

High Performance Computer Simulations of Cardiac Electrical Function Based on High Resolution MRI Datasets

Michał Plotkowiak¹, Blanca Rodriguez², Gernot Plank³, Jürgen E. Schneider⁴, David Gavaghan^{1,2}, Peter Kohl⁵, and Vicente Grau⁶

¹ LSI Doctoral Training Centre, University of Oxford, UK

`michal.plotkowiak@bnc.ox.ac.uk`

² Computing Laboratory, University of Oxford, UK

³ University of Graz, Austria

⁴ Department of Cardiovascular Medicine, University of Oxford, UK

⁵ Department of Physiology, Anatomy and Genetics, University of Oxford, UK

⁶ Department of Engineering Science and Oxford e-Research Centre, University of Oxford, UK

`vicente.grau@oerc.ox.ac.uk`

Abstract. In this paper, we present a set of applications that allow performance of electrophysiological simulations on individualized models generated using high-resolution MRI data of rabbit hearts. For this purpose, we propose a pipeline consisting of: extraction of significant structures from the images, generation of meshes, and application of an electrophysiological solver. In order to make it as useful as possible, we impose several requirements on the development of the pipeline. It has to be fast, aiming towards real time in the future. As much as possible, it must use non-commercial, freely available software (mostly open source). In order to verify the methodology, a set of high resolution MRI images of a rabbit heart is investigated and tested; results are presented in this work.

1 Introduction

The heart is an electromechanical pump, whose function and efficiency are known to be intimately related to cardiac histoanatomy. A large number of anatomical and structural factors affect cardiac electromechanical activity, but their detailed influence is poorly understood. Computer simulations have demonstrated the ability to provide insight into the role of cardiac anatomy and structure in cardiac electromechanical function in health and disease [1],[2]. The most advanced cardiac models to date incorporate realistic geometry and fibre orientation [3]. However, for each animal species, only one example of cardiac anatomy is generally used, which obscures the effect of natural variability. In addition, cardiac tissue is generally represented as structurally homogeneous, and cardiac geometry is often overly simplified, as the endocardial structures for example are not represented in detail. This limits the utility of the computational models to

understand the role of heart structure and anatomy in cardiac electromechanical function. Recent advances in medical imaging techniques allow generation of high resolution images containing a wealth of information on the 3D cardiac anatomy and structure. Among all possible techniques, magnetic resonance imaging (MRI) is the most suitable for our purposes. MRI allows acquisition of images *in vivo* as well as *in vitro*, and can provide high-quality, high resolution datasets. Thus, MRI images of whole hearts can be used to obtain highly detailed, high resolution models of cardiac anatomy and structure. Figure 1 shows two anatomical MRI sections through two different rabbit hearts, obtained using an 11.7 T MRI system (500 MHz). The information that can be obtained from these high resolution MRI datasets can be used to build the next generation of cardiac computational models with realistic and accurate representation of cardiac anatomy and structure. The goal of the present study is to develop and identify a set of methodologies to run computer simulations of cardiac electrical activity. Proposed heart models incorporate detailed description of cardiac anatomy and structure based on high resolution MRI datasets.

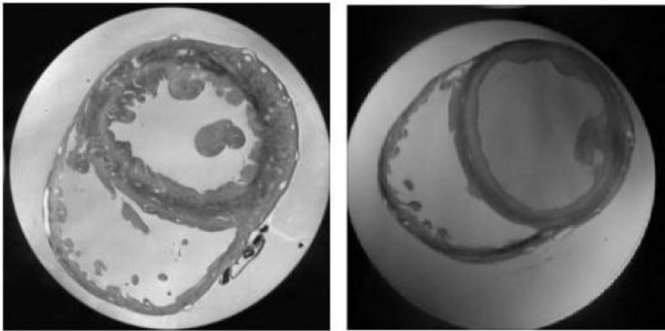


Fig. 1. MRI slices of two rabbit hearts acquired at different resolutions and different levels of contraction, the left image at $26.4 \times 26.4 \times 24.4 \mu\text{m}$, and the right image at $32 \times 32 \times 44 \mu\text{m}$

2 Methodology

This section describes the techniques to go from the high-resolution MRI images to computer simulations of cardiac electrophysiological function. This involves the use of software developed specifically for this project, based on open-source libraries (for segmentation and surface generation), as well as available software packages (for mesh generation and cardiac electrophysiology solver).

