VIRTUAL REALITY VITRECTOMY SIMULATOR

BY

PAUL F. NEUMANN
B.S., Engineering, University of Illinois at Chicago, 1986
M.S., Engineering, University of Illinois at Chicago, 1992

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2000

Chicago, Illinois

ACKNOWLEDGMENTS

I would like to thank all of the surgeons who have helped evaluate this project including Dr. Blair, Dr. De Alba, Dr. Gieser, and Dr. Pulido. Extra thanks to my committee members for all of their guidance : Tom Defanti, Jon Gieser, Andrew Johnson, Robert Kenyon, and Lewis Sadler. I would also like to Alan Millman for his physically based modeling expertise, and Dave Pape for assisting with the miniBIRD tracking system and the CAVE VR library. Special thanks to Associate Dean Mrinalini Rao for her invaluable assistance in obtaining my human testing approval. Lastly, I would like to thanks everyone at the Virtual Reality in Medicine Laboratory for their wonderful support.

PFN

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
API	Application Programmer's Interface
EVL	Electronic Visualization Laboratory
CAVE	CAVE Automatic Virtual Environment
CME	Continuing Medical Education
CPU	Central Processing Unit
CT	Computed Tomography
FEM	Finite Element Method
MRI	Magnetic Resonance Image
N/A	Not Available
NURBS	Non-Uniform Rational B-Splines
SGI	Silicon Graphics Incorporated
UIC	University of Illinois at Chicago
VR	Virtual Reality

SUMMARY

With the advent of Virtual Reality, surgical simulators have been proposed as a supplement to the education of medical personnel in regard to surgical procedures. Currently, residents and fellows obtain their surgical background from lectures, books and videotapes, and then gradually perform increasingly more complex procedures under a surgeon's supervision. A VR simulator can assist in the training process by helping residents and fellows form a 3D mental model of the anatomical space, allowing them to fine tune various procedures, providing access to a variety of pathologies, and evaluating their performance. Surgical simulators generally consist of anatomical models, physics system, pathology simulation, virtual instruments, and a physical interface. In this study, a new virtual reality system will physically simulate two types of retinal detachments in the human eye, and implement a common set of vitreoretinal instruments.

To accomplish this goal, a three dimensional computer eye model was constructed and coupled with a mass-spring system for elastic deformations. Actual vitrectomy instruments were tracked as part of the simulator's physical interface. Disease simulation was accomplished with an unique force function to emulate trapped fluid while traction was generated by reducing the spring rest lengths. Eight surgical instruments are implemented in the system including: a light probe coupled with a pick, a forceps for grasping and elevation, two types of blades for cutting, a suction cutter for removal, a laser for adhesion, and a needle for drainage. The forceps and pick instruments lock the closest vertex to their position and induces tearing when springs

exceed their assigned tolerance. A unique 3D surface cutting algorithm dynamically subdivides springs and triangles based on the cutting path. The algorithm runs in near real-time and works on any triangle surface with a mass-spring system. The suction cutter simulates material removal by attracting nearby surface elements and then subdividing them near its opening. The laser locks vertices underneath it, and visually indicates adhesion by altering the retinal texture map. Finally, the drainage needle performs a simple gas-fluid exchange before draining a simulated tear.

Four vitreoretinal surgeons evaluated the simulator over two testing sessions which were composed of three simulated surgical cases followed by a subjective Cooper-Harper survey ranking the simulator's training potential. The completed surveys indicated that with some refinements and better tracking the simulator could be a useful addition to the Ophthalmology training program.

CHAPTER 1

BACKGROUND

1.1 Ophthalmology Background

After completing a Medical Degree and one year of outside training, medical residents who are interested in focusing on eye related treatments can enroll in an Ophthalmology program. Such programs generally span three years with the following stages: clinical introduction, treating patients and an overview of subspecialities such as cornea/contact lens, glaucoma/uveitis, pediatric ophthalmology/neuro-ophthalmology, oculoplastics/ophthalmic pathology/consultation, and retina/hereditary disease. Residents receive introductory surgical training by attending related lectures, reading surgical textbooks, watching prerecorded surgical video tapes, and assisting in the basic routines of the operating room. After the introductory program, a resident can continue their study and become a post doctoral fellow by focusing on a specific subspecialty. Actual surgical training occurs during a fellowship. Fellows practice on animal carcasses and sometimes fruits such as grapes or bananas to understand the sensitivity of their surgical instruments. Participation in surgery is similar to an apprenticeship where fellows gradually performing more complex procedures under supervision.

A virtual reality simulator can assist in the beginning stage of this training process by orientating the fellows and allow them to improve their surgical performance without risk to patients. This would reduce the number of surgical complications caused by fellows and

consequentially decrease the overall time needed for surgery. A standardized simulator could provide a means of uniform accreditation for fellows and a testing environment for surgeons participating in Continuing Medical Education (Higgins et al., 1997; Durlach and Mavor, 1995). Since patients are assigned randomly to doctors, fellows may graduate without having any exposure to particular diseases. Simulators can be equipped with a database of pathologies that could give fellows exposure to rarer diseases that they will encounter later in their practice. Simulators can also help reinforce necessary hand-eye coordination required by newer minimally invasive surgery techniques. The total length of a residency program may be reduced since VR simulators can accelerate the training process (Ota et al., 1995). The expense of a VR simulator could be offset by the savings from the reduction in surgical time currently needed to train fellows, the reduced patient stays from complications, and the less dependency on animal resources. Surgical training time which is supplemented by the federal government has been reduced and may be discontinued in the future to cap health care costs. These reductions have left teaching hospitals in a vulnerable position.

1.1.1 Anatomy of the Eye

Knowledge of eye anatomy is a fundamental requirement for Ophthalmology fellows. Figure 1 shows a sagittal section of the eye with many of its internal structures labeled. The eye itself is roughly spherical with a diameter of 2.5 centimeters. On the exterior, orbital muscles insert into the sclera which is the eye's outer whitish protective coating. The clear anterior portion of the sclera protrudes outward forming the cornea which allows light to pass into the anterior chamber. The iris limits and controls the incoming light passing from the anterior aque-

ous chamber into the posterior vitreous chamber. The iris is composed of interwoven pigmented fibers and muscles which control the diameter of the pupil. The lens is located immediately behind the iris and focuses incoming light onto the retina, which is a thin transparent membrane composed of light-sensitive cells lining the rear of the posterior chamber. The lens is held in place and deformed to adjust its focal length by a series of small fibers called the zonular fibers anchored by the ciliary processes.

The posterior chamber is filled with vitreous gel that filters out certain frequencies of light, pressurizes the eye and assists the supply of nutrients to the lens and retina. Healthy vitreous gel has a 4 milliliters volume composed of 99% water with interlocking protein fibers and acids. The vitreous gel fills the chamber in a torus-like shape which is referred to as the vitreous cortex. The vitreous is attached near the optic nerve, the lens and the ora serrata which is the whitish scalloped edge of the retina. The attachment over the ora serrata extends to the outer peripheral retina as well as the ciliary epithelium referred to as the pars plana. This attachment is the strongest anchor for the vitreous and is referred to as the vitreous base.

The retina itself has a number of important features associated with vision. The macula is a small region which has the highest concentration of cone cells responsible for daytime vision. At the macula's center lies the fovea, a depression where the sharpest detail is seen. The remainder of the retina outside the macula is populated with rod cells responsible for daytime peripheral vision and night vision. Medial to the macula lies the optic disk, the location where all of the nerve fibers from the rod and cone cells exit the eye to the brain. Since the optic disk contains no rods or cones within its limits, it is a natural blind spot within the eye. Beneath the retina is

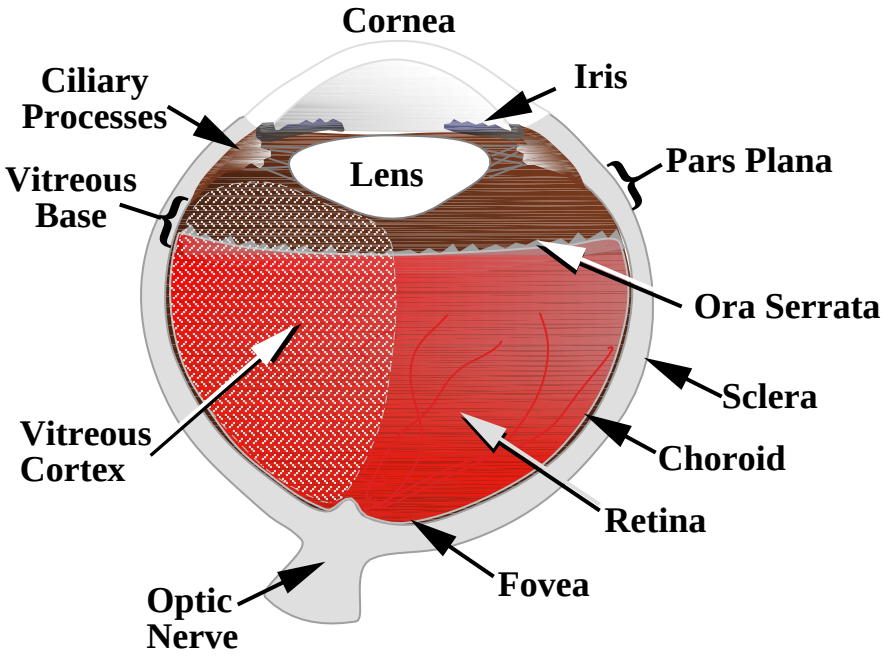


Figure 1. Section through the human eye.

a orange-ish tissue layer called the choroid which nourishes the retina. The choroid is generally more firmly attached to the retina near the optic disk and the ora serrata. Elsewhere the retina is held in place only by adhesion and the fluid diffusion through it into the choroid (Michels et al., 1990).

1.1.2 Retinal Detachments

Retinal detachments occur for many reasons. Two common types are rhegmatogenous detachment and proliferative diabetic traction. Rhegmatogenous detachment results from changes within the vitreous gel. With age, the number of acid molecules within the vitreous gel de-

creases resulting in a more liquid vitreous. If the amount of liquid vitreous is significant, the thin membrane surrounding the cortex may rupture and detach posteriorly near the optic nerve. Although a vitreous detachment may not directly affect a person's vision, gravitational and inertial forces generated by eye movements can produce traction on the vitreous base from currents in the liquid vitreous. Uneven traction on the vitreous base may result in peripheral retinal tears due to irregular attachments of the vitreous base and the thinness of the retina. Typically, these types of retinal tears are U-shaped. Once a tear has occurred, the flow of liquid vitreous can enter between the retina and the choroid. If the flow into the tear is continuous, the normal adhesive-like diffusion force is overwhelmed resulting in retinal detachment (Michels et al., 1990). Trapped subretinal fluid is often yellowish in color since the protein molecules of the liquid vitreous are absorbed slower than other molecules. If the retina remains detached, the affected cells will perish from lack of nutrients.

Another type of retinal detachment is caused by a sudden growth of fibrovascular tissue from the posterior retina onto the outer membrane of the vitreous cortex. This type of detachment is caused by proliferative traction and often occurs in diabetes. Its exact cause is not yet completely understood. As the fibrovascular tissues grow they contract, deforming the vitreous cortex. If these forces are great enough, vitreous detachment and eventually retinal detachment can result. There are three general categories of contractions: anteroposterior traction which displaces the vitreous and retina toward the front of the eye, circumferential traction which compresses radially, and perpendicular traction which compresses the vitreous and retina toward the center of the chamber (Lewis and Ryan, 1994). Fibrovascular tissue is visually white in

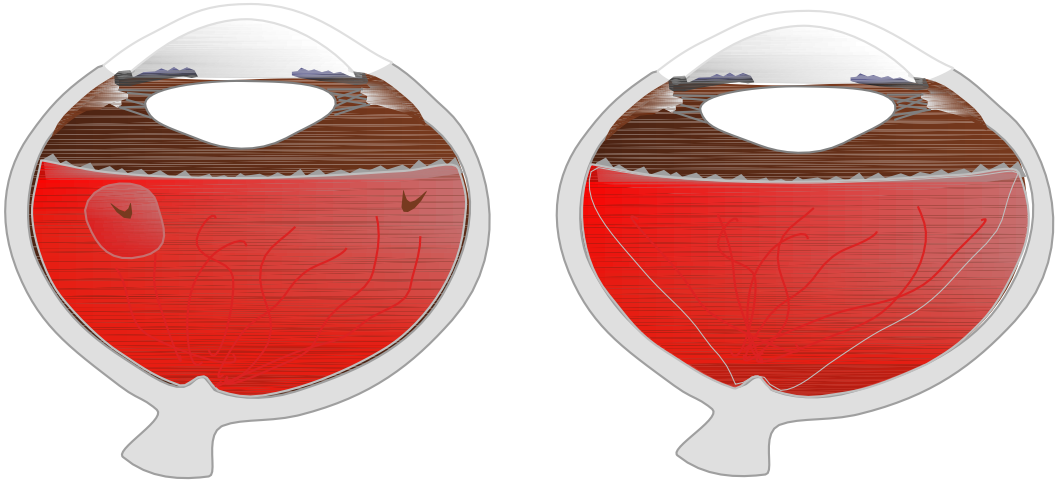


Figure 2. Rhegmatogenous (left) and tractional detachments (right).

color and grows close to the retinal surface making it difficult to remove. In addition, a patient might have a combination of rhegmatogenous and tractional detachment. Figure 2 illustrates two types of retinal detachments.

1.1.3 Vitrectomy Surgery Background

To surgically correct detachments, a vitrectomy is performed to reattach the retina, stop any bleeding and remove any fibrovascular tissue. Modern vitreous surgery is categorized as a closed divided system in which a surgeon inserts an operative instrument and a light probe into the vitreous chamber through incisions called sclerotomies made in the sclera above the pars plana section as shown in Figure 3. Additional sclerotomy is also made for fluid infusion to maintain pressure in the vitreous cavity during the operation which can be controlled by raising

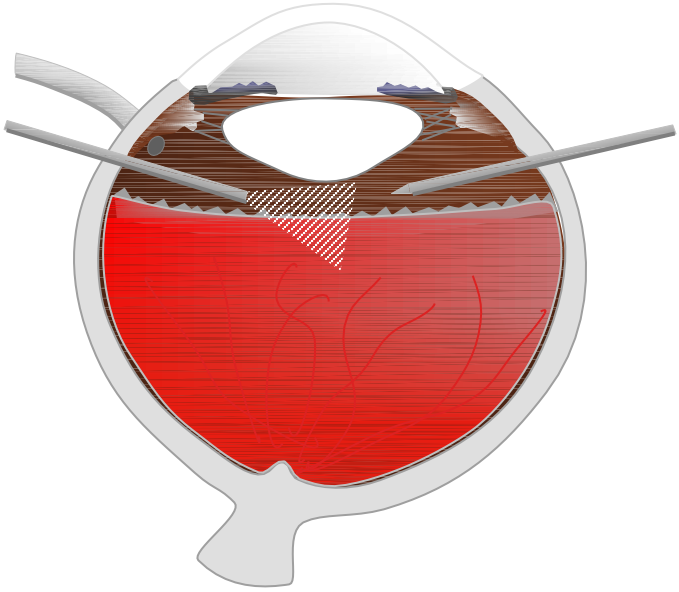


Figure 3. Closed divided surgical system.

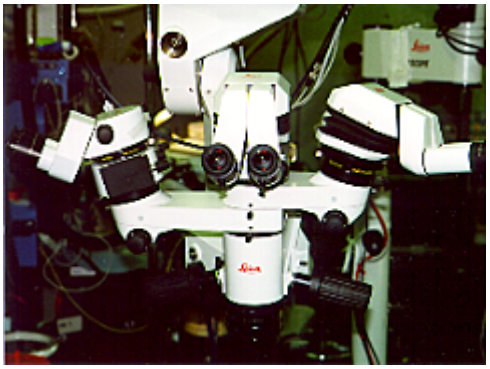
or lowering an external container. Often a corneal lens is placed on top of the patient's cornea for increased clarity into the vitreous chamber. Operative instruments range from simple forceps and picks for grasping, vitreoretinal scissors and blades for cutting, blunt drainage needles, suction cutter for removal, to more complex instruments that physically fuse the retina to the choroid such as photocoagulation laser and cryoprobe.

After creating the sclerotomies, any proliferative traction must be released by severing its connection to the retina using a blade or scissors. Once the traction is released, the retina will recede back but might not completely return to its normal position. Any remaining vitreous gel that is not attached to a fibrovascular growth or near the vitreous base should be removed from

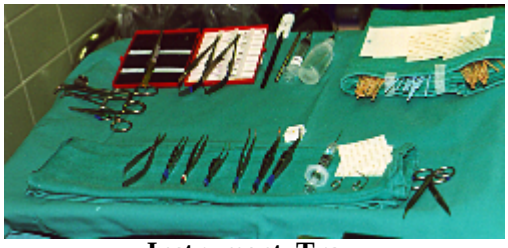
the anterior to the posterior direction. If present, unclotted blood in the vitreous cavity should be drained using a blunt needle. Remaining fibrovascular tissue should be removed by elevating it with a forceps or pick and then cutting with vitreoretinal scissors. After all traction has been removed, rhegmatogenous retinal detachments must be drained through their retinal breaks using a blunt needle. But first, a fluid-gas exchange must be performed to replace all fluids within the chamber with gas so that surrounding fluids cannot re-enter the retinal breaks during drainage. This is accomplished by having the needle remove vitreous fluid in the chamber while the infusion tube repressurizes it with gas. Because gases are lighter than liquids, the inserted gas bubble will fill the anterior section of the chamber first and then proceeds posteriorly until it reaches the tip of the needle.

Once the vitreous chamber is filled with gas, the drainage needle may be inserted into each retinal break to remove any remaining subretinal fluids. Laser photocoagulation is then applied near the retinal break to prevent further detachments. The laser instrument may also be used in the posterior region of the retina after the release of traction to prevent further detachments. Photocoagulation fuses the retina to the choroid causing the affected retinal sections to appear whitish. If subretinal fluids still exist, fusion will not occur. This can be checked visually by noting the lack of retinal color change. If necessary, a final gas-fluid exchange can be applied as a postoperative tamponade. Finally, the instruments are removed and the sclerotomies are sewn shut. Figure 4 shows the variety of vitreoretinal equipment found in the operating room.

