Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

T. Arola, M. Hannula, N. Narra, J. Malmivuo, J. Hyttinen

Software Suite for Finite Difference Method Models

Abstract—**We have developed a software suite for finite difference method (FDM) model construction, visualization and quasi-static simulation to be used in bioelectric field modeling. The aim of the software is to provide a full path from medical image data to simulation of bioelectric phenomena and results visualization. It is written in Java and can be run on various platforms while still supporting all features included. The software can be distributed across a network utilizing dedicated servers for calculation intensive tasks. Supported visualization modes are both two- and three-dimensional modes.**

INTRODUCTION

SIMULATION of bioelectric source field problem requires accurate medical image based geometry, mesh generation and source setup e.g. defining the boundary conditions for a model. For these tasks flexible, efficient and user friendly software is needed. In bioelectric source field problems huge data sets are encountered for which efficient and highly tuned simulation software is also a prerequisite.

Finite Difference Method (FDM) has been used to simulate current behavior in realistic shaped human torso models [1]. The benefit in using FDM is its ability to directly utilize a segmented data set as a mesh. Realistic shaped human models are created from medical imaging data such as magnetic resonance imaging (MRI) or computed tomography (CT) by utilizing segmentation methods to separate tissues from each others. The segmentation process is all but simple and straightforward and it has been extensively studied. The result from a segmentation process can be turned into a simulation model by assigning conductivity values and boundary conditions for calculation elements.

The main difficulty encountered in especially FDM modeling is the assignment of boundary conditions and sources. Models are highly complex and without proper 2D and 3D visualization it is often difficult to define the properties for calculation nodes. Simulation and result visualization suffer easily from the sheer number of calculation elements or lead field data encountered in FDM modeling.

To overcome these problems we have developed a software suite of three collaborating programs for FDM simulation. The idea is to provide a full path from medical images to visualization of simulation results such as potential or current distribution in the model. The software is capable of reading DICOM format image data and visualizes it in 2D and 3D. The Segmentation program incorporates various segmentation tools that can be used to create and organize segmentations. These can be turned into FDM models by assigning boundary conditions and conductivity values for the tissues in Modeler program. Sources can be defined as current sources or forced potentials. Simulation program calculates potentials or currents in the model by using a user selectable FDM stencil and formulation. Also modeling with reciprocally energized lead fields [2] can be calculated easily. The results from a simulation can be visualized together with the original imaging data, created model, or they can be layered with other simulation results. Written in Java, the software can be run also through a web browser via Java Web Start.

I. MATERIAL AND METHODS

The software has been designed to incorporate all of the necessary steps needed for FDM modeling from raw imaging data to visualization of results. This enables researchers and students alike to study modeling concepts and perform bioelectric field simulations. Generally FDM modeling workflow can be defined as:

1. Raw imaging device data
2. Filtering and segmentation
3. Model parameters and source definition
4. Simulation
5. Visualization

Java was chosen for implementation because of group's expertise in it. Java may introduce a small performance penalty if compared to C++. Recent versions of Java (1.4 and 1.5) have improved especially desktop performance significantly and enabled hardware accelerated user interfaces and 3D graphics. Java was a compelling choice because of its ability to run on multiple platforms without recompiling, rapid application development, clean object oriented design, as well as huge, well organized standard library and a number of quality open source projects. We have selected Java 1.5 as the implementation version. The 3D graphics library utilizes Java3D 1.4.

The software has been written to be highly modular. This allows the modules to be built and tested separately and effectively shields the implementation of different modules from each others. It also enables us to use the same modules with various other applications. Selected quality open source projects have been used to incorporate tested and well documented features. Most prevalent and known of those are Hibernate, Spring, Standard Widget Toolkit (SWT) and Jakarta Commons projects.

Manuscript received July 5, 2006. This work was supported in part by Academy of Finland project funding, project number 5201194.

T. Arola is with the Ragnar Granit Institute, Tampere University of Technology, Tampere, 33101 Finland. Phone: +358 331164003; e-mail: tuukka.arola@tut.fi.

1-4244-0033-3/06/$20.00 ©2006 IEEE.

The software suite consists of three different pieces of software. Each of these programs can be used separately. The first program is called Segmentator. It is used to load imaging device data and convert it into a segmented data. The second program is called Modeler and it features functionality to apply model parameters such as boundary conditions and sources to the models. The model is exported to the third program, Simulation, which runs the solver. Simulation program allows users to perform distributed calculations. The results from a simulation are transported back to Modeler in which the results can be visualized with original imaging device data as well as with segmented model.

