A Haptics and Virtual Reality Simulator for Cataract Surgery

BY

Jia Luo

B.S., Beijing Institute of Technology, 2010

Dissertation

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering and Operations Research
in the Graduate College of the
University of Illinois at Chicago, 2016

Chicago, Illinois

Defense Committee:

Prashant Banerjee, Chair and Advisor
Cristian J. Luciano
David He
William G. Myers, University of Chicago
Shameema Sikder, Johns Hopkins University

To my dearest parents,

for their unconditional love and support.

ACKNOWLEDGMENTS

I would like to gratefully thank my academic advisor, Professor Pat Banerjee, for his supervision and support in the past six years for my Ph.D. study and research. I would like to acknowledge Dr. William G. Myers and Dr. Shameema Sikder for their professional comments and suggestions. I would not be able to accomplish this research without their kindly help. I want to sincerely appreciate Dr. Cristian J. Luciano for guiding my study in the field of virtual reality as well as all the valuable help and support throughout this research.

I would like to express my gratitude to the other members of the defense committee, Professors David He and Lin Li, for all their detailed comments and helpful suggestions.

I would like to acknowledge Dr. Shun Liang for his remarkable pioneering work on the design of the capsulorrhexis simulator. I want to truly thank Patrick Kania for his help of building the virtual eye model as well as conducting experiments on the simulator.

I would also like to thank my lab mates Dr. Silvio Rizzi, Dr. Shaojie Zhang, Dr. Xiaorui Zhao, Jie Jiang, and Nihar Sheth for their immediate suggestions and help during my research.

JL

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

CHAPTER PAGE

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

LIST OF FIGURES (continued)

FIGURE PAGE

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AAO	American Academy of Ophthalmology
API	Application Programming Interface
ASCRS	American Society of Cataract and Refractive Surgery
CCD	Charge-coupled Device
CG	Computer Graphic
DC	Direct Current
DOF	Degree of Freedom
FOV	Field of View
FLTK	Fast Light Toolkit
FPS	Frames per Second
GLSL	OpenGL shading language
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HID	Human Interface Device
HIP	Haptic Interaction Point
IOL	Intraocular Lenses
LCD	Liquid-crystal Display
MCU	Microcontroller Unit
NEI	National Eye Institute

LIST OF ABBREVIATIONS (continued)

NIH	National Institute of Health
OR	Operating Room
PD	Pupillary Distance
PGY	Postgraduate Year
SDK	Software Development Kit
STL	Standard Template Library
VR	Virtual Reality
WHO	World Health Organization

SUMMARY

Cataract surgery is one of the most common ophthalmic surgical procedures. Traditional training of cataract surgery is done on artificial training materials, animals, or cadavers. Although the traditional training methods have shown acceptable results, they also present major drawbacks, including high cost of resources, low time efficiency, and even ethical issues. Computer-based training has the potential to overcome those limitations, meanwhile offering a solution with satisfactory level of realism.

A haptics and virtual reality simulator for cataract surgery is presented. As a computer-based surgical training platform, this simulator provides realistic haptic feedback as well as vivid 3D visualization. Simulations of a series of surgical tasks and exercises are implemented, including micro-dexterity, eyeball balancing, corneal incision construction, phaco sculpting, lens cracking and lens quadrants removal. Novel algorithms for recreation of visual and haptic effects as well as physics simulation encountered in cataract surgery are presented.

1 INTRODUCTION

1.1 Introduction to Cataract Surgery

Cataract (Figure 1) refers to a clouding of the natural intraocular crystalline lens of the eye which can cause a decrease in vision (AAO, 2014). Cataract is “an important cause of low vision in both developed and developing countries” (WHO, 2015). Symptoms of the cataract may include: double vision, cloudy vision, diminishing of color vibrancy, sensitivity to bright lights, etc. Figure 2 shows a scene viewed by a person with cataract and the corresponding normal vision. Cataract remains the leading cause of blindness. According to a recent assessment by WHO, “cataract is responsible for 51% of world blindness, which represents about 20 million people” (WHO, 2015).

Cataracts are very common in senior citizens. More than half of all Americans over 80 years old either have a cataract or have had cataract surgery (NEI, 2006). “As people in the world live longer, the number of people with cataract is anticipated to grow. Although most cases of cataract are related to the ageing process, a cataract may occasionally be congenital, or develop after eye injuries, inflammation, or some other eye diseases” (WHO, 2015).

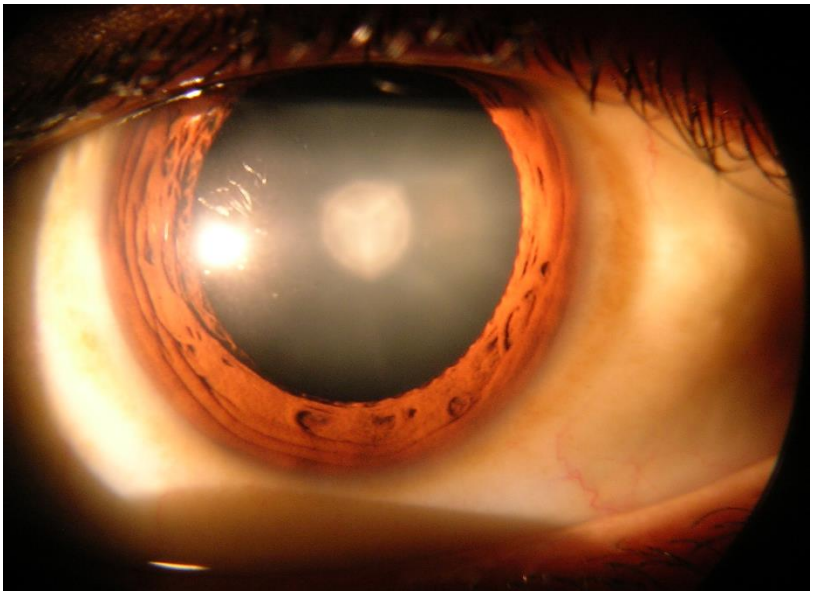

Figure 1. Cataract in human eye
Source: Rakesh Ahuja, MD, Copyright © 2016 Wikimedia Commons

Figure 2. Vision with cataract (left) compared with clear lens (right)
Source: National Eye Institute, Copyright © 2015 National Institute of Health

Cataracts are never reversible. The treatment of cataract is surgical, which removes the opaque lens replaces it by an artificial intraocular lens. The cataract surgeries are very successful in restoring sight. In more than 90% of the cases, useful vision can be restored successfully with a low complication rate.

Extracapsular surgery and phacoemulsification are two main types of surgical procedures. Extracapsular surgery is the traditional way of cataract surgery. “A long incision is made on the side of the cornea - the clear, dome-shaped surface that covers the front of the eye. The cloudy core of the lens is removed in one piece. The rest of the lens is removed by suction” (Wikipedia, 2016).

Phacoemulsification (phaco) is the most commonly used technique nowadays, accounting for more than 90 percent of the cataract surgeries performed in developed countries (Leaming, 2004). A small, self-healing corneal incision is constructed to allow insertion of the instruments. A circular tear is then performed on the anterior lens capsule to enable direct contact with the cataract lens. A phaco handpiece with a steel or titanium tip that may vibrate at ultrasonic frequency is inserted into the eye to emulsify the lens material. A lens manipulator may be used from a side port corneal incision to assist rotating the lens nucleus and cracking it into smaller pieces. The lens fragments are then emulsified and aspirated from the eye.

Recently, the femtosecond laser technology has been introduced to the cataract surgery. It replaces or assists the use of a hand-held surgical tool for several surgical procedures in phacoemulsification, including capsulorrhexis and lens fragmentation. “Initial results with an intraocular femtosecond laser demonstrate higher precision of capsulorrhexis and reduced phacoemulsification power in porcine and human eyes” (Nagy et al., 2009). However, it is still a relatively new technology and not as commonly used as the traditional phacoemulsification.

Due to the popularity of phacoemulsification, it is the type of cataract surgery that our simulator mainly focuses on.

1.2 Surgical training using virtual reality (VR) technology

“The first attempt at manipulating micro-instruments under a microscope should not be on a patient” (Caesar et al., 2003). Therefore surgeons must be well trained before performing actual operations. Traditionally, surgical residents usually practice the surgical techniques by using substitute objects, dummy bodies, animals or cadavers. Although every method has its own advantages, the limitations in terms of time efficiency, financial costs, human costs, ethical issue and so on are also non-ignorable (Haluck et al., 2000). Computer-based training is another approach in surgical education. It has the potential to not only overcome the disadvantages of those traditional training methods, but also provide many exclusive benefits.

First of all, the costs for computer-based training are much lower than those that use animals or cadavers. Computer-based training platform provides users a virtual reality (VR) environment for the operation, and no any actual material is consumed. Thus it allows trainees to practice the surgical procedures multiple times without incurring any costs of resources. In contrast, the costs for using animals and cadavers are relatively high. On top of that, due to the nature of the actual materials, operations like cutting, tearing, or piercing cannot be undone and therefore animals and cadavers are not reusable.

Besides, from realistic prospective, using animals for training may not be good enough due to the difference in anatomical structures. Though cadaveric dissection doesn't have this issue, it is not possible to provide the physiological response of living patients (Luciano, 2010). These are not the problems for VR simulation, as long as the virtual models are made according to the anatomical structure, and the simulation faithfully mimics the patient’s movement and reaction.

More importantly, the VR-based training has its own unique advantages. The virtual reality environment provides the freedom of manipulating virtual objects. With the help of the computer graphic (CG) technology, object properties like position, orientation, color, transparency or even physical boundary can be changed by demand. These exclusive features of VR can bring unique

experiences to the trainees to help them better understand the surgical procedures. For example, the inner anatomy structure can be visualized by virtually “cutting” the models; trainees may observe the surgical scene in a more comprehensive way by freely altering the view point.

In addition, by monitoring and analyzing performance in real-time, simulator can provide suggestions or warnings to the trainees during their practice. All operations in the simulator can be recorded and thus it can be replayed for reviewing the surgical technique. Moreover, computer-based simulator can provide an objective assessment of the trainees’ performance and track their progress during multiple training sessions.

With all benefits mentioned above, the computer-based simulation can be a better approach for practicing surgeries. Studies including (Gallagher et al., 2004), (Fried et al., 2004) and (Seymour, 2008) have demonstrated that, after participating VR-based training, medical students, resident physicians, and fellows had significantly improved the performance in the actual surgeries compared with untrained controls. Currently, computer-based simulation have been adopted by surgical training of many medical specialties, including ophthalmology training (Khalifa et al., 2006). Several cataract surgery simulators have been reported.

1.3 Organization

This dissertation is organized in 7 chapters. Chapter 2 presents an overview of several existing computer-based cataract surgery training platforms, as well as the previous works that serves as the foundation of my research. Chapter 3 describes the hardware improvement of the MicrovisTouch simulation platform. Chapter 4 illustrates the software design of the cataract surgery simulation, including the design of the dynamic eye model as well as models and algorithms for modifying geometry in simulation run time. Chapter 5 introduces the simulation of multiple cataract surgical procedures as well as several VR-based applications for training of surgical techniques. Chapter 6 presents a survey of tactile feedback in cataract surgery. Chapter 7 highlight the main contribution of this project and suggests possible future work.

2 PREVIOUS RESEARCH WORKS

2.1 Review of previous simulators

Several cataract surgery simulators have been reported. Examples are the following:

- Eyesi Surgical (Schill et al., 1999)
- PhacoVision (Söderberg et al., 2002)
- Phacoemulsification cataract surgery simulator (Lam et al., 2012)

A brief review for each of the simulators will be given in this section.

2.1.1 Eyesi Surgical

Designed by VRmagic Holding AG, Mannheim, Germany, Eyesi Surgical is a virtual reality simulator for intraocular surgical training. It was first presented by (Schill et al., 1999).

In the Eyesi Surgical, A mechanical model of the eye is used in the simulator. A series of pinholes are provided on the mechanical model of eye to allow instrument insertion. Three CCD cameras are used to capture the real-time images of the eye model and the instruments, which are then used by computer to calculate the position of the instruments. A virtual scenario for surgical simulation is established and presents to the user through two small LCD displays which mimic the stereo microscope used in actual cataract surgeries. Foot pedals are used for emulating the control of microscope and OR machines.

All operations including position tracking, visualization, collision detection and manipulation of virtual tissue are efficient enough to guarantee a real-time simulation (Schill et al., 1999).

The advantages of the Eyesi Surgical simulator are the following:

The hardware of the simulator mimics the environment in the OR. The patient head model lies on a height-adjustable table facing the surgeon, and user sits in the same position as surgeon does in the actual surgery. Hardware components including stereoscopic display system, foot pedals and handheld instruments are similar to the components of the actual surgical platform, enhancing the realism of the surgical simulation (VRmagic, 2014).

Eyesi Surgical provides simulation of a variety of procedures of the cataract surgery, including general scenarios like microscope handling, understanding of spatial boundaries, or the actual surgical steps, such as capsulorrhexis, hydrodissection, phaco, irrigation/aspiration, and IOL insertion.

A curriculum is provided with different cases and multiple levels of difficulty. Performance can be assessed and recorded. Immediate feedback is available after each task, and historical performance can be monitored. Third party studies (Sachdeva et al., 2011) and (Selvander et al., 2013) have shown the validity of the performance assessment.

Disadvantages of the Eyesi Surgical simulator are as follows:

The only tactile sensation available to the user comes from the physical contact between the mechanical model of eye and the handheld instruments. This oversimplifies the scenario in the surgery. Due to the lack of haptics, the tactile feedback is missing when instruments interact with the anatomy in the eye. However surgeons have reported that the tactile sensation is non-ignorable and sometimes it does play an important role in the actual surgery.

As a result of the hardware limitation of EyeSi Surgical simulator, the instruments are only trackable when the tips stay inside the eye. This makes it impossible to simulate any procedures that involves operation outside the eye. In addition, due to the same reason, the virtual instruments shown on the display may be blinking or jumping when instrument tips approach the boundary of eyeballs.

In addition, the preset pinholes around the mechanical eye model have eliminated the freedom of constructing the corneal incisions in the simulation. As we will discuss in the later chapters, it’s beneficial for the trainees to practice corneal incision construction under some specific patterns for the purposes of minimally invasive and self-healing.

Besides, the graphical resolution of the display system is only 800 by 600 pixels for each eye (Schill et al., 1999). Low resolution has limited the level of detail of the visualization, additionally, as user's eyes have to be very close to the displays, the problem of low resolution becomes more noticeable.

2.1.2 PhacoVision

PhacoVision was first introduced as a simulator for phacoemulsification by (Söderberg et al., 2002), (Söderberg et al., 2003). It was developed by Melerit Medical from Linköping, Sweden.

PhacoVision "consists of a personal computer, a 3-dimensional visual interface, a phacoemulsification handpiece, a nucleus manipulator, and foot pedals" (Laurell et al., 2004). During simulation the surgical field is seen in a microscope. Handheld tools that represent the instruments are used in the same way as in real operations. Pedals are used to control the microscope image and the phaco machine.

There are several advantages of PhacoVision. First, PhacoVision simulates the anatomy with several variations, such as different levels of cataract, variable strength of zonula, etc. This enables users practicing the cataract surgery with different cases of complications. Second, the hardware components of the PhacoVision are designed to closely resemble the actual equipment used in the OR. Third, user performance on PhacoVision can be assessed after training and

feedback is presented to the user. Performance analysis has been conducted (Söderberg et al., 2007), (Söderberg et al., 2008).

On the other hand, limitations of this training simulator are listed below. First of all, the research on PhacoVision simulator is very limited, and its current progress is unclear. Besides, this simulator also does not support haptic feedback. Trainee cannot feel the tactile sensation of the eye during practice. Additionally, the hardware seems to be lack of organizing and separated into several components. It is not convenient for assembling or transportation.

2.1.3 Simulator by Lam et al.

A “virtual reality simulator for phacoemulsification cataract surgery education and training” was introduced by a research team from Perlis, Malaysia (Lam et al., 2012), (Lam et al., 2013).

The advantages of this simulator are as follows. First of all, the simulator is equipped with two haptic devices to provide tactile feedback from the virtual scene. Besides, a research on simulating the dynamics of the human eyeball is conducted and applied to their cataract surgery simulator. Additionally, the simulator is capable of simulating several procedures in phacoemulsification surgery.

Meanwhile, the disadvantages of this simulator are the following. First, the research on this simulator is very limited. There is no any published result from a third party validating this simulator. Besides, this simulator doesn't support stereoscopic display. When 3D virtual objects are projected to a 2D display, the depth information is ignored. The lack of depth perception apparently limits the realism of visualization. Additionally, multiple pieces of hardware components need to be better organized. The relative position between two haptic devices are not fixed, and therefore a random error exists when collocating the virtual instruments with the haptic devices.

2.2 MicrovisTouch

First presented by (Banerjee et al., 2009), the MicrovisTouch (Figure 3) is a microsurgery simulation platform designed by Industrial Virtual Reality Institute in the University of Illinois at Chicago and Immersivetouch Inc. MicrovisTouch is the platform we use for the cataract surgery simulation.

The MicrovisTouch mainly consists of the following parts: a powerful workstation computer, a 3D eyepiece for stereo visualization, two large screens with a total resolution of 2160 by 1920 pixels, a pair of haptic devices for tactile sensation, foot pedals for emulating the control of surgery machine.

Figure 3. MicrovisTouch
Source: immersivetouch.com

Figure 4 shows the principle of this stereo display system. The mirrors in the eyepiece horizontally shift the field of view (FOV) for two eyes to the sides, making the image on only one monitor visible by each eye. By displaying the offset images for each eye on the corresponding monitors, the 3D stereo vision is

achieved. The angle of the mirrors can be fine-tuned to guarantee that the entire monitor is included in the FOV for users with different pupillary distance (PD). To accommodate the varying sitting height of different users, the height of the eyepiece and monitors is also adjustable.

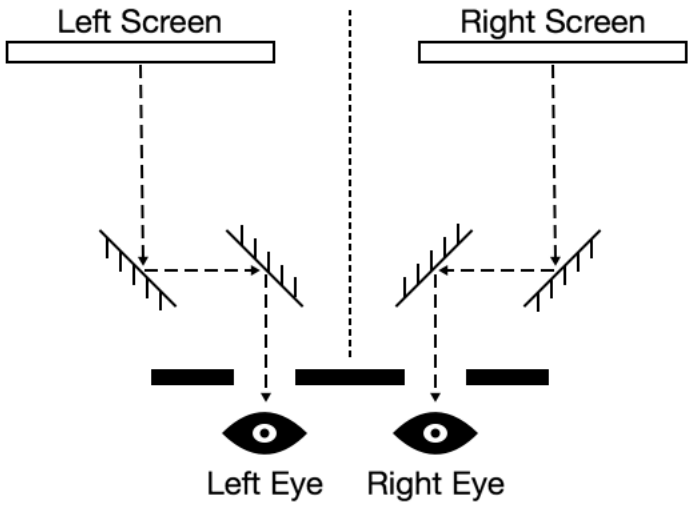

Figure 4. Principle of the stereo display of MicrovisTouch

Use of haptics is one of the most important features of MicrovisTouch. Geomagic Touch (formerly Sensable Phantom Omni) (Figure 5, left) or Geomagic Touch X (formerly Sensable Phantom Desktop) (Figure 5, right) are

the haptic devices that MicrovisTouch platform uses. “The haptic device can accurately measure the 3D spatial position (along the x-, y- and z-axis) and the orientation (roll, pitch and yaw) of its handheld stylus. The devices use motors to generate forces that push back on the user’s hand to simulate touch and interaction with virtual objects” (Geomagic, 2013). During the simulation, each haptic stylus is bound to a virtual surgical instrument. User may both see and feel as if he/she is holding the virtual instrument and operates on the virtual patient.

Figure 5. Geomagic Touch (left) and Geomagic Touch X (right)
Source: geomagic.com

2.3 Sensimmer

Sensimmer is a software library for the development of haptic and virtual reality applications. It is developed along with the evolution of the Immersivetouch (Luciano et al., 2005), (Banerjee et al., 2005), an augmented and virtual reality platform for open and percutaneous surgery simulation. Sensimmer is capable of simultaneous haptics and graphics rendering of 3D models, as well as simulating dynamic interaction among the virtual objects. To do so, Sensimmer seamlessly integrated several software libraries including Coin3D, OpenHaptics, and PhysX.

Coin3D is an open-source library for 3D graphics rendering (Coin3D, 2014). It is an implementation of Open Inventor, an object-oriented graphics engine designed on top of OpenGL. It uses a scene-graph structure to organize and render the 3D objects. Coin3D manages virtual objects, lighting, cameras as well as their properties like positions, orientations, colors, textures, etc.

OpenHaptics is a programming toolkit for haptic rendering. It is designed for Geomagic haptic devices (formally Sensable Phantom devices). “At a high level, the toolkit enables true 3D navigation, material properties, and polygonal object support for applications already using OpenGL. At lower levels, the toolkit gives developers complete access to sensor readings, device control, and direct control of force rendering” (Geomagic, 2013).

PhysX (NVIDIA, 2015) is a multi-threaded real-time physics engine developed by NVIDIA. Based on position-based dynamics (Müller et al., 2007), PhysX is capable of detecting collision between virtual objects, establishing joints between joints and applying physics laws to move objects in the virtual scene. PhysX SDK supports rigid body dynamics, soft body dynamics, cloth simulation and volumetric fluid simulation.

The above libraries are well integrated by Sensimmer. Sensimmer follows the object-oriented style of Open Inventor and applies the scene-graph structure to haptic rendering and physical simulation. More importantly, Sensimmer helps libraries communicate with each other. For instance, Sensimmer sends the force simulated by the physics engine to the haptic device so that the virtual force interaction between virtual objects is perceivable by the user.

