The Application of Streaming Video in Web-Based 3D Virtual Laboratory

S. J. Lu, C. C. Ko, B. M. Chen and C. D. Cheng
Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576
[Tel: 65-6874 6863 Fax: 65-6874 7875, Email: engp1591@nus.edu.sg]

Abstract

A web-based 3D virtual experimentation system has been successfully developed for teaching and learning purposes in the National University of Singapore (NUS). Rather than simulating the instrument display on the client as is often done in other virtual laboratories, the real-time video capture of the actual oscilloscope display is seamlessly integrated into the web-based 3D virtual environments. Our video subsystem can reflect the changes of oscilloscope display in real time through the implementation of its video capturing, transmitting, receiving and rendering modules when remote experimenters manipulate the real instruments through virtual instruments interface developed by 3D modeling. The exploitation of live video reduces the complexity of 3D modeling as well as promises a higher visual quality scene. The software and hardware structure of this video system is introduced. The customized API for the implementation of such subsystem is also proposed.

Keywords: Web-based Virtual Laboratory; Distant Education; Virtual Environment; Streaming Video; Java3D; JMF

1. Introduction

Spurred by development in computer science and network technology, the use of the Internet has been expanding exponentially. It is now extensively used as a connectivity and reference tool for commercial, personal and educational purposes. In education, the Internet opens a variety of new avenues and methodologies for enhancing the experience of learning as well as expanding educational opportunities for a larger pool of students. Specifically, distance education and non-traditional classrooms have the capability to reach more students using specialized instruction and self-paced learning.

In Electrical and Computer Engineering Department of National University of Singapore some web-based experimentation system have been established and launched for thousands of undergraduates in NUS [1-10]. However, the systems is 2D-based, which can't give the user the immersive feeling. A web-based 3D virtual laboratory system on an oscilloscope experiment has been developed recently. With a graphical user interface designed to be as natural to using real instruments as possible and LabView software package, remote users are able to gain access to actual physical apparatus, set experimental parameters, conduct and monitor the experiment in the laboratory through browsing the relevant web pages. The system enables students control real instruments through Internet and conduct experimentations anytime anywhere. It offers students an immersive feeling of doing the experiments in the real laboratory, enhances the experience of learning, and expands educational opportunities for a larger pool of students.

One of important issue for the web-based experimentation is to choose a proper means to supply the remote student the experimentation results, to let them see the effects of what they have done. One way is to transmit the experimentation data through Internet to client computer and reconstruct the waveform locally. It's a feasible and reliable way, but the visual effect is not very satisfactory. The other way is to transmit the live video captured using the video camera through Internet. The rendering of live video in 3D world is totally different from the traditional way to transmit the live video into 2D world because users need to see the video display from various view angles when they navigate in the 3D world. In this paper we construct a video subsystem. The streaming video is seamlessly integrated into the web-based 3D space using dynamic texture technique. The video feedback can reflect the waveform changes of oscilloscope display in real time through the implementation of its capturing, transmitting, receiving and rendering modules when students manipulate the real instruments remotely through web-based virtual instruments interface.

The rest of the paper is organized as follows. In Section 2, the typical user interface is described to show how VLAB helps students gain the same experience as with viewing true experiment instruments in the real laboratory. Section 3 describes the hardware and software architecture of the video system, and Section 4 proposed the customized API for the implementation of streaming video and development of remote virtual laboratory. Finally, conclusions are drawn in Section 5.

2. Interface

Figure.1 is the set-up of oscilloscope experiment in real laboratory. The oscilloscope and signal generator are connected to HTTP & Video server through the GPIB (General Purpose Interface Bus), which is an exchange medium between the instrument signals and user control data. The general CCD video camera is connected to the monitor adaptor of the server and it is responsible for capturing the oscilloscope display in real time. The communication of control signals and feedback data between the remote users and the programmable instruments is implemented using LabView, a software package produced by National Instruments. VLAB is embedded in web pages as the applets.