The graphical user interfaces for all three programs are written in SWT which allows the software to be used on all Java supported systems as well as a Java Web Start application. A GUI implemented in SWT also looks the same as native graphical user interfaces and are more responsive as Java Swing user interfaces. Generally, SWT is also considered to be easier to code.

Segmentator

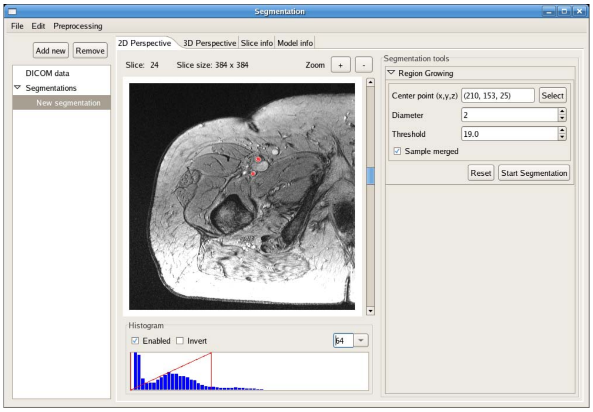
The Segmentator program is used to extract identifiable tissues from the original imaging device data. Various 2D and 3D algorithms have been implemented for segmentation purposes. Currently available algorithms include Level Sets [3], Active Contours [4], Region Growing and Thresholding. Also new techniques based on artificial intelligence and genetic algorithms are being developed. Also preprocessing tools have been incorporated. It is possible to apply various filters and normalize slice intensities within the data set. Segmented targets are collected into a tree structure for easy organization. This allows us to create and modify models with ease. Segmentations are implemented as layers and sublayers that can directly be visualized by the visualization engine. Segmentations can also be used to define a volume where other segmentations are performed and other segmentations can generally be used as seeds for other segmentations. This enables us to perform e.g. region growing segmentation in an already thresholded volume.

Modeler

Modeler is a program that is responsible for defining the model parameters as well as performing post processing operations to the simulated data. User can edit model material parameters such as conductivity values for the segmented tissues, simulation problem boundary values and sources. Sources can be defined as nodes or voxels. The actual difference between these two is dependent on the formulation and the stencil used for the problem. Currently available stencils are the traditional 6-neighbor as well as a star-shaped 18 [7] stencil. These stencils can be applied to both nodes-at-center-of-voxel and nodes-at-vertices-of-voxel formulations. The modeler creates and exports calculation matrices according to the given parameters to the Simulation program.

Simulation

Simulation is a program responsible for actually performing simulations to solve potentials and currents in the model. It implemented as a networked service. This enables us to use a dedicated calculation server to perform large simulations but it can still be run on a local computer for smaller problems. The service itself incorporates a Preconditioned Conjugate Gradient (PCG) solver with various FDM grid stencils and Diagonal- and Incomplete Cholesky (IC) preconditioners. The solver has been optimized for FDM specific calculation and utilizes the symmetric positive definitive nature of FDM matrices. However, it is possible to solve any linear equation in matrix format.



Slice: 24

Slice size: 384 x 384

Center point (x,y,z) (210, 153, 25)

Diameter 2

Threshold 19.0

☑ Sample merged

☑ Enabled ☐ Invert 64

Fig 1 Segmentator program with a Region Growing segmentation of an artery in the human thigh. The program is running on Linux with GTK+ user interface library.

General features

Visualization module [5] is used in Segmentator and Modeler. It renders two and three dimensional representations of raw imaging or model data. Multiple modalities are rendered as layers with adjustable transparencies as seen e.g. in Fig 2. 2D visualizations are drawn using as much hardware acceleration as possible. Visualization module has a support for pseudo-grey color maps making it possible to show black and white images with up to 1792 levels of gray on a standard display. Additionally the engine can render isosurfaces, isocontours, vector and potential fields. 3D visualization is done via hardware accelerated Ray Casting or by triangulating surfaces using Marching Cubes [6]. Also high quality software Ray Casting has been implemented and splatting-based visualization mode is under development. 3D hardware accelerated visualizations are implemented in Java3D 1.4. Java3D is an application programming interface on top of OpenGL or DirectX that provides a consistent interface for hardware accelerated 3D graphics. Java 3D has been implemented for various platforms.

