uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Abdelwahab Hamam
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A 3D Haptic Cararact Tele-Surgery Simulation
TITRE DE LA THÈSE / TITLE OF THESIS

N. Georganas
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

R. Liu

S. Shirmohammadi

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

A 3D Haptic Cataract Tele-Surgery Simulation

by

Abdelwahab Hamam

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements for the degree
Master of Applied Science in
Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-34072-1
Our file Notre référence
ISBN: 978-0-494-34072-1

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Medical simulations are becoming more advanced and more reliable in training medical students and new surgeons. Especially in the context of delicate surgeries, such as cataract surgery, the medical community is shifting from traditional medical training to sophisticated virtual reality simulations. In this thesis, we present a simulation built in a three-dimensional environment that aids in training medical residents to perform cataract surgeries. The application utilizes haptic devices as a means of interaction, which enriches the application by providing a sense of touch and force-feedback to users. In addition, we discuss our modeling of the environment and the decisions undertaken in the design. We also compare this 3D application with similar applications in terms of functionality but with different user interfaces, and mention what was the eye surgeon's reaction towards each interface in the training simulation. Finally, we conclude with future modifications of the application and describe a future direction.

Acknowledgments

Upon completion of this thesis, I would like to thank Dr. Nicolas D. Georganas for his continuing support and encouragement. His supervision and advice paved the road for me and directed me throughout my master's pursuit. I would also like to thank Dr. Abdulmotaleb El Saddik, Dr. Xiaojun Shen and Mr. Francois Malric for their assistance and counsel during my graduate studies. I would like to express gratitude to all my colleagues at the DISCOVER Lab, University of Ottawa for their continuous encouragement and for their warm wishes for me to succeed in my academic endeavors. Last but not least, I would like to thank my mother and my father, my sister and her husband, as well as my brother for their love and support.

This accomplishment would not have been possible without the support of God, who has given me the strength and perseverance to overcome the challenges of this thesis.

Contents

List of Figures

List of Tables

List of Acronyms

The following table lists the definition of acronyms that might have been used in the thesis:

2D	Two Dimensional
3D	Three Dimensional
API	Application Programming Interface
CPU	Central Processing Unit
CRC	Communications Research Centre
DISCOVER	Distributed and Collaborative Virtual Environments Research lab
DIVINE	Desktop-Immersive Virtual and Interactive Networked Environment
DOF	Degrees of Freedom
GUI	Graphical User Interface
HCI	Human-Computer Interaction
IEEE	Institute of Electrical and Electronics Engineering
LAN	Local Area Network
LCD	Liquid Crystal Display
MIT	Massachusetts Institute of Technology
NASA	National Aeronautics and Space Agency
SDK	Software Development Kit
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UI	User Interface
VE	Virtual Environment
VR	Virtual Reality
VRML	Virtual Reality Modeling Language

Table 1 – Acronym definitions

Chapter 1

Introduction

1.1 Motivation

In a modern world where technology is advancing rapidly, it is essential to integrate technology with various fields and sectors. Banking, transportation, and medicine are some examples of the fields that are employing technology to enhance their performance. More specifically, in medicine a lot of improvement is being achieved by using more sophisticated technology. Thus, it is vital to keep the medical sector and the technological sector in parallel, especially in patient care, physician training and hospital infrastructure. The field of medical training is the driving force behind this thesis. Jointly with the University of Ottawa Eye Institute, there was the desire to train eye surgeon residents by simulation to perform cataract surgery without interacting with a human patient for safety reasons. Therefore, it is essential to create another approach for surgical residents to acquire rudimentary skills other than the "see one, do one, teach one" approach. Although visual learning is important, training the eye-hand coordination is vital for surgeons dealing with a delicate organ such as the eye. [1] Moreover, the conventional surgical training method requires tight cooperation, between the experienced surgeon and the unskilled students who are trying to absorb the details of the surgery as it is being performed. This conventional method proves to be a cost-intensive application. It also has a drawback: as the ratio of trainee to trainer increases, the burden on both the experienced surgeon and the trainees increases as well. Even in some instances, such as trauma-related procedures where the surgeon needs to quickly stabilize the patient, teaching becomes secondary to the actual treatment of the patient. [2] An alternate approach, which medical schools are using to allow new surgeons to be more dexterous, is to let them train on dead animals' eyes or elastic rubber eyes. Nonetheless, the advancement of science and technology led this approach to become obsolete, and a more immersive and technological approach is preferred. [1] The intent of our project is to use haptic technology in order to simulate the surgery virtually. The haptic technology is a recent technology that utilizes the sense of touch in humans while navigating a computer virtual world (a more detailed explanation is found in Chapter 3).

The project entitled Hapto-Audio-Visual Environments for Collaborative Tele-Surgery Training over Photonic Networking (also aliased as CANARIE Project, as described in 3.5 CANARIE HAVE project) combines the efforts of two universities and many organizations: The University of Ottawa in collaboration with the University of Alberta, as well as the University of Ottawa Eye institute, BigBangwidth, and CANARIE. In the DISCOVER Lab at the University of Ottawa, our group started developing an application to help capture the technology of haptics in the field of Cataract Surgery Medicine. When I started working on the project and eventually on my thesis, the motivation became clear: to create the most realistic cataract eye surgery simulation using the most appropriate technology available.

1.2 Research Problem

Simulators based on virtual reality can be efficient in medical training as any traditional method used in the past. They reduce the cost and length of training. However, there is the concern of how accurately a simulator can represent real-life situations. The term referred to as simulator fidelity is an important aspect when designing and building a simulator system. Even when enhancements and modifications are made to the simulator to make it more sophisticated and reliable, the issue of proficiency remains. Trainees who become proficient on a simulator may not be proficient in the real-world clinical environment. Hence, there is always the risk of the simulator not properly replicating the clinical environment and the tasks of patient care. As such, instead of providing appropriate training for the users, they might provide negative training in the form of improper behavior.

Taking the previous points into consideration, there are some additional features that need to be incorporated in every surgical simulator:

- Simulation of how human tissue behaves under the effect of external stimuli, such as a knife and forceps in a surgery application
- Availability and real-time interactions with the medical instruments specific to the surgery taking place

Both of these points depend on the accuracy of the physical modeling of the human tissue and medical tools as well as the availability of a force-feedback device. The force-feedback device, known as the haptic device, enables the manipulation of medical tools. It also provides the tactile sensation to the user that replicates the feel of the human tissue. [3]

For this research, the surgery at focus is cataract surgery. The topic at hand is relatively new to our team in the DISCOVER Lab at the University of Ottawa. There has been some research conducted previously in the DISCOVER Lab on the simulation of trachea surgery. However, the scope of the surgery differed vastly from the scope of cataract surgery. The variation of the two surgery implementation scopes is due to various rationales. First of all, trachea surgery simulation was intended as a visual simulation, while cataract surgery is intended as a training application. As such, the restraints on the trachea surgery application realism were less than those on the cataract surgery application. Secondly, cataract surgery has more sophisticated and numerous tools than trachea surgery. Finally, the eye organ is one of the most complex organs in the human body. Just by looking at the screenshot of the anatomy of the eye in Figure 1, a person can get the feel of the complexity of the eye organ.

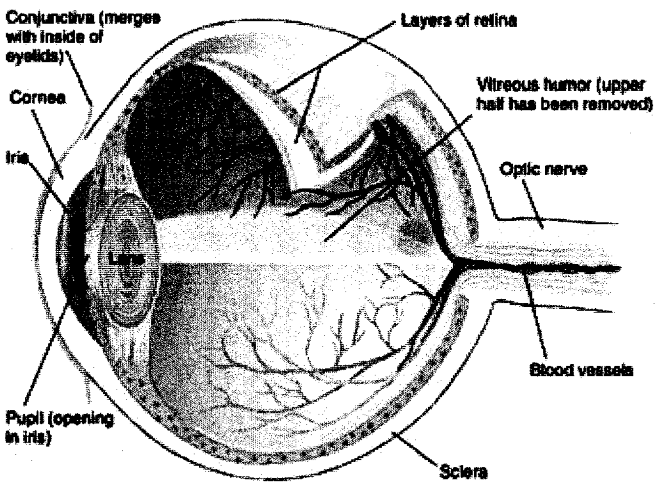


Figure 1 – Side view of the eye anatomy (http://www.bibleprobe.com/eye4.jpg)

This complexity of the human eye makes any eye surgery one that draws attention to a myriad of details. More specifically, there are a lot of details that need to be incorporated into cataract surgery simulation which otherwise can be overlooked. An example of this is how accurate an eye organ model should be, in taking into account the complexity of cataract surgery simulation requirements, but at the same time a model that does not hog the system with extra accuracy and hence extra graphics polygons in memory.

This same point applies to the surgical tools. To what degree of accuracy should the tools be modeled? How should they scale compared to the eye organ? We had to take into account all these questions before planning the application.

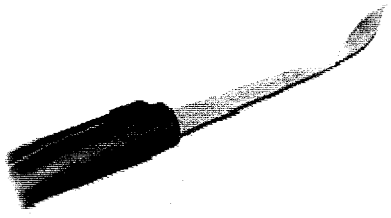


Figure 2 – Sketch of a medical knife (keratome)

In Figure 2, which is a sketch of a medical knife obtained from http://202.71.144.97/aurolab/images/03.gif , it is not clear how big this tool is compared to the eye. In addition, there are certain details that may look trivial but it is necessary to add them in the design of the tools. An illustration of such details in the figure above is the inward bend of the blade. When this tool is modeled, the blade should also be inwardly bent and the degree of the bend should be taken into account.

Another problem we had to consider at the DISCOVER Lab is the interface of the application. The most powerful feature of virtual reality applications is its human-computer interface. The way a person interacts with a virtual reality application significantly influences his or her experience. Since our application is a training program for a delicate surgery, we had to ultimately pick the most appropriate type of interface to use. We had several choices, among them are: 2D Desktop Interface, 3D immersive interface, and completely immersive display. Each interface has its own advantages and is suitable for a specific scenario and environment.

There is a wide variety of Application Programming Interfaces (APIs) that can be used with a virtual simulation. Each API has functionality for loading the graphics and creating the virtual world. It also has utilities for synchronizing the haptic data and the graphics data. The synchronization is needed due to the fact that users should feel what they see.

A final point to consider is the significance of choosing which haptic device to use in the application. Commercial haptic devices vary in the degree of freedom they are allowed to move in, as well as the amount of force-feedback they generate in response to haptic simulation of two colliding objects. There is a specific range, measured in Newton, which a haptic device can operate in, giving users the force-feedback within this range. Moreover, haptic devices tend to differ in shape, size, and the way users grasp and handle the device.

Throughout this thesis, these concepts will be discussed thoroughly and the rationale behind the decisions taken for each application design challenge will be explained.

1.3 Research Contribution

The goal of this work is to design and implement a 3D immersive cataract surgery application. The main cataract application at the DISCOVER Lab is a 2D desktop interface application, and the 3D immersive application is intended to complement it by giving more appealing 3D stereo visual to users who appreciate the 3D environment more. The work achieved is part of a bigger project that aims to contribute to research in the medical/technological simulation field.

In a nutshell, the work leading to this thesis consisted of building a 3D stereo cataract surgery simulation and included the following:

- Modifying a commercial eye model using 3D Studio Max to fit our surgical requirements
- Modeling surgical tools required to perform the surgery
- Implementing the different steps of the surgery
- Performance evaluation and usability testing of the application

Publications in IEEE conferences, resulting from the thesis:

- N.R. El-Far, S. Nourian, J. Zhou, A. Hamam, X. Shen, N.D Georganas, *A Cataract Tele-Surgery Training Application in a Hapo-Visual Collaborative Environment Running over the CANARIE Photonic Network*, IEEE International Workshop on Haptic Audio Visual Environments and their Applications, 2005.
- A. Hamam, S. Nourian, N.R. El-Far, F. Malric, X. Shen, N.D. Georganas, *A Distributed, Collaborative and Haptic-Enabled Eye Cataract Surgery Application with a User Interface on Desktop, Stereo Desktop and Immersive Displays*, IEEE International Workshop on Haptic Audio Visual Environments and their Applications, 2006.

1.4 Thesis Outline

After this introductory chapter, the rest of this thesis is organized as follows: **Chapter 2** gives a general background on the medical aspect of the cataract surgery and its benefits and advancement over time. A general description of the surgery steps is also given. **Chapter 3** discusses the concept of virtual reality and its application in the medical world and other fields along with the different tools and programming packages that is involved in building virtual reality applications. **Chapter 4** outlines the modeling of various medical tools and the virtual eye as well as the design of the environment. **Chapter 5** depicts the design and implementation of the standalone surgery 3D application using Reachin. **Chapter 6** presents the comparison of different implementations of the surgery application along with the evaluation of each application by various groups of users. Finally, **Chapter 7** concludes this thesis indicating a direction for future work.

Chapter 2

Medical Background

2.1 Cataract Surgery

A cataract is the cloudiness of the eye lens. The lens focuses the incoming light rays on the eye retina in order to view the image correctly. It functions similar to a man-made optical lens. Thus a cataract causes blurriness of vision since the light rays are partially blocked and it is harder to focus the rays on one point in the retina.

The primary cause of cataracts is aging. Every person will experience a cataract as they age. Some experience it severely by the age of 60, others will experience it mildly until later in life (90 and above). There are certain factors that will speed up cataracts such as:

- Inflammation inside the eye
- UV and IR radiation exposure
- Nutritional Deficiencies
- Genetics

There are no medications to prevent or treat a cataract. A good diet and protection such as wearing a hat or sunglasses to prevent UV exposure will slow it down. However, every patient must undergo a surgery to remove the defective lens and replace it.

Cataract surgery is the replacement of the natural lens with an artificial one. It is a complicated surgery. According to eye doctors, while it takes only around seven minutes for a skilled surgeon to perform a successful cataract surgery, it takes months for a medical resident to train and acquire the skills needed.

During cataract surgery, unexpected complications might occur, which could lead to extra effort and time. These days, the success rate of cataract surgery is 99%. Unfortunately, the complications cannot be discarded and have to be studied by cataract surgeons. Especially for inexperienced surgeons, minor side effects are inevitable.[4]

2.1.1 Advancement of Cataract Surgery in Today's World

The earliest dated form of cataract surgery goes back to the 5th century B.C. which was recorded in a Sanskrit manuscript. The method they used is known as couching (Figure 3) and it involves displacing the cataract lens within the vicinity of the eye, namely the posterior chamber. Treated patients would see better but since there is no corrective lens replaced their vision would still be blurry.

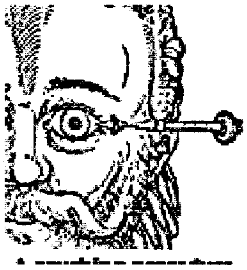


Figure 3 – A couching procedure from "Augendlenst" by G.Gartisch (1535-1606)

In 29 A.D excavations lead to the clue of a technique to breakup the cataract into smaller particles, which will facilitate its absorption. The technique is known as needling. Afterwards some techniques also emerged to further remove the cataract, the most common of which was in 1753 where the cataract was removed through an incision using the thumb. It was then that the concept of cataract surgery has emerged.

There were two major problems at that time however. The first problem is that anesthetics were not yet invented. The surgery was performed with the aid of strong assistants who would hold the patients head still such that it would not move during the surgery procedure (Figure 4). General anesthesia was introduced in 1840 in the form of eye drops and was used for surgical procedures which took care of that problem. Nonetheless, up until the present era it was necessary that the patient be heavily medicated and sedated. This is to assure that the eye does not move during important surgical steps such as incision, in which eye movement can lead to serious injury in the iris. These days the surgery steps as well as the chemicals used have evolved and just local anesthetic is sufficient.

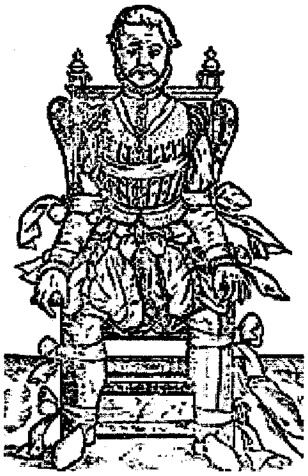


Figure 4 – Man restrained for eye surgery from "Augendlenst" by G.Bartisch (1535 -1606)

The other obstacle after they coined Cataract Surgery in 1753 was the fact that there was no replacement lens available. Since the lens was used to correct the vision, the surgery was not quite complete in our standards today. Actually, there were many advancements of the surgery during the 20th century in which different techniques and chemicals were used to dissolve or freeze the lens and extract it from the eye, but they all lacked a replacement lens. One of those techniques is still used in present day surgery, the technique of emulsifying the lens, in which the lens is broken down using ultrasonic vibrations.

It was not until 1940 that the intraocular lens was introduced in England by Harold Ridley and made the surgery entirely effective as now there is a corrective plastic lens that is permanently placed instead of the crystalline lens. Of course, there was much modification on the lens' design and material since 1940. [5]

2.1.2 Summary of Cataract Surgery Procedures

The steps involved in the surgery are the driving force behind the virtual simulation. In order to implement and design the application, each individual step had to be studied thoroughly. As none of the technical team members were familiar with medical background of the surgery, a series of doctors' interviews were undertaken at the University of Ottawa Eye Institute and at the University of Ottawa DISCOVER Lab. Moreover, videotaped surgery was studied and analyzed to break down the surgery as a whole into discrete, individual steps for better and easier visualization. Lectures of eye anatomy were attended by the team members for more detailed understanding of the complex eye organ.

From the above-mentioned work we were able to summarize cataract surgery into seven consecutive steps starting from step zero where the patient is sedated to step six where the new lens is inserted.

Below is a more detailed list of cataract surgery steps accompanied by the description of the surgical tools required to operate with:

0. Sedation; dilation of the iris; preparation of the instruments; all these steps are preliminary steps and were not included in the simulation.

Among these steps is attaching a tool (Figure 5) to open the eye wide open during the surgery. Although this step was not simulated, the attaching tool was modeled to add more authenticity to the application. The tool to be used is attached to the upper and lower eye-lid and it looks like this:

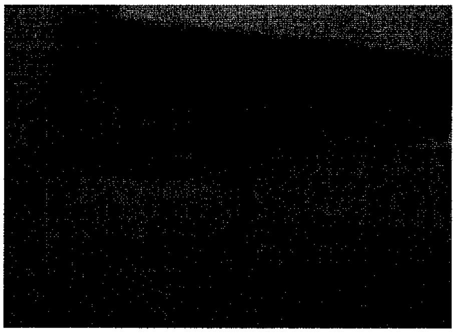


Figure 5 - Tool for opening the eye lid during the surgery

1. Incision: The first actual step of the surgery starts with an incision of the cornea. The tool utilized is the Keratome; a sharp blade with a small diameter, shown in Figure 6.

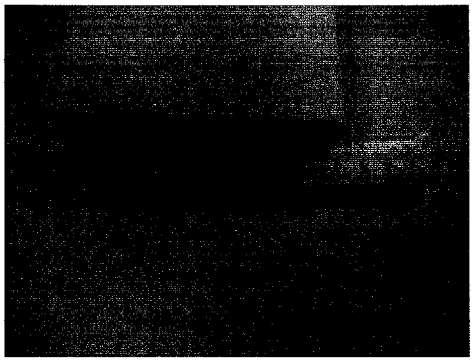
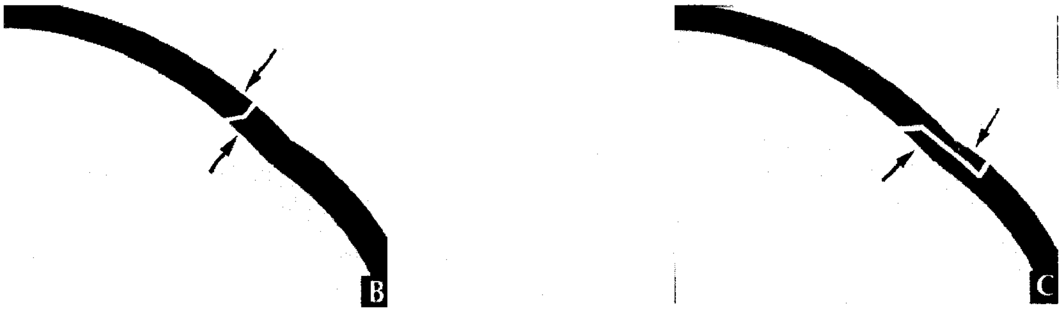


Figure 6 – Keratome: Tool for making incisions

The small diameter of the Keratome is convenient for making two or three-plane incision in the cornea, which facilitates wound healing and keeping liquids in the eye (Figure 7).