2.1 Data Acquisition

The MRI data were acquired using an 11.7 T (500 MHz) MRI system, consisting of a vertical magnet (bore size 123 mm; Magnex Scientific, Oxon, UK), a Bruker

Avance console (Bruker Medical, Ettlingen, Germany), and a shielded gradient system (548 mT/m, rise time 160 μ s; Magnex Scientific, Oxon, UK). For the MRI signal transmission/reception, dedicated quadrature driven birdcage RF-coils were used. Scanning was performed using a fast gradient echo technique for high-resolution gap-free 3D MRI. Images of coronary perfusion-fixed rabbit hearts, embedded in agarose, were acquired with an in-plane resolution of 26.4 μ m x 26.4 μ m, and an out-of-plane resolution of 24.4 μ m. All the methodological details were described previously in [5].

2.2 Segmentation

Segmentation of the MRI datasets is the first step towards extracting anatomical information for incorporation into computational models of cardiac electrophysiology. Segmentation may be described as the process of labelling each voxel in a medical image to indicate its tissue type or to which anatomical structure each voxel belongs to. As a first step, in this study, the aim was to segment high resolution MRI of rabbit hearts in order to generate an accurate description of the epicardial and endocardial structures. For this purpose an application, based on Insight Toolkit libraries (ITK, www.itk.org), was developed.

ITK is an open-source software system originally developed to support the Visible Human Project¹. In recent years it has become a standard tool for biomedical image segmentation and registration. Segmentation in the present study is performed using the fast marching method, one of the family of level set methods.

Level Set Method. Level set methods [6] are a family of numerical techniques for tracking the evolution of contours and surfaces. The main idea is to embed the evolving contour/surface in a higher dimension function ψ . The contour C is represented as the zero level set of this function. In the context of image segmentation, level sets are generally used to evolve a contour/surface using the evolution of the higher dimension function. A large number of variations have been presented in the literature, for which a comprehensive review is out of the scope of this paper. In the formulation introduced in [6], the level sets function is generally initialized with a signed distance map to an initial surface, and then evolves guided by a speed function combining internal (generally related to contour regularity) and external (generally linked to image features) influences. We use a simplified level set method called the fast marching method [7]. In this method it is assumed that the speed function $F > 0$, which means that the front is always going forward. This has the advantage of having to consider each voxel only once, thus making the algorithm significantly faster. We chose this method because of both reduced computational requirements (necessary when dealing with large 3D datasets as in this case) and for convenience reasons (available implementation in ITK). The position of the moving front can be characterized by calculating its arrival time $T(x, y)$ as it crosses each point (x, y) on the plane or space. Thresholding the arrival time provides a segmentation of the image.

¹ www.nlm.nih.gov/research/visible/visible_human.html

In the current application, we initialize the contour by using a set of automatically generated seeds. These are located using a set of heuristic rules involving the intensity of a block centered at each image voxel. A highly conservative threshold was used here to ensure that all the seeds belong to the object. The images were pre-processed, using an anisotropic filter, before application of the fast marching algorithm.

2.3 Surface Generation

In order to generate surface data for further volume meshing and visualization, special algorithms were applied, based on the Visualization Toolkit (VTK) package libraries. For a more detailed description refer to [8].

The proposed surface generation application contains three main elements: marching cubes, decimation and smoothing. Its role is not only to generate a spatial object from binary 2D data but also to prepare structures of interest for finite element meshing.

Marching Cubes. The marching cubes algorithm produces a triangular mesh by computing isosurfaces from volumetric data [9]. A cube is defined by the values at its eight vertices, corresponding to eight voxels in the original 3D image. When one or more vertices of a cube have values less than the specified isovalue, and one or more have values greater than this value, the voxel contributes some component to the isosurface. Determining which edges of the cube are intersected by the isosurface, a triangular patch can be created. The final surface representation is obtained by connecting the patches from all cubes.

Decimation. Marching cubes usually generate a large number of polygons, and so this has to be reduced before generating a finite element mesh. We used the decimation algorithm from [10], available in VTK. Decimation is designed to reduce the total number of triangles in a mesh, while preserving the original topology [8]. The proposed decimation is an iterative process, where each point in a triangle mesh is visited. Three basic steps are carried out for each of the points. In the first one, the local geometry and topology in the neighbourhood of the point are classified. Next, the vertex is assigned to one of possible five categories: simple, boundary, complex, edge, or corner point. Finally, using a decimation criterion that is based on a local error measure, it is determined whether the point can be removed. If the criterion is satisfied, the point and associated triangles are deleted and the resulting hole is re-triangulated.

