HAVE 2005 – IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa, Ontario, Canada, 1-2 October 2005

A Cataract Tele-Surgery Training Application in a Hapto-Visual Collaborative Environment Running over the CANARIE Photonic Network

Naim R. El-Far, Saeid Nourian, Jilin Zhou, Abdelwahab Hamam, Xiaojun Shen, and Nicolas D. Georganas
Distributed & Collaborative Virtual Environments Research Laboratory
School of Information Technology and Engineering
University of Ottawa, Ottawa, ON, Canada
{naim, saeid, jzhou, ahamam, shen, georganas}@discover.uottawa.ca

Abstract *- In this paper, we will discuss our efforts in developing a hapto-visual application that medical students can use to gain valuable experience in performing eye cataract surgery on a human. Our application will support three scenarios: an instructor and a trainee in geographically distant locations interacting in real-time having the instructor tele-mentor the trainee using haptic devices at both ends; a trainee learning the surgical procedure by means of perceptual cues (e.g. projective lighting); and finally, a trainee performing the surgery without any guidance. Collaboration and remote interaction is done over the CANARIE CA*net4 photonic network.*

Organizationally, project development responsibilities are split into four functional components: the GUI, the graphic rendering, the haptic rendering (which includes collision detection and collision response by means of a physics engine), and networking. In this paper we will discuss our work on the four components, comment on the challenges we are facing and expect to face, and also discuss remaining work.

Keywords *– Tele-surgery, haptic, hapto-visual, virtual environments, CANARIE, CA*net4.*

I. INTRODUCTION

The utility of a training hapto-visual cataract surgery application is undeniable; medical students can begin surgical training sooner, in larger numbers, and with less risk to patients. Furthermore, the application scenarios are replicable on demand, meaning that students can practice at any time, avoiding the preparation time needed to setup an elaborate training operations room (whether training is carried out on human eyes for the more advanced students, or on animals eyes – a method widely used in preparing students to take the next step and operate on a human eye – for the less experienced student).

As mentioned above, we have broken down the application into four components: the GUI, the graphic rendering, the haptic rendering, which includes collision detection and collision response, and networking.

The GUI not only allows the user to change the tools in hand and view the virtual scene, but it also contains a videoconferencing component that allows real-time visual and auditory communication between the instructor and trainee in the tele-mentoring mode.

As for the graphic and haptic rendering as well as the base networking layer, we are creating a customized framework for tele-haptic surgery. Of interest is that we are integrating some and extending some open source haptic (and hapto-visual) APIs to best suit our application scenario. A high-fidelity, collaborative haptic API is necessary to ensure realism. Currently available haptic application frameworks can ensure high enough fidelity but no real support for distributed collaboration (e.g., Reachin API). There are no haptic APIs currently available that offer both, the main lacking area being support for real-time, distributed collaboration. In this paper, we propose our own haptic framework that addresses both requirements keeping in mind the graphical demands of tele-surgery.

Similar is the case in the sub-area of collision detection. While there has been extensive work in the field especially for stand-alone haptic applications, more development is necessary to customize a collision detection algorithm for collaborative environments and high-fidelity, high-resolution haptics (since in operating on a human eye, errors in fractions of a centimeter can prove costly and irreversibly damaging). The collision response module of the application is an extension of an already existing physics engine called xPHEVE developed at the DISCOVER lab [1]. The aforementioned physics engine is being modified and advanced again to better fit the applications' two main parameters: high-fidelity haptics, and collaborative environments. Finally, and at the base of all these functional layers, sits a networking layer that is challenged with utilizing the CANARIE network in its fullest capacity to ensure that all remote functional components run smoothly and transparently to the users.

II. THE GUI

The GUI was programmed using VB .net. Following is a screenshot and an explanation of the various components:

As shown in Figure 1 below, the vast majority of display space is dedicated to the eye and tools. The eye image shown in the figure above is a placeholder; the actual application will have a 3D model in place of the image. Graphics modeling is discussed in the next section of this paper. The components shown on the bottom of the screenshot in Figure 1 allow for the user to choose which tool goes into which hand, interact through teleconferencing with a mentor, navigate the scene (allowing the model to be seen from angles not achievable in real life; for example from inside the eye out, behind the eye, etc), and also receive visual cues reflecting haptic pressure on the model.

