A Sketch-based Interface for Modeling Myocardial Fiber Orientation that Considers the Layered Structure of the Ventricles

KENSHI TAKAYAMA[1], TAKASHI ASHIHARA[3], TAKASHI IJIRI[1], TAKEO IGARASHI[1, 2], RYO HARAGUCHI[4], and KAZUO NAKAZAWA[4]

[1]Department of Computer Science, the University of Tokyo; [2]JST/ERATO;
[3]Department of Cardiovascular Medicine, Heart Rhythm Center, Shiga University of Medical Science; and
[4]National Cardiovascular Center Research Institute

Abstract: We propose a sketch-based interface for modeling the myocardial fiber orientation required in the electrophysiological simulation of the heart, especially the ventricles. The user can create a volumetric vector field that represents the myocardial fiber orientation in two steps. First, the user defines a depth field over the three-dimensional (3D) ventricular model to create layers of myocardium. Then, the user can peel these layers and draw strokes on them to specify the myocardial fiber orientation in each layer. We represent the 3D ventricular model as a tetrahedral mesh and perform Laplacian smoothing over the mesh vertices to interpolate the vector field defined by the user-drawn strokes. Our method also allows the user to perform deformations on volumetric models of myocardial fiber orientation, which is very important for studying heart disease associated with morphological abnormalities. We created several examples of myocardial fiber orientation and applied them to a simplified simulator to demonstrate the effectiveness of our method.

Key words: electrophysiological heart simulation, myocardial fiber orientation, sketch-based interface.

––––

To better understand the mechanisms of fatal cardiac arrhythmias, we previously developed a large-scale heart simulator that calculates the excitation propagation through a realistic three-dimensional (3D) human ventricular model called “Virtual Heart” [1]. A supercomputer was used to solve a large system of differential equations containing about 45 million variables, which represented the behavior of 5.64 million volumetric units of myocardium. Various electrophysiological phenomena were visualized using this simulation system. With the rapid growth of computer hardware technology, the speed and accuracy of the simulation will increase in the future.

However, creating a realistic 3D ventricular model is still difficult and time-consuming because the model consists of various parameters such as the ion channel activities, the excitation sequences of the subendocardial layers, and the myocardial fiber orientation, which is especially important because the speed of excitation propagation largely depends on it. The speed is about three times faster in the longitudinal direction of myofibers than in the transverse direction. In our simulation system [1], the myocardial fiber orientation was hard-coded and could not be modified easily by the user.

To resolve this issue, we subsequently developed a sketch-based interface for modeling myocardial fiber orientation [2]. In this system, the user can create a volumetric vector field that represents the myocardial fiber orientation easily and quickly by drawing strokes on the surfaces or cross sections of the 3D ventricular model. However, a user study revealed a limitation of the method: since the ventricles consist of many myocardial layers [3], it is not intuitive for the user to draw strokes on cross sections of the 3D ventricular model to specify the internal fiber orientation. Instead, according to a user’s comment, it would be better if the user could peel the layers of myocardium and draw strokes on each layer to specify the internal fiber orientation.

Based on this user feedback, we propose a sketch-based interface for modeling myocardial fiber orientation that allows the user to peel the layers of myocardium and draw strokes on each layer to specify the myocardial fiber orientation. Our method also makes it possible to deform the entire volumetric model of myocardial fiber orientation, which is very important for studying morphological abnormalities, such as hypertrophic cardiomyopathy, dilated cardiomyopathy, and ventricular aneurysm. We created several models of myocardial fiber orientation using our system and applied them to a simplified electrophysiological heart simulator [4] to demonstrate the effectiveness of our system.

METHODS

System overview. The input to the system is the tetrahedral mesh of the 3D ventricular model, and the output is a volumetric vector field that represents the myocardial fiber orientation. The input mesh model was obtained as follows: the shape of a living human heart was obtained using magnetic resonance imaging (MRI) data that was converted into a polygonal mesh using Marching Cubes [4]. Then, this was simplified and tetrahedralized using TetGen library [5]. The user can create the myocardial fiber orientation embedded in this tetrahedral mesh model in two steps: by defining a depth field and specifying the fiber orientation. The user can also deform the shape of the 3D ventricular model after modeling the myocardial fiber orientation. We describe the details of each in the following.

