An Interactive Virtual Reality Environment for Analysis of Clinical Atrial Arrhythmias and Ablation Planning

Axel Loewe[1], Emanuel Poremba[1,2], Tobias G. Oesterlein[1],
Nicolas Pilia[1], Micha Pfeiffer[2], Olaf Dössel[1], Stefanie Speidel[2,3]

[1]Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[2]Institute for Anthropomatics and Robotics, KIT, Karlsruhe, Germany
[3]Translational Surgical Oncology, National Center for Tumor Diseases (NCT), Dresden, Germany

Abstract

Atrial arrhythmias like atrial fibrillation and atrial flutter are a major health challenge in developed countries. Radiofrequency ablation performed via intracardiac catheters is a curative therapy for these reentrant arrhythmias. However, the optimal location of ablation lesions is not straightforward to determine, particularly for complex activation patterns. Thus, a clinical need for tools to intuitively visualize complex activation patterns and to provide a platform to evaluate different ablation strategies in dry runs is apparent.

Here, we present a virtual reality system that allows to interactively simulate atrial excitation propagation and place ablation lesions. Our software builds on the IMHOTEP framework for the Unity3D engine and implements a multithreaded model-view-controller design pattern. Excitation propagation is computed using a fast marching approach considering refractoriness. Interactive rewind and playback is supported through a combination of the flyweight pattern for simulation data with complete snapshots for key frames. The system was evaluated in a user study using the HTC Vive™ headset including two controllers. For high fidelity virtual reality interaction, a minimum frame rate of 60 per second is required. In a biatrial anatomical model comprising 36,059 nodes (Figure 1), even complex activation patterns with multiple wavefronts could be simulated and rendered down to 2x slow motion (1 sec activation sequence displayed during 2 sec wall time) on a desktop machine. Results of the user study suggest added value regarding the comprehension of arrhythmias and ablation options and very good intuitiveness of the user interface requiring almost no teach-in. The virtual reality tool is ready to be used for educational purposes and prepared to import personalized models supporting diagnosis and therapy planning for atrial arrhythmias in the future.

1. Introduction

Atrial flutter and atrial fibrillation are the most common sustained arrhythmias in humans and one of the major health challenges in developed countries. Despite the high prevalence and severe implications of these arrhythmias, medicine is lacking reliable treatment options for a large share of patients [1,2]. While both the clinical systems employed to map atrial arrhythmias in patients and our understanding of basic patho-mechanisms governing such arrhythmias have made substantial progress in the last years, the success rates of catheter ablation to cure them remain suboptimal. Part of the reason is that today's electroanatomical mapping systems acquire a massive amount of data, which is not adequately reflected by available visualization methods. Moreover, no method exists to evaluate and test different ablation approaches in dry runs in an environment considering the individual characteristics of a patient. This lack often leaves the physician with the 'learning by burning' paradigm.

In this paper, we present a method (*FaMaS-VR*) that allows to interact with an individual patient model in real time in a virtual reality setting. Besides facilitating the comprehension of the individual reentry mechanism at work through the intuitive 3D user interface, it allows to interactively explore and test different ablation strategies in a model reflecting the characteristics of the individual patient.

2. Methods

In this section, the methods for excitation propagation simulation, for interactive playback and rewind and the software architecture are detailed.

A triangular surface mesh is used to represent the anatomy of the individual patient's atria. To calculate excitation propagation, we use an implementation of the

ISSN: 2325-887X DOI:10.22489/CinC.2017.125-118

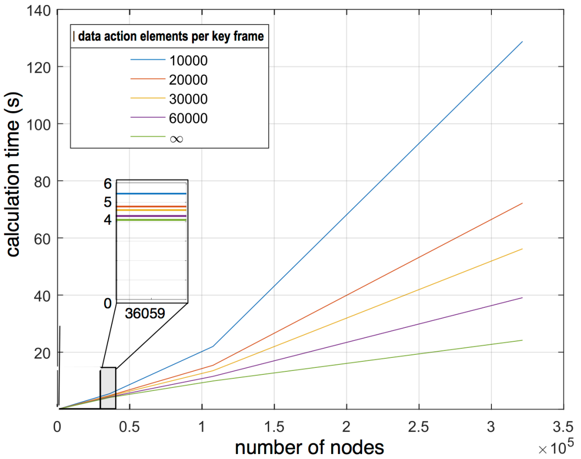

Figure 1. The calculation time for one regular sinus node activation increased linearly with the number of nodes in the model without key framing. The smaller the number of data action elements per key frame, the less linear the relation between calculation time and number of nodes.

Table 1. Data model for each node. 104 bytes are required per node to store all relevant information for a heterogeneous, anisotropic re-entry simulation.

