Medicine Meets Virtual Reality 14
J.D. Westwood et al. (Eds.)
IOS Press, 2006

A Software Framework for Surgical Simulation Virtual Environments

Lee A. Belfore II [a,1], Jessica R. Crouch [a] Yuzhong Shen [a] Sylva Girtelschmid [a] and Emre Baydogan [a]
[a] *Old Dominion University*

Abstract. Surgical simulators are an integration of many models, capabilities, and functions. Development of a working simulator requires the flexibility to integrate various software models, to support interoperability, and facilitate performance optimizations. An object oriented framework is devised to support multithreaded integration of simulation, deformation, and interaction. A demonstration application has been implemented in Java, leveraging the features that are built into the language including multithreading, synchronization, and serialization. Future work includes expanding the capabilities of the framework with a broader range of model and interactive capabilities.

Keywords. Surgical simulation, Framework, Java

Introduction

Surgical simulators are interactive systems that model surgical procedures on simulated biological tissues along with other processes relevant to surgical procedures. The simulation process is complicated by several scientific and technological limitations. Scientific limitations include an incomplete understanding of the properties of biological tissues and processes as well as identifying those aspects of a simulated procedure most amenable to training transfer. Technological limitations include the inability to provide real time performance for models at the desired fidelity. In addition, input devices may not provide a satisfying surrogate that is consistent with the manipulations required in a surgical procedure. In order to provide grounding to manage these issues and facilitate research, a flexible software architecture is necessary to support simulator system development. In light of these issues, an object-oriented framework is proposed to support straightforward integration of the various functional components composing a surgical simulator. The work described here is part of a larger project [1]. This paper is organized in five sections including an introduction, a summary of related works, an overview of the software framework, presentation of a demonstration, and finally a summary and future work.

[1] Correspondence to: Lee A. Belfore, Dept. ECE, Old Dominion University. Tel.: 757-683-3746; Fax: 757-683-3220; E-mail: lbelfore@odu.edu.

1. Related Work

Surgical simulation holds the promise of giving medical practitioners the ability to perform some of their training on simulators, raising the expertise of practitioners when they first interact with patients. In Reference [2], a technology independent language CAML is introduced as a foundation for medical simulation systems. Specific to the work here, Spring [3] offers a framework for surgical simulation that includes many models for surgical intervention, a mass-spring model for soft tissue modeling, and distributive collaborative functionality. Finally, the GiPSi system [4] provides a framework with broad capabilities that organ level simulations of physical, functional, and spatial modeling.

2. An Overview of the Software Framework

The software framework is built upon an abstract class structure that holds the relevant parameters and references to support communication and interaction among the respective objects. The abstract classes are then extended to provide a specific functionality. All objects fundamentally share a common link in the form of the basic geometry which each act upon and update. Communicating this information among the objects and synchronizing the updates enables all to receive the most current information in a consistent fashion. Further, having different objects in separate threads limits the interactions among objects to the pure geometry information and provides a pathway to a coarse granularity parallel implementation through the use of threads. Among the geometry linked objects can include the tissue simulation, dynamic texturing, collision detection, remeshing, visual rendering, and haptics rendering. Each of these can function autonomously, updating and responding at its necessary inherent update rate, provided up-to-date geometry is available. Figure 1 provides a generic perspective on the interaction among these threads. The wound debridement simulation companion paper [1] describes an application where this software framework could be used.

3. Demonstration Implementation

A prototype implementation of this framework has been developed in Java using the Java3D API. This platform was selected for convenience as it includes all of the powerful features of Java, i.e. object oriented class structure, multithreading, serialization, and synchronization. Furthermore, the Java platform provides some inherent capabilities that can be leveraged in collaborative applications. Among the features of the implementation, abstract classes are defined to hold information that must be synchronized among threads. Abstract classes define the generic features for each major thread and are extended to provide the functional implementations desired. Furthermore, different functionalities, i.e. model enhancements, model evaluation, can be easily replaced and evaluated without disrupting the rest of the application. Figure 2 shows an example screen capture of the demonstration application.

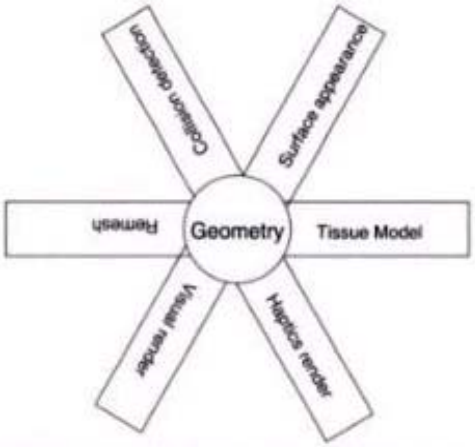

Figure 1. Software Architecture

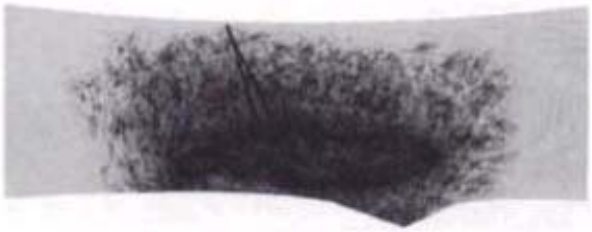

Figure 2. Example Image

4. Summary and Future Work

In this paper, we have presented a software framework that can be used in the development of surgical simulation virtual environments. We plan to continue to develop the framework, to investigate its operation on a C++ platform for enhanced performance, and work towards simpler integration with other libraries and commercial products.

Acknowledgements

This project was a collaborative effort between the Virginia Modeling, Analysis and Simulation Center (VMASC) at Old Dominion University and the Eastern Virginia Medical School. Partial funding was provided by the Naval Health Research Center through NAVAIR Orlando TSD under contract N61339-03-C-0157, and the Office of Naval Research under contract N00014-04-1-0697, entitled "The National Center for Collaboration in Medical Modeling and Simulation". The ideas and opinions presented in this paper represent the views of the authors and do not necessarily represent the views of the Department of Defense.

References

[1] J. Seevinck, M.W. Scerbo, L.A. Belfore, L.J. Weireter, J.R. Crouch, Y. Shen, F.D. McKenzie, H.M. Garcia, S. Girtelschmid, E. Baydogan, and E.A. Schmidt, A simulation-based training system for surgical wound debridement, *Proceedings of the Medicine Meets Virtual Reality 14*, Long Beach, CA, January 24-27, 2006.

[2] S. Cotin, D.W. Shaffer, D. Meglan, M. Ottensmeyer, P. Berry, and S.L. Dawson, CAML: A general framework for the development of medical simulations. In H. Pien (Ed.), *Digitization of the Battlespace V and Battlefield Biomedical Technologies II, Proceedings of SPIE*, Vol. 4037, 2000, pp. 294-300.

[3] K. Montgomery, C. Bruyns, J. Brown, G. Thonier, A. Tellier, J.C. Latombe, Spring: A general framework for collaborative, real-time surgical simulation, *Proceedings of Medicine Meets Virtual Reality 9*, Newport Beach, CA, January 23-26,2001.

[4] M.C. Cavusoglu, T.G. Goktekin, F. Tendick, and S. Sastry, GiPSi: An open source/open architecture software development framework for surgical simulation, *Proceedings of the Medicine Meets Virtual Reality 12*, Newport Beach, CA, January 14-17, 2004, pp. 46-48.