Two –plane incision

Three-plane incision[1]

Figure 7 – Two and three-plane cornea incisions

To guarantee the eye is fixed in one place during the incision step the surgeon holds the eye with forceps, which is shown below in Figure 8. The forceps are held with one hand while the Keratome with the other hand.

---

[1] Plane incision pictures taken from Cataracts presentation by S. El_Defrawy. [4]

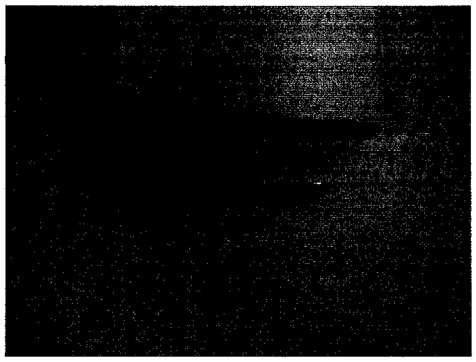


Figure 8- Forceps with straight tips

2. Capsulorhexis: This step involves the rupture of the lens capsule in order to reach the cloudy lens and clear it.

There are two ways to initiate this step. The initiation involves creating a flap in the lens capsule. The easier of these two is to utilize the Keratome used in the incision step to create the flap. The other option is to use a special tool, shown in Figure 9, which has a sharp tip to create the flap.

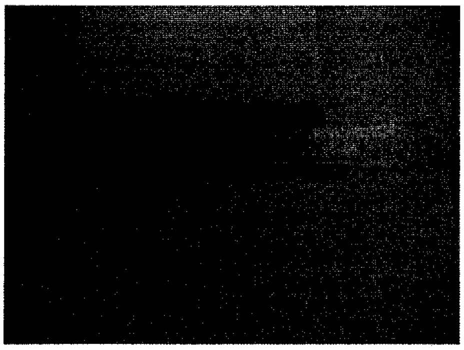


Figure 9 - Sharp edge tool for creating flap or aid in phacomelusification

The tip of this tool is sharp and is bent to allow for greater control. For the simulation purpose, this tool has to be modeled even if the trainee decides to use the Keratome to make the flap. The reason is that this tool has other functions, for instance it is used in the phacoemulsification process to destroy the lens.

After the flap is created, it is grabbed with special forceps that is bent inwards (Figure 10) and the flap is swirled in a circular motion to produce an opening in the lens capsule. This step is hard on the surgeon and has many complications as such the lens might 'run-out'. The analogy of this, as the doctor put it, is like peeling a tomato on one side and then the inside of the tomato runs out and does not stay within the rest of the tomato skin.

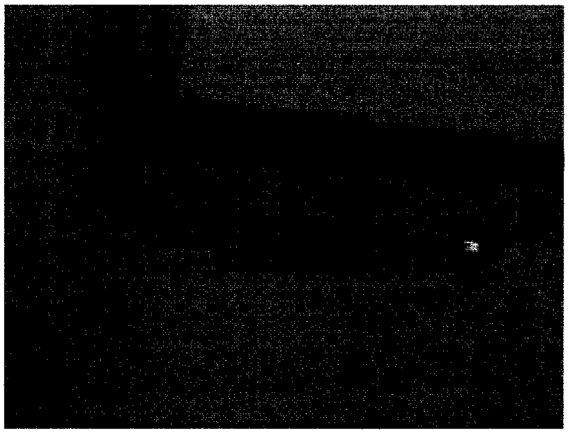


Figure 10 - Forceps with the tips bent inwards

3. Hydro dissection: To destroy the lens, a fluid is injected around the lens to do the preliminary dissection. This fluid is injected through a special syringe that has a bent tip, shown in Figure 11.

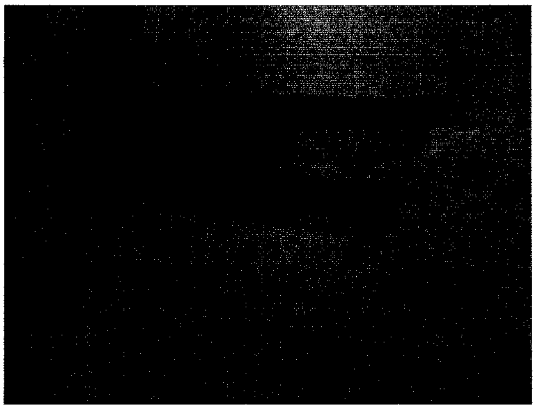


Figure 11 - Syringe to insert fluid into the eye

4. Phacoemulsification: This is the most crucial step for the lens destruction. With the aid of the sharp-tip tool, discussed above in Figure 9, held in one hand, the phacoemulsification tool shown in Figure 12 is held in the other hand and is inserted in the capsule hole created in the capsulorhexis step. Using ultraviolet beams this tool destroys the lens into tiny peaces.

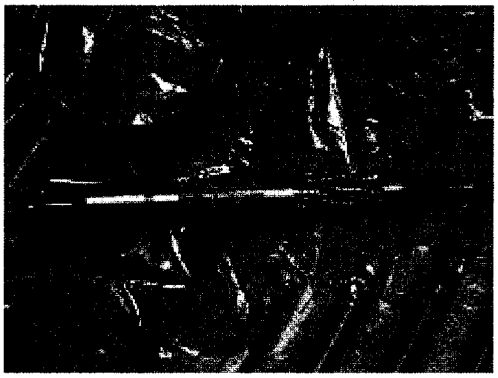


Figure 12 – Phacoemulsification tool

5. Cortical cleanup: After phacoemulsification, the destroyed parts of the lens remain in the lens capsule and must be removed. This step clears the capsule before the new lens can be inserted. The cortical cleanup can be done manually using the tool in Figure 13 or

it can be done through a suction machine using the same tool but in the latter case, the surgeon hooks up the tool to the suction machine via the tube that is attached to the tip of the tool.

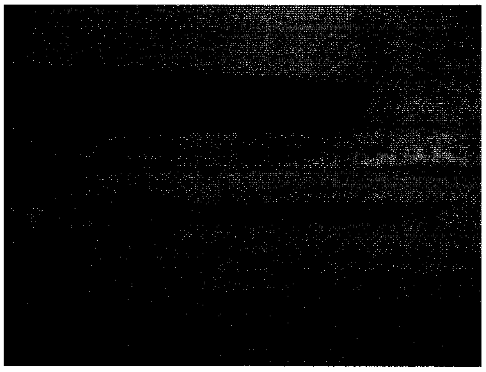


Figure 13 – Cortical cleanup tool

For that reason the white tube accompanying the tool is optional for connecting it to an electrical suction machine. For our simulation purposes this is not necessary, since most surgeons in the University of Ottawa Eye Institute do it manually without the machine.

6. Inserting the lens: The last step in the surgery is to insert the artificial lens. There are two ways to do this: Manually or using an injector that is shown below:

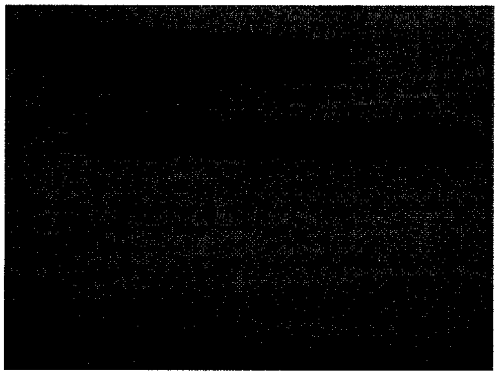


Figure 14 – Lens injector

Using the tool is easier; because the lens is attached to the tool and when injected, the lens spreads automatically to fill the lens capsule.

2.2 Cataract Surgery Current Training Procedure

As mentioned in the introduction, there has been a gradual shift in the training methods of surgery in general. The shift is geared toward technological advancements such as virtual simulators. However, over the years, it has been hard to come up with a surgical educational environment that would replace the apprenticeship environment of the operating room. This was achieved in other areas. Probably one of the most successful applications that replace the need for students to be in the operating room is the flight simulator for pilot training. In surgery, ophthalmic surgery specifically, there are no environments as effective as the flight simulators. [1] There has been attempts to get the ophthalmic surgery simulators as close to reality as possible, but it is still not sufficient to base medical students training solely on these simulators. One such attempt is the EyeSi system from VR Magic Labs in Germany, shown in Figure 15. [6]

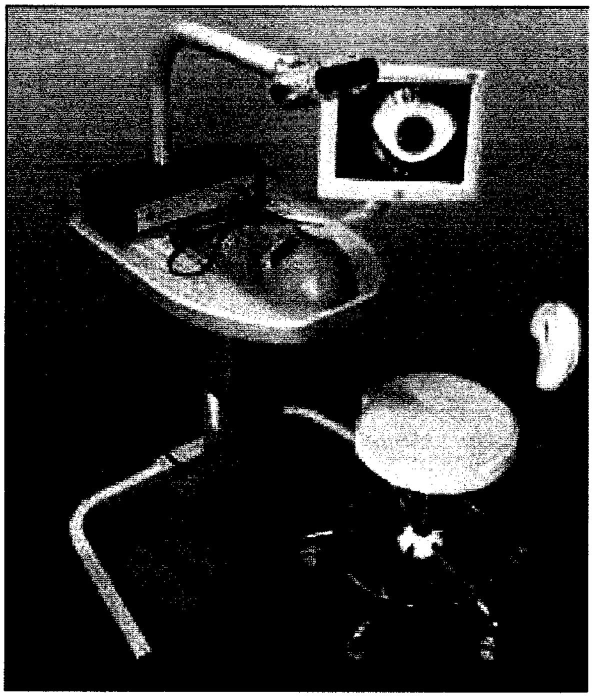


Figure 15 – EyeSi Ophthalmic Surgery Simulation

This surgical simulator provides the tools and environment to help surgeons acquire the skills needed for a vitreoretinal surgery. The goal is to shift from the traditional training methods gradually. However, there are certain shortcomings that still need to be overcome in order to achieve a state of complete reliability, similar to the reliability of the flight simulators in aviation. One such shortcoming is the addition of complications scenarios that might arise during an ophthalmic surgery. [8]

Previous training methods included analyzing surgery videos. Those conventional 2D videos have been recorded and analyzed for a long while by the medical community, but they are regarded universally by surgeons to be marginally effective for several reasons:

- The viewer is just a passive observer that cannot interact with the video. For example the user cannot change the view of the video.
- Subtle yet critical motions can be missed during the viewing.

- Video provides few depth cues and offers only linear control over the playback's timing.
- Watching the video over and over again will provide the trainee with same experience, rather than allowing a new or changed perspective.

However, the surgical videos have advantages. They provide the viewer with data that can build a trainee's knowledge base. Nonetheless, it comes short in providing the user with experience. [1] As one author describes in his paper for theoretical issues in experiential data [9] "experiences are not data." He describes experiences as "an intangible process of interaction among humans and the world that has its existence in human minds".

To compensate for the missing experience, training has been done on dead animals' eyes. This form of training was once very popular in the medical community, but it is losing its popularity gradually as substitute forms are emerging, such as the virtual simulation form of training. The animals' eye training, referred to by surgeons as wetlab training, is very high in cost. It needs a time overhead for preparations of the specimen to be operated on as well as cleaning the setup after the training is done. Besides, one of the serious limitations of this method is that animal eyes are not identical to human eyes in its physical properties such as size, shape, and feel and structure of the tissues. [8]

One doctor, expressing hope in the future of virtual training, summarizes the inadequacy of previous training methods and says: "The way we have learned ophthalmic surgery has been to read, to observe, and then to operate- on real live eyes on real live patients. We may do some wetlab training, but there is only so much we can learn from operating on pigs' eyes. When a young surgeon hits the road, the patients carry the risk." [6]

Chapter 3

Technical Background

3.1 Virtual Reality

In 1965, Ivan Sutherland, described in his concept paper “The Ultimate Display” a completely immersive virtual world where computer graphics would be so realistic they are even tangible. In his own words Sutherland says: “The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in...” [10]

This description, being depicted in 1965, would be so futuristic at that time and hard to envision. Yet, with the advancement of technology nowadays, we can relate more to this concept. We are still not there yet, having a computer controlling matter, and it still could be out of reach. However, the birth of virtual reality applications and the ability to interact with them through haptic devices brings us one step closer.

One misconception of virtual reality is the idea of telepresence, in which a user is immersed in a remote environment. Another misconception is interchanging the terms virtual reality with augmented reality. Augmented reality is imposing computer graphics on top of real images. The fact that real images are involved in the simulation, augmented reality does not fall under the category of virtual reality.

Some people associate virtual reality with only people wearing masks and goggles and interacting with the environment using gloves. This is one form of virtual reality but it is not the only categorization of it.

So what is virtual reality? A good definition is given in the book "Virtual Reality Technology" and it states: "Virtual reality is high-end user-computer interface that involves real-time simulation and interactions through multiple sensorial channels. These sensorial modalities are visual, auditory, tactile, smell, and taste." [11]

From this definition, we can deduce that virtual reality is a form of Human-Computer Interaction (HCI) where graphics are realistic enough, preferably in 3D, and immerses users by involving all of there senses from vision to touch.

It is worthy to note though that although taste and smell senses may be engaged, few applications do so. It may become common in the future, but present time applications focus more on the vision and the sense of touch in a virtual environment.

There are three important features in virtual reality referred to as "the Three I's of Virtual Reality". The three features are:

- Interaction: Users can interact with the virtual reality environment
- Immersion: Users engage their senses within the virtual reality environment
- Imagination: As virtual reality often involves a solution to a real-life problem in medicine, engineering, and military, the simulation performance depends on users' imagination.

The three I's of virtual reality are represented in Figure 16.

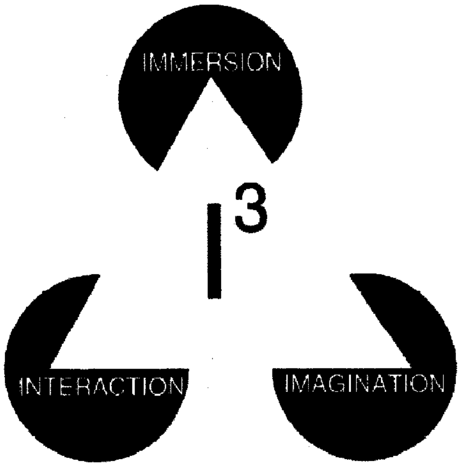


Figure 16 – The three I's of virtual reality, immersion-interaction-imagination [12]

3.1.1 History of Virtual Reality

The concept of virtual reality is not relatively new. Although it has been commercialized and widely accepted as a technology in the last decade, pioneers have been building the concept gradually from the 1960's. Table 2 presents a timeline of the progress of virtual reality over time [[11],[13]].

Year	Inventor	Product Name	Relation to VR
1962	Morton Heilig	Sensorama Simulator (Figure 17)	First virtual reality video arcade
1965	Ivan Sutherland	"The Ultimate Display"	Concept paper given definition to VR
1966	Ivan Sutherland	Head-mounted Display	View immersive images
1971	Fredrick Brooks	Three Dimensional Collision Force	Robotic arm that was a precursor to the haptic technology
1977	Dan Sandin, Richard Sayre, and Thomas Defanti	The first dataglove	Interaction through body movement

1981	NASA	Liquid Crystal Display-based HMD	Focus the image close to the eyes without effort. Majority of VR head-mounted displays still use this technique
1982	Bonnie MacBird	Tron	The first computer- generated movie
1983	Myron Kreuger	Video Space	First virtual environment
1984	William Gibson	Neuromancer	The term 'Cyberspace'
1985	Scott Fisher	Sensing Glove (Figure 18)	Integrated the glove to the NASA LCD-Head-Mounted simulation
1987	Michael Piller (Writer)	Star Trek – The Next Generation	The Holodeck, idea of immersive VR
1993	IEEE	San Diego Conference	First IEEE VR conference
1995	Silicon Graphics	VRML 1.0	Virtual Reality Modeling Language
1999	Larry and Andy Wachowski	The Matrix	Virtual Reality movie grosses $750M worldwide

Table 2 – Advancements of virtual reality over time

The following figures are related to the previous table

Figure 17 – The Sensorama Simulator prototype [14]

Figure 18 – NASA prototype with sensing glove [11]

3.2 Virtual Reality Applications

There has been a wide use of virtual reality applications. It has been used in different areas of industry from flight simulators to medical simulators. Even in the gaming industry, virtual reality inclusion is picking up there. An example is Halo2 game by Microsoft, running on the Xbox gaming console. In addition to the 3D graphics, the console comes equipped with a vibrating controller that vibrates differently according to the events in the game. There is a distinguished vibration for a character dying, shooting, or falling. This VR approach makes the game more exciting to players. [15] Another game that is completely haptic in nature is being developed at the DISCOVER lab, and utilizes the Phantom Omni haptic device from Sensable Technologies. [16] The game is called Hapticast and it is a multiplayer first person shooter game. The haptics interface was added for more immersion and excitement [17]. Figure 19 shows a person playing the game.

Figure 19 – Hapticast Game

There are certain advantages of using a VR application. The most apparent advantage is that it is appealing to the user. In addition, there are certain practical advantages that a VR application can provide. Since this thesis is about medical applications, the advantages listed below belong to medical simulators, but comparisons can be made with other VR applications. The advantages include [3]:

1. Medical simulators allow trainees to return to the same procedure or task several times in order to improve their surgical skills.
2. Medical simulators eliminate the need for other expensive alternatives, such as training on animal eyes.

3. The haptic feedback provides realistic simulation of objects including the real behavior of tissues and medical objects.
4. Medical simulators provide a real-time learning environment for surgeons.

Fundamentally, VR applications provide an enriching application to the user. They also have industry advantages such as cost reduction of simulators' training. [3]

To take an idea of how diverse the type of VR applications can be, Table 3 shows some VR applications in industry and education [15]:

Company	Type of Application	Description
Center for Advanced Studies, Research and Development in Sardinia (CRS4)	Medical	Catheter Insertion, mastiodectomy, and bone dissection applications
Distributed & Collaborative Virtual Environments Research (DISCOVER) Laboratory at the University of Ottawa	Industrial Training	An application intended to train technicians to replace faulty ATM switch board.
DISCOVER Lab	E-Commerce	Showcasing a car in a showroom where users can interact with the car haptically
Novint	Virtual Reality Dental Training System	Used to train dental students
Reachin	Oil Industry Application (GeoEditor)	Plug-in for an earth modeling software, designed for geologist in the oil industry to model VR simulations
Team GAMMA at the University of North Carolina, Chapel Hill	ArtNova: Artistic Appliction	VR modeling and applying texture to 3D objects

Table 3 - Virtual Reality applications efforts and research

3.3 Haptic Devices

The term haptics is derived from the Greek verb “haptesthai” which means “to touch”. Nowadays the term refers to the science of sensing and manipulation through the sense of touch. It specifically refers to the human touch and force-feedback with an external environment, whether the environment is real or virtual. From a computer science and technological perspective, the environment is usually virtual and haptics refers to the force-feedback that an individual experiences when interacting with the virtual world.

Haptics have been included in a variety of applications [18], which include:

- Data Visualization: Data visualization exploits interactive diagrams to solve a given problem. Including haptic devices to interact with the diagrams will enrich the experience.
- Medical Simulation: Using haptic devices in simulations will allow simulating many medical procedures. Our cataract surgery simulation is an example of that.
- E-commerce: In e-commerce applications, the transactions between the customer and merchandiser will be more comfortable, if customers were able to haptically feel the texture and test the product of the goods they are about to purchase.
- Education: Haptics can be used for geometrical and mathematical models constructions.
- Entertainment: In the entertainment sector, haptics are most widely used in gaming. Vibrating joysticks are very popular.

