4D Model Reconstruction of Patient-Specific Cardiac Mesh from Segmented Contour Lines

***Chi Wan Lim[1], Yi Su[1], Si Yong Yeo[1], Gillian Maria Ng[1], Vinh Tan Nguyen[1], Liang Zhong[2], Ru San Tan[2], Kian Keong Poh[3,4], Ping Chai[3]**

[1] Institute of High Performance Computing, A*STAR, Singapore
[2] National Heart Centre Singapore, Singapore
[3] Cardiac Department, National University Heart Center, National University Health System, Singapore
[4] Department of Medicine, Yong Loo Lin School of Medicine, National University of Singapore, Singapore

*Corresponding author: limcw@ihpc.a-star.edu.sg

Abstract

We propose an automatic algorithm for the reconstruction of a set of patient-specific dynamic cardiac mesh model with 1-to-1 mesh correspondence over the whole cardiac cycle. This work focus on both the reconstruction technique of the initial 3D model of the heart and also the consistent mapping of the vertex positions throughout all the 3D meshes. This process is technically more challenging due to the wide interval spacing between MRI images as compared to CT images, making overlapping blood vessels much harder to discern. We propose a tree-based connectivity data structure to perform a filtering process to eliminate weak connections between contours on adjacent slices. The reconstructed 3D model from the first time step is used as a base template model, and deformed to fit the segmented contours in the next time step. Our algorithm has been tested on an actual acquisition of cardiac MRI images over one cardiac cycle.

Keywords: Mesh Reconstruction, Cardiac, Patient-Specific, MRI, 4D

Introduction

The cardiovascular diseases are becoming more commonplace in modern societies. One of the ways which physicians use to view the internal cross-sectional images patient’s heart with cardiovascular diseases is by Magnetic Resonance Imaging (MRI). Inclusion of computational methods in cardiac analysis can allow us to further understand and visualize what we are unable to obtain from static 2D images, such as blood flow behavior (hemodynamics) in the chambers. To achieve this, the reconstruction of the 4D (spatial temporal) mesh model of the heart is an important and critical requirement. In this paper, we propose a framework to directly reconstruct a 4D right heart model from segmented contours drawn on MRI images.

Related Work

Cardiac reconstruction, either static (3D) or dynamic (4D), has traditionally focused on the ventricular regions due to their simpler morphology and the relative ease of border delineation on CT and MRI scans, as compared to the atrial regions. This differentiation is further compounded for MRI-based methods due to its relatively large spatial intervals between slices. A comprehensive review of the different modeling techniques from cardiac images prior to 2001 was presented by Frangi et al. [1].

One of the first studies to directly construct a 3D model of the heart is by McQueen and Peskin [2], which employed idealized cones and ellipsoids for modeling the left and right ventricles for simulating cardiac hemodynamics [3]. Using 3D echocardiography, Corsi et al. [4] employed the level set method and marching cubes algorithm to reconstruct a 3D model of the LV. Zhukov et al. [5] proposed to deform a sphere model using dynamic remeshing and curvature estimation methods to produce high quality meshes of the heart. Montagnat and Delingette [6] extended the deformable surface framework by introducing time-dependent constraints such as temporal smoothing and trajectory constraints. Sermesant et al. [7] fitted various image modality together using a non-rigid registration approach to create a mesh model of both the LV and RV (right ventricle) using the GHS3D commercial software. A noteworthy attempt to reconstruct a hugely detailed static animal heart model is presented by Plank et al. [8], which uses a 9.4T MR system that is able to generate MR datasets with isotropic resolution of up to $20\mu m$.

One of the biggest obstacles in 4D heart modeling has been the modeling of the passageways between the ventricles and the atriums and the complex vessels that connects them to the rest of the body. During cardiac contraction, these vessels deform, possibly in directions perpendicular to the imaging planes. This can cause difficulties during the 4D reconstruction process, as the vessels themselves moves in and out of the imaging planes, and thus their boundaries on these planes are manifested in a constant series of merging and splitting processes. In our work, we propose a methodology that aims to mitigate this issue by utilizing a tree structure to link and track the contours throughout the cardiac cycle, in order to reconstruct a 4D cardiac mesh model with 1-to-1 vertex correspondence.