Smoothing. Mesh smoothing is a method of shifting points of a mesh that can significantly improve its quality (i.e. appearance and shape), without modifying mesh topology. Smoothing applications improve isosurfaces by removing surface noise. We have used Laplacian smoothing, a general smoothing technique that has been used successfully in other applications [8].

2.4 Finite Element Mesh

The majority of the leading cardiac electrophysiology simulators (such as the one used in this study and described in the next section) utilize tetrahedral finite element meshes as an input. Therefore, from the model cardiac surfaces, generated as described in the previous section, a tetrahedral finite element mesh was generated. First, a surface mesh composed of triangles or rectangles was generated, and then the volume mesh composed of tetrahedral elements was fitted into the cardiac volume. The most common unstructured meshing algorithms are Delauney triangulation and advancing front methods. There are many different automatic mesh generating tools available. However, meshes generated in this way may contain poorly shaped or distorted elements that cause numerical problems. For instance the size of dihedral angles is very important; if too small then the corresponding number of the elemental matrices increases, if too large then the discretization error in the finite solution increases.

The meshing program used in this project, Tetgen², performs Delauney tetrahedralization using algorithms presented in [11]. It also includes algorithms for quality control, e.g. Shewchuk's Delaunay refinement algorithm. This algorithm ensures that no tetrahedra in the generated mesh have a radius-edge ratio greater than 2.0. The reason for choosing TetGen is that it is freely available, contains state-of-the-art tetrahedralization algorithms, and offers good mesh quality control.

2.5 Electrophysiological Simulations

The finite element meshes were generated specifically to conduct simulations of ventricular electrophysiological activity, using one of the most advanced bidomain cardiac simulators available to date, namely Cardiac Arrhythmias Research Package (CARP)³. CARP uses computational techniques to solve the cardiac bidomain equations [12] defined as:

$$\nabla(\sigma_i \nabla V_m) + \nabla[(\sigma_i + \sigma_e)\phi_e] = -I_{stim} \quad (1)$$

$$\nabla(\sigma_e \nabla \phi_e) = -I_{ion} - I_{stim} \quad (2)$$

$$V_m = \phi_i - \phi_e; \quad (3)$$

where ϕ_i and ϕ_e are intra- and extracellular potentials, σ_i and σ_e are intra- and extracellular conductivity tensors, and I_{ion} and I_{stim} are the volume densities of the transmembrane and stimulated currents.

In the bidomain model, membrane kinetics are represented by a system of ordinary differential equations that are used to compute the total transmembrane ionic current I_{ion} . CARP uses an expanded library of ion models and plugins (augmentations) called LIMPET. As the animal species in this study is rabbit, the Puglisi-Bers rabbit Ionic model described in [13] was used, which consists of

² <http://tetgen.berlios.de/>

³ <http://carp.meduni-graz.at/>

a system of 17 ordinary differential equations (ODEs) that describe the electrophysiological behaviour of ion channel currents, pumps and exchangers in rabbit ventricular cells. Bidomain models are often used for studies of defibrillation, simulating the application of strong shocks to the heart, which requires representation of the extracellular space. However, simulations of cardiac propagation often use monodomain models, which can be obtained from the bidomain model by assuming $\phi_e = 0$. The detailed representation of cardiac structure incorporated in the meshes results in a large number of mesh nodes and therefore in computationally very demanding simulations, despite CARPs efficiency. Thus, simulations required the use of high performance computing such as offered by the UK National Grid Service (NGS) (www.grid-support.ac.uk).

3 Results and Discussion

Here we present results for each step of the model development: segmentation, surface generation, finite element mesh, and electrophysiological simulations using a high resolution MRI dataset of the rabbit heart.

3.1 Segmentation Results

MRI data were reconstructed and stored in the form of 1440 2D TIFF-images with a resolution of 1024x1024 pixels. For our purpose, the TIFF-images were down-sampled by a factor of 4 on each axis in order to speed up the segmentation process. The same segmentation method can be applied to the full resolution data, however, it would require the use of high performance computing capabilities such as the NGS, for which our segmentation software is not ready at this stage. As a first step in the model development process, we focused on segmentation of the endocardial and epicardial surfaces. Even though, in theory, the fast marching process could work from a single seed, in practice a reduced number of seeds was prone to produce leakage in areas where gradient values were small. In our segmentation program, we generate about 4000 seed points for each MRI slice. The results of segmentation of the rabbit heart are shown in Fig.2.