- Arts and Designs: Haptics opens new channels for virtual modeling. The modeling includes sculpturing and painting.

Haptic devices have seen a surge in the 1990's. The first attempt to commercialize haptic devices was by SensAble Technologies cofounder Thomas Massie. In his thesis, when he was a Master's student at Massachusetts Institute of Technology (MIT), entitled "Initial Haptic Explorations with the Phantom: Virtual Touch through Point Interaction" [19], Massie describes the Phantom haptic device which possess a pen-like structure and can interact with virtual objects through poking with the tip of the pen. More on the Phantom is described in the next section.

Other commercial haptic devices have emerged, and others were created for research purposes. Table 4 surveys the haptic devices available commercially; the table is adapted from a Master's thesis of a DISCOVER Lab member [15].

Company	Product	Description
FCS Control Systems	HapticMASTER	A 3 degrees-of-freedom haptic device that combines high position and force resolutions with an extended workspace and a maximum of 250 N of force output.
Immersion	Haptic Workstation	Immersive hapto-visual interface with goggles and haptic devices fitting over both hands with range of motion pivoting at the shoulder. The Haptic Workstation is made up of two Immersion CyberForce systems.
Immersion	CyberForce	A 6 degrees-of-freedom armature that communicates force feedback (both translational and rotational) to the arm and hand.
Immersion	CyberGrasp	A 1 degree-of-freedom exoskeleton

		device that fits over a hand and provides forced feedback to each finger individually.
Immersion	CyberTouch	A vibro-tactile feedback device that can fit over the fingers and the palm of a hand.
Immersion	CyberGlove	A sensor glove that measures position and bend of the hand and fingers.
MPB Technologies	Freedom 6S	A 6 degrees-of-freedom haptic device with high position and force resolution. Stylus based.
SensAble Technologies	PHANTOM Omni	First haptic device cost-effective enough to be marketed directly to consumers. Stylus based.
SensAble Technologies	PHANTOM Desktop	High-fidelity haptic feedback through nominal position resolution of about 0.023 mm. Stylus based.
SensAble Technologies	PHANTOM Premium 3.0/6 DOF	Offers full arm movement pivoting at shoulder with six degrees of freedom in movement. Nominal position resolutions are: ~ 0.02 mm translational; 0.0023 degrees rotational (yaw and pitch); and 0.008 degrees rotation (roll). Stylus based.

Table 4 – Haptic Devices Survey

3.3.1 SensAble Phantom

The Desktop Phantom from SensAble Technologies is a six Degrees of Freedom (DOF) metallic haptic device, which is distinguished with precise positioning input. Its portable design makes it suitable for haptic research and simulations such as medical simulation.

The range of force feedback is on the x, y, and z axis only, making it three DOF force feedback device [20]. The Phantom Desktop is shown in Figure 20.

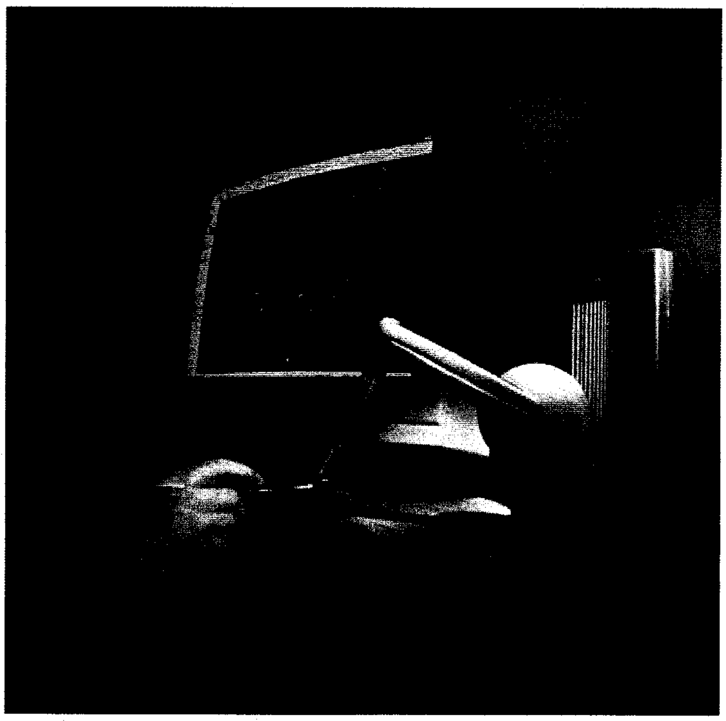


Figure 20 – SensAble Technologies Phantom Desktop

3.3.2 MPB Freedom 6S

The Freedom 6S from MPB technologies is a 6 DOF device both in position and force feedback. The Freedom 6S is bulky in size but has a light touch and accurate force feedback. One distinguished feature of the Freedom 6S is that it can be purchased in two forms. One form is the left-hand model and the other is the right-hand model, depending on the preferred orientation of the developer [21]. The MPB Freedom 6S is shown in Figure 21.

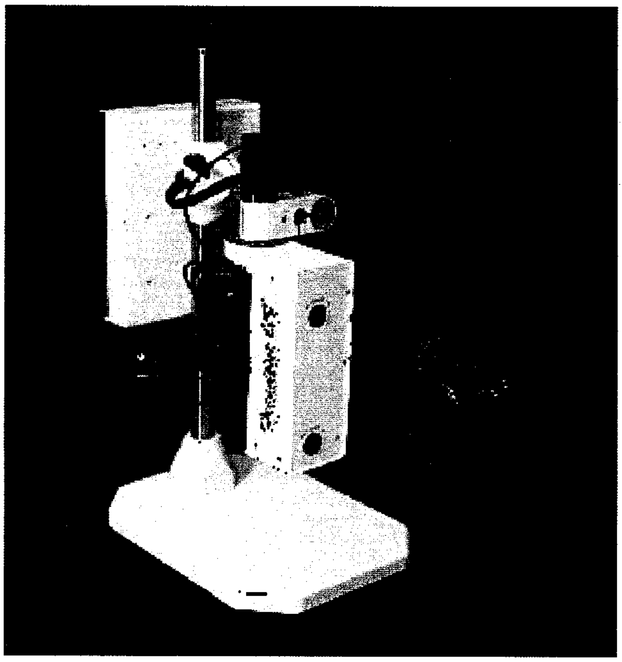


Figure 21 – MPB Technologies Freedom 6S

3.3.3 Comparing Phantom Desktop with Freedom 6S

The following table compares the Phantom Desktop with the Freedom 6S in terms of their specification [15]:

	Phantom Desktop	**Freedom 6S**
Degrees of Freedom	6 DOF positioning, 3 DOF haptic feedback	6 DOF
End Effector	Stylus	Stylus
Workspace	~2,700 cm^3	~13,000 cm^3
Position Resolution	20 μm	20 μm
Force Resolution	Impedance force control: 7.9 N	Impedance force control: 2.5 N over 60 seconds or 0.6 N continuous

Table 5 – Comparison between Phantom Desktop and Freedom 6S

For our project we decided to use the Phantom since it has a compact design, which makes it easier to manipulate delicate organs such as the eye organ.

3.4 Haptic Application

3.4.1 Ghost SDK

The Ghost SDK from SenAble Technologies contains the Ghost API. The API consists of only haptic rendering which classifies it as a first generation API. The API does not contain any graphic rendering information. The synchronization is done through method call back from the application that contains the scene to the haptic process (Figure 22).

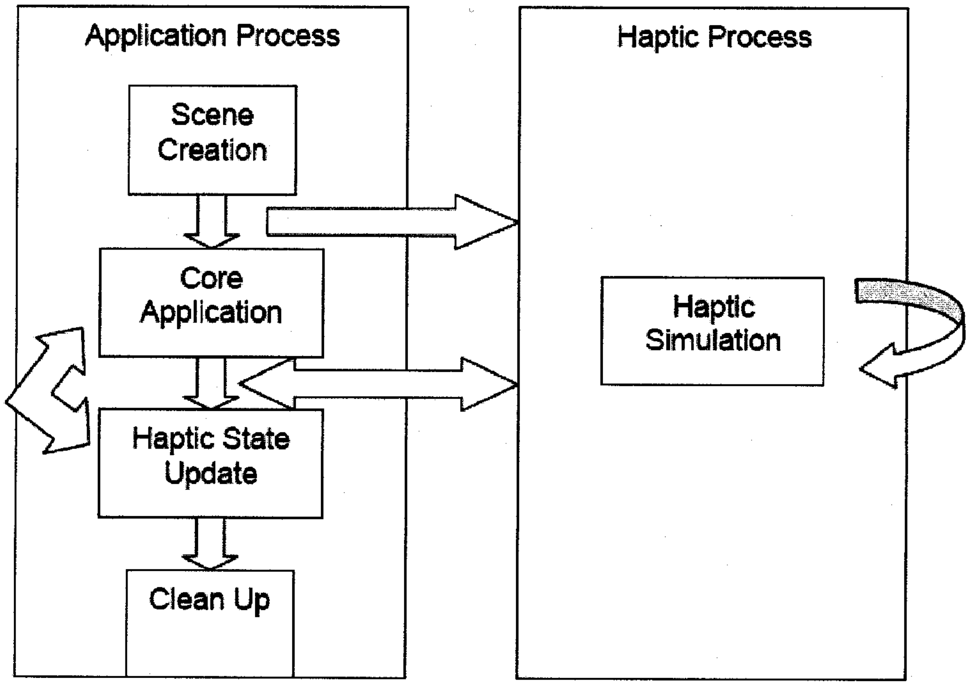


Figure 22 - Application using the Ghost SDK

The Ghost API is based on the scene graph paradigm. The API is in C++ and it is extensible through inheritance classes. [22]

3.4.2 Reachin Display and API

Reachin Technologies is a company located in Stockholm, Sweden. It specializes in human-computer interaction via haptics; the sense of touch. It manufactures hardware related products named Reachin Display as well as software related products known as Reachin API. Through the combination of the hardware and software products, Reachin facilitates the development of applications related to three dimensional environments that are touch-enabled. [23]

Reachin Display is a set of hardware that allows the user to view and interact with the object as if in the real world. The ergonomic features of the Display, viewed in Figure 23, permit the co-location of the 3D view and haptic response while utilizing a minimum space.

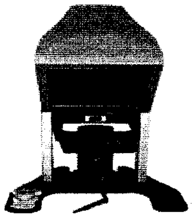


Figure 23 – Reachin Display

There are other advantages of the Display which is made of aluminum such as the stereo view done through a glass located on top of the haptic device that reflects the stereo monitor. The location of the haptic device allows the user to interact intuitively with the virtual objects as the user is viewing the environment through the glass. Note however that the user should be wearing a 3D eyeglass. [24]

Reachin Display integrates a Phantom force feedback device from SensAble Technologies. The Phantom Desktop (Figure 24), described earlier, has a portable design, which makes it easy to integrate with Reachin Display.

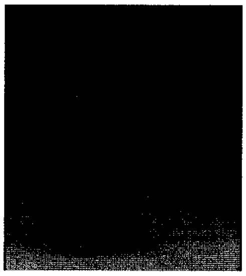


Figure 24 – Phantom Desktop

The Reachin API is a development platform that allows the user to build 3D haptic applications using one of those three programming languages: C++, Python, or VRML (Virtual Reality Modeling Language). The advantage of the API is the prewritten libraries and classes that allow the haptic, graphics, and audio synchronization. It also allows the user to implement the required force feedback easily using the library available. [25] More on Reachin API will be discussed in Chapter 5.

3.4.3 eTouch

eTouch SDK from Novint is an SDK that provides the means of building a 3D environment with sense of touch, sight, and hearing enabled. Most important feature is the fact that eTouch provides a graphical user interface that allows the developer to easily integrate the 3D graphics with the haptic feedback. eTouch API enables the user to control a 3D cursor in the environment and experience the force feedback through that cursor. The API supports multiple haptic devices including the Phantom haptic device from SensAble Technologies. [26]

3.5 CANARIE HAVE project

The CANARIE HAVE (Haptic Audio-Visiula Environment) project is funded by Canada's advanced Internet organization referred to as CANARIE (www.canarie.ca). They have provided the CAnet^4 a high speed dedicated network linking the University of Alberta to the University of Ottawa. The goal of the project is aimed at developing a hapto-audio-visual environment to optimize the medical training experience, especially in surgical training of ophthalmic surgery.

University of Ottawa and the University of Alberta provided the virtualized application of the surgery. The University of Ottawa Eye Institute provided the medical research support, while BigBangwidth provided the Lightpath Accelerator to produce huge data rates (Up to 10 Gigabit) across the CAnet^4 network. [27]

Chapter 4

Graphics Modeling and Environment Design

Graphics and the environment are very important aspects of simulation. For surgical simulations in particular the use of 3D graphics modeling has been gaining popularity. [28] Some examples of high-end interactive graphics used in surgical simulations are:

- Endoscopic simulator that contains a simplified representation of the colon [29]
- A simulation of tendon transfer operations with simplified geometry [30]
- CT and MRI generated graphic models for simulation [31]
- Simulation of laparoscopic surgical procedures in real time [32]

This chapter goes through the modeling of the eye organ and the surgical tools needed for the surgery. It also covers the user interface used in the application, and how to actually navigate through the application.

4.1 Virtual Eye Modeling

The eye organ is a complex organ. The anatomy of the eye is very detailed. Most eye structures modeled is part of the human face and the finite details are obscured as it is not necessary when looking at the face from a distant.

One of the few papers that discuss the detailed modeling of the eye is called "A Virtual Environment and Model of the Eye for Surgical Simulation". [28] In their paper they have developed a meticulous eye organ and the surrounding eyelids. Their model takes into account mathematical angles and they use nurbs to model the exterior of their eye. From the exterior the eye looks like a high-end facial model (Figure 25), so they have hidden the details from the viewer.

Figure 25 – Exterior view of an eye model using nurbs

For our specific cataract surgery application, we did not require that amount of details. We needed enough details to perform the surgery, but at the same time do not take a lot of memory in order for the haptic application to be rendered in minimal amount of time. The following section describes the anatomy of the eye, followed by description of what we did to model our eye with proper requirements.

4.1.1 Anatomy of the Eye

In its core structure, the eye consists of three layers, as shown in Figure 26 [33]

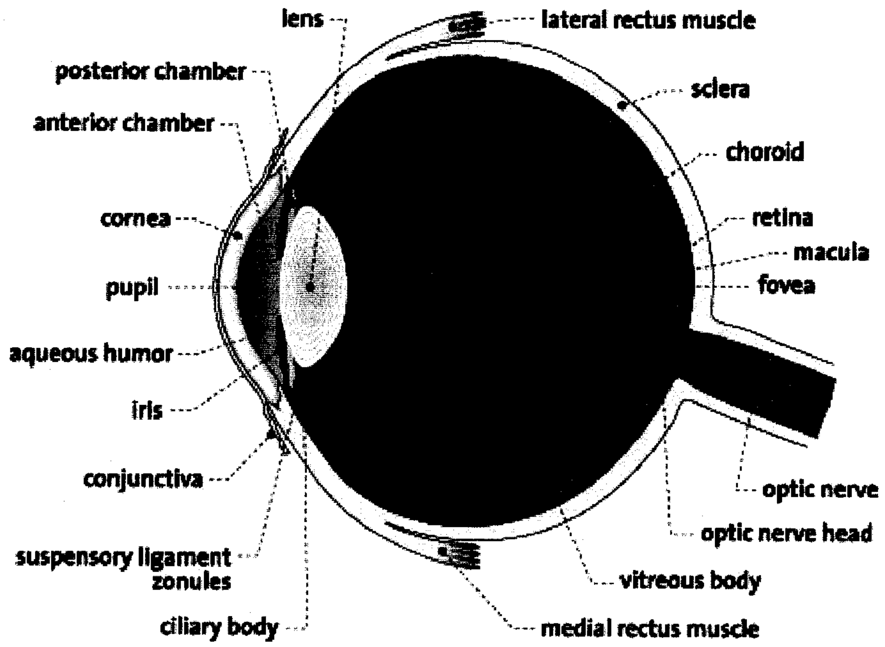


Figure 26 – Anatomy of the eye

The first layer called sclera is responsible for maintaining the shape of the eye. The front end of the sclera is called cornea which is a transparent section responsible for refracting light entering the eye. The fluid beneath the cornea, the aqueous humor, provides support to the cornea and prevents it from collapsing. In ophthalmic surgery cornea is the entry point to the interior of the eye. For example in cataract surgery, the first step is incision to the cornea which provides a passage to the interior.

The middle layer has several functionalities such as nourishment. It contains ciliary bodies and the iris, which is the pigmented area visible through the cornea. The iris provides the eye with its color (brown, green, blue …etc). During the cataract surgery, surgeons tend to bypass the iris without incisions, through the opening in the iris called the pupil. Depending on the patient, infrequently surgeons have to extend the iris in order extend the pupil area.

Below the iris resides the lens capsule. The lens capsule contains the eye lens responsible for focusing the incoming light. It is in this part of the eye where cataract occurs. And the lens is the destination for eye cataract surgeons wishing to destroy the cataract lens and replacing it with an artificial one.

The last layer is called the retina. It is connected to the optical nerves. It is the visually receptive area, and communicates with the brain for signal and image interpretation. The third layer is usually not operated on in ophthalmic surgery. Particularly in cataract surgery, the third layer is irrelevant. [[4], [28]]

4.1.2 Eye Modeling for the Cataract Surgery

The modeling of the eye started by purchasing a commercial eye model, which we used as a precursor for developing our own model. The commercial model we have purchased from www.exchange3d.com contains 16,000 polygons and screenshots of the model is shown in Figure 27.

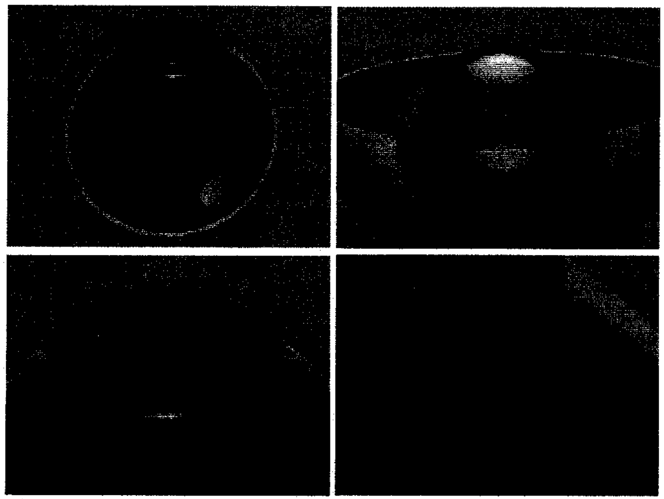


Figure 27 - Eye Model with 16,000 Polygons

The purchased model is in .max format, which can be viewed and edited with 3D Studio Max software [35]. It represents a cross-sectional view of the eye as it can be seen from the screenshots above. All the three layers of the eye, described in the previous sub-section, are presented. What was important to us is that the entire eye sub-organs essential to the cataract surgery are visible and emphasized, namely: cornea, iris, and lens capsule.

The .max format is a proprietary format, owned by Autodesk Company the developers of 3D Studio Max software. 3D Studio Max is a high-end modeling software allowing us to create sharp images with high polygon ratio. This ratio, although results in superb quality, it does not completely fulfill our application's requirements for two reasons:

1. The high polygon ratio, will consume system memory, rendering it virtually unsuitable for real-time interactions. Since our application is a training simulation, real-time requirements are stringent and they should be accounted for.
2. The Reachin API we used to create the application, support only one graphics format which is the Virtual Reality Modeling Language (VRML) format. Hence it was required that we convert all the graphics to VRML.