2.4 Capsulorrhexis simulator

Capsulorrhexis is a procedure in the cataract surgery. The aim of the procedure is to create a continuous curvilinear tear on the anterior lens capsule. Capsulorrhexis provides access to the lens material inside the capsule, while remaining the capsular bag for the artificial lens placement.

Developed by (Liang, 2009), (Liang, 2010), the capsulorrhexis simulator (Figure 6) is the first cataract surgery simulation that runs on the MicrovisTouch platform. “High fidelity haptic feedback is rendered to provide the sense of real

surgery feelings. The deformable capsule is modeled with position-based dynamics with improved performance over traditional mass-spring models. A customized tearing algorithm with full controllability is developed to simulate the tearing/capsulorrhexis procedure. A combination of vertex and fragment shaders is adopted to provide improved realism on visual effect" (Liang, 2010).

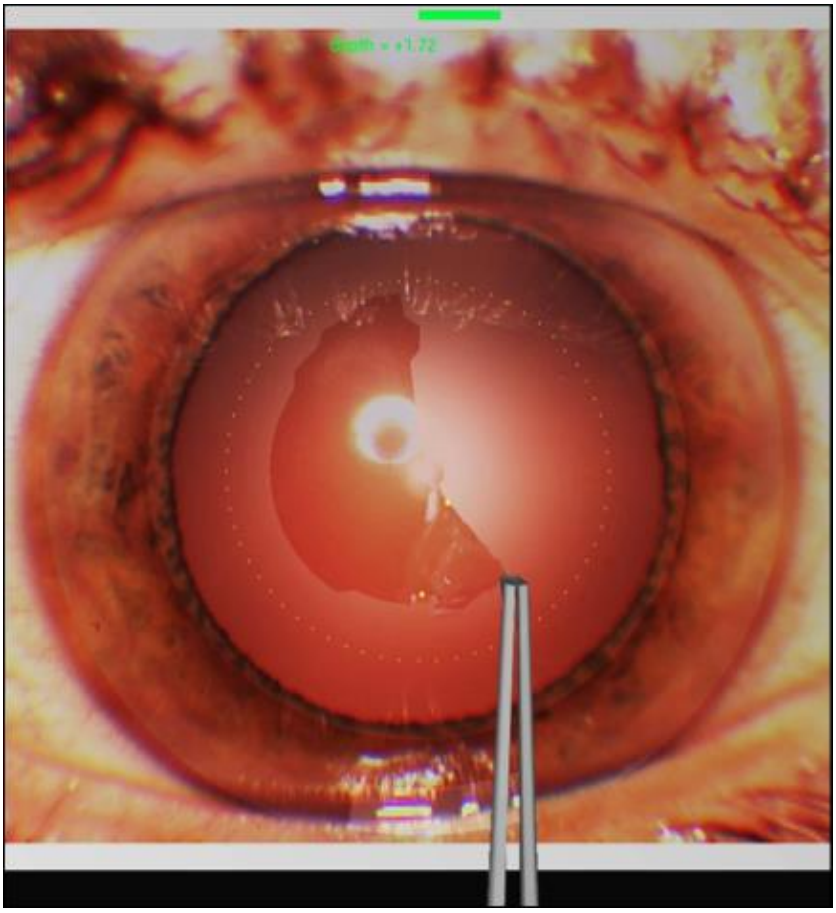

Figure 6. Capsulorrhexis simulator

The capsulorrhexis simulator monitors the entire performance and evaluate it in several aspects including circularity, accuracy and fluency. The circularity score measures the shape similarity between the capsulorrhexis and a circle with ideal radius. The accuracy score reflects the number of times the lens or cornea being touched by the tip of the instrument. If the instrument stretches the corneal incision excessively, the score will also be negatively impacted. The fluency score indicates the number of times the tearing is performed for the entire capsulorrhexis. An overall score is calculated as the weighted average of the above three scores.

Several studies have been conducted to investigate the validity of the capsulorrhexis simulator. (Banerjee et al., 2012) showed a significant concurrent validity of the circularity of the capsulorrhexis between the simulator and live surgeries. (Sikder et al., 2015) evaluated the performance of the capsulorrhexis simulator by two rounds of testing in ophthalmology residents in the US and Saudi Arabia. “The average scores in all measured metrics demonstrated statistically significant improvement (except for circularity, which trended toward improvement)” in the follow-up testing compared with the baseline assessment conducted 6 months before (Sikder et al., 2015).

3 HARDWARE IMPROVEMENTS ON THE MICROVISTOUCH SIMULATION PLATFORM

3.1 Introduction

The cataract surgery simulator is developed on the MicrovisTouch simulation platform. With its visual and haptics interfaces, the MicrovisTouch platform provides a virtual reality environment for the surgical training. Along with the development of the cataract surgery simulators, improvements have also been accomplished in the hardware design of the MicrovisTouch platform.

As the most important factor of evaluating virtual reality platforms, realism is always in the focus of attention when designing the cataract surgery simulator. From the hardware prospective, the simulator should emulate the environment of the actual operating room. The operation on the simulator should be approximate enough to the actual surgery so that the kinesthetic memory of the training can effectively improve the surgical skill.

Ergonomics is another key principle. The simulation platform should be designed with consideration of human factors. A comfortable experience helps user concentrate on the training. In addition, the interaction of the user and the simulator should be straightforward while effective.

To achieve these goals, several new components have been introduced to the simulation platform.

3.2 Pressure sensor

Haptic device can detect the movement of user's hand, but it's not possible to perceive the pressure of user's fingers. Meanwhile, surgical instruments like forceps or pliers need to be manipulated by finger pressure. To address this issue, a force sensing resistor is attached on the haptic stylus where the thumb finger locates (Figure 7). When the finger presses on the resistor, its value of resistance will change according to the force applied. Therefore, this force sensing resistor can serve as the sensor for the finger pressure. In the capsulorrhexis simulator, it is used to control the forceps to tear the anterior lens capsule. The finger pressure determines the opening and closing of the virtual forceps. Sufficient pressure keeps forceps firmly closed, and the lens capsule can be grasped and torn.

Figure 7. Pressure sensor

3.3 Phaco pedal

In the phacoemulsification surgery, a special foot pedal is used to control the phaco machine. In order to provide a more realistic training environment, a same phaco pedal is introduced to the MicrovisTouch platform. Figure 8 shows the phaco pedal.

Figure 8. Phaco pedal

The phaco pedal consists of a main foot pedal that supports both vertical and horizontal movement as well as four side buttons at the corners of the pedal. Status of the main pedal and side buttons will be output as several electronic signals.

In the actual surgery, the phaco machine is controlled by the main vertical pedal, with four positions triggering different functionalities:

- Position 0 – Off
- Position 1 – Irrigation

- Position 2 – Irrigation and aspiration
- Position 3 – Irrigation, aspiration and phaco power

In our simulator, the above effects will be simulated respectively when the pedal is pressed down to the corresponding positions.

3.4 Human interface device (HID) hub

Both the pressure sensor and the phaco pedal take inputs from the human action and generate electronic signals as outputs. They can be called as human interface device (HID). A hub device is designed to monitor HIDs including a phaco pedal and up to two pressure sensors, and relay the input of those devices to the computer. The hub generates a DC electronic signal to reflect the resistance of each pressure sensor, and scale the signals from the phaco pedal to a proper voltage range. All electronic signals are then read by the programmable microcontroller unit (MCU) in the hub device. The MCU converts the analog signals from the pressure sensors and phaco pedal into digital values, and send the information to the computer by serial communication. Figure 9 shows the HID hub.

Figure 9. HID hub

3.5 Head Mannequin

To stabilize the hand and improve precision, surgeons usually support their wrist on patient's forehead or the side of the face during cataract surgery. Since the patient model only exists in the virtual scene on the MicrovisTouch platform, the user can only rest the hand on the tabletop, or simply have the wrist hanging. This will make the hand movement of using the simulator different from performing the actual surgery. To address this issue, a silicone mannequin head has been added to the tabletop of the simulator. The head mannequin can be placed in different orientations so that operator may rest the wrists on the mannequin's forehead or either side of the face.

The mannequin also provides a position reference so that user may quickly locate the workspace of the virtual instrument, which is relatively small due to the scale of the human eye. Just as the collocation of the haptic styluses and the virtual instrument, the mannequin overlaps with the patient model in the virtual scene. In this way, manipulating haptic styluses on top of the mannequin matches the interaction between the virtual instruments and patient model. Figure 10 shows the head mannequin with two haptic devices.

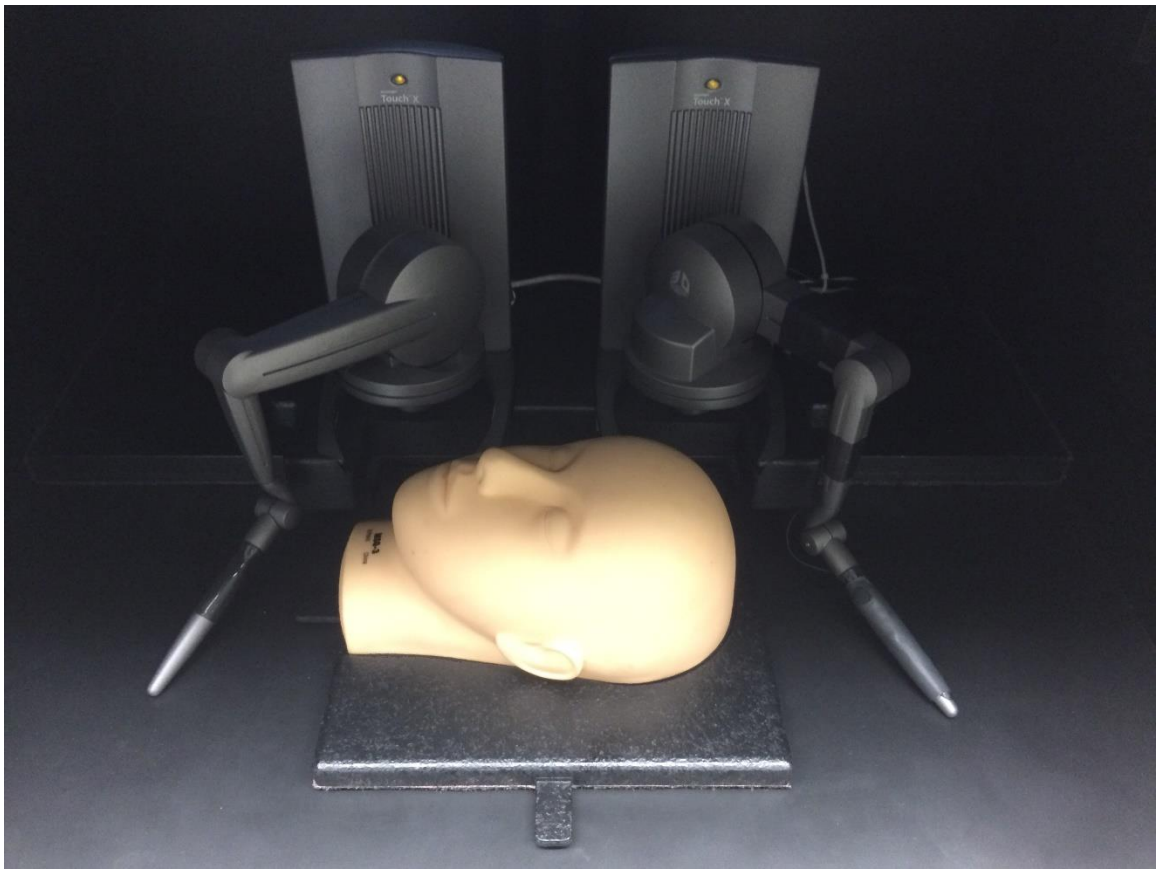

Figure 10. Head mannequin with haptic devices

3.6 Summary

Several improvements have been made on the hardware design of the MicrovisTouch platform, making the simulation more realistic and ergonomic. Figure 11 shows the hardware composition of the cataract surgery simulator.

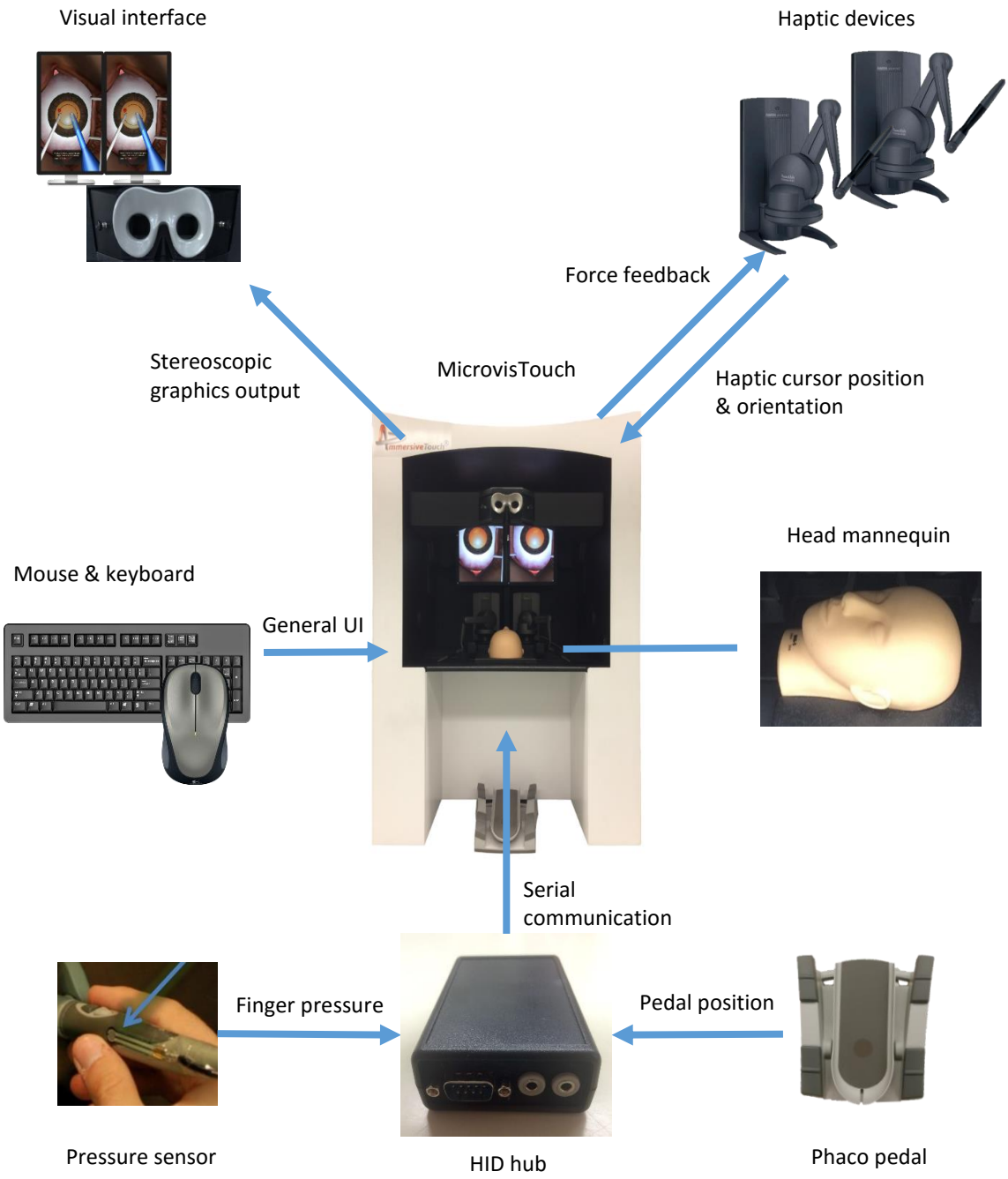

Figure 11. Hardware composition of the cataract surgery simulator

4 SOFTWARE DESIGNS OF THE CATARACT SURGERY SIMULATOR

4.1 Model of the Human Eye

4.1.1 Overview

Cataract surgery is performed under the microscope. The entire workspace of the surgery is within the eye. Correspondingly, in the simulations, all procedures are performed on top of or inside the eye. Therefore the virtual eye model is the foundation of the VR-based simulation.

For all non-haptic-based training platforms, the eye model either sacrifices the flexibility and realism by having a series of predefined entry points, or has a limited life of use due to the fact that operations including piercing or cutting cannot be undone. The haptic interface does not present those problems, as it allows the virtual incision to be constructed at any location on the virtual eyeball, and the operation can be repeated over and over without destroying any actual material.

The virtual human eye model (Figure 12) is responsible of both visualization and haptic feedback in the simulation. It “consists of realistic 3D polygonal meshes representing all major anatomical structures, including cornea, sclera, iris, lens, retina, and eyelid, surrounded by a virtual 3D model of the patient’s face” (Luo et al., 2016). Vivid color and textures are applied to the model for realism of graphics rendering.

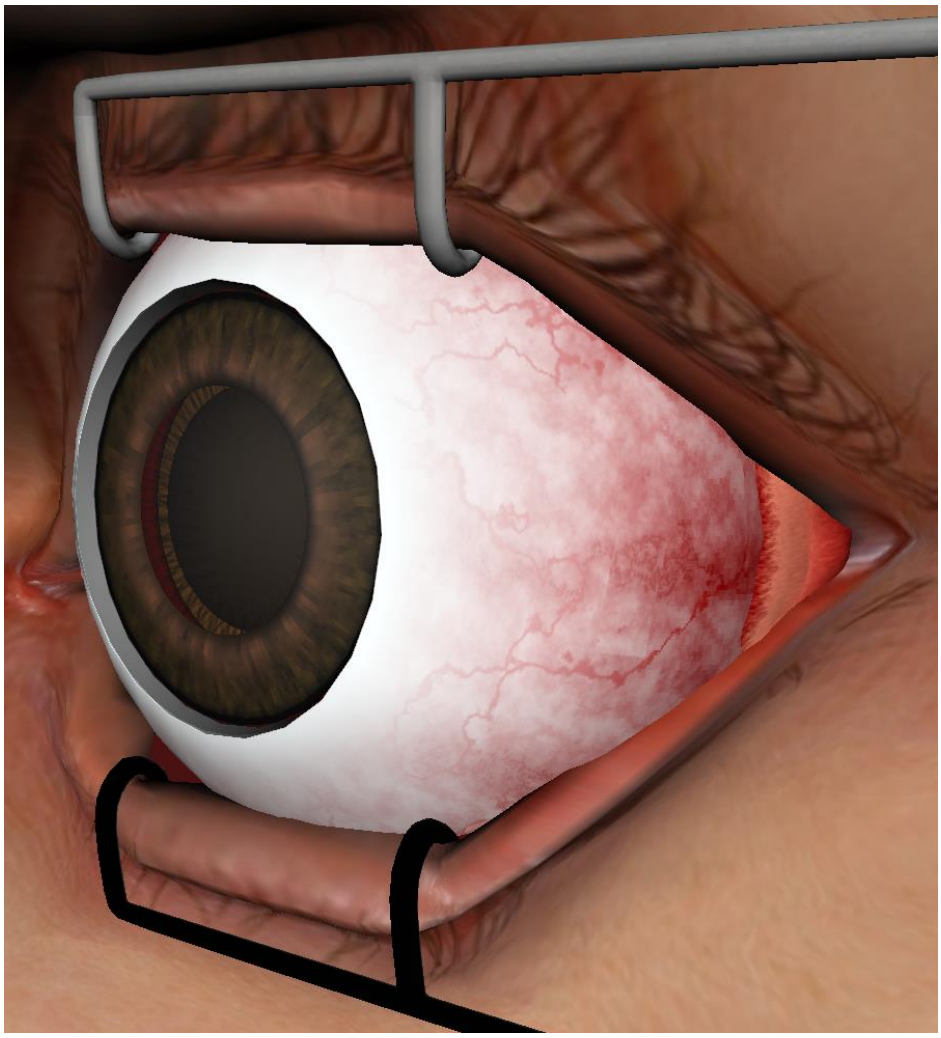

Figure 12. 3D eye models

4.1.2 Haptics Interaction

The haptic interaction between the virtual instruments and the human eye models is achieved by the proxy-based method implemented in the Haptic Library API (HLAPI) of OpenHaptics. The proxy, also known as the “god-object” (Zilles et al., 1995), follows the Haptic Interaction Point (HIP) and remains outside the touchable face of the shape. In the simulation, the proxy determines the tip position and orientation of the virtual surgical instrument. When the virtual instrument touches the virtual eye model, it visually stays outside of the mesh surface.

The haptic feedback is computed by pulling a virtual spring-damper from the proxy position to the HIP. Therefore, the haptic force is proportional to the instrument depth into the model. By properly configuring the haptic properties of stiffness, damping and friction coefficients for each shape, the tactile sensation for different parts of the eye can be vividly emulated.

4.1.3 Deformation

To simulate the deformation of the virtual eye model, a GPU based elastic-object deformation algorithm (Luciano et al., 2007) has been applied. When the eye model is touched by the virtual instrument, the polygonal surface is deformed by moving vertices around the proxy position towards the HIP, and updating the normal vectors within the deformed area. Figure 13 shows the sclera being deformed when touched by an instrument.

