GENERATION OF AN ATLAS-BASED FINITE ELEMENT MODEL OF THE HEART FOR CARDIAC SIMULATION

M.Sc. Vasiliev Evgeny
Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia
eugene.unn@gmail.com

Abstract: *In this paper an algorithm for creating an atlas-based finite element heart model for cardiac simulation is described. This model is used to simulate the propagation of electrical impulses of the heart. An important feature of this model is that it contains conductive paths and fibrous tissue, which makes it possible to make more realistic calculations of the propagation of electrical signals. The model created from anatomical segments of the heart surface is defined by a polygonal mesh. The algorithm presented in the article offers a means to create models of various accuracy.*

Keywords: HEART MODEL, CARDIAC SYSTEM, RECONSTRUCTION, RAY CASTING, FE-MODEL, TETRAHEDRAL MESH.

1. Introduction

In order to simulate an accurate electrical system of the human heart, it is necessary to have a detailed heart model containing different types of tissues: the direction of the muscle tissue, conducting fibers, fibrous rings, etc.

Nowadays there is a number of finite element models (FE models) of heart for modeling electricity signals. For example, in some systems of heart electrical modeling, the rabbit heart model is used [R. Bordasa, 2011] [Arevalo HJ, 2016]. The rabbit heart model was constructed from a high resolution MRI dataset, with the use of an intensity based level-set filter.

The first developed 3D models of cardiac anatomy were simplistic models based on geometric shapes, this approach is still in use for applications where the anatomical validity is not important for the purpose of the model [Colli Franzone P, 1998] [Sermesant M, 2006]. Currently, researches use simple geometric shapes and make parameterized models based on segmented heart images [P. Lamata et al, 2014].

In many articles, studies are based on models built on the segmentation of CT [Deng D, 2012] [Aslanidi OV, 2013] or MRI [Plotkowiak M, 2008] [Lopez-Perez & Sebastian, 2015] scans. Models of this type very anatomically accurate, but usually they describe only one heartbeat phase, rather than the whole cycle.

Our algorithm enables automatic marking of vertex the type of tissues after the grid generation, because manual marking of hundreds of thousands and more tetrahedrons makes no sense.

2. Source data

As the initial data, anatomical segments of heart surface from Plastic boy anatomy (Plastic boy store) were used. We have several types of meshes (Figure 1):

- the outer surface of the heart;
- left atrium;
- left ventricle;
- right atrium;
- right ventricle;
- mitral valve;
- tricuspid valve;
- atrioventricular node;
- sinoatrial node;
- His bundle;
- Bachmann bundle;
- Signal ways.

We use the heart surface polygonal model to generate a tetrahedral FE model. We can vary the total number of vertices and tetrahedrons in the FE model.

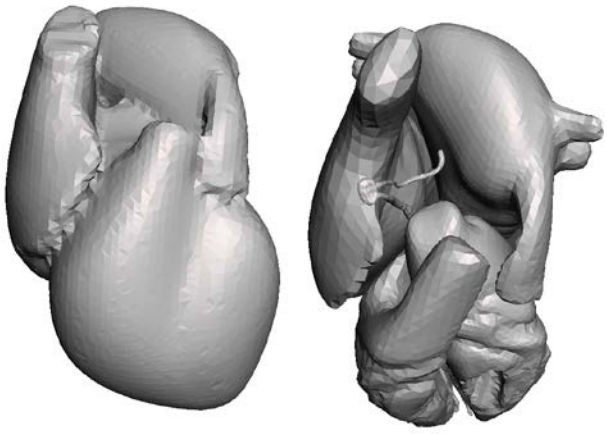

Fig. 1 *The outer surface of the heart and inner tissues.*

3. Algorithm of model generation

The algorithm consists of several parts: FE model generation; classification of atriums and ventricles; removal of cavities; cardiovascular system classification.

Step 1. Generation of the FE model.

Generation of a three-dimensional finite element mesh is labour-intensive process, that is why there are few high-quality open-source 3D mesh generators. We used Netgen for mesh generation, because Netgen is an open-source framework with LGPL v2 license, it is well-tested, and has its own user community. Netgen uses the outer surface of the heart in stl format, and yields a FE model consisting of tetrahedrons (Figure 2).