Fortunately, 3D Studio Max allows graphic conversion to different formats among them is the VRML format. Following each conversion, there is a loss of quality however. This loss can be attributed to the difference between the two formats, .max and VRML. VRML is an ASCII format that can be modified through text editors, and can be loaded into browsers, such as HTML would. There are tools that allow development of VRML files using interactive graphical software, in which the developer needs to know minimal

VRML syntax, an example of such software is Cosmo Worlds by Computer Associates who discontinued supporting the software[36]. However most of the development of VRML files is text based such as the HTML when it first came out. VRML has a hierarchical composition, meaning it has a parent-children relationship between the nodes that comprise the graphics [37].

3D Studio Max format on the other hand, the .max format, can only be loaded and edited in the 3D Studio Max software. It has a binary format, and it saves certain information that cannot be transferred when performing the transformations such as lighting, shading, and camera angles. All this data is lost when using the 3D Studio extractor, the utility which converts the graphics to the required format. The most important feature that is lost during the transition is the texture add-ons. In 3D Studio Max, certain textures can be specified during the modeling, which add realism to the application. During the VRML conversion, these textures are not visible.

We have completed the eye model in 3D Studio Max by continuing the other cross section and resizing certain sub-organs, one such is making the cornea a little more concaved. For visibility purposes we chose a different color schema for the eye. Some colors chosen were for testing purposes, and were later changed to the original colors. For example, we chose to make the cornea, which is transparent in nature, appear purple in order to see the marks of the cutting when performing an incision with a keratome. The post-modified eye is shown in Figure 28.

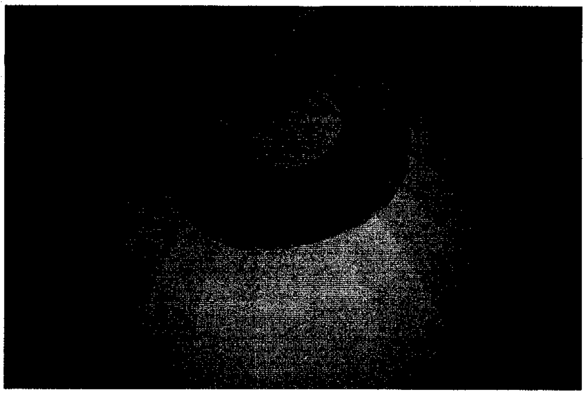


Figure 28 – Eye organ modified for the cataract surgery

After the modification in 3D Studio Max, we converted our model to VRML. As expected, the components of the eye were unaffected in terms of functionality, but there graphics lost quality. To compensate for the loss of quality, we had to do elaborate texture-mapping through editing of the VRML file by grouping certain nodes together and applying the required texture. The most remarkable texture applied is the texture for the iris shown in Figure 29. In Figure 30, an excerpt of the VRML code is shown that deals with adding the texture to the iris.

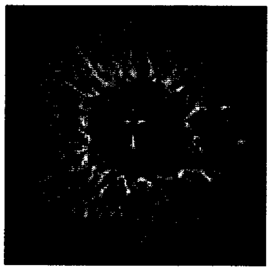


Figure 29 – Texture for the iris

```
DEF iris Transform {
      children    Shape {
      appearance  DEF irisAppear Appearance {
        material  DEF irisMat Material {
          ambientIntensity    0.119048
          diffuseColor  0 0.8 0
          specularColor 0.32 0.4 0.4
          emissiveColor 0.07 0.13 0.22
          shininess     0.54
        }

        texture   ImageTexture {
          repeatS TRUE
          repeatT TRUE
          url     "iris2.png"
        }

      }

      geometry    DEF iris-FACES IndexedFaceSet {
      .............

      }
```

Figure 30 – Code for inserting texture into VRML iris

To make the eyeball more realistic we added a shiny white texture to it, and then added shades of red color, to mimic the red blood vessels. The texture for the eye ball is shown in Figure 31.

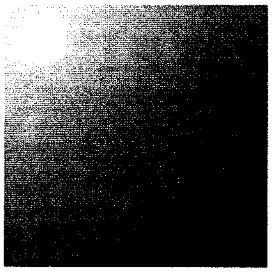


Figure 31 – White shade texture

After all the modifications and texture mapping, the final eye model result looked as in the screenshot of Figure 32. Note, however, that the eye is intended to be viewed in 3D environment using special glasses, and the screenshot does not reflect the view exactly.

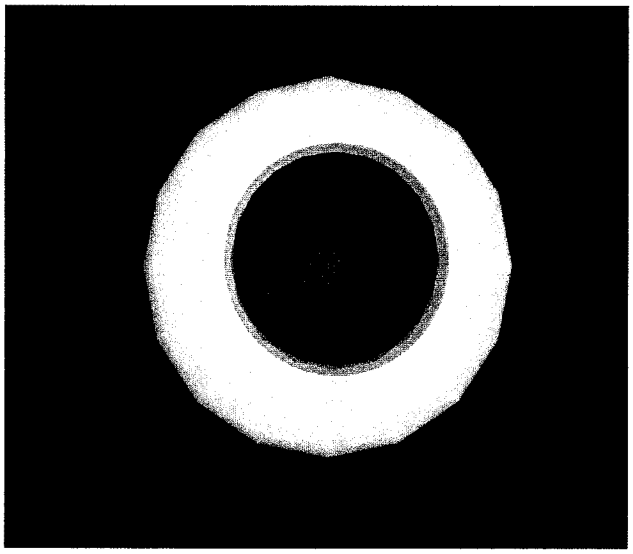
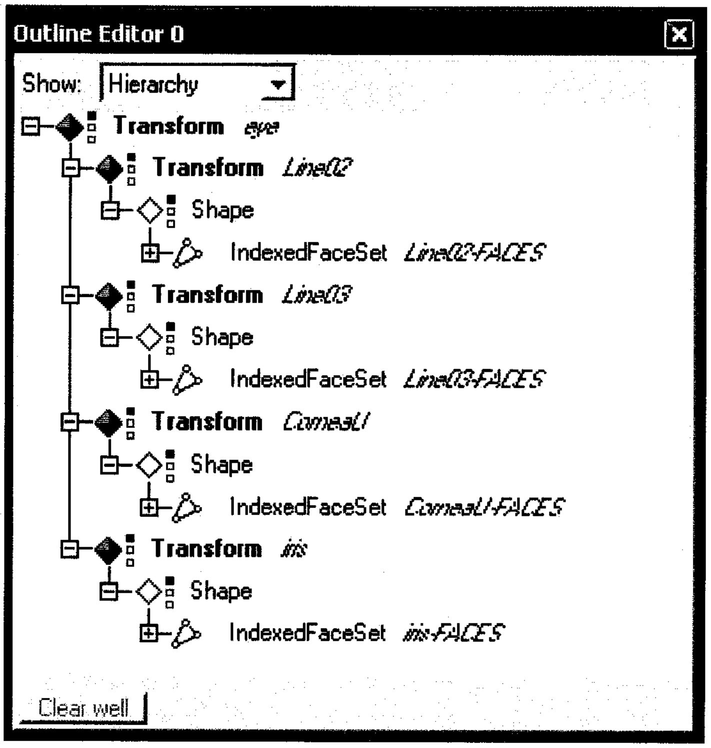


Figure 32 – VRML eye used in the simulation

To simplify the algorithm of the actual surgery, given the complexity of the eye, we designed the VRML with the following hierarchy in Figure 33, taken from the VRML software Cosmo Worlds:



Show: Hierarchy

Figure 33 – Hierarchy organization for the VRML eye

Node Eye is the parent node, and it has four children: Line02, Line03, CorneaU, and iris. Line02 and Line03 in the above figure, refers to the lens capsule and the white part of the eye respectively. For code maintenance, they should be renamed, but any instance appearing in the source code should in effect be renamed in the future.

This hierarchal structure allows us to manipulate the eye node as whole through the parent node Eye, but at the same time it allows manipulation of sub-nodes through our source code. For example, we can restrict the cutting to CorneaU (the cornea), while restricting the keratome from cutting any other sub-organ, thus redirecting the trainee to the appropriate cutting place.

4.2 Medical Tools Modeling

Initially, we did a survey on all the tools required for the surgery. This survey was deducted from watching surgery videos, and lecture notes from medical doctors [4]. The initial survey is given in Table 6.

Surgical Stage	Tools
a- Local Anesthesia	Syringes
b- Cornea incision	
i- Scleral Tunnel	- Forceps - Crescent knife
ii- Side Port incision	- 30^{o} Blade - Hydrodissection cannula
iii- Clear Cornial incision	Keratome (may use 1 or 2 blades)
c- Penetration in anterior chamber	Keratome (may open tunnel with viscoelastic)
d- iris bypassing	
i- incision	(not preferable)
ii- stretching	Hooks
e- Capsulorhexis	- Keratome/ or -Bent Forceps
f- Cloud Removal	
i- Hydro dissection	Hydro dissection Tube
ii- Phacoemulsification	- Phacoemulsification instrument - Forceps
iii- Divide and Conquer	- Specialized instrument - Forceps
iv- Stop and Chop	- Specialized instrument - Forceps
v- Phaco Chop	- Specialized instrument - Forceps
g- Cortical Cleanup	Suction Tube
h- lens placement	- Intraocular Lens - Lens Injector

Table 6 – Initial survey of tools required for surgery

To support our survey we have visited the University of Ottawa Eye Institute to see the actual tools they use, and to take measurements and photos. The result of the visit was summarized in section 2.1.2 Summary of Cataract Surgery Procedures. The photographs of the tools in section 2.1.2 Summary of Cataract Surgery Procedures were the photos taken at the eye institute. The ruler that appears in the background was used to take the dimension of the tools.

Our initial survey proved to be relatively accurate. There were some options that doctors do not usually undertake, and they prefer to take the most common procedure. For example, in Table 6 under cloud removal, there are five options to destroy the lens containing the cataract. The most common option is option two which is phacoemulsifying the lens. Therefore, when modeling the tools only the phacoemulsification tool was modeled as it is the preferred method.

Some tools where modeled for completeness but were not actually used in the implementation. The peripheral steps were not implemented, only the core steps were. The peripheral steps include: Giving local anesthetic, iris bypassing, and cortical cleanup. These steps aid in the surgery but are either rarely used (iris bypassing), or done by an aid such as a nurse (local anesthetic).

The modeling of the tools took similar approach to the modeling of the eye. We have built the tools from scratch, except for the syringe which we obtained from a 3D models website. Modeling was done in 3D Studio Max, and then converted to VRML. No special texture was used except for the intraocular lens, which is the artificial lens to be replaced. The loss of texture when converting to VRML wasn’t substantial. All the tools were under one primary node in VRML, this way we manipulate the tool as whole unit. One consideration we took when modeling the tools is the naming of the nodes that describes the primary node. In VRML, each node has children nodes that describe the shape, texture, the color coordinates, the coordinate index...etc. Each tool we modeled has the same naming sequence for all these sub-nodes; the general format is given in Figure 34, where 'TOOL' is replaced by the tools name for each tool. The figure also shows that there is only one top node definition referred to by the tools name. This hierarchy approach and naming convention allow us to write only one class in C++ to load every tool by just specifying the name of the tool in the constructor of the class.

```
DEF TOOL Transform {
  children  Shape {
    appearance    DEF TOOLAppear Appearance {
      material    DEF TOOLMat Material {
      ambientIntensity  0.6418
      diffuseColor      0.7882 0.7882 0.7882
      specularColor     2.39 2.39 2.39
      shininess   1
      transparency      0
      }

    }

    geometry      DEF TOOL-FACES IndexedFaceSet {
      coord DEF TOOL-COORD Coordinate {
      point [ -88.02 0 45.9,
              -0.1927 0 -132.8,
             .........
            ]
      }
```

Figure 34 – Excerpts from the general structure of a VRML file describing the tool

Another consideration we took when modeling the tools is its size compared to the eye. The tools models had a realistic eye size over tool size ratio. We deduced this ratio from the measurements we took in the Eye Institute. The tools modeled are displayed in the figures below.

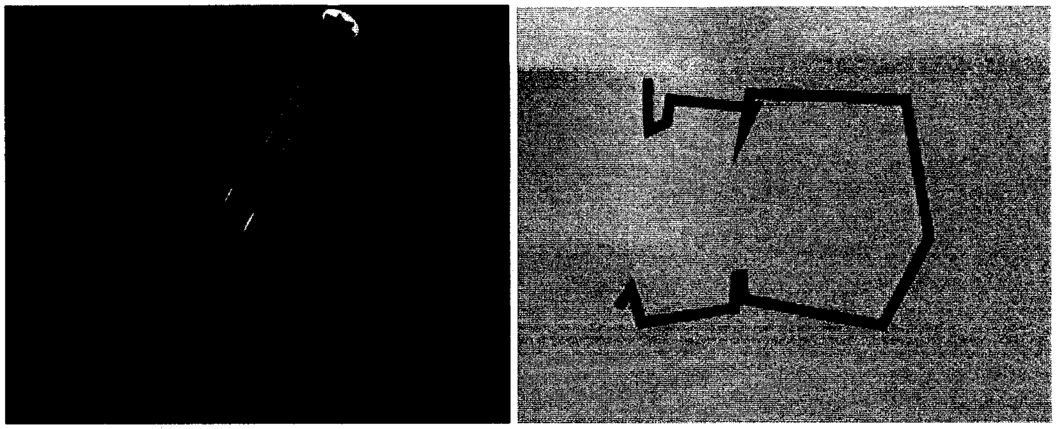


Figure 35 – Syringe (left) and Eye Holder (right)

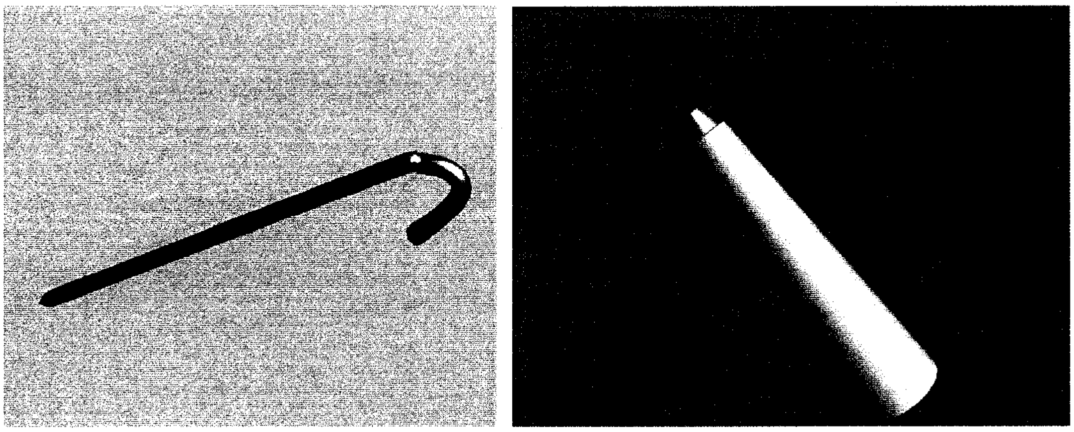


Figure 36 – Hook (left) and Crescent Knife (right)

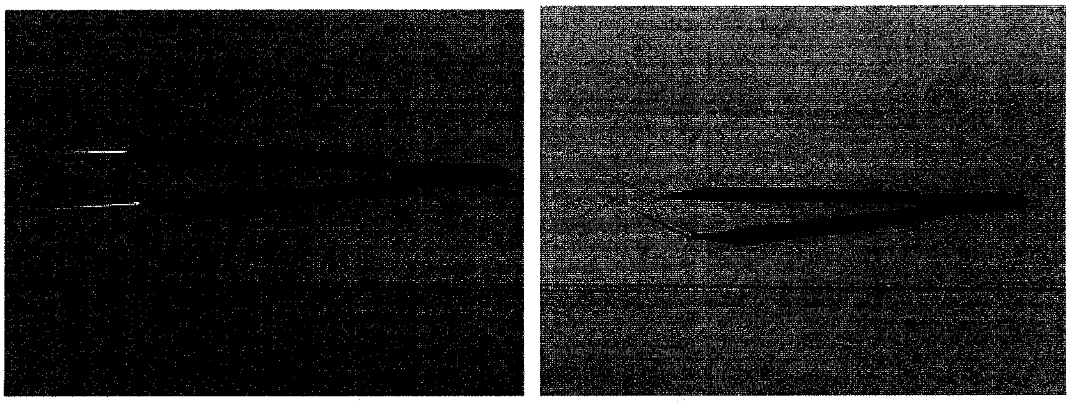


Figure 37 – Straight Forceps (left) and Bent Forceps (right)

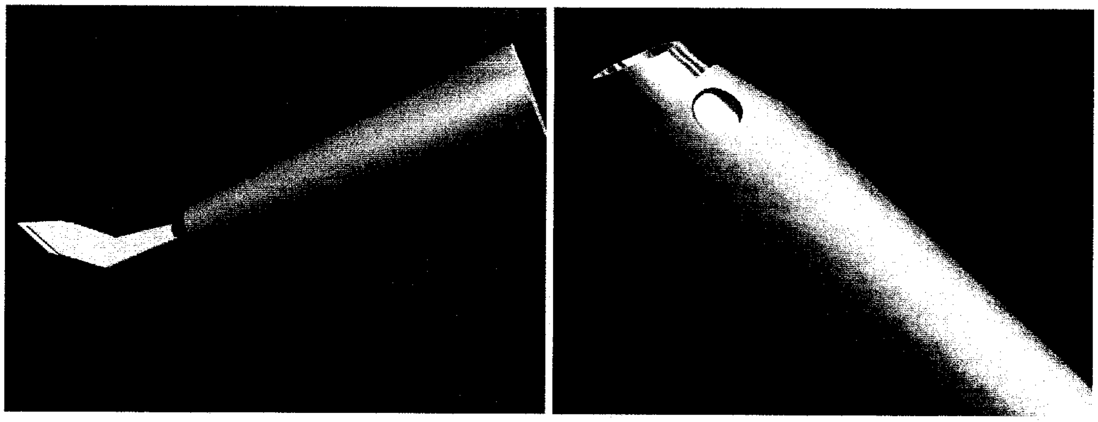


Figure 38 - Keratome (left) and Phacoemulsification Device (right)

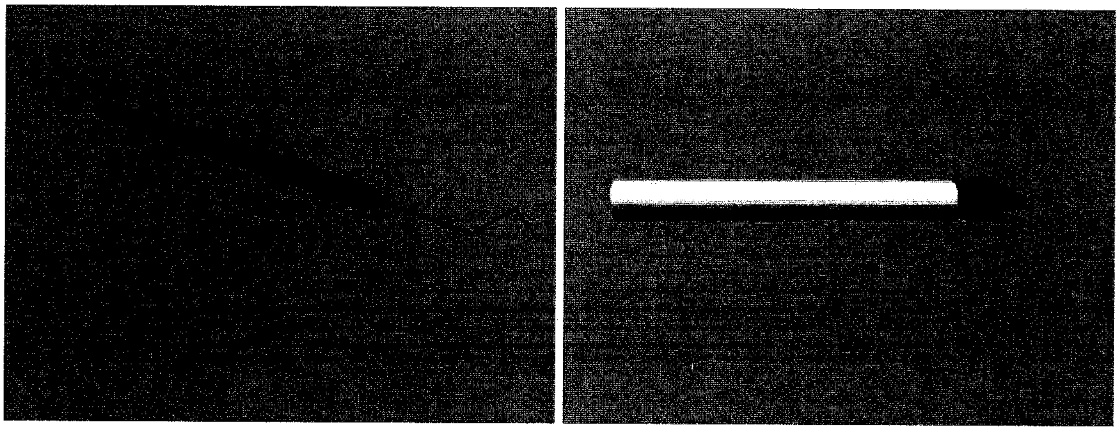


Figure 39 – Sharp edge tool for capsulorhexis (left) and Cortical Cleanup tool (right)

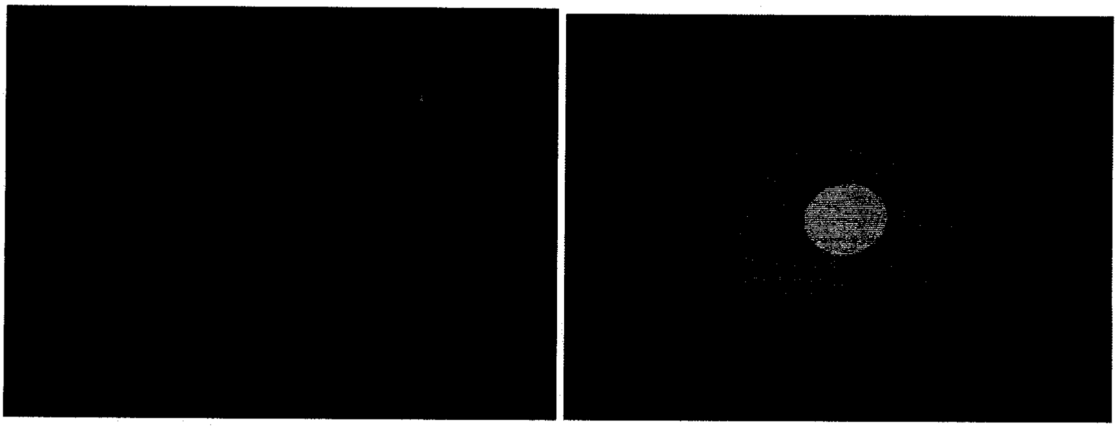


Figure 40 – Lens Injector (left) and Intraocular Lens (right)

4.3 User Interface and Environment

When creating the user interface of the application, we had the following considerations to take into account:

1. What is the best way a user can interact with the environment and the interface? Is it better to manipulate the surgery with the haptic device, and maneuver the user interface with a keyboard and mouse? Or is it better to exclude the usage of the keyboard and mouse altogether and confine users' interaction of the surgery and the user interface through the haptic device?
2. Given the 3D environment, how rich should the user interface be? Is it subtle to design a regular user interface that looks similar to a 2D environment, or should special consideration be taken for a 3D environment?