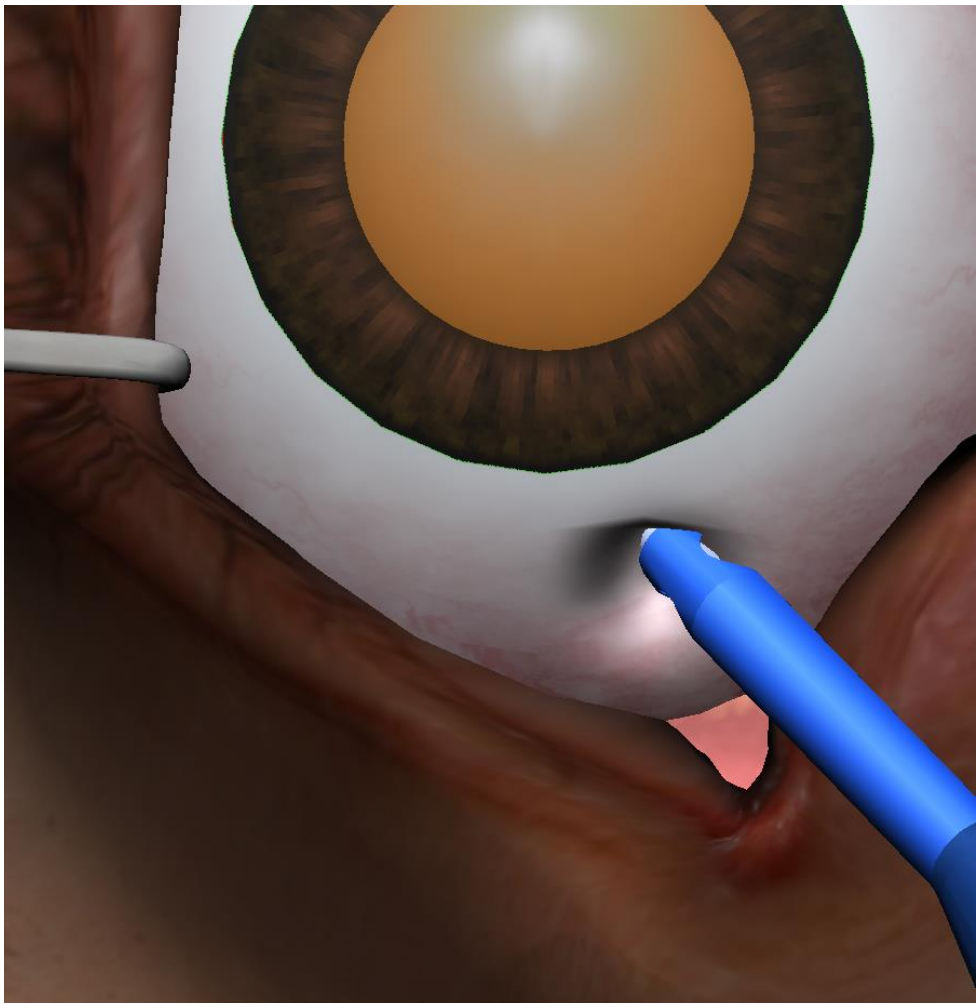

Figure 13. Deformation of the eye model

4.2 Haptic Effects

In addition to proxy-based haptic rendering, which simulates the interaction between the instruments and the eye, several haptic effects are introduced to

simulate ambient sensations that exist throughout the workspace, such as gravity, and tactile feedbacks that occurs under certain conditions, including friction and normal force. Haptic effects can also be used to simulate the mechanical vibration of the surgical instruments.

HLAPI of OpenHaptics toolkit handles the haptic effects. To simulate the force effects smoothly, the force is computed and updated on servo loop that refreshes at 1000 fps to guarantee a smooth haptic feedback. Sensimmer API provides a handy interface to customize the force effects and combine them into the scene graph. Several haptic effects have been implemented as follows.

4.2.1 Constant Force Effect

Usually, the weight of the surgical instrument is less than that of the haptic stylus. To compensate the weight difference, a constant force effect is introduced. The force output to the haptic stylus is a vector that points to the opposite of the gravity direction, with the constant magnitude that reflects the difference of gravity, as (4.1) shows:

$$\boldsymbol{F_{const} = G_I - G_S} \tag{4.1}$$

where $\boldsymbol{G_I}$ is the gravity of the surgical instrument, and $\boldsymbol{G_S}$ is the gravity of the stylus. This effect is enabled throughout the simulation.

4.2.2 Fulcrum Effect of Cornea Incision

In actual cataract surgeries, instruments are operated through a corneal incision. After the instruments pop through the eyeball, their freedom of motion is limited by corneal incisions. However, the point-based collision detection algorithm implemented in OpenHaptics cannot prevent the virtual instruments from deviating from the pop-through points. In order to replicate the restriction of the incision in the virtual scenario, a fulcrum haptic effect was implemented.

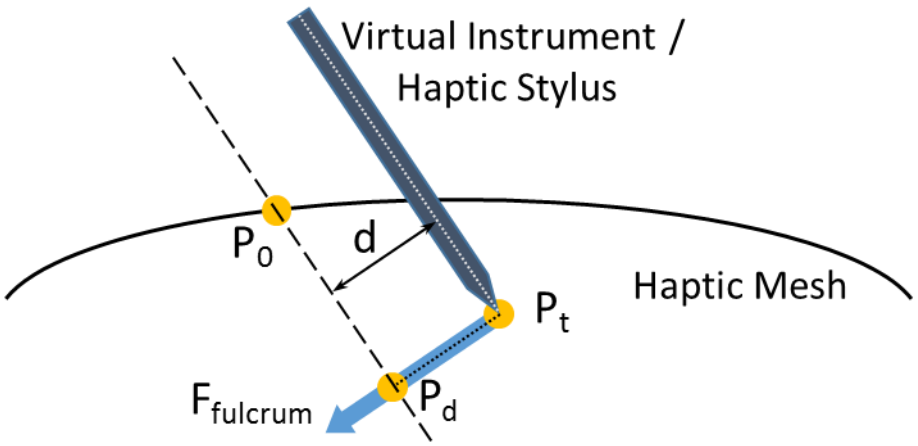

Figure 14. Principle of fulcrum haptic effect

The principle of the fulcrum effect is shown in Figure 14. The position of the corneal incision serves as the fulcrum point (P_0). In each haptics frame, the desired position of the instrument tip (P_d) is computed by projecting the current

position of the tip (P_t) to the fulcrum line, which is passing through the fulcrum point and parallel to the virtual instrument / haptic stylus (i.e. the dash line in Figure 14). A force $F_{fulcrum}$ proportional to the distance (d) between the current proxy and its desired position is applied to the haptic device to pull the deviated instrument back to the fulcrum line. The fulcrum force is shown in (4.2):

$$\boldsymbol{F_{fulcrum}} = Md\frac{\boldsymbol{P_d} - \boldsymbol{P_t}}{|\boldsymbol{P_d} - \boldsymbol{P_t}|} \tag{4.2}$$

where M is the magnitude of the force effect. This haptic effect keeps the instrument in the incision, while allowing free entering and exiting; the rotation of the instrument is also unlimited. The fulcrum effect is triggered when the virtual instrument is inserted into the eyeball. When the instrument is removed, the effect stops accordingly.

4.2.3 Friction Effect during Instruments Insertion

When surgical instrument is inserted into the eyeball, friction occurs between the instrument and the corneal incision. To achieve a smooth experience, a mass-spring-damper system is introduced to compute the resistance. The mass-spring damper system is subject to an oscillatory force $\boldsymbol{f_s}$ as well as a damping force $\boldsymbol{f_d}$:

$$\boldsymbol{F} = \boldsymbol{f_s} + \boldsymbol{f_d} = -k(\boldsymbol{p} - \boldsymbol{x}) - c\boldsymbol{v} \tag{4.3}$$

where $\boldsymbol{p}$ is the proxy position of the haptic device, $\boldsymbol{x}$ is the displacement of the mass, $\boldsymbol{v}$ is its velocity, k is the spring constant, and c is the viscous damping coefficient. $\boldsymbol{x}$ and $\boldsymbol{v}$ are either computed by the previous frame, or obtained from the initial condition. The friction applies only along the direction of instrument insertion (pushing the incision sidewise does not generate friction as the relative position between the instrument and the eye does not change). Therefore, the resistance applied to the haptic device can be expressed as the projection of the inertia force along the direction of the instrument, as equation (4.4) shows:

$$\boldsymbol{F}_{\boldsymbol{friction}} = \left(\boldsymbol{F} \cdot \hat{\boldsymbol{I}}\right)\hat{\boldsymbol{I}} \tag{4.4}$$

where $\hat{\boldsymbol{I}}$ is the unit vector parallel to the tip of the instrument.

According to equation (4.3), the acceleration $\boldsymbol{a}$ of the point mass can be obtained by Newton's second law, as the following:

$$\boldsymbol{a} = \frac{\boldsymbol{F}}{m} = \frac{-k(\boldsymbol{p} - \boldsymbol{x}) - c\boldsymbol{v}}{m} \tag{4.5}$$

where m is the mass. The displacement and velocity of the point mass can then be determined by Euler integration, i.e.

$$\boldsymbol{v}_{\boldsymbol{t}} = \boldsymbol{v}_{\boldsymbol{t-1}} + \boldsymbol{a}\Delta t \tag{4.6}$$

$$\boldsymbol{x_t = x_{t-1} + v_t \Delta t} \quad (4.7)$$

where Δt is time elapsed between two frames. Note that the point mass states computed in equation (4.6) and (4.7) will be used to determine the inertia force in the next frame.

The friction effect is started once the tip of the instrument is engaged into the corneal incision. The effect is stopped when the instrument is removed from the eye. Initial conditions of the point mass is necessary for the first frame, as follows:

$$\boldsymbol{v_0 = 0} \quad (4.8)$$

$$\boldsymbol{x_0 = p} \quad (4.9)$$

where $\boldsymbol{p}$ is the proxy position of the haptic device.

4.2.4 Vibration Effect of Phaco Handpiece

Phaco handpiece is used to sculpt and remove the cataract lens. When in use, phaco power brings vibration to the instrument, which is perceivable by the operator. To mimic the same experience on the simulator, a vibration effect is applied to the haptic stylus. The output force follows a sinusoidal pattern, as (4.10) shows:

$$\boldsymbol{F_{vibration}} = A\sin(2\pi f t)\hat{\boldsymbol{I}} \quad (4.10)$$

where A is the amplitude of the sine wave, f is the frequency of vibration, t is the time, and $\hat{\boldsymbol{I}}$ is a unit vector. Vibration effect is toggled along with the simulation of phaco power, which is controlled by the foot pedal. The frequency of the vibration effect can be variable.

4.3 Dynamics of the Eye Model

4.3.1 Kinematic Joint

In the actual surgery, the human eye can be rotated and moved by instruments inserted through corneal incisions. Meanwhile, “its freedom of motion is restricted by anatomical structures including the orbit, eyelids, and extraocular muscles” (Luo et al., 2016).To be specific, the human eye permits a 120-degree horizontal rotation and a 60-degree vertical rotation. It also allows cyclotorsion within a small range. Translational motion of the human eye is mostly restricted, but a slight axial motion perpendicular to the eye orbit is possible when external force is applied. Figure 15 shows the range of motion of the human eye.

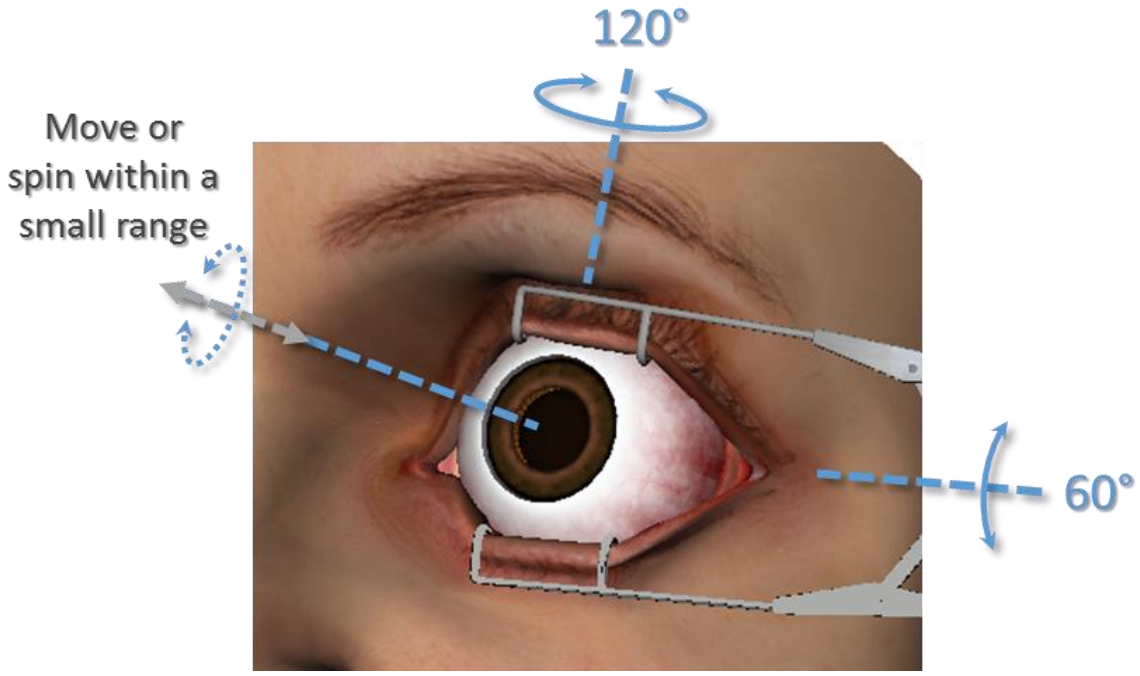

Figure 15. Range of motion of the human eye

To present the restricted range of motion of the eye in the simulation, an articulated kinematic joint model is implemented based on rigid body dynamics (Luo et al, 2016). Joint limits are set for each degree of freedom to allow the rotation and translation of the eye model within the range of motion same as actual human eye. To simulate the physiological response of the extraocular eye muscles, spring and damping are introduced to manipulate the joint. In this way, the eye model can preserve the range of motion even it is exceedingly pushed or rotated by the external force.

4.3.2 Force Interaction

"According to Newton's third law of motion, when the instrument is pushed by the fulcrum force, a reaction force on the opposite direction is applied to the eye model. This reaction force rotates the eye model, which in turn changes the position of the incision in 3D space. Since the incision serves as the fulcrum point, the fulcrum haptic effect itself is eventually affected. This force interaction mechanism (Figure 16) improves the realism of the interaction between virtual instruments and the eye model" (Luo et al, 2016).

When two haptic devices are used in the simulation, both of them may apply force to the virtual eye model. In this case, the eye rotation is determined by the fulcrum force for both instruments. The new orientation of eye changes both fulcrum positions, which then feeds back to the fulcrum force applied on both instruments. For instance, when one instrument pushes against the incision, another inserted instrument will move along with the eye rotation. When both instruments pushes the eye with a zero sum of the torque, the eye should keep stable. Figure 17 shows the feedback mechanism between the physics engine and two haptic devices.

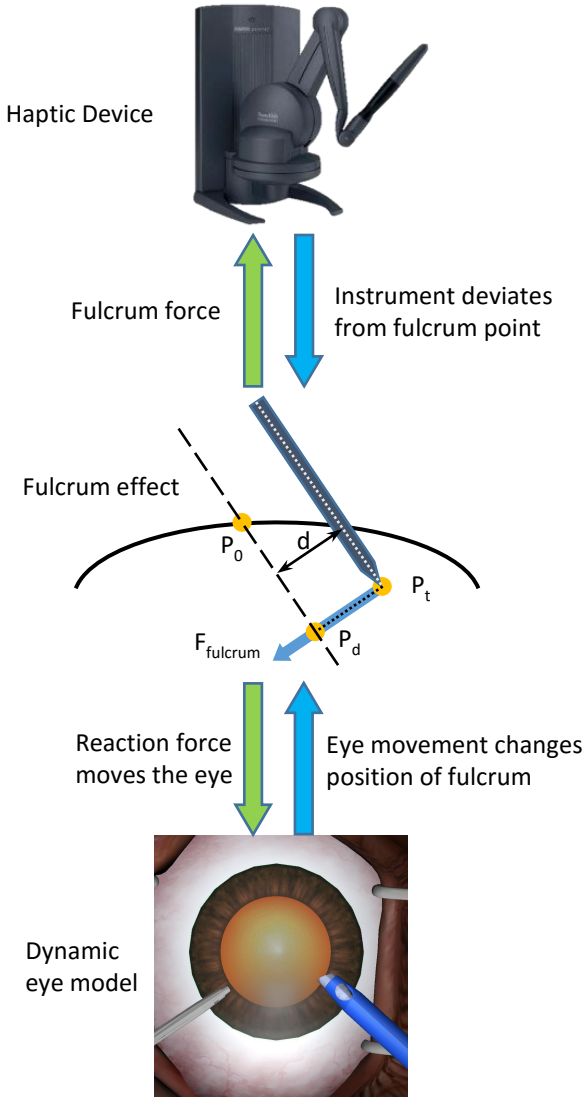

Figure 16. Force interaction mechanism of the eye model

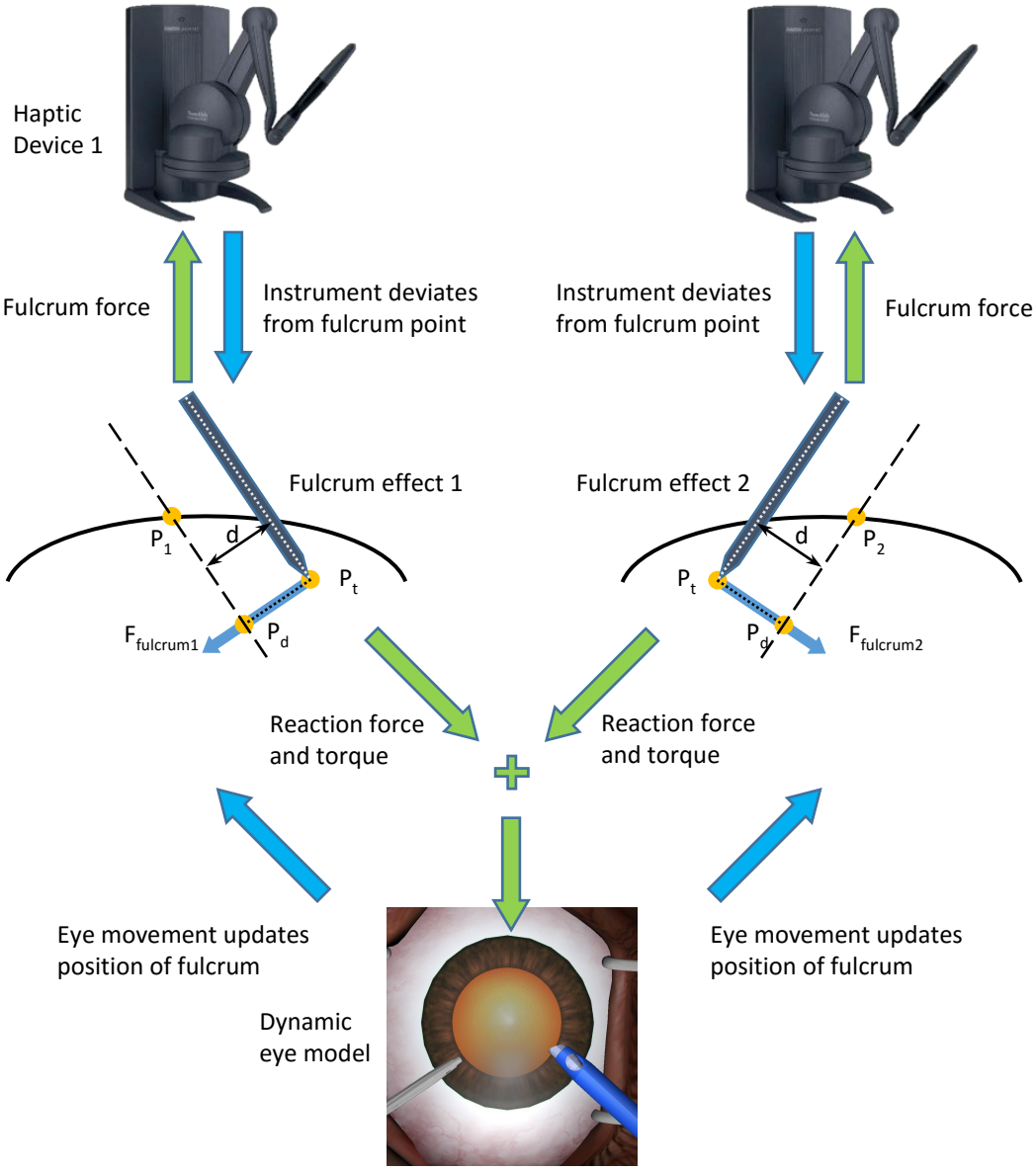

Figure 17. Force feedback mechanism with two haptic devices

4.4 Modeling of Cataract Lens

To simulate the emulsification of cataract lens, the shape of the model needs to be modified during simulation. Regular polygonal meshes, however, can hardly describe the modified geometry, especially the when a lot of changes have taken place. Therefore a 3D model that describes the whole volume of the object is more appropriate for modeling of cataract lens. Two kinds of 3D model – volumetric model and tetrahedron mesh model, are implemented. They are adopted in simulation of different surgical procedures.

Mathematical model of human lens is introduced to precisely define the shape of the 3D mesh. For convenience sake, the rest of this chapter defines the equatorial plane of the lens model as the X-Y plane, and the optical axis as Z axis. According to (Rama et al., 2005), continuous asymmetric bi-elliptical model is commonly used to describe the human lens. Equation (4.11) describes the Y-Z section of the human lens:

$$y^2 + (1 + Q)z^2 - 2Rz = 0 \qquad (4.11)$$

where “y is the radial distance from the surface vertex along the vertex plane, z is the surface sagitta, R is the vertex radius of curvature, and Q is the conic asphericity” (Smith et al., 2009). Moving the ellipse center to the origin, the equation becomes

$$y^2 + (1+Q)z^2 - \frac{R^2}{1+Q} = 0 \qquad (4.12)$$

This equation describes both anterior and posterior halves of the lens with different coefficients R and Q. For anterior surface, $z \geq 0$; for posterior surface, $z \leq 0$. Figure 18 shows the drawing of the Y-Z section of the bi-elliptical model of the human lens using equation (4.12).