Vitrector Machine

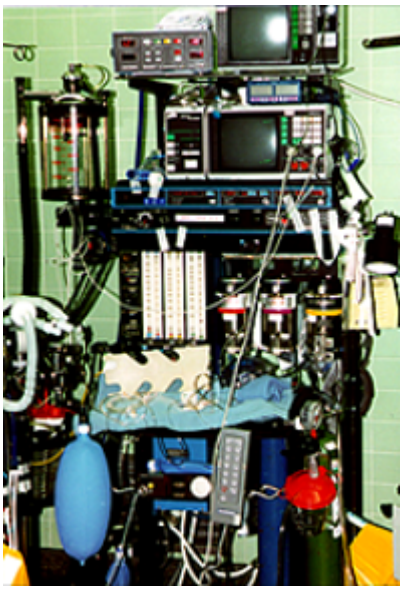


Microscope



Instrument Tray

Foot Petals



Monitoring Equipment

Laser and Light Sources

Figure 4. Vitreoretinal surgical equipment.

1.2 Previous Research

Although the human eye has been studied for hundreds of years, modeling the eye for simulation or for virtual environments is relatively new. The eye's small size and fine structures make it difficult to reconstruct using standard imaging modalities. An early reconstruction of an eye with an ocular tumor was created by segmenting and stacking ultra-sound images to visualize a radiation treatment plan (Lytle, 1989). Another eye model was developed at the University of Auckland to prototype their telesurgery research project (Sagar et al., 1994). Their model focused on superficial and extrinsic eye structures such as the cornea, iris, and surrounding eyelashes and skin. The goal of the project was to train surgeons to perform radial keratotomy on a cornea model using finite element method. Another surgical simulator involving an eye model was prototyped at Interactive Media Technology Center which used a simple model with a tactile stylus that controlled several virtual instruments (Sinclair et al., 1995). Their software package allowed a user to cut the sclera and insert a phacoemulsifier to remove a cataract. Their goal was to develop a general surgical workstation to train fellows and surgeons. The current status of the project is not known.

The model used in this project was originally built in 1988-1989 using published statistical population averages for its internal structures and traced contours from a cadaver for its exterior orbit structures (Parshall, 1991). Figure 5 shows a lateral, external view of the eye model complete with extrinsic musculature.

Figure 5. Eye model.

1.3 Virtual Reality Surgical Simulators

Virtual reality surgical simulators generally consist of five components: anatomical models, physics system, pathology simulation, physical interface, and virtual surgical instruments. Anatomical models are computer representations of surfaces or volumes which approximate the form of the human body. A physics system enables these models to response to computed forces similar to the behavior of their real-world counterparts. A pathology simulation represents an anomaly that needs correction through a surgical procedure. It can be time dependent such as a simulated aneurysm, or fixed such as a tumor. The physical interface includes the real-world devices of the simulator. Lastly, virtual instruments allow the users to interact with the

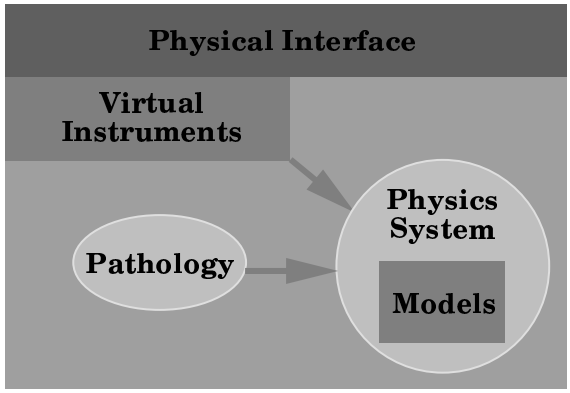


Figure 6. Surgical simulator schematic diagram.

anatomical models through the physical interface. Figure 6 shows a schematic diagram for VR surgical simulators.

1.3.1 Modeling

The creation of medical computer models is a tedious process. The basis for models is a sequential series of tomographic images obtained from physical cadaver slices such as the National Library of Medicine's Visible Human dataset, or from anatomical patient scans using Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) systems, or finally from statistics. For a surface reconstruction, the boundary edge of the desired structure is segmented from the remainder of the image, stacked sequentially in space, and then fitted to a surface representation such as a polygon mesh or Non-Uniform Rational B-Splines (NURBS) surface. Volume representations incorporate the entire image dataset - not just boundaries. Various structures can be emphasized by mapping different rendering attributes to the images' intensity

values. Volume models generally require more computation than surface models due to their increased complexity.

1.3.2 Physics System: Deformable Bodies

To achieve more realistic behavior such as deformation and tearing, static computer models can be coupled with physical properties allowing them to respond to external and internal forces much like their real-world counterparts. Within the human body, a rigid skeletal system supports layers of flexible soft tissues, organs, muscle and skin. In addition, a majority of the body's structures are pressured by the flow of fluids. Rigid structures react uniformly to applied external forces and are governed by the laws of classical mechanics. Flexible structures absorb and transmit some of their applied force into their internal structure causing shape deformation.

Deformations can be quantified as a relationship between stress, strain and material composition. Stress is defined as the internal force per unit area, while strain is measured by the relative displacement of a body from its rest position. When plotted against each other, a general stress-strain relationship can resemble the curve shown in Figure 7. Stress-strain curves generally begin with a linear elastic region whose slope is called Young Modulus. Within this region, deformations are reversible after removal of external forces. Beyond this region, deformations are nonlinear and generally irreversible or inelastic until a fracture point is reached.

Early computer graphics' deformation algorithms were not physically based, but were essentially modeling techniques. Hierarchical transformations which tapered, twisted, bent and sheared models (Barr, 1984), and free-form deformations which warped space to distort the original model (Sederberg and Parry, 1986) were used to obtain a particular deformation shape.

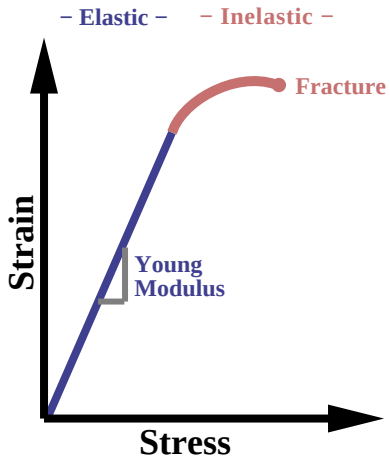


Figure 7. A hypothetical stress-strain curve.

Mass-string lattices introduced a system for physically simulating deformations. This type of system assigns a corresponding mass value to each vertex and then interconnects vertices with damping strings to maintain the model's shape. Springs generate a linear restoring force when stretched beyond their original rest length. To compute the system's next state or next time step, each vertex accumulates its external applied forces and interior forces from its neighboring springs. Then the vertex's new acceleration can be solved by dividing its total force by its mass. To compute the vertex's new position at the end of each time step requires integrating it ahead in time using a forward finite difference method such as Euler, Verlet, or Runga-Kutta (Press et al., 1996).

Since forces can only propagate to their immediate neighboring vertices during a time step, there is a limit to the vibrational frequencies that can be simulated using this technique (Parent,

1996). To overcome this difficulty, an implicit integration scheme can be applied in such a manner that forces are passed to a vertex's neighbors in a similar fashion to the application of a low pass imaging filter (Witkin et al., 1994). This method requires the solution of a linear system of equations which can be subdivided into tridiagonal matrices for effective computation.

Deformation analysis has been a primary application of the mechanical engineering community for evaluating design tolerances. The finite element method (FEM) has become one of their most robust algorithms. The overall goal of the method is to minimize the total energy of the system (Astley, 1992). The method first requires that the model be subdivided into subelements such as cubes, tetrahedrons, and hexahedrons which have known energy equations over their volume. Each subelements' potential and strain energy is determined in terms of the unknown relative displacement of its node vertices. The energy equations are then assembled together into a large sparse matrix whose dimensions are the total number of degrees of freedom at each node of a subelement. Finally, the system is minimized by applying a numerical method such as Gaussian elimination or LU reduction so that each node's displacement is solved. Since the finite element method is discrete, displacements are interpolated between subelements and can be refined to converge towards the exact solution of the system.

Although the finite element method has been applied to mechanical engineering tasks, its use within the computer graphics community has been slowed by its computational requirements. While mechanical engineering's emphasis has been on accuracy and tolerance, computer graphics is more interested in producing near real-time animations using the method. In addition, the base subelements of FEM generally do not include 3D surfaces, the prime modeling staple for

computer graphics. To this end, new approximations of a model's strain energy were based on weighted averages of stretching and bending tensors (Terzopoulos et al., 1987) which could also be extended to include inelastic deformations and fractures (Terzopoulos and Fleischer, 1988). Further optimizations can be applied to superquadrics using this procedure (Metaxas and Terzopoulos, 1992). Unfortunately, this method still requires significant computational overhead and is not applicable to general surfaces. A more flexible method uses 3D orientated particles constrained to a surface with bending and stretching potentials (Szeliski and Tonnesen, 1992). This method allowed surfaces to be molded, joined and cut. But for a given fixed surface, the computation of a particle's neighbors may outweighed its benefits.

Some computational trade-offs to the finite element method have been proposed such as just solving for the surface nodes (Bro-Neilsen, 1996), using the boundary integral formulation (James and Pai, 1999), limiting the degrees of freedom of a node such as excluding torques, and preprocessing displacements into a tensor array (Cotin et al., 1996). Mass-spring networks can be improved compared to the finite element method by making the stiffness coefficients proportional to the spring's adjacent surface area (Gelder and Wilhelms, 1997; Wilhelms and Gelder, 1997). More recent applications of finite element methods related to computer graphics have been the simulation of muscles (Chen and Zeltzer, 1992), cornea deformation (Sagar et al., 1994), and facial modeling (Koch et al., 1996).

1.3.3 Pathology Simulation

Disorders afflicting the human body can vary greatly. Causes range from genetics, diseases, trauma, aging, or from undetermined causes. Simulating disorders within a computer system

would require modeling form using standard modeling acquisitions, to running algorithms that simulate growth, decay and interaction with the surrounding regions. Time dependent measurements of tissue function can be obtained directly from or more often indirectly through newer imaging modalities such as Positron Emission Tomography (PET), functional MRI (fMRI) and Magnetic Resonance Angiography (MRA).

1.3.4 Physical Interface

A simulator's physical interface consists of its real-world devices plus the computer system generating the simulation. Interfaces vary depending on the context of their environments. For a flight simulator, an actual plane cockpit may be used with virtual scenery giving the illusion of flight. Ideally for a surgical procedure, the entire operating room would be reproduced and a monitor or microscope would become the viewport into the virtual patient's anatomical environment.

1.3.4.1 Virtual Reality

Virtual Reality despite its current popularity, is not a new concept but rather one based upon the foundation of interactive computer graphics. Essentially, VR encompasses multimedia applications and supplements them with a three dimensional interface. The standard computer interface simulates a two dimensional desktop in which application windows can be manipulated with a 2D locator or mouse. For simplification, the user's viewpoint is center constrained within the desktop reducing the degrees of freedom. Virtual Reality extends the desktop interface into three dimensions by placing the user inside a partial or complete computer simulated environment which is object centered rather than viewer centered. In addition, VR can incorporate

extra feedback mechanisms such as stereo vision with user-centered perspective, directional audio and tactile sensations. 3D locators such as wands, flying mice, or data gloves enable users to interact with their compute generated environments. Many VR applications react 'naturally' to the user since their position and orientation are monitored in order to constantly update the environmental perspective. An adequate response time is critical for VR to maintain its fluidity to the user. VR allows its users the freedom to explore their environments at their own pace and direction.

The current surge of virtual reality technologies within the last decade has been driven primarily by a quantum leap in computer hardware rendering and processor computation speeds. The origins of virtual reality can be traced back to the mid-1960's Ivan Sutherland's augmented reality head-mounted display (Sutherland, 1965), followed by the Jim Clark's fully computer-generated display (Clark, 1974) in the 1970's, and early VR flight simulators (Rolfe and Staples, 1986). Early VR applications remained simplistic or required expensive specialized hardware to sustain a real-time frame rate. As a result, the computer graphics community focused more on advancing realistic rendering such as ray tracing, radiosity, texture mapping and physically driven animation. Since the majority of these algorithms required minutes to hours of computation, the primary output was prerecorded animations which made the viewer a simple observer of the authors' virtual worlds. During this time, the growth of personal computers and video games launched multimedia applications that integrated computer graphics, audio, and video into their designs. With faster and more economical graphics boards and correspond-

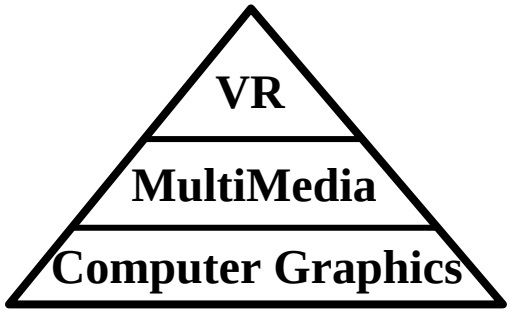


Figure 8. VR hierarchy.

ing increases in computational speed, Virtual Reality has become more accessible. Figure 8 illustrates the foundations of Virtual Reality.

1.3.4.1.1 VR Medical Applications

A number of fields such as medical education, surgical training and planning, rehabilitation, and psychiatry can benefit from the use of Virtual Reality technology (Kaltenborn and Rienhoff, 1993; Durlach and Mavor, 1995). Medical education pertaining to human anatomy and pathology is an obvious application for Virtual Reality. A VR simulator with a digital model of the human body could supplement cadaver dissections by simulating the body's natural appearance, and allowing students the opportunity to change their viewpoint and scale factor within the model to observe a perspective unobtainable in real life (Hohne et al., 1996). Standard visualization techniques could also be applied to reveal the complex relationships between body systems. Although a reasonable full human body simulator plus the technology to process it in real-time is years away, researchers have focused on specific sections of the body

such the abdomen (Satava, 1993b; Merril et al., 1994; Cover et al., 1993), the eye (Sagar et al., 1994; Sinclair et al., 1995), the heart (Mahoney, 1994), and the skeletal and muscular systems.

Little has been published on integrating virtual reality into medical curricula and evaluating its effectiveness. Tactile feedback may be critical in some surgical simulations where medical personnel must learn the correct tactile pressure and response to their actions. Currently, virtual surgical simulators have received more attention than any other medical VR applications. These range from laparoscopic surgery (Satava, 1993b; Merril et al., 1994; Ota et al., 1995), radiocaitonomy (Sagar et al., 1994), cataract removal (Sinclair et al., 1995), arthroscopy (Ziegler et al., 1995) and trauma (Delp et al., 1997). In addition, the new fields of telemedicine and telesurgery allow doctors to consult with or perform surgery on remote patients, share their expertise, or provide emergency care regardless of location (Satava, 1993a; Hunter et al., 1993). VR technology is also being applied in the rehabilitation of disabled patients (Treviranus, 1994), analyzing the walking patterns of patients with Parkinson's disease (Emmett, 1994; Riess and Weghorst, 1995). Finally, behavioral therapy can also benefit from VR technology by helping patients overcome psychological phobias such as fear of heights or flying (Hodges et al., 1995; Hodges et al., 1996).

1.3.5 Surgical Instruments: Cutting

Surgical instruments vary on their functionality, but can be quantified into five general categories: separating, pulling, fusing, removing, and information gathering. Of these, cutting presents the greatest challenge for surgical simulation. An early related method propagated fractures within an object's surface when stress forces exceeded a certain limit by inserting

positional discontinuities in the area where the largest elastic displacement occurs (Terzopoulos and Fleischer, 1988). Another cutting technique used a constrained particle system, but surface area was lost when particle bonds were broken due to the Delaunay surface generation method (Szeliski and Tonnesen, 1992). Another method developed for plastic surgery radially projected a screen based cutting path onto a finite element mesh attached to a laser scanned facial model (Pieper et al., 1992).

Another finite element method moved a template over a 2D surface to define the cutting boundary (Song and Reddy, 1995). For cutting to occur, the user's force vector from the input stylus had to overcome the shear strength of the virtual surface. At this time, this method has not been extended to three dimensions. A more recent method used a single bilinear reference plane to cut into a voxel array derived from a series of CT and MRI images (Mazura and Seifert, 1997). The voxel array was stored in an octree data structure with tetrahedral substructures for finite element analysis. This algorithm outlined an extensive procedure to subdivide tetrahedra based on the cutting plane. Response time ranged from several seconds for subdivision plus another twenty minutes for the cut's depression to become visible due to the finite element computation. Another approach was taken by (Delp et al., 1997), who used boolean operations on 3D surface geometry, but the system lacked an underlying physics system. Other researchers have demonstrated cutting techniques but have not published their algorithms (Sagar et al., 1994; Sinclair et al., 1995). One novel proposal incorporates fuzzy logic to quantitatively evaluate a user's cutting performance within a simulator (Ota et al., 1995).

1.4 Training Issues

Much of the purported potential attributed to medical simulators is based upon the training success achieved in the aerospace industry with flight simulators. Flight simulators have been developed throughout most of this century and became one of the first effective simulators to incorporate virtual reality. As a result, a number of performance and training assessment techniques have been developed to measure flight simulator effectiveness. A trainee's performance is a ranking of their decision-making process and motor skills compared to another performance such as their last session, or to an expert session. Training assessment is the amount of skills that trainee acquired from the simulator that can be applied to real-world tasks. One measure of this transference rate is the Transfer Effectiveness Ratio which is the difference between the transfer performance of a control group to a simulator-based group over the total simulator time (Blaiwes et al., 1973). This ratio is typically used to relate the savings in time from the simulator compared to direct training in the real-world task. Part-task approach subdivides training into subtasks in the belief that overall performance will be improved and be more cost efficient (Wightman and Lintern, 1985). Since simulators can only approximate real-world tasks, simulators must be judged by whether they are sufficient for training. Imperfections in a simulator may lead to negative transfer or learned behavior which is counterproductive to the real-world task, but reinforced by the simulator.