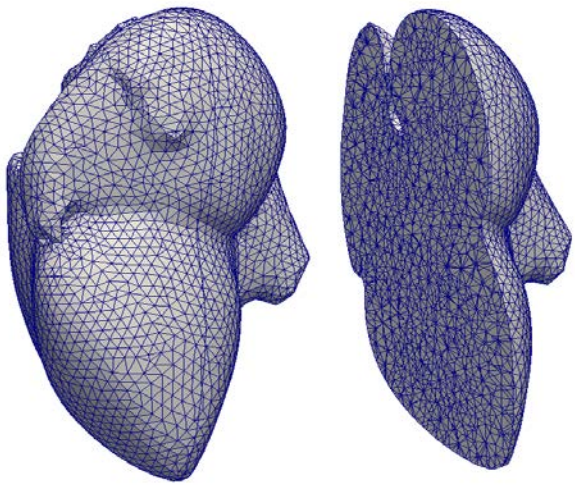

Fig. 2 *Generated FE-model*

Step 2. Removing tetrahedrons of atria and ventricles cavities.

After the first step we have the FE model, which consists of tetrahedrons inside atria and ventricles cavities and we delete them. To remove them we need to mark vertices for deleting. We mark vertices located inside tissue meshes (Figure 1) via the ray casting algorithm (J. D. Foley, 1990). A two-dimensional version of the algorithm is shown in Figure 3.

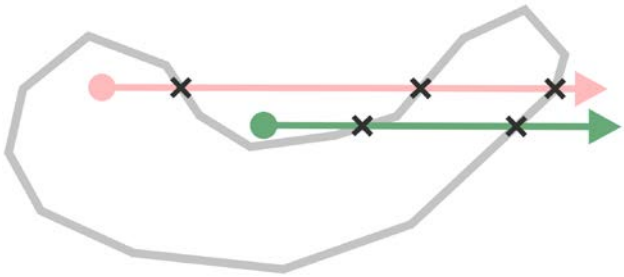

Fig 3 *Even-rule algorithm (2D schema). If the point is on the inside of the polygon then it will intersect the edge an odd number of times, and outside otherwise.*

We start the ray from a tested vertex in any fixed direction and calculate the number of intersections with heart tissue shells. If the point is outside of the shell the ray will intersect it an even number of times. If the point is inside the shape then the ray will intersect the edge an odd number of times.

Having tested vertices with all shapes, we can delete tetrahedrons consisting of marked vertices. After deleting we obtain the model shown in Figure 4.

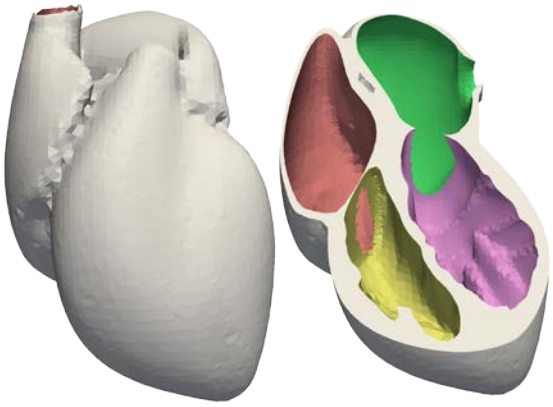

Fig. 4 *FE-model with removed cavities*

Step 3. Marking up the electrical conduction system of the heart.

To mark up the cardiac conduction system we use the ray casting algorithm for checking whether the point is inside or outside the shell again. As a result we have a model with marked atriums, ventricles and cardiac conduction system.

4. Parallel realization

The brunt of the calculation is marking points according to tissue type. We calculate it with the ray casting algorithm. This algorithm is good for parallel calculations, because we can test a lot of points with the same shell together. Parallel realization of the ray casting algorithm: we divide all vertices into blocks whose size equals size of the GPU thread block and perform calculations for blocks independently; inside the block, the ray casting algorithm is performed for all vertices simultaneously.

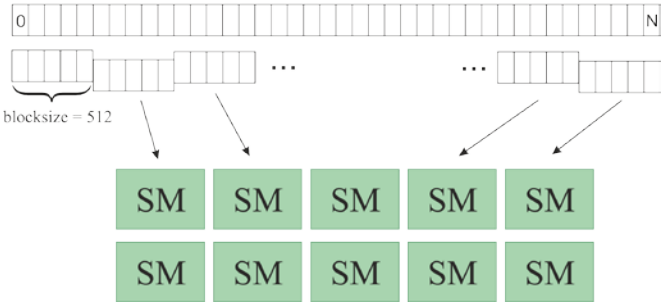

Fig. 5 *Data parallelization scheme for the ray casting algorithm. SM - Streaming Microprocessor*

Parallelization scheme is shown in Figure 5. All vertices are divided into blocks of the same size. On a GeForce GTX 680 (GK104), the optimal block size was found to be 512 threads. Scheduler allocates vertex blocks for processing on Streaming Multiprocessors automatically.