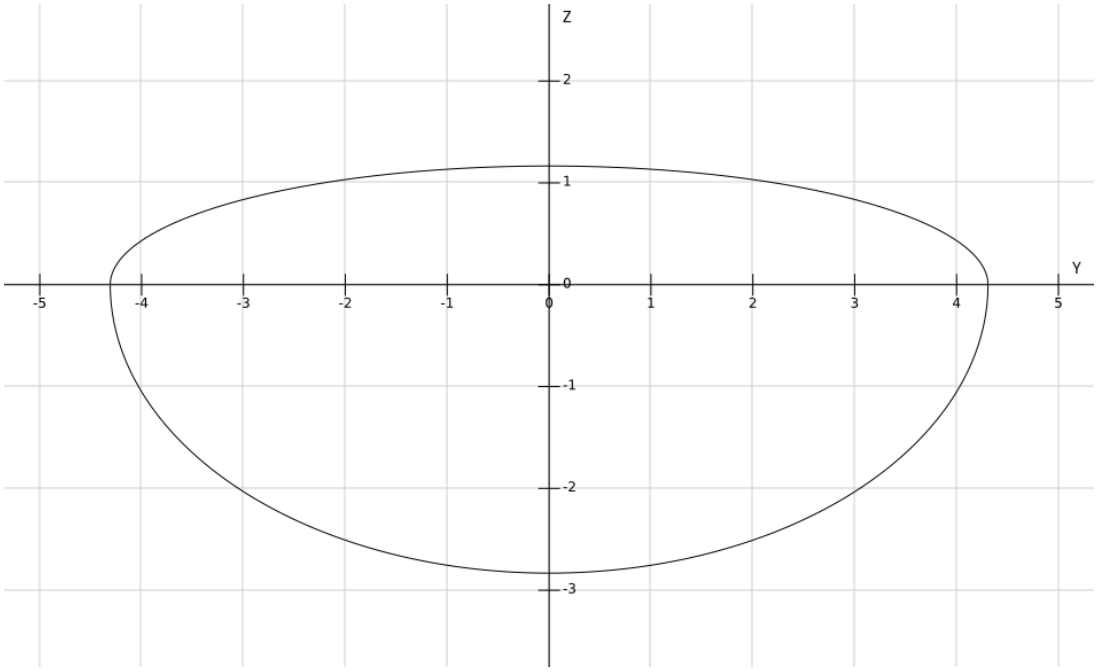

Figure 18. Bi-elliptical model of lens

Due to the rotational symmetry of the lens, the shape of the lens in 3D space is derived from equation (4.12):

$$x^2 + y^2 + (1 + Q)z^2 - \frac{R^2}{1 + Q} = 0 \qquad (4.13)$$

This mathematical model is used to guide the construction of the 3D mesh of human lens. By adjusting values of R and Q, this mathematical model is able to describe various human lenses with different ages.

4.5 Volumetric Model

The volumetric model is one of the 3D models we use to represent the human lens in the virtual scene. It contains a 3 dimensional cubic grid, with each volume element, or voxel represented by a scalar value. The shape of the model can be obtained by defining an isosurface, which is a surface that represents all points of a constant value (i.e. isovalue) within the volume.

4.5.1 Model Construction

The volumetric model of lens can be generated by stacking up the cross sections of the lens geometry in different layers. The cross sections of the lens are circles, and the radii in different layers can be derived from equation (4.12), as the following:

$$r = \sqrt{\frac{R^2}{1+Q} - (1+Q)z^2} \tag{4.14}$$

where r is the radius of the circle at layer z.

By assigning one constant value c_1 to the voxels inside the lens, and another constant value c_0 to the voxels outside the lens for all 2D slices, the volumetric model of lens is achieved. The shape of the lens can be represented by an isosurface with the isovalue between c_0 and c_1.

4.5.2 Volume Rendering

Marching cubes algorithm (Lorensen et al., 1987) is used to render the volumetric model. It extracts the polygonal mesh from the isosurface. The algorithm contours the whole volume, taking eight neighbor voxels at a time, forming a small cube. By comparing the isovalue with the value of these 8 voxels, the portion of the isosurface passing through this cube can be determined, and polygons are generated to represent the isosurface. Combining polygons from all small cubes, the entire surface of the lens is achieved. Figure 19 shows the graphics rendering of the volumetric model using marching cubes algorithm.

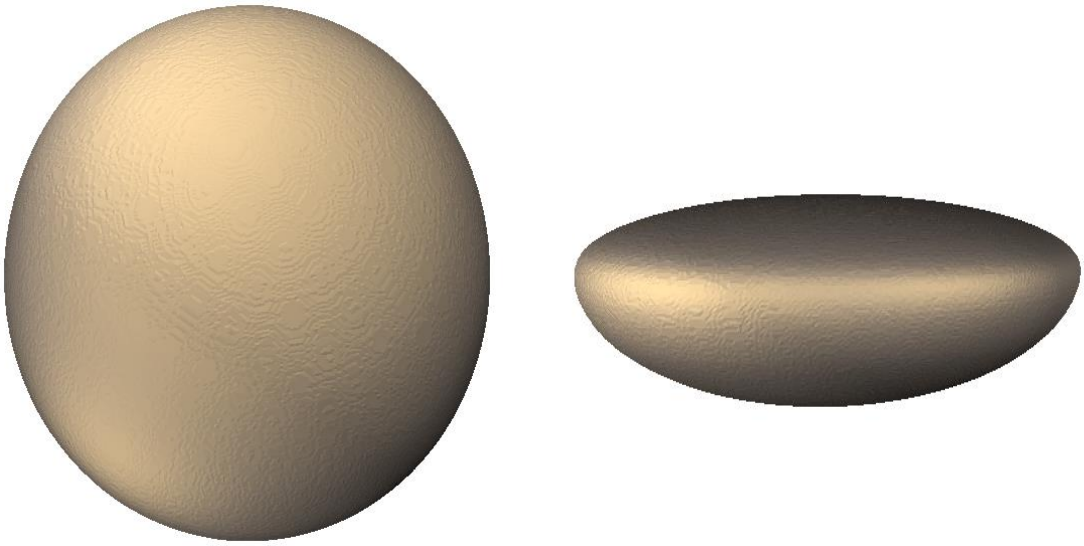

Figure 19. Volumetric model of human lens

Marching cubes algorithm usually generates the surface mesh with a large number of polygons. However the performance of OpenHaptics is poor for shapes with too many polygons. To address this issue, a haptic rendering algorithm (Rizzi et al., 2007) which provides haptic feedback directly from the volumetric model is used. It is an efficient algorithm as it only needs a small portion of voxels near the haptic interaction point to detect collision with the shapes and generate force feedback.

As described in Section 4.5.1, the volume is composed of voxels with two arbitrary values c_0 and c_1. In this case, the marching cubes algorithm may extract

the sharp edges from all small cubes that locate on the surface of the lens geometry, which does not comply with the smooth ellipsoid-like shape of the actual lens. To achieve a smooth surface for both visualization and haptic rendering, Gaussian smoothing needs to be applied to the initial volume. The equation of a Gaussian function in one dimension is shown in equation (4.15):

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{4.15}$$

where σ is the standard deviation of the Gaussian distribution. To apply smoothing in three-dimension space, the products of Gaussian functions on all x, y and z dimensions are needed:

$$G(x, y, z) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \tag{4.16}$$

Values from this distribution are used to build a convolution matrix, which is then applied to the original volumetric data. This can be simplified by applying a precomputed Gaussian kernel to the voxels. The new value of a voxel is calculated as a weighted average of its neighboring elements. The larger the distance to the center voxel, the smaller weight is assigned to the neighboring voxel; the original value from the center voxel has the largest weight. This smoothing kernel can be effectively applied to the volumetric data by parallel computing using GPU.

4.5.3 Run Time Mesh Modification

The values of the voxels can be modified in run time, and the isosurface changes accordingly. This is a convenient way to update the shape of the lens during the simulation of phaco grooving. When phaco power is turned on, the voxels near the tip of phaco handpiece are modified. Since the haptic rendering algorithm only relies on the isosurface, it automatically adjusts the force output to reflect the changes. After reapplying the marching cubes algorithm to the modified volume, the polygonal mesh of the lens surface is also updated. To reassure a smooth surface in graphics and haptics, the Gaussian smoothing needs to be applied again within the subdivision of the volume that contains the voxel changes.

4.6 Soft Body Model

"Soft body" is another kind of 3D model representation that uses a polyhedral mesh to describe not only the surface of the virtual object, but the entire volume of the geometry. Just like polygonal mesh, which describes the surface as a set of adjacent polygons, polyhedral mesh represents the volume by a group of adjacent polyhedrons. The most commonly used polyhedral mesh is the tetrahedral mesh, which is composed only by tetrahedrons.

4.6.1 Mesh Construction

An algorithm of 3D mesh generation (Rineau et al., 2007) is used to construct the tetrahedron mesh of the human lens. It is implemented based on

the method of Delaunay refinement introduced by (Chew, 1993) and (Ruppert, 1995) and pioneered in 3D by (Shewchuk, 1998). This algorithm generates isotropic simplicial meshes discretizing 3D domains, bounded and representable as a set of faces. Several quality criteria including facet size, facet angle and cell size may be applied when generating the meshes.

When building the tetrahedron mesh of the human lens, a triangle mesh with all vertices satisfying Equation (4.13) in Section 4.4 serves as the surface boundary. The meshes with different tetrahedron density can be generated by configuring the size of tetrahedrons.

4.6.2 Graphics Rendering

A series of graphics rendering algorithms for tetrahedron meshes have been implemented. The naive algorithm is to render all four triangles of each tetrahedron. This algorithm is capable of visualizing the inner structure of the tetrahedron mesh. It keeps a fair representation of the shape surface even when some tetrahedrons are removed from the mesh. However, rendering vertices and triangles inside the shape surface takes extra time and is usually unnecessary. Moreover, when the mesh is rendered with semi-transparent material, all inner triangles become visible, which in turns leads to an undesired broken-glass-like visual effect.

Another graphics rendering algorithm is surface mesh binding. This algorithm uses a polygon mesh as shape surface and binds it to the superficial vertices on the tetrahedron mesh. The bound surface mesh deforms along with the movement and distortion of tetrahedrons. The advantage of this algorithm is that it can effectively render the mesh with much better visual effects, as the surface polygon is capable of providing much more details by having higher vertex density and rendering with vivid textures. However, this algorithm cannot properly render the mesh when tetrahedrons are dynamically modified or removed, as the polygon mesh only describes the original surface and is fixed with the vertices on the surface of the tetrahedron mesh.

In the simulation of cataract surgery, the model of human lens is not always opaque, and the shape needs to be changed in run time. Therefore, neither of the above algorithms are appropriate. A new algorithm has been implemented to render the dynamically changed geometry.

Due to the adjacency of the tetrahedral mesh, all triangles inside the mesh are always shared by two tetrahedrons. If we put all triangles from every tetrahedrons together, each inner triangles must appear twice, with opposite normal directions. Similarly, a triangle is on the mesh surface if and only if it appears only once. The surface rendering algorithm uses this feature to distinguish all external triangles of the tetrahedral mesh. At the initialization stage of the simulation, the algorithm traverses through all triangles, and identify each

triangle by indices of its three vertices. The triangles should be consider to be the same if all three indices match regardless of the order.

OpenGL uses the order of the vertices to distinguish the orientation of triangle. For three indices a, b, and c, there are 6 permutations, namely {a,b,c}, {a,c,b}, {b,a,c}, {b,c,a}, {c,a,b}, and {c,b,a}. Among them {a,b,c}, {b,c,a}, and {c,a,b} represent the triangle with the same normal direction (hereby defined as “up”), while {a,c,b}, {b,a,c}, and {c,b,a} denote the triangle facing the opposite direction (defined as “down”). Therefore, the triangle can be described by a sorted indices set with a specified normal direction (“up” or “down”).

A map container of C++ Standard Template Library (STL) is established to distinguish the surface triangles. The sorted indices set is used as the key to identify the triangle in the map. The value of map element enumerates the normal direction(s) of the triangle(s) in the tetrahedron mesh. Each elements has one of these four possible values: “0” (indicating the triangle with this indices combination does not exist), “up”, “down”, “up+down” (indicating triangles with both up and down normal exist in the map). After adding all triangles to the map, all elements with the value of “up+down” represents the triangles inside the tetrahedron mesh, which are ignored for graphics rendering. The map elements with the value of “up” or “down” denotes the triangles on the surface, which are to be rendered. For the element with the value “up”, a triangle with ascending indices {a,b,c} is added to the surface polygon, while a descending set {c,b,a} is

added when a map element has a value of “down”. In this way, all triangles on the surface are properly rendered.

When the mesh is modified during simulation, one or more tetrahedrons may be removed. To update the surface mesh accordingly, the map elements should be modified to reflect the changes. The elements whose values change from “up” or “down” to “0” denote the surface triangles that belong to the removed tetrahedrons, which should be removed from graphics rendering. Meanwhile, all triangles whose values change from “up&down” to “up” or “down” are the ones that just emerged from the inside, and should be added to the rendering list.

To sum up, this algorithm is efficient as it drastically reduces the number of triangles to be rendered. In addition, this algorithm is compatible with all semi-transparent visual effects, as no inner triangle is involved in graphics rendering. Moreover, this algorithm can properly render the updated surface when the tetrahedron mesh is modified in run time.

4.6.3 Subtriangulation Shader

The graphics rendering algorithm described in Section 4.6.2 provides a way to generate a polygonal mesh to represent the surface of the tetrahedral model. The level of detail of the polygon mesh is however limited by the number of vertices on the surface. Moreover, small fragments separated from the model may be composed of only a few tetrahedrons. The sharp angles of tetrahedron

may not necessarily reflect the actual geometry of the fragments. To improve the visual quality of the surface mesh, a subtriangulation shader is implemented using OpenGL shading language (GLSL). The shader uses Curved PN Triangles Algorithm (Vlachos et al., 2001) to refine the triangles into curved patches with higher order of normal variation.

The shader uses the three vertices $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, and three vertex normal $\boldsymbol{N}_1$, $\boldsymbol{N}_2$, $\boldsymbol{N}_3$ of a triangle as input. It substitutes the geometry of the flat triangle with a cubic Bezier triangle, which can be described by the following equation:

$$\boldsymbol{p}(s,t,u) = \sum_{i+j+k=3} \boldsymbol{p}_{ijk} \frac{3!}{i!\,j!\,k!} s^i t^j u^k \qquad (4.17)$$
$$s.t. \quad 0 \le s,t,u \le 1, s+t+u = 1$$

where $\boldsymbol{p}_{ijk}$ are the control points of the Bezier triangle, and s, t, u are the barycentric coordinates inside the triangle. (Vlachos et al., 2001) described an algorithm to determine the control points of the curved PN triangles, as the following.

First, place all $\boldsymbol{p}_{ijk}$ evenly at the intermediate positions of the input triangle. Vertex coefficients $\boldsymbol{p}_{300}$, $\boldsymbol{p}_{030}$ and $\boldsymbol{p}_{003}$ represents the vertices of the original triangle. Their position should remain unchanged. Six tangent coefficients locate at trisection points of all three edges. They are moved to the projection on tangent plane defined by the normal of the nearest vertex. Finally, the center

point $\boldsymbol{p}_{111}$ should be moved to the average of all six tangent coefficients, and continue its motion for another half of the distance traveled in the same direction. Figure 20 illustrates the curved surface generated by this algorithm.

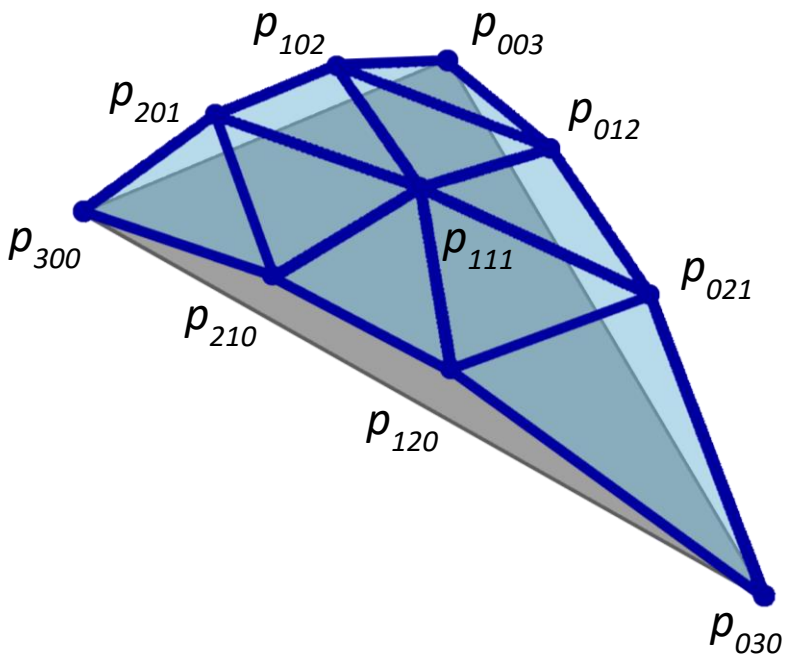

Figure 20. Subtriangulation using curved PN triangles

In addition, a quadratically varying normal n (Equation 4.17) is computed for smooth shading of the curved PN triangle.

$$\boldsymbol{n}(s,t,u) = \sum_{i+j+k=2} \boldsymbol{n}_{ijk} s^i t^j u^k \quad (4.17)$$
$$s.t. \quad 0 \le s,t,u \le 1, s+t+u = 1$$

Among all 6 control points, $\boldsymbol{n}_{200}$, $\boldsymbol{n}_{020}$, and $\boldsymbol{n}_{002}$ are the corner coefficients, which should match the normal of three vertex of the input triangle. The intermediate normal coefficients $\boldsymbol{n}_{110}$, $\boldsymbol{n}_{011}$, and $\boldsymbol{n}_{011}$ locate at the midpoint of each triangle edges. A naive way to compute this mid-edge normal is the linear interpolation of the vertex normals on both sides, however this may ignore the shape variation of the curved edge. (Van Overveld et al., 1997) suggests that the mid-edge normal should be constructed by averaging the vertex normals and reflect it across the plane perpendicular to the edge.

4.6.4 Soft Body Dynamics

Soft body dynamics (Nealen et al., 2006) uses a set of particles to simulate the motion and physical properties of the soft bodies. The particles are located at every vertices of the tetrahedron mesh. The edges of the tetrahedrons defines the spring-damper constraints between the particles, which enable the deformation of the mesh while retaining its shape to some degree. When a shape interacts with the soft body, collisions forces deforms the mesh by moving the particles away from the rest position. After the interaction, spring-damper constraints restore the tetrahedrons back to their original shapes. The geometry of the soft body mesh can be modified by removing tetrahedrons together with the corresponding particles and spring-damper constraints. When the mesh is

separated, each piece may move individually, since there is no constraints between different pieces.

The haptic rendering of tetrahedron mesh relies on the soft body dynamics. A rigid body is bound to the haptic stylus by a spring-damper model. When a force is applied, the rigid body deviates from the rest position, which in turn pulls the spring-damper and generate the haptic effect. To achieve haptic rendering of the soft body meshes, two-way interaction between soft body and rigid body is enabled. When the rigid body on haptic stylus collides with the soft body, a response force is applied to the rigid body, and eventually it is converted to the actual force and output to the haptic stylus.

4.7 Comparison of the Lens Models

As illustrated in Section 4.5, the volumetric model uses 3D discretely sampled data set to define the virtual object. Meanwhile, the soft body model introduced in Section 4.6 uses adjacent polyhedrons to describe objects. Both of these models can be used to simulate the removal of cataract lens since they are capable of real-time modification of the geometry. The comparison between these two models are presented below.

The soft body model outperforms the volumetric model in the following ways. First of all, the soft body model is composed of a series of tetrahedrons. When the whole mesh divides into multiple fragments, each piece can move

freely and independently. This is not the case for the volumetric models. The output of the marching cubes algorithm is a single surface mesh. Even if the lens is cracked into separated pieces, they are still considered as one mesh and the relative positions between the pieces are fixed.

Besides, the actual human lens is a soft tissue which may deform under pressure, and recover the original shape when force is removed. For the soft body model, the spring-damper model between vertices may properly simulate this deformation. However, the surface mesh generated by volume rendering is irrelevant to the external force and thus the shape is rigid.

Additionally, the computational complexity of updating geometry of the soft body model is faster than that of the volumetric model. For soft body model, the update involves only removing several triangles from the surface mesh as well as adding a few new triangles to it. However, volumetric model requires the execution of the marching cubes algorithm to regenerate the triangles of the surface mesh. This brings a big overhead during graphics rendering and may decrease the frame rate of the simulation and even affect the overall performance.

On the other hand, the volumetric model has several advantages compared with the soft body model. First of all, the haptic rendering of the volumetric model is precise and effective. The force is computed in the servo loop, which enables a

more realistic haptic feedback. However the physical simulation of the soft body model is computationally intensive, especially when multiple shapes interact with the soft body model simultaneously.

Besides, compared with the soft body mesh, volumetric model describes the geometry with higher level of detail. The number of voxels in the volumetric model depends on the dimension of the cuboid dataset. For example, the volume dimension of the cataract lens model is 256 by 256 by 128, making the total number of voxels to be 256×256×128, i.e. 8,388,608. For soft body models, due to the computational intensity of physics interaction, the maximum number of vertices is limited to several thousands. Therefore, the volumetric model may potentially provide more details in the geometry. This is more noticeable when simulating the lens sculpting, which brings high complexity to the lens geometry.

Based on the difference discussed above, the two kinds of model are adopted in simulation of different procedures. The simulation of lens sculpting (Section 5.5) chooses the volumetric model as it provides more details. The simulation of phacoemulsification (Section 5.6) uses the soft body model as it is capable of being separated into individually mobile fragments.

4.8 Red Reflex

In the actual surgery, a red reflex, which refers to the reddish-orange reflection of light from the eye's retina, can be observed on the lens. When the

lens tissue gets thinner during phacoemulsification, the rex reflex become more obvious. Therefore this visual effect is usually used as an indication of the remaining depth of the lens during lens sculpting.

