A FRAMEWORK FOR INTEGRATING VIRTUAL SURGERY MODULES

Yingcai Xiao, Sudheer Regatti, Shanthanand Kutuva
Department of Computer Science, University of Akron, Akron, OH 44325, USA

Narender P. Reddy, Xiaosheng Gao, S. I. Hariharan, Saket Kulkarni
College of Engineering, University of Akron, Akron, OH 44325, USA

ABSTRACT

This paper presents the ASVC (Acquisition, Simulation, Visualization and Controller) framework for integrating heterogeneous virtual surgery modules. The framework extends the well-known MVC (Model-View-Controller) system architecture to make virtual surgery systems flexible and efficient. The framework can be used to construct stand-alone or server-client based virtual surgery systems. The former can be used for pre-surgery practice and planning, and the latter is suitable for surgical e-training. A stand-alone version has been implemented at the University of Akron and a server-client based system is under construction.

KEYWORDS

Virtual Surgery, Framework, eTraining, Human Computer Interaction.

1. INTRODUCTION

Innovations in virtual reality and associated technologies have created growing opportunities for applications in every major discipline. Virtual Reality (VR) is a pseudo space of simulated environment which gives the user the feeling of being immersed in the environment [2]. Virtual Reality has applications in areas like entertainment, medicine, engineering and many other disciplines such as rehabilitation [8,9]. Virtual surgical simulation is one of the most intriguing applications in the medical field. Virtual surgical simulation can be used in surgical training. VR surgery involves tissue squeezing, tissue cutting, and manipulation [10,12]. Cutting in VR environment requires a dynamic and interactive control of the cutting location, cutting force, etc on the virtual object (virtual tissue). Song and Reddy [12] have developed a technique of measuring user-exerted forces and employed them for real time control of cutting, which demonstrated the proof of concept for VR surgery. The present study extends their work to provide a more comprehensive analysis and realistic experience. Designing systems for VR surgery is complex and requires integration of various modules for acquisition of user-exerted forces, real time processing of user-exerted forces and using them to accurately compute tissue deformations, and rendering of the deformed virtual tissue. The key issue in VR surgery is that the geometry of the virtual tissue changes constantly, and the deformation should be accurately computed and rendered in real time. The integration and coordination of all the activities present a complex problem. The purpose of the present paper is to report on the development of an integration framework.

2. SYSTEM DESIGN AND IMPLEMENTATION

Our framework is designed to integrate three major modules of a virtual surgery system (Figure 1). (1) **Data Acquisition:** This module deals with the data acquisition from the surgical tool held by the user, including the force applied on the tool, and the angle of the tool movement. This acquired user data is forwarded to the simulation module. (2) **Simulation:** This module creates the simulated data of the surgery, based on the acquired user input from the data acquisition module. The data generated by the simulation module includes the location of the nodal points, cell topology (which could change during the virtual surgery process), and stresses at each node. The simulation module generates all this data for each time step. (3) **Visualization:** This module deals with the rendering of the time-dependent simulated data. Surgical tools are also rendered for a more realistic animation of the surgery process. The location and orientation of the surgical tools are controlled by the user data acquired from the data acquisition module. (4) In order to effectively communicating between different modules, a **controller** module is added to the framework. It provides standard application programming interfaces (APIs) and network connections for all other modules to transmit data to each other. This ASVC (Acquisition-Simulation-Visualization-Controller) framework can be thought of as an extension to the well-known MVC (Model-View-Controller) [3] software architecture. The equivalents are: Simulation to Model, Visualization to View and Controller to Controller. The equivalent modules perform similar functions as defined in the MVC architecture.

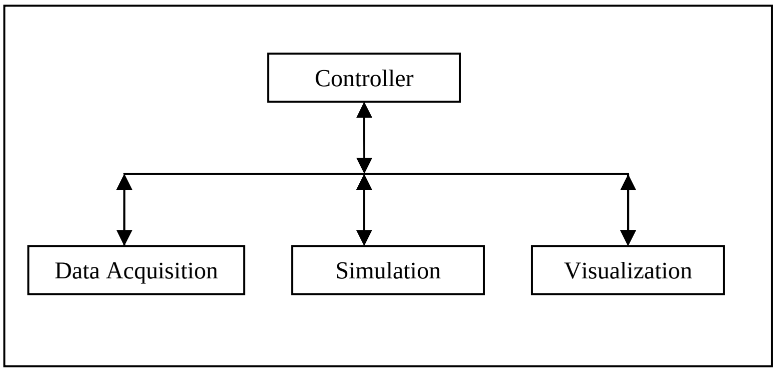

Figure 1. The ASVC Framework.

Using the controller module to bridge the communications between modules works well with the virtual surgery system which could be composed of heterogeneous modules. Modules in the system may be written in different programming languages due to the availability of the required software. Data acquisition module connects to the data input device which often comes with software drivers written in Visual Basic. Simulation is frequently done using finite element method (FEM) [1,6,7,12,15]. FEM software is commonly written in Fortran. Popular visualization libraries, for example, the Visualization Toolkit [11], are written in C++. Special care is needed to transfer data between modules written in different programming languages. For instance, data arrays are column-major in Fortran, but row-major in C++. As a bridge, the controller module can translate data from column-major to row-major when it transfers data from a Fortran based simulation module to a C++ based visualization module.