Overview of Method

The raw data input to this algorithm are the segmented contour lines, which are drawn on a set of short-axis MRI images over one cardiac cycle. We proceed to establish connectivity relationships between the segmented contours lines in two ways, inter-frame and intra-frame. Connectivities within the same time step (intra-frame) relates to the morphological structure of the heart in that particular time step; connectivities across adjacent time steps (inter-frame) provide information related to the motion of the heart across that time step.

The main challenge occurs during 2D contour topological changes, when there are no 1-to-1 inter-frame connections between contours. To handle this, we first collect the characteristics features of the 2D contour topological change, such as its vertical motion and the location of the ridge feature. Thereafter, we apply a 2D sine-based deformation function using the ridge feature as its center to induce the vertical motion, and therefore ensuring that the contours can be associated in a 1-to-1 manner across that time step.

Constructing a 3D model of the First Cardiac Frame

In this section we describe how we perform filtering to remove unwanted connections between segmented contours from adjacent slices, and then detail the reconstruction process of the 3D heart model, based on the MR images from the first time step.

Tree-Based Connectivity

The set of connectivity relationship between contours from adjacent slices and across time steps are termed as a tree-based connectivity. The 3D reconstruction makes use of the connectivities between adjacent slices (intra-frame connectivities), while the deformation stage makes use of the connectivities between adjacent frames (inter-frame connectivities). Each contour can have four different types of connections, *above* and *below*, which are intra-frame connections; and *prev* and *next*, which are inter-frame connections. The tree-based connectivity is established by comparing contours between adjacent slices (both inter- and intra-frame) based on a similarity index function. The similarity index is computed based on the proportion of the perimeter of the contour that lies within the interior of the other contour when both contours are superimposed on the same plane. The higher of the two proportion values is then taken as the similarity index (see Figure 1).

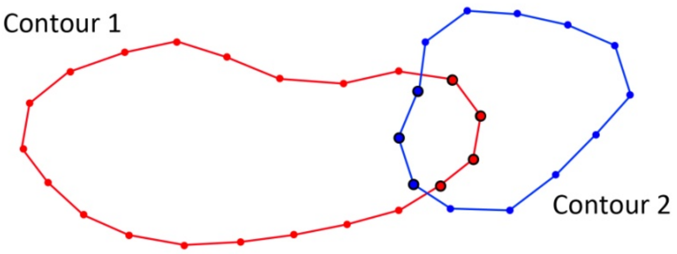

Figure 1. Determining connectivity between contours. The similarity index based on the higher of the two proportions is 23.08% (3 out of 13 points from contour 2).

In cases where there is only one pairing with non-zero similarity index, the assignment is straightforward. Otherwise, two situations could happen, either branching (intra-frame)/topological change (inter-frame) has occurred, or one of the similarity index is a very weak one. To decide between both potential situations, weak connections are filtered away by setting a minimum similarity index threshold of 30%.

Reconstruction from Segmented Contour Lines

For reconstruction of the 3D initial cardiac mesh model from the first frame, we extended the works of Barequet and Sharir [22] by incorporating the concept of contour connectivity into it such that only those contours that are connected will have a surface formed between them. One potential issue that might arise using their approach is that some of the reconstructed surface triangles might have all their three vertices belonging to the same contour, thereby making it flat when the contours are projected back. If another flat triangle exists on the other side of the contour, the resulting surface will contain a collapsed triangle-pair, thereby forming a disconnected internal volume. In order to prevent such problematic configuration, we inject new vertices into the triangulation before lifting by using the chordal axis transform (CAT) approach [23]. This CAT approach introduces new vertices which essentially act as a set of "lifting" points to break up the flat triangles to prevent a collapsed configuration. This also helps to create a relatively smoother volume.