Defining a depth field. The ventricles consist of myocardial layers, as shown in Fig. 1 [3]. To create such layers, the user defines a depth field inside the 3D ventricular model by putting several points on the model that correspond to four regions of the ventricles: the epicardial layers, septal myocardial layers between the left and right ventricles, right ventricular free wall, and left ventricular free wall. Each region is associated with a color (red, yellow, green, and blue, respectively), and the user can put colored points on the 3D ventricular model by clicking the mouse on the surfaces (Fig. 2a). If the position where the user wants to put a point is hidden by other parts of the model, the user can cut off such parts by drawing a free-form stroke across the 3D ventricular model (Fig. 2b) and can place the point on the desired position (Fig. 2c).

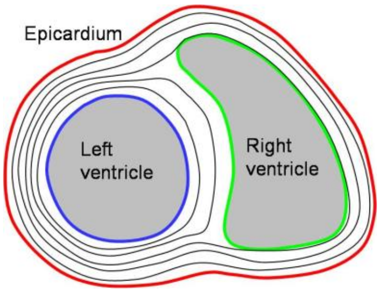

Fig. 1. The myocardial layers of the ventricles.

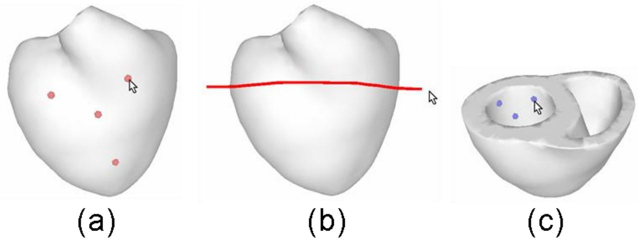

Fig. 2. The user puts several points on the surfaces of the 3D ventricular model by clicking the mouse (a). Cutting the model (b) allows the user to put points on the internal regions (c).

After putting several points on the 3D ventricular model (Fig. 3), the user asks the system to compute a smoothly varying depth field inside the 3D ventricular model. The system assigns each of these user-specified points a depth value corresponding to the region (0, 0.5, 1, and 2, respectively), and uses them as constraints to compute a smoothly varying scalar depth field in the 3D space by using thin plate spline interpolation [6]. The resulting depth field is visualized as a smoothly varying color distribution (Fig. 4a). The system also generates the surfaces of the myocardial layers by extracting the iso-surfaces from the depth field using Marching Tetrahedra [7] (Fig. 4b). If the resulting depth field is not satisfactory, the user can add or remove points and let the system recompute the depth field in a trial-and-error manner.

Specifying the myocardial fiber orientation. Next, the user specifies the myocardial fiber orientation by drawing strokes on each myocardial layer. The user chooses a layer of myocardium with the mouse wheel and draws strokes on it by dragging the mouse (Fig. 5a). Cutting the model, similar to Fig. 2c, allows the user to draw intramural myocardial fiber orientation as well. The sys-

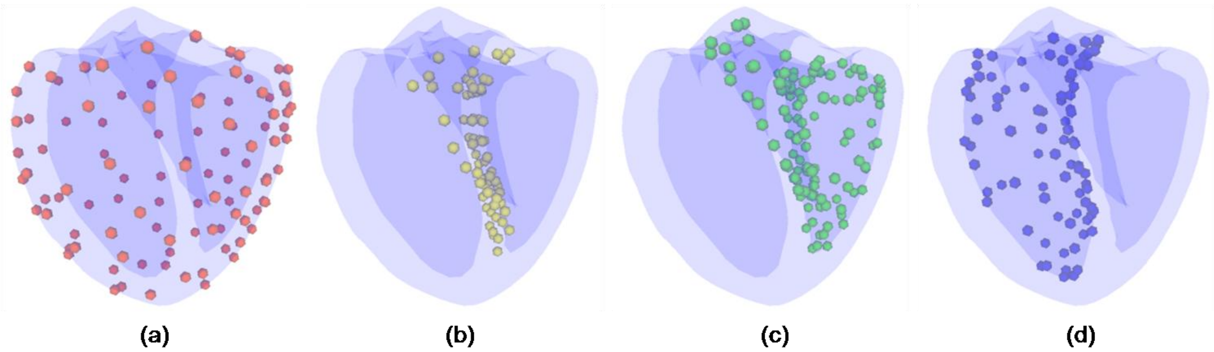

Fig. 3. User-specified points on the four parts of the ventricles used to construct the depth field: the (a) epicardial layer (depth 0), (b) septal layer (depth 0.5), (c) right ventricle (depth 1), and (d) left ventricle (depth 2).