To achieve the red reflex effect in the simulation, a per-vertex color buffer is used. The graphics material assigned to each vertex is determined by the remaining depth of lens at that position. By interpolating the colors of all vertices, the surface mesh will be rendered with a gradually changed color that reflects the lens thickness.

The remaining thickness is obtained by computing the distance between the vertex $V(\mathrm{x}_v, \mathrm{y}_v, \mathrm{z}_v)$ and its orthogonal projection $P(\mathrm{x}_p, \mathrm{y}_p, \mathrm{z}_p)$ on the lens posterior surface. As the projection line is parallel to the lens optical axis (i.e. Z axis), the vertex and its intersection point should have the same X and Y coordinates, and the remaining thickness equals to the difference of Z coordinates:

$$d = z_v - z_p \tag{4.17}$$

The Z coordinate of the projection point z_p can be computed by solving the lens posterior surface equation 4.12, as follows.

$$\mathrm{z}_p = -\frac{\sqrt{R^2 - (1+Q)(x_v^2 + y_v^2)}}{(1+Q)} \tag{4.18}$$

Finally, the color at each vertex $C_v(r, g, b, \alpha)$ is computed by linear interpolation, as the following:

$$C_v = \frac{d}{d_L} C_L + \left(1 - \frac{d}{d_L}\right) C_R \tag{4.19}$$

where d_L is the thickness of the lens, $C_L(r, g, b, \alpha)$ is the original color of the cataract lens, and $C_R(r, g, b, \alpha)$ is the color of red reflex.

4.9 Phaco power

Phaco power is the mechanical energy generated by the phaco needle for breaking up the cataract. It "represents a combination of frequency and stroke length of the needle motion. The frequency refers to the speed the needle changes the direction of movement. It is measured in cycles per second, or Hertz. The stroke length represents the actual distance the phaco needle travels during vibration" (Fine et al., 2004). The stroke length of the phaco tip can be controlled by surgeons using phaco pedal.

The phaco power enables when foot pedal reaches position 3. Depressing the pedal further increases the phaco power. The full pre-set power is achieved when full excursion of the pedal is reached. In the MicrovisTouch simulation platform, the phaco pedal outputs its position by an analog signal. The HID hub periodically reads the signal and from the pedal and sent to the simulation application.

The pedal controls the level of phaco power, which determines the ease of removing materials. Therefore, in the simulation the reading from the pedal determines the speed of geometry change. In the simulation that uses volumetric lens model, the pressed distance of the phaco pedal is proportional to the reduction of the scalar values of in voxels. It also determines the size of region where voxel modification takes place. In the simulation using soft body model, the time interval of the soft body removal is determined by the pressure applied on the pedal.

When phacoemulsification is enabled, a vibration haptic effect is added to the haptic stylus to emulate the tactile sensation of the actual phaco handpiece. The frequency of vibration changes along with the level of the phaco power. As the frequency falls within the acoustic zone, the vibration on the haptic stylus also serves as a sound feedback indicating of the level of phaco power.

5 SURGICAL PRECEDURES SIMULATION AND TRAINING MODULES

5.1 Micro-dexterity practicing module

5.1.1 General purpose

Tremor is a shaking movement which usually appears in hands and arms. Everyone has some hand tremor, however surgeons are required to keep their hands steady during surgery. This “micro-dexterity practicing module includes a hand-eye coordination task that helps trainees identify the tremor and improve the hand stability” (Luo et al., 2016). This is a preliminary training module for the cataract surgery.

5.1.2 Overall design

The main interface of the micro-dexterity practicing module is shown in Figure 21. The 3D eye model set is used to emulate a scenario of the eye surgery. A tiny sphere is moving along a predefined path. The objective of this simulation is to move the instrument together with the sphere and keep the tip of the instrument close to the center of this moving sphere. The color of the sphere changes as an indication of the distance between the instrument and the sphere. The default path for the sphere is a circle which locates inside the eye and right above the lens.

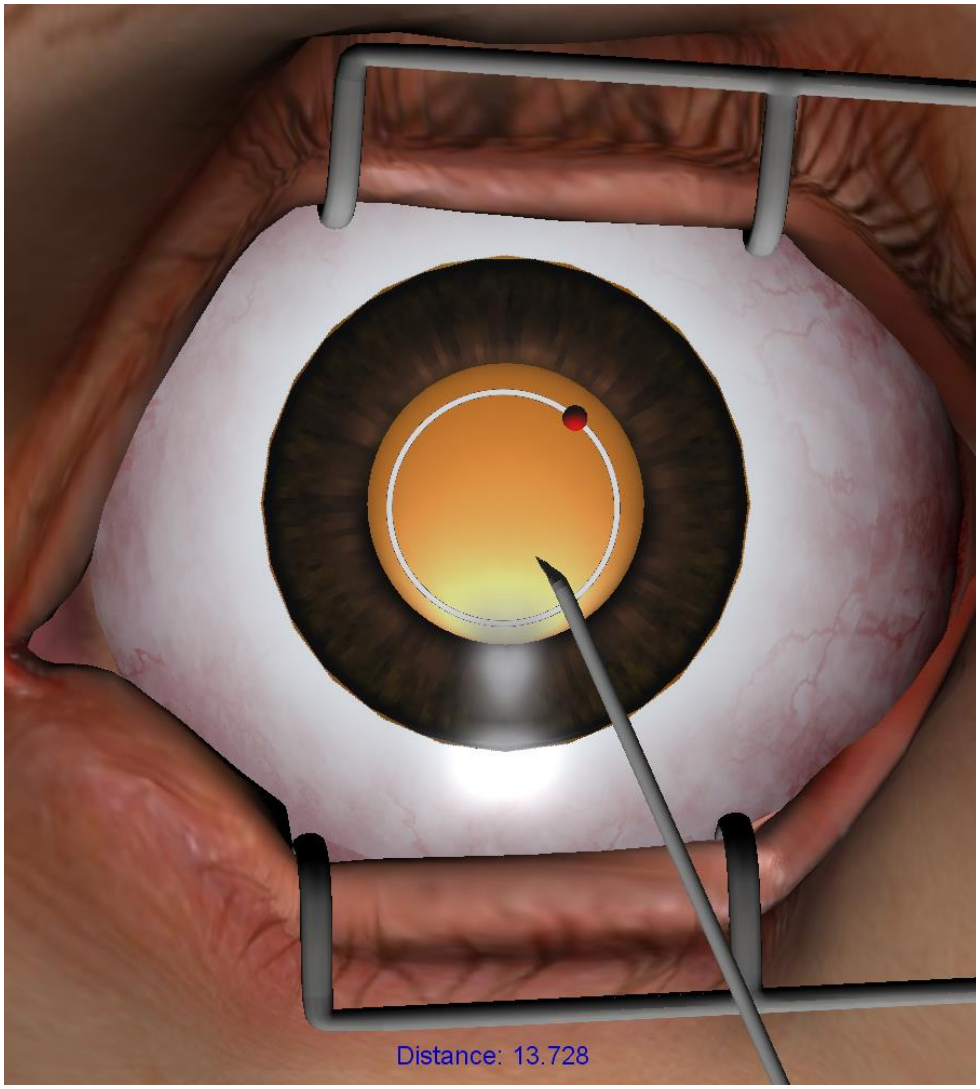

Figure 21. Interface of the micro-dexterity practicing module

5.1.3 Task customization

The application provides a variety of options to customize the training task. The path of the moving sphere can be customized to accommodate the need of mimicking any type of the motion of the surgical instruments. Besides, the speed of the sphere can be adjusted for different level of difficulty. If certain parts of the

eye model interfere with the customized training task, they can be rendered with semitransparent colors, or even removed from the scene.

In this application, the haptic feedback of the eye model is disabled by default in order to have user focus on hand-eye coordination. If a customized practicing scenario requires haptic feedback of any specific parts of the eye model, they can be turned on separately. In addition, the haptic properties including stiffness, friction or pop-through factor can be modified at the user interface.

5.1.4 Graphical user interface

A graphical user interface (GUI) (Figure 22) has been designed for this application. It is implemented using the Fast Light Toolkit (FLTK), a cross-platform C++ GUI toolkit which “supports 3D graphics via OpenGL and its built-in GLUT emulation” (FLTK, 2012).

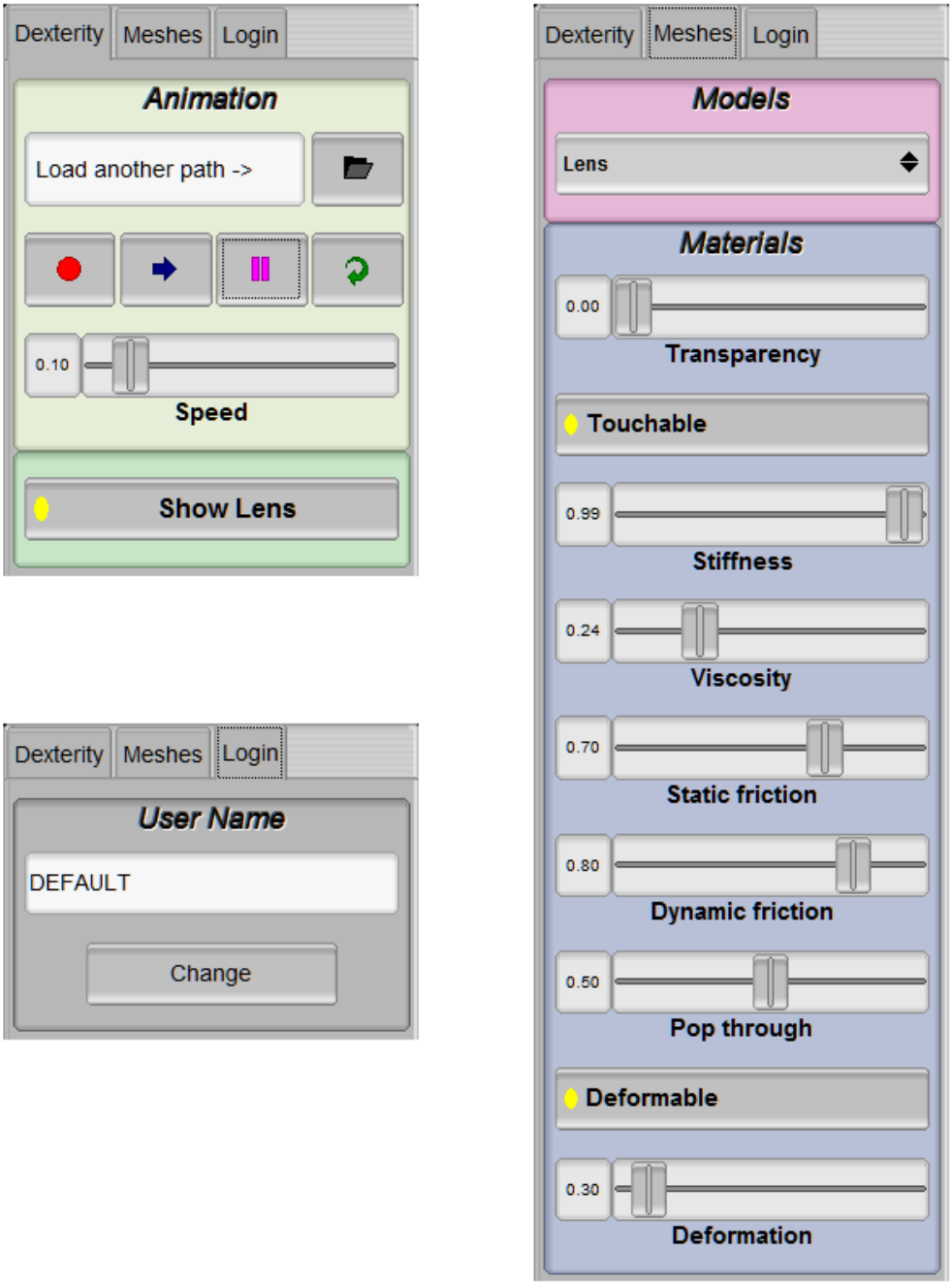

Figure 22. Graphical user interfaces

The GUI works as a control panel that allows users to adjust the settings of the simulation. The GUI includes three tabs. The “Dexterity” tab is used to control the sphere animation, including changing paths, toggling moving direction and adjusting speed. It is also responsible for initiating the performance evaluation. The “Meshes” tab provides several sliders which manage the haptic and visual properties of every parts of the 3D models. The “Login” tab allows user to specify a username, which serves as an identifier in the performance records.

5.1.5 Performance Assessments

The training performance can be recorded and evaluated. During recording, the application will keep measuring the distance between the tip of the instrument and the running sphere in every frames of the simulation. The recording stops when the sphere finishes moving a full cycle. The mean and standard deviation of all the distance data is then calculated. The mean of the distance indicates the level of user’s hand-eye coordination, and the standard deviation measures the evidence of a tremor. A grade will be given based on the calculation, and displayed on the screen. The scores are also saved into the database which collects all the historical users’ performance.

5.2 Eyeball balancing practicing module

5.2.1 General Purpose

Eyeball balancing is a basic while important technique of ophthalmic surgeries including the cataract surgery. Surgical instruments should be carefully

operated in the eye and the force applied to the eye should be limited and balanced. Otherwise, external forces may push the eye away from the center, which moves the surgical workspace outside of the microscope field of view. Additionally, exceeding forces may overly rotate the eye and may potentially enlarge the incision or even damage the eye. A VR software has been developed for the MicrovisTouch platform to help trainees manage the movement of surgical instruments and improve the technique of eyeball balancing.

5.2.2 Overall design

The interface of the eyeball balancing practicing module is shown in Figure 23. The eye model is displayed on the screen. The purpose of this training is to minimize the movement of the eyeball while manipulating the virtual instruments to finish a specific task in the eye. The training can be conducted by either hand, or with the coordination of both hands.

The task shown in Figure 23 is designed to mimic the movement of instrument in capsulorrhexis, a procedure which construct a circular tear on the anterior lens capsule. The trainee is expected to operate one instrument through a corneal incision, and follow the sphere that spins on a circular trajectory within the anterior chamber. As described in Section 4.3, an articulated joint model manages the movement of the eye model. When instrument pushes the incision on the side, a force is applied to the fulcrum and drive movement of the eyeball. To minimize the eye movement, the haptic stylus should be rotated rather than

shifted when moving the instrument along with the sphere. Operator should keep the instrument passing through the center of the incision. A second haptic device can be introduced to emulate the operation of a second instrument, which helps stabilize the eyeball by applying a force opposite to the eye movement.

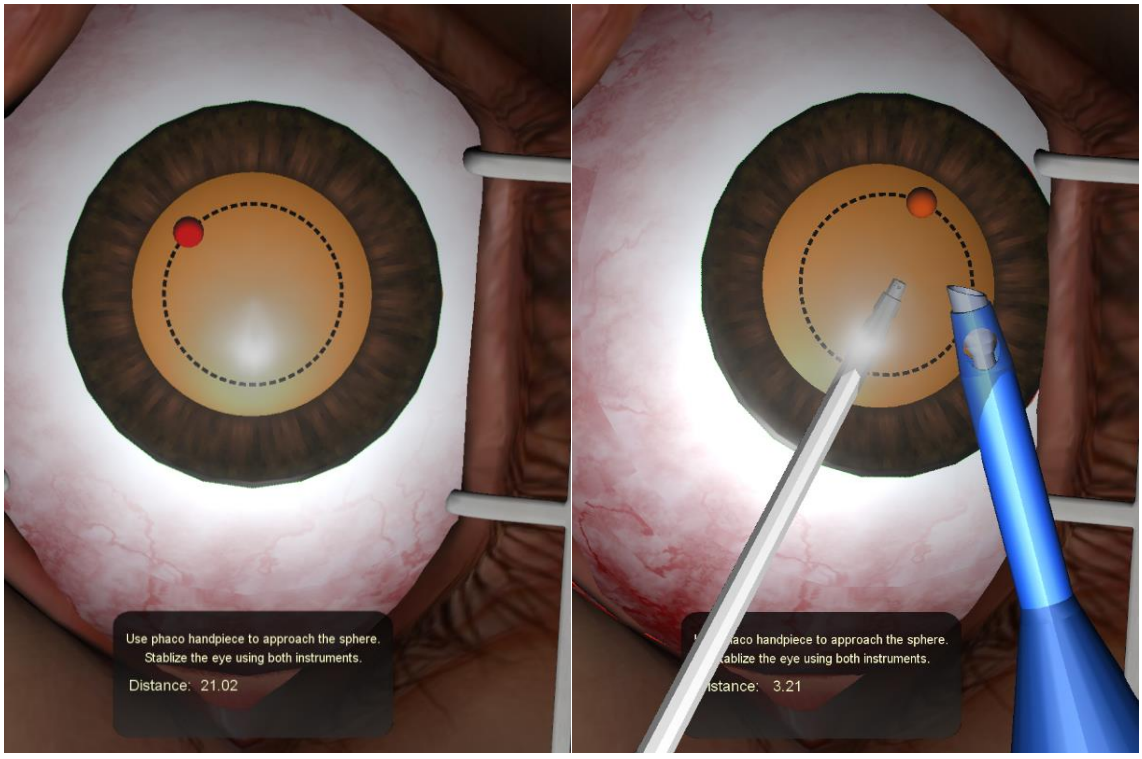

Figure 23. Eyeball balancing practicing module

5.2.3 Performance Assessments

The application calculates the distance between the sphere and the instrument tip to assess the precision of the desired operation. To evaluate the eyeball balancing, the application measures the angular deviation of the eyeball in each simulation frame. Statistics including mean and standard deviation of the angle reflect how much the eye model rotates during the training. Small mean and standard deviation may indicate better eyeball balancing.

5.3 Corneal incision simulation module

This simulation module serves as a training platform for the construction of corneal incisions.

5.3.1 Background information

In phacoemulsification surgery, two incisions should be made on the corneal limbus to allow inserting surgical instruments into the eye. The first incision is constructed at the beginning of the surgery, which provides a channel for instruments to operate within the anterior chamber. The second incision is created after capsulorrhexis, supporting the following procedures that requires simultaneous use of two surgical instruments.

The aim of this procedure is to construct a stepped, self-sealing, clear corneal incision of the correct length. A self-sealing corneal incision avoids fluid leaking, and therefore avert the need for any sutures. “Too short a tunnel may

not self-seal; too long a tunnel may cause increased risk of corneal burns and make access without corneal distortion difficult" (Caesar et al., 2003).

The corneal incision is constructed by a diamond-shaped blade called keratome. To construct a self-sealing incision, a special technique is necessary to shape the incision in a specific pattern. Biplanar pattern is a popular way to construct the incision. "Enter the cornea at the limbus at a shallow angle with the heel of the keratome almost flat on the globe. Once the tip of the keratome is engaged, immediately raise the heel of the blade and pass forwards in the corneal stroma for the first 1.0 mm, before stopping and elevating the wrist to achieve a much steeper angle to penetrate the anterior chamber until the blade is parallel to the iris plane" (Caesar et al., 2003). Figure 24 illustrates the entire instrument trajectory with biplanar pattern in the cross-section view.

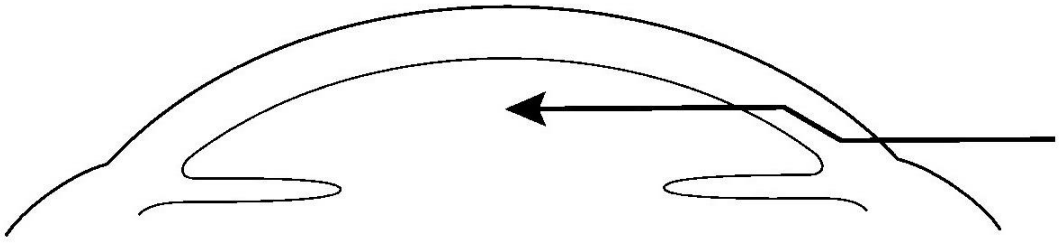

Figure 24. Instrument trajectory with biplanar pattern
(Luo et al., 2016, Copyright © 2016 IEEE)

5.3.2 Application Designs

In the simulation, the virtual model of keratome (Figure 25) is manipulated by the haptic stylus. Compared with straight instruments, whose position and orientation can be easily determined by the 3D coordinates and rotation matrix of the haptic interaction point, the angled blade of keratome brings complication when binding the instrument model with the haptic stylus.

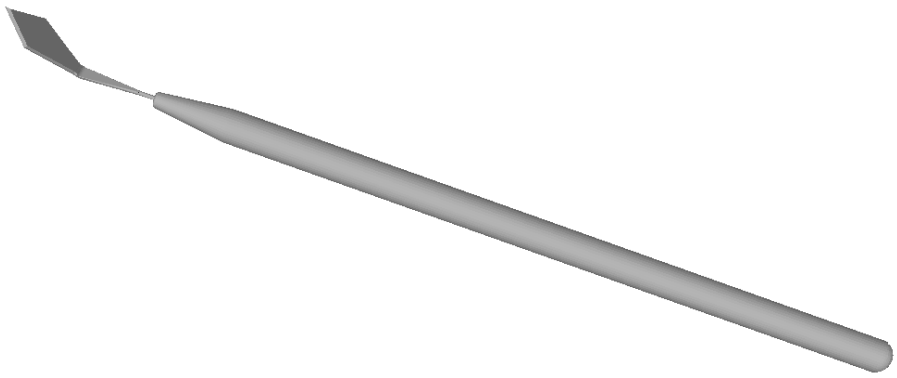

Figure 25. Keratome model

To realistically simulate the use of keratome, the handle should always be aligned with the haptic stylus. As the tip of the keratome is not aligned with its handle, it is not located on the tip of the haptic stylus. To deal with this issue, the

haptic interaction point needs to be moved from the tip of the stylus to the proper position of the instrument tip. The transform is determined by the orientation of the stylus as well as the length and angle of the blade. In this way, the handle of keratome is aligned with the stylus, while the haptic interaction always occurs at the tip of the instrument.