One validation method called the Cooper-Harper characteristic scale allows pilots to rank the handling of a flight simulator (Rolfe and Staples, 1986). This ten point scale is a subjective measurement which asks a set of refined questions to determine if the simulator is unsatisfactory,

needs improvements, or is sufficient. Figure 9 shows a modified Cooper-Harper scale in full detail (Wierwille and Casali, 1983). Another important aspect of a simulator is its workload which is the amount of mental ability required by the trainee to perform an assigned task. As workload increases, the trainee can initially compensate and maintain good performance, but with further increases, performance drops rapidly until the trainee is overloaded resulting in low performance (O'Donnell and Eggmeier, 1986). Workload assessment can be done subjectively, based on performance measurements or by physiological measures.

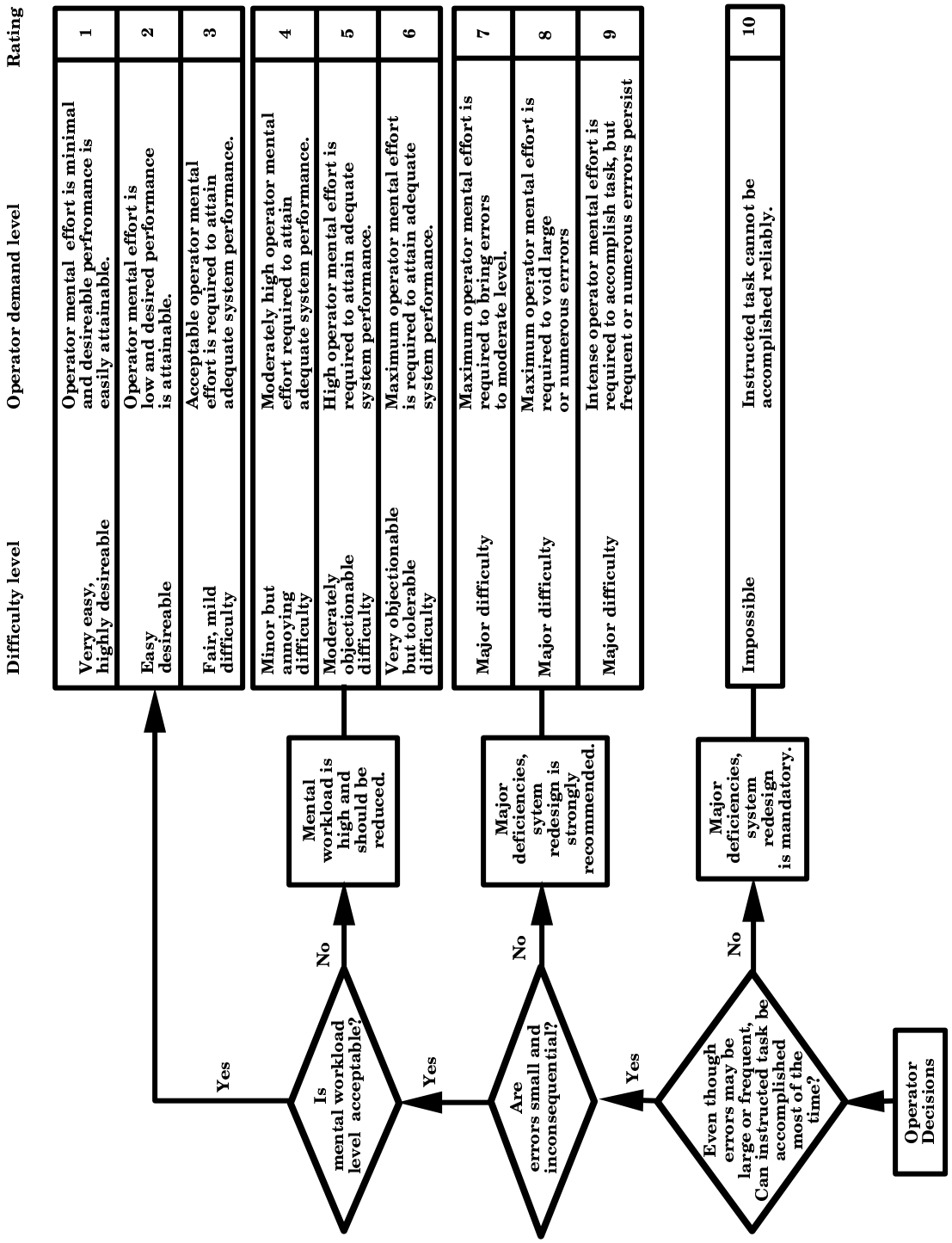


Figure 9. A modified Cooper-Harper handling characteristics scale (Wierwille and Casali, 1983).

CHAPTER 2

VIRTUAL REALITY VITRECTOMY SIMULATOR

2.1 Introduction

Since surgical simulators are relatively new, many of their algorithms have not been completely defined, or have not been published. The override challenge of simulators is to maximizes realism while maintaining interactive frame rates. Since a vitrectomy procedure has not been simulated on a computer before, a manageable scope of work was defined with the available computer resources. Preparatory and closing sections of the surgical procedure were not included since repairing the retina is the primary objective. To compute the required physics, a mass-spring system was selected because of its fast computation and popularity. To simulate cutting and material removal, the mass-spring system had to modified to permit dynamic insertions and deletions of springs and mass points. In addition, a mass distribution scheme had to be devised so that mass is conserved as new vertices are inserted along the cutting boundary. To simulate retinal detachment, traction could be easily accomplished by reducing the default rest lengths of springs resulting a contraction force. The rhegmatogenous case was more challenging since in general fluid mechanics can not be computed interactively. Since the force exerted on the retina is the primary effect in this phenomenon, a unique force function was constructed which physically simulates the retina model being detached. Surgical cases were assembled in a modeling package, brought into the simulator and then saved to disk once equilibrium had

been established. A near complete instrument set was constructed to provide the necessary functionality to repair the simulated cases. Many of these surgical instruments have not been simulated before. As a result, a new cutting algorithm was developed because none of the previous algorithms provided a workable solution for a mass-spring system. The remaining sections of this chapter describe the implementation in detail including models, mass-spring system, detachment simulation, physical interface and each of the simulated surgical instruments.

2.2 Models

Professor Robert Parshall (Department of of Biomedical Visualization, University of Illinois at Chicago) originally generated the cross sections for the eye model. Interior structures such as the sclera, cornea, iris, lens, ciliary bodies and retina were derived from cross-sections from published statistical averages on anatomical curvature and thickness (Parshall, 1991). Original cross sections were 2.2 times larger than life for historical reasons. Because the VR surgical focus is confined to the posterior vitreous chamber, extrinsic orbital structures were eliminated to reduce rendering time. Symmetrical interior structures were generated by sweeping cross sections around the eye's polar axis to construct Non-Uniform Rational B-Spline Surfaces which were then polygonalized into regular triangular meshes or triangular sets based upon shape preservation and display rate. Forty ciliary processes were formed on three primary cross sectional curves.

Other models such as the surgical instruments and infusion tube were modeled using normal conventions. All models were constructed within the Alias Studio software package and exported to a custom file format through a conversion program that uses Alias' OpenModel API library.

A custom file format was created to retain spring state information for physically based models and for consistency with nonphysical or static models. A description of the file format is outlined in Appendix A. When models are read into the CAVE environment, the default centimeter units are converted directly into units of one foot resulting in a 30.5 magnification of the original system. With this magnification, the retinal model fills the entire viewport of the monitor.

A radial texture map coordinate generator was also integrated into the conversion program since it is a natural mapping for many of the eye's symmetrical structures. Radial mapping was computed by dividing the approximate arc length of a vertex over the total length of the cross section. This radial mapping was used on the sclera, choroid, retina, and a discontinuous version for the iris. The choroid color texture map is a modified version of a medical illustration painted by Dr. William Stenstrom (Bell and Stenstrom, 1983) while the iris color texture map is a photograph (Long and McNally, 1992). All static models were stored in an OpenGL display list for efficiency if shadowing was not enabled. Figure 10 shows exterior and internal views of the VR model with the pick instrument and light probe inserted into the posterior chamber.

2.3 Physics System

2.3.1 Structures

Flexible structures such as the retina, vitreous gel, and fibrovascular tissue were modeled as above, but the conversion program fitted a vector spring through each edge. The spring's rest length and direction were initialized using the two vertices of an edge, and its stiffness coefficient was computed based on its surrounding surface area (Wilhelms and Gelder, 1997). The contraction of the fibrovascular tissue model was accomplished by scaling down the springs' nat-

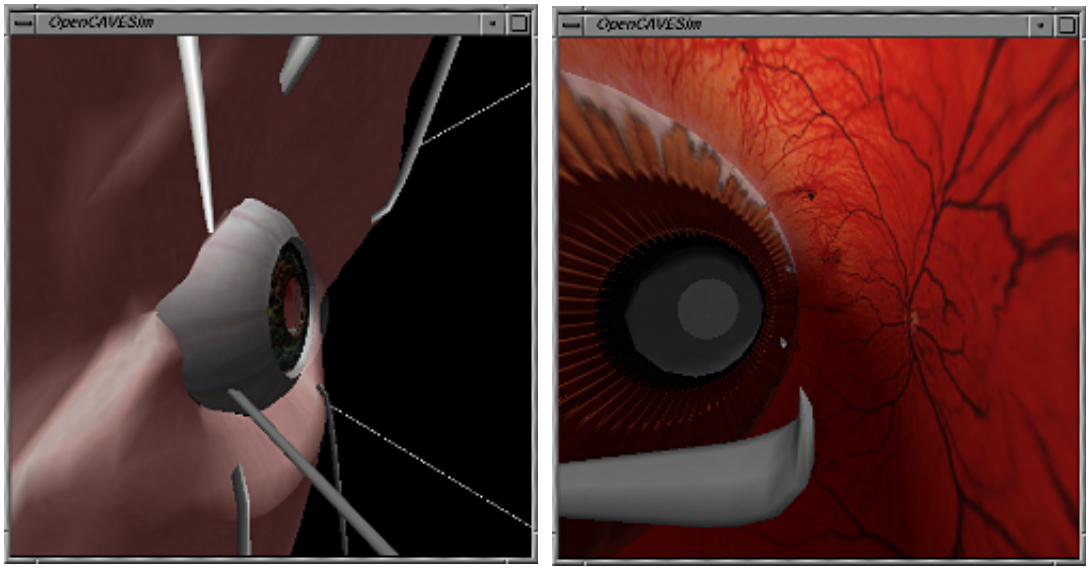


Figure 10. Exterior and internal views of the VR model.

ural rest length by a percentage. A global percentage could be assigned to the entire structure, or individual percentage could be set for a particular spring. Attachments between different physical surfaces, such as fibrovascular tissue and retina, were implemented through instance vertices, which were not integrated but simply contribute their local forces to the master vertices. Instance vertices were identified automatically during the conversion process. In addition, individual model vertices can be anchored or fixed to prevent them from being affected by forces. Fixed vertices within the retinal model mimicked the retina's natural attachment locations in its outer peripheral region near the ora serrata and around the optic nerve. The vitreous gel

was approximated as stacked coronal cross-sections aligned with the retinal vertices so that they would deform together.

2.3.2 Flexible Body Algorithm

The physical system is composed of three layers: an integration module, vector springs and physical vertices. The integration module updates polygonal surfaces composed of vector springs connected to physical vertices. Physical vertices consist of the usual fields such as position, surface normal, texture coordinates with additional fields for velocity, acceleration and force vectors, mass value, adjacent spring list, a possible instance list, local force functions list, and finally a fixed/instance flag variable. In addition, the vertex's position, normal and flag variable were double buffered so that the computation process can update these quantities without interfering with the display processes. A complete list of physical fields can be found in Appendix B.

The vector spring was created by Alan Millman, Electronic Visualization Laboratory, University of Illinois at Chicago. Its significance is that these types of springs maintain their orientation as well as their length. Fields consist of a rest direction vector, an active flag, and stiffness and dampening coefficients. Vector springs work well for flexible bodies since they maintain their spacing and orientation. The force of a vector spring is computed with Equation 2.1.

$$force = ((rest\_direction - current\_direction) * Stiffness) - (\Delta Velocity * Dampening) \quad (2.1)$$

Surfaces contain a list of physical vertices and vector springs as well as a list of active global force functions such as gravity, intraocular pressure or viscosity. The system integration is started by computing the system's next time step which is based on the CAVE's timer in six substeps before rerendering all of the models' geometry. For each substep, all springs compute their forces, then each physical vertex sums up its external forces by accumulating its adjacent springs' forces and adding any global or local force functions. Using this accumulated force, a vertex's new acceleration can be derived by simply dividing the force vector by its mass. A Verlet update was then used to determine the vertex's new position as outlined below:

$$Position+ = (Velocity * \Delta Time) + (Acceleration * \Delta Time^2)$$

$$Velocity+ = (Acceleration * \Delta Time) + (\Delta Acceleration * \Delta Time^2)$$

$$Acceleration+ = \Delta Acceleration$$

Once all of the substeps are finished, surface normals were recomputed before the scene is redrawn. Since the vertices' position, normal and flag variable were double buffered, the computational process computed the new time step while the display processes were redrawing the last computed one. To update the drawing processes, the computational process toggled an index to have the drawing processes render the newly computed state. The integration scheme outlined above can be parallelized by forking off extra slave computational processes and assigning each a unique section of springs, vertices and normals to update. Synchronization is required so that all spring forces are computed before the vertices' positions are updated, and

conversely so that the spring forces of the next substep are not computed before all vertices of the previous substep are finished. One exception is that the normal computation cannot be parallelized because adjacent triangles may be assigned to different processes which will in turn try and access similar vertices.

For insertions and deletions, the vertex, spring and triangle lists were maintained in large shared memory arrays with extra space for insertions. Each array has three associated markers: one for the active linked list, another for the deleted entries and finally a pointer to free space. When a new entry is requested, the deleted entries are checked first and if none are present, a new entry is returned and added to the table's active list. Cutting generally increases the number of vertices, springs and triangles while the suction cutter removes more entries than it inserts. For integration, the active entries are partitioned between the number of available computational processes at each time step.

A simple collision detection routine was implemented due to the computational overhead on a full detection system. Physical models were prevented from leaving the vitreous chamber by a large boundary sphere placed around the choroid model. This boundary stopped the retina, vitreous gel and fibrovascular tissue models from penetrating the choroid model. During integration if a vertex moves outside the boundary sphere's radius, its exact intersection with the boundary sphere is computed and then its position, velocity, and acceleration are reflected around the sphere's normal. Unfortunately the choroid model is not exactly spherical so anteriorly the boundary sphere does not aligned with its surface permitting some penetration. No other collision detection is performed.

2.3.3 Physical Properties

In order to drive a physical simulation, measured real-world elasticity material parameters such as the Young Modulus, Poisson coefficient and mass are needed for the retina, fibrovascular tissue, vitreous gel, and liquid vitreous so that they behave properly. Appropriate force magnitudes for intraocular pressure and viscosity are also needed. However as of this writing, approximate values have been substituted in order to run the computations. Some values have been published: relative density of the retina is 1.0174 g/ml compared to liquid vitreous 1.0033 g/ml (Michels et al., 1990), and intraocular pressure is approximately 13 mmHg near the retina (Fatt and weissman, 1992). Different pathologies may also alter these values as a particular disease progresses. Newer scanning devices such as Magnetic Resonance Elastography can now measure elastic properties of a particular patient through the use of acoustic strain waves (Muthupillai et al., 1996). In the future, this technique may make general elasticity properties more easily available.

Mass distribution was accomplished in the simulator using an equation similar to Wilhelms' stiffness coefficient equation (Wilhelms and Gelder, 1997), by assigning a vertex's mass proportional to its neighboring surface area at its initial rest position. When springs are subdivided or removed, inserted vertices and their immediate neighbors are flagged before integration and have their mass values recomputed based upon their new triangle configuration. Ideally as an object deforms and surface changes, mass should be redistributed after every time step. However, this would degrade response time so to maintain efficiency, mass is updated locally based on the initial mass to surface area ratio. This method conserves mass within the system. As

triangles are removed from the system, the total mass of a structure is decreased according to the percentage of surface area lost.

Global force functions defined within the system include gravity, intraocular pressure and viscosity. The force of gravity is computed by dividing the earth's gravitational acceleration constant by the vertex's mass. Intraocular pressure is the radial force on the retina from absorption of nutrients. Viscosity is a drag force through a fluid field such the vitreous gel. Viscosity is computed by scaling the inverse of the vertex's velocity vector.

2.4 Pathology

Several pathological cases were constructed that required surgical correction: anteroposterior traction, rhegmatogenous detachment, a combination of perpendicular and circumferential traction, and simple fibrovascular tissue blockage. To approximate a rhegmatogenous detachment, a simple local radial force function was implemented which pulled selected retinal vertices toward toward the center of the vitreous chamber. This scheme physically simulates the shape of a detachment without incorporating any fluid mechanics. Retinal tears were initially texture mapped for simplicity, but in later simulations were modeled in 3D. Traction cases were modeled by constructing circular fibrovascular tissue and attaching it to the posterior section of the retina model via common instance vertices. Reducing the tissue model's springs natural rest lengths generated an inward contraction force, which caused the model to shrink pulling the retina with it. Figure 11 shows the retinal model in profile in an undeformed natural position, followed by the same model constrained by a fibrovascular tissue model.

