



Deliverable D10.4.1

Hugo Talbot, Federico Spadoni, Maxime Sermesant, Nicholas Ayache, Hervé Delingette

► To cite this version:

Hugo Talbot, Federico Spadoni, Maxime Sermesant, Nicholas Ayache, Hervé Delingette. Deliverable D10.4.1. [Research Report] 2011, pp.25. hal-00918216

HAL Id: hal-00918216

<https://hal.inria.fr/hal-00918216>

Submitted on 13 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



euHeart

Personalised & Integrated Cardiac Care: Patient-specific
Cardiovascular Modelling and Simulation for *in silico*
Disease Understanding & Management and for Medical
Device Evaluation & Optimization

FP7-ICT-2007

IP Contract no: 224495

Deliverable D10.4.1

***Progress report on the development of a prototype simulator for
radiofrequency ablation***

Due date of deliverable: 31.03.2012

Actual submission date: 31.03.2011

Start date of project: 01 June 2008

Duration: 48 months

Responsible WP Leader: INRIA

Revision: <Final Version>

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

Project Number	224495
Project Title	euHeart
Project Duration	48 months: 01 June 2008 - 31 May 2012

Deliverable Number	D10.4.1
Contractual Delivery date	31.03.2012
Actual delivery date	31.03.2011
Title of Deliverable	Simulator for RF Ablation
Contributing Work package	10
Dissemination level	Restricted to a group specified by the consortium
Author (s)	Hugo Talbot, Federico Spadoni (INRIA), Maxime Sermesant (INRIA), Nicholas Ayache (INRIA), Hervé Delingette (INRIA),

Abstract:

This deliverable describes the work developed in the course of the Workpackage 6 of the euHeart project,

Contents

1	Introduction	4
2	Simulator Objective and Setup	4
2.1	Objectives of Endovascular Simulation of Radiofrequency Ablation	4
2.2	Simulator components	5
2.3	The Hardware Setup	5
2.4	The Software Components	7
2.5	Main Challenges	7
3	EndoVascular Simulation	9
3.1	Reconstruction of the beating heart and vessels	9
3.2	Main Principles of the Simulation of Endovascular Navigation	10
3.2.1	Physics simulation of catheter	11
3.2.2	Collision Detection	12
3.2.3	Simulation of X-ray	14
3.3	Results on the Simulation of Endovascular Navigation in the Beating Heart	16
4	ElectroPhysiology Simulation	18
4.1	Biophysical Models of Cardiac Electrophysiology suitable for fast computation	19
4.2	Optimized Mitchell-Schaeffer Model	19
4.3	Results of Optimized Mitchell-Schaeffer Model	20
5	Discussion and Conclusion	22
5.1	Conclusions	22
5.2	Limitations	22
5.3	Perspectives	22
6	Acknowledgements	22
	Appendix	23
	Notation	23
	Bibliography	25

1 Introduction

This deliverable (D6.3.2) is part of euHeart Workpackage 10 (WP 10)

2 Simulator Objective and Setup

2.1 Objectives of Endovascular Simulation of Radiofrequency Ablation

The objective of task 10.4 is to create a prototype to simulate the ablation of cardiac tissue through catheterization. The practical interest of such simulator are :

- to train cardiologists / electrophysiologists to perform cardiac radiofrequency ablation and more precisely ablation on VT cases.
- to rehearse a complex procedure based on patient specific datasets.
- to help the development and design of new catheterization cardiac procedures that can be tested *in silico* on personalized cases.

There are a limited number of endovascular simulators including:

- Angio mentor[[Sim](#)] from Simbionix
- Cardiac Rythm Management simulator[[Men](#)] from Mentice (Gothenburg, Sweden).
- CathLab Simulator[[CC](#)] from CAE Healthcare (Montreal, Canada).

Those commercial products are limited because they rely on pre-stored electrophysiology data not on electrophysiology modeling. More precisely, they typically restrict significantly the possible positions and behaviors of pacing and ablating catheters in order to display a limited number of biosignals that have been previously stored. In this workpackage, we aim at simulating physically all biosignals such that a complete freedom is let to the user.

Endovascular simulators have also been developed at various academic centers:

- INRIA and CIMIT [[LCDN06](#)] [[CDL+05](#)],
- Nanyang Technological University [[CCM+11](#)], and National University [[LRRV00](#)] in Singapour,
- SORA Imperial College London Biosurgery [[LHG+09](#)].

We have chosen to develop this prototype in close link with the Workpackage 6 which is dedicated to the planning of cardiac radiofrequency ablation. More specifically, the targeted procedure of this prototype is the simulation of the radiofrequency ablation of tissue of patients suffering from ischemic Ventricular Tachycardia (VT). This electrophysiological pathology is characterized by the

presence of scars (necrotic tissue caused by an infarct) in the myocardium surrounded by a peri-infarct zone where necrotic, fibrosed and healthy tissue can be found. The heterogeneity of tissue modifies the conduction velocity property of the depolarization and repolarization waves as well as their distance (action potential duration) as a function of the wave speed (restitution properties). Those changes are such are under certain circumstances, spiraling waves may be created around the scar regions thus leading to tachycardia and even in some cases fibrillation.

The objective of the simulator prototype is to simulate the ablation of myocardial tissues in the left ventricle in order to suppress the possibility to obtain re-entry patterns.

A typical scenario of simulation would be the following :

1. Navigation of a guidewire from the femoral vein / artery into the right / left atrium. The guidewire is handled by a user through a hardware tracking device (see section 2.3). The navigation is controlled by a simulated X-ray view of the thorax where the catheter and the heart are visible.
2. Insertion of a guidewire in the apex of right ventricle going through the tricuspid valve. The catheter inserted in this guidewire would be used for pacing the heart with various stimuli.
3. Insertion of a guidewire in the left ventricle next to the scar region. The catheter inserted in this guidewire would be used for performing radiofrequency ablation and measuring extra-cellular potential in the endocardial surface of the left ventricle.
4. Simulation of pacing (overdrive, S1-S2) from the catheter in the right ventricle. The extra-cellular potential measurements in the left ventricle is simulated and displayed on the screen.
5. Simulation of radiofrequency ablation from a catheter located in the left ventricle. The extent of ablated tissue depends on some power setting and the duration of ablation (which can be controlled by a foot-pedal).
6. The simulation of pacing and ablation are iterated until the re-entry patterns can no longer be observed.

2.2 Simulator components

2.3 The Hardware Setup

The simulation hardware is mainly composed of :

- a screen where X-ray images and biosignals are displayed.
- a computer with mouse and keyboard. It consists of PC computer with 8 cores running Microsoft Windows and with a latest graphics card (with Graphical Processing Unit).
- a catheter tracking device inside which real catheters are inserted and manipulated by a user.

The hardware used to track guidewires and catheters has been acquired from the Mentice corporation as the VIST-C system. This system can be connected to a computer (see Fig 1). A C++ driver library allows to read the position of guidewires and catheters and to set the position of a brake in order simulate some force feedback.