Pondering upon these issues, we have decided that using only the haptic device as an input device is a better fit for the user. Especially in a 3D environment sometimes the viewing screen is far enough from the main computer, so the mouse and keyboard can be out of reach. We did not want the user to be constantly struggling when shifting between performing the surgery and navigating the environment and interface. Another reason is the way a user can be immersed in the 3D display when wearing special glasses and looking through the special 3D screen of the Reachin Display that accessing the keyboard or mouse can be a nuisance. Some users may feel uncomfortable with wearing the 3D glasses without looking in the 3D screen that they have to toggle wearing the glasses each time they decide to click on a different button in the user interface.

Taking the second point into consideration, a 3D environment varies from a 2D one. In a 2D platform, if the environment is clustered it is trivial to users, as most 2D applications posses a rich interface. For the 3D platform a clustered interface will reflect a negative navigation experience to users, as they will feel more comfortable with a lot of free space in the environment. Thus, we have decided to keep the user interface as simple as possible without any complications. We have only provided essential buttons to the user and designed them to occupy minimal space, leaving users enough free space to roam around.

When the application loads up, the eye is showed in the center of the screen, with a single button at the top right corner labeled 'Tools'. The haptic interface controls a default virtual pen that loads with the application. The default pen, provided by Reachin API, has a black round tip, that is the point of contact with eye, and with the 'Tools' button.

Clicking on the 'Tools' button, will bring up a list of tools buttons that users can select as well as a 'Quit' button at the end. Selecting any of the tools button, or the quit button is done with the haptic device by pushing the virtual pen toward the button. When the quit button is selected the application exits, but when a tool button is selected the selected tool will load up replacing the virtual pen. Consequently, the manipulation is done through the selected tool, whether performing a surgical procedure on the eye or selecting another button from the list. In essence, the tools will have two functions: performing a surgical procedure or acting as a haptic input device for controlling the environment.

Figure 41 shows screen shots of the application, one with the knife selected and an incision is being performed, and the other one when the 'Tools' button is clicked and a list of tools appears, and a selection of one of the tools is being performed via the virtual pen. It can be noted that the screenshots are inverted. This is due to the Reachin Display which inverts the image correctly through the glass that the user looks in, thus requiring the image on the regular monitor to be inverted upside down.

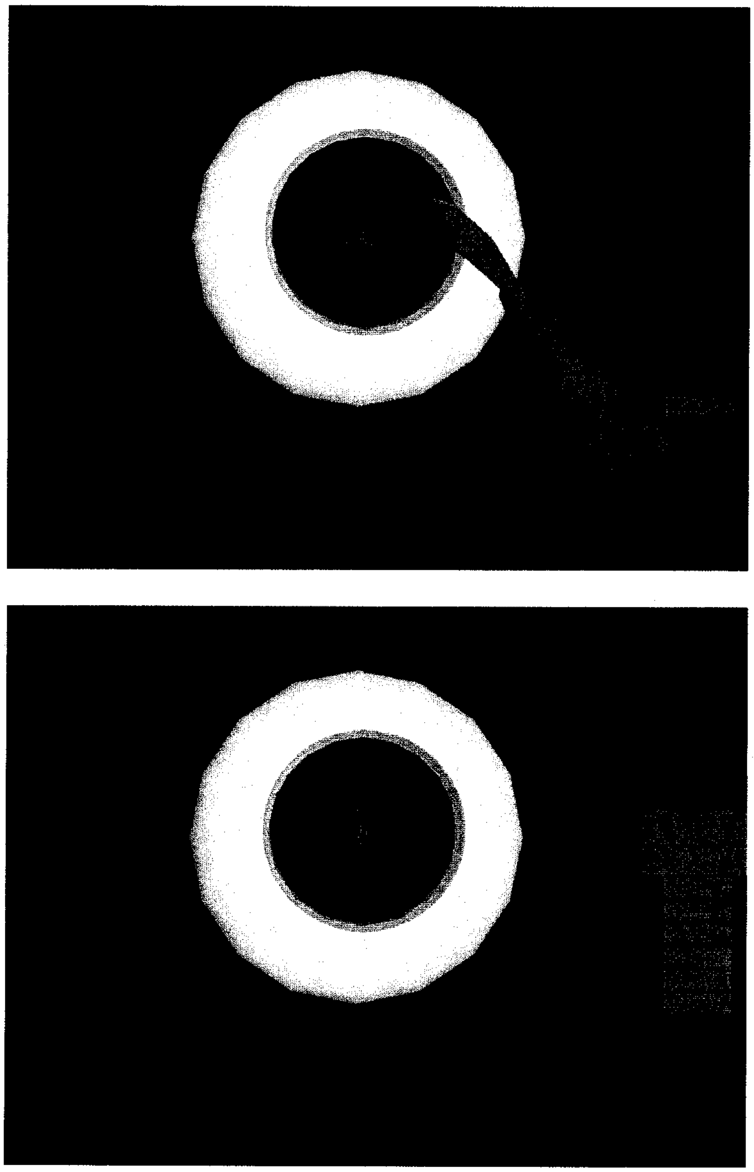


Figure 41 - 3D Immersive simulation screenshot (2D and inverted)

Chapter 5

Design and Implementation Using Reachin Display

This chapter is concerned with the process of building the application using Reachin API. The first section describes general design organization, followed by some hardware limitation present in Reachin Display, the hardware the application is intended to run on. Finally, the implementation of each step of the surgery is discussed.

5.1 General Design and Organization

5.1.1 Reachin API Revisited

In Chapter 3, we gave an overview of Reachin API. As our application is written using Reachin API, we will explain it in more details in this chapter.

A formal definition of Reachin API is given in their Programmer's Guide [38] and states : " The Reachin API is an API for defining the interacting with multi sensory three-dimensional virtual worlds. In particular, we target the co-location of graphics and haptics to interface with the primary senses of interaction.

More Concretely the Reachin API is a multi-sensory rendering engine that integrates visual and haptic rendering through the use of one single scene graph manager."

The definition is better visualized in Figure 42. [38]

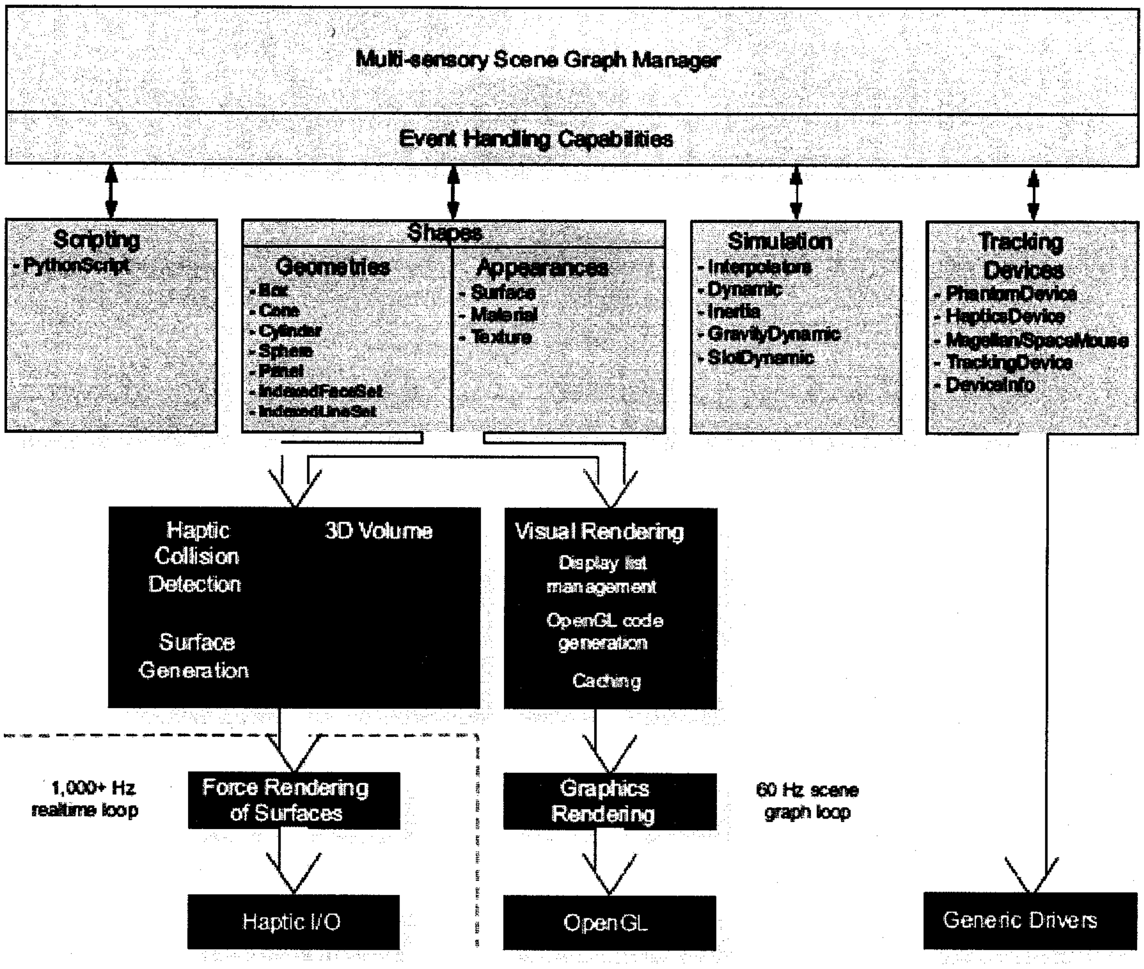


Figure 42 – A conceptual overview of the major modules present in the Reachin API

The Reachin API is based on the scene graph concept, which is a hierarchical data structure holding information about the 3D scene. The information includes geometry of the objects, their positions, their attributes, and certain information about the scene such as viewing positions and light sources. Scene graph is an abstraction layer over lower level graphics APIs such as OpenGL. There are two advantages of using scene graphs over lower level APIs:

1. Scene graph is declarative (what?) instead of procedural (how?). This allows for lesser complexity in coding.

2. Scene graph allows optimization in graphics rendering. This optimization is nearly impossible to get with low level APIs.

The top level in Figure 42 is the multi-sensory scene graph manager. It is responsible for maintaining the integrity of the scene all together. The multi-sensory scene encloses both haptic rendering and graphics rendering. Haptic rendering loops in a faster rate than graphics rendering and synchronization is achieved through multi-threading loops. The multi-sensory scene graph manager hides the aspect of multi-threading from the developer.

Event handling capabilities layer, located directly below the multi-sensory scene graph manager, is a framework that allows developers to manage different modules in the scene coherently and define how they should interact within the scene.

Beneath that is the module layer. Here, the modules of the scene is defined and allows the developer to build their scene from haptic devices to the graphics involved. There are a wide variety of modules available and the figure only shows some of the modules that can be used.

The final layer interfaces with low-level drivers and APIs such as the haptic drivers and OpenGL. It abstracts the programmer from the technicality of the rendering.

5.1.2 Cataract Surgery Simulation Architecture

Our design of the cataract surgery simulation architecture is broken down into three components: EyeTransform, Tools, and User Interface. Starting with the EyeTransform, it has four children as seen in Figure 43.

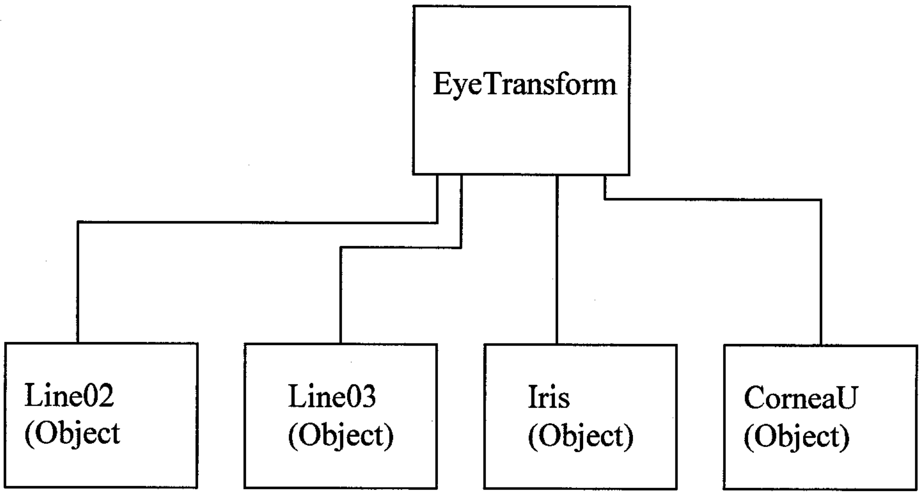


Figure 43 – EyeTransform component architecture

The EyeTransform class has four children corresponding to the VRML file describing the eye (refer to Chapter 4). This approach allows us to treat each of the eye parts as separate object, thus performing separate operations on each of the eye sub-organs as required.

Each sub-organ of the eye belongs to the Object class. The Object class is a class implemented to load each part of the eye and assign the required event handling properties that are appropriate. The object class architecture is shown in Figure 44. The Object Class is a subclass of the Collider group class provided by Reachin API. This class provides the method necessary to make the members of Object class collidable when interacting with other objects such as the virtual pen provided by the API. The Object class has four nodes under its main Transform node: Scale, Shape, Translation, and Rotation. Each node set the properties suggested by its name. The Shape node in turn has two children: Appearance and Geometry. The Geometry node is where all the haptic data is defined such as the elasticity upon collision. The Geometry node also set the conditions of the property of the graphics whether it is deformable, cutting enabled, or can be destroyed (erased). As it will be shown later this is one of the conditions that has to be set in order to perform a surgical operations such as cutting.

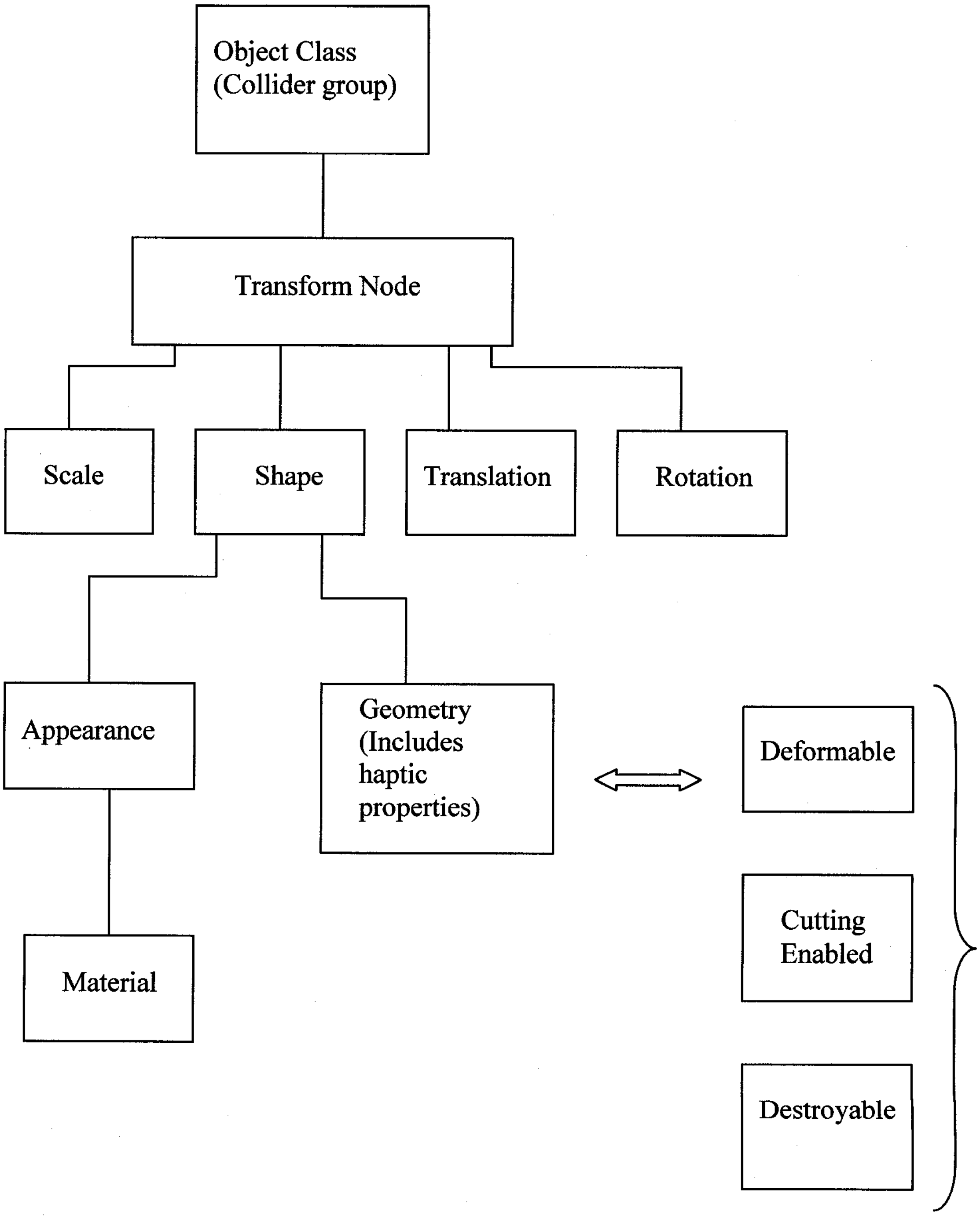


Figure 44 – Object Class Architecture

The second component in the architecture is the tools component shown in Figure 45.