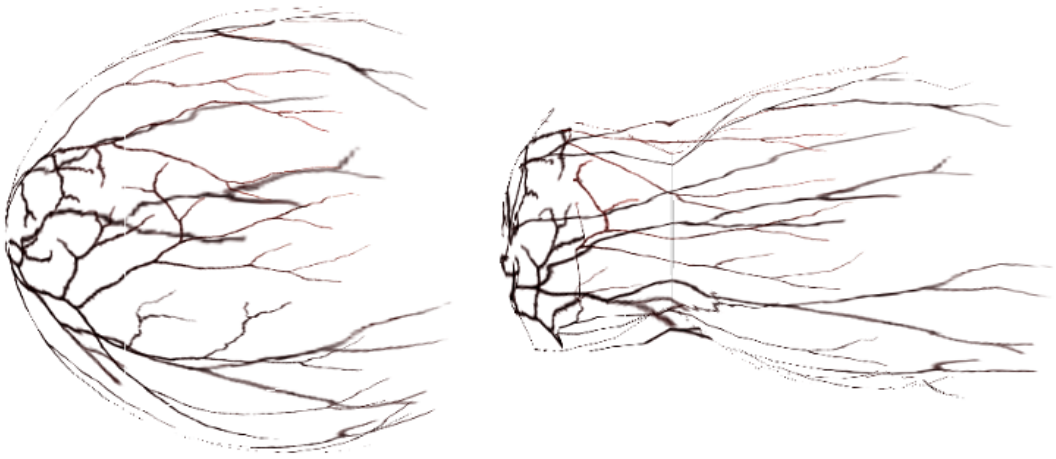


Figure 11. Retinal model: normal anatomy and deformed by traction.

2.5 Physical Interface

Current VR implementations vary greatly in their features and performance. The VR system developed in this project, shown in Figure 12, consists of a Silicon Graphics Onyx^2 rack workstation with InfiniteReality^2 Engine graphics board, duel R10000 250MHz CPU, 1280 megabytes of main memory, 256 megabytes of texture memory, two Ascension miniBIRD trackers attached to disposable vitrectomy instruments, and CrystalEyes stereo glasses. The monitor is a 21 inch color screen with 1280 by 1024 stereo resolution. Since the miniBIRD system was magnetic, its transmitter and sensors were placed on a smaller wooden table underneath the main computer table. This was necessary because metal in the main table and monitor interfered with the

sensitivity of the miniBIRD system. To elevate the subjects' hands, several books with wrist pads were used. Head tracking was not enabled due to a lack of an available serial port.

The system is compatible with the CAVE and Immersadesk VR systems developed by the Electronic Visualization Laboratory at UIC (Cruz-Neira et al., 1993). Although this VR system doesn't support tactile feedback, vitreoretinal surgical maneuvers do not generate strong contact forces, and are generally regarded as an entirely visual procedure. With two CPUs in the current system, parallel integration was attempted but did not yield an increase in performance due to the synchronization overhead. Parallelization routines were tested on another system with eight additional computational processes and achieved a noticeably faster response time.

Software is comprised of C++ classes created by using the OpenGL graphics library and the CAVE library which emulates our application running within the CAVE environment (Pape, 1996). The CAVE library supplies a framework for applications by initializing window parameters, monitoring tracking devices and rendering synchronization. CAVE applications are generally composed of two independent sections: a main controlling process which performs initialization and the application's computation, and rendering routines that simply redraw the generated scene. Since the rendering and computational processes must share information, the application's data was placed in shared memory and semaphores were used to synchronize updates.

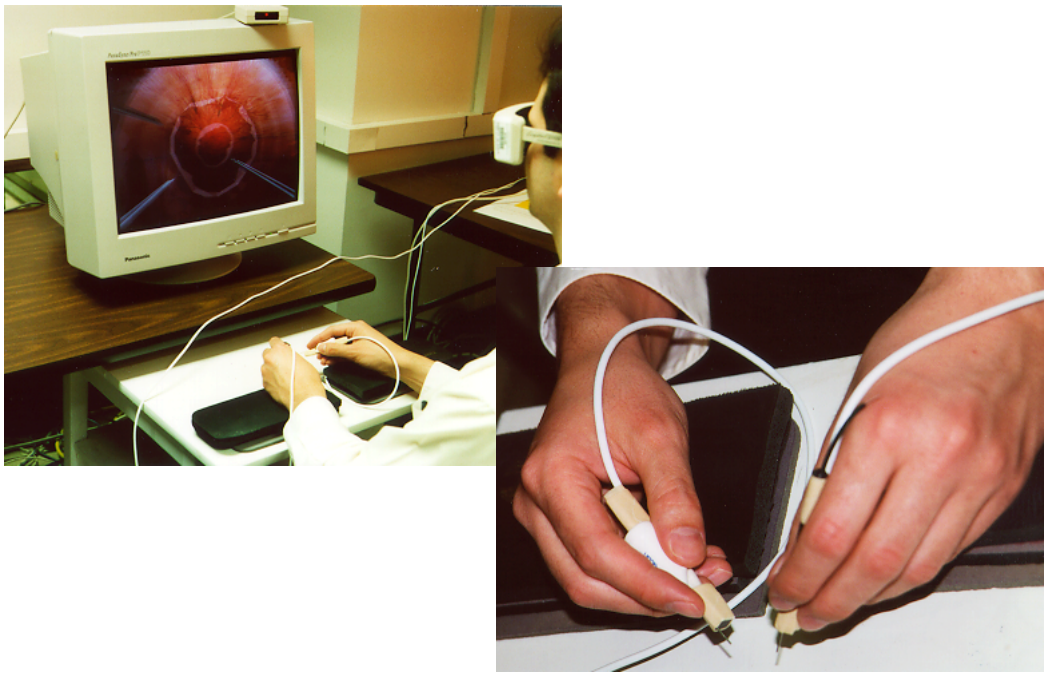


Figure 12. A fellow using the simulator.

2.6 Interaction

2.6.1 Wand Interface

Surgical instruments are inserted into the posterior chamber above the pars plana through a sclerotomy. Through these incisions, instruments are extended and rotated within the cavity. In order to move an instrument to match the sensor position, the instrument's extension was computed first by determining the directional vector from the insertion position to the sensor's value. Since the Z axis is the default alignment for all instruments, instruments were translated along the Z-axis by the magnitude of this vector. The cross product of the directional vector and the Z axis is the axis of rotation, while its magnitude is the sine value of the needed rotation. The twist of an instrument can be directly obtained from the sensor's value. Instruments are always rotated about their incision point. Figure 13 illustrates the extension and rotation of the blade instrument to match a new sensor location.

The intersection of an instrument with a surface such as the fibrovascular tissue or retinal model involves testing each vertex of the surface against a small spherical region surrounding the sensor's position. If a spring needed to be tested then a general line/sphere intersection routine is used. Since an intersection test can be computationally expensive due to the large number of vertices and springs in a scene, each surface declares in advance which instruments it will react with. In this way, only the required surfaces were tested for intersections. The intersection test was not parallelized. Generally the intersection of an instrument and a surface is only of interest when the user is ready and has positioned the instrument. Since the miniBIRD tracking system did not have any input selection devices such as buttons, switching between instruments

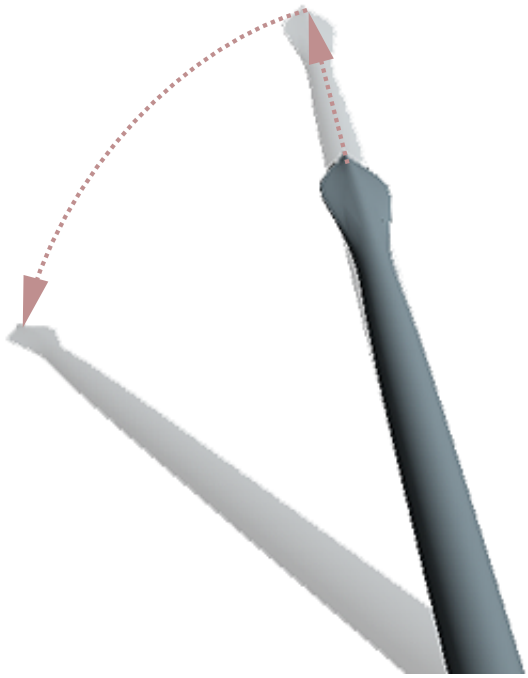


Figure 13. Instrument alignment.

TABLE I

KEYBOARD MAPPING

Key	Action
Left Ctrl	Toggles Activation of Light Probe's Pick
Right Alt	Toggles Activation of Operative Instrument
Right Ctrl	Switch to Next Operative Instrument
F5	Zoom In
F6	Zoom Out
F9	Next Scene
F1	Toggle Fill/Line Mode
F2	Toggle Flat/Smooth Shading
F3	Toggle Labels for Debugging
F4	Toggle Integration
F12	Saves Physical Structures to Disk

and activating them were mapped to particular keys on the keyboard. Table I summarizes the active keys within the program. Figure 14 summaries all of the available surgical instrument models.

2.6.2 Surgical Instruments

2.6.2.1 Blade

Eight surgical instruments: two blades, suction cutter, pick, forceps, laser, and drainage needle plus a light probe are currently implemented in the simulator. None of the instruments

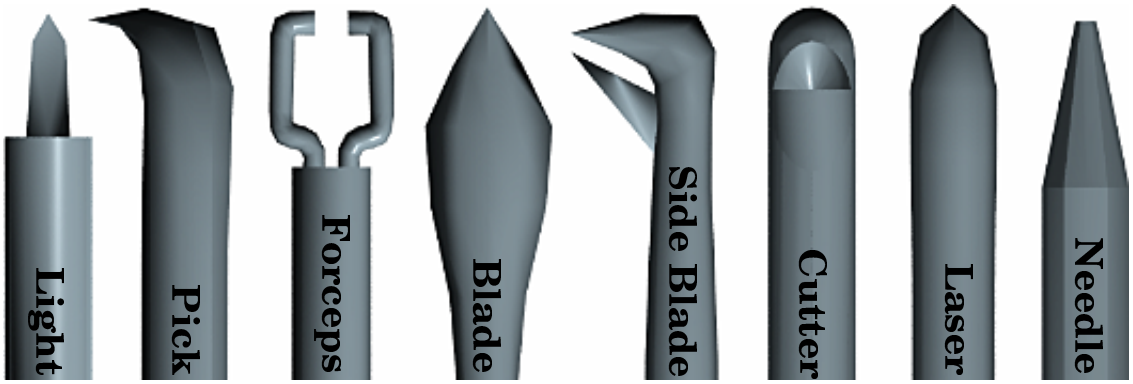


Figure 14. Surgical instrument models.

has any moving parts. The blade's cutting algorithm first samples the blade's motions, intersects this path with the surface springs and then subdivides surface triangles based upon the new spring configuration. For simplicity, the blade's path is assumed to be moving at a relatively slow rate with no high frequencies artifacts. Also, the surface triangles should be smaller than the blade's path so that the path can be reconstructed by connecting spring intersections with a linear interpolation function. Sampling the blade's motion can be done by discretely recording the instrument's position and orientation at each CAVE frame. If the current sample does not deviate in a linear fashion from the previous two samples, the previously recorded sample is dropped in favor of the current one. When two consecutive instrument positions have been determined, their orientations are averaged to form a common plane through both samples. So that the user does not have to intersect the surface exactly in order to cut it, a small offset along the common orientation is assumed. The two sample points plus their offsets form an

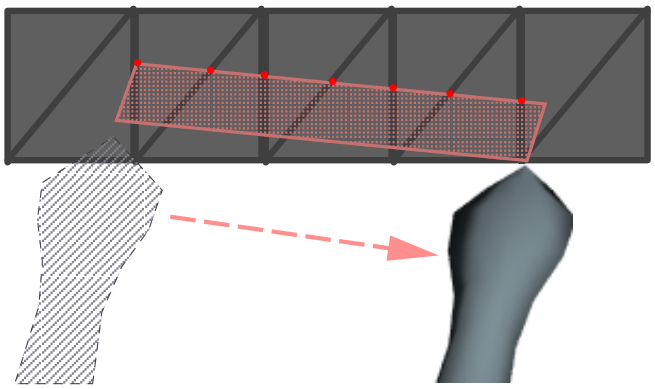


Figure 15. Cutting intersection parallelogram.

intersection parallelogram which is compared with the surface springs. Figure 15 illustrates a sample intersection parallelogram.

2.6.2.1.1 Intersection and Subdivision

To determine which springs intersect the parallelogram, two steps are performed: a standard line-plane intersection routine (Glassner, 1990) followed by a quadrilateral inverse mapping (Haines, 1989). A parallelogram can intersect a triangle in six ways as illustrated in Figure 16.

When a spring is confirmed by the intersection routine, two new subsprings are created around the intersection point and linked back to the original spring. In the vertex to vertex condition, the spring is copied and each vertex's spring list is separated based upon which side of the parallelogram it falls. Any triangles adjacent to the affected spring are placed on a subdivision list. After all springs have been intersected, each affected triangle is subdivided

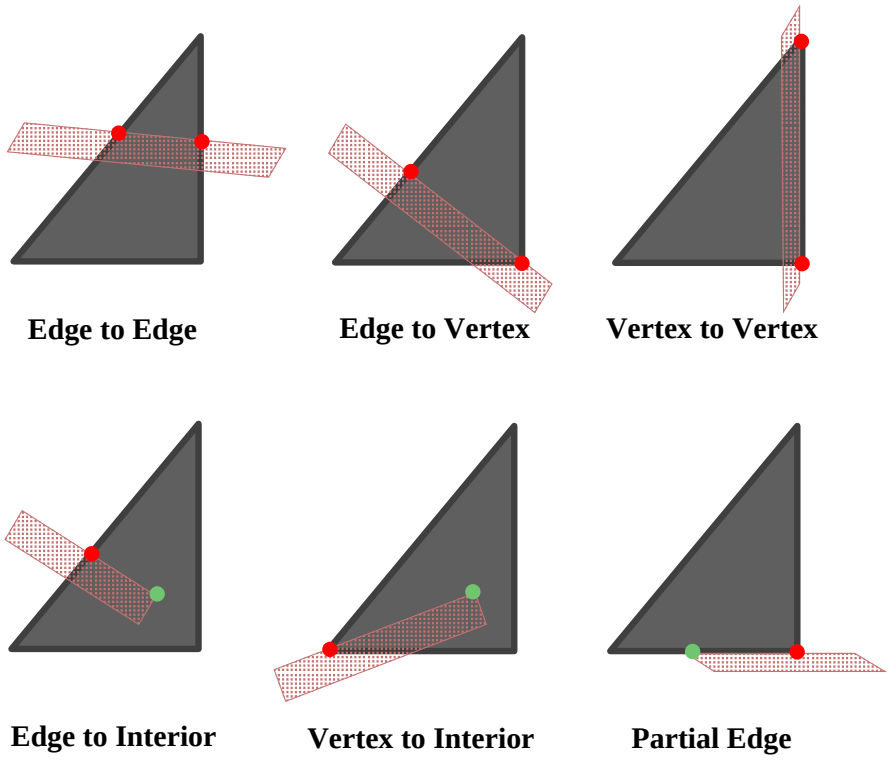


Figure 16. Triangle intersection types.

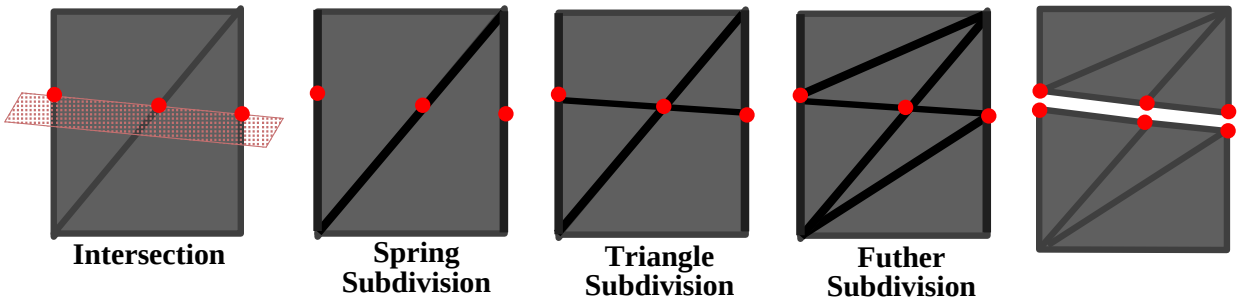


Figure 17. Triangle subdivision.

along its new subsprings by forming a counterclockwise list of vertices from its new spring list. This vertex list can be separated between intersection pairs to subdivide the triangle. If after subdivision, a quadrilateral is formed such in an edge to edge case, it is subdivided once again along its diagonal to generate two subtriangles.

Newly created interior springs must compute their rest length vectors through vector addition of the adjacent vectors within their triangle. After all triangles have been subdivided, stiffness coefficients for each newly created spring is computed based upon their adjacent surface area. Figure 17 illustrates the steps of the subdivision process. Since the drawing processes are in parallel to the computation process, the detection of new triangles must be synchronized with the beginning of the drawing cycle by double buffered their active flag. Figure 18 shows a sequence of screen images of the blade cutting a simple surface. In these images, the red outline is the current parallelogram while the purple line indicates the next parallelogram's position.

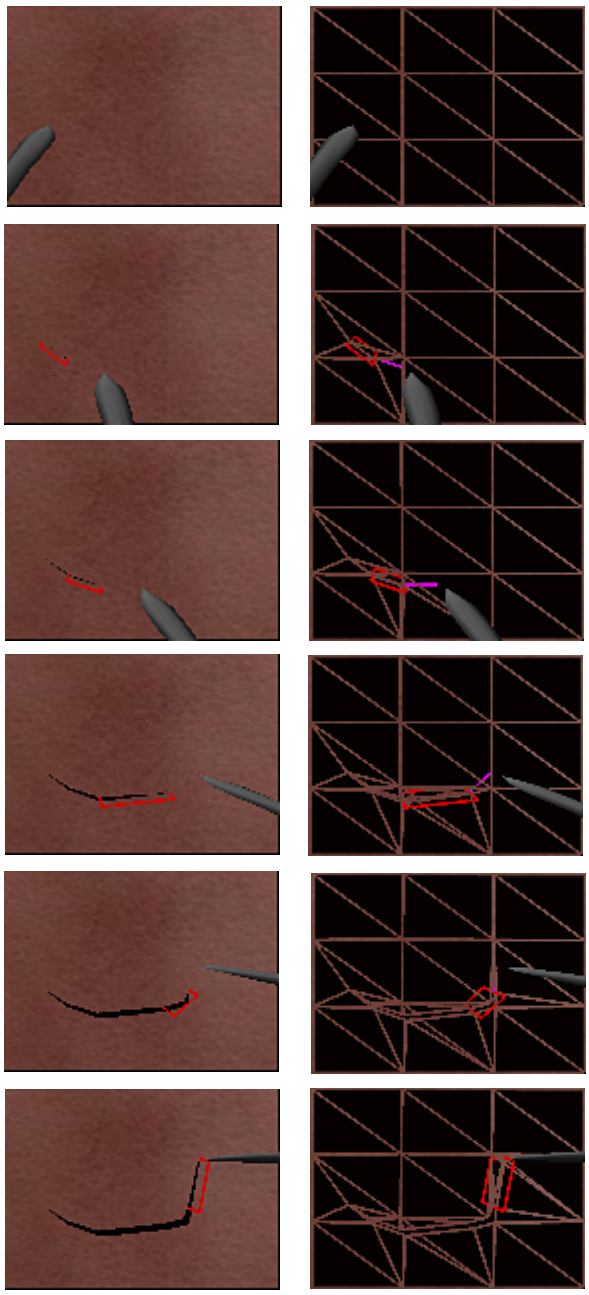


Figure 18. Cutting sequence: shaded model (left) and spring wireframe (right).