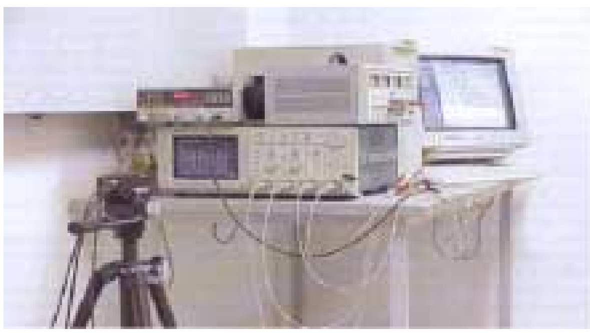

Figure 1 Laboratory Set-up of Oscilloscope Experiment

A remote laboratory for an oscilloscope experiment should have a realistic and user-friendly interface through which users can control the real instruments. One important consideration is that if the user sees a non-realistic interface, a poor impression will be created and the experiment is not easy to be conducted further. The graphic interface of VLAB on the client side is implemented by HTML, Java Media Framework and Java 3D. Fig 2a and Fig 2b shows the front panels of the oscilloscope and signal generator, the two main instruments in the laboratory. The experiment is set up by “dragging” various leads and cables to connect relevant points of the circuit to the two instruments. The live video is introduced into the virtual experimentation environment. Unlike the playback of streaming video in 2D world, the video in 3D virtual environment can be viewed from different view angle. The live video capture of the actual oscilloscope display is seamlessly integrated into the web-based 3D virtual space through the combination of JMF (Java API) and the dynamic texture technique. The video feedback can reflect the waveform changes of oscilloscope display in real time once students change the setting and parameters of the real instruments through web-based virtual instruments interface.

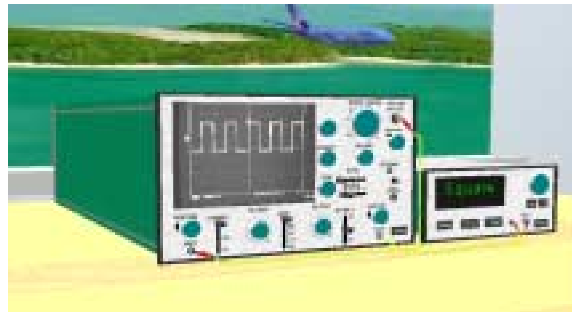

Figure 2 One View of User Interface of VLAB

3. System Architecture

3.1 Software Structure

Figure 3 summarizes the software structure of the system. A HTTP & Video server running with Win2000 hosts the web pages for browsing and the. Virtual experimentation environments, virtual experimentation instruments and video receiving and rendering modules are all embedded in HTML files as java applets. They are downloaded for running on the client machine to provide web-based 3D virtual user interface. Video capturing and transmitting modules both run on video server as daemon. To monitor and control the real lab instruments, LabView is installed on the HTTP server to interchange the signals between the real programming instruments located in laboratory and web-based virtual instruments built by 3D modeling. A Java Plugin-enabled browser such as Internet Explore or NetScape is necessary for the client computer. Several main parts of software system structure are presented as below:

- LabVIEW has been chosen to implement local instrument control. It is a powerful instrumentation and analysis programming environment for PCs running Microsoft Windows series and various other operating system. In our implementation, the WWW server accepts the parameters from the client and then passes these parameters to LabVIEW. The LabVIEW program invokes a local instrument control program that commands the attached instruments via the GPIB interface. Based on the GPIB port number assigned to the instrument, the commands for modifying settings or initiating specific actions are transferred through the GPIB card and cable to the corresponding instrument, which interprets the commands and takes appropriate actions.

- The video subsystem is implemented using Java Media Framework (JMF) API, which is based on the Real-Time Transport Protocol (RTP). It is an application programming interface for incorporating time-based media into Java applications and applets. In our implementation, the video subsystem consists of four modules that are responsible for the video capturing, transmitting, receiving and rendering functions respectively. Capturing and transmitting module run as the daemons on video server, while the receiving and rendering module are embedded as applets in HTML files stored in HTTP server, which are downloaded to client computer for requesting the live video from the HTTP server. The installation of JMF API in both the video server and client PC is necessary of the running of video subsystem.

