matrix solver are implemented and tested on a 2.40 GHz Intel Core2 Quad computer with both GNU/Linux and Windows environments. These algorithms are coded by extending vtkAlgorithm in Visualization ToolKit (VTK) and named as vtkModeling [11]. Codes and documents can be downloaded in the vtkExtend category of Sourceforge. For comparison, TPS is also tested.

The initial surface mesh of a generic LV model has 2833 vertices and 5662 polygons. The image data is MR data with seven SA slices and two LA slices. 270 landmarks are generated from SA image data and surface mesh. The whole 4-filter framework takes about 15 seconds with our matrix solver and less than 1 second with professional solver TAUCS. Note that the first pass of LSO can be performed offline, by which the speed can be improved. TPS takes 3.5 seconds with the same data sets and professional solver.

Figure 2 shows the effect of each step of our framework. Figure 3 compares results of our algorithm (red object) and TPS (green object). Applying TPS directly on the initial mesh results in bad shape due to its irregular shape and degenerated triangles. For fairness, we apply TPS on the first pass of the LSO. The TPS result is shrunk in regions far away from control points like the bottom, since TPS tries to smooth the whole shape and removes high frequency. Due to non-uniform scaling and shearing, TPS cannot guarantee a rigid deformation on the top.

Triangle shapes are also measured and compared. We measure our success with the radius ratio mapped to $[0, 1]$ as

$$t_i = 2\frac{r}{R} \qquad (6)$$

where R and r are the radii of the circumscribed and inscribed circles respectively. This way, $t_i = 1$ indicates a well shaped triangle, $t_i \in [0, a]$ means degenerate triangle. Figure 4 compares T_i value of our algorithm and TPS. t_{mean} of our mesh is 0.8932. t_{mean} of TPS mesh is 0.8628. Note that TPS is applied on the first output of LSO, whose triangle quality is already improved.

4. CONCLUSION

We presented a novel framework to reconstruct a surface mesh from sparse tMRI. The resulting high-quality mesh can be employed as the input of deformable models to reconstruct the LV motion. We also introduced LSO and LSD to the medical imaging community. The algorithm is efficient,

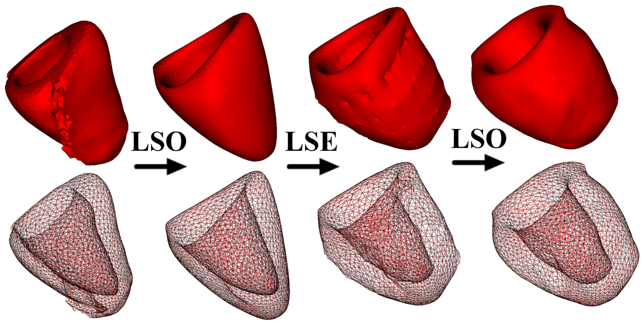

Fig. 2. The effects of our algorithm framework showed with polygon mesh and wire-frame. From left to right, they are initial surface mesh of a generic LV model, the first LSO filter, LSD filter and the second LSO filter.

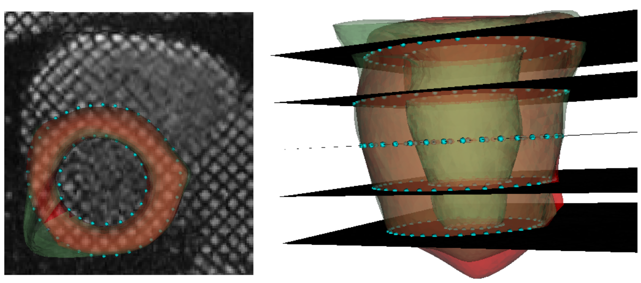

Fig. 3. Results from our algorithm (red) and TPS (green) are displayed with tMRI data. TPS result is over-deformed around control points but under-deformed in other regions.

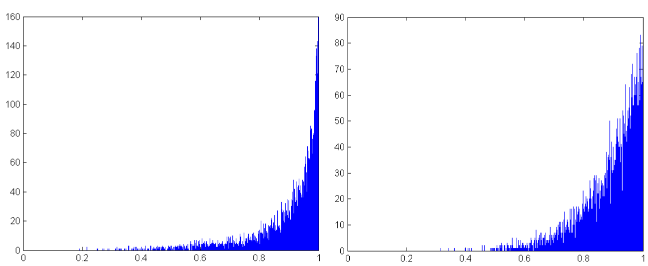

Fig. 4. Triangle quality measured by T_i histogram. The left one is obtained from our algorithm. The right one is obtained from TPS.

the resulting mesh represents the image data well, and the triangle quality is suitable for input of deformable models. Experiments are designed to compare our algorithm to TPS. LSO and LSD are implemented as VTK classes, which can be easily used in many projects. In the future we will focus on volume data reconstruction.

5. REFERENCES

[1] Xiaoxu Wang, Joel Schaerer, Suejung B. Huh, Zhen Qian, Dimitris N. Metaxas, Ting Chen, and Leon Axel, "Reconstruction of detailed left ventricle motion from tmri using deformable models," *FIMH 2007*, pp. 60–69, 2007.

[2] FL Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations," *IEEE Trans. PAMI*, vol. 11, pp. 567–585, 1989.

[3] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa, "Laplacian mesh optimization," in *GRAPHITE '06*, New York, NY, USA, 2006, pp. 381–389, ACM.

[4] Olga Sorkine, Yaron Lipman, Daniel Cohen-Or, Marc Alexa, Christian Rossl, and Hans-Peter Seidel, "Laplacian surface editing," in *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 2004, pp. 179–188.

[5] D. N. Metaxas and D. Terzopoulos, "Dynamic 3d models with local and global deformations: Deformable superquadrics," *IEEE Trans. PAMI*, vol. 13, pp. 703–714, 1991.

[6] Xiaoxu Wang, Ting Chen, Shaoting Zhang, Dimitris Metaxas, and Leon Axel, "Lv motion and strain computation from tmri based on meshless deformable models," *MICCAI2008*, pp. 636–644, 2008.

[7] Ting Chen, Xiaoxu Wang, Dimitris Metaxas, and Leon Axel, "Fast motion tracking of tagged mri using angle preserving meshless registration," *MICCAI2008*, 2008.

[8] Shaoting Zhang, Lixu Gu, Pengfei Huang, and Jianfeng Xu, "Real-time simulation of deformable soft tissue based on mass-spring and medial representation," *CVBIA, LNCS3765*, pp. 419–426.

[9] Z. Qian, D.N. Metaxas, and L. Axel, "Extraction and tracking of mri tagging sheets using a 3d gabor filter bank," *Proceedings of Intl Conf. of the Engineering in Medicine and Biology Society*, 2006.

[10] Huang X. and D.N. Metaxas, "Metamorphs: Deformable shape and appearance models," *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 30, pp. 1444–1459, 2008.

[11] Shaoting Zhang, Xiaoxu Wang, and Dimitris Metaxas, "Technical report: Vtkmodeling, an example of extending vtk," .