Field	Size
status	4 bytes
timestamp	4 bytes
last timestamp	4 bytes
ERP	4 bytes
CV	4 bytes
ERP duration	4 bytes
fiber tensor	64 bytes
position	12 bytes
index	4 bytes
Sum	104 bytes

fast marching algorithm that accounts for anisotropic and heterogeneous tissue properties and considers refractoriness [3,4]. As shown in Table 1, 104 bytes define the state of a node. For an atrial model comprising 30,000 nodes, this sums up to 3.05 MB. The complexity of the fast marching algorithm is $\mathcal{O}(NlogN)$ as can be seen in Figure 1. For memory-limited systems, the data per node can be reduced to 20 bytes at the expense of up to 10x longer calculation times.

To provide interactive rewind and fast forward functionality, two fundamental approaches can be employed. Either the complete state of the system is saved for each frame or the complete history of the system is saved using the flyweight pattern. The former approach requires around 1.5 GB of data per second simulation time and therefore does not scale well. The latter approach makes use of data action elements representing atomic changes in the state of the system (e.g. one node recovering from refractoriness). Reversing the data action elements allows to move back in time. One regular sinus node activation produces around 190,000 data action elements with a total memory footprint of less than 30 MB for a model comprising 30,000 nodes. While this method is very memory-efficient, large jumps in time require to process a large number of data action elements, which causes a time lag. For these reasons, we adopted a hybrid scheme saving a complete snapshot of the state of the system (‘key frame’) after a certain number of data action elements have accumulated (cf. Figure 1). In this way, a jump to any point in time can be realized by finding the closest key frame and then applying only a few action elements to reconstruct the correct state. This approach minimizes both the memory footprint and the time lag for fast forward or rewind operations.

FaMaS-VR builds on the *IMHOTEP* framework providing interfaces for patient data management, visualization, and user interaction via cutting-edge devices (Figure 2). We used Vive™ (HTC, Taoyuan, Taiwan) virtual reality goggles paired with two haptic feedback controllers providing an immersive environment. Patient data can be provided in DICOM and Blender formats and supplemented with additional metadata in the JSON format. For visualization, *IMHOTEP* uses the *Unity3D* engine. The functionality of *Unity3D* can be extended by linking to pre-compiled libraries or providing code in C# or JavaScript being compiled just in time. C# implementations need to be based on the Mono framework, which implements the .NET standard, comes with a garbage collector, and facilitates platform independence.

FaMaS-VR is implemented using two threads as shown in Figure 3. The visualization thread is implemented according to the model-view-controller pattern. The visualization data is updated according to a time-

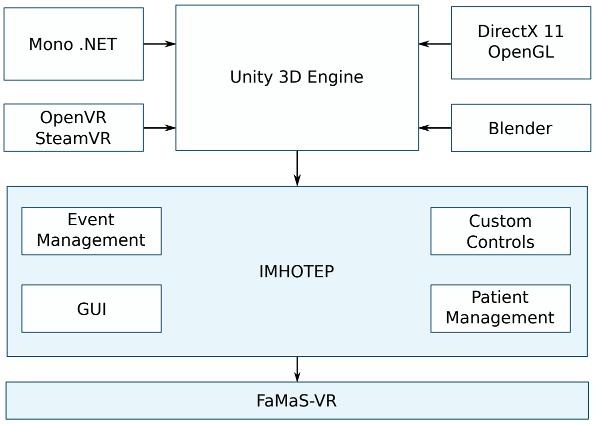

Figure 2. Software architecture. The *IMHOTEP* framework provides the GUI and communication interfaces for *FaMaS-VR*. The *IMHOTEP* framework itself is based on the *Unity3D* engine.

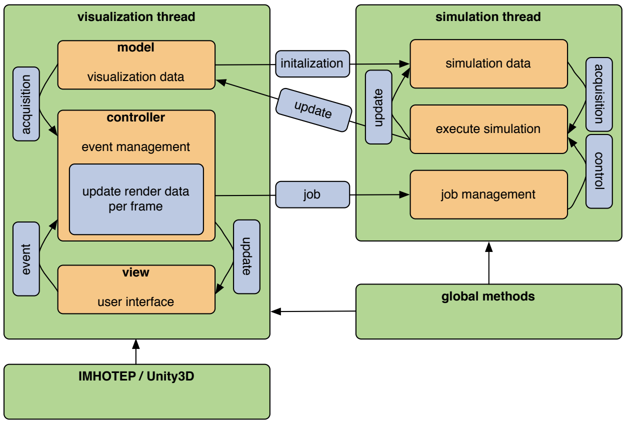

Figure 3. The implementation uses separate threads for visualization and simulation. The visualization thread implements the model-view-controller pattern and controls the simulation thread via job objects.

controlled scheme based on the *Unity3D* frame-by-frame rendering. The controller of the visualization thread and the simulation thread are controlled via discrete events.