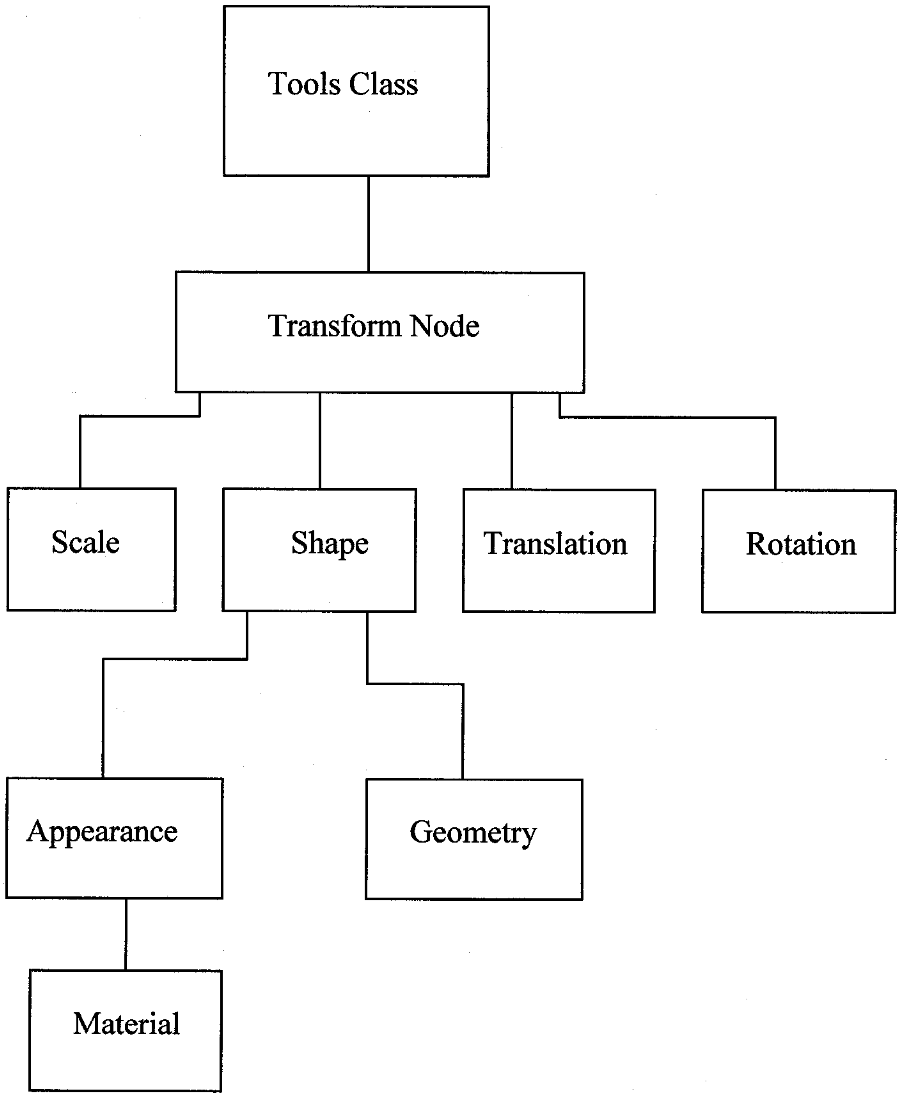


Figure 45 – Tools component architecture

The tools component is similar in architecture to the Object class. However it is not a Collidable class, meaning that there is no force feedback generated when the tools collide with another object. Nonetheless force feedback is produced automatically from Reachin API when the tools replace the Reachin's virtual pen which has a default force feedback at certain point of contact (the tip).

The third component of the architecture is the user interface. It consists of two components the Graphical Buttons displayed and the Button Tracker. Graphical buttons is what appear on the screen for the user to haptically control. The Button Tracker is the component that tracks the action on the side button of the Phantom haptic device. On the Phantom there is a button that can be pressed during the simulation, in which we programmed to perform miscellaneous functions. The miscellaneous functions include opening and closing the Forceps and replacing the lens, depending on the tool selected.

Our overall architecture can be summarized as shown in Figure 46.

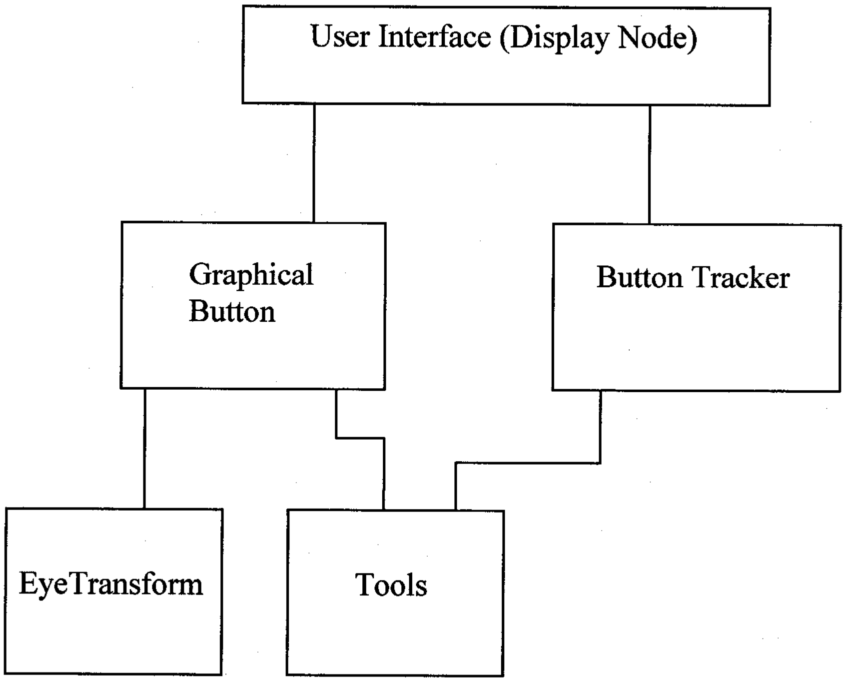


Figure 46 – Higher view architecture of the cataract surgery

The user interface resides on top of the architecture. It is linked to Reachin API's Display Node which starts the scene and the management between graphics and haptic rendering. The Button Tracker and Graphical Button are the components of the UI. The Button Tracker constantly checks if the button on the Phantom haptic device is pressed down and takes action depending on the Tool selected.

The Graphical Button is linked to the Tools and EyeTranform. It manages the other condition in which a procedure can be performed. As mentioned earlier, one condition is set through the Geometry Node of the Object class. The other condition is set upon selecting the Graphical Button. For example, cornea incision will be performed if the knife is selected and the knife touches the cornea (which has the cutting enabled property set to true in its geometry defintion).

5.2 Hardware Limitations

There were certain hardware limitations when programming our application. The cataract surgery is done with both hands in real life, usually with the left hand holding the eye with a forceps to limit its movement. On Reachin Display we had only one haptic device, and we could not simulate the functionality of the other hand. One work around is to make the eye more rigid as if standing still, this way we will mimic the functionality of holding the eye still. Another limitation is the involvement of specialized hardware used for the surgery that cannot be simulated by a haptic device. The most apparent example is in the phacoemulsification step. In order to activate the phaco tool, a footstep (shown in Figure 47) has to be pressed down. If this is to be incorporated into our application in the future, special hardware, acting as a footstep, should be placed below the person using the application in order to simulate the activation of the phaco tool when this hardware is pressed down.

Figure 47 – Phacoemuslification foot step

5.3 Design and Implementation of Surgery Steps

This section describes the specificities of major surgery steps.

5.3.1 Cornea Cutting

The cutting is activated when the keratome touches the cornea. Upon collision detection by Reachin API using the Device Info :getNumContacts () method, the algorithm in its core sense starts deleting the triangular vertices closer to the keratome proxy.

Calculating the closest vertices was based on the distance between the vertex and the keratome proxy. The algorithm generates a list of possible vertices that are close to the proxy and deletes the closest of these vertices and the corresponding triangle they form. There was a challenge of how many vertices we should delete depending on how wide we need the incision to be. We optimized that with a moderate size incision based on the surgery videos we have seen.

5.3.2 Eye Deformation

Eye deformation was set through Reachin API. In the API there are certain methods that can be used to set the properties of the geometry in terms of deformation. We chose values that resemble the eye physical properties closely in terms of elasticity and friction. Our reference was a plastic eye obtained from the University of Ottawa Eye Institute, which resembled, in terms of physical properties, the real human eye.

A snippet of the code used to set the elasticity is given in Figure 48.

```
// set the field values for the surface nodes
  my_surface->starting_friction->set(0.8);
  my_surface->stopping_friction->set(0.4);
  my_surface->dynamic_friction->set(0.6);
  my_surface->stiffness->set(300);
```

Figure 48 – Setting eye properties values

'my_surface' field refers to the surface of the eye for which the friction and stiffness values are set. The values were obtained through trial and error, until we were satisfied with force feedback from the eye deformation upon contact with surgical tools.

5.3.3 Lens Destruction

The lens destruction algorithm was activated upon collision of phacoemulsification tool with the eye lens. The algorithm calculates the closest vertices surrounding the point of contact and deletes the polygons in a circular area in an outward bound one by one, emulating the destruction of the lens.

5.3.4 Lens Replacement

After the destruction of the lens, the artificial lens should be replaced. When users select the intraocular lens from the list of tools, the button tracker takes a note of that, thus allowing the user to insert the new lens in the correct position when the button of the Phantom is pressed.

The following images is taken at the DISCOVER Lab, showing a user interacting with Reachin 3D application.

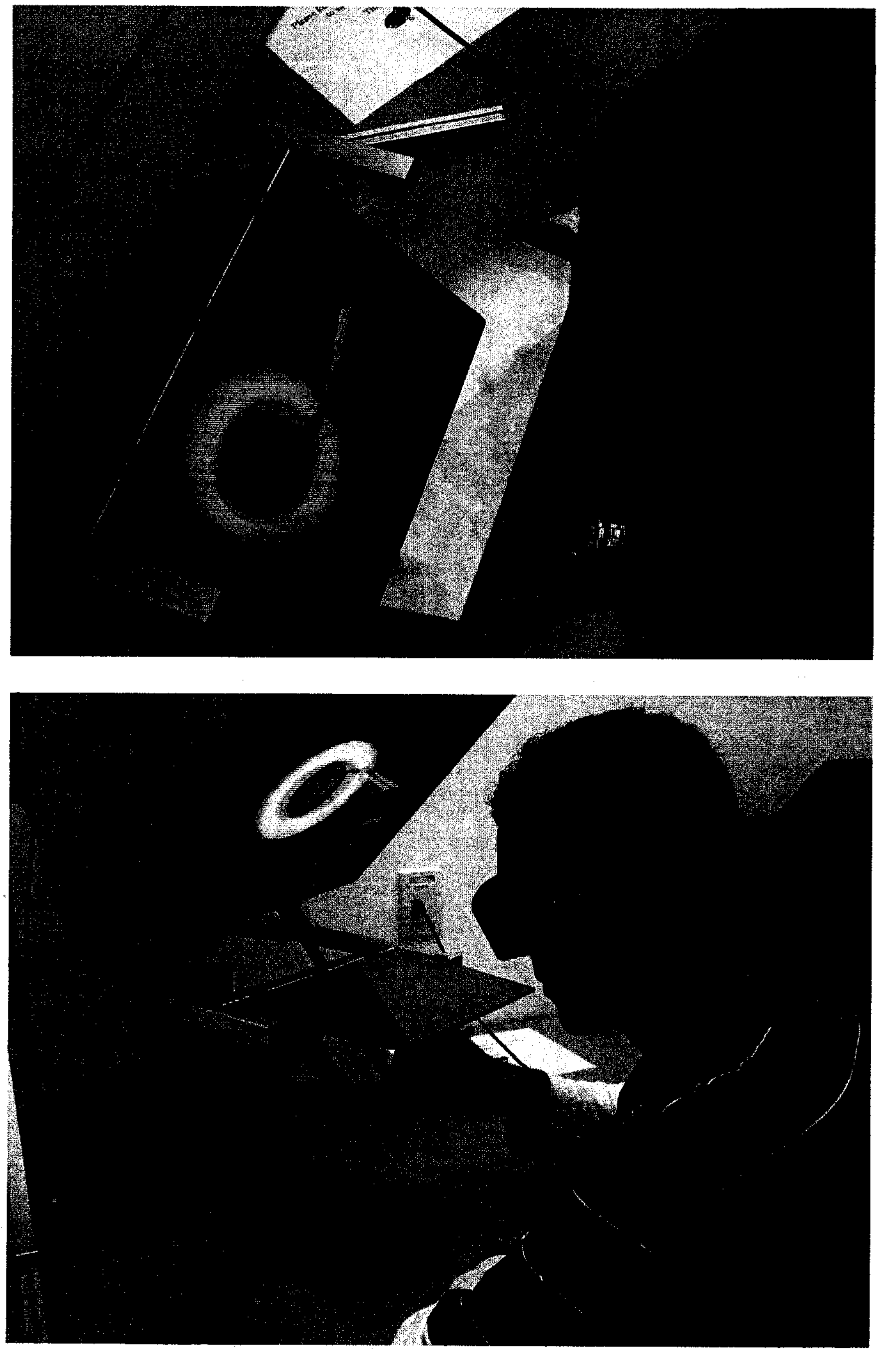


Figure 49 – User on Reachin Display going through the Cataract Eye Surgery Simulation

Chapter 6

Evaluating and Comparing the Application

At the DISCOVER Lab, we came up with three different implementations for cataract surgery. The purpose was to appeal to a wider range of audience depending on their viewing preference. Some users like 3D display, others lack the 3D vision and prefer a simpler 2D desktop interface, yet others like to be completely immersed in the environment and prefer the graphics to be as realistic as possible.

In this thesis, we went through the 3D interface application so far. In this chapter, we will go over two additional implementations done by other DISCOVER Lab members which are the 2D Desktop interface application and DIVINE Immersive interface application which is still being developed at the time of writing this thesis.

In this chapter also, we will mention the evaluation we conducted with eye surgeons and other personnel involved in the project. Their initial feedback is discussed, with their recommendations.

6.1 Description of Other Implementations

The other two applications that were developed at the DISCOVER Lab for simulating cataract surgery are the 2D Desktop Interface Application and the DIVINE Immersive Application. Currently, the 2D Desktop application is functional application with a tele-mentoring capability, while the DIVINE application displays immersive graphics of the eye model and the tools but there are no haptic data being rendered. The tele-mentoring feature of the 2D desktop application allows for remote training, when a surgeon is guiding a trainee in a different location. The following subsections describes the two applications, the description is taken from a paper we wrote entitled "A Distributed, Collaborative and Haptic-Enabled Eye Cataract Surgery Application with a User Interface on Desktop, Stereo Desktop, and Immersive Displays". [[39],[40]]

6.1.1 Desktop 2D Interface Application

The 2D surgical simulation was built on a framework for developing haptic-enabled surgery applications. This framework is being developed at the DISCOVER Lab and contains the several layers shown in Figure 50. The geometrical models of the organs and surgical tools are at the bottom of the hierarchy. They are used as the basis of the virtual environment that is created and managed by the Scene Graph Layer. The dynamics of the scene are controlled by the Physics Layer which assures that the interaction of the objects in the scene conforms to the laws of physics [41]. The inputs to the Physics Layer that ultimately result in the scene interactions are generated by the Control Layer. These interactions are usually triggered by the inputs received from the input devices or higher layers. The Application Layer is responsible for the overall flow of the simulation. It is here that the implementation of the components is dependant on the type of surgery that is being simulated. Finally, the Configuration Layer is strategically located on top of all others in order to have a clear view of the overall operation. It is responsible for general tasks such as recording a simulation session or reconfiguring the general parameters of the application.

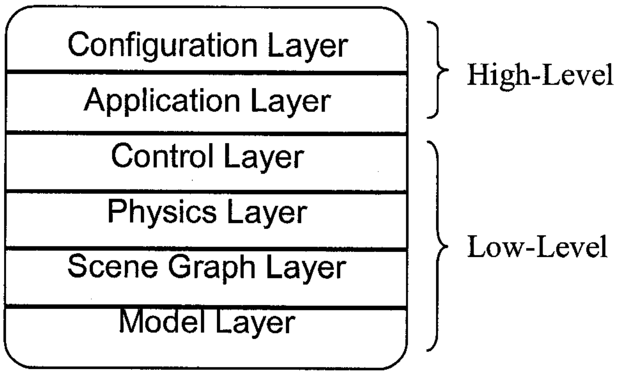


Figure 50- Layered Architecture of proposed framework

Each of the design layers presented above consist of one or more key components each with their own set of functionalities. Figure 51 shows these components along with the general inter-component data flow. The 3D models needed for the simulations (such as organs and surgery tools) are usually stored in some ASCII or binary format such as VRML or 3DS. The model loader component loads these models, converts them to scene-graph and sends them to the graphics rendering component for display.

In addition to rendering, the graphics rendering component makes these models available to the Physics and Collision Engines. Instead of manipulating the graphics directly, the higher layer components issue high-level commands to the physics engine for moving or manipulating the models. The physics engine, in coordination with the collision engine, generates physically realistic results that correspond to the incoming high-level commands.

The components at the control layer are in charge of controlling the virtual scene by issuing commands to the lower layers based on the requested tasks. The *remote client* for example is in charge of synchronizing the scene so that the remote and client users perceive a same simulation in a distributed application. The *tools manager* is specific to surgical applications and it provides a convenient means of managing the collection of tools to be used in the surgery procedure. The *haptic control* is a generic component for all haptic devices. It continuously receives inputs from whichever haptic device assigned to it; it will then translate those inputs into high-level commands that can be used for variety of tasks such as controlling the position of a surgical tool. Specifically for this application the haptic control uses either OMNI or MPB haptic devices. The choice of which haptic device to use can be simply set through the configuration component. The remote client component uses a TCP connection for graphics data (to assure consistency) and UDP for haptic data (for higher data rate).

The *simulation* component is the brain of the simulation, as it utilizes all other components in order to simulate a particular surgery procedure. It is therefore highly customizable and extensible by the application developers. The *user interface* and *training* components are also highly customizable but they additionally offer a rich set of features that the application developers can choose from instead implementing their own.

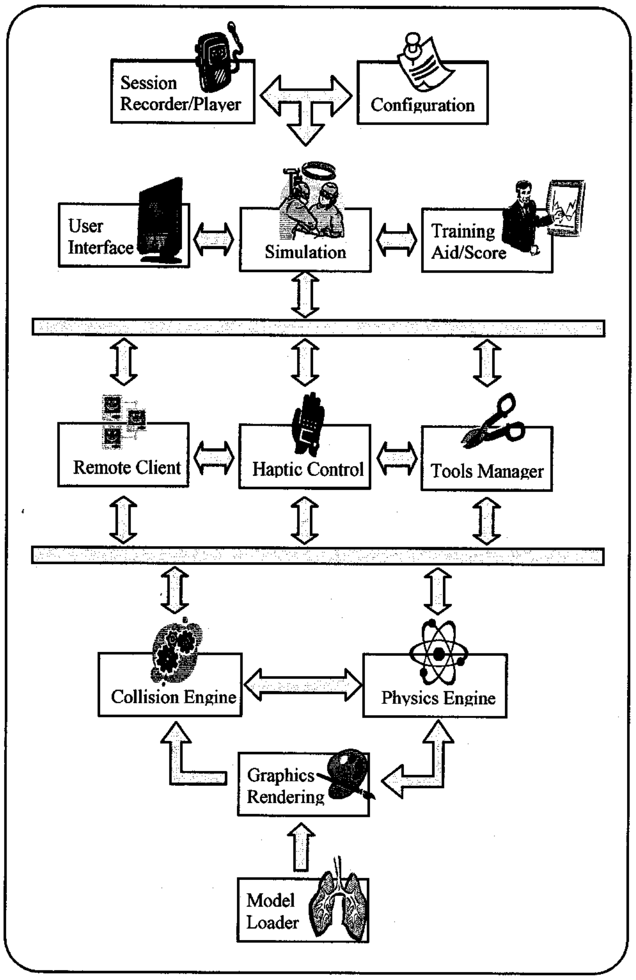


Figure 51- Major components of the proposed framework

Most surgical simulations and training applications require that the sessions be recorded for later analysis. This is particularly useful in training applications when the trainer might want to go over the simulation again in order to pinpoint the trainee's mistakes. The *session recorder* component is in charge of the task of recording and replaying simulations. The *configuration* component reads script files that outline general configuration settings that are needed by the various components in the framework.