The eye model described in Section 4.1 is used in the simulation of corneal incision. Modifications are made to meet the specific need of this procedure. In this simulation, the movement of keratome within the corneal stroma needs to be recognized and recorded. The simulator should recognize the moment when the blade enters the cornea, and when it is fully engaged. Therefore, in addition to the external surface of cornea that is already included in the eye model, it is necessary to model the internal surface as well. Both surfaces serve as the boundaries of haptic rendering and performance monitoring.

The haptics properties of the cornea model is properly set according to the stiffness and smoothness of the actual human cornea. The friction haptic effect introduced in Section 4.2.3 is used to present the resistance when keratome moves within the corneal stroma. This haptic effect generates a force which opposes the direction of motion and is in proportion to the magnitude of the moving speed. The fulcrum effect which keeps the instrument within the incision is also included with the following modifications. As the keratome blade does not align with the handle, the fulcrum effect is computed based on the direction of the

diamond blade, rather than the orientation of the haptic stylus. Besides, the magnitude of the resistance is reduced due to the sharpness of the blade.

5.3.3 Performance recording and visualization

In the simulation, several colors are used to render the virtual instrument as indications of different events. By default, the keratome is rendered in grey, which is color of the actual instrument. The green color indicates the blade has cut the external surface of the cornea and the tip is within corneal stroma. When the keratome turns blue, it means the tip has entered the anterior chamber. If the instrument accidentally touches the lens or iris, it will be colored in red to remind the user.

The performances of constructing the corneal incision is monitored by recording the keratome trajectory. When the tip of the keratome is engaged into the cornea, the simulator will start recording the position of the blade tip. In every frame of the simulation, a new position is recorded. A series of line segments that connects all recorded points is rendered as the trajectory of the keratome. After the keratome penetrates into the anterior chamber, the recording stops and the complete trajectory is available. The green curve in Figure 26 shows the trajectory of the keratome after the incision is fully constructed.

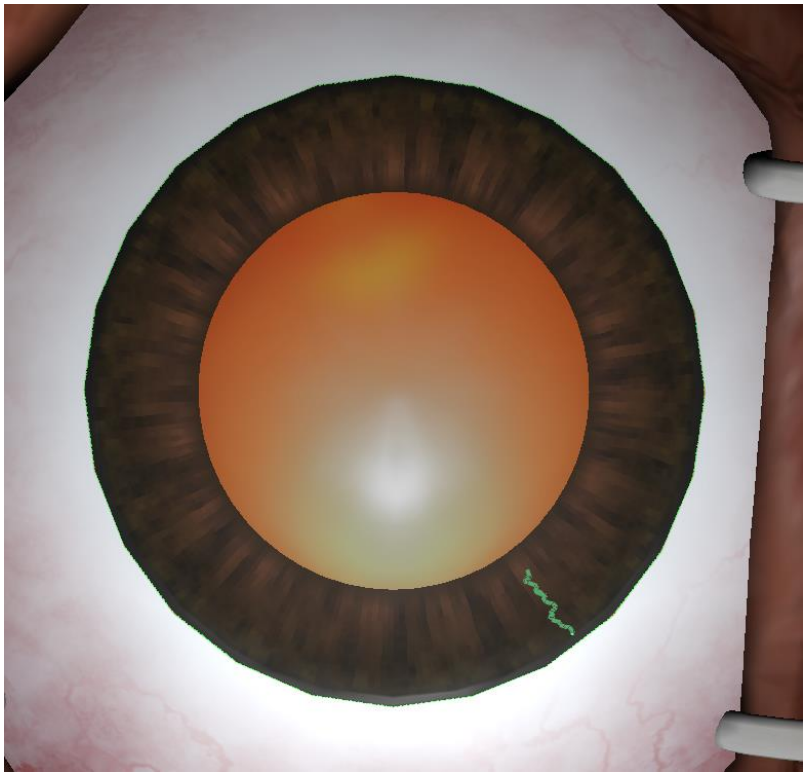

Figure 26. Trajectory of the keratome in simulation

The prospective of the microscope is from above the eye and pointing down. Although the trajectory is rendered in 3D, the shape of the incision may not be clearly shown due to the prospective. Therefore, another two views of the trajectory are implemented. The cross-section view shows trajectory projected to the cross-section plane that contains the zenith of the cornea and the starting point of the incision. Another view is perpendicular to both the microscope view and the cross-section view, showing the trajectory along the direction of the incision. Figure 27 shows these two views of the same keratome trajectory shown in Figure 26.

The visualization of the keratome trajectory is a unique feature in virtual reality-based simulator, as for those actual-based training platform, it is not possible to highlight the trajectory nor alter the prospective.

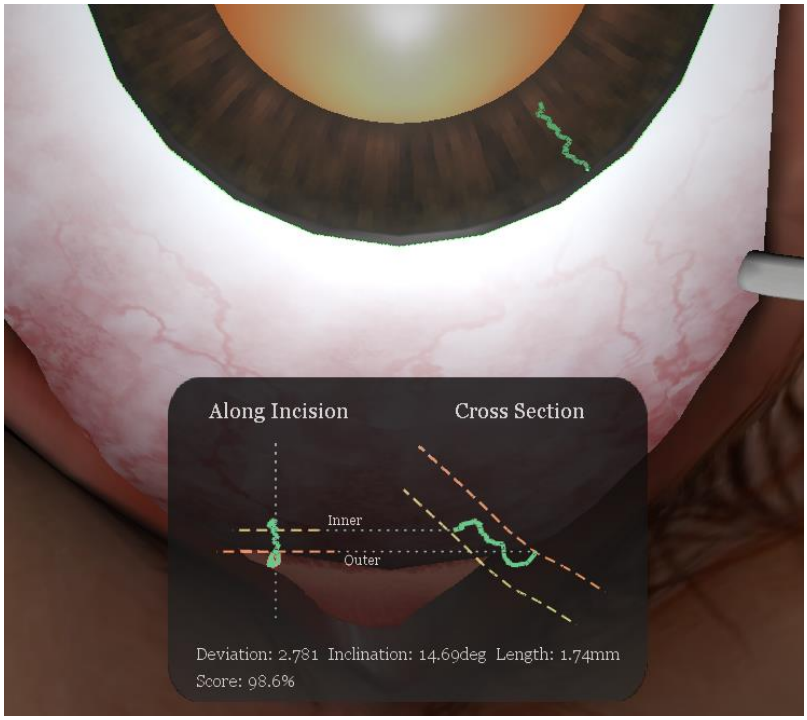

Figure 27 Trajectory in along-incision view (left) and cross-section view (right) (Luo et al., 2016, Copyright © 2016 IEEE)

5.3.4 Evaluation

The simulator enables automatically evaluating the performance in several aspects. First, it measures the length of the incision. The total length of the wound should be 1.5 - 2.0 mm. Second, the simulator measures the inclination of the incision. The middle portion of the incision should be parallel to the cornea, therefore the inclination of the wound should be approximate to the inclination of the cornea in the same area. Third, the simulator calculates the wound's level of deviation from the center line. The corneal incision is supposed to be constructed towards the center of the eye, so the ideal level of deviation should be zero. By comparing the measurements of incision length, inclination and deviation with ideal values, the simulator can effectively assess the quality of the corneal incision.

Besides, the visualizations of the keratome trajectory may assist instructor to evaluate the performance of trainees. The cross-section view of the incision shown in Figure 27 can be used to compare with the ideal keratome trajectory recommended by the instructor, e.g. the biplanar pattern shown in Figure 24.

5.4 Improvements of the capsulorrhexis simulator

The capsulorrhexis simulator described in Section 2.4 is the first software module of the MicrovisTouch platform. Improvements have been made to enhance the realism of the simulator and improve the user experience.

Several changes have been applied to the graphics interface. The warning notification and the depth indication bar are moved into the field of view so that they are more noticeable when trainees focus on surgical procedure. Different colors are applied to the depth bar to distinguish the position of the instrument. To increase realism of simulation, different models of surgical instruments are loaded for their corresponding procedures.

A pressure sensor introduced in Section 3.2 replaces the button which controls the forceps. It allows gradually opening and closing the forceps according to the finger pressure. A serial communication server is added to the software to receive readings from the pressure sensor.

5.5 Phaco sculpting simulation module

This module simulates the surgical procedure of constructing grooves on the cataract lens.

5.5.1 Background information

The aim of phaco sculpting is to “progressively sculpt a set of trenches of optimal depth in the center of the lens nucleus” (Caesar et al., 2003). The walls of each trench are used as the leverage to facilitate easy cracking of the nucleus. Medical institutions may have various preference on the shapes of trenches, including the Swiss cross shape, the Y-shape or the single straight trench. The

trenches must be wide enough to allow the phaco tip and a second instrument to be place side by side, and deep enough to allow cracking.

For the trenches of the Swiss cross shape or Y-shape, rotating the nucleus is necessary after finishing each groove so that the phaco tip can construct the next groove within the range of motion restricted by the corneal incision. To rotate the epinucleus and nucleus together within the capsule, two instruments including the phaco handpiece and the lens manipulator are needed. Therefore this procedure requires two-handed operation.

5.5.2 Overall design

The main interface of this simulation module is shown in Figure 28. Two haptic devices are used to manipulate the phaco handpiece and the lens manipulator. The phaco pedal controls the energy level of phacoemulsification.

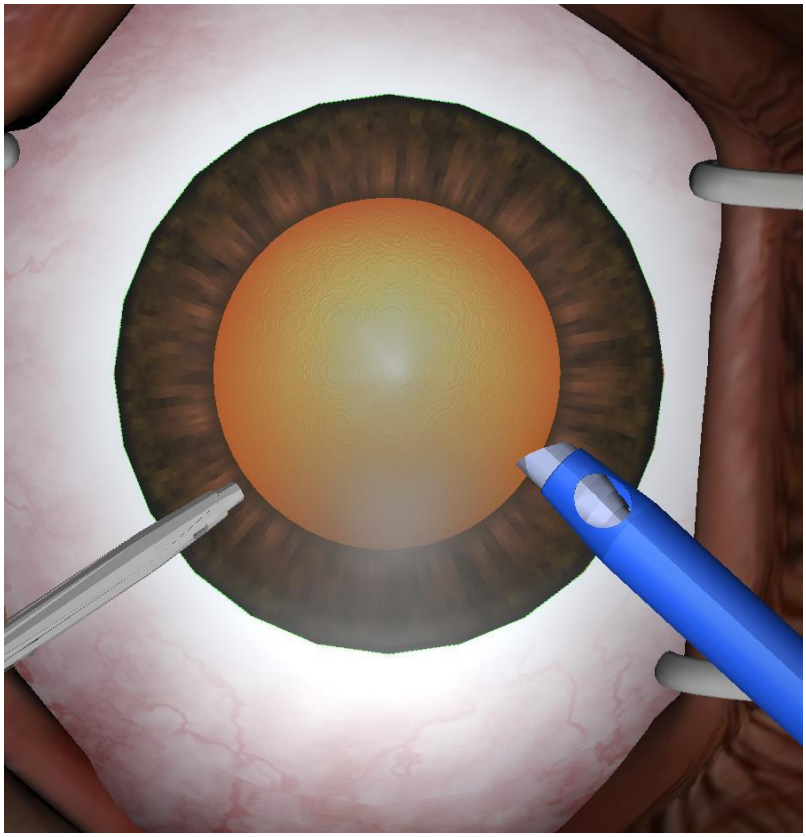

Figure 28. Interface of the phaco sculpting simulation module

Begin the simulation by sculpting the nucleus. Move the tip of phaco handpiece to the center on top of the lens. Depress the phaco pedal to position 3 to enable the phaco power. Move the haptic stylus gradually to advance the instrument towards the side of the lens to construct a groove. Release the pedal from position 3 to power off the phaco power on the backstroke. Repeat the operation till the groove reaches the desired depth. When the first trench is finished, rotate the lens using both instruments by pushing the inner walls of the

groove. After rotating the lens with desired angle, sculpt another groove. Repeat the operation until all grooves are sculpted.

5.5.3 Simulation of lens sculpting

During the simulation of lens sculpting, the geometry of the lens model is dynamically modified. Chapter 4 introduced two kinds of modeling method and corresponding algorithms for the cataract lens model. This simulation module uses the volumetric model, as it is capable of showing more details of the grooves.

When phaco power is switched on, the lens tissue near the phaco needle is emulsified and removed. To simulate this, the software computes the position relationship between the volumetric model and the tip of the phaco handpiece. The voxels are modified when they are close enough to the instrument tip. To keep the continuity of the haptic surface, the Gaussian smoothing algorithm is applied to the modified area. The graphics representation of the lens is updated by applying the Marching cubes algorithm to the modified volume.

The rotation of the lens is achieved by physical simulation. To apply physical simulation to the volumetric model, a phantom object (i.e. a shape excluded from visualization and haptic rendering) is loaded into the virtual scene, overlapping the cataract lens. The lens phantom is linked to the eye model through a revolute joint, which binds their relative position and only allows the

lens to rotate along the axis perpendicular to its equatorial plane. The transform of the volumetric lens model is connected to the phantom shape, therefore the movement of the phantom always applies to the lens model. In this way the lens model can be rotated by external force while remaining in its position in the eye.

In the actual surgery, the lens nucleus is rotated when two instruments push both ends of a groove to the opposite direction. When an instrument pushes against the groove from inside, volume haptic rendering generates a normal force to the haptic device to keep it from going further. At the same time, a reaction force is applied to the lens phantom on the opposite direction, which generates a torque applied to the revolute joint. The sum of the torques from both instruments eventually drives the lens rotation. Spring-damper model is applied to the joint to simulate a smooth rotation.

5.6 Phacoemulsification simulation module

This application simulates a series of procedures in phacoemulsification, including sculpting a Swiss cross shape groove, crack the lens nucleus into four quadrants, then emulsify and remove each quadrant one by one.

5.6.1 Background information

The purpose of phacoemulsification is to safely remove the lens through a small incision. A small incision has many benefits including no need of sutures, faster visual rehabilitation, a significant less morbidity, not inducing unwanted

astigmatism and so on. To remove the cataract lens without constructing a large incision, it is necessary to separate the lens into small fragments within the eye.

Phaco sculpting creates a Swiss cross shape trench in the center of the lens nucleus. The grooves provide mechanical advantage for cracking the nucleus. Use phaco handpiece and lens manipulator to separate the nucleus into four quadrants. The quadrants should be individually mobile so that in advance of being emulsified, each piece can be pulled into the central safe zone to avoid damaging the iris or the lens capsular bag. By applying the high vacuum and phaco power to the lens fragments, the fragments of cataract lens are fully emulsified and removed from the eye.

5.6.2 Overall design

This simulation requires manipulating two virtual instruments by a pair of haptic devices. Phaco pedal is included to simulate the control of phaco power. Head mannequin can be used to locate the workspace in the virtual scene and provide wrist support. The haptic-based 3D eye model is included to provide tactile feedbacks. Position-based dynamics simulates contacts between objects and manages the eye rotation.

Phacoemulsification dynamically modifies the geometry of cataract lens. Besides, this procedure separates the lens nucleus into multiple pieces that

move individually. To achieve both of the above effects in the simulation, the soft body model is chosen to describe the cataract lens.

The lens model is simulated by the soft body dynamics of PhysX. The physics engine enables object-to-object collision detection, which improves the realism of interaction between virtual objects. Due to the level of computational complexity, it does not support contacts between the soft body and complex shapes including triangle mesh, cloth or another soft body. Only simple shapes including spheres, cuboids and capsules are able to interact with the soft body. Therefore, to simulate the interaction between the surgical instruments and the lens model, each haptic cursor is bound with a capsule phantom shape, representing the instrument tip in the physical simulation.

5.6.3 Lens container

The lens nucleus stays within the lens capsule. Meanwhile, it may rotate when external force is applied. When cracked apart, the lens fragments should move freely within the capsule of lens. For the virtual model set, a container is designed to hold the lens model in its position, and define the boundary of its movements. Obviously, the container needs to contact with the soft body, therefore it needs to be built as a composition of simple shapes.

To limit the range of motion, the container is designed as a hollow which approximates the shape of the lens capsule. Two cuboids serve as the floor and

ceiling, and three set of cuboid are placed around the lens model as walls on the side (Figure 29, right). The shape precision of the container depends on the number of cuboids used in each set of wall. Meanwhile, the computational complexity of collision detection is proportional to the number of shapes in contact with the soft body model. Experiments suggest the optimal number of faces in each set of walls should be 6 (Figure 29, left), which may best balance the shape precision and the fluency of the simulation. In all, the container is assembled by a total of 20 cuboids. As the lens container is an artificial object which only serves to keep the soft body model, its collision with all shapes but the lens soft body model is ignored. In addition, the lens container is set to be neither visible nor touchable.

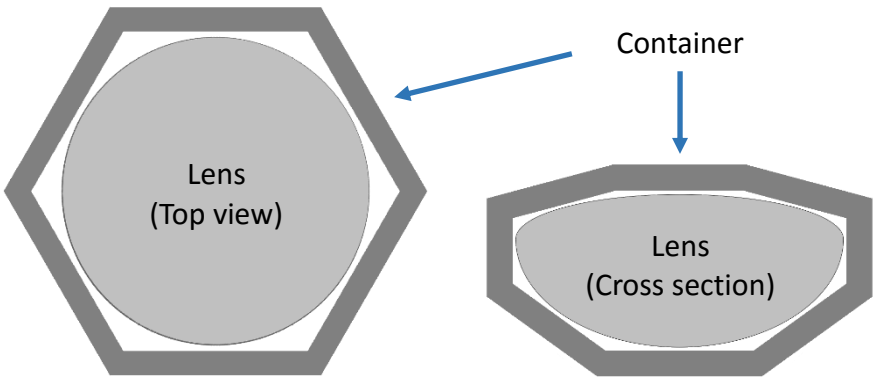

Figure 29. Geometry of the lens container
Left: top view (side walls only); right: cross section view

5.6.4 Simulation of phacoemulsification

To simulate phacoemulsification, the geometry of the lens model is modified in run time. The model can be modified by the soft body aspirator, which is an object with a special functionality that eliminates all soft body vertices colliding with it. When a vertex is removed from the soft body model, all tetrahedrons sharing that vertex are withdrawn. In the simulation, a soft body aspirator is attached to the tip of the phaco handpiece. When the phaco pedal is pressed down to position three, the removal of soft body is triggered. The functionality of the soft body aspirator is enabled and disabled periodically, with the interval proportional to the pressed distance of the phaco pedal.

When one or more vertices are removed from the model, the graphics rendering algorithm deletes triangles that contain this vertex, and add triangles that just emerged to the surface. Subtriangulation shader is applied to the updated surface mesh, while the red reflex effect is also updated to reflect the remaining depth of the lens. Figure 30 shows the lens model being modified during soft body removal.

In the procedure of constructing a cross-shaped trench, the lens nucleus needs to be rotated 90 degrees to enable sculpting on different direction. Soft body dynamics provides a more realistic and natural way to simulate the lens rotation. As mentioned above, a capsule shape overlapping with the virtual instrument is bound to the haptic interaction point. This capsule provides shape-

to-shape contact with the soft body lens model. In this way, not only the tip but the whole portion of the instrument inside the eye can touch the lens and apply force to it. In addition, the lens container limits the range of motion of the lens model, so that when pushed by the instrument, the lens naturally rotates within the container.

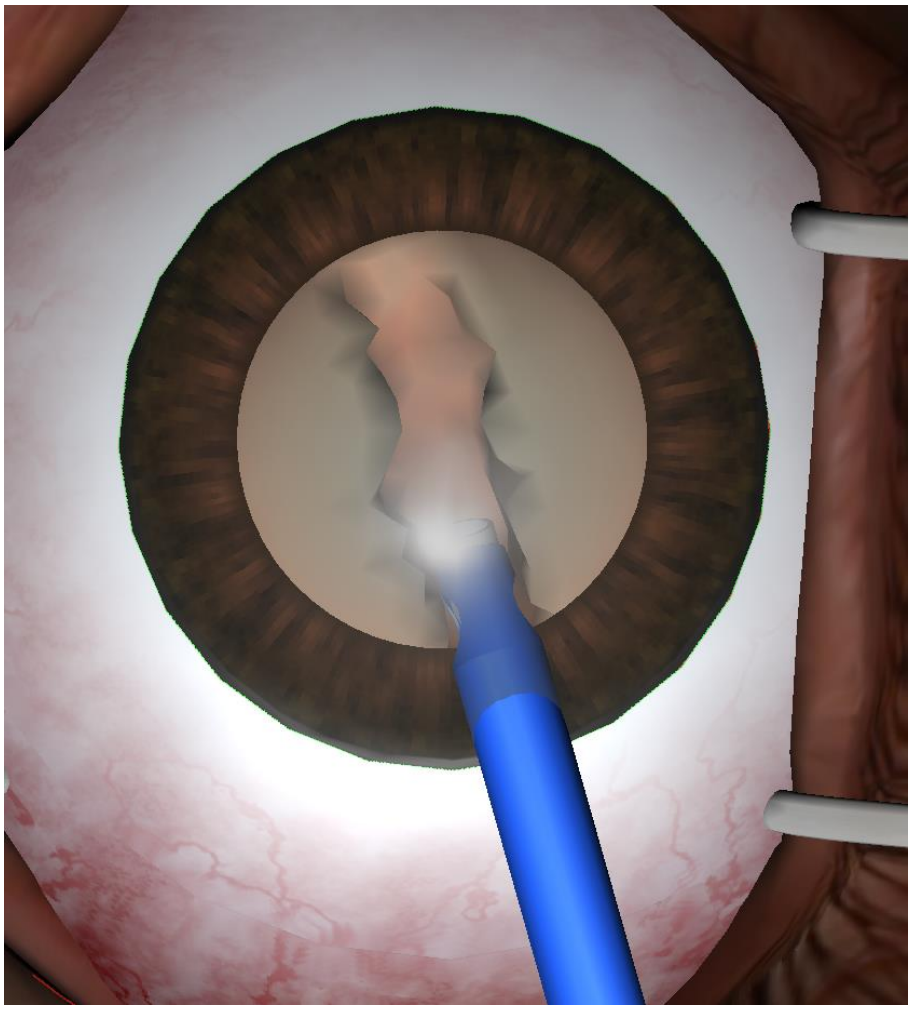

Figure 30. Soft body removal

5.6.5 Simulation of nucleus cracking

With a groove of Swiss cross shape being constructed with enough depth, the lens nucleus is ready for cracking. In this procedure, both instruments are engaged in the trench, pushing on opposite walls apart. The simulation takes several criteria to decide the timing of lens cracking. First, the trench should be deep enough. Besides, the force applied by the instruments are sufficient. In addition, the pivot points of both instruments locate in the lower part of the groove, and on opposite walls of the same groove arm. If any condition does not meet, the lens may only be deformed or rotated without cracked apart.