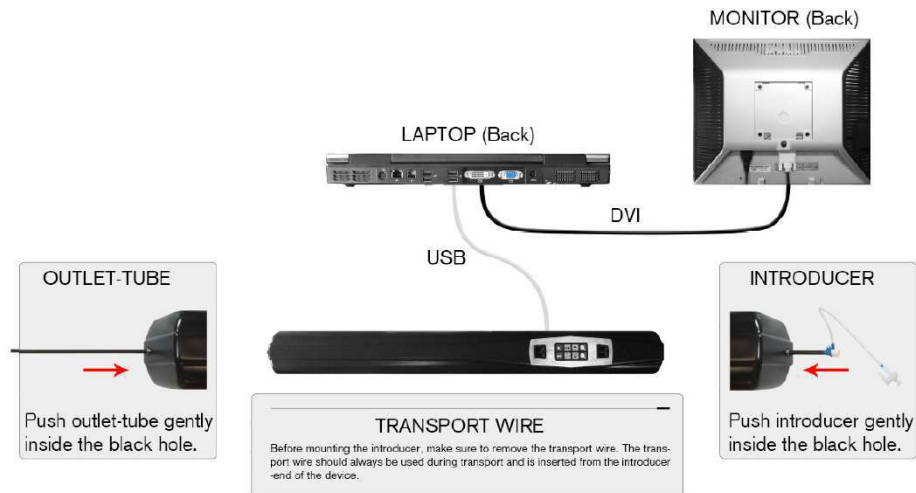


Figure 1: (Left) Hardware used to track up to three catheters and guidewire; (Right) Example of a user using the simulation setup.

This hardware setup is therefore quite sophisticated in order to simulate realistically the navigation of catheters inside the heart. Example of a user manipulating the simulator is shown in Fig. 3.



Figure 2: (Left) Hardware used to track up to three catheters and guidewires; (Right) Simulation hardware setup.

2.4 The Software Components

The software components that are needed to develop the simulator include :

- **Hardware driver.** This component allows to read the positions of catheters and guidewires from the Vist-C system and to set the brake status.
- **Catheter Navigation.** It takes as input the motion of real catheters and guidewires in the tracking device and simulate the motion of virtual catheters inside the vessels and the heart. In particular, it computes the contact between catheters and vessels walls and models the mechanical behavior of catheters. This component is detailed in section 3.2.
- **X-ray Visualization.** It simulates the visualization of the thorax and catheters on a X-ray imaging system. This is detailed in section 3.2.3.
- **Cardiac Motion Simulator.** It simulates the cardiac motion of both ventricles and atria. This component is detailed in section 3.3.
- **Cardiac Electrophysiology Simulator.** It simulates the propagation of transmembrane and extracellular potential inside the myocardium. It is detailed in section 4.
- **Pacing and Radiofrequency Ablation Simulation.** It models the injection of currents from pacing leads located inside the right ventricle and the action of ablation catheters.
- **BioSignals Visualization.** it visualizes the simulated electrophysiology signals on the screen.

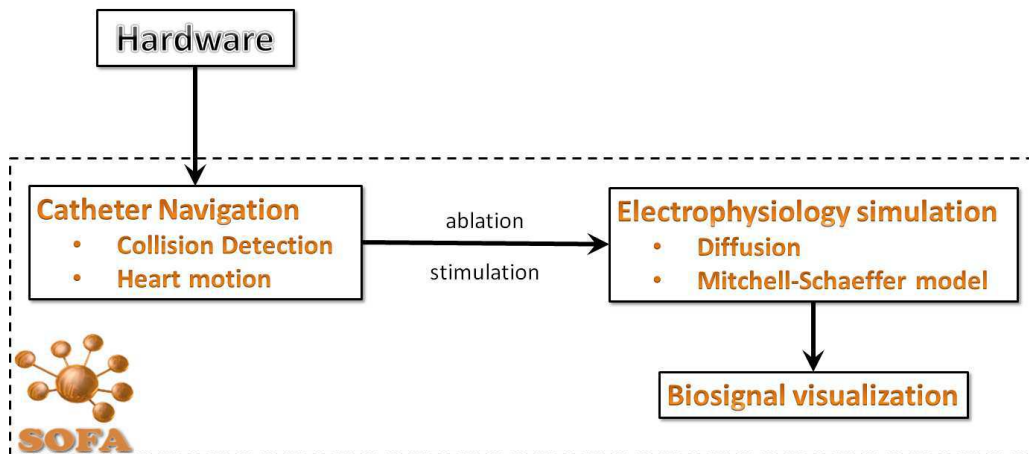


Figure 3: Structure of the simulator.

2.5 Main Challenges

We are planning to develop a new generation of cardiac endovascular simulator based on physics-based modeling of the cardiac electrophysiology. However, the following challenges needs to be met :

- **Real-time simulation of electrophysiology.** Simulation of cardiac electrophysiology usually requires very fine mesh of heart easily reaching almost a million (tetrahedral) elements. Thus simulation of electrophysiology propagation requires several hours or even days on large computers. A important challenge is to make this computation in real-time.
- **Simulation of Catheter Navigation inside moving structures.** Most simulation of catheter navigation has been done on static vascular structures. It is required to perform this navigation on a beating heart. Difficulty especially arises to properly compute the contact between catheters and vessel walls.
- **Software Integration.** This prototype requires to integrate two physical simulations (mechanical model of the catheters and electrophysiological model of the heart) together with a user interface.

3 EndoVascular Simulation

3.1 Reconstruction of the beating heart and vessels

2-3 patients were recruited for the VT-Stim study in WP6. More details on the patients' history and data acquisition is given in deliverable D6.3.2

The patients data acquisition comprised of pre-procedure image acquisition, involving 3D SSFP for heart anatomy, Thorax MR scan for BSPM mapping, cine MRI and tagged MRI for cardiac motion analysis, and High-resolution Late enhancement MRI (LE-MRI) for imaging the infarction extent for ischemia cases.

3D SSFP MR images were used to segment the patient-specific heart anatomical model of atria and ventricles, with the Philips plugin tool implemented in GIMIAS. A rigid registration based on DICOM headers was then used to register the cine MRI with the heart anatomical model from 3D SSFP MRI. Thus a 3D model of the whole heart was developed and registered with cine-MR for cardiac motion analysis (Fig. 4).

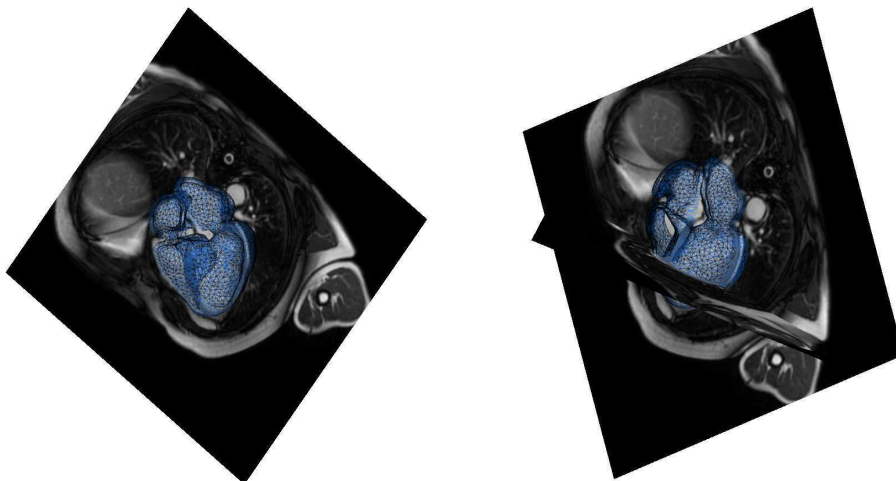


Figure 4: registered 3D mesh of the heart

As a pre-processing step, before the estimation of cardiac motion, we re-ordered the cardiac phases in the 3D+t cine MRI, so as to register the first cardiac phase with the phase of the whole heart model obtained from SSFP MR.