For the cataract surgery simulation our implementation uses a high resolution model of the eye and several specialized surgical tools. These models are loaded by the model loader and rendered accordingly (Figure 52). The xPheve physics engine [41] is used for realistic cutting and deformation effects. The collision engine used for this application uses a simple algorithm to detect collision between a line segment (surgical tools are represented by a line segment) and polygons. The haptic control uses either OMNI or MPB haptic devices. The choice of which haptic device to use can be simply set through the configuration component. The remote client component uses a TCP connection for graphics data (to assure consistency) and UDP for haptic data (for higher data rate).

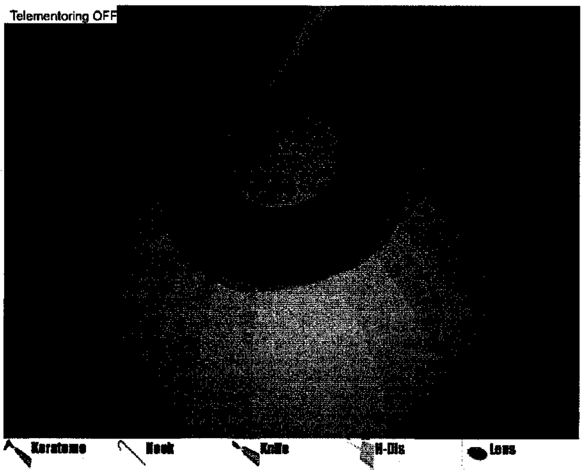


Figure 52 – 2D Desktop Interface screenshot

6.1.2 DIVINE Immersive Application

Immersion can be defined as the state of deep and natural involvement a user experiences when interacting within some space. A computer graphic display is said to be immersive when it is rendering on its display mediums, specifically for some user's view, in a way to make him or her perceive some virtual space as part of the real physical space. In other words, if an immersive display were to show a coffee cup on a tabletop display medium, then that coffee cup should remain in the same place on that tabletop no matter the viewer's position and angle of sight, exactly as it would, had there been a real coffee cup on that tabletop. To achieve this coherency, the display platform has to account for the user's movement (head and body, position and rotation) and display the graphics in such a way that the displayed objects look as if they were physically there and viewed from a given angle and position. The most common Immersive Projection Technology (IPT) setup is the CAVE (Cave Automatic Virtual Environment) system. Typically, a CAVE system constitutes of several large display screens to fill as much of a user's field of view as possible. With video projectors, computer systems that can synchronously display in stereo on these multiple screens, a 3D position and orientation sensor, and stereo glasses complete a classic CAVE setup. By entirely surrounding the user with viewpoint specific stereo rendering, "full" immersion can be achieved whereas all the perceived physical space is virtual space. This full immersion is very difficult to achieve by reason of its requirements. The integration of a multitude of technologies, all with their own practical limits, imposes severe restrictions in the end product, with a price tag exponentially proportional to the size and number of screens.

In contrast, our DIVINE (Desktop-Immersive Virtual and Interactive Networked Environment) system allows the display of immersive computer graphics on a desktop-like workspace more appropriate for close interaction, and alleviates some of the problems related to a fully immersive CAVE. A major advantage making it suitable for finer use is the greater pixel density offered by concentrating the video projection on smaller and closer screens. Also of significance is its better affordability and its much smaller space requirements.

Worthy of noting here is that DIVINE was custom designed from the ground up by DISCOVER lab members and was built by a local company. It was designed to be versatile in its use. Its main purpose is to provide an immersive visualization experience for two users (one per corner unit) and allow them to directly interact within the projected space. For maximum immersive effect, the video projection had to be done in stereo, and a position/orientation tracker integrated in the system. For the users to be able to extend their arms and move their head in the projected space (for natural manipulation and visualization), three orthogonal rear-projection screens were used: a table at waist height, and two walls to form a corner. Another consideration in the design was the need for a well lit environment for video-capture of the user (not used in the context of this application). Placement for the cameras that would provide some good viewing angles of the user without interfering with the immersive projection was another constraint. Furthermore, only a limited space was available for such a system.

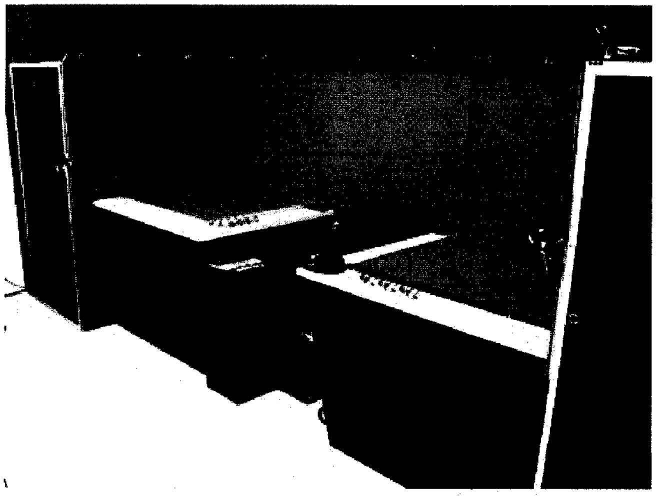


Figure 53 – The DIVINE system at the University of Ottawa Discover Lab

Considering all these requirements, DIVINE is composed of two corner visualization volumes and one central projection screen. Figure 53 shows the hardware setup. This central screen can combine the two corners into a long, almost seamless virtual window. Each screen projection in the system is driven by one workstation and two LCD projectors. Circular polarization filters in front of each projector and non-depolarizing screen materials and first-surface mirrors were used to provide the stereo projection capability.

The workstations are dual-Xeon based, with nVIDIA Quadro FX 3000 video cards, capable of 3D stereo on independent left and right outputs. Projectors are SANYO PLC-XT16 with the wide-angle lens option and network controller. They are stacked in left and right pairs using an extruded aluminum structure. The circular polarizing filters (3M HNCP37% x0.30") left and right were purchased in large sheets and cut to 7" squares. They are placed ~5" from the lens in front of each projector using simple metal brackets and paper clips. This distance is required to prevent the filter's plastic from melting because of the considerable heat generated by the projectors. The screen material used is STEWART filmscreen 150 on 1/4" plastic backing. Each screen is 4'x3' (for a 1024x768 resolution's 4/3 aspect ratio). The mirrors are DaLite PFS 90% reflective mylar mirrors.

Each corner unit can be used independently. Having two usable IPT in this way allows us to experiment with collaborative virtual reality applications in a relatively small space. Collaboration can be directly observed by third parties by having the two systems side-by-side in the same room.

For rendering 3D graphics on an IPT, each of its constituent screens must display the proper view projection of the graphics relative to the eyes of the user (approximated by using a 3D tracker attached to his/her head). Several software packages propose to do this IPT rendering.. In this work, we explore the use and limitations of some of these offerings in the context of our eye cataract surgery application, having “mixed and matched” software and hardware combinations in order to come up with a generic framework that we could use to run our 3D applications in an immersive environment, using different display platforms (e.g. desktop, HMD, CAVE, DIVINE, etc) with no need to recompile our code. Some of the software packages that we have studied include: For networking and cluster management, VRCO CAVElib [42] and VRJuggler [43]; For graphics rendering, SGI OpenGL Performer [44], OpenSceneGraph [45], Virtools Dev VR authoring software [46] with Physics pack and VR pack, and basic OpenGL[47]; And for tracking using our Ascension Flock of Birds tracker, VRCO trackd [48], and the University of North Carolina’s VRPN [49].

After several trial-and-error runs, we built a programming framework that allows any 3D application to run on DIVINE provided that the graphics are rendered by OpenSceneGraph, the physical tracking be done by VRCO trackd, and the networking/cluster-management be done by VRCO CAVElib.

The above combination was arrived at in stages: First of all, we favored VRJuggler to manage our computer cluster mainly because it was open-source and supported OpenSceneGraph, the library we were using for our graphics programming. However, we could not translate DIVINE’s physical coordinate system to that of VRJuggler due to a bug in VRJuggler so we switched to CAVElib. VRCO’s trackd works well with CAVElib so we chose that as our tracking software. We had an issue getting OpenSceneGraph to call CAVElib functions, but we were able to solve this problem.

The following figure shows the eye and surgical tools loaded into the DIVINE system. Note however that it is hard to catch the immersive graphics on a still photo.

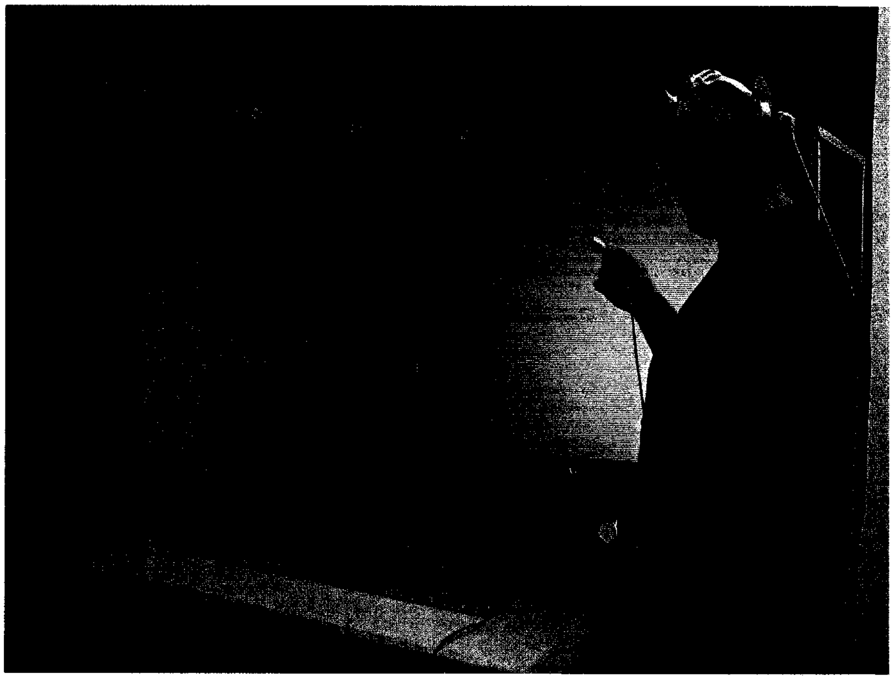


Figure 54 – Eye and surgical tools loaded into the DIVINE system

6.2 Application Testing

We have tested the application with different classes of users. Our intention was to get the feedback from different perspectives in order to evaluate our three implementations and see how they relate to each perception. This section deals with the statistics and findings of the testing we have conducted, while the next section discuss how different classes of users adapted to the application.

6.2.1 Use Cases

Before beginning the testing, users were invited to the DISCOVER lab, and they were introduced to the various haptic devices we have, and especially the haptic devices used for the surgery. An overview of what we are simulating is explained, and in particular our goal of the application is stated. We explain our three implementations and what the advantages of each are.

Our use case consisted of users conducting the surgery steps on the 2D Desktop interface application and the 3D Immersive application on Reachin. Users also used the DIVINE system to view the immersive eye and surgical tools, and manipulate the position of the eye while trying to select different tools. The position of the eye is relevant to the user and it can increase the immersion of the eye, by making it "pop-up" into space, according to the users' preference.

Users are reminded to take certain points into consideration while performing the surgery among these points are:

- How realistic is the eye model, and which application is more appealing to the user?
- How realistic are the surgical tools?
- How easy is it to maneuver in the environment and interact with the user interface (such as switching between the tools)?
- Is it intuitive to manipulate the tools with stylus-based haptic device?
- How does the eye cutting and deformation relate to the user?
- Is the tele-mentoring feature an effective learning tool?

- Do 3D graphics add more value to the application, and if so, are 3D graphics sufficient or are they better off as immersive graphics?

After users went through the testing procedure they are asked to fill a questionnaire. The questionnaire take into account all the above points but with greater details. The questionnaire is included in Appendix A – Questionnaire Form Used to Evaluate Cataract Surgery Simulations.

Often there were instances when the user case had to be modified. This happens when users cannot make it to our lab. Instead we have to go to their location and carry our equipment. By doing so, the only platform we can perform the testing on is the 2D Desktop application as it is the only application that is portable. The other two applications cannot be moved due to the bulk size of the hardware involved namely: the Reachin Display, and the DIVINE system. Moreover, the licensing system of the Reachin API does not permit us to use the software outside the DISCOVER local area network (LAN).

In that case, users only test one interface of cataract surgery simulation; the 2D Desktop interface. They are notified that there are other applications in development in the DISCOVER lab and that they are welcome to come and see them sometime. When they fill out the questionnaire, they only fill the relevant questions concerning the 2D Desktop application and they disregard the other questions that concern Reachin display application or the DIVINE application.

As a result, we evaluated the 2D Desktop application more frequently than the other two. However we were able to get enough feedback for all three, particularly from medical doctors, which are our intended audience to begin with.

6.2.2 Application Statistics

From the questionnaire in Appendix A we gathered statistics based on some of the questions provided. The values provided in the charts below are the average values of all the evaluations we have obtained. We averaged all values obtained from all users. There were 15 users in total in which the majority were medical residents. The Diagrams below are divided into sections that correspond to the sections in the questionnaire.

The diagram in Figure 55 deals with the first section of the questionnaire. The question is how well does the haptic device stylus represent the *tools* handle? The diagram shows the average value of each tool, where five is the highest score.

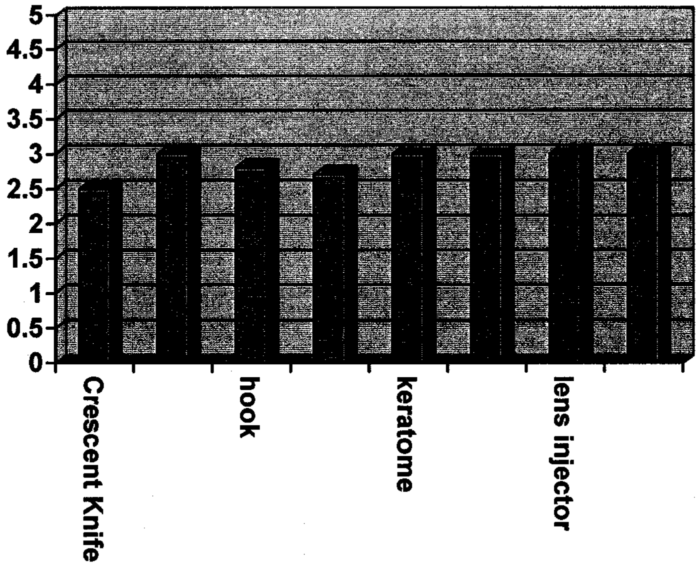


Figure 55 – Section 1: Haptic Input Statistics

Figure 56 and Figure 57 are related to section two of the questionnaire; they are related to the Graphical User Interface (GUI). Figure 56 compares the three applications based on the easiness of switching the tools, while Figure 57 compare the three applications in terms of how realistic the graphics are.

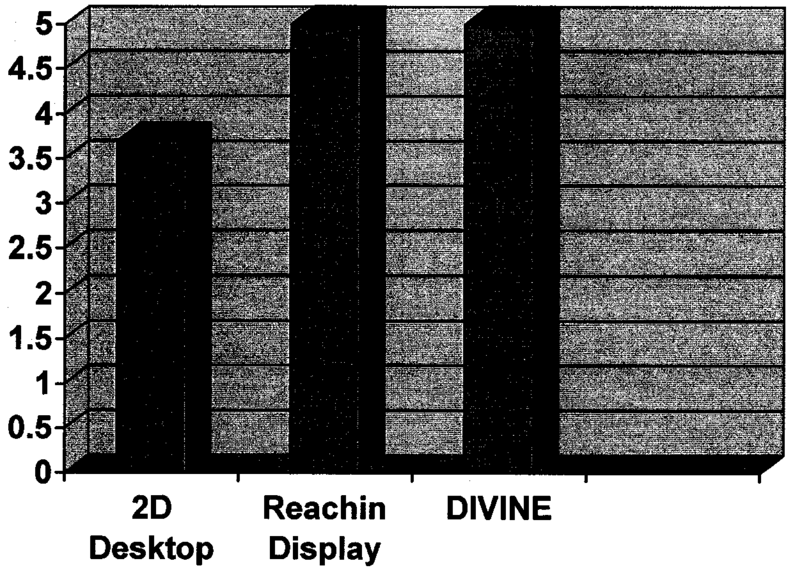


Figure 56 – Section 2: GUI Statistics: Easiness of switching between tools

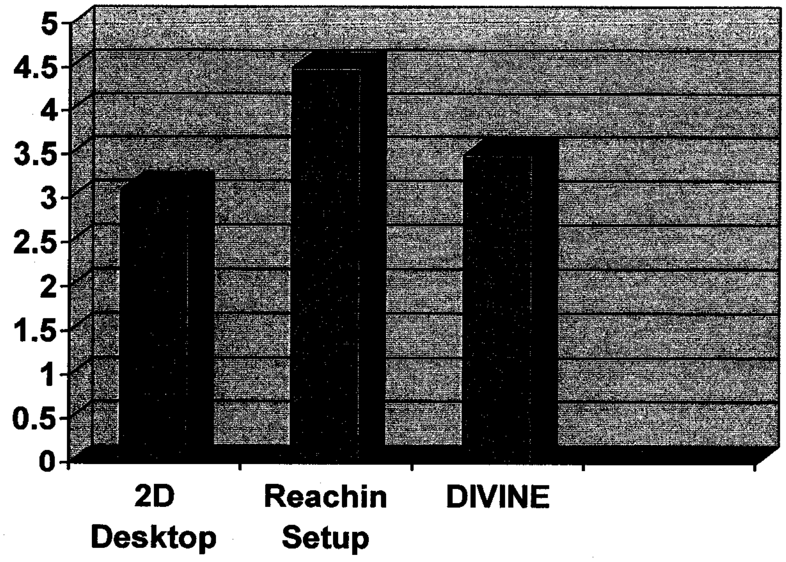


Figure 57 – Section 2: GUI Statistics: realism of the graphics displayed

The next figure refers to section three of the questionnaire. Figure 58 shows the comparison of the haptic feedback of both applications: 2D Desktop and Reachin 3D interface (DIVINE application is not setup to give haptic feedback yet). Note that tissue pulling is not setup in the 3D application that is why it is omitted from the figure.

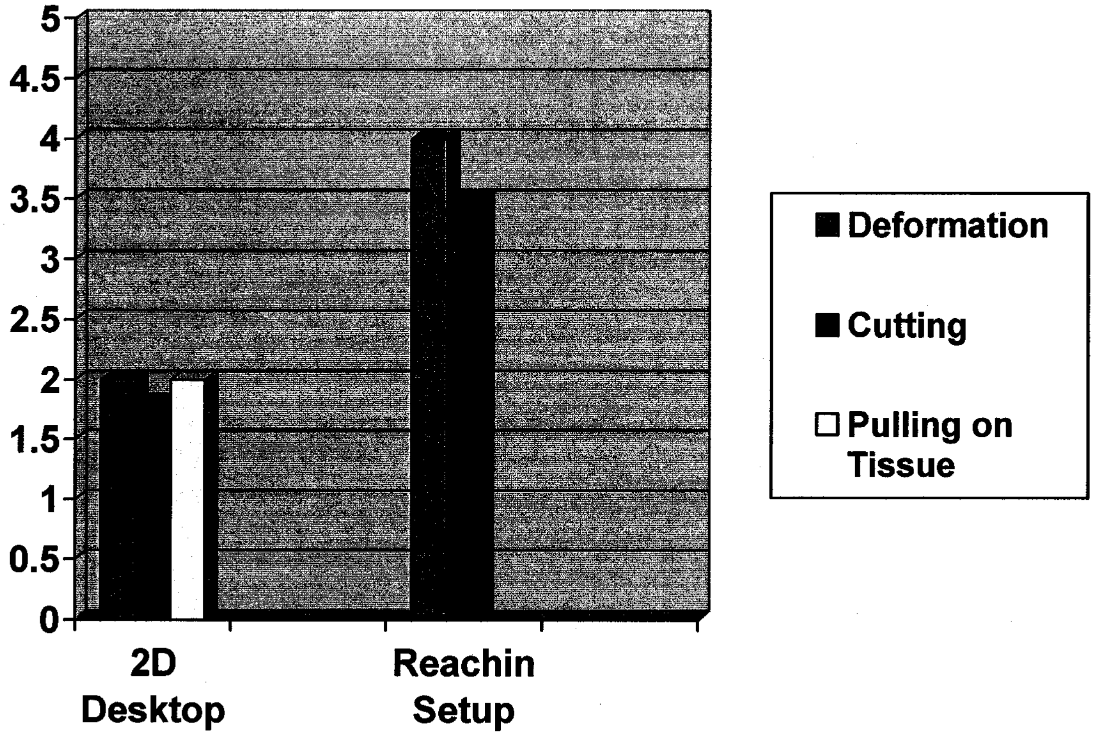


Figure 58 - Section 3: Haptic Feedback statistics

Other evaluation parameters were calculated and the average is given as the following:

- Section 4: Tele-Mentoring: The tele-mentoring feature averaged 3.5/5
- Section 5: Overall application
  - The overall application's value as a learning tool was 2.9/5
  - The realism of the application was 2.7/5

6.2.3 Performance Evaluation

When the applications were running locally, they were running normally with no apparent delay. We needed to measure the performance over the network for the tele-mentoring feature. The first test we did was running the 2D Desktop interface application with the tele-mentoring feature in the local area network. There was an apparent delay that was due to a buffering problem and one colleague working on the project fixed it. After fixing the problem, we wanted to test the performance over a dedicated network, such as the CANARIE network. We went to the Communications Research Center which is linked to our lab through CANARIE connection and tested the application over there. We did not do an objective test, but we did a subjective one by running the application on both sides and communicating results over the telephone. We deduced that the performance of the application is acceptable and adheres to the standard we have set. The delay was minimal.