To simulate lens cracking, a capsule-shaped cracking gadget is loaded to the simulation scene to remove tetrahedrons. It is located in the middle of the pivot points of two instruments, and placed inside the groove parallel to the equatorial plane of the lens. Since this object is not a part of the actual surgical instrument, and is only for assisting the simulation of lens cracking, it is excluded from visualization and haptic rendering, and physics engine ignores its collision with any objects other than the soft body lens model. All particles (vertices) touched by this object are removed from the soft body model. The cracking gadget gradually moves downward when force applied to the groove. When it eventually travels through the entire lens, the cracking is completed. Rotate the

lens model 90 degrees and crack again from another arm of the groove. Figure 31 shows the simulation of nucleus cracking.

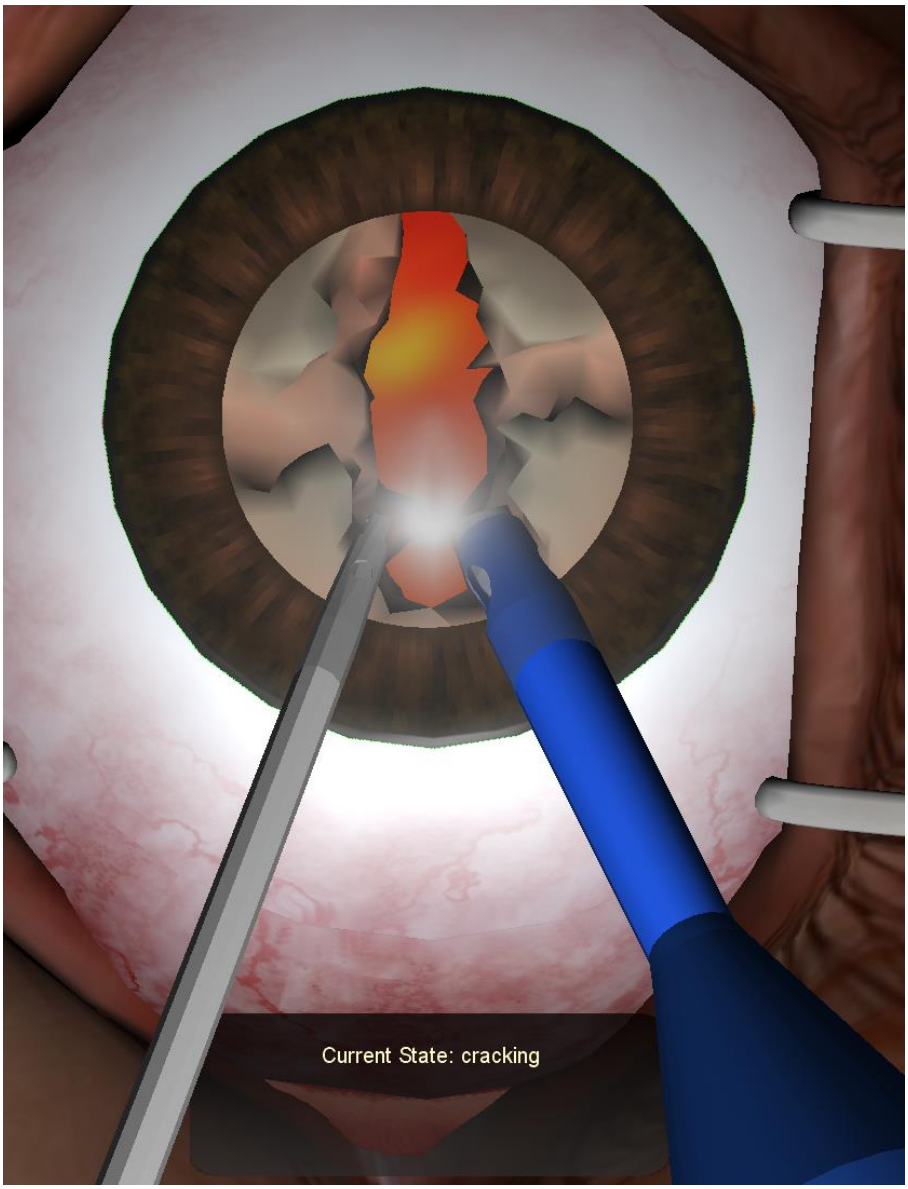

Figure 31. Nucleus cracking

5.6.6 Simulation of quadrant removal

When the lens model is separated into multiple fragments, they should be individually mobile so that the force applied to one piece has no direct impact to the rest of the model. Additionally, detecting collisions between lens fragments is necessary to prevent any piece going through another. However the algorithm of self-collision in PhysX engine is of very high computational complexity and may bring a huge run-time overhead to the simulation, especially when the number of tetrahedrons is large.

As a useful and fast workaround, the interaction between lens quadrants can be achieved by adding attachments to the lens model. Soft body particles can be attached to rigid body shapes and fixed at a position in shape's coordinate frame. As the shape moves, this position will change in the global frame, pulling the soft body with it. Optionally, the soft body may move the attachment as well. When the lens cracking is initiated, four rigid body shapes are attached to the lens model, serving as cores of each lens quadrants. The attachment is represented by a simple shape, whose contact with the soft body is supported by the physics engine. Cuboid shapes are chosen as the attachment to simulate the sharp edge of the lens quadrants. Figure 32 demonstrates the placement of the lens attachments. Note that the attachments are transparent in visualization and excluded in haptic rendering.

Figure 32. Lens attachments

When the phaco pedal is stepped down to position two or position three, the aspiration is enabled, which brings the lens fragments towards the phaco tip. In the simulation, the aspiration is implemented by pulling the lens fragment in front towards the phaco tip. A ray is emitted from the tip towards the front. Ray-casting algorithm can identify the lens particle hit by the ray, if there is one. A force is

then applied to the identified particle to pull it towards the tip. If the pedal is at position three, the soft body aspirator at the instrument tip removes the lens particles nearby, as described in Section 5.6.4.

When a lens quadrant is partially removed, the attachments may interfere with the removal of the lens particle. In addition, when particles on the outside are removed, the attachment shape emerges, which may result in unnecessary or inaccurate collision. Under this circumstance, the attached rigid body should be removed. Similar to the aspiration simulation, ray-casting helps find out the attachments in front of the phaco tip. The rigid shape is detached from the soft body model when phaco power is applied to the corresponding lens quadrant for some considerable time. Figure 33 shows the simulation of quadrant removal.

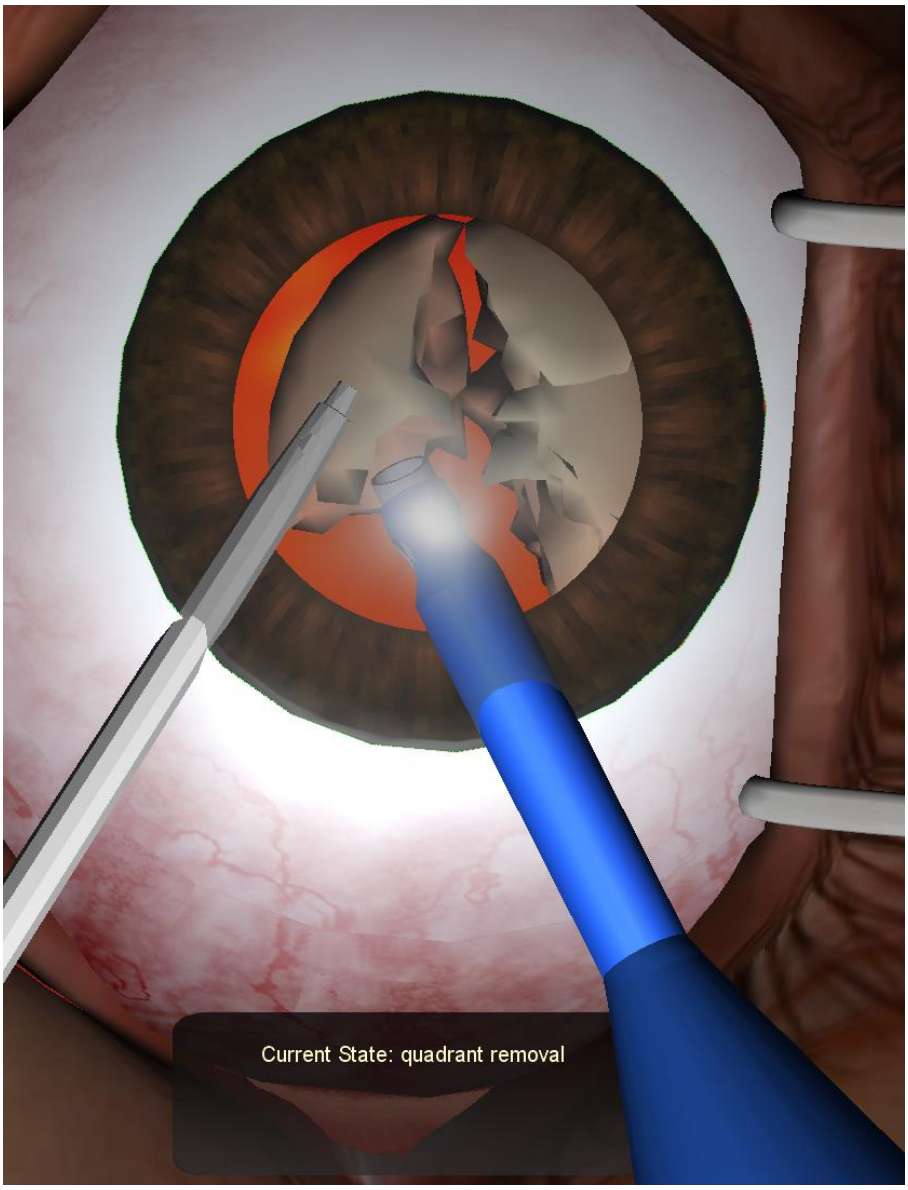

Figure 33. Simulation of quadrant removal

6 SURVEY OF TACTILE FEEDBACK IN CATARACT SURGERY

During a survey performed at Harvard Medical School (Huynh et al., 2008), the majority of the participating ophthalmologists agreed that haptic feedback has an essential role in ophthalmic surgeries. To investigate the importance of haptics in different procedures of the cataract surgery, a more detailed survey on the cataract surgery is conducted. The survey was done by seeking voluntary responses by Google Forms from members of ASCRS (American Society of Cataract and Refractive Surgery) and responses from the Wilmer Eye Institute, Johns Hopkins University through an open solicitation.

6.1 Methods

The target population of the survey is ophthalmologists including residents, fellows and practitioners.

The survey is made up of five sections. The section 1 surveys the backgrounds of the participants, including their level of medical experience, years of experience on cataract surgery and their home institution. Participants are then asked to rate the overall importance of the tactile sensation in the cataract surgery on a scale from 1 to 5, where 1 means “not important” and 5 means “critically important”. Section 2 asks participants for their opinions on the intensity of haptic sensation in all major surgical procedures in the cataract surgery. Section 3 discovers the level of tactile feedback of several common surgical

scenarios. For all questions in section 2 and 3, participants rate the haptic sensation on a scale of 1 to 5, where 1 means the haptic feedback is almost imperceptible, and 5 means there is very clear sensation. Section 4 and 5 of the survey focus on surgeons who have experience on another cataract surgery simulator EyeSi Surgical (see Section 2.1.1), which does not support haptic rendering. The same questions in section 2 and 3 are used to investigate how much the haptic feedback is perceived when practicing cataract surgery on this simulator.

The entire survey questions are listed in the Appendix A.

6.2 Results and analysis

51 participants from 28 medical institutions participated the survey. Among the participants there are 38 practitioners, 12 residents and 1 fellow. The year of experience on cataract surgery ranges from 0 (residents in PGY 1) to 42 years, with the average of 15.68 years and the median of 15 years.

The majority of the participants agreed on the importance of the tactile sensation in cataract surgery. The average rating from all 51 responses is 4.10. Specifically, 35.3% of the participants selected the highest scale of 5, and 49% chose 4 in their response (Figure 34). Only a small fraction of participants gave a rate of 2 or 3, and none of the participants rated the importance of haptics with only the scale of 1.

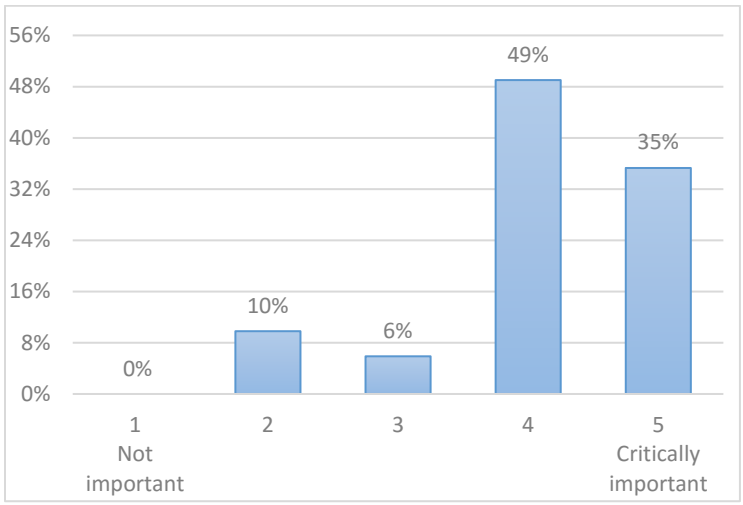

Figure 34. Overall importance of the tactile sensation in cataract surgery

Figure 35 shows the mean and standard deviation of the rating of haptic perception for actual surgical procedures. Several facts are revealed from the statistics. First of all, with high average rating and relatively low standard deviation, participants reported clear haptic sensation in the surgical procedures including corneal incision construction, cracking the lens nucleus, rotating the lens nucleus and IOL insertion. Haptic feedback is relatively strong in the procedures including side port construction, phaco sculpting and quadrants removal by the lens manipulator. For cutting and tearing in capsulorrhexis, the average rating is relatively low, while the standard deviation is relatively high.

This indicates the haptic sensation of performing capsulorrhexis is less significant compared with other procedures, and the opinion varies among participants.

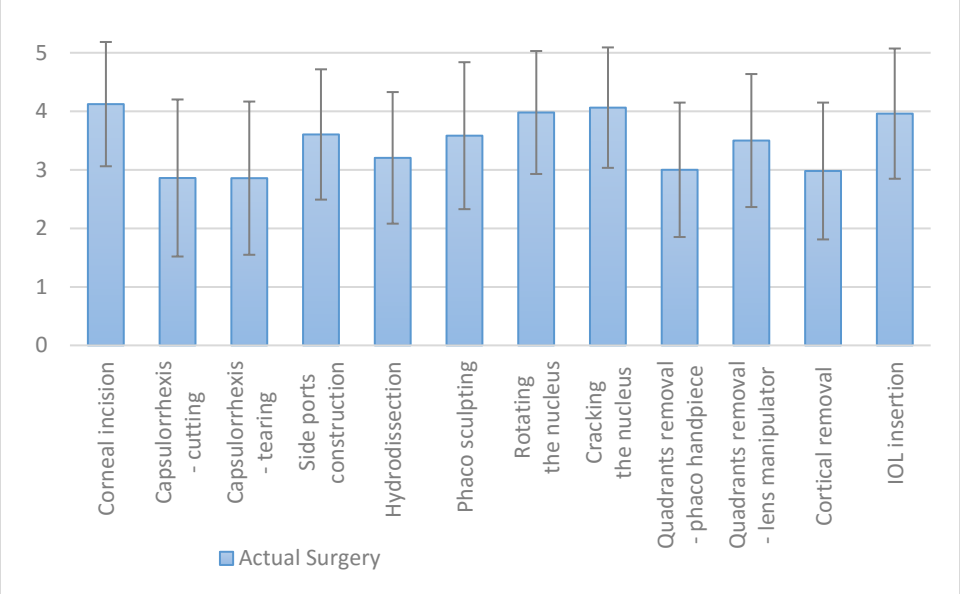

Figure 35. Rating of haptic perception for actual surgical procedures

The statistical data is shown in shown in Table I, with the surgical procedures sorted by the average rating of haptic perception.

Table I
Scale of haptic perception in major surgical procedures

Procedures	Average Rating	Standard Deviation
Corneal incision construction	4.122	1.062
Cracking the nucleus	4.063	1.029
Rotating the nucleus	3.979	1.051
IOL insertion	3.960	1.113
Side ports construction	3.604	1.113
Phaco sculpting	3.583	1.256
Quadrants removal - lens manipulator	3.500	1.137
Hydrodissection	3.204	1.124
Quadrants removal - phaco handpiece	3.000	1.149
Cortical removal	2.980	1.169
Capsulorrhexis - cutting	2.860	1.342
Capsulorrhexis - tearing	2.857	1.309

Figure 36 shows the average and standard deviation of the scale for the haptic feedback on several scenarios in the surgery. According to the responses, the haptic perception during instruments insertion is evident. Inserting a blunt instrument provides a more significant haptic feedback compared with a sharp instrument. Besides, when the instrument is inserted, pushing against the corneal incision has clear haptic feedback, especially when pushing the incision on the sides. Meanwhile, the tactile sensation of accidentally damaging the lens posterior capsular bag or unexpected touching the corneal from inside is not

obvious. The standard deviation of all data of this section are relatively small, indicating high reliability of the results. Table II shows the detail statistics.

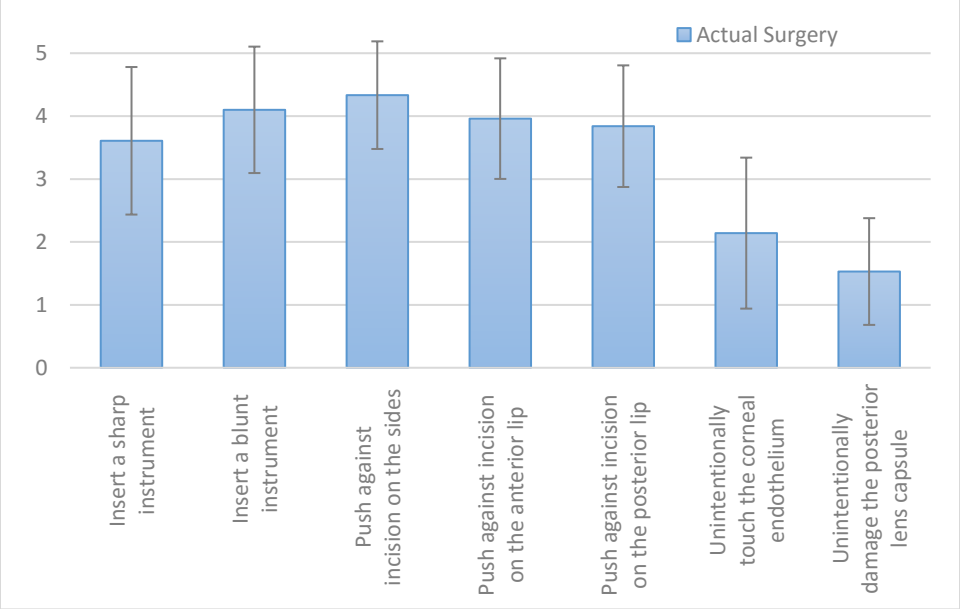

Figure 36. Rating of haptic perception for selected surgical scenarios

Table II

Scale of haptic perception in selected scenarios of cataract surgery

Scenario	Average Rating	Standard Deviation
Push against incision on the sides	4.333	0.856
Push against incision on the anterior lip	3.960	0.958
Push against incision on the posterior lip	3.840	0.967
Insert a blunt instrument	4.100	1.005
Insert a sharp instrument	3.608	1.173
Unintentionally touch the corneal endothelium	2.140	1.200
Unintentionally damage the posterior lens capsule	1.529	0.848

Among all participants, 12 of them have experience of using EyeSi simulator for cataract surgery training. According to their response, the haptic feedback on this simulator is generally insignificant compared with the actual surgery among all surgical procedures and all studied surgical scenarios with clear haptic perception. Figure 37 and Figure 38 illustrate the difference of the average rating between the actual cataract surgery and the simulation on EyeSi platform.

