GPU-based High Performance Wave Propagation Simulation of Ischemia in Anatomically Detailed Ventricle

Lei Zhang[1, 2], Changqing Gai[1], Kuanquan Wang[1], Weigang Lu[1], Wangmeng Zuo[1]

[1]Harbin Institute of Technology, Harbin, China
[2]Harbin University, Harbin, China

Abstract

In this study, we present a framework to simulate multi-scale wave propagation of ischemia which leverages the high- performance computing capacity of Graphic Processing Units (GPU). To cope with the no-flux boundary condition and address the branch deficiency, a phase-field method is employed. An on-the-fly visualization method of two dimensional simulation results is proposed. A fusion volume visualization method of both simulation result and anatomy structure data is provided to show ischemia with supporting spatial information in the heart. The experimental results show that the speedup of our GPU-based simulation method is about at least 20 times faster than that of conventional CPU-based simulation method.

1. Introduction

Myocardial ischemia is caused by the reduction or blocking of the blood supplying to myocardium, and usually leads to abnormal wave propagation and probably results in sudden death. Action potential propagation simulation of ischemia in 3D anatomically detailed ventricle is valuable in studying the mechanisms and dynamics of ischemia-induced re-entry and arrhythmia [1]. However, such simulations are computationally intensive [2], making efficient numerical solution urgently necessary.

With the advent of GPU, general-purpose computing on GPUs (GPGPU) becomes an emerging technology to enhance computational efficiency [3, 4]. Instead of CPUs, GPGPU uses GPUs to do large simulations in parallel. Comparing to CPUs which containing about 8 or more cores processor, GPUs are massively parallel single instruction multiple data processing units with 128~240 stream processors. Researchers has take advantage of GPU to simulate electric activity of mammalian hearts and the speedups are satisfactory [5, 6]. In this study, we present a GPU-based framework to simulate the ischemia wave propagation of human heart and visualize the results with different rendering approaches.

The remainder of this paper is organized as follows. In Section 2, we first introduce the GPU high-performance computing (HPC) architecture, then briefly introduce the TNNP model, and finally present our simulation approach and the visualizing method of the simulating results. In Section 3, we evaluate our proposed framework. Finally, in Section 4 we provide our conclusion and future work.

2. Methods

The main workflow of the proposed framework is shown in figure 1. First the anatomical data of human heart is used to provide the geometrical information. Second the phase-field method is used to compute the spatial information of each point in the human heart with the help of geometrical information. Then a file which contains the phase-field value of each point is generated. After that the file can be used to provide the information which is used to simulate the wave propagation under condition of ischemia. And the file is fetched into the main memory and transferred into the texture memory of the graphic card. The simulation method is executed on the GPU. After the simulation results generated, they then are stored onto the hard disk or directly visualized via the methods described in the subsection on the screen. The subsections are going to provide the detailed implementation of the framework in this study.

2.1. GPU HPC Architecture

With the GPU's rapid evolution, modern GPU is not only a powerful graphics engine but also a highly parallel programmable processor with HPC capability [7]. Also the way to drive GPU became easier with the help of CUDA (Compute Unified Device Architecture) architecture [8]. CUDA is a heterogeneous hardware and software co-processing architecture for general-purpose computing supported by both GPU and CPU. Code runs on GPU side is called kernel. When kernel is executed, it is implemented as a series of threads running independently in parallel. In this study, the simulation is implemented as kernel running on the GPU side.

ISSN 0276−6574 Computing in Cardiology 2011;38:469−472.