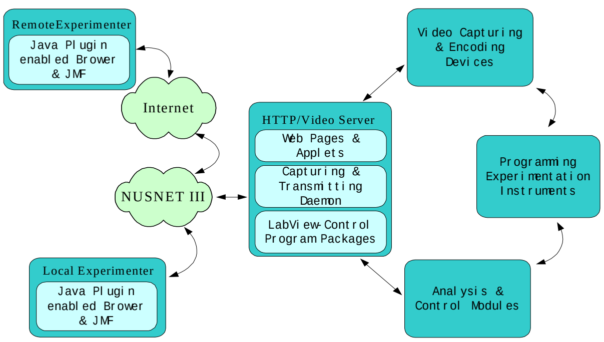

Figure 3 Software Structure of System

- Web-based 3D virtual laboratory are implemented using Java 3D API 1.3. Java 3D is an application programming interface used for writing 3D graphic application and applets. It provides a full support for the animation and behavior. In our implementation, the virtual environment and experimentation instruments are all modeled using Java 3D. The user actions such as picking, dragging are all realized using this API. All these 3D model and user action classes are embedded as applets in HTML files stored in HTTP.
- In client side, the Java Plug-in enabled browser such as Microsoft Explorer and Netscape is necessary to view the virtual user interface. Java 3D and JMF are also needed to be installed in the client computer.

3.2 Hardware Structure

Figure 4 gives a block diagram of the main hardware structure and components in the system, which can be divided into the following five subsystems.

- HTTP & Video server hosts the whole web-based 3D virtual laboratory (applets), including the video subsystem. LabView control software packages are also running on this server. The server is equipped with video capture card, which is connected to the video camera. With the Ethernet card, the server is connected to the Internet through the NUSNET-III-network on the campus of National University of Singapore (NUS). It accepts the connection requests and sends the live video of oscilloscope display to the remote users. Additionally, the HTTP server receives command strings from the client computer through a TCP/IP channel for instrument controls and analog input/output.

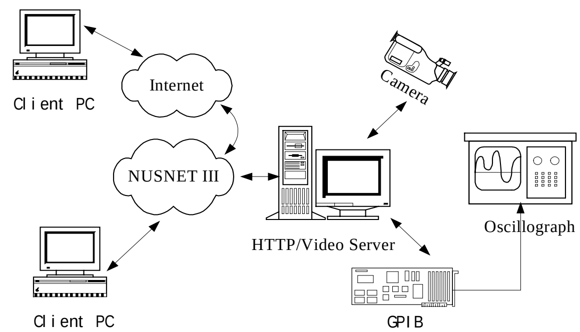

Figure 4 Hardware Structure of System

- Video Camera is a common CCD digital camera. It captures the waveform changes of oscilloscope display and encodes the captured media data. It's connected to the video capture card of video server.
- The programmable experimentation instruments such as oscilloscope and signal generator are connected to the HTTP & Video server through a GPIB card. These GPIB-based instruments are ready to accept and execute SCPI (Standard Commands for Programmable Instruments) commands defined in IEEE488.2. Any data resulted from the execution of the command will also be sent back to the server through the GPIB.

4. Customized API

In this paper, a customized API library for the capturing, transmitting, receiving and rendering of streaming video in the virtual 3D environments as well as the construction of 3D virtual experimentation environments, virtual experimentation instruments is proposed. Using customized video API, video capturing, transmitting, receiving and rendering can be easily implemented in the development of similar web-based 3D virtual laboratory. The customized 3D components API also facilitate the development of virtual experimentation environments and virtual experiment instruments. Through these customized API, later developers can establish their own web-based experiment systems without needs of expertise of 3D modeling and streaming video manipulation in virtual spaces. After imported, all the classes can be objectified, inherited or overridden catering to the particular requirements of users. It reduces the repeated laboring and speed up the later system development in relevant fields.