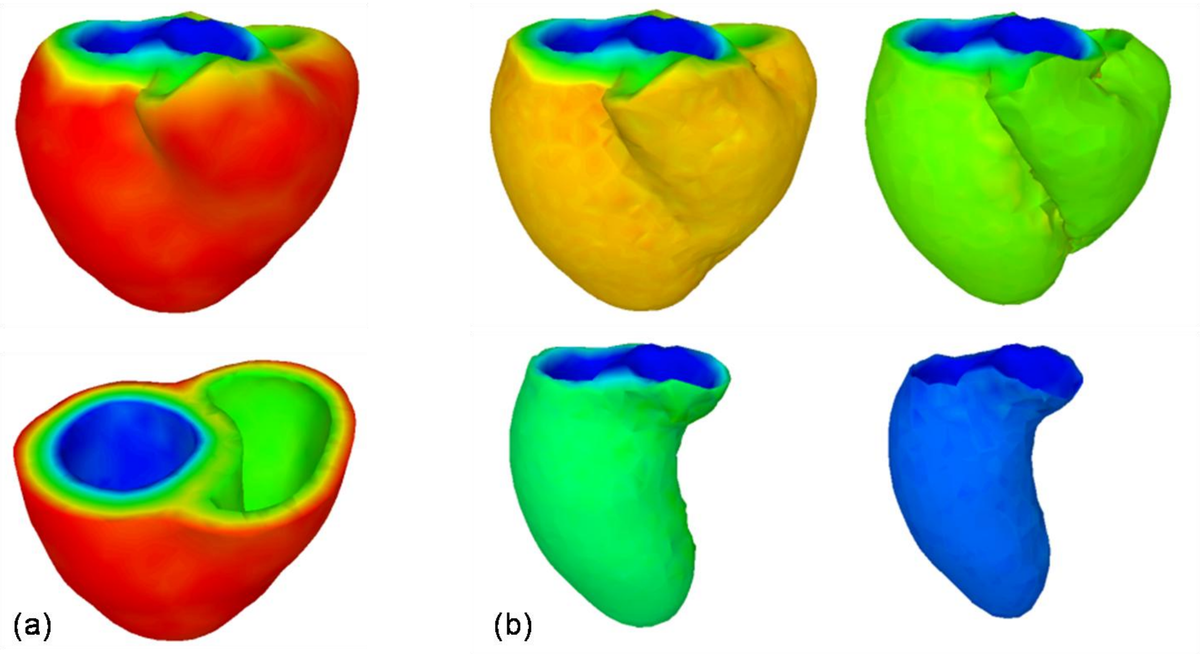

Fig. 4. (a) Computed depth field visualized as a smoothly varying color distribution. (b) The layers of ventricular myocardium extracted from the depth field.

tem then interpolates these user-drawn strokes to obtain the myocardial fiber orientation varying smoothly inside the 3D ventricular model. The system also generates many streamlines running along the fibers to visualize the interpolation result (Fig. 5b).

Laplacian smoothing. To interpolate the vector field defined by the user-drawn strokes, we perform Laplacian smoothing [8] on the tetrahedral mesh vertices for each *x*-, *y*-, and *z*-component of the vector field. This technique minimizes the differences between the values assigned to the mesh vertices while satisfying the user-specified constraints as much as possible.