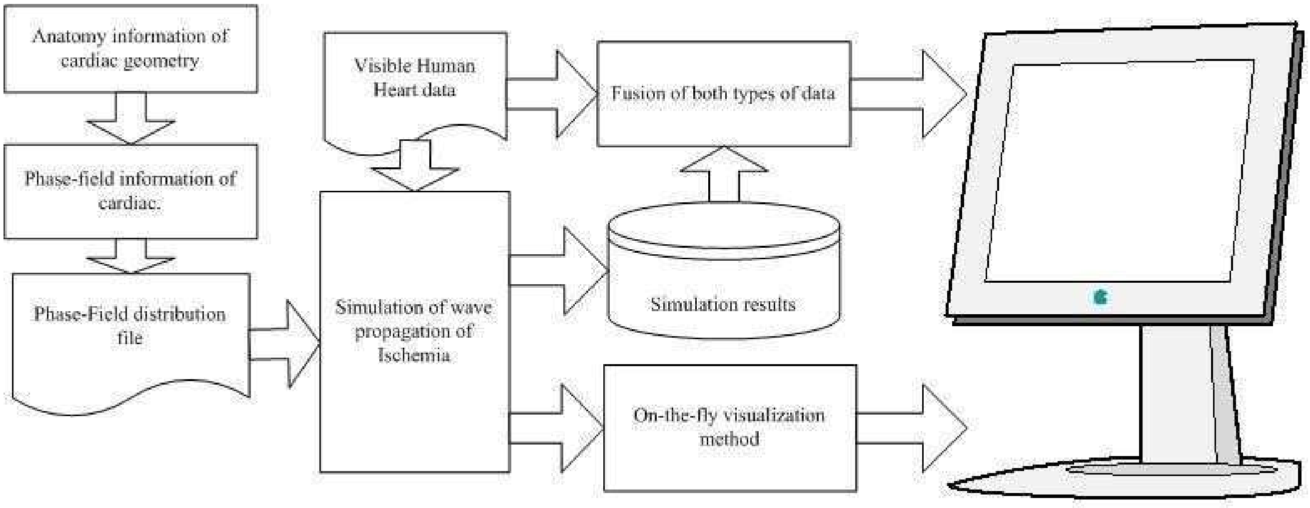


Figure 1. The architecture of the presented framework.

2.2. Model of cardiac tissue

We use the TNNP model [9] of human ventricle to represent the excitable dynamics of cardiac tissue. The following equation defines the transmembrane voltage V:

$$\begin{cases} \dfrac{\partial V_m}{\partial t} = -\dfrac{I_{ion}+I_{stim}}{C_m} + \nabla\cdot(D\nabla V_m) \\ I_{ion} = I_{Na}+I_{K1}+I_{to}+I_{Kr}+I_{Ks}+I_{CaL}+I_{NaCa}+I_{NaK}+I_{pCa}+I_{pK}+I_{bCa}+I_{bNa} \end{cases} \quad (1)$$

where ∇ denotes the gradient operator, C_m is the cellular capacitance, and D is the diffuse tensor. I_{ion} and I_{stim} are sum of all transmembrane ionic currents and externally stimulus current, respectively.

2.3. Numerical approach

To automatically handle the boundary conditions of anatomical heart geometries, a phase-field method is employed [10]. An auxiliary field ϕ is introduced, which has a value of 1 inside the ventricle and 0 outside the ventricle but within the bounding box [11]:

$$\phi(\xi) = \begin{cases} 1, \xi \in \Omega_{ventricle} \\ 0, \xi \in (\Omega_{bounding-box} - \Omega_{ventricle}) \end{cases} \quad (2)$$

The following equation is used to solve the value of ϕ:

$$\frac{\partial \phi}{\partial t} = \xi^2 \nabla^2 \phi - \frac{\partial G(\phi)}{\partial \phi} \quad (3)$$

where ξ is used to control the width of the interface between ventricle and the bounding box. $G(\phi)$ is a double-well function with minima at $\xi \in \Omega_{ventricle}$ and $\xi \in (\Omega_{bounding-box} - \Omega_{ventricle})$. We choose the function as follows:

$$G(\phi) = -\frac{(2\phi-1)^2}{4} + \frac{(2\phi-1)^4}{8} \quad (4)$$

and Eq. 4 is modified as:

$$\phi\frac{\partial V}{\partial t} = -\phi\frac{I_{ion}+I_{stim}}{C_m} + \nabla\cdot(D\phi\nabla V) \quad (5)$$

The contributions of the phase-field to our framework are twofold. On one side, we can unify our simulation scheme. When the anatomy structure changes, we just need to change the phase-field distribution. On the other side, the GPU implementation can avoid the judgments of the boundaries which will lead to branch causing deficiency. In this study, Eq. 5 is integrated with a space step 0.15mm and a time step 0.02ms. ξ is 0.33mm, and the forward Euler scheme is adopted.