Ventricular motion (Fig. 5, Fig. 6) was then estimated from cine MRI using a Demon's based registration framework, which is explained in details in [MPS⁺11] a on ilog demons

Last, the estimated deformation field was resampled on to the vertices of the ventricles in the whole heart model.

The atrial motion was then computed simulating a simple elastic model and constraining the boundaries vertices between the atria and the ventricles with the ventricles motion. We plan to improve the atrial motion adding more constraints estimated from cine MRI. The motion for the cava vein is computed with the same technique used for the atria but an additional force should be apply during

the simulation to keep the vein and the right atrium connected.

Thus, we obtained a patient-specific beating heart model, to integrate in the simulation of endovascular navigation.

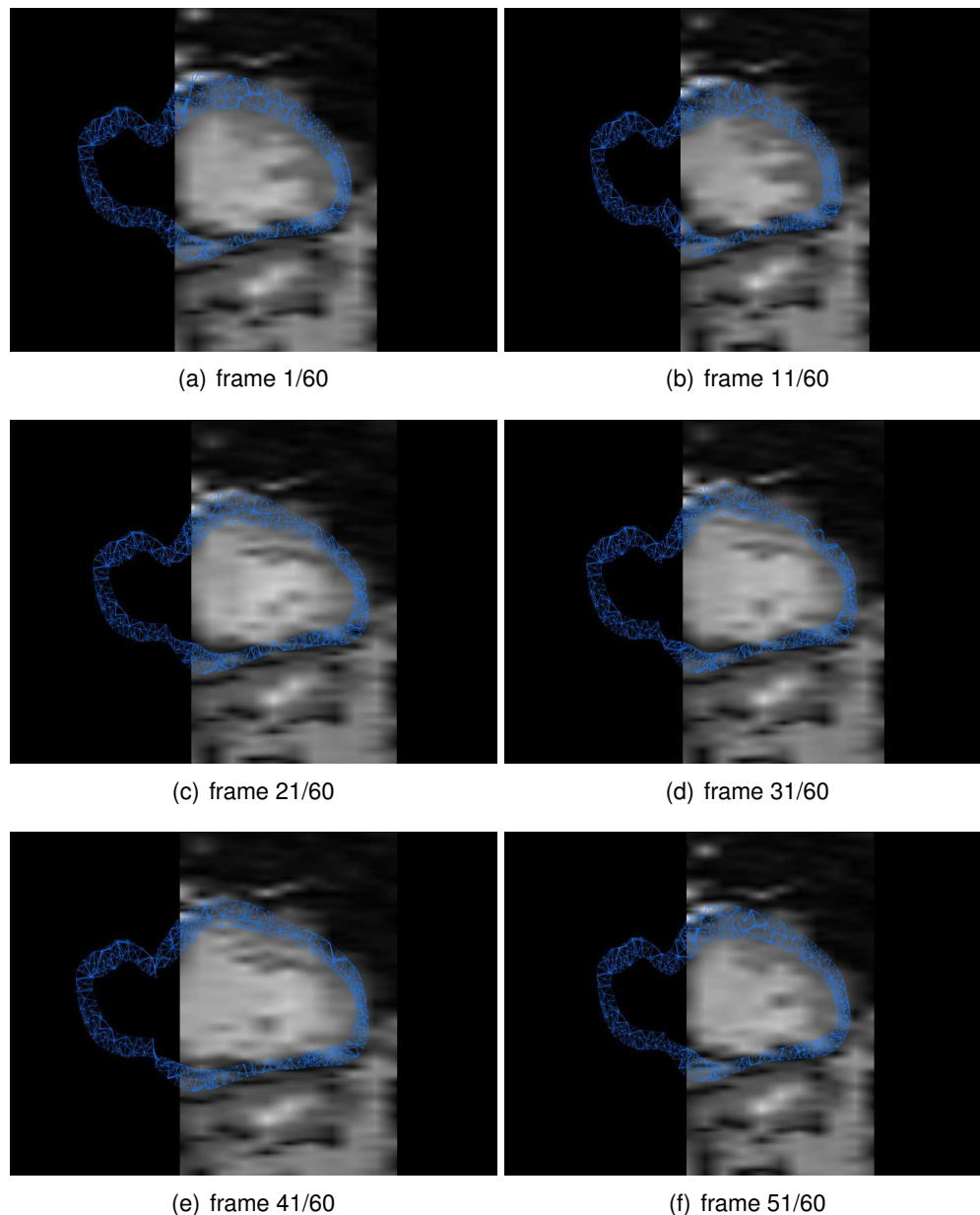


Figure 5: short axis view of the ventricular motion sequence

3.2 Main Principles of the Simulation of Endovascular Navigation

To simulate the endovascular navigation we take advantage of the multi-modal representation of the objects in SOFA. This important feature allows the possibility of using different geometries of the same object to describe the physics, graphics and collision model.