More importantly, the controller module can set up the software interface standards for all other modules. One of the main advantages of the MVC architecture is that by enforcing such interface standards, other modules can be easily changed. For example, one can replace a FEM based simulation module by a fuzzy logic [4,5,13,14] based simulation module as long as both were written to follow the same interface standard set by the controller.

The proposed ASVC framework can be used to construct two types of virtual surgery systems: the stand-alone system and the server-client system. In a stand-alone system, all modules are integrated into one compact program running on a single computer. Such a system is good for a surgeon to conduct pre-surgery practice and pre-surgery planning. An important issue for this type of systems is to make sure the modules collaborate with each other but don't block each other at run-time. This requires the modules to be threaded such that each module is running as a parallel thread and all the modules in the system can carry out their tasks simultaneously. The motion of the surgical tool is captured at regular time intervals by the data acquisition module while the simulation module simulates the surgical operation as fast as it can using the most current input data from the acquisition module. At the meantime, the visualization module either renders the newly generated simulation data when it's available or waiting for the user to interact with the current display (zooming into a specific surgical region, changing viewing angle, etc.)

In an ASVC server-client virtual surgery system, a server program containing the data acquisition and controller modules is hosted on a server computer while the client software containing the visualization module is running on other networked client computers. The server-client virtual surgery system is ideal for an expert surgeon to demonstrate a surgical procedure to a number of trainee surgeons who may not even reside at the same location. The surgical tool is attached to the server computer for the expert surgeon to operate and the visualization module on the client computers renders the simulated surgical process for the trainee surgeons to view. The main issue of a SAVC server-client virtual surgery system is where to leave the simulation module: the server or the client? If the simulation module is on the server but not on the client, massive simulated data will have to be transmitted over the network, which will drastically delay the viewing at the clients. The problem will multiply as the number of clients increases. We propose to have the simulation module on the client side so that only the user data from the data acquisition model needs to be transmitted. User data is many thousands of times less than the simulated data. Transmitting of only user data makes it possible to visualize a virtual surgery demonstration in real time.

Figure 2 illustrates the architecture of a ASVC server-client system. The simulation module is also included in the server software so that the expert surgeon who demonstrates the surgery procedure could interactively operate the surgical tool. The addition of the Network Communication module on both the server and client sides isolates other modules from the details of the network connection and data transmission. The Global Data module in the controller captures the state of the surgery and stores the latest simulated data including the current geometric configuration of the operating subject. This enables a client to join an ongoing session. The most recent simulated data with the current geometric configuration are fetched over to the client as the correct initial conditions for the client simulation module to get started. After that only new user data (the movement of the surgical tool) needs to be sent to the client to continue the animation.

An ASVC stand-alone system has been implemented at The University of Akron to simulate tissue cutting. Two simulation modules have been constructed. One of them uses the finite element method and was written in Fortran. The other uses fuzzy logic and was written in C++. Experiments have been conducted using the fuzzy logic modules. Initial results showed the system can simulate tissue cutting in real time. Figure 3 is a screenshot of the display from a tissue cutting experiment simulated by the system. Further quantitative experiments are being conducted to measure the accuracy of the system.

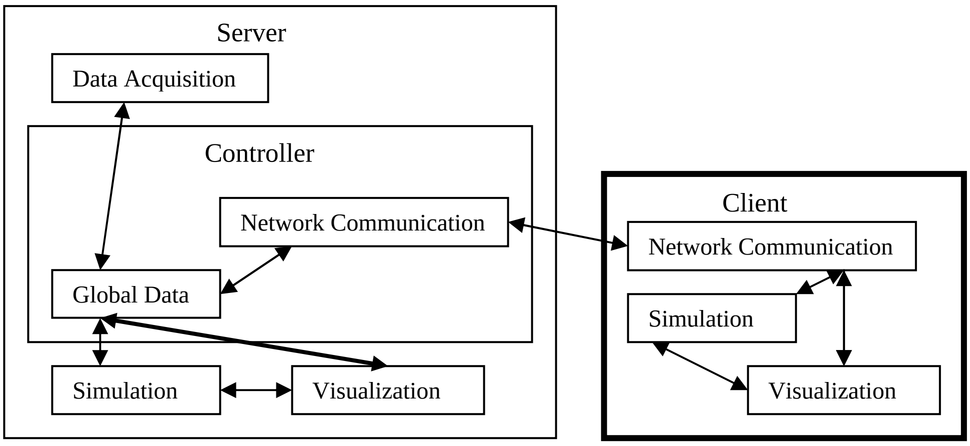

Figure 2. The ASVC Server-Client Architecture.