3. Results

Our implementation of the fast marching algorithm in the virtual reality environment was evaluated for biatrial anatomical models of different resolutions. For high fidelity virtual reality interaction, a minimum frame rate of 60 per second is required. For a model comprising 36,059 nodes, even complex activation patterns with multiple wave fronts could be simulated and rendered down to 2x slow motion (1 sec activation sequence displayed during 2 sec wall time) on a desktop machine (AMD FX-8350 8x4 GHz, 24 GB RAM, GeForce GTX 970). For practical applications, far higher slow motion factors would be sufficient as the activation is far too rapid to comprehend the pattern in real time. The C# garbage collector caused rendering lags of up to 500 ms when cycling through the large number of orphaned data action elements [5]. To reduce this problem, we recycle data action element objects in order to reduce the workload of the garbage collector. This pooling approach did dramatically reduce both the frequency and the duration of garbage collector-induced lags.

Figure 4 shows a screenshot of the *FaMaS-VR* user interface. The toolbar allows to place ablation lines, mark slow conducting tissue to represent e.g. fibrotic regions, and to place triggers. The design and implementation of the interface were evaluated in a user survey comprising 21 students and researchers in the fields of biomedical engineering and clinical electrophysiology with widely varying experience in virtual reality applications. While the users affirmed ease of use and intuitiveness in general, the time control leaves room for optimization (Figure 5). Particularly the placement of zones of slow conduction and ablation lesions was rated "rather intuitive" or "very intuitive" by >95% of the participants. The users completed the tasks of marking slow conducting tissue and placing ablation lesions in an average time of 84 s (range: 35-230 s) with an average error of 0.8% (range: 0.19-1.4%) without clear correlations between level of virtual reality experience and performance indicating an intuitive user interface.

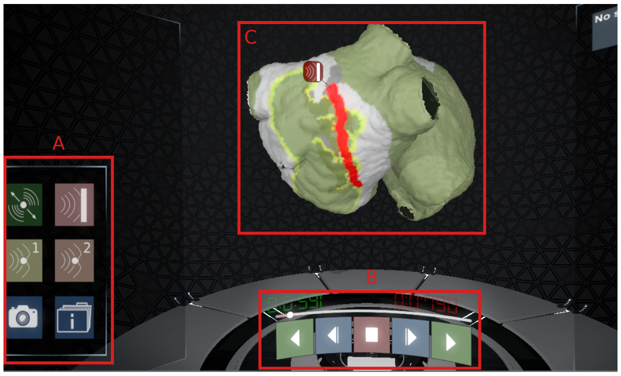

Figure 4. *FaMaS-VR* user interface. The toolbar (A) is linked to the controller in the left hand and provides tools to set stimuli, define tissue properties, place ablations, as well as to save and load simulation snapshots. The time control (B) allows to stop, rewind, and fast forward the simulation. The simulation model (C) shows the activation wave dynamics, moves according to the movement of the operator's head and can be rotated and manipulated with the controller in the right hand.

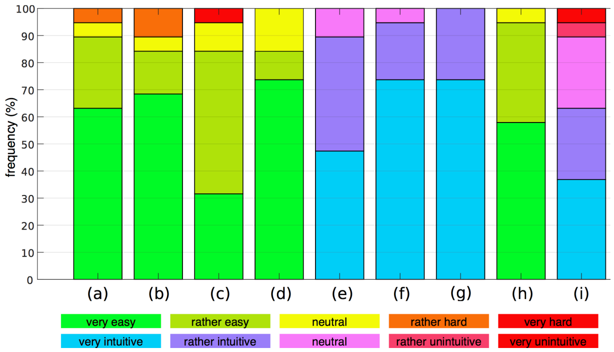

Figure 5. User study results for ease of use and intuitivity for the aspects (a) positioning and zooming, (b) marking slow conducting tissue, (c) placing ablations, (d) toolbar menu, (e) tools, (f) marking slow conducting tissue, (g) placing ablations, (h) time control, (i) time control.

4. Discussion

In this paper, we presented a virtual reality system that allows to interactively simulate atrial excitation propagation and place ablation lesions. The Eikonal model describing excitation propagation and solved using the fast marching scheme in our proposed setup has been shown to realistically reproduce regular activation and stable forms of reentry [6,7]. Thus, it is well suited for the simulation of stable, repetitive arrhythmias like atrial flutter. For more complex and chaotic patterns like atrial fibrillation, effects as e.g. source sink mismatch would need to be considered.