The customized classes are created for the manipulation of streaming video in web-based remote manipulations and the construction of virtual experimentation instruments, virtual experimentation environments. Some representative class examples are listed below:

- public class VlabJMFAPI implements ControllerListener, Runnable {}
- public class VTrans implements Runnable{}
- public class Button3D{}
- public class Mouse_Click{}

Class VlabJMFAPI performs the receiving and rendering function of video subsystem. Specified the 3D object that live video will map to, it processed the video data and renders it onto the surface of 3D objects as the object's dynamic texture, which forms parts of object's appearance attributes. Class VTrans implements the video capturing and transmitting work. It can capture and transmits the live video or the media data stored in video server, depending on the specific media locators the program chooses. Class Button3D and class Wall3D are used to create the virtual instruments and virtual environments, while class Mouse_Click is implemented to responds the user's behaviors. After imported, all these classes can be objectified, inherited or overridden according to the users' requirements. The customized API is based on the JMF, but it extends the JMF. The standardization speeds up the later development. VLAB, a web-based 3D virtual laboratory on an oscilloscope experiment is roughly introduced in this paper.

5. Conclusion

The playback of streaming video in web-based 3D virtual environments is introduced in this paper. Rather than simulating the instrument display on the client as is often done in other virtual laboratories, the real-time video capture of the actual oscilloscope display is seamlessly integrated into the web-based 3D virtual space. The customized API for the manipulation of streaming video in web-based remote manipulations and the construction of web-based virtual laboratory are also proposed. Because the downloading of applets, transmission and rendering of live video demand much time, the video subsystem is fairly slow and can only be used on campus at present. Next stage, we will try to study the adaptive frame rate techniques to improve the video system speed.

References

[1]. C. C. Ko, B. M. Chen, S. Y. V. Ramakrishnan, C. D. Cheng, Y. Zhuang and J. Chen, A web-based virtual laboratory on a frequency modulation experiment, IEEE Transactions on Systems, Man, and Cybernetics, in press.

[2]. C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang and K. C. Tan, Development of a web-based laboratory for control experiments on a coupled tank apparatus. IEEE Transactions on Education, VOL. 44. No. 1. p.76-86 2001

[3]. C. C. Ko, B. M. Chen, S. H. Chen, V. Ramarkrishman, R. Chen, S. Y. Hu and Y. Zhuang. A large scale web-based virtual oscilloscope laboratory experiment. IEE Engineering Science and Education Journal, Vol. 9. No. 2 pp. 69-76, 2000.

[4]. C. C. Ko, B. M. Chen, K. P. Chan, C. D. Cheng, G. W. Zeng and J. Zhang. A Webcase Virtual Laboratory on a Frequency Modulation Experiment, in the Proceedings of 40th IEEE Conference on Decision and Control, Orlando, FL, 2001

[5]. J. Zhang, J. Chen, C. C. Ko, B. M. Chen and S. S. Ge. A Web-Based Laboratory on Control of a Two-Degree-of-Freedom Helocopter. IEEE Conference on Decesion and Control, Orlando, FL, 2001

[6]. V. Ramakrishnan, Y. Zhuang, S. Y. Hu, J. P. Chen, C. C. Ko, B. M. Chen and K. C. Tan. Development of a web-based control experiment for a copled tank apparatus, Proceedings of the 2000 American Control Conference, Chicago, Illinois, USA, pp. 4409-4413, 2000

[7]. S. H. Chen, R. Chen, V. Ramakrishnan, S. Y. Hu, Y. Zhuang, C. C. Ko and B. M. Chen. Development of remote laboratory experimentation through Internet. Proceedings of the 1999 IEEE Hong Kong Symposium on Robotics and Control Hong Kong, p. 756-760. 1999

[8]. Arno Schodl, Richard Szeliski, David H.Salesin, Irfan Essa. Video Texture SIGGRAPH 2000.

[9]. Java3D API 2.3.1 Sun Microsystem http://java.sun.com/products/java-media/3D/index.html

[10]. Java Media Framework API Sun Microsystem http://java.sun.com/products/java-media/jmf/index.html