4D Morphing of the Heart Model

The goal of the deformation process is to modify the heart model of a particular frame to fit the contours of the next frame without changing the mesh connectivity. To implement the deformation process, we split the heart model into logical sub-meshes using the MRI planes as the partitioning planes. As each sub-mesh (except the topmost and bottommost) is sandwiched between two partitioning planes (see Figure 2), we can compute the deformation for each sub-mesh independently. Using an example for illustration (see Figure 3), consider two adjacent contours *X* and *X'* that has a surface ζ between them; we wish to find a mapping of the points on *X* (source points) onto the contour *X'* (target points). To achieve a good mapping, we project *X* and *X'* onto a common plane. Next, the intersection points between *X* and *X'* are then used as breaking points to split both X and X' into segments. Each individual segment in X is then matched against its corresponding segment in X', and the points in each segment are then mapped from X and X' proportionally. Using both sets of source points and target points, we generate a radial basis function (RBF) interpolant [32] of the form

$$\tilde{f}(x) = \sum_{k}^{N} w_k \phi\,(\|x - x_k\|),$$

where the kernel function $\phi: \mathbb{R}^+ \to \mathbb{R}$ is univariate and radially symmetric, and w_k is a set of 3-dimensional weights. Our choice of the kernel function is the Hardy Multiquadric, $\phi\,(\mathrm{r}) = \sqrt{r^2 + c^2}$, where c is a constant defined individually for each source point, based on the distance to its closest neighbor. By determining the weights of the RBF, we are able use it to interpolate where the new surface ζ' lies on.

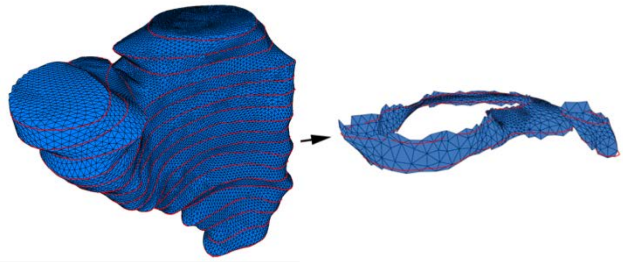

Figure 2. A Sub-mesh is a subset of the 3D mesh that is bounded by a pair of contour lines.

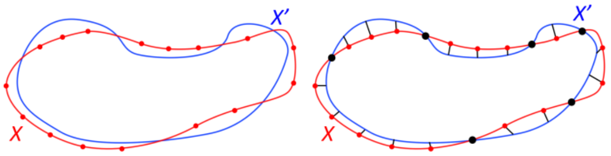

Figure 3. Contour matching for generating a source-target point set.

Handling 2D Contour Topological Changes

A 2D contour topological change occurs when there is no 1-to-1 inter-frame matching of contours during the contour matching process. This contour slice is then referred to as the *incident* slice. For such situations, we first determine the direction of the vertical motion that is occurring at the contour, then locate the ridge feature, and finally apply a 2D sine-based deformation function centralized at the ridge feature. The goal is to make sure that there is a 1-to-1 inter-frame matching of all the contours.

In order to determine the direction of the vertical motion, we first need to detect the location of the ridge in both frames where the 2D contour topological change is deemed to have occurred. The location of the ridge can be easily determined by noting where a many-to-one (or one-to-many) intra-frame connectivity exists. By comparing the location of the ridge between the two adjacent frames, we can then determine the direction of the vertical motion, i.e., moving upwards or downwards. For the case of upward vertical motion, the slice below the *incident* slice is referred to as the *complement* slice. For the case of a downward vertical motion, the *complement* slice is the slice above the *incident* slice.

A ridge feature is a path described by a sequence of edges that are roughly equidistant from the two contours. The desired effect is to apply a stronger vertical deformation at the source of the branching, while gradually reducing the strength as we move further away from it. We use a 2D sine-based function, with its center localized along the ridge feature, as the vertical deformation function. To extract the ridge, we have to first locate the two end points of the ridge. Assuming that the *incident* slice contains contour *A* and the *complement* slice contains two contours, *B* and *C*, we proceed to project all the three contours onto a common plane. For each vertex point on A, we compute its nearest distance to both B and C. For vertex points where its nearest distance to both B and C do not differ by more than 10%, we term it as the middle band (see Figure 4). Typically, two contiguous and disjointed middle bands will be formed. However, in the unlikely case where multiple contiguous middle bands are formed, we select the two largest bands and discard the rest. A point in each band is identified such that the difference between its distance to both B and C is the smallest. These two points are then selected as the end points of the ridge. The ridge is represented by the geodesic path computed between these two end points. A simple greedy algorithm that always select the next edge that result in closer distance to the end location is sufficient to extract the ridge. Figure 4 shows an idealized image of the ridge feature that is formed between two adjacent slices.