0-7803-9377-5/05/$20.00 ©2005 IEEE

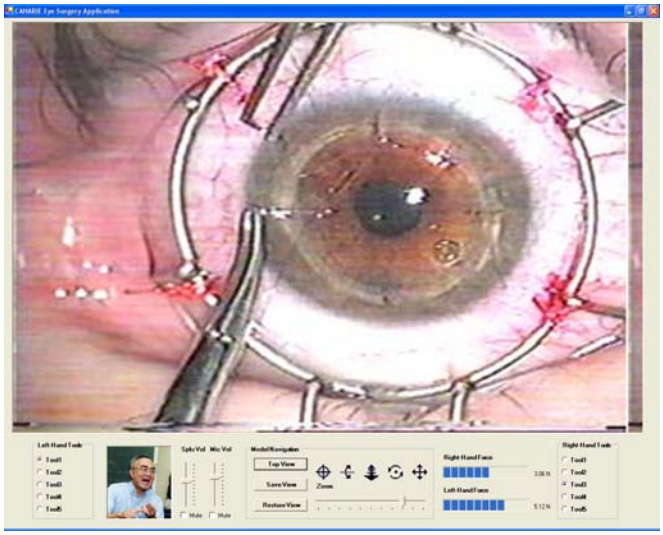

Figure 1: GUI Screenshot

III. GRAPHICS

For the purposes of our application, we need to accurately and optimally (in terms of file size) model the human eye and the surgical instruments.

For the eye, we started with a commercially available model [2] that we bought and modified to better fit our needs (i.e. detail, size, color, etc). The final model after modification is shown in Figure 2. VRML was chosen for the file format because of its hierarchical, object-oriented properties. The VRML format is also widely supported and can be loaded into web browsers which would allow in the future for observers to watch the surgery over the World Wide Web. VRML is also widely supported by newer modeling languages such as X3D. Finally, using VRML allowed for quick deployment of a stand-alone prototype [3] that was tested on the Reachin Display platforms [4] available to us; the Reachin API has a utility (ReachinLoad) that allows haptic properties be added to VRML nodes making for a custom VRML-based format that could be loaded rapidly using the Reachin API running on Reachin Displays.

As for the surgical tools needed for the application, they were modeled using 3D Studio Max. The references used for designing the tools' shapes and textures were cataract surgery websites as well as specifications provided by the University of Ottawa Eye Institute.

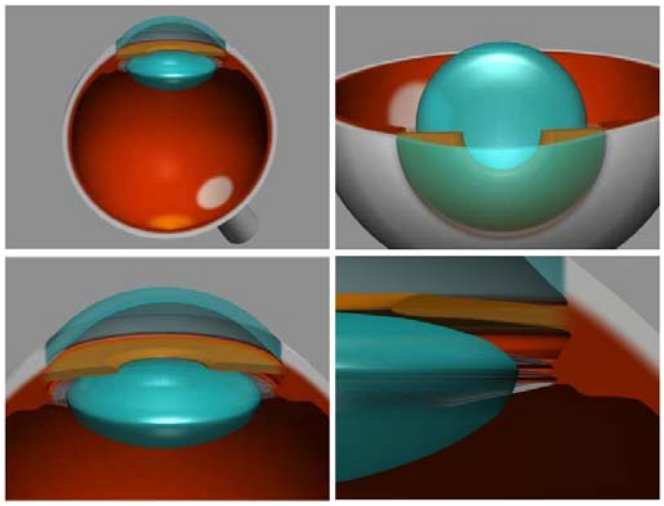

Figure 2: Eye Model with 16,000 Polygons

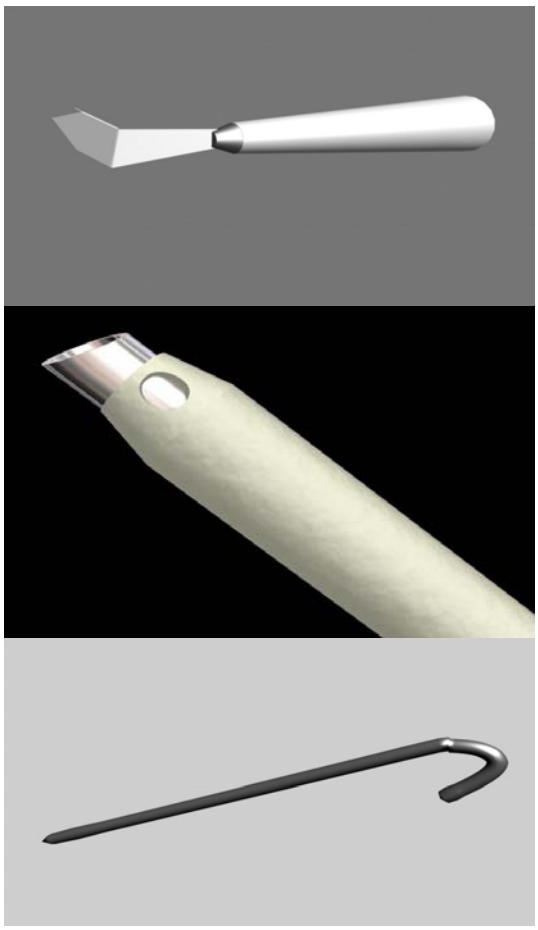

Figure 3: Sample models of the surgical tools developed. From top to bottom: a keratome, a phacoemulsification tool, and a hook.