3.2 Surface Generation Results

The marching cube algorithm was used to generate 3D isosurfaces from the segmented 2D slices. Due to the large size of data (about 200 MB), a decimation algorithm was applied. Parallel algorithms are in the process of being developed to allow handling of the full resolution dataset. Decimation allowed reduction of the datasize by about 50% while still maintaining a very detailed structure. Finally, to improve the quality of decimated isosurfaces smoothing was applied. The final isosurfaces are presented in Fig.3.

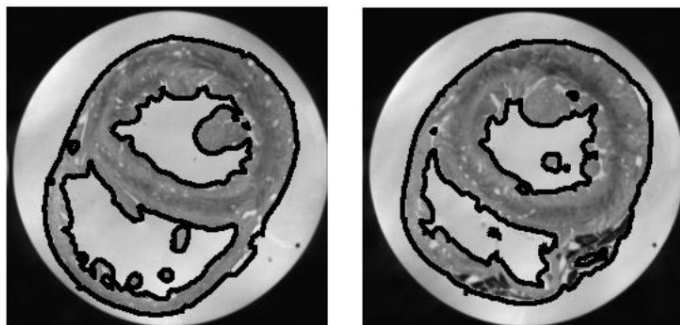


Fig. 2. Segmentation of MRI slices of rabbit heart. The black outline on the MRI slices shows the segmented boundaries of ventricular structure.

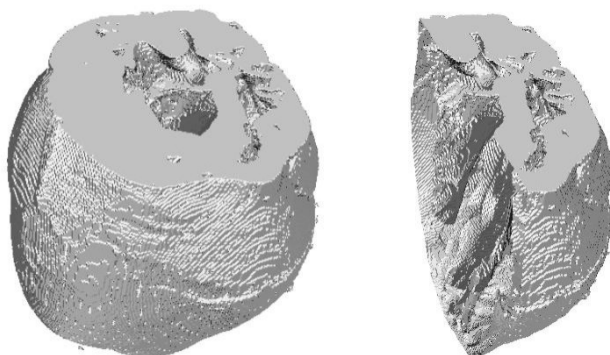


Fig. 3. Rabbit heart isosurfaces generated using marching cubes algorithm. Surfaces were decimated and smoothed using VTK functions.

3.3 Finite Element Mesh

As a result of the steps described above, we obtain an isosurface in *stl* format, fully compatible with the mesh generator TetGen that we use here. In order to ensure a good mesh quality, i.e. a mesh where the radius-edge ratio is smaller than 2.0 for all tetrahedrals and maximum element volume is constrained, a set of TetGen switches was used⁴. The final mesh consisting of 828,476 nodes and 3,706,400 tetrahedral elements, is shown in Fig.4.

3.4 Electrophysiological Simulations Results

The output tetrahedral mesh from TetGen had to be converted into a format compatible with the CARP solver. All simulations for the generated models were

⁴ <http://tetgen.berlios.de/>

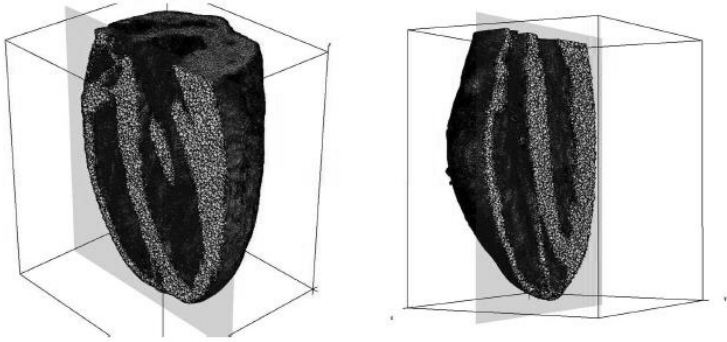


Fig. 4. Different cuts through the finite element mesh for rabbit heart containing about 4 million tetrahedral elements



Fig. 5. Different stages of electrical propagation in the developed model. Transmembrane potential values at each epicardial node are shown using a grey scale, where black is resting potential and white is depolarized.