Regarding the computational load, the C# Mono implementation could provide high fidelity simulation output down to 2x slow motion in a model comprising 36,059 nodes. While this is fast enough for practical application, there is room for improvement and higher resolution models by outsourcing the fast marching algorithm in a C++ library. Compared to the C# Mono implementation, the C++ code performed 136x faster. The main reason for this difference is the implementation of sorted lists in C# Mono vs. C++ STL. Outsourcing parts of the algorithm in a C++ library would also reduce the workload for the garbage collector, however requiring strict separation of managed and unmanaged code.

For future translational perspectives regarding patient-specific evaluation of ablation patterns, the workflow for model import in the clinical setting needs to be streamlined. Anatomical models could be augmented with rule-based a priori information like myocyte orientation [8], or physiological information like disease-state [9], genotype [10], or medication [11]. Advances in mapping systems and appropriate signal processing methods will facilitate patient-individual and spatially resolved parameterization of tissue properties like conduction velocity and duration of the effective refractory period [12]. To pave the way for clinical application, experience and efforts in modelling and biosignal analysis need to be joined [13].

In the current state, the system can be used for training and educational purposes. When equipped with a set of relevant and challenging arrhythmia cases, *FaMaS-VR* can provide an ideal environment for physicians in training to acquire experience and skills in diagnosing, comprehending, and eventually curing arrhythmia by placing appropriate ablation lesions.

By facilitating comprehension of complex arrhythmia activation patterns via intuitive visualization and by evaluating ablation options in interactive dry runs, virtual reality environments like *FaMaS-VR* can support diagnosis and therapy planning in the future.

Acknowledgments

We gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) through CRC 1173.

References

[1] Kirchhof P, Benussi S, Kotecha D, et al. 2016 ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS. Eur Heart J. 2016;37:2893–962.

[2] Verma A, Jiang C-Y, Betts TR, et al. Approaches to catheter ablation for persistent atrial fibrillation. N Engl J Med. 2015;372:1812–22.

[3] Loewe A. Modeling human atrial patho-electrophysiology from ion channels to ECG : Karlsruhe: KIT Scientific Publishing; 2016.

[4] Sermesant M, Konukoglu E, Delingette H, et al. An Anisotropic Multi-front Fast Marching Method for Real-Time Simulation of Cardiac Electrophysiology. In: Functional Imaging and Modeling of the Heart. 2007;4466:160–9. (Lecture Notes in Computer Science; vol. 4466).

[5] Boehm H-J. Space efficient conservative garbage collection. In: Proceedings of the ACM SIGPLAN 1993. 1993;28:197–206.

[6] Jacquemet V. An eikonal-diffusion solver and its application to the interpolation and the simulation of reentrant cardiac activations. Comput Methods Programs Biomed. 2012;108:548–58.

[7] Wallman M, Smith NP, Rodriguez B. A comparative study of graph-based, eikonal, and monodomain simulations for the estimation of cardiac activation times. IEEE Trans Biomed Eng. 2012;59:1739–48.

[8] Wachter A, Loewe A, Krueger MW, et al. Mesh structure-independent modeling of patient-specific atrial fiber orientation. Curr Dir Biomed Eng. 2015;1:409–12.

[9] Loewe A, Wilhelms M, Dössel O, et al. Influence of chronic atrial fibrillation induced remodeling in a computational electrophysiological model. Biomed Tech / Biomed Eng. 2014;59:S929–32.

[10] Loewe A, Wilhelms M, Fischer F, et al. Arrhythmic potency of human ether-à-go-go-related gene mutations L532P and N588K. Europace. 2014;16:435–43.

[11] Loewe A, Lutz Y, Wilhelms M, et al. In-silico assessment of the dynamic effects of amiodarone and dronedarone on human atrial patho-electrophysiology. Europace. 2014;16:iv30-iv38.

[12] Trächtler J, Oesterlein TG, Loewe A, et al. Virtualizing clinical cases of atrial flutter in a fast marching simulation including conduction velocity and ablation scars. Curr Dir Biomed Eng. 2015;1:405–8.

[13] Dössel O, Lenis G, Loewe A, et al. Atrial Signals – Modeling Meets Biosignal Analysis. In: IFMBE Proceedings. 2017 Jun 11;65:723–6.

Address for correspondence:
Axel Loewe, PhD; Institute of Biomedical Engineering,
Karlsruhe Institute of Technology (KIT)
Fritz-Haber-Weg 1, 76131 Karlsruhe, Germany
publications@ibt.kit.edu