More precisely, suppose we are solving for the *x*-component x_i assigned to each vertex $\mathbf{v}_i \ (i = 1, \ldots, n)$. The Laplacian δ_i is defined as

$$\delta_i = x_i - \sum_{j \in N_i} w_j^i x_j \quad , \qquad (1)$$

where N_i is the index set of one-ring neighboring vertices of $\mathbf{v}_i$, and w_j^i represents the corresponding weight. To make the interpolation result smoother on the same layer than among different layers, we set the weights as

$$w_j^i = \frac{\exp(-\lambda_r r_{ij}^2 - \lambda_d d_{ij}^2)}{\sum_{k \in N_i} \exp(-\lambda_r r_{ik}^2 - \lambda_d d_{ik}^2)} \quad , \qquad (2)$$

where r_{ij} is the Euclidean distance between $\mathbf{v}_i$ and $\mathbf{v}_j$, d_{ij} is the difference between the depth values sampled at $\mathbf{v}_i$ and $\mathbf{v}_j$, and λ_r and λ_d are predefined constants that control the smoothness in the depth direction (currently, we set them to 1 and 5, respectively).

The goal is to minimize all these Laplacians while satisfying the user-specified constraints. As the constraints are given in the form of user-drawn strokes and each stroke consists of a series of small segments, we treat each segment as a normalized constraint vector defined at a discrete point in the 3D space. When constraint vector $\mathbf{c}$ is given at 3D position $\mathbf{p}$, we search for tetrahedron T in the mesh that is closest to $\mathbf{p}$. The constraint is then given in the form of a weighted sum as

$$w_1 x_{i_1} + w_2 x_{i_2} + w_3 x_{i_3} + w_4 x_{i_4} = c_x, \qquad (3)$$

where $i_1, \ldots, i_4$ are the indices of the four vertices of T and c_x is the *x*-component of $\mathbf{c}$. The weights $w_1, \ldots, w_4$ are given as

$$w_j = \frac{\exp(-\lambda_r r_j^2 - \lambda_d d_j^2)}{\sum_{k=1}^{4} \exp(-\lambda_r r_k^2 - \lambda_d d_k^2)} \quad , \qquad (4)$$

where r_j is the Euclidean distance between $\mathbf{p}$ and $\mathbf{v}_{i_j}$, and d_j is the difference between the depth values sampled at $\mathbf{p}$ and $\mathbf{v}_{i_j}$.

Minimizing all the Laplacians (Eq. 1) while satisfying the collection of the constraints (Eq. 3) in a least squares sense forms a sparse linear system, which can be solved quickly by LU decomposition (we use UMFPACK library [9]). We perform a similar procedure for the *y*- and *z*-components, and finally combine them into normalized vectors.

Converting tetrahedra into voxels. Since the vector field obtained above is only defined at the positions of the tetrahedral mesh vertices, we compute the volumetric vector field by converting tetrahedra into voxels. For each tetrahedron in the mesh, we first scan it in the *x*-, *y*-, and *z*-directions to obtain a set of voxels that belong to the tetrahedron. Then, we assign each voxel a depth value that is interpolated linearly from the depth values assigned to the four vertices of the tetrahedron to which it belongs. Finally, the vector value assigned to each voxel is computed as the weighted sum of the vectors assigned to the four vertices of the tetrahedron to which it belongs, where the weights are similar to Eq. 2. After obtaining the volumetric vector field, we can apply various existing visualization techniques such as streamlines (Fig. 5b) [10]. The volumetric data, where each voxel contains the *x*-, *y*-, and *z*-components of myocardial fiber orientation vector, can be exported to other applications, and can also be

