Motivations

• tMRI is a non-invasive way to track the in vivo myocardial motion during cardiac cycles (Figure 4). The data is usually very sparse.
• Reconstructing 3D LV motion from tMRI can assist doctors to diagnose cardiac diseases earlier, and can also be used for 3D strain analysis.
• We use deformable models to reconstruct the LV motion[1]. But, how to initialize the deformable model (get initial reconstructed surface)?
• Since the data is sparse, we can fit a generic LV mesh to a specific tMRI data. The generic mesh is manually segmented from a dense data.
• Thin Plate Spline (TPS) is a traditional approach to align two objects given landmarks. However, TPS allows shearing and non-uniform scaling (green point in Figure 4). How to obtain a better shape with higher triangle quality?

Methods

Figure 1. Algorithm framework.

• The input is the surface mesh of a generic model, model landmarks and image landmarks. 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. LSO [3] is employed twice. The first pass smooths the initial surface mesh. The second pass improves the triangle quality of output. The quality is usually measured by radius ratio. LSD [4] is employed to locally deform the model mesh to image data according to model and image landmarks.
• [2] describes how to obtain landmarks. In the following we will focus on these important filters.

Global deformation:

• 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 or SVD [5].

Figure 2. Laplacian coordinates.

Laplacian surface optimization:

• Laplacian coordinates (uniform weights): \( \delta_i = v_i - (1/\delta) \Sigma \delta_j \delta_j \), \( d \) is the valence of \( v_i \). Cotangent weight is \( \frac{1}{2}(\cot \alpha + \cot \beta) \) (green line in Figure 2).
• Denote the weight matrix as L, then \( \Lambda = L \), where \( \Lambda \) is matrix form for \( \delta_i \) and \( V \) is matrix form for vertices’ Euclidean coordinates.
• The smoothed object can be solved by minimizing the quadratic energy: \( E(V') = \Sigma (v_i' - v_i)^2 + \Sigma \delta_i (v_i - v_i')^2 \), where \( v_i' \) are anchor points.
• ILV[4] tries to smooth the object by minimizing the differences between vertices. \( \Sigma (v_i - v_i')^2 \) keeps anchor points unchanged.
• Equals to a \((n+m) \times n\) overdetermined linear system, which can be solved by conjugate gradient:

\[
\begin{bmatrix}
I & L \end{bmatrix} \begin{bmatrix}
V' \\
V \end{bmatrix} = \begin{bmatrix}
0 \\
V \end{bmatrix}
\]

Laplacian surface deformation:

• The same formulation as LSO.
• Minimize this energy function: \( E(V') = \Sigma \delta_i \delta_i + \Sigma \Sigma \delta_i (v_i - v_i')^2 \), where \( C \) is the set of control points. However, it’s just like TPS, which allows shearing and non-uniform scaling.
• The main idea of LSD is to compute an appropriate transformation \( T_i \) for each vertex and plugged into the energy formula:

\( E(V') = \Sigma T_i \delta_i \delta_i + \Sigma \Sigma \delta_i (v_i - v_i')^2 \).
• \( T_i \) is an approximation of the isotropic scaling and rotations when the rotation angle is small, which is true in our framework.
• The above formula can be minimized iteratively by finding \( T_i \) and apply it on each vertex until converge.

Results

Figure 3. Effect of each fitting step in local deformation.

Figure 4. Comparison between TPS (green) and our algorithm (red).

Figure 5. Triangle qualities (2R/1). Left – our algorithm, right – TPS.

Kernel codes (LSD and LSO) are available at:
http://sourceforge.net/projects/vtkextend/

References:

[1] Xiaoxu Wang: LV Motion and Strain Computation from MRI Based on Deformable Models, MICCAI08
[2] Xiaoxu Wang: Reconstruction of detailed left ventricle motion from multi-modal deformable models, PMB07
[3] Andrew Nixder: Laplacian mesh optimization, GRAPHITE06
[4] Olga Sorkine: Laplacian surface editing, SGP04