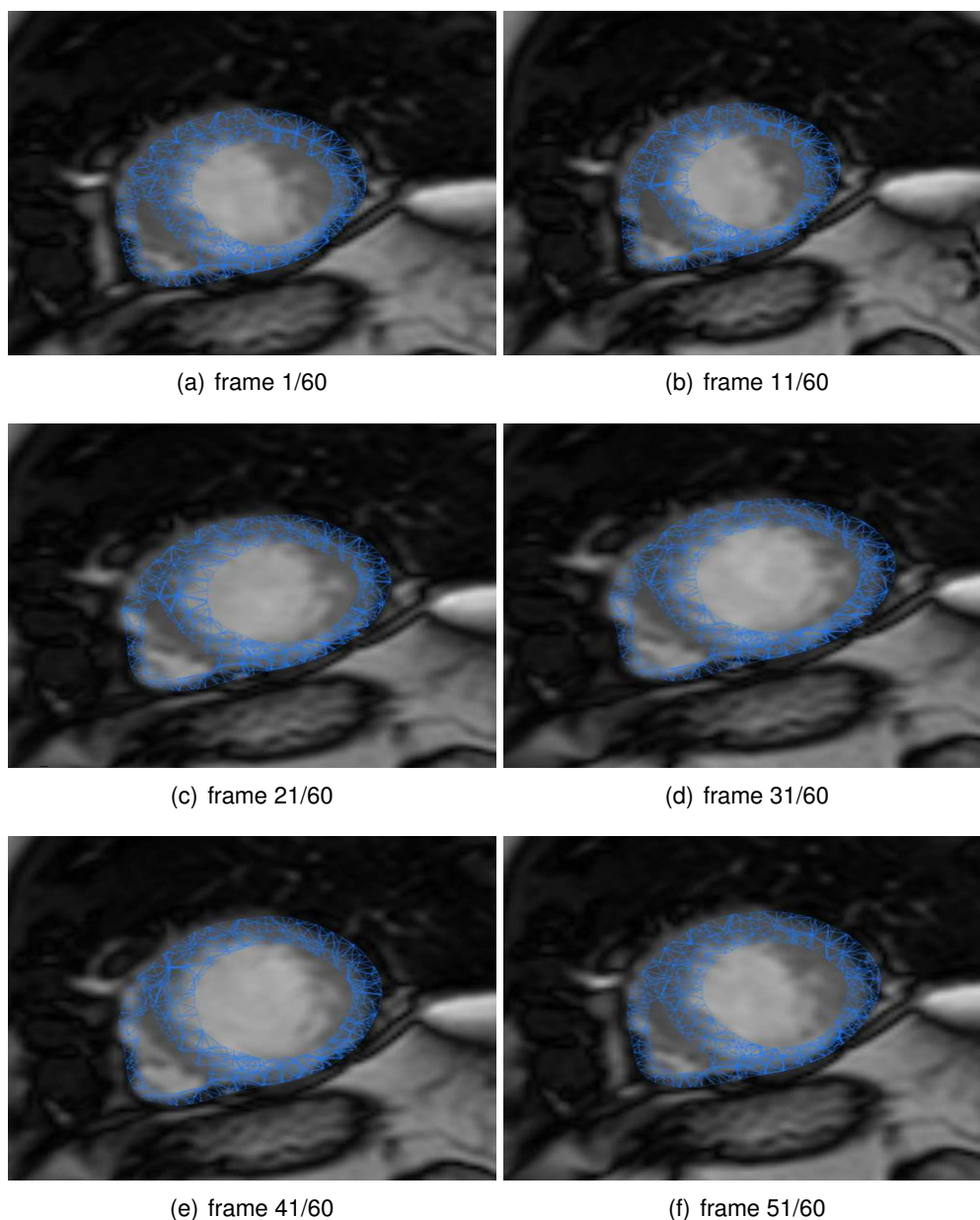


Figure 6: short axis view of the ventricular motion sequence

3.2.1 Physics simulation of catheter

To model the wire-like structure of the catheter and guidewire, a mathematical representation based on three-dimensional beam theory. The implementation used is described in [DCLN06] and [DMD⁺08]. The beam element model consists in a series of beam elements in a serially linked structure (Fig. 7, Fig. 8).

The dynamics of the mechanical system of connected beams is described by the following equation

$$M\ddot{x} + D\dot{x} + Kx = F \quad (1)$$

M is a diagonal mass matrix. D is the damping matrix. K is the global stiffness matrix computed by summing the contributions of each beam element stiffness matrix K_e . Due to the serially connected beam structure of the system K is a symmetric and tridiagonal matrix with a band size of 12. F is the sum of all the external and constraint forces acting on the mechanical system. Each beam extremity referred as node has 3 angular 3 spatial position degrees of freedom. The beam element stiffness matrix K_e is a 12x12 symmetric and it relates the degrees of freedom of node of a beam element to the forces and torques applied to them.

$$K_e u = f \quad (2)$$

where $u = x - x_0$ is the displacement vector of the degrees of freedom of the beam. The element stiffness matrix K_e depends on Young Modulus, Poisson's ratio, geometry and mass of the beam.

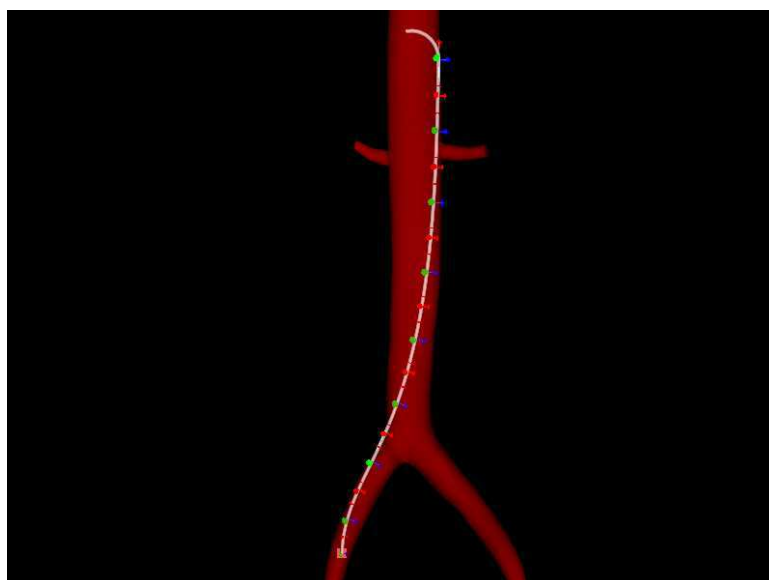


Figure 7: catheter constraint in the vessel

Contacts resolution is another important problem to solve during the navigation. We assumed the interaction between the catheter and the precomputed animated heart and the vessel walls is only one way. It means that heart and vessel walls motion can't be influenced by catheter. The contacts between the catheter and the heart and the vessel walls are solved implicitly using the Lagrange multiplier techniques.

Equation (1) is integrated in time using an implicit Euler integration scheme and then solved using the Thomas algorithm for block tridiagonal matrices.

3.2.2 Collision Detection

The collision models of the catheter, the vessels and the heart are represented as a set of geometric primitives. The wire-like structure of the catheter and guidewire

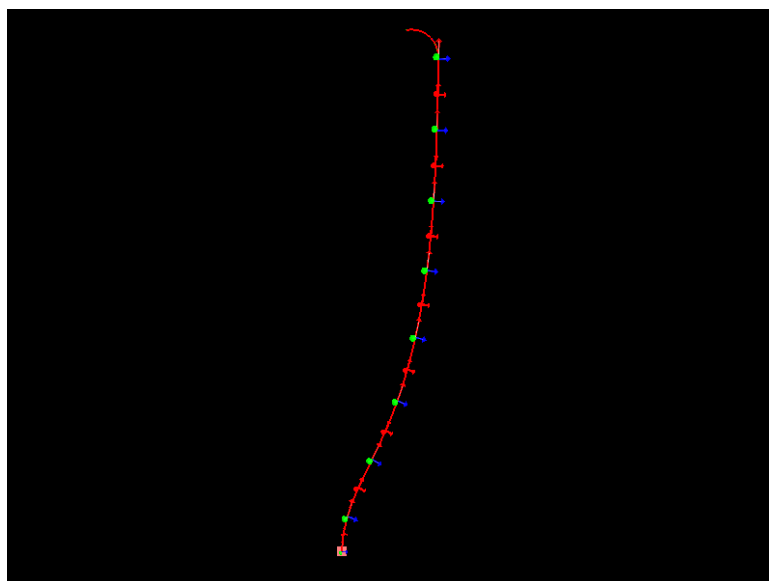


Figure 8: beam structure used to model the catheter

is represented as a series of points connected with segments (Fig. 10(a)). The vessels and the heart are represented as a set of triangles (Fig. 9(a)).