The data loader in Segmentator and Modeler is capable of reading DICOM format imaging device data as well as various other formats used in previous software developed at our institute. Also database connectivity has been implemented and models, as well as model parameters can be loaded from a database. This becomes exceedingly important when using a model with calculated lead fields. An indexed database can efficiently find lead vectors for the specified leads.

II. RESULTS

We have implemented a software suite for FDM models that allows us to perform the entire modeling process as described.

Java has shown its cross-platform benefits and the software works identically in all of the tested system in terms of both usability and performance. Tested systems include Windows XP and Linux (Fedora Core 5) installations running Java Sun 1.5 Virtual Machine and MySQL database server.

Performance and usability has been high on priority during the development process. Graphics performance is comparable to commercial FEM or FDTD software both in 2D and 3D. The engine produces visualizations for hardware accelerated 3D view such as in Fig 4 within seconds. Performance of 3D visualizations is dependent on the graphics card present but generally the user can rotate and zoom the model in real time. The solver also performs surprisingly well but quantitative comparison to a pure C implementation has not been performed.

We segmented one nerve cell and simulated a single dipole source in the produced model. The segmentation produced consists of surrounding media, nerve cell hull and nerve interior. Conductivity values were set to 100:1:100 respectively. Fig 2 shows a 2D visualization of the nerve cell with the dipole. Potentials can be seen spreading into conductive surrounding media.

As en example of 3D visualization, we have visualized a segmented human head obtained by segmenting Visible Human Man data (Fig 3). The human head segmentation is difficult and takes considerably long time due to the massive amount of details in the data. The data size segmented and visualized was 700x700x130 voxels (in total 63.7 million voxels). We have also obtained a CT scan set of a rat. The rat data consisted of 250x457x719 voxels (in total 82.1 million voxels) and it has been obtained with a new 64 headed CT scanner at Tampere University Hospital (Fig 4). The rat data was easy to segment and a model data can be visualized in 3D in a couple of minutes. Visualizations are done in both cases by surfacing produced segmentations with Marching Cubes algorithm.

All of these examples have been performed on a standard desktop 2,8GHz Pentium 4 PC with 2GB of RAM running Fedora Core 5 Linux and Windows XP operating systems.

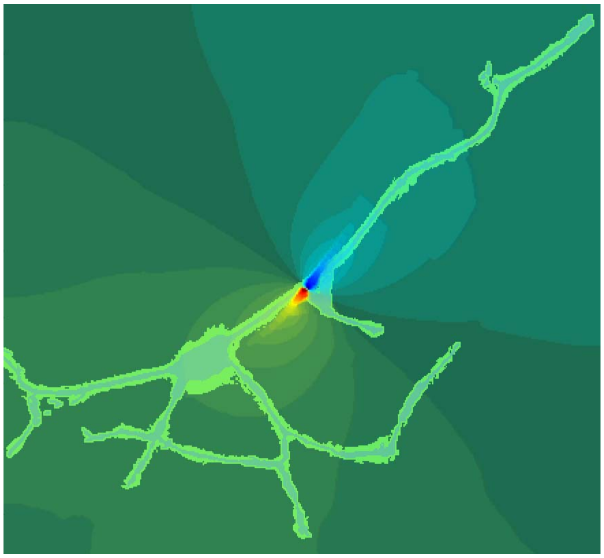


Fig 2. FDM model of a real nerve cell layered with a simulated current source and generated potentials.

Fig 3 shows a 3D visualization of the human head obtained by segmenting Visual Human Man data set. The complete model consists of 8 tissues but here only four types of tissues are visible. Visualization is done by triangulating segmentation surfaces and rendering them with hardware acceleration. Total triangle count for the complete scene is 3.2 million

Fig 3 3D visualization of a segmented Visible Human Man data set with skull, eyes and eye lids visible.

Fig 4 presents a 3D visualization of a rat obtained from a CT data set. Visualization is done by triangulating segmentation surfaces and rendering them with hardware acceleration. Triangle count for the complete scene is 5.2 million.