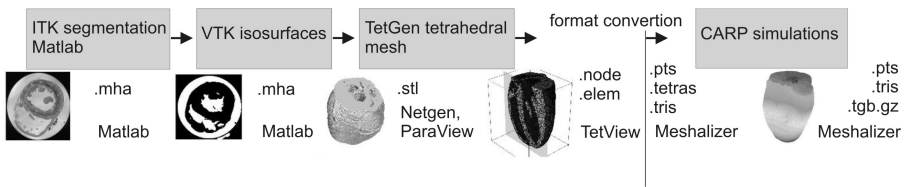


Fig. 6. Proposed model development pipeline illustrating main applications, methods, file formats, and visualization programs used

carried out using a monodomain model. There is no special preparation needed for the bidomain calculation, however for the sake of obtaining a simple electrical propagation the monodomain mode is sufficient and computationally more efficient. Figure 5 shows transmembrane potential distribution on the epicardium at several time points during electrical propagation from apex to base following the application of the electrical stimulus.

4 Conclusions

The main contribution of this paper is the design and implementation of an application pipeline, that uses high resolution MR images to create individualized anatomically detailed heart models that are compatible with advanced cardiac electrophysiology simulators (such as CARP). Techniques such as the ones presented here, by generating models with unprecedented realism and level of detail, and introducing natural variability between individuals, have the potential to strengthen and broaden electrophysiological models as a fundamental tool in cardiac research. The main techniques applied in the presented model development are: segmentation, which uses fast marching algorithm to extract ventricular geometry; surface generation that uses marching cubes algorithm and some surface processing (decimation and smoothing), and finite element mesh generation using Delauney tetrahedralization. The main parts of the developed pipeline are presented in Fig.6. The main objectives of this work were to demonstrate the feasibility of the process, and to propose a working prototype. Most of the methods used are amenable to improvement; for instance, more sophisticated segmentation algorithms will be needed in order to extract different anatomical structures, such as papillary muscles and blood vessels, and adaptive meshing techniques may be applied for creating more efficient finite element models. The electrophysiological model used here has some limitations, and in order to be useful in particular applications additional information, such as fibre orientation or cell type distribution, will have to be included. This can be obtained using additional segmentation techniques, or by including information from different image modalities such as histology. In addition, cardiac electro-mechanically coupled solvers are currently being developed that will allow simulation of cardiac electromechanical activity using the meshes developed through the proposed pipeline.

Acknowledgements

This work was supported by a LSI DTC scholarship (to M.P.), an MRC Career Development Award (to B.R.), a Marie Curie Fellowship (to G.P.), and a BBSRC grant (BB E003443 to P.K.).

References

1. Kerckhoffs, R.C.P., et al.: Computational methods for cardiac electromechanics. *Proc. IEEE* 1994, 769–783 (2006)
2. Rodriguez, B., et al.: Differences between left and right ventricular geometry affect cardiac vulnerability to electric shocks. *Circ. Res.* 97, 168–175 (2005)
3. Vetter, F.J., McCulloch, A.D.: Three-dimensional analysis of regional cardiac function: A model of rabbit ventricular anatomy. *Prog. Biophys. Mol. Biol.* 69, 157–183 (1998)
4. Nielsen, P.M.F., et al.: Mathematical-model of geometry and fibrous structure of the heart. *Amer. J. Physiol.* 260, H1365–H1378 (1991)

5. Burton, R., et al.: Three-Dimensional Models of Individual Cardiac Histoanatomy: Tools and Challenges. *Ann. NY Acad. Sci.* 1080, 301–319 (2006)
6. Osher, S., Sethian, J.: Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics* 79, 12–49 (1988)
7. Sethian, J.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge (2002)
8. Schroeder, W., et al.: *The Visualization Toolkit*, 3rd edn. Kitware Inc (2004)
9. Lorensen, W., Cline, H.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics* 21(3), 163–169 (1987)
10. Schroeder, W., et al.: Decimation of Triangle Meshes. *Computer Graphics* 26(2), 65–70 (1992)
11. Si, H., Gaertner, K.: Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. In: *Proceeding of the 14th International Meshing Roundtable* (2005)
12. Vigmond, E.J., et al.: Solvers for the cardiac bidomain equations. *Prog. Biophys. Mol. Biol.* 33, 10 (2007)
13. Puglisi, J., Bers, D.: LabHEART: an interactive computer model of rabbit ventricular myocyte ion channels and Ca transport. *Am. J Physiol. Cell. Physiol.* 281(6), 2049–2060 (2001)