A unique situation arises when a triangle is partially intersected, such as in the case where the cutting path begins or ends within a triangle's interior, or when a surface slopes away from the cutting parallelogram. In this case, a triangle will only have one edge intersected so a matching interior intersection must be computed. This can be derived by first locating the opposite edge from the intersection which has a valid plane intersection, but an invalid quadrilateral inverse mapping coordinates. Since the boundary of the parallelogram lies at zero or one in parametric space, the intersection parametric coordinates can be linearly interpolate to the opposite edge's coordinates. This derives the interior point position in parametric space which then can be converted into Cartesian space. Once the interior intersection point is determined, the original triangle can be subdivided into three subtriangles through this location.

Three new internal springs from the original vertices to the interior point will be created during this process. Determining the rest length vectors for these new springs must be done in two steps. The first step is to project one of the new interior springs through the entire triangle and solve for its total rest length from the original springs. Once the complete rest length for this extended edge is computed, it can be clipped back to its original size. The remaining internal springs can then be solved normally through vector addition of their neighbors. The subtriangle which contains the original valid intersection can then be subdivided into two separate subtriangles in order to close or open the cutting boundary. Figure 19 illustrates this process.

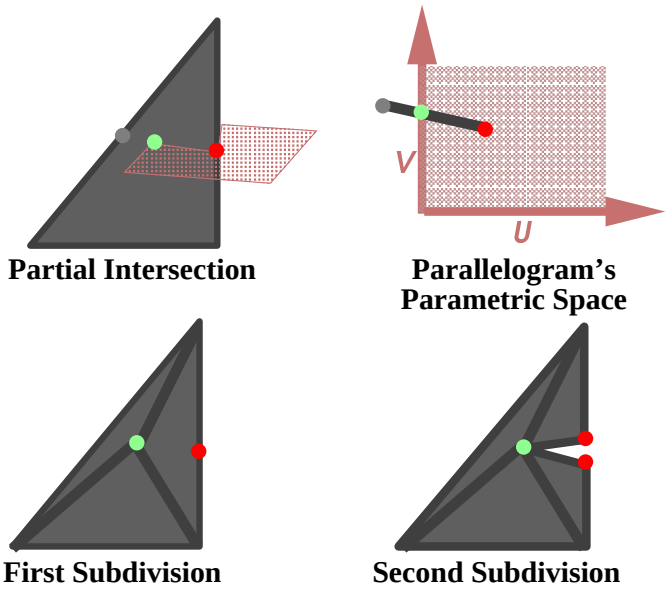


Figure 19. Interior subdivision.

2.6.2.1.2 Special Cases

Special cases occur when a parallelogram just grazes a spring or triangle. In the spring case, only one intersection point is generated. Grazed triangles are subdivided in half with a new interior spring. A triangle is categorized as grazed when the parallelogram's intersected length is less than 10% of the total projected length. In a vertex intersection case if a neighboring triangle is only grazed then the intersection is canceled. Figure 20 illustrates a grazed spring condition before and after processing.

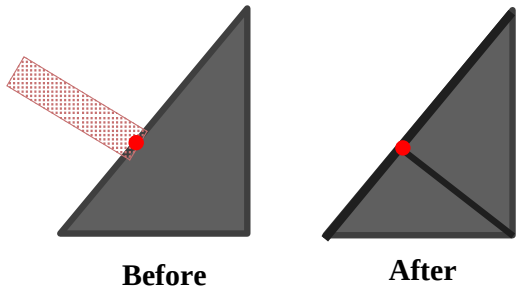


Figure 20. Grazed spring.

2.6.2.2 Suction Cutter

The suction cutter simulates material removal by attracting nearby vertices towards its opening and then clipping them when they are too close. No published algorithm has simulated this particular instrument. Vertices are drawn to the cutter's opening with an attraction force function which is assigned to all intersecting vertices within its surrounding hemisphere. The attraction force magnitude is scaled by the angle between the direction from vertex to the cutter's opening and the current suction direction. Vertices are clipped when they intersect a smaller nested removal hemisphere inside the attraction field. Vertex removal is similar to the cutting algorithm except no double intersection vertices and edges are needed. Subtriangles containing the intruding vertices are only computed for their undeformed surface area which is used to decrease from the object's total surface area and mass. If a vertex exits the attraction hemisphere, the force function is removed from it and a small reverse force is added to slow

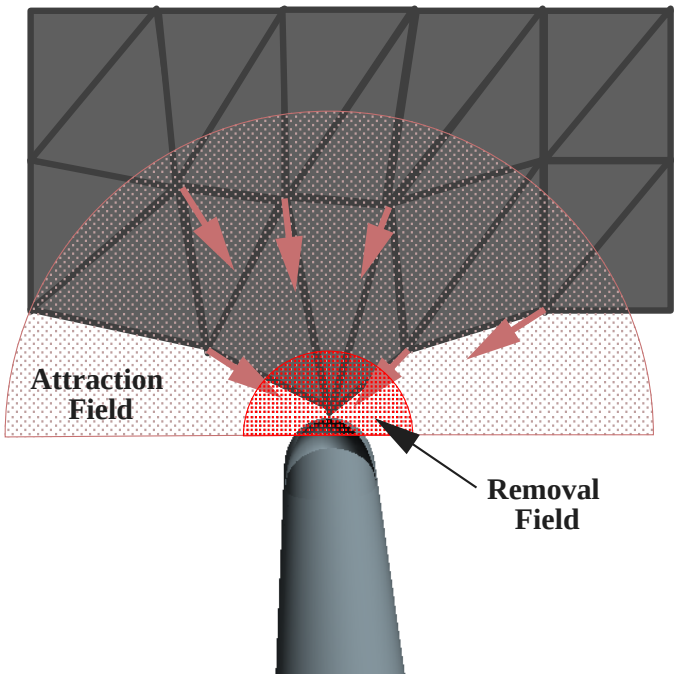


Figure 21. Suction cutter attraction and removal hemispheres.

its momentum. Figure 21 illustrates the cutter's hemispheres while Figure 22 shows all of the possible intersection cases for a triangle with the suction cutter's removal hemisphere.

One problem encountered with this algorithm is that a large number of new vertices are generated with small mass values. When these new vertices are accelerated by the suction's attraction field, they make large positional jumps over a single time step. Although this is the correct behavior from the integration routine, it is highly undesirable for the stability of the system. To correct this behavior, large accelerations are clamped to a maximum allowable value. Only the magnitude is effected by this procedure, the acceleration direction was not

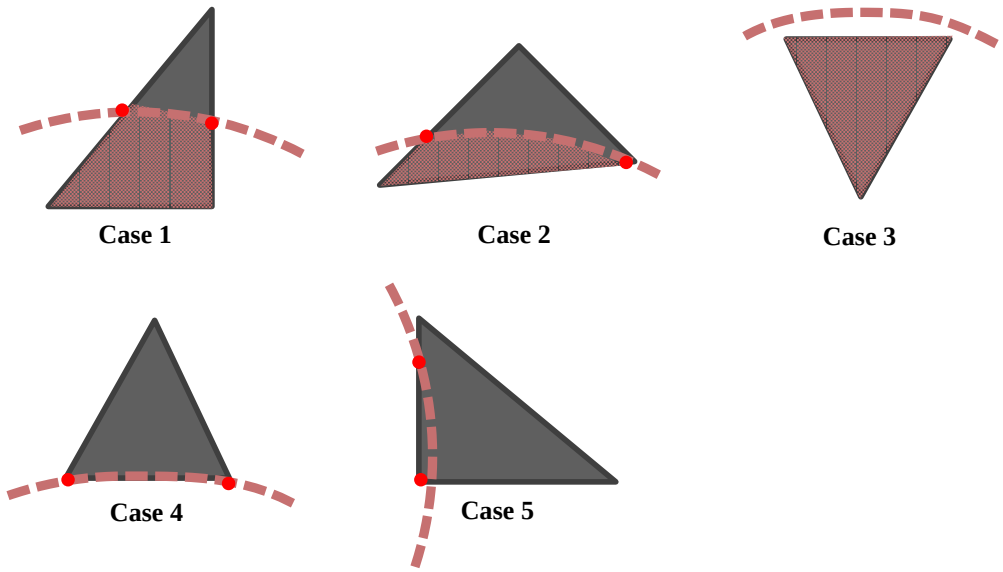


Figure 22. Suction cutter intersection cases.

altered. Since new vertices are generated on the boundary of the removal hemisphere, the majority of them will be removed in the next time frame as they are drawn in by the attraction field.

2.6.2.3 Pick/Forceps

The pick and forceps are primarily used to elevate tissue. The light probe is coupled with the pick while a forceps is available as a general operative instrument. Both have the same functionality. When either one intersects a surface, all surface vertices within a small intersection sphere are linked to the instrument's position and move with it as the instrument moves. Nearby vertices react by the force propagation of their shared springs. Figure 23 shows a screen

image of the forceps instrument elevating a cut boundary. The light probe was implemented with a local light source or with the use of projected texture maps (Segal et al., 1992) to produce spotlight and shadowing effects.

To implement fractures, a spring is subdivided into two detached subsprings if it is stretched beyond a set tolerance assigned to the surface. Tissue from the human retina can be stretched from 35% to 55% of its length before tearing (Michels et al., 1990). Tearing is again similar to the cutting algorithm with some exceptions. When a spring reaches its limit, two new subsprings are created about its center with two independent fracture vertices. Again, any adjacent triangles to the spring are placed on a subdivision list. After all springs have been updated, each triangle on the subdivision list is divided according to the number of fractures. If the triangle has only one fracture spring then it is subdivided into two subtriangles, by connecting the fracture point to the vertex located on the triangle's opposite side. If the triangle has two fractured edges, then it is processed identically to the cutting algorithm. Three fractures would subdivide the triangle into four separate triangles with the center triangle being completely detached from the others. Figure 24 illustrates a possible fracture condition and its resulting triangle configuration.

2.6.2.4 Drainage Needle

Before draining trapped fluid underneath a retinal tear, a fluid-gas exchange is performed to remove any fluid in the vitreous chamber above the tear. Although computing the exact boundary between the gas and fluid regions cannot be done in a reasonable amount of time, a height ramp function based upon the Z coordinate can approximate two regions with different intraocular pressure and viscosity values. When the needle is activated within the chamber,

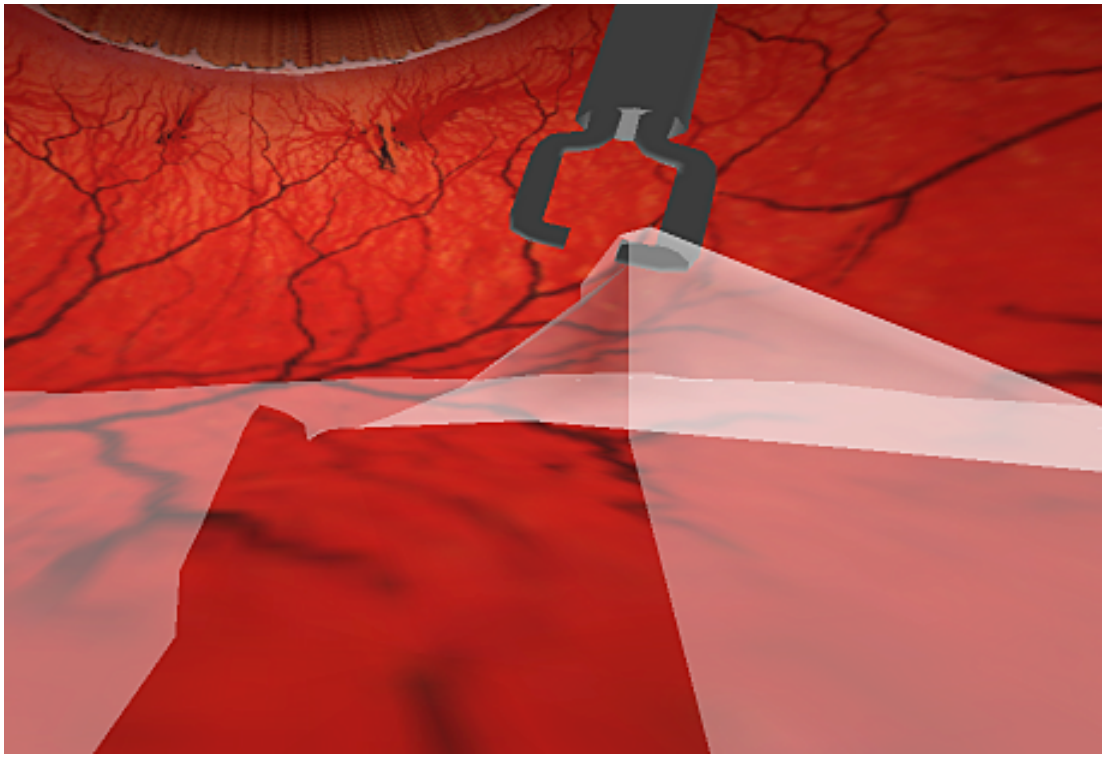


Figure 23. Forceps elevating a cut boundary.

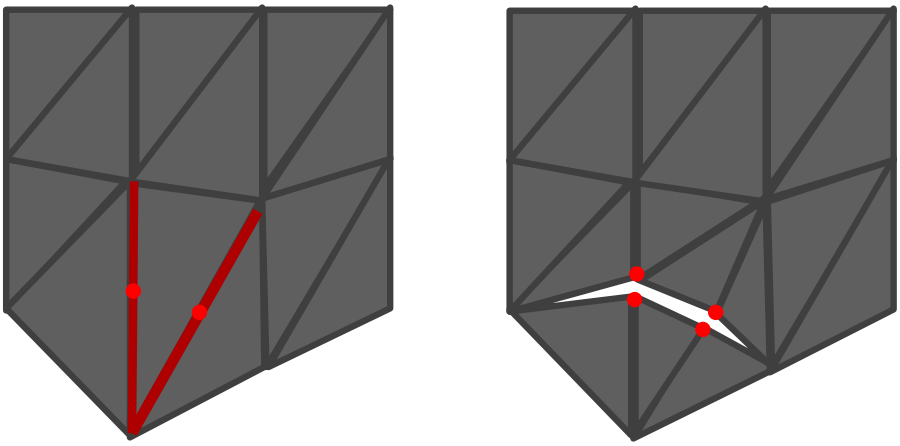


Figure 24. Fracture example.

its Z coordinate is compared to a global fluid level variable that is initialized to the top of the chamber. If the needle's position is lower than this value, then the fluid level variable is decremented for each time step until it reaches the needle's current position. A transparent plane visually indicates the current position of the fluid boundary within the chamber. During integration, intraocular pressure and viscosity force functions can determine a vertex's region by comparing its Z coordinate with the current fluid level variable.

To simulate trapped fluid underneath a retinal tear, a detachment force function pulls a set of retinal vertices toward the center of the vitreous chamber. When the needle is used to drain a tear, it checks to see if all of its intersecting vertices are above the current fluid region, or else drainage will not take place. If so, then the detachment force function's magnitude is decreased every time step while the needle is activated until the magnitude reaches zero.

2.6.2.5 Laser

The laser instrument adheres the retina to the choroid providing it does not have any external forces acting upon it. The affected retinal tissue is destroyed and changes to a whitish color. In the simulator, retinal vertices are selected with a line distance routine based on the laser's orientation vector (Glassner, 1990). If the vertex is near its rest position, then it is flagged as fixed so that it cannot be integrated and its texture coordinates are averaged with other intersected vertices. If the vertex has a force acting on it such as the detachment force then the vertex is ignored. If the majority of the intersected vertices are flagged as fixed, then a small whitish circular template is blended with the retinal texture map at the group's average coordinates. For aiming, a green project line is drawn from the laser's tip to the retina. Figure 25 shows a screen image of the laser instrument adhering around a simulated retinal tear.