To minimize the computation cost of collision test between primitives two more representation of the objects are used, the bounding volumes and the bounding volume hierarchies (BVH).

The geometric primitives are encapsulated in axis aligned bounding boxes. These bounding boxes are simple geometrical objects that allows faster overlap test than primitives. Using bounding boxes allow for fast overlap rejection tests because it is necessary only to test for collisions against primitives when the bounding box intersection test is passed. When primitives overlap this additional test increases the computation time but in most situations few primitives are really close enough for their bounding volumes to overlap and it results in a great performance boost.

Second step is to arrange the bounding volumes into a tree hierarchy (Fig. 9(b), Fig. 10(a)) so that the bounding volumes forms the leaf nodes of the tree. These nodes are then grouped and recursively enclosed within a larger bounding volumes resulting in a binary tree structure with a single bounding volume at the top of the tree. This hierarchy of bounding volume prevent all the children to be examined if their parent volume is not intersected.

The collision test between two bounding volume hierarchies consists in establishing a descent algorithm rule and iteratively evaluated it and check against the overlap bounding volumes until leaf nodes are reached or no overlapping bounding volumes are found. A depth-first search traversal algorithm is used as a rule for how to descend the trees when their top-level volumes overlap.

All the primitives of each collision model are mapped into the physics models and at every time step their position is updated according to the simulation or animation behavior. This implies that before proceeding with the collision detection step all the bounding volume boxes and the tree hierarchy must be recomputed

at every time step.

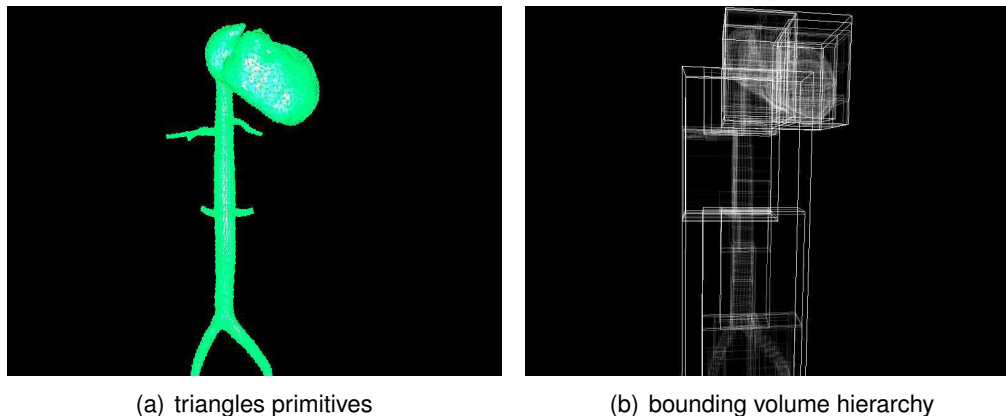


Figure 9: cava vein and heart collision model

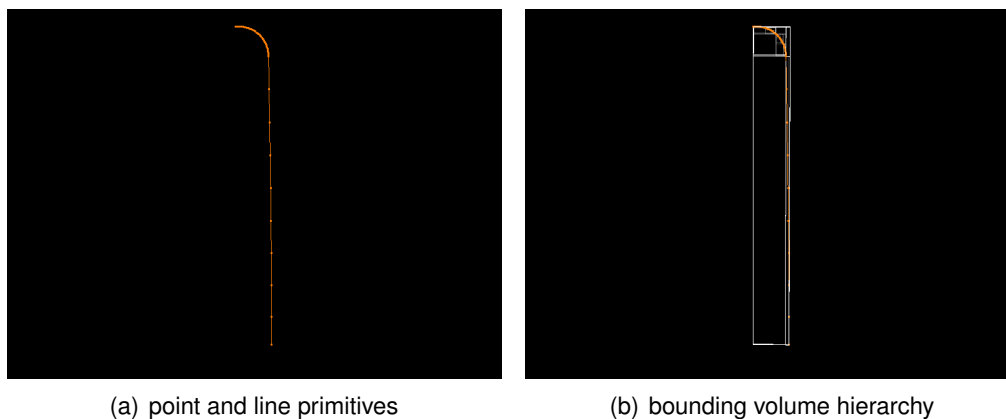


Figure 10: catheter collision model

3.2.3 Simulation of X-ray

According to the radiation absorption law the intensity of the primary beam passing through a medium has an exponential attenuation. Ignoring scattering and secondary emission of radiation the intensity is given by:

$$I_{out} = I_{in}e^{-\mu l} \quad (3)$$

where:

- I_{in} initial intensity of the beam
- I_{out} final intensity of the beam
- μ absorption coefficient of the medium
- l path length in the medium

the attenuation coefficient μ for each medium depends on the energy of the incident beam, on the material density and on the atomic number of the medium. For the Simulation of x-ray rendering we use an algorithm that can be suitable for the 3d surface models. The algorithm is described in [VGF⁺09] and our approach differs only in the implementation. Taking advantage of the current GPU hardware, we can reduce the number of rendering pass and speed up the simulation.

The first step to evaluate (3) is to compute the path length μ travelled by each ray of the beam through every object. Considering one 3D object the path length of each ray can be computed just considering all the intersection points between the ray casts from the x-ray source and the surfaces of the object.

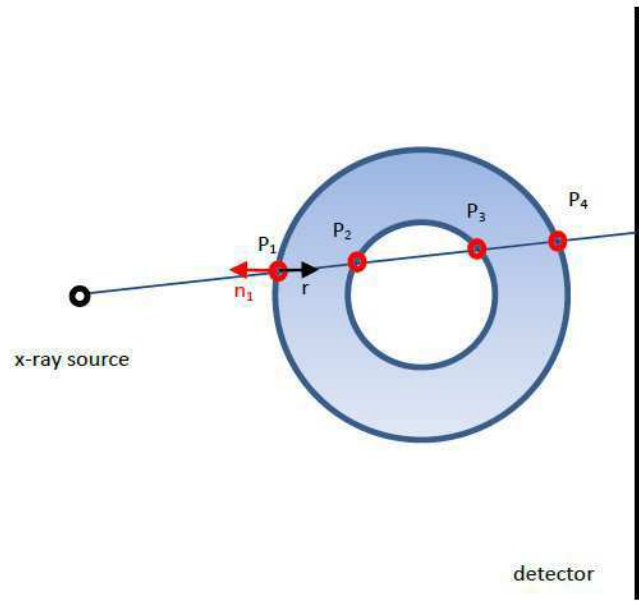


Figure 11: x-ray beam passing through an object

Considering the 2D representation in Fig. 11 the path length of the ray n in the medium is given by:

$$l_n = (P_2 - P_1) + (P_4 - P_3) \quad (4)$$

We need to add or subtract the positions of each intersection point P_n considering if the ray penetrates or leaves the object. To get this information we can compare the dot product between the direction of the ray casts and the normals of the surface. if the dot product is negative the ray is penetrating the object and if it's positive the ray is leaving the object. A drawback of this method is that only works if we have exactly the same number of penetrating and leaving intersecting point for each ray. This means that the method only works with closed 3D object.

$$l_n = \sum_i \text{sgn}(\vec{r}_i \cdot \vec{n}_i) P_i \quad (5)$$

The second step is to compute all the exponent term for each beam multiplying the μ absorption coefficient of each object crossed by the path length l_n of

the ray n . We consider the absorption coefficient as a constant and we can just multiply l_n by μ and then sum for each crossed object:

$$\sum_{obj} (\mu l_n) = \sum_{obj} \mu_{obj} \times \left(\sum_i \text{sgn}(\vec{r}_i \cdot \vec{n}_i) P_i \right) \quad (6)$$

In the final step we can compute the intensity attenuation for each ray computing the exponential

OpenGL implementation: the x-ray rendering algorithm is implemented entirely on the GPU using OpenGL and GLSL shading language.