As the maximum length of the tetrahedron edge decreases, the number of vertices and tetrahedrons and the time for their processing increase exponentially (Figure 6), and it takes more than more than 6 hours for sequential algorithm to process 2 million vertices.

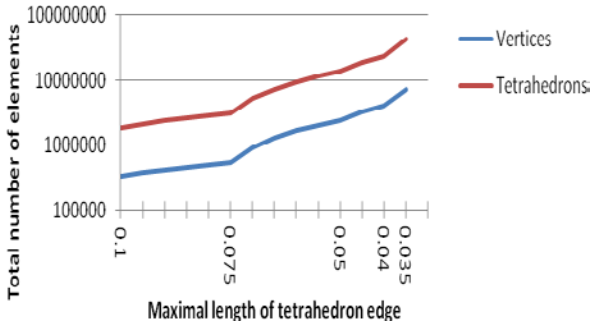

Fig. 6 *Ratio between the length of tetrahedron edge and the number of elements. Y axis has logarithmic scale*

We used CUDA for the parallel version. To measure the performance, a computer with the following characteristics was used: CPU Intel Core-i7 3820 3.6 GHz and GPU Nvidia GTX 680 4Gb 1006 MHz 1536 cores. The results are shown in the table below. Algorithms were tested on a mesh consisted of 2389529 vertices and 13690149 tetrahedrons.

Table 1: *Speed-up of the ray casting algorithm with CUDA*

Type of shape	Count of triangles	CPU msec	CUDA msec	Speed-up
AV	328	30995	100	310
SAV	590	54228	165	328
SA	792	82990	223	372
B	830	74979	221	339
F	2454	222461	606	367
RA	2646	247647	645	383
RV	3026	329501	771	427
His	4216	449535	1044	430
LA	5686	473853	1318	359
LV	7180	714429	1735	411
Total	27748	26800618	6828	392

We can see from the table that the speed-up when using the CUDA realization of the algorithm compared to the serial version is about 400x. Accelerating structures will increase the speed of the algorithm even more.

5. Conclusions

Having completed all the steps of the algorithm, we obtain a model of the heart, consisting of elements that belong to one of 10 types of tissue, to simulate the propagation of electrical heart signals along conductive paths of the cardiac system. With an optimized algorithm we can easily update our mesh after each tissue model correction, because the updating process takes some minutes rather than hours as is the case with sequential algorithms.

6. Acknowledgements

The Authors acknowledge Ministry of Education and Science of Russia (Contract № 02.G25.31.0157).

References

Arevalo HJ, B. P. (2016). Computational rabbit models to investigate the initiation, perpetuation, and termination of ventricular arrhythmia. *Progress in biophysics and molecular biology* , 185–194.

Aslanidi OV, N. T. (2013). Application of micro-computed tomography with iodine staining to cardiac imaging, segmentation, and computational model development. . *IEEE Trans Med Imaging.* , 32-8.

Colli Franzone P, G. L. (1998). Spread of excitation in 3-D models of the anisotropic cardiac. *Effects of fiber architecture and ventricular geometry* , 131-71.

Deng D, J. P. (2012). An image-based model of the whole human heart with detailed anatomical structure and fiber orientation. . *Comput Math Methods Med.* , 2012-16.

J. D. Foley, A. v. (1990). *Computer Graphics: Principles and Practice. The Systems Programming Series.* Addison-Wesley, Reading.

Lopez-Perez, A., & Sebastian, R. (2015). Three-dimensional cardiac computational modelling: methods, features and application. *BioMedical Engineering OnLine* .

P. Lamata et al. (2014). An automatic service for the personalization of ventricular cardiac meshes. *J. R. Soc. Interface* .

Plastic boy store. (n.d.). Retrieved august 2017, from Plastic boy anatomy models store: http://www.plasticboy.co.uk/store/

Plotkowiak M, R. B. (2008). High performance computer simulations of cardiac electrical function based on high resolution MRI datasets. *Int Conf Comput Sci 2008* , 571-80.

R. Bordasa, K. G. (2011). Rabbit-Specific Ventricular Model of Cardiac Electrophysiological Function including Specialized Conduction System. *Prog Biophys Mol Biol* .

Sermesant M, M. P.-M. (2006). Cardiac function estimation from MRI using a heart model and data assimilation: advances and difficulties. *Med Image Anal* , 642–56.