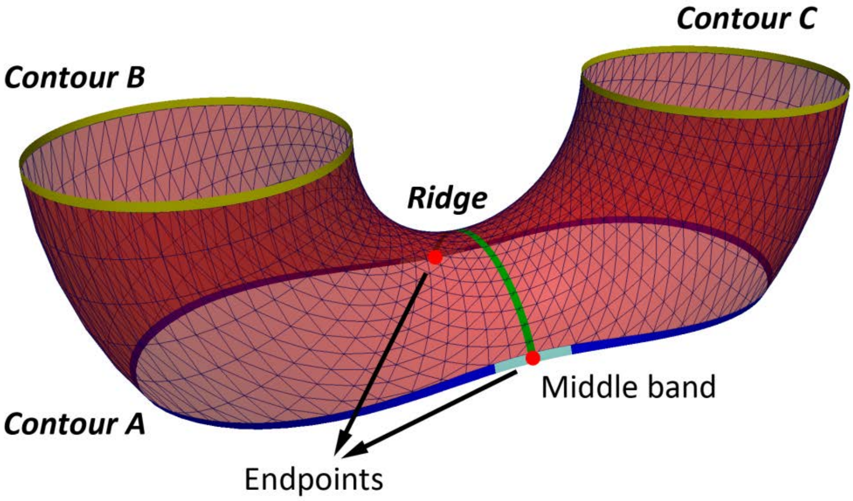

Figure 4. Ridge Detection.

To induce smooth vertical deformation, we have to expand the affected regions to include other sub-meshes around the incident slice. This allows the deformation to gradually spread its effect across several slices, thereby creating a smoother transition to the next frame. Hence, we considered two layers of adjacent sub-meshes above and below the incident sub-mesh. In total, six different slices are affected, labeled from 1 (topmost) to 6 (bottom). The deformation employs a 2D sine function, with its center located along the ridge feature. This allows us to apply the largest deformation at the ridge feature, while gradually reducing its effects, as we move away from the ridge. We proceed to project all the affected contours located on the six slices onto a common perpendicular plane, together with the ridge feature. On this perpendicular projection, the parameter λ is set to be the furthest 2D distance between the ridge and any point on the incident slice. We then use a 2D sine function to compute the amount of vertical displacement VD_p of each point p on the contour as a function of its 2D distance to the ridge D_{pr}.

$$VD_p = \begin{cases} \delta\left(\frac{R}{2}\right)\left[\sin\left(\pi\left(\frac{2D_{pr}+\lambda}{2\lambda}\right)\right)+1\right] & if\ D_{pr} < \lambda \\ 0 & if\ D_{pr} \geq \lambda \end{cases}$$

The value R acts as a control variable to adjust the amount of deformation experienced by each individual contour. The parameter δ refers to the inter-slice distance. An R value of 1 shifts a vertex by δ upwards, while an R value of -1 does the opposite. By setting R independently, as illustrated in Table 1, we can then adjust and smooth the effect of the deformation across the slices. After inducing the vertical motion, a 1-to-1 inter-frame correspondence can be established for each slice. We can then proceed using the standard deformation process to deform the mesh into the next time step.

Table 1: Listing of applied R values.

Slice	*R* Value	
Position	Upward	Downward
1	0.2	-0.2
2	0.4	-0.5
3	0.7 (Incident)	-1.2 (Complementary)
4	1.2 (Complementary)	-0.7 (Incident)
5	0.5	-0.4
6	0.2	-0.2

Results and Discussion

We implemented the 4D heart deformation algorithm in C++ (with no multi-threading optimization) on an i7 core 3.07 GHz machine and tested it on a set of MRI scan of the right heart of a healthy patient. The MRI data was acquired using the steady state free precession sequence with retrospective electrocardiographic gating, and consists of 25 contiguous image slices 5 mm apart covering the mid of the right atrium to the apex of the right ventricle, with 25 frames per slice per heartbeat. The contours of the endocardial surface are segmented manually by a cardiologist using the CMRTools software developed by Cardiovascular Imaging Solutions Ltd.