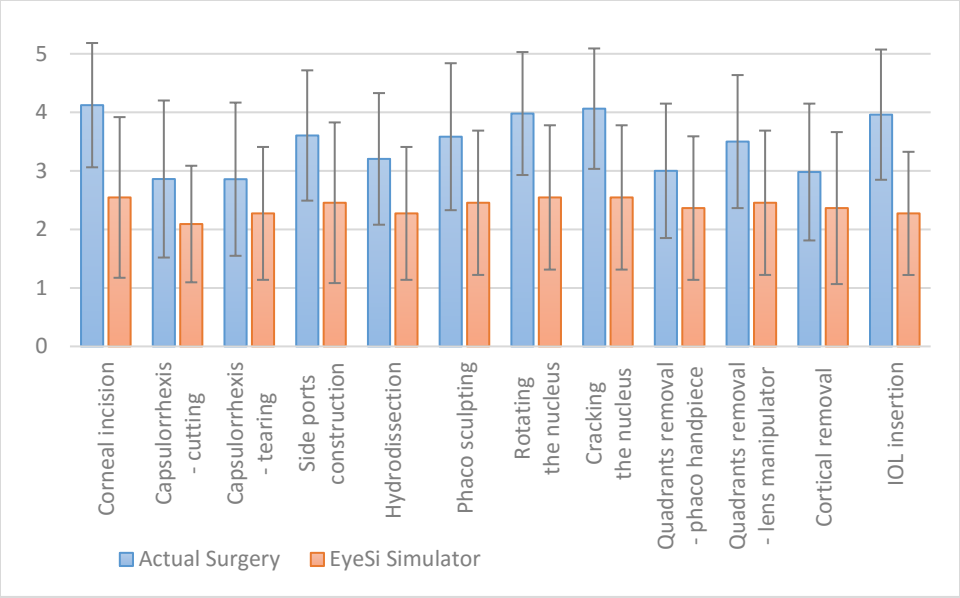

Figure 37. Comparison of haptic feedback for major surgical procedures

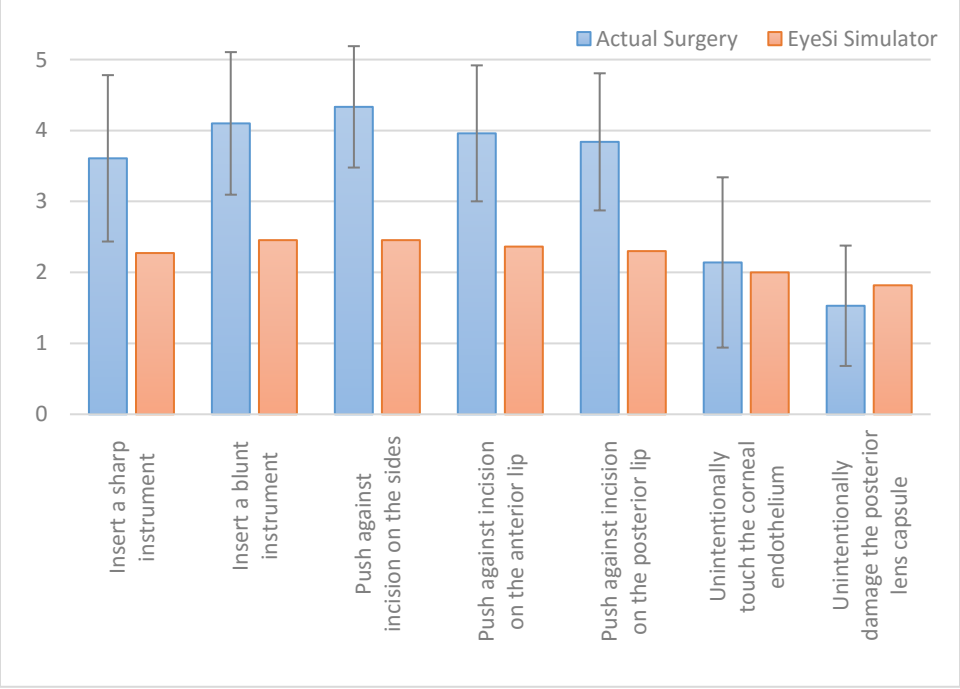

Figure 38. Comparison of haptic feedback in selected scenarios

6.3 Discussion

The responses demonstrates the importance of haptic sensation in cataract surgery. The result of the survey has motivated the design and implementation of a haptic-based cataract surgery simulator. The MicrovisTouch platform includes simulation of multiple cataract surgical procedures with notable haptic feedback, including corneal incision construction, side port paracentesis, phaco sculpting, nucleus rotation, nucleus cracking, and quadrant removal.

The survey reveals the significance of the haptic feedback when instruments interact with the corneal incision. Correspondingly, the haptic experience in this scenario is enhanced in the cataract simulation. The joint model applied to the eye model enables the eye rotation when external force is applied. The fulcrum haptic effect generate resistance when instrument pushes the corneal incision. The friction effect emulates the resistance during insertion of instrument.

The survey clearly indicates the lack of tactile feedback on the EyeSi simulator. With the enhancement of haptic rendering, the realism of experience on the MicrovisTouch platform may outperform the EyeSi simulator, or other simulators which do not support haptics.

7 CONCLUSION

The integration of visual and haptic interface has enhanced the level of realism of the cataract surgery simulator. Dynamic engine vividly simulates the force interaction between objects in the virtual scene. Special visual artifacts provide guidance to the interaction, while haptic effects present the force in the virtual scene to user's hand.

The contributions in the design and implementation of the cataract surgery simulator are listed as follows. First, a haptic and physics based dynamic virtual eye model has been designed. Haptic effects including fulcrum and friction emulate the force interaction between the eye and the surgical instruments. Managed by an articulated joint model of point based dynamics, the eye model is able to rotate by external force within a limited range.

Besides, two sets of cataract lens model have been designed using volume rendering and soft body dynamics respectively. Modification of geometry in simulation run time is possible. The visualization and haptic interaction of both models have been implemented. A special visual effect of red reflex is investigated and implemented on both lens models. The volumetric model describes geometry with higher level of detail, which benefits the simulation of lens sculpting. The soft body model is capable of being separated into multiple

individually mobile fragments, therefore it is used in the simulation of lens cracking and quadrants removal.

In addition, a series of surgical procedures simulation as well as several VR-based training modules have been implemented. The simulator is capable of conducting a step by step training of cataract surgery. Objective performance assessment methods have been designed for training modules as well as surgical simulation.

The haptics and virtual reality cataract surgery simulator could be further enhanced in a number of ways. First of all, validations of the simulation modules by expert ophthalmologists can be conducted to evaluate its capabilities for training residents and fellows. Besides, to improve the performance assessment of the simulation, advanced algorithms including pattern recognition can be implemented to thoroughly analyze the user operation. Additionally, a comprehensive curriculum can be designed to help residents develop the knowledge of cataract surgery using this simulator.

CITED LITERATURE

American Academy of Ophthalmology: EyeWiki – Cataract, http://eyewiki.aao.org/Cataract/

Banerjee, P.P., Edward, D.P., Liang, S., Bouchard, C. S., Bryar, P. J., Ahuja, R., Dray, P., Bailey, D. P.: Concurrent and face validity of a capsulorhexis simulation with respect to human patients. Studies in Health Technology and Informatics, 173:35-41, 2012.

Banerjee, P. P., Liang, S., Edward, D. P., Palmieri, P. A., Batt, S., Mudumbai, R.: Simulating Capsulorrhexis Procedure with High Performance Graphic and Haptic Feedback: A PILOT STUDY. Investigative Ophthalmology & Visual Science 50(13):6115-6115, 2009.

Banerjee, P., Luciano, C., Dawe, G., Florea, L., Steinberg, A. D., Drummond, J., Zefran, M. (The Broad of Trustees of the University of Illinois): COMPACT HAPTIC AND AUGMENTED VIRTUAL REALITY SYSTEM. U.S. 7,812,815 B2, Appl. 11/338,434, 24 Jan 2006; 12 Oct 2010.

Caesar, R., Benjamin, L.: Phacoemulsification Step by Step. Elsevier Press, 2003.

Chew, L. P.: Guaranteed-quality mesh generation for curved surfaces. In: Proceedings of the ninth annual symposium on Computational geometry, pp. 274-280. ACM, 1993.

Coin3D, https://bitbucket.org/Coin3D/coin/wiki/Home/

Fast Light Toolkit, http://www.fltk.org/

Fine, I. H., Packer, M., Hoffman, R. S.: Power modulations in new phacoemulsification technology: improved outcomes. Journal of Cataract & Refractive Surgery, 30(5):1014-1019, 2004.

Fried, G. M., Feldman, L. S., Vassiliou, M. C., Fraser, S. A., Stanbridge, D., Ghitulescu, G., Andrew, C. G.: Proving the value of simulation in laparoscopic surgery. Annals of Surgery. 240:518-528, 2004.

Gallagher, A. G., Cates, C. U.: Approval of virtual reality training for carotid stenting: what this means for procedural- based medicine. JAMA. 292:3024-3026, 2004.

Geomagic, http://www.geomagic.com/

Haluck, R. S., Krummel, T. M.: Computers and virtual reality for surgical education in the 21st century. Archives of Surgery, 135(7):786-792, 2000.

Huynh, N., Akbari, M., Loewenstein, J. I.: Tactile Feedback in Cataract and Retinal Surgery: A Survey-Based Study, Journal of Academic Ophthalmology, 1(2):79-85, 2008.

Khalifa, Y. M., Bogorad, D., Gibson, V., Peifer, J., Nussbaum, J.: Virtual reality in ophthalmology training. Survey of Ophthalmology, 51(3):259-273, 2006.

Lam, C. K., Sundaraj, K., Sulaiman, M.: Virtual simulation of eyeball and extraocular muscle reaction during cataract surgery. Procedia Engineering, 41:150-155, 2012.

Lam, C. K., Sundaraj, K., Sulaiman, M.: Virtual Reality Simulator for Phacoemulsification Cataract Surgery Education and Training. Procedia Computer Science, 18:742-748, 2013.

Laurell, C. G., Söderberg, P., Nordh, L., Skarman, E., Nordqvist, P.: Computer-simulated phacoemulsification. Ophthalmology, 111(4):693-698, 2004.

Leaming, D. V.: Practice styles and preferences of ASCRS members—2003 survey. Journal of Cataract & Refractive Surgery, 30(4): 892-900, 2004.

Liang, S., Banerjee, P. P., Edward, D. P.: A high performance graphic and haptic curvilinear capsulorrhexis simulation system. 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5092-5095. IEEE, 2009.

Liang, S.: Design and Validation of a High Performance Continuous Curvilinear Capsulorrhexis Simulator. Ph.D. dissertation, University of Illinois at Chicago, Chicago, Illinois, 2010.

Lorensen, W. E., Cline, H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In ACM Siggraph Computer Graphics. 21(4):163-169. ACM, 1987.

Luciano, C., Banerjee, P., Florea, L., Dawe, G.: Design of the ImmersiveTouch: a high-performance haptic augmented virtual reality system. In Proc. Of Human-Computer Interaction (HCI) International Conf., Las Vegas. 2005.

Luciano, C. J., Banerjee, P. P., Rizzi, S. H. R.: GPU-based elastic-object deformation for enhancement of existing haptic applications In: 2007 IEEE International Conference on Automation Science and Engineering, pp. 146-151. IEEE, 2007.

Luciano, C.: Open Surgery Training Simulator Using Haptics and Augmented Reality Technologies. Ph. D. dissertation, University of Illinois at Chicago, Chicago, Illinois, 2010.

Luo, J., Kania, P., Banerjee, P. P., Sikder, S., Luciano, C. J., Myers, W. G.: A part-task haptic simulator for ophthalmic surgical training. 2016 IEEE Symposium on 3D User Interfaces (3DUI), pp. 259-260. IEEE, 2016.

Melerit Medical, http://www.melerit.se/

Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. Journal of Visual Communication and Image Representation, 18(2): 109-118, 2007.

Nagy, Z., Takacs, A., Filkorn, T., Sarayba, M.: Initial clinical evaluation of an intraocular femtosecond laser in cataract surgery. Journal of refractive surgery, 25(12), 2009.

National Eye Institute, https://www.nei.nih.gov/health/cataract/cataract_facts/

Nealen, A., Müller, M., Keiser, R., Boxerman, E., Carlson, M.: Physically based deformable models in computer graphics. Computer Graphics Forum, 25(4): 809-836, Blackwell Publishing Ltd, 2006.

NVIDIA PhysX, http://www.geforce.com/hardware/technology/physx/

Rama, M. A., Pérez, M. V., Bao, C., Flores-Arias, M. T., Gómez-Reino, C.: Gradient-index crystalline lens model: A new method for determining the paraxial properties by the axial and field rays. Optics Communications, 249(4):595-609, 2005.

Rineau, L., Yvinec, M.: A generic software design for Delaunay refinement meshing. Computational Geometry, 38(1):100-110, 2007.

Rizzi, S. H., Banerjee, P. P., Luciano, C. J.: Automating the extraction of 3d models from medical images for virtual reality and haptic simulations. In: 2007 IEEE International Conference on Automation Science and Engineering, pp. 152-157. IEEE, 2007.

Ruppert, J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation. Journal of algorithms, 18(3):548-585, 1995.

Sachdeva, R., Traboulsi, E. I.: Performance of patients with deficient stereoacuity on the EYESi microsurgical simulator. American Journal of Ophthalmology, 151(3):427-433, 2011.

Schill, M. A., Wagner, C., Hennen, M., Bender, H. J., Männer, R.: Eyesi–a simulator for intra-ocular surgery. In: Medical Image Computing and

Computer-Assisted Intervention–MICCAI'99, pp. 1166-1174. Springer Berlin Heidelberg, 1999.

Selvander, M., Åsman, P.: Cataract surgeons outperform medical students in Eyesi virtual reality cataract surgery: evidence for construct validity. Acta Ophthalmologica, 91(5):469-474, 2013.

Seymour, N.E.: VR to OR: a review of the evidence that virtual reality simulation improves operating room performance. World Journal of Surgery. 32:182-188, 2008.

Shewchuk, J. R.: Tetrahedral mesh generation by Delaunay refinement. In: Proceedings of the Fourteenth Annual Symposium on Computational Geometry, pp. 86-95. ACM, 1998.

Sikder, S., Luo, J., Banerjee, P., Luciano, C., Kania, P., Song, J., Al-Kahtani, E., Edward, D. P., Towerki, A. A.: The use of a virtual reality surgical simulator for cataract surgical skill assessment with 6 months of intervening operating room experience. Clinical Ophthalmology (Auckland, NZ), 9:141-149, 2015.

Sikder, S., Tuwairqi, K., Al-Kahtani, E., Myers, W. G., Banerjee, P.: Surgical simulators in cataract surgery training. British Journal of Ophthalmology, 98(2), 154-158, 2014.

Singh, A., Strauss, G. H.: High-Fidelity Cataract Surgery Simulation and Third World Blindness. Surgical Innovation, 1553350614537120, 2014.

Smith, G., Atchison, D. A., Robert Iskander, D., Jones, C. E., Pope, J. M.: Mathematical models for describing the shape of the in vitro unstretched human crystalline lens. Vision Research, 49(20):2442-2452, 2009.

Söderberg, P. G., Laurell, C. G., Artzén, D., Nordh, L., Skarman, E., Nordqvist, P., Andersson, M.: Computer-simulated phacoemulsification improvements. In: International Symposium on Biomedical Optics, pp. 76-80. International Society for Optics and Photonics, 2002.

Söderberg, P. G., Laurell, C. G., Nordqvist, P., Skarman, E., Nordh, L.: Virtual cataract surgery: clinical evaluation. In: Biomedical Optics, pp. 62-66. International Society for Optics and Photonics, 2003.

Söderberg, P., Laurell, C. G., Simawi, W., Skarman, E., Nordqvist, P., Nordh, L.: Performance index for virtual reality phacoemulsification surgery. In: Biomedical Optics, pp. 64261B-64261B. International Society for Optics and Photonics, 2007.

Söderberg, P., Laurell, C. G., Simawi, W., Skarman, E., Nordh, L., Nordqvist, P.: Measuring performance in virtual reality phacoemulsification surgery. In:

Biomedical Optics, pp. 684412-684412. International Society for Optics and Photonics, 2008.

Van Overveld, C. W. A. M., Wyvill, B.: An algorithm for polygon subdivision based on vertex normals. In: Computer Graphics International, 1997. Proceedings, pp 3-12. IEEE, 1997.

Vlachos, A., Peters, J., Boyd, C., Mitchell, J. L.: Curved PN triangles. In: Proceedings of the 2001 Symposium on Interactive 3D Graphics, pp 159-166. ACM, 2001.

VRmagic, https://www.vrmagic.com/simulators/eyesi-surgical/

Wikipedia: Cataract, https://en.wikipedia.org/wiki/Cataract/

World Health Organization, http://www.who.int/topics/cataract/en/

Zilles, C. B., Salisbury, J. K.: A constraint-based god-object method for haptic display. In: Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on, Vol. 3, pp. 146-151. IEEE, 1995.

APPENDICES

A. Survey of Tactile Feedback in Cataract Surgery

A paper by Nancy Huynh, MD, Mona Akbari, and John I. Loewenstein, MD has conducted a preliminary survey on the importance of the tactile feedback in ophthalmological surgeries. A more detailed survey on the cataract surgery is now being conducted.

- Your level of medical experience

  □ Medical Student □ Resident
  □ Fellow □ Practitioner

- Please specify your medical institution:

  _____ .

- Years of experience on cataract surgery: _____.

- Overall, how important is tactile feedback (touch or feel of tissue-instrument interactions) to you in cataract surgery?

1	2	3	4	5
(Not important)				(Critically important)

Please rate how much you feel in your hands during actual cataract surgery.

Score of 1 stands for almost imperceptible, and score of 5 stands for a very clear sensation.

- Construct the corneal incision by keratome

1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

- Puncture the anterior capsule with cystotome

1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Tear the capsule with capsulorrhexis forceps
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Construct the side port incision by keratome / 15-deg blade
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Inject BSS during hydrodissection
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Sculpt the cross on the lens during nucleus dissection with phaco handpiece
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Rotate the nucleus with phaco handpiece and lens manipulator
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Crack the nucleus with phaco handpiece and lens manipulator
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Remove quadrants of nucleus with phaco handpiece
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Help quadrants removal with lens manipulator
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Perform cortical material removal with I/A handpiece
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Perform IOL insertion
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Insert a sharp instrument (e.g. cystotome, capsulorrhexis, lens manipulatior, etc.) through the corneal incision
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Insert a blunt instrument (e.g. phaco handpiece, I/A handpiece, IOL injector, etc.) through the corneal incision
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Push against the sides of the corneal incision with an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Push against the anterior lip of the corneal incision with an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Push against the posterior lip of the corneal incision with an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Unintentionally touch the corneal endothelium with the tip of an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Unintentionally damage the posterior lens capsule with the tip of an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

- Have you experienced any other tactile sensation during cataract surgery?

Also if you have used the Eyesi Surgical simulator before, please rate how much you feel in your hands during simulation on Eyesi.

Score of 1 stands for almost imperceptible, and score of 5 stands for a very clear sensation.

• Construct the corneal incision by keratome				
1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Puncture the anterior capsule with cystotome				
1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Tear the capsule with capsulorrhexis forceps				
1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Construct the side port incision by keratome / 15-deg blade				
1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Inject BSS during hydrodissection				
1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Sculpt the cross on the lens during nucleus dissection with phaco handpiece				
1	2	3	4	5
(Almost imperceptible)				(Very clear sensation)

• Rotate the nucleus with phaco handpiece and lens manipulator
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Crack the nucleus with phaco handpiece and lens manipulator
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Remove quadrants of nucleus with phaco handpiece
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Help quadrants removal with lens manipulator
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Perform cortical material removal with I/A handpiece
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Perform IOL insertion
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Insert a sharp instrument (e.g. cystotome, capsulorrhexis, lens manipulatior, etc.) through the corneal incision
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Insert a blunt instrument (e.g. phaco handpiece, I/A handpiece, IOL injector, etc.) through the corneal incision
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Push against the sides of the corneal incision with an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Push against the anterior lip of the corneal incision with an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Push against the posterior lip of the corneal incision with an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Unintentionally touch the corneal endothelium with the tip of an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Unintentionally damage the posterior lens capsule with the tip of an instrument
1 2 3 4 5 (Almost imperceptible) (Very clear sensation)

• Have you experienced any other tactile sensation during simulation on Eyesi?

• Questions or comments

B. IEEE Copyright Policy for Thesis / Dissertation Reuse

Title:
A part-task haptic simulator for ophthalmic surgical training

Conference Proceedings:
2016 IEEE Symposium on 3D User Interfaces (3DUI)

Author:
Jia Luo; Patrick Kania; P. Pat Banerjee; Shammema Sikder; Cristian J. Luciano; William G. Myers

Publisher:
IEEE

Date:
19-20 March 2016

Thesis / Dissertation Reuse
The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © [year of original publication] IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.

VITA

NAME:	Jia Luo
EDUCATION:	Ph.D., Industrial Engineering and Operations Research University of Illinois at Chicago (UIC), Chicago, Illinois, 2016
	B.S., Information Engineering Beijing Institute of Technology (BIT), Beijing, China, 2010
RESEARCH EXPERIENCE:	January 2016 – May 2016 Research Aide Argonne National Laboratory, Lemont, Illinois
	August 2010 – May 2016 Research Assistant Industrial Virtual Reality Institute, UIC, Chicago, Illinois
TEACHING EXPERIENCE:	May 2011 – May 2016 Teaching Assistant Department of Mechanical and Industrial Engineering, UIC, Chicago, Illinois
HONORS:	Link Foundation Fellowship on Advanced Simulation (2014 – 2015)
	Best Undergraduate Dissertation Award, BIT (2010)
	Scholarship for Excellence, BIT (2006 – 2010)
PUBLICATIONS:	**Luo, J.**, Kania, P., Banerjee, P. P., Sikder, S., Luciano, C. J., Myers, W. G.: A part-task haptic simulator for ophthalmic surgical training. *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 259-260. IEEE, 2016.
	Sikder, S., **Luo, J.**, Banerjee, P., Luciano, C., Kania, P., Song, J., Al-Kahtani, E., Edward, D. P., Towerki, A. A.: The use of a virtual reality surgical simulator for cataract surgical skill assessment with 6 months of intervening operating room experience. *Clinical Ophthalmology* (Auckland, NZ), 9:141-149, 2015.
	Kozak, I., Banerjee, P., **Luo, J.**, Luciano, C.: Virtual reality simulator for vitreoretinal surgery using integrated OCT data. *Clinical ophthalmology* (Auckland, NZ), 8:669-672, 2014.