The algorithm consists in two rendering passes. In the first pass (Fig. 12(a)) we compute all the exponential term and render it in a floating point texture attach to a Frame Buffer Object. The camera position is set up to match the x-ray source position.

We use the vertex shader to compute the position of each vertex and to determine if the vertex normal is facing or not the source of the beam. To be able to use only one rendering pass the idea is to assign for each vertex his front and back color value to 1 and -1 activating the backfacing writing OpenGL state option. In this way the GPU processing pipeline automatically will use as color in the fragment shader the front or back according to the direction of the normal of the vertices. To take in account all the vertices in the rendering processing pipeline the depth test and the face culling must be disabled.

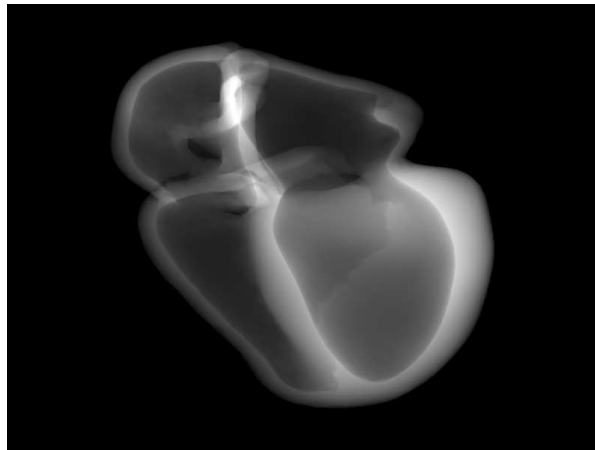
In the fragment shader We sum according with the sign of the color the product of distance between the position of each fragment and the point of view with the absorption coefficient and store it in a floating point texture. To sum all the contribution from each object crossed the OpenGL blending state must be activated. The color clamping must be disabled to let value greater than 1 and smaller than 0 be accumulated in the floating point texture.

Using the correct blender factor we end up to have in the first rendering pass the value of the exponent expressing the total length path multiplied by the absorption coefficient. The second pass (Fig. 12(b)) only a fragment shader is used to compute the exponential attenuation for each fragment and render it directly to the render buffer. Finally some brownian noise textures are added and blended in the render buffer (Fig. 12(c)).

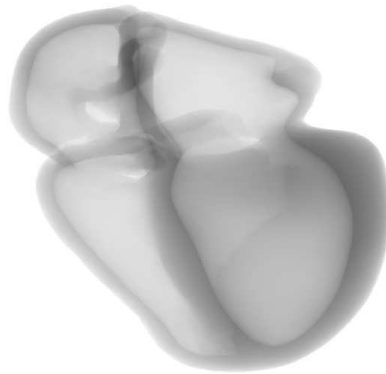
3.3 Results on the Simulation of Endovascular Navigation in the Beating Heart

The results for the navigation in the vessels (Fig. 13) are quite promising and there are still two problems to solve when the catheter navigates inside the beating heart.

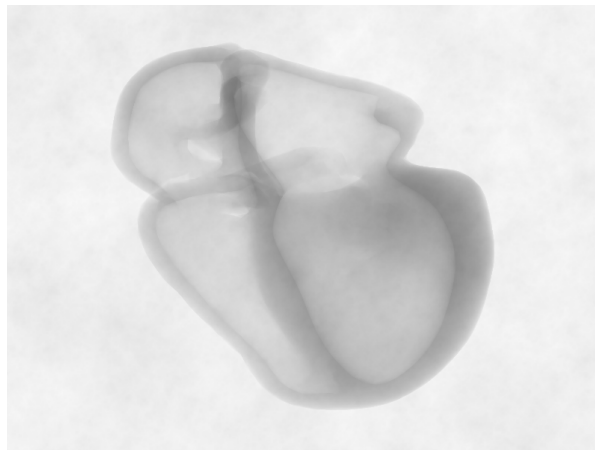
The first is to find the right way to solve the contacts between the catheter and the animated surface of the heart in order to get a better behavior of the collision response force. The second is to reduce the simulation speed drop when many contacts are generated. In this situation the computational cost for the contacts resolution become very high and the frequency of the simulation step can easily drop to few frames per second.



(a) first pass



(b) second pass



(c) with noise texture

Figure 12: X-Ray rendering of the heart. μl render in a floating point texture (a). Absorbed intensity (3) computed and rendered in the render buffer (b). Final blended with a brownian noise texture (c).