3. CONCLUSION

A new framework, the ASVC (Acquisition-Simulation-Visualization-Controller) framework, for integrating heterogeneous virtual surgery modules has been presented. The ASVC framework, which extends the well-known MVC (Model-View-Controller) system architecture, makes virtual surgery systems flexible and efficient. A stand-alone implementation of the framework has demonstrated that real time simulation of tissue cutting is possible. Quantitative experiments are being conducted to measure the accuracy of the system. A server-client implementation is under way at the University of Akron.

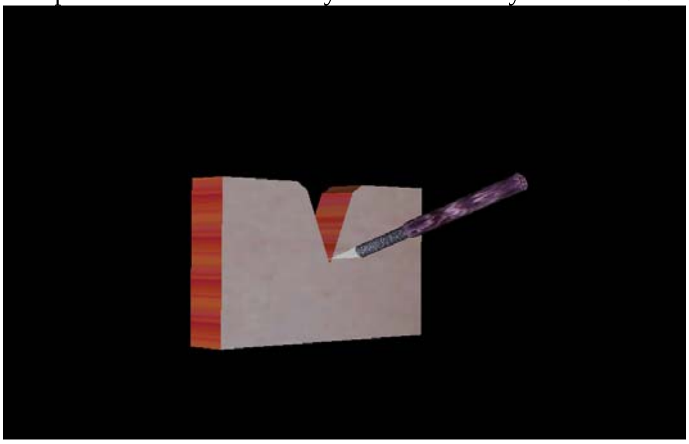

Figure 3. Virtual Tissue Cutting.

REFERENCES

[1] Berkley, J., *Haptic Rendering of Deformable Bodies Using Real-time Finite Element Analysis: an Application to Suturing Simulation,* doctoral thesis, University of Washington, 2002.

[2] Burdea, G. and Coiffett, P., *Virtual Reality Technology*, Wiley, New Jersey, 2003.

[3] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M., *Pattern-Oriented Software Architecture, Vol. 1: A System of Patterns,* Wiley, 1996.

[4] Chien, C, Fuzzy Logic in Control Systems: Fuzzy Logic Controller, *IEEE Trans Syst Man Cybern Part II*, Vol. 20, pp. 429-434, 1990.

[5] Das, A, Reddy, N.P., and Narayanan, J., Hybrid Fuzzy Logic Committee Neural Networks for Recognition of Acceleration Signals Due to Swallowing, *Computer Methods & Programs in Biomedicine*, Vol. 64, pp. 87-99, 2001.

[6] Keeve, E., Girod, S., Pfeifle, P., Girod, B., Anatomy-Based Facial Tissue Modeling Using the Finite Element Method, *Proceedings of IEEE Visualization 1996*, pp. 21-28.

[7] Mor, A. and Kanade, T., Modifying Soft Tissue Models: Progressive Cutting with Minimal New Element Creation, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2000 Lecture Notes in Computer Science*, Vol. 1935, pp. 598-607, 2000.

[8] Reddy, N.P., Biomedical Engineering in the Twenty First Century, *Recent Advances in Biomedical Engineering*, editor: D.C. Reddy, Tata-McGrawHill Publishing Co., New Delhi, India, pp. 183-187, 1994.

[9] Reddy, N.P., Virtual Reality in Rehabilitation, *Innovations in Rehabilitation Engineering*, editor: R. Cooper, IEEE Engineering in Medicine and Biology Conference, 16th Annual International Conference, 1994.

[10] Reddy, N.P. and Sukthankar, S, *Electromyographic Virtual Reality System*, US Patent Serial No. 5482051, Jan. 9, 1996.

[11] Will Schroeder, Ken Martin and Bill Lorensen, *The Visualization Toolkit --- An Object-Oriented Approach to 3D Graphics*, Kitware (www.ktiware.com), 2003.

[12] Song, G.J. and Reddy, N.P., Tissue Cutting in Virtual Environments, *Interactive Technology and the New Paradigm for Healthcare*, Editors: R.M. Satava, K. Morgan, H.B. Seiburg, R. Mattheus, and J.P. Cristensen, IOP Press and Ohmsha, Amsterdam, pp. 359-364, 1995.

[13] Suryanarayanan, S., Reddy, N.P., and Canilang, E.P., A Fuzzy Logic Diagnosis System for Classification of Pharyngeal Dysphagia**,** *International J. Biomedical Computing*, Vol. 38, pp. 207-215, 1995.

[14] Suryanarayanan, S., Reddy, N.P., and Gupta, V., An Intelligent System with EMG-Based Joint Angle Estimation for Telemanipulation, *Medicine Meets Virtual Reality: Healthcare in the Information Age***,** Editors: Weghorst, S.J., Sieburg, H.B. and Morgan, K.S., IOP Press, Amsterdam, pp. 4546-4552, 1996.

[15] Takahashi, E. and Kaneko, T., Facial Surgery Simulation Using 3D Bubble Mesh, *Systems and Computers in Japan*, Vol. 32, No. 1, pp. 1-9, 2001.