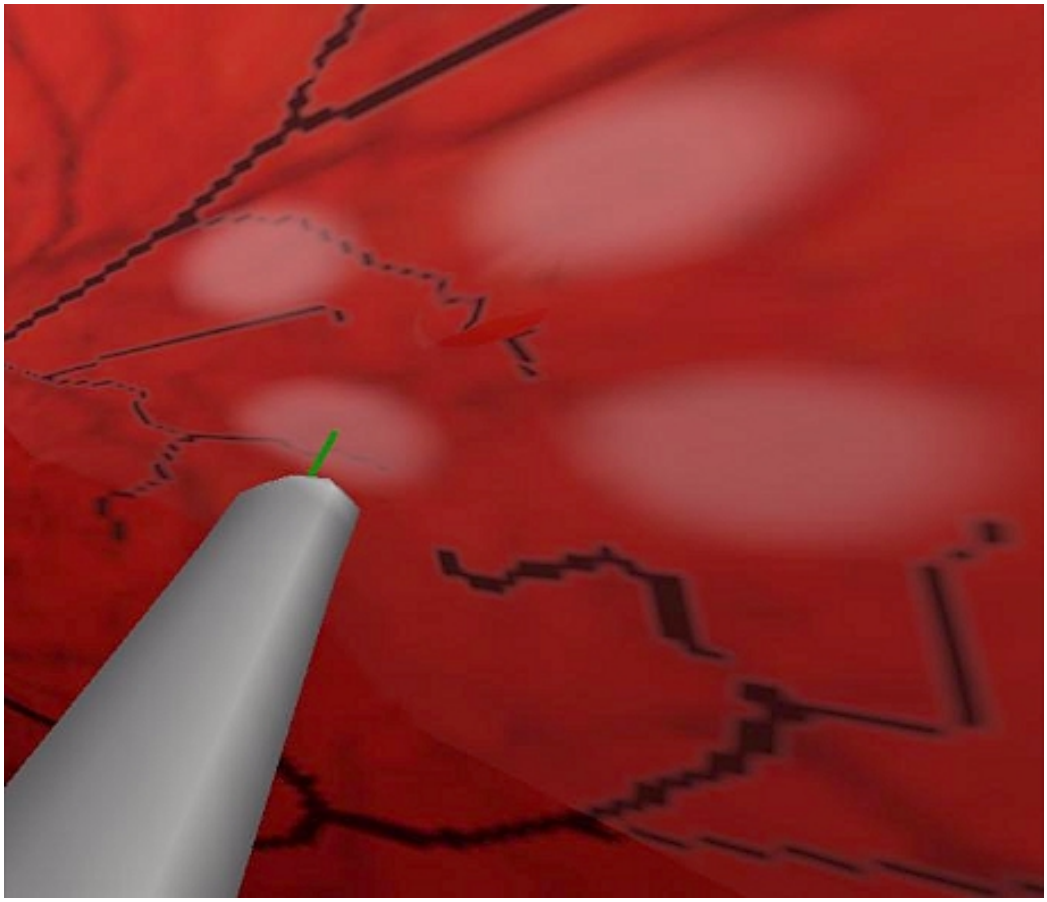


Figure 25. Laser adhering around a tear.

CHAPTER 3

EVALUATION

As a first step in evaluating the simulator, an initial survey was designed to determine the simulator's training potential for ophthalmology fellows. To this end, a set of four vitreoretinal specialists (three surgeons and one advanced fellow) were asked to process three simulated surgical cases and then rank the simulator's training potential through a modified Cooper-Harper survey. This procedure is similar to an initial pilot survey administered in flight simulators. Once the ranking is known, subjects were asked to comment on the image quality, modeling detail, tracking accuracy, physical interface, instrumentation and an optional write-in category. Since instrumentation is the focus of this study, a detail ranking is also included for each instrument. A space at the end of the survey permits subjects to enter any general comments. The complete survey form can be found in Appendix C. There was no time limit set on any of the testing. Subjects could take as little or as much time as they needed on any particular case or exercise. All trials were performed in the Virtual Reality in Medicine Laboratory. Figure 12 shows one of the fellows using the simulator during the first trial.

3.1 First Trial

Subjects were first presented with five practice exercises with each surgical instrument. The motivation for the practice exercises was to permit the subjects to adjust to the physical interface. Practice exercises consisted of removing small sections of gel, cutting along an outlined

path, pulling on virtual tissue, draining a simple tear and then adhering it. Figure 26 shows screen images of the practice exercises. Afterwards, subjects were presented with three surgical cases and were asked to repair any retinal detachments with the given instruments. The cases consist of anteroposterior traction, rhegmatogenous detachment, and a combination of perpendicular and circumferential traction as shown in Figure 27. Afterwards, subjects filled out the survey form.

Since the miniBIRD tracking system does not have any input selection devices such as buttons, switching instruments and activating them was accomplished by mapping them to particular keys on the keyboard. To make it easier for the subjects, the author himself handled these keys during the evaluation sessions so that the subjects could maintain focused and maintain their hand positions. A default OpenGL lighting model was used in the first trial due to some performance issues with the projective spotlight and shadows. A number of lessons were apparent with the first trial:

Axes Reversed: The first was that the Y and Z axis were reversed from the subject's point of view. Since a vitrectomy is performed with the patient laying down, surgeons associate depth with vertical decent. This orientation confusion was especially noticeable in the practice exercises when the light probe did not produce full illumination at a vertical orientation. Surprisingly not all of the subjects complained about this situation. When asked later, one subject replied that he did notice that the axes were reversed but he could accommodate both systems.

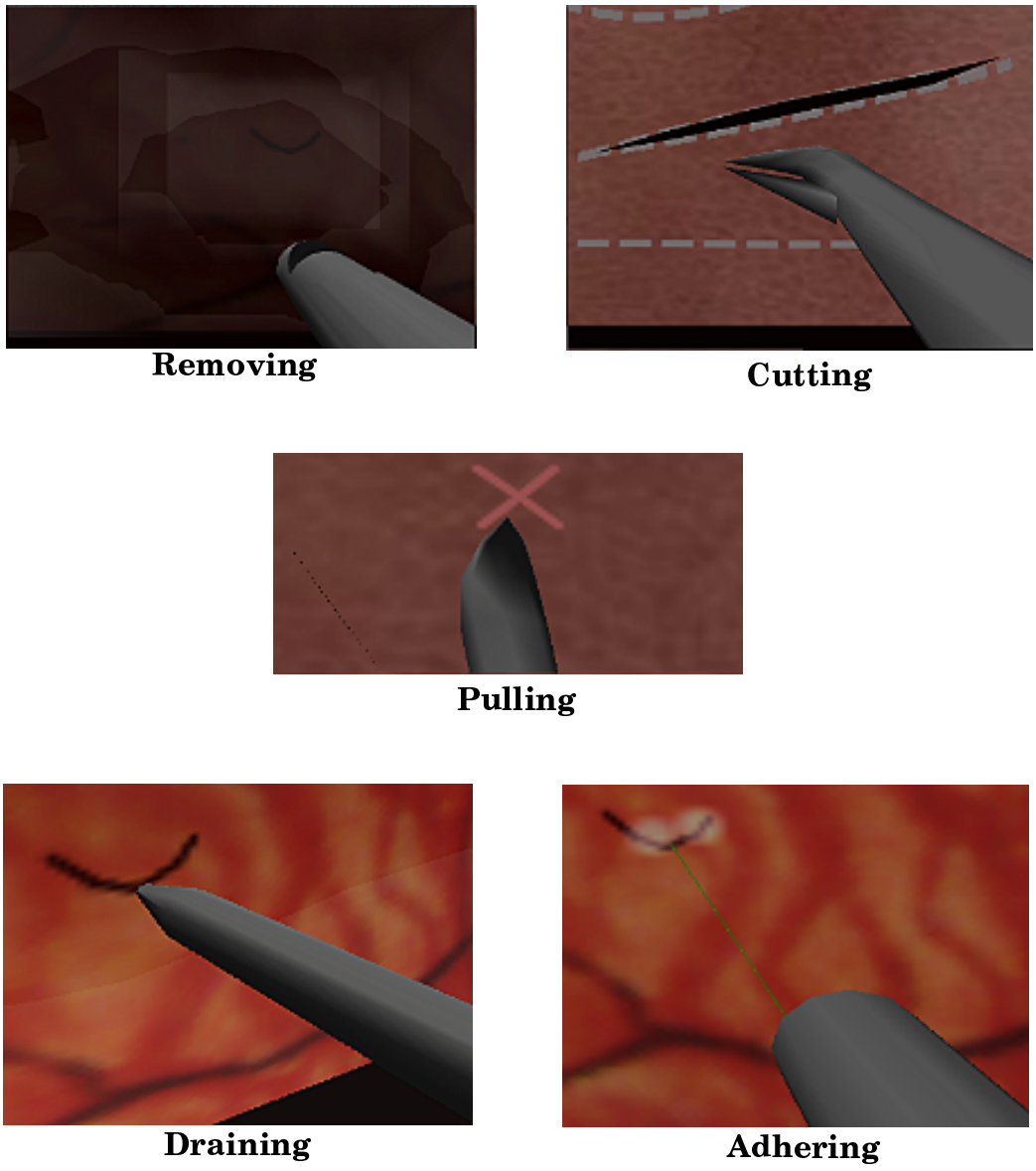
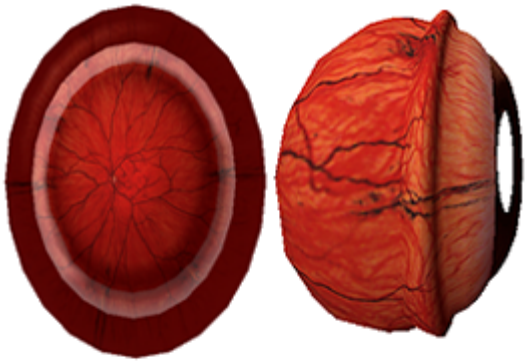
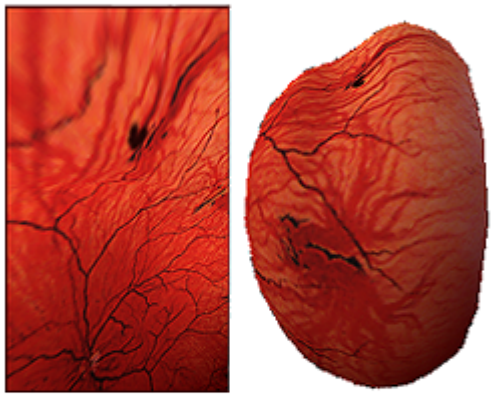


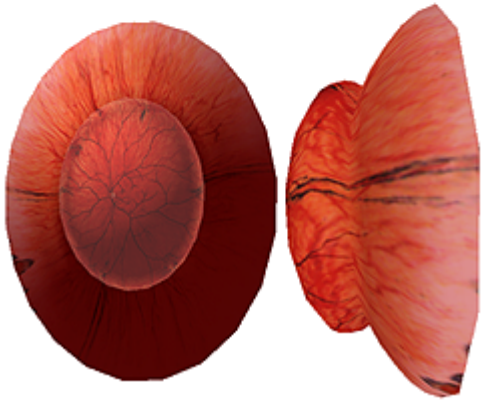
Figure 26. Practice exercises.



Case #1: Traction



Case #2: Tear



Case #3: Traction II

Figure 27. First set of simulated surgical cases.

Practice Unnecessary: The subjects did not understand the need or motivations for the the practice exercises. In fact, most of them were confused by the exercises. Some of the reasons were that the exercises were all exactly parallel to the screen which is usual in a vitrectomy. The skin texture on the cutting and pulling exercises was out of place. The reversed axes was glaring problem on the drainage exercise. The practice exercises also lengthened the testing period.

Hand Placement: There was also some confusion over hand placement. Most subjects recommended building a sphere that could represent the eye ball with two insertion holes to alleviate this dilemma. This is similar to the ideas developed in another vitrectomy simulator called EyeSi (Schill et al., 1999). Figure 29 shows a prototype of their system. Unfortunately, the miniBIRD tracking system lacks the required sensitivity to operate in the volume of the human eye.

Instrument Comments: General instrument comments ranged from replacing the pick with a forceps, wanting more immediate feedback from the cutting blade, reducing the removal hemisphere of the suction cutter, stabilizing the accuracy of the laser, and giving the needle a more blunt shape. The suction cutter instrument was used predominately and was generally well accepted. In addition, it was learned that the laser instrument can be used to adhere the entire peripheral retina after traction to prevent further detachments.

Replace First Case: In terms of the surgical cases, the first anteroposterior traction case was viewed as uncommon. A number of subjects suggested replacing it with the more common procedure of peeling a fibrovascular membrane.

Second Case - Tear: In the second case involving a tear, all of the subjects remarked that there was no trapped fluid. Upon further questioning, it was discover that the retinal texture map that was used for this purpose, was a combination of the retinal vessels plus the choroid pigment. To indicate detachment, two map were required so that the retinal vessels would be displaced from their default positions over the choroid. In addition, detached regions should have a slightly opaque-ish appearance.

General Comments: Another general comment was that view into the retinal chamber looked too deep to be realistic. An unsettling observation is that subjects quickly learned that there was no general collision detection so that they could go through any structure without consequences.

3.2 Second Trial

The goals of the second trial were to: 1) developed better surgical test cases, 2) emphasize the the other instruments besides the suction cutter, and 3) to shorten the overall testing time. The second trial was conducted roughly a month after the first trial. Figure 28 show the three surgical cases used for this trial. The following changes were made :

Practicing Dropped: The practice exercises were dropped since all of the subjects were familiar with the system.

Gel Sufficient: Since the vitreous gel models were ranked favorably, it was decided to exclude them from the second trial to improved the system performance and shift the focus to

the other instruments. In the first trial, removing the gel in the tear case took roughly fifteen minutes which was excessive in terms of the desired length of a session.

Improved Retina: The main improvement was that the retinal model was texture mapped with a new alpha map consisting of just the retinal vessels as shown in Figure 11. The choroid model was added underneath and was textured mapped with an opaque map without these vessels. A simple collision detection boundary sphere was fully debugged so that majority of the retinal model would not penetrate the choroid model. Again since the choroid model is not completely spherical, the retinal model can penetrate the choroid model posteriorly.

Tear Case: Two of the surgical cases were completely remodeled. The tear case contained a new retina model with a true three dimensional U-shaped tear in its anterior. This was accomplished by trimming the NURBS retinal model. A slight opaque-ish appearance was accomplished by drawing a gray region in the retinal alpha mask of its texture map. To improve performance, the quadrant of the retina without the detachment was not physically modeled.

Low Membrane Case: A new case was generated with a low fibrovascular membrane attached at the optic nerve and covering the macular region. Subjects had to peel the membrane and then cut or remove it at its attachment. No retinal detachment was simulated in this particular case.

Traction Case: The perpendicular traction case was simplified by just having one plane rather than the initial two as well the new retinal and choroid maps.

Minor Improvements: The miniBIRD system was reconfigured placing the Z axis in the vertical direction, shadowing was turned on, a zoom feature was implemented so that subjects could magnify their viewpoint onto the retina.

All of the instruments were updated:

Forceps: The pick instrument was replaced with a forceps model and was modified to only select the nearest vertex. Previously the pick instrument selected any vertex within a small radius at its tip. This change was prompted because the pick was often used to separate cut boundaries, but with its original behavior it would automatically select both sides.

Sideways Blade: A modified version of the cutting blade was constructed with a sideways blade that cuts in individual strokes. The sideways blade has a simpler interface because it only requires a small fixed cutting parallelogram at its tip. As a result, it lead to a more direct cutting experience.

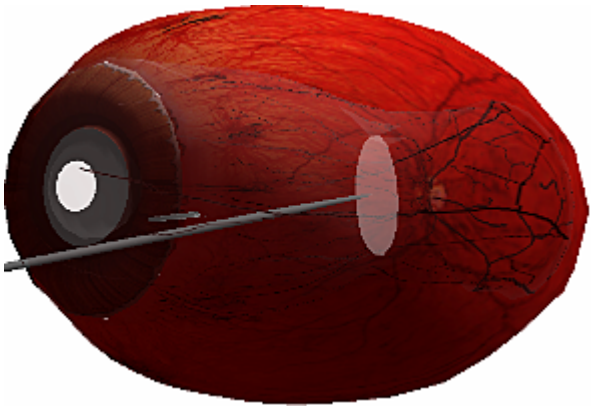
Laser: The laser instrument was modified to do a weighted average on its selected vertices based on the distance to its projection line.

Cutter: The suction cutter removal radius was shorten.

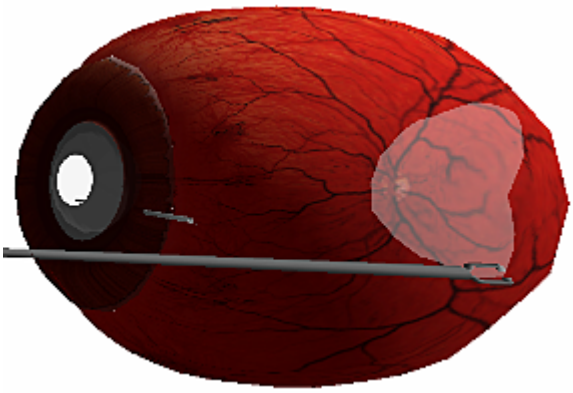
Comments from the subjects regarding the second trial were:

Tracking Problems Continue: Subjects continued to complain about the tracking. In the membrane case, subjects generally had to use both instruments together to pull on the

Case #1: Tear



Case #2: Traction



Case #3: Membrane

Figure 28. Second set of surgical cases.

membrane with the light probe's pick while cutting with the operative instrument. Often their hands would collide making this task awkward to accomplish.

Tear Correction: The newer surgical cases received a positive response due to the correct retinal and choroid texture maps. When asked if the detachments looked authentic, all of the subjects tested replied yes which was a substantial improvement over the first session. The tear case was well received, but the retina took longer than expected to return to its normal rest position after drainage. Also, the detached opaque-ish tint should have been reduced after drainage, but at this time the tint is hard-coded into the retinal texture map.

Subjects noted that the eyeball is generally rotated during drainage so that the optic nerve is at the lowest point. Rotation of the eyeball is done by gently pressing the instruments against the surrounding sclera. Rotations are also done to better insert the needle into retinal tears.

Membrane Case: Subjects remarked that the membrane should not return to its default position but stay roughly where it was left. A possible solution would be to add a viscosity force on the membrane to slow its return. Another comments was that some membranes have a weak attachment, meaning that when pulled they simply detach from the retina. Surgeons generally test a membrane by gently pulling on it before deciding to cut. All attachments in the simulator are strong due to the default instance vertex methodology.

Cutting and Pulling: One subject commented that the sideways blade should pull the tissue up like a forceps before cutting.

TABLE II

COOPER-HARPER RANKING

	Subject 1	Subject 2	Subject 3	Subject 4
Trial 1	5.5	8	8.5	8
Trial 2	5.0	8	5.5	7.5
+/- Change	-0.5	0.0	-3.0	-0.5

Summarizes the overall Cooper-Harper rating based on the survey questions. Refer to Table III which shows the general Cooper-Harper scale from Figure 9.

Laser Another subject remarked that the laser projects a circle on the retina rather than a full projection line as currently implemented.