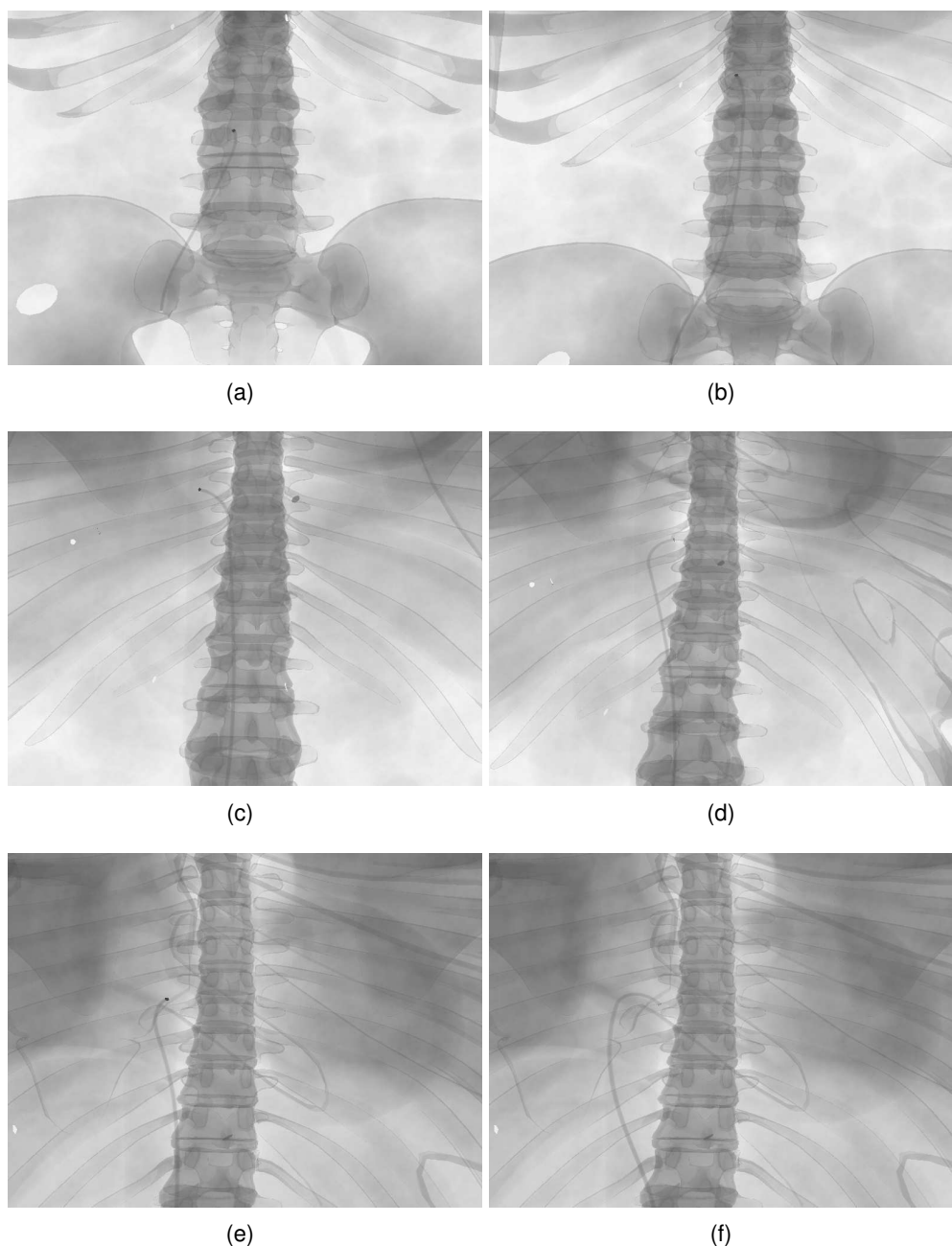


Figure 13: screenshots of the endovascular navigation

4 ElectroPhysiology Simulation

After considering the navigation aspect, we now focus on the simulation of the cardiac electrophysiology. This step is the crucial step of the simulation. The myocardial cells are polarised, i.e. there is a potential difference between the inside and the outside of the cells. This potential difference is called action potential. The electrical wave propagating through the heart walls is due to ion exchanges between cells, i.e. modifying the action potential.

To meet the requirements of our training simulator, our computations have to be fast enough to allow user interaction. We also want to simulate the thermoablation procedure. In this part, we will detail different biophysical models and concentrate on the Mitchell-Schaeffer model that we chose. We will propose optimizations on this model for faster computations. Finally, we will present our current results.

4.1 Biophysical Models of Cardiac Electrophysiology suitable for fast computation

Since the last decades, several models have been developed to simulate the cardiac electrophysiology. These models can be classified depending on their complexity and on the cardiac scale that is considered: (i) the biophysical models are complex models including many parameters and simulating the electrophysiology at a cellular scale ([TNNP04]). (ii) Phenomenological models are simplified models ([RF61], [AP96], [FK98], [MS03]) derived from the biophysical model. This second category includes much less parameters and captures the action potential shape and its propagation at the organ scale. (iii) The Eikonal models ([Kee91], [Jac10]) correspond to a static non-linear PDE for the depolarisation time derived from the previous models. These models enable to simulate wave propagation but they can't account for complex physiological states (such as reentries) and the model parameters have no physiological meanings. Due to the requirements about the computation time, we focused on the phenomenological models.

4.2 Optimized Mitchell-Schaeffer Model

The model that we chose is the Mitchell Schaeffer model because of the following reasons: (i) it has only 5 parameters that we detail below, (ii) each parameter has a physiological meaning and (iii) it provides a better estimation of the AP compared to other phenomenological models (as the Aliev-Panfilov model [AP96]).

The MS model is a two-variable model derived from the Fenton Karma model. The equations describing the model are written in 7:

$$\begin{cases} \partial_t u = \operatorname{div}(D\nabla u) + \frac{zu^2(1-u)}{\tau_{in}} - \frac{u}{\tau_{out}} + J_{stim}(t) \\ \partial_t z = \begin{cases} \frac{(1-z)}{\tau_{open}} & \text{if } u < u_{gate} \\ \frac{-z}{\tau_{close}} & \text{if } u > u_{gate} \end{cases} \end{cases} \quad (7)$$

where u is a normalized transmembrane potential¹ and z is the gating variable associated to the sodium ion influx, thus depicting the repolarization phase. The diffusion term is defined by an 3x3 anisotropic diffusion tensor $D = d \cdot \operatorname{diag}(1, r, r)$ so that the conductivity in the fiber direction is 2.5 times greater than in the transverse plane ($r = \frac{1}{(2.5)^2}$). d is the diffusion coefficient.

¹The MS is a monodomain model since it is expressed according to the transmembrane potential whereas bidomain models depends on both intra- and extra-cellular potential.

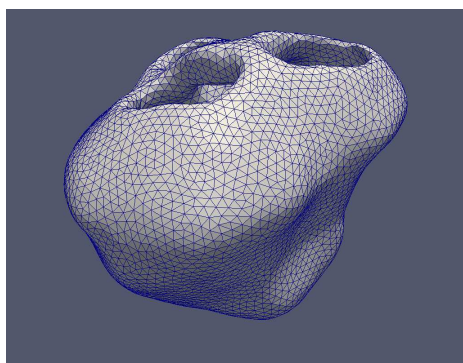


Figure 14: 3D patient-specific mesh (with 65547 linear tetrahedra).

The parameters τ_{in} and τ_{out} define the repolarization phase whereas the constants τ_{open} and τ_{close} manage the gate opening or closing depending on the change-over voltage u_{gate} . The term $J_{stim}(t)$ is the stimulation current applied in the pacing area. The default values (describing the common action potential) of these parameters are given in the work of Mitchell Schaeffer.

Targeting real-time computations is very challenging due to the stiffness of the equations involved. In order to use larger time steps, we first propose an adaptive parametrisation of the Mitchell-Schaeffer model. By modifying the time constants τ_{in} and τ_{out} of the model, we slightly decrease the slopes of the potential shape. This optimization keeps accurate computations of conduction velocity while allowing the use of larger time steps.

The other optimization results in a GPU implementation of the electrophysiology. Previous work [BCG⁺11] already focused a GPU implementation for electrophysiology. However till now only 2D simulations have been studied. We implemented the 3D anisotropic diffusion and the Mitchell Schaeffer term on GPU with the help of CUDA programming to decrease the computation time using recent methods.

4.3 Results of Optimized Mitchell-Schaeffer Model

A simulation dedicated to a training system assumes that cardiologists will interact with it. These optimizations detailed previously allowed us to reach computations less than 3 times slower than real-time, thus making our simulation interactive.

	CPU	GPU	GPU with lower CG tolerance
Mean computation time for one time step (ms)	25.4	3.89	2.17
real-time ratio (slower than real-time)	75.8	7.04	2.99

Table 1: Performance comparison between CPU and GPU.

Our simulations already allows to stimulate any area of the heart in real-time

(constraint with $u = z = 1$). This allows to simulate the stimulation done by the surgeon before the ablation in order to set the definitive diagnosis.