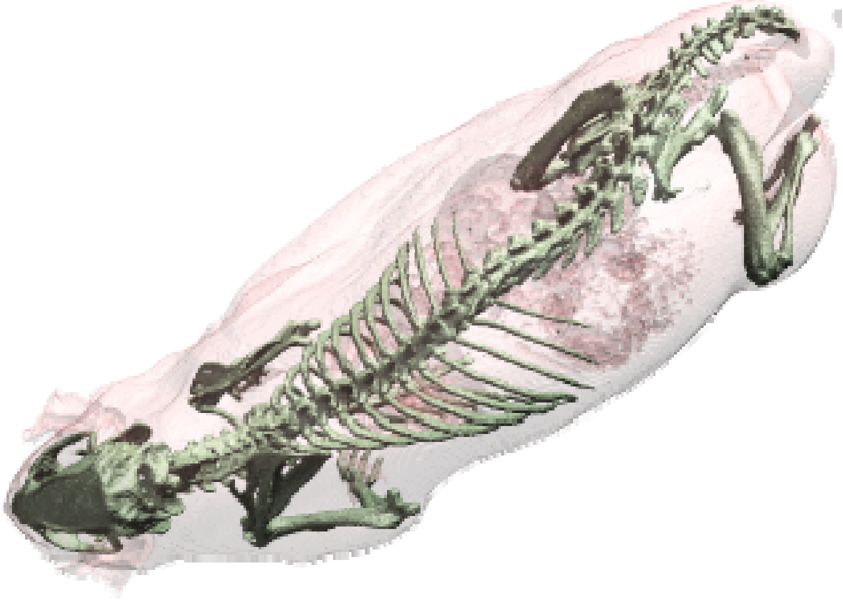


Fig.4. 3D visualization of a segmented rat model obtained from 64 head CT scanner. Visible tissues are the bone and a general soft tissue.

III. DISCUSSION

The software has been under development for a couple of years. It has been implemented with collaboration of Ragnar Granit Institute researchers and Tampere University Hospital. To our knowledge it is the only one of its kind available. The features are especially developed for FDM model extraction, management, simulation and visualization. Commercial software exist for finite element (FEM) and finite difference time domain (FDTD) simulations. If compared to commercial FDTD software our software lacks 3D computer aided design (CAD) tools. There are no plans to develop such features. The software suite is intended to be used to simulate bioelectric phenomena which in our case do not necessitate CAD capabilities since in FDM the model itself acts as a mesh and e.g. no mesh generation is needed.

Our software is being used in various research projects as well as for education purposes at our institute to demonstrate FDM modeling concepts and 3D visualization. It can be applied for image analysis and visualization e.g., segmentation, volumetric and structural data analysis and 3D visualization for which our software has been successfully used in both clinical cases as well as for research purposes.

One could argue whether Java is the right choice for the job at hand. However, we are confident that benefits from development speed and the possibility to run the software in various operating systems exceed the slightly increased memory consumption and small possible performance penalties compared to a C++ implementation.

We have been implementing distributed computing system which enables us to perform scheduled calculations on multiple machines for a single simulation. This lets us to calculate bioelectric simulations with huge models utilizing the power of researchers' desktop computers. In the future we will also further develop the software and incorporate enhanced features such as better 3D visualization and network data compression. Some features such as complete automatic source definition are still missing and will be developed in the future. Currently, the ground work for the platform has been provided and its usability has been evaluated.

The software is not currently available as open source and the final license will be determined later.

IV. CONCLUSIONS

We have introduced our modeling software suite developed for finite difference modeling. The suite is designed to incorporate whole modeling process from raw imaging data extraction and manipulation to simulation and visualization of created models and simulated results such as potentials and currents. The software is written in Java which enables the software to run on various hardware environments and operating systems.

REFERENCES

[1] Hyttinen J. “Development of Regional Aimed ECG Leads Especially for Myocardial Ischemia Diagnosis”. Doctoral thesis. Tampere University of Technology, Tampere, Finland, 1994.

[2] Malmivuo J, Plonsey R. “Bioelectromagnetism – Principles and Applications of Bioelectric and Biomagnetic Fields”. Oxford University Press, NY, USA, 1995.

[3] Osher S. and Sethian J. “Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation”, J. of Computational Physics, vol.79, 1988, pp.12-49.

[4] Kass M., Witkin A. P. and Terzopoulos D. “SNAKES: Active Contour Models”, International Journal of Computer Vision, 1987, pp.321-331.

[5] Arola et al. “A Java-based Multiplatform High Performance 2D and 3D Visualization Engine for Medical Image Data Sets”. EMBEC’05 Proceedings, 2005.

[6] Lorensen W., Cline H: “Marching cubes: A high resolution 3D surface construction algorithm’” Proc. of the 14th annual conf. on Computer graphics and interactive techniques, 1987. p.163-169.

[7] Saleheen H.I., Ng K.T.”New Finite Difference Formulations for General Inhomogeneous Anisotropic Bioelectric Problems” IEEE Trans Biomed. Eng., vol. 44, 1997.