General Comments: All of the tested subjects felt the projective spotlight and shadows were a significant improvement.

3.3 Results

Below are the summary tables collated from each testing session.

TABLE III

GENERAL COOPER-HARPER SCALE

Rating	Difficulty
1	Very Easy, Highly Desireable
2	Easy, Desireable
3	Fair, Mild Difficulty
4	Minor but Annoying Difficulty
5	Moderately Objectionable Difficulty
6	Very Objectionable but Tolerable Difficulty
7	Major Difficulty
8	Major Difficulty
9	Major Difficulty
10	Impossible

Summarizes the general Cooper-Harper scale. Please refer to Figure 9 for a complete diagram.

TABLE IV

ERROR RATING

	Trial	Subject 1	Subject 2	Subject 3	Subject 4
Overall	1	Low	Medium	Somewhat High	Medium
	2	Low	Medium	Low	Somewhat Low
Image Quality	1	Low	Medium	Low	Medium
	2	Low	Somewhat Low	Low	Somewhat Low
Modeling Detail	1	Low	Medium	Somewhat Medium	Medium
	2	Low	Somewhat Low	Low	Somewhat Low
Tracking Accuracy	1	Low	Somewhat High	Medium	Somewhat High
	2	Low	Somewhat High	Low	Medium
Physical Interface	1	Low	Medium	Medium	Somewhat High
	2	Low	Medium	Low	Somewhat Low
Instru-mentation	1	Low	Medium	Medium	Medium
	2	Low	Medium	Low	Somewhat Low

Table IV shows the error level assigned by all subjects for both trials.

TABLE V

WORKLOAD RATING

		Tracking	Physical Interface	Instrumentation
Subject 1	Trial 1	Somewhat High	Somewhat Medium	Low
Subject 1	Trial 2	Somewhat High	Low	Low
Subject 3	Trial 2	High	Medium	Low

Table V estimates the workload only for those subjects who felt the simulator's error level was sufficiently low.

TABLE VI

INSTRUMENT RATING

	Trial	Subject 1	Subject 2	Subject 3	Subject 4
Pick	1	Fair	N/A	N/A	Fair
	2	Very Good	Very Good	Good	Very Good
Blade	1	Very Good	N/A	N/A	Fair
	2	Very Good	Good	Good	Very Good
Cutter	1	Very Good	Good	Very Good	Very Good
	2	Very Good	Very Good	Excellent	Very Good
Laser	1	Very Good	N/A	N/A	Good
	2	Very Good	N/A	Good	Fair
Needle	1	Very Good	N/A	Good	Good
	2	Very Good	Very Good	Excellent	Very Good

Table VI summarizes rating for each instrument.

CHAPTER 4

DISCUSSION

Given that the simulator was a first prototype, it was not a surprising that it received a higher Cooper-Harper rating indicating that improvements were required. The simulator's ranking did improve between trials. A number of factors have contributed to this negative perception including tracking problems, simplistic physical interface, instrument inconsistencies, and lack of realism.

Tracking was the most common complaint throughout the trials. The miniBIRD system did not have the required precision expected by the surgeons. Retinal microsurgery often demands "surgical accuracy on the order of tens of microns, a scale at or near the limit of human positional ability" (Gupta et al., 1999). The miniBIRD system has the static positional resolution of 0.02 of an inch. In addition, the miniBIRD system was scaled up six times to try and increase its range within a smaller area without letting the noise level degrade its overall signal. This adaptation was insufficient and all subjects had to over extend their hand moments to compensate. The main advantage of the miniBIRD system is that it could be integrated immediately into the simulator since its programming interface was already part of the CAVE library. Other tracking systems would require that their interfaces to be re-encoded before any development could begin. It seems doubtful that any currently available 3D tracking system can accurately record vitreoretinal maneuvers.

Instruments in general received an average error rating, probably to due to geometry dependent artifacts in their design. The pick selected the nearest vertex which sometimes was not exactly where the tip was located. Fracturing was highly dependent on the original polygonal structure, and most subjects noted that fractures should be closer to the contact point. The laser suffered the worst artifacts since it is highly vertex dependent. A general ray tracing intersection routine would probably be needed to compute an exact texture map coordinate for the adhesion circle rather than averaging nearby vertices. The drainage needle functioned simplistically without any real fluid mechanism simulation. The suction cutter was generally well received overall.

Cutting did lag behind the user's current position since the system was trying to maximize the current cutting plane. A parameter can be adjusted for finer resolution, but a tolerance should be retained for the noise level inherit in 3D tracking systems. Some polygonal artifacts did occur since the system rounds to the nearest vertex to avoid generating small springs. Small springs however are generated in particular cases and are flagged as inactive, since in general they are unstable under the used integration method. Cut boundaries naturally separate because the objects themselves are put under tension by reducing their spring rest vectors by a small percentage. If a parallelogram is completely contained within a triangle and doesn't intersect any edge, the algorithm does not process it at this time. The sideways blade gave a more direct response since it has a fixed cutting plane.

Image quality and modeling were ranked lower in terms of error. Designing surgical cases is still tedious. More complex 3D fibrovascular tissues were desired, but modeling structures in

three dimensions and maintaining alignment with the retinal vertices were problematic. The membrane tissue is the most complex structure. This model is actually a trimmed copy of the macular region of the retinal surface which had been flatten.

Another related problem is with compressibility the current physics system. Some fibrovascular tissue can shrink significantly and pull the retina surface down to a very small circumference. One approach to model this shrinkage is to scale down the spring rest vectors and spring stiffness coefficients by a large percentage. However, this method did not achieve large compressions as expected. The vitreous gel is also highly compressible, but within these simulations it seemed to reinforce the retina, probably because the gel model was attached to the retinal model and because there was no internal collision detection.

In terms of training potential, the overall vitrectomy procedure could be accomplished, and functionality of the surgical instruments was judged sufficient. However, the lack of collision detection permitted negative training since subjects learned that they could move instruments through any surface without repercussions. For example in the planar perpendicular traction case, some subjects placed the suction cutter through the fibrovascular tissue model before removing it. In reality, an incision is required before inserting the cutter because a collision may induce retinal tears. Given this deficiency, practical surgical cases should minimize instrument penetration by designing pathologies in the anterior section of the vitreous chamber. Good examples are the membrane case and retinal tear model used in the second trial.

Even though the anteroposterior traction case used in the first trial was viewed as uncommon, simulators can expose fellows to rarer cases that they will eventually encounter in their practice.

4.0.1 Conclusions

A prototype virtual reality vitrectomy simulator was assembled with two types of retinal detachments. Eight different surgical instruments were implemented. A near real-time physically based cutting algorithm was accomplished on 3D polygonal models with a mass-spring system.

Although the majority of subjects felt that improvements to the simulator were necessary, all indicated that the surgical procedure could be accomplished with relatively low error rate. To minimize negative training, surgical cases should be designed with a nonobstructed anterior pathology. All subjects seemed genuinely enthusiastic about the concept of a simulator indicating that with refinement and better tracking, the simulator could be a useful addition to the fellows' training program.

4.0.2 Future Work

Research in this project involves three general areas: virtual reality simulation of a pathology within the human body, duplication of surgical instruments' functionality, and the physical reproduction an operating room. In terms of simulating retinal detachment, a future goal is to develop a more physically driven simulation that would include a comprehensive fluid mechanics component to replicate liquid vitreous and fluid-gas exchanges. A 3D Navier-Stokes equation is most often used to represent fluid fields, but its computational requirements are generally

not within interactive frame rates. A more recent algorithm based on modified equations has demonstrated near real-time fluid flow (Stam, 1999). The small contained environment of vitreous chamber may provide some simplification, but trapped subretinal fluid has exhibited non-Newtonian pseudoplastic behavior at low flow rates (Michels et al., 1990).

Collision detection between all deformable structures (including self intersections) and surgical instruments would require a general N-body processing algorithm such as I-COLLIDE (Cohen et al., 1995) published by the University of North Carolina at Chapel Hill. A N-body algorithm compares each surface triangle to every other possible triangle in the scene. A cylindrical boundary may be sufficient for a quick intersection test for instruments. With more computational resources, some instrument such as the cutting blade should be active at all times, and not turned on when ready. Moving parts on instruments would also improve the simulated surgical experience.

A challenging goal for the simulation of surgical instruments would be geometry independence. Surgeons do not want to know about the internal structure of a simulator. They want to see the proper result happen when they activate an instrument. Many of the instruments in this simulator suffered from some underlying modeling dependence. The cutting path was approximated by spring intersections. The laser's adhesion circle was positioned by nearby vertices. The suction cutter clipped vertices. All in some way exposing the underlying geometry.

In terms of modeling, the surface representation of the vitreous gel is simplistic. A new volumetric approach called an Enhanced 3D ChainMail algorithm could represent the gel as a whole within interactive rates (Schill et al., 1998). This algorithm also has the advantage that

it can represent inhomogeneous material. More investigation would be required to see if this algorithm can be optimized while voxels are being interactively removed by the suction cutter, or compressed volume-wise.

The most deficient area of the simulator was in its physical interface. This is due to the prohibitive cost to acquire the appropriate surgical equipment and then interfacing it to a common computer system. Ideally, the physical interface should consist of the same operating room equipment as shown in Figure 4. A stereomicroscope, foot petals, instrument trays, and control systems should be included in an ideal interface. An example of a better interface was constructed for the recent EyeSi project that consists of a suspended mechanical eye with two insertion holes for instruments (Schill et al., 1999). Figure 29 shows a photograph of the EyeSi's prototype interface. All of the subjects suggested building a physical eyeball representation to help orient themselves.

If more funding was available, it would be useful to expand the scope of the project to include the surgical preparatory and closing steps so that fellows could be trained in the entire surgical procedure. It would involve additional instruments such as suturing as well as additional physical models such as the eye lids, sclera and additional membranes.

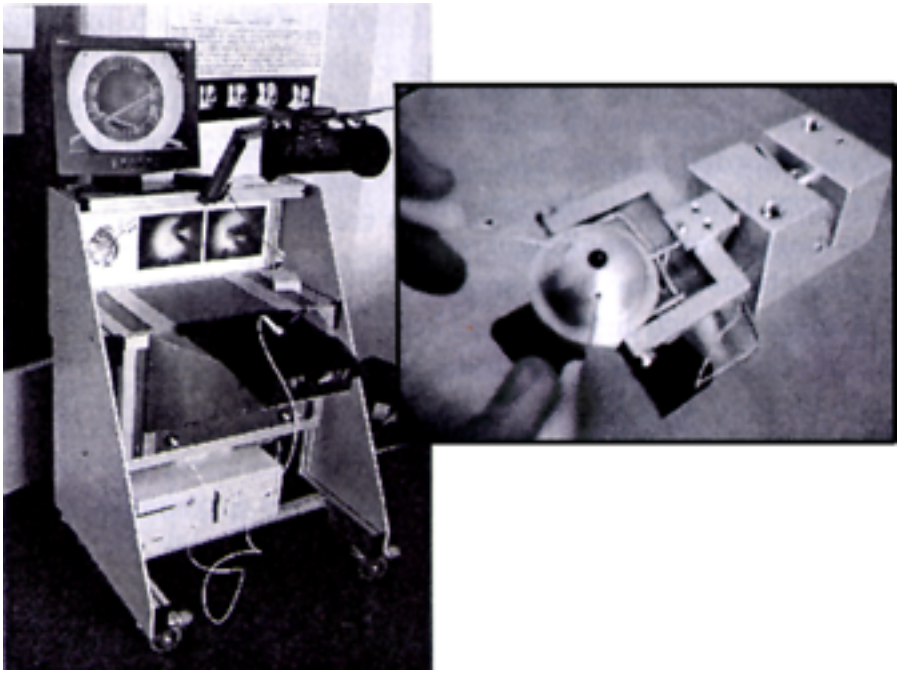


Figure 29. EyeSi prototype physical interface.

APPENDICES

CHAPTER 5

APPENDIX A

5.1 File Format

The sections below outline the internal file format used to store scene objects. A '#' represents a numerical place holder while '...' means that the last entity can be replicated multiple times.

5.1.1 Static Triangle Mesh

```
Tmesh {
    Rows Cols = { # # }
    Flags = { Backface Active WrapAround }

    # Vertices {
      Vertex # {
          position = ( # # # )
          normal   = < # # # >
          texture  = [ # # # ]
      }
      ...
    }
}
```

5.1.2 Static Triangle Set

```
Tset {
     Flags = { Backface Active }

     # Vertices {
       Vertex # {
           position = ( # # # )
           normal   = < # # # >
           texture  = [ # # # ]
       }
```

```
      ...
    }
    # Triangles {
        ( # # # )
        ( # # # )
        ...
    }
}
```

5.1.3 Physical Triangle Set

```
PhysicalTSet {
    Space: # Vertices, # Triangles, # Springs
    Forces = { Gravity ... }
    Total Mass = #
    Total Surface Area = #
    Spring Global Scale: Length # Stiffness #
    Stretch Factor = #%
    Instruments = { Pick Cutter Blade Laser }
    Flags = { Backface Active }

    Vertices {
        Vertex # {
            position = ( # # # )
            normal   = < # # # >
            texture  = [ # # # ]
            velocity       = < # # # >
            acceleration   = < # # # >
            mass           = #
            Flags    = { Fixed Inst Active }
            Forces   = { Fluid ...}
            # Triangles [ # # # ... ]
            # Springs [ # # # ... ]
        }
        ...
    }

    Triangles {
        Triangle # {
            Vertices ( # # # )
            Springs  [ # # # ]
        }
```

```
        ...
    }

    Springs {
        Spring # {
            vertices  = { # # }
            rest direction = < # # # >
            current direction = < # # # >
            strength damping = { # # }
            scale = #
            Flags = { Active }
        }
        ...
    }

    # Instances {
        # has # instance { # # # ... }
        ...
    }
}
```

CHAPTER 6

APPENDIX B

6.1 Physical Field Components

Below is the class member fields for all the physically based structures. Static members were primarily used for global lists such as the active linked list, the delete/recycle linked list, and marked linked list which was used for entities which had to updated later. Fields with an array of two were double buffered so that the computation process could compute a new value while the drawing processes were rendering the previously value.

6.1.1 Vertex

```
class PhysicalVertex {
    static PhysicalVertex *VertexList, *DeleteList, *MarkList, *MarkEnd;
    static unsigned int    MaxVertices, free_index, num_Deleted, MaxForces;

    int                    ID;
    bool                   Active, Fixed, Inst, Split, NewVertex, Deleted, Torn;
    Point                  position[2];
    Vector                 normal[2], velocity, acceleration, force;
    GLfloat                texture[3];
    double                 mass;

    unsigned int           num_triangles, max_triangles, num_springs,
                           max_springs, num_forces, num_insts;
    PhysicalTriangle     **triangle;
    VectorSpring         **spring;
    ForceGenerator        *LocalForce[5];
    PhysicalVertex       **inst_vertex;
    PhysicalVertex        *master_vertex;
    PhysicalTSet          *tset;
    PhysicalVertex        *next, *prev, *nextMarked;
};
```

6.1.2 Triangle

```
class PhysicalTriangle {
    static PhysicalTriangle *TriangleList, *DeleteList, *MarkList, *MarkEnd;
    static unsigned int      MaxTriangles, free_index, num_Deleted;

    int                      ID;
    PlaneSign                Sign;
    bool                     Active[2];   // doublecheck Active flag
    PhysicalVertex          *vertex[4];
    VectorSpring            *spring[4];
    unsigned char            num_vertices, num_springs;
    double                   area;
    PhysicalTSet            *tset;
    PhysicalTriangle        *next, *prev, *nextMarked, *childTriangle[4];
};
```

6.1.3 Spring

```
class VectorSpring {
    static VectorSpring  *SpringList, *DeleteList, *MarkList, *MarkEnd;
    static unsigned int   MaxSprings, free_index, num_Deleted;
    static double         Epsilon;

    int                   ID;
    bool                  Active, InPlane, Integrate, Torn;
    IntType               intersection_type;
    PhysicalVertex       *pnt1, *pnt2, *replace_pnt1, *replace_pnt2;
    int                   num_triangles;
    PhysicalTriangle     *triangle[4];
    Vector                rest_direction, current_direction;
    double                strength, dampening, rest_length, epsilon_offset;
    Vector                force;
    VectorSpring         *subspringA, *subspringB;
    double                intersection_t, intersection_u, intersection_v;
    VectorSpring         *next, *prev, *nextMarked;
};
```

6.1.4 Triangle set

```
class PhysicalTSet : public Object {
    static PhysicalTSet    **Set;
    static unsigned int     *NumSets, MaxForces;

    PhysicalVertex          *VertexTable;
    VectorSpring            *SpringTable;
    PhysicalTriangle        *TriangleTable;
    double                   total_mass, total_area, stretch_factor;
    unsigned int             num_forces;
    ForceGenerator          *GlobalForce[10];
};
```

CHAPTER 7

APPENDIX C

7.1 Evaluation Survey

Virtual Reality Vitrectomy Simulator
Evaluation Survey
Session #_____

Subject Number_____ Date: _____

Thank you for participating in our vitrectomy simulator project. Please take a moment and fill out this short survey indicating your impressions on the training potential of this simulator for other Ophthalmology Fellows.

Even though errors may be large or frequent, can the vitrectomy procedure be accomplished most of the time?

NO YES
Skip to Page 5

Are errors small and inconsequential?

NO YES
Skip to Page 2

Is mental workload level acceptable?