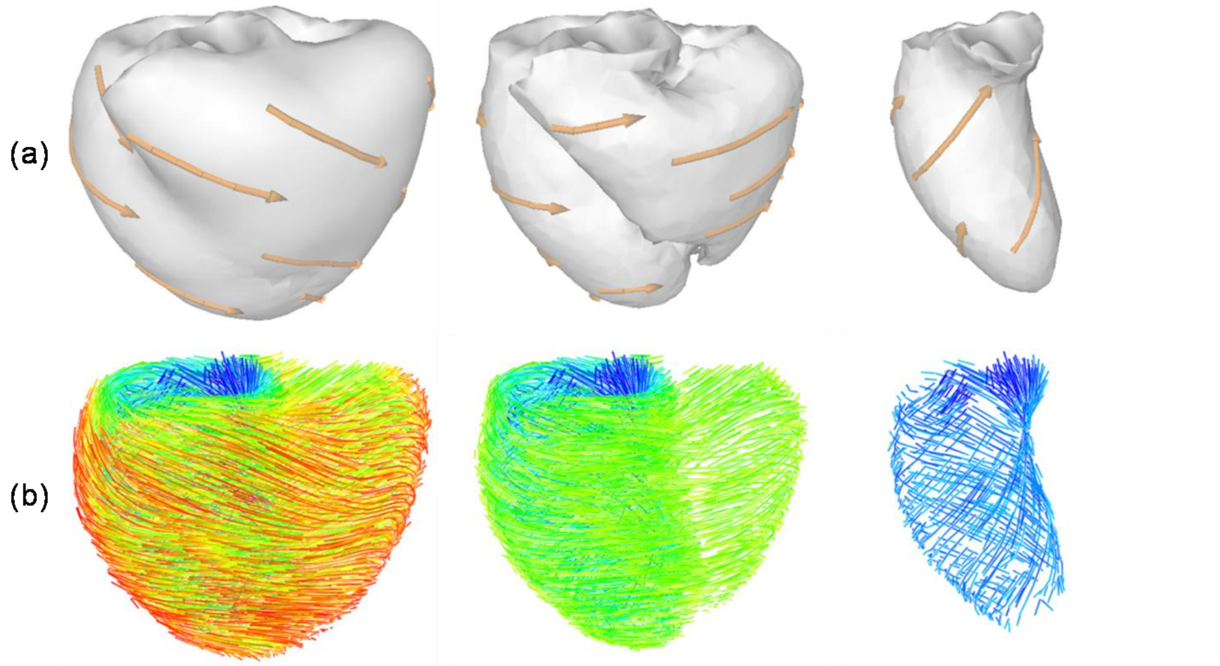

Fig. 5. (a) Drawing strokes on the layers of ventricular myocardium. (b) Streamlines visualizing the interpolated myocardial fiber orientation in the ventricles.

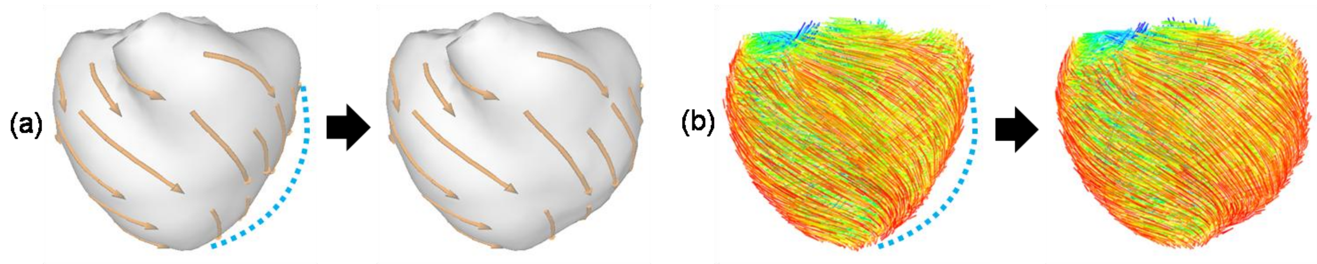

Fig. 6. Deforming the model. (a) The strokes are transformed according to the deformation. (b) The volumetric model is recomputed.

converted into any specific data format available for electrophysiological simulations in the ventricles.

Deforming the shape. We can deform the entire shape of the volumetric model after creating the myocardial fiber orientation. This is accomplished by first deforming the tetrahedral mesh and then recomputing the deformed volumetric model of the myocardial fiber orientation. In our current prototype system, the user deforms the tetrahedral mesh by moving the individual mesh vertices manually. However, it is possible to use other techniques of deforming tetrahedral meshes [11].