2.4. On-the-fly simulation data visualization

In this subsection, we introduce the method we used to visualize the two dimensional (2D) data. Using conventional method, we first do the simulation, then transfer the simulation results from CPU to GPU, and visualize the results with the help of visualizing toolkits. The disadvantages are lost of time-vary feather of the data and need of large storage and GPU-CPU data transfer. Our method is to visualize the simulation result on the fly. After the simulation result at current time step completed, our method will visualize the current result and do next time step simultaneously, which avoid GPU-CPU data transfer in conventional CPU based simulation. Figure 2 shows our rendering results of a time sequence. The method is implemented using the Graphics Interoperability with OpenGL provided by CUDA. Also spatial locality of 2D texture memory is harnessed to increase the performance.

2.5. Visualization fusion of different type data

Multimodal visualization is the visualization of more than one data set from different modalities [12]. In this section, we provide a multimodal visualization method to visualize the multimodal volume data which include both simulation and anatomy data. We adopt the GPU-based ray casting algorithm to simultaneously visualize two types of data. The two different modalities can enhance the in-depth analysis. The ischemia parts which are necessary might be visible in simulation data and not in the other one and vice versa. Therefore, it is possible to see the necessary part for a heart diagnosis in the combination of two different data sets. The rendering results of traditional ray casting method and our proposed rendering method are shown in figure 3. With the help of ghost from anatomy data, the ischemia parts are supported by spatial information.

3. Results

In this section, we use the female heart data of Visible Human Project providing anatomically detailed information, to evaluate the performance of our proposed simulation method.

The simulations are run on an Intel(R) Core (TM) 2 Duo CPU E7500, NVIDIA Tesla C1060 GPU. The visualizations are performed on a NVidia GeForce 9600 GPU. The wave propagation simulation of ischemia kernel code is written in CUDA 3.2. The on-the-fly data is implemented by OpenGL and the shaders of visualization fusion of both types of data is coded by Cg 3.0. The comparison of the traditional simulation method using CPU and our proposed method tested on 2D simulation is shown in Table 1.

Table 1. 2D simulation on CPU and GPU.

CPU	Tesla C1060 GPU
350ms	11ms

The speedup is about 31.8.

When the proposed simulation method applied to 3D wave propagation, the speedup is about 25.7 according to the data listed in Table 2.

Table 2. 3D simulation on CPU and GPU.

CPU	Tesla C1060 GPU
2700s	105s

Results show that the GPU-based method provided in this work can greatly shorten the simulation time comparing to the conventional CPU method.

4. Discussion and conclusions

In this paper, we present a GPU-based framework for simulating the effects of ischemia on wave propagation. We also provide an on-the-fly visualization method for 2D simulation results and a modified ray casting method for 3D simulation results. The experimental results show that the speedup is at least 20 times faster than traditional simulation method. And we find a new way to analyzing the simulation results, not only analyzing after storage but also on-the-fly.

In the future, we will study on visualization method fusion of time varying functional data and anatomical data, and eventually integrate our method to a GPU-based analysis platform with both cardiac simulation and visualization functionalities.

Acknowledgements

The work is partially supported by the National Natural Science Foundation of China (NSFC) under Grants No. 60872099 and No. 61173086.

References

[1] Bernus O, Verschelde H, Panfilov AV. Reentry in an anatomical model of the human ventricles. International Journal of Bifurcation and Chaos 2004; 13(12): 3693-3702.

[2] WG Lu, KQ Wang, WM Zuo, TJ Liu, HG Zhang. Simulation of Effects of Ischemia in 3D Human Ventricle. Computer in Cardiology. 2009: 477 – 480.

[3] Lindholm E, Nickolls J, Oberman S, Montrym J. NVIDIA Tesla: A Unified Graphics and Computing. IEEE Micro 2008; 28(2):39-55.