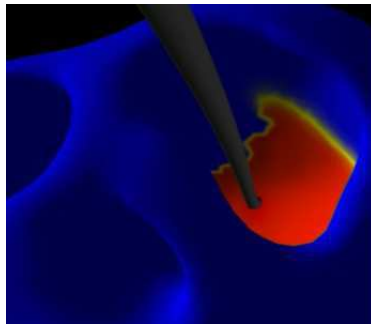


Figure 15: Excitation of the pericardium with a tool monitored by the user.

The procedure of thermo-ablation is also handled by our computations. The cardiac cells treated by thermo-ablation can not conduct the current anymore. These cells can be seen as zero conductivity area. To simulate this procedure, we have to take these changes into account in the system resolution which can be done without affecting the computation times with our solvers.

These results are therefore very encouraging in the development of our thermo-ablation simulator. Some improvements are still to be done on the GPU code. New methods are currently developed to reach real-time computations. The interaction of a catheter inside the ventricule is also a work in progress.

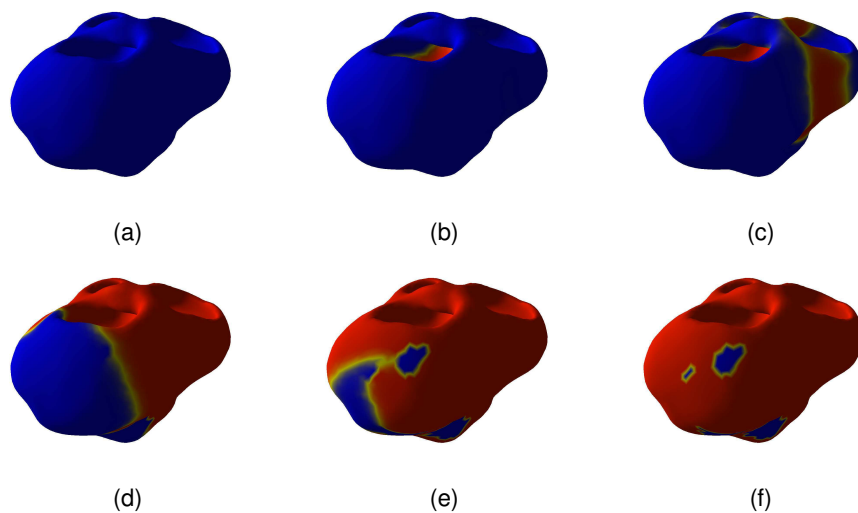


Figure 16: Depolarization wave (in red) propagating through the heart.

5 Discussion and Conclusion

5.1 Conclusions

5.2 Limitations

5.3 Perspectives

6 Acknowledgements

The authors would like to thank

Appendix

Bibliography

- [AP96] R. R. Aliev and A. V. Panfilov. A simple two-variable model of cardiac excitation. *Chaos, Solitons and Fractals*, 7(3):p. 293–301, 1996.
- [BCG⁺11] E. Bartocci, E. M. Cherry, J. Glimm, R. Grosu, S. A. Smolka, and F. H. Fenton. Toward real-time simulation of cardiac dynamics. In *Proceedings of the 9th International Conference on CMSB*, pages p. 103–112. ACM, 2011.
- [CC] CAE-CathLab. <http://www.cae.com/en/healthcare/home.asp>.
- [CCM⁺11] Patricia Chiang, Yiyu Cai, Koon Hou Mak, Ei Mon Soe, Chee Kong Chui, and Jianmin Zheng. A geometric approach to the modeling of the catheterheart interaction for vr simulation of intra-cardiac intervention. *Computers Graphics*, 35(5):1013 – 1022, 2011.
- [CDL⁺05] S. Cotin, C. Duriez, J. Lenoir, P. Neumann, and S. Dawson. New approaches to catheter navigation for interventional radiology simulation. In James Duncan and Guido Gerig, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, volume 3750 of *Lecture Notes in Computer Science*, pages 534–542. Springer Berlin / Heidelberg, 2005.
- [DCLN06] C. Duriez, S. Cotin, J. Lenoir, and P. Neumann. New approaches to catheter navigation for interventional radiology simulation 1. *Computer Aided Surgery*, 11(6):300–308, 2006.
- [DMD⁺08] Jrmie Dequidt, Maud Marchal, Christian Duriez, Erwan Kerien, and Stephane Cotin. Interactive simulation of embolization coils: Modeling and experimental validation. In Dimitris N. Metaxas, Leon Axel, Gabor Fichtinger, and Gbor Szekely, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2008, 11th International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part I*, volume 5241 of *Lecture Notes in Computer Science*, pages 695–702. Springer, 2008.
- [FK98] F. Fenton and A. Karma. Vortex dynamics in three-dimensional continuous myocardium with fiber rotation. *Chaos*, 8(1):p. 20–47, 1998.
- [Jac10] V. Jacquemet. An eikonal approach for the initiation of reentrant cardiac propagation in reaction-diffusion models. *Biomedical Engineering, IEEE Transactions on*, 57(9):p. 2090 –2098, Sept. 2010.

- [Kee91] James P. Keener. An eikonal-curvature equation for action potential propagation in myocardium. *Journal of Mathematical Biology*, 29:p. 629–651, 1991.
- [LCDN06] Julien Lenoir, Stephane Cotin, Christian Duriez, and Paul Neumann. Interactive physically-based simulation of catheter and guidewire. *Computers Graphics*, 30(3):416 – 422, 2006.
- [LHG⁺09] Vincent Luboz, Chris Hughes, Derek Gould, Nigel John, and Fernando Bello. Real-time seldinger technique simulation in complex vascular models. *International Journal of Computer Assisted Radiology and Surgery*, 4:589–596, 2009.
- [LRRV00] W. Lawton, R. Raghavan, S.R. Ranjan, and R.R. Viswanathan. Tubes in tubes: catheter navigation in blood vessels and its applications. *International Journal of Solids and Structures*, 37(22):3031 – 3054, 2000.
- [Men] Mentice. <http://www.mentice.com/>.
- [MPS⁺11] Tommaso Mansi, Xavier Pennec, Maxime Sermesant, Hervé Delingette, and Nicholas Ayache. iLogDemons: A demons-based registration algorithm for tracking incompressible elastic biological tissues. *International Journal of Computer Vision*, 92(1):92–111, 2011.
- [MS03] C. Mitchell and D. Schaeffer. A two-current model for the dynamics of cardiac membrane. *Bulletin of Mathematical Biology*, 65:p. 767–793, 2003.
- [RF61] Richard and FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):p. 445–466, 1961.
- [Sim] Symbionix. <http://symbionix.com/simulators/angio-mentor/>.
- [TNNP04] K. H. W. J. Ten Tusscher, D. Noble, P. J. Noble, and A. V. Panfilov. A model for human ventricular tissue. *American Journal of Physiology - Heart and Circulatory Physiology*, 286(4):p. 1573–1589, April 2004.
- [VGF⁺09] F. P. Vidal, M. Garnier, N. Freud, J. M. Ltang, and N. W. John. Simulation of x-ray attenuation on the gpu. pages 25–32, June 2009.