After the tetrahedral mesh is deformed, we transform each of the user-drawn strokes into the corresponding deformed position using a piecewise linear mapping $\phi : \mathbf{R}^3 \mapsto \mathbf{R}^3$, which maps an arbitrary 3D position $\mathbf{p}$ in the original model onto the corresponding position $\mathbf{p}' = \phi(\mathbf{p})$ in the deformed model. To compute $\mathbf{p}'$, we search for the tetrahedron T in the original mesh that is closest to $\mathbf{p}$ and compute the barycentric coordinates $\lambda_1, \dots, \lambda_4$ as

$$\begin{aligned} \lambda_1 \mathbf{v}_{i_1} + \lambda_2 \mathbf{v}_{i_2} + \lambda_3 \mathbf{v}_{i_3} + \lambda_4 \mathbf{v}_{i_4} = \mathbf{p} \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \end{aligned} \quad (5)$$

where $i_1, \dots, i_4$ are the indices of the four vertices of T. We can then calculate $\mathbf{p}'$ as

$$\mathbf{p}' = \lambda_1 \mathbf{v}'_{i_1} + \lambda_2 \mathbf{v}'_{i_2} + \lambda_3 \mathbf{v}'_{i_3} + \lambda_4 \mathbf{v}'_{i_4}, \quad (6)$$

where $\mathbf{v}'_i$ is the deformed position of $\mathbf{v}_i$.

We transform all the user-drawn strokes appropriately using this map (Fig. 6a), and perform Laplacian smoothing again to obtain the deformed volumetric model of the myocardial fiber orientation (Fig. 6b). Note that this deformation technique is applicable not only to the myocardial fiber orientation, but also to other spatially varying parameters such as the ion channel activities and excitation sequence of the subendocardial layers.

RESULTS

As shown in Fig. 7, we created three sample models of myocardial fiber orientation and applied them to the simplified simulator that we had developed previously [12]. The models shown in Fig. 7a and 7b have the same 3D shape, but their myocardial fiber orientations differ. We can observe the influence of this difference on the excitation propagation in the simulation result. The model shown in Fig. 7c was created by deforming the model shown in Fig. 7a. The simulation result shows that the myocardial fiber orientation was mapped appropriately from the original model to the deformed one.

Our current prototype system is implemented using C++ and OpenGL on a notebook PC with a 2.33-GHz CPU and 1.0 GB of RAM. The tetrahedral mesh we used consists of 3,162 vertices and 13,157 tetrahedra. The total computation time, including the construction of the depth