[4] Luebke D. CUDA: Scalable parallel programming for high-performance scientific computing. International Symposium on Biomedical Imaging 2008; 836-838.

[5] Sato D, Xie Y, Weiss JN, Qu Z, Garfinkel A, Sanderson AR. Acceleration of cardiac tissue simulation with graphic processing units. Medical and Biological Engineering and Computing 2009; 47(9): 1011-1015.

[6] Vigmond EJ, Boyle PM, Leon LJ, Plank G. Near-real-time simulations of bioelectric activity in small mammalian hearts using graphical processing units. Engineering in Medicine and Biology Society 2009; 3290-3293.

[7] Nickolls J, Dally WJ. The GPU Computing Era. IEEE Micro 2010; 30(2): 56-69.

[8] NVIDIA. NVIDIA CUDA Programming Guide. http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf. 2010.

[9] Tusscher KT, Noble D, Noble PJ, Panfilov AV. A model for human ventricular tissue. Heart and Circulatory Physiology 2004; 286(4):1573-1589.

[10] Fenton FH, Cherry EM, Karma A, Rappel WJ. Modelling wave propagation in realistic heart geometries using the phase-field method. Chaos 2005; 15: 013502.

[11] Bueno-Orovio A, Pérez-García VM, Fenton FH. Spectral

methods for partial differential equations in irregular domains: the spectral smoothed boundary method. Journal of Scientific Computing 2006; 28(3): 886-900.
[12] Haidacher M, Bruckner S, Kanitsar A, Gröller ME. Information-based Transfer Functions for Multimodal Visualization. Eurographics Workshop on Visual Computing for Biomedicine 2008; 101-108.

Address for correspondence.

Kuanquan Wang
Mailbox 332, Harbin Institute of Technology
Harbin 150001, China
E-mail: wangkq@hit.edu.cn

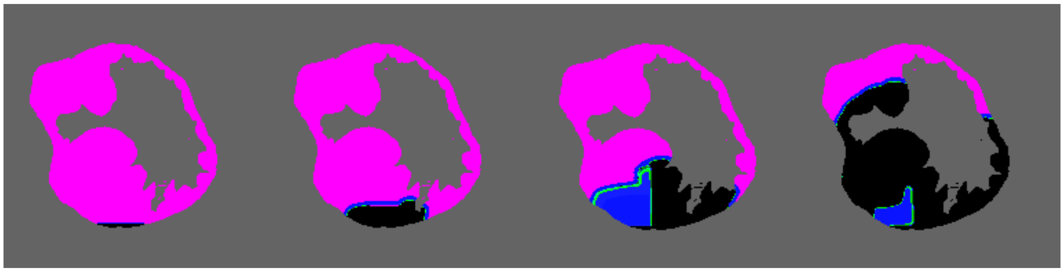
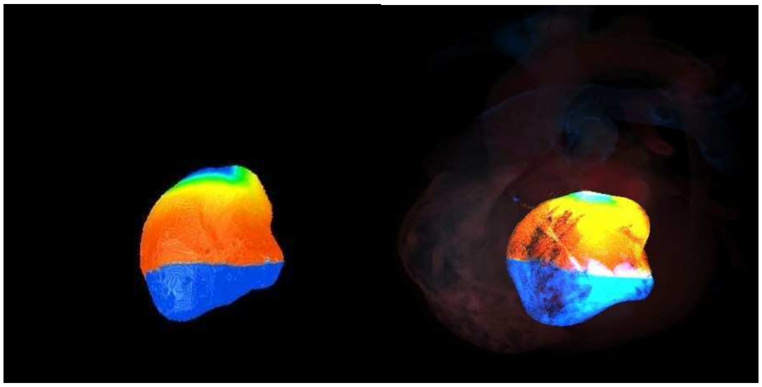


Figure 2. Rendering results of a 2D simulation time sequence using on-the-fly rendering method.



(a) (b)

Figure 3. Rendering results of 3D wave propagation simulation. (a) is the traditional ray casting method and (b) is the proposed method in this study.