Having used 3D Studio Max allows the tool models to be converted into different formats in the future if necessary or so desired. One of the challenges faced during the modeling of the tools was balancing model quality and the model's file size. We chose to create moderate quality models with

optimized file sizes paying special attention to details at points where the tool model would interact with the other objects in the scene (e.g. the eyelids, the lens, etc), which allows for more realistic graphic interaction at points of contact. Figure 3 shows a few tools that we have modeled.

IV. HAPTICS

Three different available haptic simulation frameworks, namely: OpenHaptics from Sensable Inc. [5], e-Touch from Novint Inc. [6], and CHAI3D from Stanford University [7], were studied.

The OpenHaptics toolkit is a high level API, and is not open source. It implements simple collision detection and response algorithms and hence is not suitable for a sophisticated eye surgery simulation. Furthermore, it only supports Phantom haptic devices.

e-Touch is an open source framework for haptics control. It offers a haptic graphic user interface and is more of an SDK than an API. The fact that e-Touch is open source is very helpful for our purposes.

CHAI3D is a simplified haptic framework. It generalizes the variances in a typical haptic application and provides correspondent base classes. Typical variances include the device, graphic model format, virtual object, collection detection algorithms, and force rendering algorithms.

We are planning to take the advantages of existing APIs and propose our own framework. This framework will include the following components:

- Graphics loader: Modeling 3D objects directly with graphic APIs such as OpenGL is difficult and inefficient. The graphics provides the functionalities of loading 3D objects that are modeled in different formats. We start our work with VRML.
- Physics engine: Discussed in more detail later, a physics engine should be open and extensible. For instance, different collision detection algorithms should be easily plugged into the application.
- State synchronization: A state synchronization module is used to coordinate graphic rendering, haptic control, and a global dynamic simulation thread. Synchronization of these three operations is critical for consistent and concurrent visual and haptic feedback.
- Device manager: different haptic devices will be supported in the simulation application. Haptic devices typically vary in the degrees of freedom, input and output resolutions, and force output.

V. NETWORKING

Participants in our application could be either passive observers, who are not equipped with haptic devices, or they could be active participants who are capable of haptically intervening in the scenario. Figure 4 below shows the architecture we are using that would allow for all class participants to be supported. This hapto-visual environment (HAVE) architecture is based on our previous work in [8].

Three different network connections are involved in this architecture. A local Haptic Real Time Controller (HRTC) is linked with other remote HRTC(s) through a haptics channel (1 in Figure 4) and communicates with its virtual environment (VE) station through a local channel (2 in Figure 4). The VE graphics stations are interconnected over collaborative virtual environment (CVE) middleware. For our CVE middleware, we choose HLA/RTI [9], the IEEE standard for distributed simulations and modeling (3 in Figure 4).

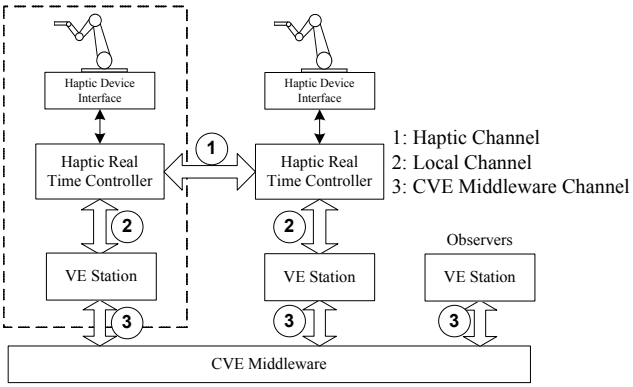

Figure 4: HAVE architecture.

Phantom Desktop haptic device from Sensable Technologies Inc. [10] and Freedom 6S hand controller from MPB Technologies Inc. [11] will be used in the development. Phantom devices provide 6 input degrees of freedom (DOF) and 3 output DOFs, while the Freedom 6S supports 6 DOFs for both input and output (the translational x, y, and z, as well as roll, pitch, and yaw).

The HRTC relies on a hard real time operating system to guarantee the stability of the control loop and to minimize the delays in the network stack. The HRTC is also capable of compensating for network latency, so increased or unreliable latency in the network stack provided by the host operating system will either increase the complexity of the latency compensation algorithms or decrease the effective separation of tele-haptic collaborators. A challenge of ours is to find the right balance given the characteristics of the host machines as well as the network.

VI. SPECIFIC CONSIDERATIONS

A. Collision Detection

Two major tasks in haptic interaction paradigms are collision detection and collision response. The latter is discussed below in the sections on our work on a physics engine, cutting, and geometry deformation.