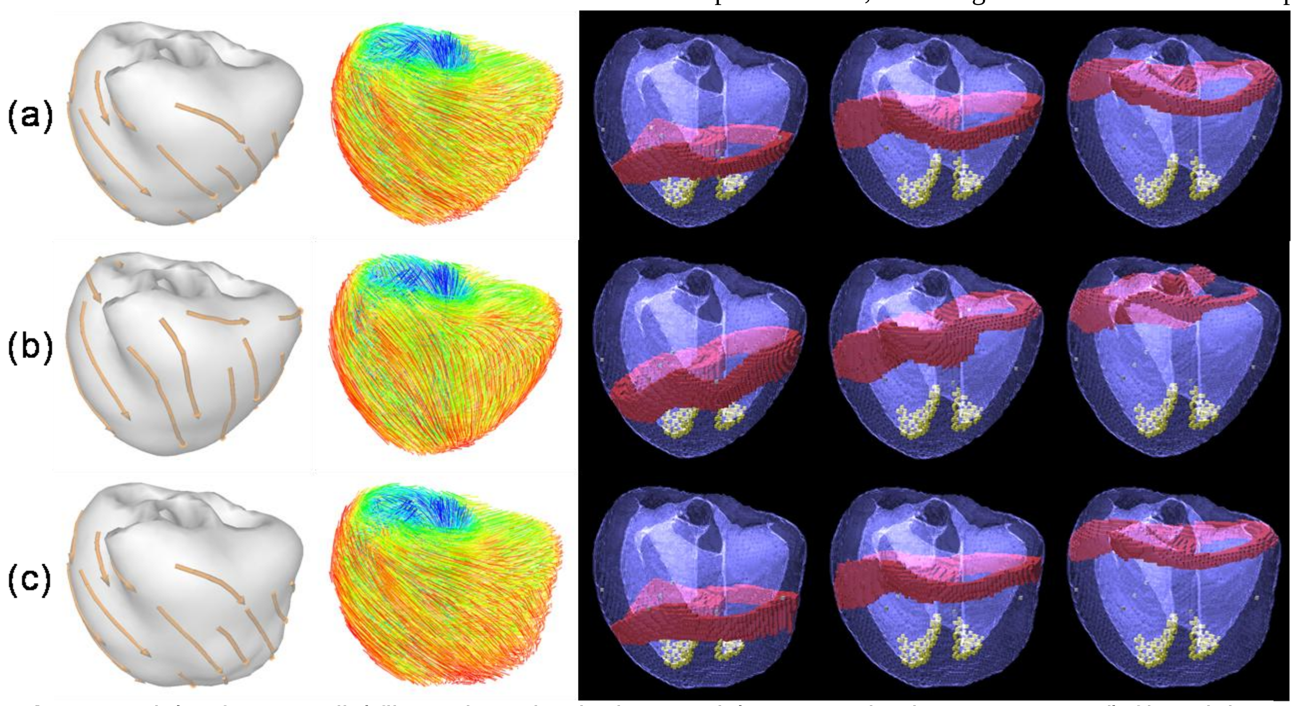

Fig. 7. Models of myocardial fiber orientation in the ventricles created using our system (left) and the simulation results (right).

field, interpolation of the user-drawn strokes, and conversion of tetrahedra into voxels, was less than 5 seconds.

DISCUSSION

Since the user can create models of myocardial fiber orientation quickly and easily using our system, our method facilitates realistic 3D heart simulation with the goal of clinical application. Our method is also effective for studying the influence of the myocardial fiber orientation on the excitation propagation by comparing the simulation results of various heart models with the same 3D shape, but with different myocardial fiber orientations. Finally, our technique of deforming volumetric models allows the user to obtain a variety of heart models with different 3D shapes effortlessly by deforming a single base model with predefined parameters. This makes it much easier to study the influence of the morphological change in the ventricles on the excitation propagation.

There are some limitations to our method. One serious drawback is that the distribution of the myocardial fibers in the real world is so complicated that our method might not be able to create a precise model. Several regions in the heart, such as the cardiac apex and septal myocardial layers between the left and right ventricles, have very complex distributions of myocardial fibers [13, 14]. It might be necessary to develop more sophisticated user interfaces to deal with these specific regions.

Another issue, which is also related to the above, is that the myocardial fibers in the real world actually have no distinction between the front and back orientations, which implies that there can be some regions in the ventricles where the myocardial fiber orientation cannot be represented by smoothly varying vector field. Tensor field representation might be more appropriate for our purpose, because it can represent certain orientation fields that cannot be represented by smoothly varying vector field, as shown in Fig.8. Zhang et al. proposed an interactive design tool for two-dimensional (2D) tensor field that can create orientation fields such as those shown in Fig. 8 [15], but extending their technique to 3D tensor field is not trivial.

Finally, we need to investigate how to support the modeling process. Currently the user has to design the myocardial fiber orientation from scratch based only on his or her expert knowledge. Since it has become possible to obtain the myocardial fiber orientation to some extent using diffusion tensor MRI [16], we will be able to help the user's modeling process with such data. We plan to continue working on these issues.

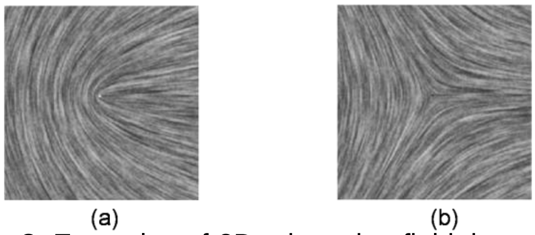

Fig. 8. Examples of 2D orientation field that cannot be represented by smoothly varying vector field: the (a) wedge and (b) trisector.

The first author was funded by the Information-technology Promotion Agency (IPA). This study was partially funded by the Ministry of Education, Culture, Sports, Science and Technology (Japan), a Grant-in-Aid for Scientific Research (B) 18300157, and the Leading Project for Biosimulation.