Reconstructing a 4D Heart Model

Using the MRI data from the first frame, we reconstruct a 3D mesh model of the right heart (17012 vertices with 34020 triangles). Using this 3D mesh model, we generate a 4D right heart model based on the deformation algorithm. Each of the 25 3D mesh model conforms to the corresponding input segmented contours lines and maintains vertex correspondence throughout. In Figure 5, a snapshot of the sequence of the 4D right heart model taken at different time steps is shown. We track the quality of the mesh model based on the conformity to the input contours and the quality of the surface triangles over the 25 time steps.

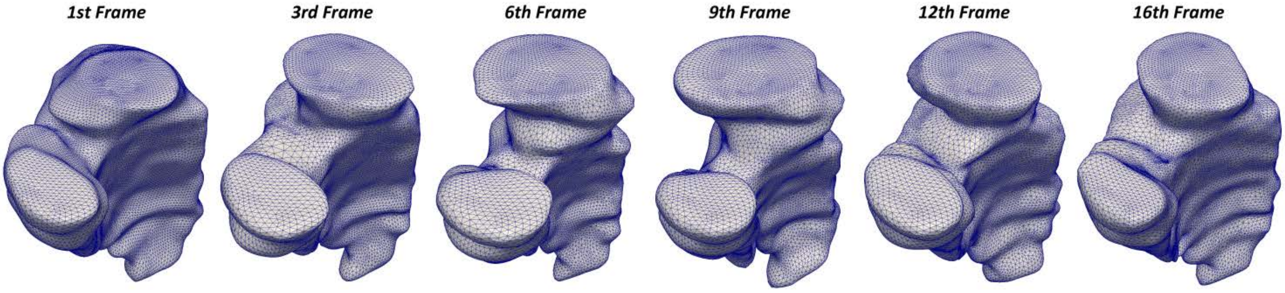

Figure 5. 4D Right Heart Model Reconstruction.

For tracking conformity, we sampled points along the input segmented contours at every 0.2mm interval and recorded its nearest distance (geometrical deviation) to the 3D mesh model. For each frame, we record the average geometrical deviation over all the input contours. Finally, we tabulate the mean and standard deviation of the averaged geometrical deviation over all the frames. The mean and standard deviation are 0.044mm and 0.003mm respectively. Taken as a percentage of δ (5mm) for this dataset, this equate to around a mean deviation of around 0.89%, which is a very low percentage value and is unnoticeable visually.

In terms of mesh quality, we can see from Table 2 that the majority of the triangles are within an acceptable quality range of between 40-80 degrees. Even as the 3D mesh deforms over the cardiac cycle and undergoing 2D contour topological changes with significant deformation, the percentage of acceptable quality triangles remains stable throughout (85% to 89%). On the other end of the spectrum, the number of bad quality triangle (< 20 degrees) occurs at a very low rate of 0.88 per frame, or 0.002% of the total triangle count.

Table 2: Distribution of Triangle Angle Quality.

Dataset No		% of angles (in degrees) between				
		40-50	50-60	60-70	70-80	Total
1	First Frame	15.42%	30.67%	28.56%	15.01%	89.67%
	Mid Frame	17.07%	28.52%	26.76%	14.67%	87.02%
	Last Frame	18.16%	27.24%	25.84%	14.37%	85.61%

2D Contour Topological Changes

From the dataset, 2D contour topological changes are found to occur at 4 different time steps. We illustrate how our algorithm is able to resolve it in a series of screen shots in Figure 6. In the figure, a red plane depicting the image scanning plane of the MRI, is shown to intersect with the 3D model. We observed the phenomenon of two distinct contours merging and splitting on the red plane, which is consistent with the drawn segmented contours in the dataset.