NO YES
Skip to Page 3 Skip to Page 4

Please circle the description below which best describes the amount of error in the simulator :

HIGH Somewhat HIGH MEDIUM Somewhat MEDIUM Low

For each category below, please circle its error amount :

Image Quality	HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low
Modeling Detail	HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low
Tracking Accuracy	HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low
Physical Interface	HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low
Instrumentation	HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low
Other: _____ _____	HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low

General comments about the error level in the simulator :

Proceed to Page 5 ->

Page 2

Please circle the description below which best describes the amount of mental workload you experienced in the simulator :

HIGH	Somewhat HIGH	MEDIUM	Somewhat MEDIUM	Low

For each category below, please rate its mental workload :

Tracking	HIGH	Somewhat HIGH	MEDIUM	Somewhat	Low MEDIUM
Physical Interface	HIGH	Somewhat HIGH	MEDIUM	Somewhat	Low MEDIUM
Instrumentation	HIGH	Somewhat HIGH	MEDIUM	Somewhat	Low MEDIUM
Other: _____	HIGH	Somewhat HIGH	MEDIUM	Somewhat	Low MEDIUM

General comments about the workload level in the simulator :

Proceed to Page 5 ->

Please circle the description below which best describes the amount of mental workload you experienced in the simulator :

VERY EASY Somewhat EASY EASY Somewhat FAIR FAIR

For each category below, please rate its mental workload :

Tracking	VERY EASY	Somewhat EASY	EASY	Somewhat FAIR	FAIR
Physical Interface	VERY EASY	Somewhat EASY	EASY	Somewhat FAIR	FAIR
Instrumentation	VERY EASY	Somewhat EASY	EASY	Somewhat FAIR	FAIR
Other: _____	VERY EASY	Somewhat EASY	EASY	Somewhat FAIR	FAIR

General comments about the workload level in the simulator :

Proceed to Page 5 ->

Page 4

For each surgical instrument, please rate its performance based upon its simulated functionality:

Pick Excellent Comments:
Very Good _____
Good _____
Fair _____
Poor _____

Blade Excellent Comments:
Very Good _____
Good _____
Fair _____
Poor _____

Cutter Excellent Comments:
Very Good _____
Good _____
Fair _____
Poor _____

Laser Excellent Comments:
Very Good _____
Good _____
Fair _____
Poor _____

Needle Excellent Comments:
Very Good _____
Good _____
Fair _____
Poor _____

Continue on next Page ->

Please list anything below that you felt should have been included in the simulation:

Please list any other general comments here:

THANK YOU!

CITED LITERATURE

Astley, R. J.: Finite Elements in Solids and Structures. Chapman & Hall, 1992.

Barr, A.: Global and local deformations of solid primitives. In Computer Graphics Proceedings, pages 21–30. SIGGRAPH, 1984.

Bell, F. C. and Stenstrom, W. J.: Atlas of the Peripheral Retina. W.B. Saunders Company, 1983.

Blaiwes, A. S., Plug, J. A., and Reagan, J. J.: Transfer of training and the measurement of training effectiveness. Human Factors, 15(6):523–533, 1973.

Bro-Neilsen, M.: Surgery simulation using fast finite elements. In Visualization in Biomedical Computing, eds. K. H. Hohn and R. Kikinis, Lecture Notes in Computer Science #1131, pages 529–533. Springer-Verlag, September 1996.

Chen, D. T. and Zeltzer, D.: Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. In Computer Graphics Proceedings, pages 89–98. SIGGRAPH, 1992.

Chen, J. X., da Vitoria Lobo, N., Hughes, C. E., and Moshell, J. M.: Real-time fluid simulation in a dynamic virtual environment. IEEE Computer Graphics and Applications, 17(3):52–61, May-June 1997.

Clark, J. H.: . Doctoral dissertation, University of Utah, 1974.

Cohen, J., Lin, M., Manocha, D., and Ponamgi, K.: I-collide: An interactive and exact collision detection system for large-scaled environments. In Interactive 3D Graphics Conference Proceedings, pages 189–196. ACM, 1995.

Cotin, S., Delingette, H., and Ayache, N.: Real time volumetric deformable models for surgery simulation. In Visualization in Biomedical Computing, eds. K. H. Hohn and R. Kikinis, Lecture Notes in Computer Science #1131, pages 534–540. Springer-Verlag, September 1996.

Cover, S. A., Ezquerra, N. F., O'Brien, J. F., Rowe, R., Gadacz, T., and Palm, E.: Interactively deformable models for surgery simulation. IEEE Computer Graphics and Applications, 13(6):68–75, November 1993.

Cruz-Neira, C., Sandin, D. J., and DeFanti, T. A.: Surround-screen projection-based virtual reality: The design and implementation of the cave. In Computer Graphics Proceedings, pages 135–142. SIGGRAPH, 1993.

Delp, S. L., Loan, P., Basdogan, C., and Rosen, J. M.: Surgical simulation: An emerging technology for training in emergency medicine. Presence, 6(2):147–159, April 1997.

eds. N. I. Durlach and A. S. Mavor Virtual Reality Scientific and Technological Challanges. Washington D.C., National Academy Press, 1995.

Emmett, A.: Virtual reality helps steady the gait of parkinson patients. Computer Graphics World, pages 17–18, November 1994.

Fatt, I. and weissman, B. A.: Physiology of the Eye. Butterworth-Heinemann, 1992.

Gelder, A. V. and Wilhelms, J.: Varying spring constants for accurate simulation of elastic material. In Computer Graphics Visual Proceedings, page 153. SIGGRAPH, 1997.

Glassner, A. S.: Useful 3d geometry. In Graphic Gems I, ed. A. S. Glassner, pages 297–300. London, AP Professional, 1990.

Gould, H. and Tobochnik, J.: An Introduction to Computer Simulation Methods. Addison-Wesley, 1988.

Gupta, P. K., Jensen, P. S., and de Juan Jr, E.: Surgical forces and tacile perception during retinal microsurgery. In Medical Image Computing and Computer-Assisted Intervention - MICCAI, Lecture Notes in Computer Science #1679. Springer-Verlag, 1999.

Haines, E.: Essential ray tracing algorithms. In An Introduction to Ray Tracing, ed. A. S. Glassner, pages 59–64. London, Academy Press Limited, 1989.

Higgins, G. A., Merrill, G. L., Hettinger, L. J., Kaufmann, C. R., Champion, H. R., and Satava, R. M.: New simulation technologies for surgical training and certification: Current status and future projections. Presence, 6(2):160–172, April 1997.

Hodges, L. F., Kooper, R., Meyer, T. C., Rothbaum, B. O., Opdyke, D., de Graaff, J. J., Williford, J. S., and North, M. M.: Virtual environments for treating the fear of heights. Computer, 28(7):27–33, July 1995.

Hodges, L. F., Watson, B. A., Kessler, G. D., Rothbaum, B. O., and Opdyke, D.: Virtually conquering fear of flying. IEEE Computer Graphics and Applications, 16(6):42–49, November 1996.

Hohne, K. H., Pflesser, B., Pommert, A., Riemer, M., Schiemann, T., Schubert, R., and Tiede, U.: A 'virtual body' model for surgical education and rehearsal. Computer, 29(1):25–31, January 1996.

Hunter, I. W., Doukoglou, T. D., Lafontaine, S. R., Charette, P. G., Jones, L. A., Sagar, M. A., Mallinson, G. D., and Hunter, P. J.: A teleoperated microsurgical robot and associated virtual environment for eye surgery. Presence, 2(4):265–280, Fall 1993.

James, D. L. and Pai, D. K.: Accurate real time deformable objects. In Computer Graphics Proceedings, pages 65–72. SIGGRAPH, 1999.

Kaltenborn, K. and Rienhoff, O.: Virtual reality in medicine. Methods in Information in Medicine, 32(5):407–417, 1993.

Koch, R. M., Gross, M. H., Carls, F. R., Buren, D. F. V., Fankhauser, G., and Parish, Y. I. H.: Simulating facial surgery using finite element models. In Computer Graphics Proceedings, pages 421–428. SIGGRAPH, 1996.

Lewis, H. and Ryan, S. J.: Medical and Surgical Retina. Mosby-Year Book, 1994.

Long, M. E. and McNally, J.: The sense of sight. National Geographic, pages 3–41, November 1992.

Lytle, W.: Simulated treatment of an ocular tumor. ACM SIGGRAPH Video Review, July 1989.

Mahoney, D. P.: Virtual science. Computer Graphics World, pages 20–26, July 1994.

Mazura, A. and Seifert, S.: Virtual cutting in medical data. In Transformation of Medicine Through Communication, ed. K. Morgan, volume 39, pages 420–429. Medicine Meets Virtual Reality Conference, IOS Press, January 1997.

Merril, J. R., Roy, R., and Raju, R.: Virtual reality for trade shows and individual physician training. Virtual Reality Systems, 1(3):40–44, Spring 1994.

Metaxas, D. and Terzopoulos, D.: Dynamic deformation of solid primitives with constraints. In Computer Graphics Proceedings, pages 309–312. SIGGRAPH, 1992.

Michels, R. G., Wilkinson, C. P., and Rice, T. A.: Retinal Detachment. The C.V. Mosby Company, 1990.

Muthupillai, R., Rossman, P. J., Lomas, D. J., and Greenleaf, J. F.: Magnetic resonance imaging of transverse acoustic strain waves. Magnetic Resonance in Medicine, (36):266–274, 1996.

Neumann, P., Sadler, L., and Gieser, J.: Virtual reality vitrectomy simulator. In Medical Image Computing and Computer-Assisted Intervention - MICCAI, eds. W. M. Wells, A. Colchester, and S. Delp, Lecture Notes in Computer Science #1496. Springer-Verlag, 1998.

O'Donnell, R. D. and Eggmeier, F. T.: Workload assessment methodology. In Handbook of Perception and Human Performance: Sensation and Perception, eds. K. R. Boff, L. Kaufman, and J. Thomas, chapter 42. John Wiley & Sons, 1986.

Ota, D., Loftin, B., Saito, T., Lea, R., and Keller, J.: Virtual reality in surgical education. Computers in Biology and Medicine, 25(2):127–137, 1995.

Pape, D.: A hardware-independent virtual reality development system. IEEE Computer Graphics and Applications, 16(4):44–47, July 1996.

Parent, R.: Computer animation: Algorithms and techniques. http://www.cis.ohio-state.edu/ parent/book/preface.html, August 1996. class animation notes.

Parshall, R. F.: Computer-aided geometric modeling of the human eye and orbit. Journal of Biomedical Communication, 18(2):32–39, 1991.

Pieper, S., Rosen, J., and Zeltzer, D.: Interactive graphics for plastic surgery: A task-level analysis and implementation. In ACM Proceedings Interactive 3D Graphics, volume 3, pages 127–134, 1992.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P.: Numerical Recipes in C. New York, New York, Cambridge University Press, 1996.

Riess, T. and Weghorst, S.: Augmented reality in the treatment of parkinson's disease. In Interactive Technology and the new Paradigm for Healthcare, eds. R. M. Satava, K. Morgan, H. B. Sieburg, R. Mattheus, and J. P. Christensen, volume 18, pages 298–302. Medicine Meets Virtual Reality, IOS Press, January 1995.

Rolfe, J. M. and Staples, K. J.: Flight Simulation. Cambridge University Press, 1986.

Sagar, M. A., Bullivant, D., Mallinson, G. D., Hunter, P. J., and Hunter, I. W.: A virtual environment and model of the eye for surgical simulation. In Computer Graphics Proceedings, pages 205–212. SIGGRAPH, 1994.

Satava, R. M.: The modern medical battlefield. Technical report, Advanced Research Projects Agency, Arlington, Virginia 22203, August 1993.

Satava, R. M.: Virtual reality surgical simulator: The first steps. Surgical Endoscopy, 7:203–205, 1993.

Schill, M. A., Gibson, S. F., Bender, H. J., and Manner, R.: Biomechanical simulation of the vitreous humor in the eye using an enhanced chainmail algorithm. In Medical Image Computing and Computer-Assisted Intervention - MICCAI, eds. W. M. Wells, A. Colchester, and S. Delp, Lecture Notes in Computer Science #1496. Springer-Verlag, 1998.

Schill, M. A., Wagner, C., Hennen, M., Bender, H.-J., and Manner, R.: Eyesi - a simulator for intra-ocular surgery. In Medical Image Computing and Computer-Assisted Intervention - MICCAI, Lecture Notes in Computer Science #1679. Springer-Verlag, 1999.

Sederberg, T. and Parry, S.: Free-form deformation deformation: A sculpturing tool for 3d geometric modeling. In Computer Graphics Proceedings, pages 151–160. SIGGRAPH, 1986.

Segal, M., Korobkin, C., van Widenfelt, R., Foran, J., and Haeberli, P.: Fast shadows and lighting effects using texture mapping. In Computer Graphics Proceedings, pages 249–252. SIGGRAPH, 1992.

Shene, C.-K.: Computing the intersection of a line and a cylinder. In Graphic Gems IV, ed. P. S. Heckbert, pages 353–355. London, AP Professional, 1994.

Sinclair, M. J., Peifer, J., and Haleblian, R.: Computer-simulated eye surgery. The Journal of the American Academy of Ophthalmolgy, 102(3):517–521, March 1995.

Song, G.-J. and Reddy, N. P.: Tissue cutting in virtual environments. In Interactive Technology and the new Paradigm for Healthcare, eds. R. M. Satava, K. Morgan, H. B. Sieburg, R. Mattheus, and J. P. Christensen, volume 18, pages 359–364. Medicine Meets Virtual Reality, IOS Press, January 1995.

Stam, J.: Stable fluids. In Computer Graphics Proceedings, pages 121–128. SIGGRAPH, 1999.

Sutherland, I. E.: The ultimate machine. In Proceedings of IFIP, volume 2, pages 506–508,582–583, 1965.

Szeliski, R. and Tonnesen, D.: Surface modeling with oriented particle systems. In Computer Graphics Proceedings, pages 185–194. SIGGRAPH, 1992.

Terzopoulos, D. and Fleischer, K.: Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In Computer Graphics Proceedings, pages 269–278. SIGGRAPH, 1988.

Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K.: Elastically deformable models. In Computer Graphics Proceedings, pages 205–214. SIGGRAPH, 1987.

Treviranus, J.: Virtual reality technologies and people with disabilities. Presence, 3(3):201–207, Summer 1994.

Wierwille, W. W. and Casali, J. G.: A validated rating scale for global mental workload measurement applications. In Proceedings of the Human Factors Society Twenty-Seventh Annual Meeting, pages 129–133, 1983.

Wightman, D. C. and Lintern, G.: Part-task training for tracking and manual control. Human Factors, 27(3):267–283, 1985.

Wilhelms, J. and Gelder, A. V.: Anatomically based modeling. In Computer Graphics Proceedings, pages 173–180. SIGGRAPH, 1997.

Witkin, A., Baraff, D., Blum, M., Thumrugoti, U., and Monheit, G.: Physically based modeling: Principles and practice. SIGGRAPH Course Note #19, August 1997.

Witkin, A., Baraff, D., and Kass, M.: An introduction to physically based modeling. SIGGRAPH Course Note #32, July 1994.

Ziegler, R., Mueller, W., Fischer, G., and Goebel, D. M.: A virtual reality medical training system. In ComputerVision, Virtual Reality and Robotics in Medicine, ed. N. Ayache, Lecture Notes in Computer Science #905, pages 282–286. Springer-Verlag, April 1995.

VITA

NAME: Paul Francis Neumann

EDUCATION: B.S., Engineering, University of Illinois at Chicago, Chicago, Illinois, 1986

M.S., Engineering, University of Illinois at Chicago, Chicago, Illinois, 1992

Ph.D., Engineering, University of Illinois at Chicago, Chicago, Illinois, 2000

PUBLICATIONS: P.Neumann: Near Real-Time Cutting. SIGGRAPH Visual Proceedings, Technical Sketch, 2000.

P.Neumann, L.Sadler and J.Gieser: Virtual Reality Vitrectomy Simulator. Medical Image Computing and Computer-Assisted Intervention (MICCAI Proceedings), Lecture Notes in Computer Science #1496, Springer-Verlag, 1998.

R.Gore: Neandertals. National Geographic, January 1996. (Images and Photograph)

M.Rasmussen, B.Blumenthal, S.Borysewicz, and
P.Neumann: The Global Interactive Museum Using Mosaic.
2nd International World Wide Web Conference, 1994.

P.Neumann and L.Sadler: Visualization Package for Laser Scanned 3D Geometry. SPIE Medical Imaging VI, Image Capture, Formatting and Display Proceedings, 1993.

GRANTS: A.Steinberg and P.Neumann, UIC Council for Excellence in Teaching and Learning $10,000 grant to develop a dental training simulator. (January 2000)

MEMBERSHIP: ACM, SIGGRAPH

Virtual Reality Vitrectomy Simulator

Paul F. Neumann, Ph.D.
Department of Electrical Engineering and Computer Science
University of Illinois at Chicago
Chicago, Illinois (2000)

With the advent of Virtual Reality, surgical simulators have been proposed as a supplement to the education and training of medical residents and post doctoral fellows. A virtual reality simulator can convey the three dimensional anatomical space, provide a means to fine tune surgical skills, and present a variety of pathologies ranging from rare to general cases. Surgical simulators consist of anatomical models, a physics system, a pathology simulation, virtual surgical instruments, and a physical interface. In this study, a new virtual reality system will physically simulate two types of retinal detachment in the human eye: rhegmatogenous detachment and proliferative traction. In addition, a common set of viteoretinal instruments has also been implemented.

To accomplish this task, a three dimensional computer eye model was constructed and coupled with a mass-spring system for elastic deformations. Actual surgical instruments were tracked as part of the simulator's physical interface. Rhegmatogenous detachment was physically simulated with an unique force function, while traction was accomplished by reducing the natural rest length of the surface springs. Eight surgical instruments were implemented including: a light probe coupled with a pick, forceps for grasping and elevation, two types of blades for cutting, suction cutter for removal, laser for adhesion, and a drainage needle. The forceps and pick instruments lock the nearest vertex to its position and induce tearing if springs exceed a threshold. A unique 3D surface cutting algorithm subdivides springs and triangles

based on the cutting path interactively. The suction cutter simulates material removal by attracting nearby surface elements and subdividing them once they're near its opening. The laser locks nearby vertices and indicates adhesion by altering the retinal texture map. Finally, the drainage needle performs a simple gas-fluid exchange.

Four vitreoretinal surgeons evaluated the simulator over two separate testing sessions. A session consisted of three simulated surgical cases followed by a subjective Cooper-Harper survey ranking the simulator's training potential. The completed surveys indicate that with some refinement and better tracking the simulator could be a useful addition to the Ophthalmology training program.