REFERENCES

1. Nakazawa K, Suzuki T, Ashihara T, Inagaki M, Namba T, Ikeda T, Suzuki R. Computational analysis and visualization of spiral wave reentry in a virtual heart model. In: Yamaguchi T, editor. Clinical application of computational mechanics to the cardiovascular system. Springer; 2000. p. 217-241.
2. Takayama K, Igarashi T, Haraguchi R, Nakazawa K. A sketch-based interface for modeling myocardial fiber orientation. In: Butz A, Fisher B, Krueger A, Olivier P, Owada S, editors. Smart graphics. Springer; 2007. p. 1-9. doi:10.1007/978-3-540-73214-3_1.
3. Nielsen PM, Le Grice IJ, Smaill BH, Hunter PJ. Mathematical model of geometry and fibrous structure of the heart. Am J Physiol Heart Circ Physiol. 1991;260:H1365-H1378.
4. Lorensen WE, Cline HE. Marching cubes: a high resolution 3d surface construction algorithm. SIGGRAPH Comput Graph. 1987;21(4):163-169. doi:10.1145/37402.37422.
5. Si H. On refinement of constrained Delaunay tetrahedralizations. In: Pebay PP, editor. Proceedings of the 15th international meshing roundtable. Springer; 2006. p. 509-528. doi:10.1007/978-3-540-34958-7_29.
6. Turk G, O'Brien JF. Shape transformation using variational implicit functions. In: SIGGRAPH '99: proceedings of the 26th annual conference on computer graphics and interactive techniques. ACM Press; 1999. p. 335-342. doi:10.1145/311535.311580.
7. Treece GM, Prager RW, Gee AH. Regularised marching tetrahedra: improved iso-surface extraction. Computers and graphics. 1999;23(4):583-598.
8. Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rössl C, Seidel H-P. Laplacian surface editing. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on geometry processing. ACM; 2004. p. 175-184. doi:10.1145/1057432.1057456.
9. Davis TA. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. , ACM Trans Math Softw. 2004;30(2):165-195. doi:10.1145/992200.992205.
10. Zhukov L, Barr AH. Heart-muscle fiber reconstruction from diffusion tensor MRI. In: Proceedings of the 14th IEEE Visualization. IEEE Computer Society;2003. p. 597-602. doi:10.1109/VISUAL.2003.1250425.
11. Müller M, Dorsey J, McMillan L, Jagnow R, Cutler B. Stable real-time deformations. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on computer animation. ACM; 2002. p. 49-54. doi:10.1145/545261.545269.
12. Haraguchi R, Igarashi T, Owada S, Yao T, Namba T, Ashihara T, Ikeda T, Nakazawa K. Electrophysiological heart simulator equipped with sketchy 3-d modeling. In: Wu JL, Ito K, Tobimatsu S, Nishida T, Fukuyama H, editors. Complex medical engineering. Springer; 2007. p. 107-116. doi:10.1007/978-4-431-30962-8_10.
13. Ashihara T, Namba T, Yao T, Ozawa T, Kawase A, Ikeda T, Nakazawa K, Ito M. Vortex cordis as a mechanism of postshock activation: arrhythmia induction study using a bidomain model. J Cardiovasc Electrophysiol. 2003;14(3):295-302. doi:10.1046/j.1540-8167.2003.02408.x.
14. Ikeda T, Kawase A, Nakazawa K, Ashihara T, Namba T, Takahashi K, Sugi K, Yamaguchi T. Role of structural complexities of septal tissue in maintaining ventricular fibrillation in isolated, perfused canine ventricle. J Cardiovasc Electrophysiol. 2001;12(1):66-75. doi:10.1046/j.1540-8167.2001.00066.x.
15. Zhang E, Hays J, Turk G. Interactive tensor field design and visualization on surfaces. IEEE Trans Vis Comput Graph. 2007;13(1):94-107. doi:10.1109/TVCG.2007.16.
16. Scollan DF, Holmes A, Zhang J, Winslow RL. Reconstruction of cardiac ventricular geometry and fiber orientation using magnetic resonance imaging. Ann Biomed Eng. 2000;28(8):934–944. doi:10.1114/1.1312188.