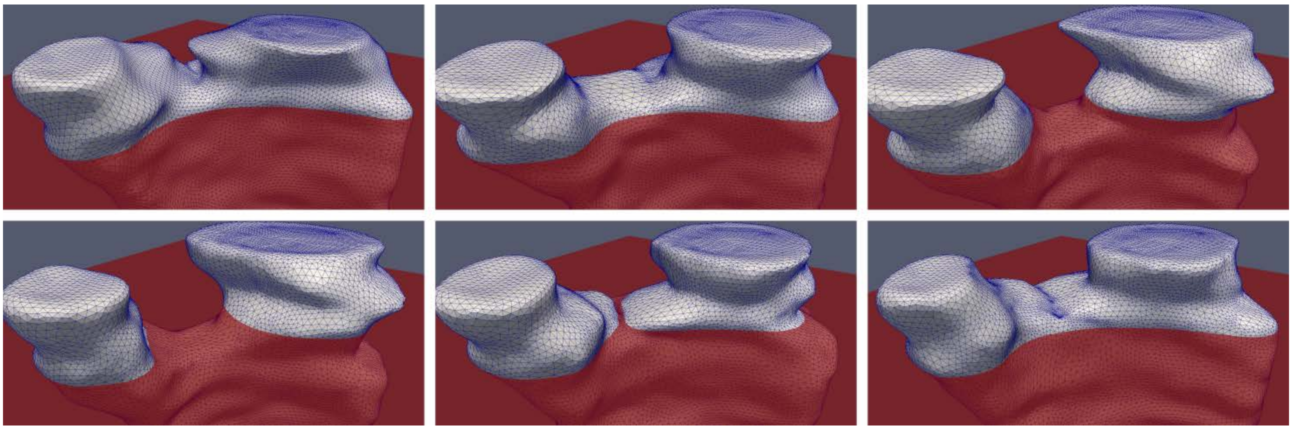

Figure 6. Resolving 2D Contour Topological Changes.

Computation Timing

Generally, the average computational time is around 6.4 milliseconds per vertex for normal frames, and 7.8 milliseconds per vertex for frames with 2D contour topological changes. There is little timing variability among frames, with a standard deviation of 0.3 milliseconds and 0.17 milliseconds respectively. Typically, for a 3D mesh model with a vertex count of 20000, it would take approximately 2 minutes to complete a frame. To generate a 4D right heart model consisting of 20 frames would take around 40 minutes to complete, which is considered to be adequate for practical usage.

Conclusion

In this paper, we presented a methodology for the automatic 4D reconstruction of a patient specific cardiac mesh model using segmented contour lines from MRI images. Our methodology is able to handle the inter-frame 2D contour topological changes that occur during the cardiac cycle and produces a sequence of good quality 3D mesh models with 1-to-1 vertex correspondence that conforms to the input contours. Such a model is useful for downstream computational simulation purposes.

References

Frangi AF, Niessen WJ, Viergever MA (2001) Three-dimensional modeling for functional analysis of cardiac images: A review. IEEE Transaction on Medical Imaging 20: 2–25.

McQueen DM, Peskin CS (2000) A three-dimensional computer model of the human heart for studying cardiac fluid dynamics. ACM SIGGRAPH Computer Graphics 34: 56–60.

Kovάcs SJ, McQueen DM, Peskin CS (2001) Modelling cardiac fluid dynamics and diastolic function. Phil Trans R Soc Lond 359: 1299–1314.

Corsi C, Saracino G, Sarti A, Lamberti C (2002) Left ventricular volume estimation for real-time three-dimensional echocardiography. IEEE Transaction on Medical Imaging 21: 1202–1208.

Zhukov L, Bao Z, Guskov I, Wood J, Breen D (2002) Dynamic deformable models for 3D MRI heart segmentation. Proceedings of SPIE 4684: 1398–1405.

Montagnat J, Delingette H (2005) 4D deformable models with temporal constraints: Application to 4D cardiac image segmentation. Medical Image Analysis 9: 87–100.

Sermesant M, Forest C, Pennec X, Delingette H, Ayache N (2003) Deformable biomechanical models: Application to 4D cardiac image analysis. Medical Image Analysis 7: 475–488.

Plank G, Burton RAB, Hales P, Bishop M, Mansoori T, et al. (2009) Generation of histoanatomically representative models of the individual heart: tools and application. Philosophical Transactions of the Royal Society A 367: 2257–2292.

Barequet G, Sharir M (1996) Piecewise-linear interpolation between polygonal slices. Computer Vision and Image Understanding : 251–272.

Prasad L (2007) Rectification of the chordal axis transform skeleton and criteria for shape decomposition. Image and Vision Computing 25: 1557–1571.

Roussos G, Baxter BJC (2005) Rapid evaluation of radial basis functions. Journal of Computational and Applied Mathematics 180: 51–70.