6.3 User Adaptability Testing

This section is an acknowledgment of the feedback we got from different users. The idea of having several classes of users test the application is to determine how people with different background in terms of haptic technology and medical sciences adapt to the application. We, as researcher in the DISCOVER Lab, are used to dealing with haptic devices and we work with them on a regular term. This makes us biased toward the devices and we needed opinion from people novice to this technology.

Medical doctors who are experienced in doing the cataract surgery are biased from a medical point of view as they did the surgery many times and they are extremely familiar with the steps and the tools.

We wanted the application to address different users, which is why we performed the testing with users that have different backgrounds. However we always kept in mind that our target users were medical residents who have a background on the cataract surgery but did not practice it much.

One feedback we got was from a health researcher who did not have a background in haptics or cataract surgery. She adapted well to the haptic devices and agreed that the force feedback was interesting and feels great however she had no point of reference to compare it to a real cataract surgery and couldn't answer whether the simulation was realistic or not. She suggested having a beginner's demo version that has an instruction box in the corner for first time users, e.g. guide the user to pick up a tool and perform the cutting.

Another feedback we got was from an experienced eye surgeon at the University of Ottawa Eye Institute. Dr. Andrew Merkur visited our lab and tested the three applications. He found the applications to be very helpful as a learning tool. He suggested minor modifications to the system. He said some tools need to be oriented correctly such that they are easier to manipulate by the haptic devices. For example, the Keratome needs to be adjusted so the first cut will be at the bottom of the cornea not the left or right side.

Some tools were needed for the core operations of the surgery, while others were needed for enhancement steps. For instance, the hook tool is not mainly used, but can be used if the path to the eye lens was obscure and the iris needs to be expanded. It is preferable to remove those tools so that they will not cause confusion.

As far as the anatomy goes, there was a slight problem with the lens rigidity. The lens was loose and stretches, while in reality it is supposed to be fixed and breaks easily when touched. In the 2D application the orientation of the eye needs to be adjusted. If that is difficult to achieve, shadowing can be added to help with the orientation.

The doctor said that the applications are very impressive. He found the haptic device to be appropriate for most of the steps of the surgery but he hinted that for phacoemulsification maybe specialized hardware would be more appropriate. He found that the 3D immersive application is more realistic, but the 2D application when modified will better suite our goals as a training application, since the 3D vision can be deceiving and lacking in some people.

Based on his recommendation and help we were able to setup a testing session at the Eye Institute. We went there and ophthalmic surgery residents had the chance to test the 2D Desktop application. Some residents were used to handling surgery tools, but they had difficulty in manipulating the haptic device. They did not find it intuitive at the beginning as they were not familiar with this technology. But as they worked with the devices for a longer time, they got the feel for manipulating the virtual environment. They suggested more time for orientation for residents who intend to use this application. They saw value in the application and said in the future it will be exciting to train on the VR application as opposed to animal eyes.

6.4 Survey Results

As mentioned earlier, more testing has been done on the 2D Desktop application. The results obtained do not necessary reflect that bias in testing. In general, however, 3D environment seemed more appealing to users than 2D environment, which is reflected in the averages obtained from the survey. The next step is to weigh these scores with practicality, as the 2D Desktop interface is more practical application, due to its less stringent requirements for special display.

All of our scores were above average, which suggests the value of the applications. There are still areas to improve on, but with future modification, and more haptic awareness higher results will be achieved.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Cataract surgery simulation demonstrated to be a promising application. The reviews from surgeons were positive and they concurred that this could be a first step toward a prosperous future for medical training.

This thesis provided a detailed walkthrough in the development of a 3D immersive cataract surgery. In Chapter 2 and Chapter 3, we have described the background material for the medical cataract occurrence and a step by step description of the medical surgery that relieves the patient from the disease. We also provided a technological background on the rise of VR reality applications and the inclusion of haptic devices in the applications. An overview of the project we are working on in the DISCOVER lab was also given.

In Chapter 4, we have reviewed our design of the 3D objects required for the surgery; the eye organ and the surgical tools. The challenges we faced during the modeling and decisions we undertook was explained. Screenshots of our final models were presented. We gave a tour of our 3D environment and the rational behind the design.

In Chapter 5, we went into the specificities and details of our architecture of the 3D simulation utilizing the Reachin Display and API. We also brushed on the functionality of the surgical steps implemented and which steps needs more refinements in the future.

Chapter 6 started by giving an overview of other efforts being conducted at the DISCOVER lab. As there are three applications for cataract surgery simulation being built in parallel, we compared the three and listed the advantages of each. We summarized the results of our usability testing, and addressed the feedback given by medical doctors and other professionals in the industry.

7.2 Future Work

There are certain modifications that can be done in the future. The peripheral steps omitted during the design can be implemented to add authenticity to the application. We could start from the sedation process and then end with the remedies applied after the surgery. We can go further and design a human face and include our eye model within the face to make it more realistic.

The hardware limitations discussed in this thesis should be addressed. However, obtaining customized hardware for the surgery can be expensive. Weighing the advantages of the additional hardware against the cost disadvantage can be measured and a decision can be made based on that.

This application is geared to be a training guide for novice surgeons. If the medical community decides to change the direction of training methods, the application should be extremely reliable and trustworthy. This implies constant maintenance of the code as well as constant feedback from experienced surgeons and modification to the application to include their suggestions. Complications that may arise during the surgery can be incorporated into the application. Finally it would be nice to have a scoring system to measure how efficiently users perform the virtual surgery. The scoring system can be also used to monitor improvement over time.

The future looks promising, and we hope we can take our application another step further.

Appendix A – Questionnaire Form Used to Evaluate Cataract Surgery Simulations

Usability Questionnaire

Thank you for helping us in assessing the usability of our eye cataract surgery application.

This questionnaire is made up of six sections covering different aspects of the application. We estimate that this survey will take you 8 – 12 minutes to complete.

If at any time, you have a question then please do not hesitate to ask.

Thank you again.

Section 1: Haptic Input

1. Have you ever handled eye cataract surgery tools before?

Yes No *If your answer is "No," please proceed to Question 11.*

2. How well does the haptic device stylus represent the *crescent knife's* handle?

Not at All				*Exactly*
1	2	3	4	5

Comment:

3. How well does the haptic device stylus represent the *forceps'* handle?

Not at All				*Exactly*
1	2	3	4	5

Comment:

4. How well does the haptic device stylus represent the *hook's* handle?

Not at All				*Exactly*
1	2	3	4	5

Comment:

5. How well does the haptic device stylus represent the *hydrodissection tool's* handle?

Not at All *Exactly*

1 2 3 4 5

Comment:

6. How well does the haptic device stylus represent the *keratome's* handle?

Not at All *Exactly*

1 2 3 4 5

Comment:

7. How well does the haptic device stylus represent the *lens*?

Not at All *Exactly*

1 2 3 4 5

Comment:

8. How well does the haptic device stylus represent the *lens injector's* handle?

Not at All *Exactly*

1 2 3 4 5

Comment:

9. How well does the haptic device stylus represent the *phacoemulsification tool's* handle?

Not at All *Exactly*

1 2 3 4 5

Comment:

10. How well does the haptic device stylus represent the *syringe*?

Not at All *Exactly*

1 2 3 4 5

Comment:

Section 2: Graphical User Interface

11. How easy is it to switch between surgical tools on the *2D Desktop* setup of the application?

Difficult *Easy*

1 2 3 4 5 N/A

Comment:

12. How easy is it to switch between surgical tools on the *Reachin Display* setup of the application?

Difficult *Easy*

1 2 3 4 5 N/A

Comment:

13. How easy is it to switch between surgical tools on the *DIVINE* setup of the application?

Difficult *Easy*

1 2 3 4 5 N/A

Comment:

14. How realistic are the graphics displayed on the *2D desktop*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

15. How realistic are the graphics displayed on the *Reachin Setup*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

16. How realistic are the graphics displayed on the *DIVINE Setup*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

17. Which graphic output setup do you think best portrays the *anatomy of the eye*?

2D-Desktop Setup Reachin Setup DIVINE Setup

Comment:

18. Which graphic output setup do you think best simulates *real-life cataract surgery*?

2D-Desktop Setup Reachin Setup DIVINE Setup

Comment:

Section 3: Haptic Feedback

19. Have you ever performed eye cataract surgery on humans before?

Yes No *If your answer is "No," please proceed to Question 26.*

20. On the *2D Desktop,* how realistic was the haptic feedback you felt when performing *touching and pushing (deforming)*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

21. On the *2D Desktop,* how realistic was the haptic feedback you felt when performing *cutting*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

22. On the *2D Desktop,* how realistic was the haptic feedback you felt when *pulling on tissue*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

23. On the *Reachin Setup,* how realistic was the haptic feedback you felt when performing *touching and pushing (deforming)*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

24. On the *Reachin Setup,* how realistic was the haptic feedback you felt when performing *cutting*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

25. On the *Reachin Setup,* how realistic was the haptic feedback you felt when *pulling on tissue*?

Not at all Realistic *Just like Reality*

1 2 3 4 5 N/A

Comment:

Section 4: Tele-Mentoring

26. How do you rate the *tele-mentoring feature*?

Not at all Helpful *Extremely Helpful*
1 2 3 4 5 N/A

Comment:

Section 5: Overall Application

27. How do you rate the overall application's value as a learning tool?

Not at all Helpful *Extremely Helpful*
1 2 3 4 5 N/A

Comment:

28. How do you rate the realism of the application?

Not at all Realistic *Just like Reality*
1 2 3 4 5 N/A

Comment:

29. Do you think perceptual cues are appropriate to medical training?

30. In your opinion , which means fits the demands of medical training, such as telementoring, perceptual cues and others?

Section 6: Background Information

29. How familiar are you with haptic devices? (Please circle one)

1 What is a haptic device?
2 Familiar with them but never used them
3 Have used them before but only once or twice
4 Use them regularly
5 Have programming and/or engineering knowledge of them

30. How familiar are you with eye cataract surgery? (Please circle one)

1 What is eye cataract surgery?
2 I know what it is but have no knowledge of the procedure
3 Familiar with the procedure but not its details
4 Know the procedure in detail
5 Performed the procedure before (training and/or real surgery)

31. Age: ____ (optional)

32. Gender: Male Female (optional)

33. Profession: ______________________________

Other comments:

References

[1] G. Welch, A. State, A. Ilie, K. Low, A. Lastra, B. Cairns, H. Towles, H. Fuchs, *Immersive Electronic Books for Surgical Training, IEEE Multimedia*, July-September 2005, pp 22-35

[2] J. Luursema, P. Kommers, *A Surgical Virtual Learning Environment*, unpublished manuscript, available at http://www.gw.utwente.nl/cpe/medewerkers_cpe/luursema_j_m, 2004.

[3] G. Aloisio, L. T. De Paolis, A. De Mauro & A. Zuccalà, *Surgical Training System with a Web Portal for Virtual Environment Searching*, Modelling in Medicine and Biology VI, 2005.

[4] S. El-Defrawy, *Basic and Clinical Science of Cataract*, University of Ottawa Eye Institute, Basic Science Course Presentation, 2001.

[5] *History of Cataract Surgery*, The Foundation of the American Academy of Ophthalmology Website, http://www.eyecareamerica.org/eyecare/museum/exhibits/online/cataract , accessed Nov, 2006.

[6] P. Wysong, S. Henahan, *Virtual surgery replaces traditional wetlab courses*, available at http://www.meleritmedical.se/html/html_news/pdf/virtualsurgery.pdf, accessed Dec. 8, 2006.

[7] JV. Rossi, D. Verma, GY. Fujii, RR. Lakhanpal, SL. Wu, Ms. Humayun, *Virtual Vitreoretinal Surgical Simulator as a Training Tool*, Retina the Journal of Retinal and Vitreous Diseases, 2004.

[8] A. Fareed, *Surgical Simulator Helps Train Young Surgeons*, available at http://www.retinalphysician.com/article.aspx?article=100203, accessed Dec. 8, 2006.

[9] M. Davis, *Theoretical Foundations for Experiential Systems Design*, Proc. ACM MultiMedia 2003 Workshop on Experiential Telepresence (ETP 2003), ACM Press, 2003, pp. 45-52.

[10] I.E, Sutherland, *The Ultimate Display*, in Proceedings of IFIPS Congress, vol. 2, pp 506 – 508, May 1965.

[11] G.C. Burdea, P. Coiffet, *Virtual Reality Technology*, Second Edition, John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2003.

[12] G. Burdea, Virtual Reality Systems and Applictions [short course], in Electro '93 International Conference, Edison, NJ.

[13] A Brief History of Virtual Reality, http://www.bilawchuk.com/mark/history.html, accessed Dec. 13, 2006

[14] Supported Education Infrastructure Project Website, Virtual Reality Homepage, http://ei.cs.vt.edu/~history/Tate.VR.html, accessed Dec. 13, 2006

[15] N.R. El-Far, *UNISON: a Generic Framework for Hapto-Visual Application Development*, University of Ottawa Master's thesis, 2004.

[16] Sensable Technologies Website, Phantom Omni Homepage, http://www.sensable.com/haptic-phantom-omni.htm, accessed Dec. 13, 2006.

[17] S. Andrews, J. Mora, J. Lang, W. Lee, *HaptiCast : A Physically-Based 3D Game with Haptic Feedback*, Future Play, 2006

[18] M. Eid, M. Orozco, A. El Saddik, *A Guided Tour in Haptic Audio Visual Environment and Applications*, Accepted for publication in the International Journal of Advanced Media and Communication, to appear in vol.1, n.3, 2007.

[19] T. Massie, *Initial Haptic Explorations with the Phantom: Virtual Touch through Point Interaction*, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.

[20] SensAble Technologies website, Phantom Desktop homepage, http://www.sensable.com/haptic-phantom-desktop.htm, accessed on Dec. 18, 2006.

[21] MPB Technologies website, Freedom 6S homepage, http://www.mpb-technologies.ca/mpbt/haptics/hand_controllers/freedom/description.html, accessed on Dec. 18, 2006.

[22] SensAble Technologies Technical Staff, *Ghost SDK Version 4 Programmer's Guide*, SensAble Technologies, 2002.

[23] Reachin Technologies, http://www.reachin.se/, accessed on Dec. 15, 2006.

[24] Reachin Technologies, Reachin Display, http://www.reachin.se/products/Reachindisplay/index.asp, accessed on Dec. 15, 2006.

[25] Reachin Technologies, Reachin API, http://www.reachin.se/products/ReachinAPI/index.asp, accessed on Dec. 15, 2006.

[26] Novint Website, E-Touch homepage, http://www.novint.com/etouch2.htm, accessed on Dec. 19, 2006.

[27] CAnet^4 Advanced Applications Program, Hapto-Audio-Visual Environments for Collaborative Tele-Surgery Training over Photonic Networking, Statement of Work, internal report.

[28] M.A. Sagar, D. Bullivant, G.D. Mallinson, P.J. Hunter, *A Virtual Environment and Model of the Eye for Surgical Simulation*, International Conference on Computer Graphics and Interactive Techniques, 1994, pp. 205-212.

[29] J. Baillie, P. Jowell, W. Evangelou, W. Bickel, *Use of computer graphics simulation for teaching of flexible sigmoidoscopy*, Endoscopy 23, 1991, 126-129.

[30] S. Delp, P. Loan, M. Hoy, F. Zajac, S. Fisher, J. Rosen, *An Interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures,* IEEE Transactions on Biomedical Engineering, 37, 8, 1990.

[31] D. Hasimoto, T. Dohi, M. Tzuzuki, Y. Horiuchi, K. Chinzei, M. Suzuki, Y. Idezuki, *Development of a computer aided surgery system: Three dimensional graphic reconstruction for treatment of liver cancer*, Surgery, 1991, pp. 589-596.

[32] D. Sims, *The Point where Lines Converge*, IEEE Computer Graphics and Applications, 13, 1993, 7-9.

[33] All about vision website, Eye anatomy explained webpage, http://www.allaboutvision.com/resources/anatomy.htm, last accessed Dec. 11, 2006

[34] Human Eye Model [Search Result], Exchange3D. Accessed on Sept 8, 2005. www.exchange3d.com

[35] Autodesk Website , 3D Studio Max Homepage, http://usa.autodesk.com/adsk/servlet/index?id=5659302&siteID=123112, last accessed Dec. 11, 2006

[36] Computer Associates website, Cosmo Homepage, http://www.ca.com/cosmo/, last accessed Dec. 11, 2006.

[37] The Virtual Reality Modeling Language Specification, http://www.topedge.com/panels/cg/vrml/spec/index.html, last accessed Dec. 11, 2006

[38] Reachin Technologies AB Technical Staff, *Reachin API 3.1 Programmer's Guide*, Reachin Technologies AB, 2002.

[39] N.R. El-Far, S. Nourian, J. Zhou, A. Hamam, X. Shen, N.D Georganas, *A Cataract Tele-Surgery Training Application in a Hapo-Visual Collaborative Environment Running over the CANARIE Photonic Network*, IEEE International Workshop on Haptic Audio Visual Environments and their Applications, 2005.

[40] A. Hamam, S. Nourian, N.R. El-Far, F. Malric, X. Shen, N.D. Georganas, *A Distributed, Collaborative and Haptic-Enabled Eye Cataract Surgery Application with a User Interface on Desktop, Stereo Desktop and Immersive Displays*, IEEE International Workshop on Haptic Audio Visual Environments and their Applications, 2006.

[41] S.Nourian, "xPheve: Extensible Physics Engine for Virtual Environments" , University of Ottawa, 2004

[42] CAVELib, http://www.vrco.com/CAVELib/OverviewCAVELib.html. Accessed on Sept 1, 2006.

[43] VRJuggler, http://www.vrjuggler.org/, accessed on Sept 1, 2006.

[44] SGI – OpenGL Performer, http://www.sgi.com/products/software/performer/, accessed on Sept 1, 2006.

[45] OpenSceneGraph, http://www.openscenegraph.net/. Accessed on Sept 1, 2006.

[46] Virtools, http://www.virtools.com/solutions/products/virtools_dev_new3_5.asp, accessed on Sept 1, 2006.

[47] OpenGL, http://www.opengl.org/, accessed on Sept 1, 2006.

[48] Tackd, http://www.vrco.com/trackd/Overviewtrackd.html, accessed on Sept 1, 2006.

[49] VRPN, http://www.cs.unc.edu/Research/vrpn/, accessed on Sept 1, 2006.