Collision detection, the focus of this section, in the context of our application is simply the detection of point contact between the stylus representation (e.g. surgical tool, haptic

cursor, etc) and any haptic surfaces in the virtual environment. Collision detection is a complex problem and has been a hot research topic since the late 90s. There have been many proposed algorithms for efficient detection of collisions between 3D geometries. Some are more efficient than others but imply some limitations on the shape of geometries. Our aim in this application is to develop a hybrid collision detection algorithm mainly based on the work in [12].

B. Physics Engine

To recreate a hapto-visual environment that simulates real life, laws of physics need to be implemented in the virtual world. To fulfill the need of physical laws, a physics engine named xPHEVE [1], previously developed at the DISCOVER lab, is being modified and built upon. xPHEVE simply is a library that contains implementations of physical laws. For performance issues associated with xPHEVE having been originally written in Java and Java3D, xPHEVE is being rewritten in C/C++ and OpenSceneGraph [13] for graphics rendering.

C. Cutting and Geometry Deformation

In eye-surgery, cutting and deformation occur on thin surfaces as opposed to other surgery simulations where larger volumetric organs are involved. In eye surgery simulations, a volumetric approach to haptic modeling is unnecessary. Instead, we have constructed test models that are composed of triangular meshes. A number of test cases were evaluated in an attempt to cut the geometry by deleting or separating the triangles that are in contact with a cutting edge [3]. As for our approach to deformation, our work is currently focused on using springs and manipulating elastic properties of haptic surfaces. Development in these areas is only at its beginning phases.

VII. REMAINING WORK

This project is slated to finish between the second and third quarters of 2006, and a lot of work remains to be done. Unless otherwise noted, all aspects discussed in this paper have seen substantial progress. Major challenges lying ahead not touched upon in previous sections are in developing an end-to-end teleconferencing system to work with this training application; developing specific network hardware and software infrastructure based on CANARIE's CA*net4; and finally integrating all components in one working application.

VIII. CONCLUSION

In this paper we have presented our work in developing a hapto-visual eye cataract surgery training application, the benefits of which are clear and highly advantageous over current training methods such as surgery on animal and human eyes. In developing such an application, all components need to be geared towards high performance; graphics need to be realistic yet small enough to load rapidly, haptics need to be of very high resolution and fidelity, and collaboration needs to be over a high speed network running an architecture that takes into account network delay and latency. In developing such an application, we address all these problems and challenges.

REFERENCES

- [1] S. Nourian, "xPheve An Extensible Physics Engine for Virtual Environments". MSc Thesis. 2003.
- [2] Human Eye Model [Search Result], Exchange3D. Accessed on 8/9/2005. www.exchange3d.com
- [3] Y. Xu, "Making Incisions on an Eye Model in Reachin Display". 2/9/2005. Internal report. DISCOVER Lab at the University of Ottawa, Ottawa, ON, Canada.
- [4] Reachin Display. Reachin Inc. Accessed on 8/9/2005. http://www.reachin.se/products/reachindisplay/
- [5] Open Haptics Toolkit. Sensable Technologies. Accessed on 8/9/2005. http://www.sensable.com/products/phantom_ghost/OpenHapticsToolkit-intro.asp
- [6] e-Touch. Novint Technologies. Accessed on 8/9/2005. http://www.novint.com/etouch2.htm
- [7] Computer Haptic and Active Interfaces. CHAI3D. Accessed on 8/9/2005. http://www.chai3d.org/index.html
- [8] X. Shen, F. Bogsanyi, G. Ni, N.D. Georganas, "A Heterogeneous Scalable Architecture of Collaborative Haptics Environments", Proc. IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications (HAVE 2003), Ottawa, ON, Canada, September 2003.
- [9] HLA/RTI. Defense Modeling and Simulation Office. Accessed of 8/9/2005. https://www.dmso.mil/public/transition/hla/
- [10] Phantom Desktop. Sensable Technologies. Accessed on 8/9/2005. http://www.sensable.com/products/phantom_ghost/phantom-desktop.asp
- [11] MPB Freedom 6S. MPB Technologies. Accessed on 8/9/2005. http://www.mpb-technologies.ca/mpbt/haptics/hand_controllers/freedom/freedom.html
- [12] P. Liu, X. Shen, N.D. Georganas, G. Roth, "Multi-resolution Modeling and Locally Refined Collision Detection for Haptic Interaction", Proc. 3DIM 2005: The Fifth International Conference on 3-D Digital Imaging and Modeling, Ottawa, ON, Canada, June 2005.
- [13] OpenSceneGraph. OpenSceneGraph.org. Accessed on 8/9/2005. http://www.openscenegraph.org/