

# VR for Microsurgery

---

**Team:** May1702

**Client:** Dr. Ben-Shlomo

**Advisor:** Dr. Keren

**Email:** [med-vr@iastate.edu](mailto:med-vr@iastate.edu)

**Website:** <http://may1702.sd.ece.iastate.edu/>

***Team Members/Role:***

*Maggie Hollander –Leader*

*Eric Edwards – Communication Leader*

*Sam Glancy – Webmaster/Communications*

*David Weinstein - Webmaster*

*David Robideau – Key Concept Holder*

*Kyle Rohlfing – Key Concept Holder*

|  |    |
|--|----|
| <b>1. Introduction</b>                 | 2  |
| 1.1 Project statement                  | 2  |
| 1.2 Purpose                            | 2  |
| 1.3 Our Solution                       | 3  |
| <b>2. Design</b>                       | 3  |
| 2.1 System Specification               | 3  |
| 2.2 Activity Solutions                 | 3  |
| 2.2.1 Incisions                        | 3  |
| 2.2.2 Injections                       | 4  |
| 2.2.3 Suturing                         | 4  |
| 2.2.4 Microscopes                      | 5  |
| 2.3 Requirements                       | 6  |
| 2.3.1 Functional Requirements          | 6  |
| 2.3.2 Non-Functional Requirements      | 6  |
| 2.4 Implementation                     | 6  |
| 2.4.1 Demonstration                    | 6  |
| <b>3. Testing Process</b>              | 7  |
| <b>A. Appendix - Operation Manual</b>  | 8  |
| A.1 Getting Started                    | 8  |
| A.2 Tools and Navigation               | 8  |
| A.4 Microscopes                        | 9  |
| A.3 Operating                          | 9  |
| A.3.1 Incisions                        | 10 |
| A.3.2 Sutures                          | 10 |
| A.3.3 Injections                       | 10 |
| <b>B. Appendix - Alternate Designs</b> | 11 |
| B.1 Initial Design & Plan              | 11 |
| B.1.1 Pros                             | 11 |
| B.1.2 Cons                             | 11 |

# 1. Introduction

## 1.1 Project statement

Microsurgery is a very specialized form of surgery done using microscopes. An example of microsurgery is removing a cataract from an animal's eye. Training for this procedure and microsurgery in general requires immense skill and is highly costly to practice over and over again. The cost is due to the necessary test subjects, like cadavers/preserved organs, and equipment. A different way to perform microsurgery training is needed to reduce cost and provide a more flexible and reliable system. For our project we will be researching and developing the basis for a system to improve microsurgical training using virtual reality. Virtual reality is still a relatively new market and many tools are currently not in place to do fine detailed work. However, VR is advancing daily and having a system in place that can be expanded upon when new technologies are released will provide for a fast moving prototype. We will be using an HTC Vive Virtual Reality Headset in order create an interactive virtual environment to aid in training microsurgical procedures.

## 1.2 Purpose

Currently, most microsurgical training falls into one of two camps. The first is training performed in a physical environment. This type of training is highly common and typically involves the trainee practicing and performing microsurgical procedures under an operating microscope. Generally, these training procedures are performed on either mannequins or non-living animals. Training in this way has both pros and cons. On the positive side, it allows trainees to practice using a real operating microscope and with real microsurgical instruments, the same kind they will be using in the field. It also allows for the user to have the tactic sensation of interacting with real or synthetic blood vessels, nerves, and tissue. On the other hand, training in a physical environment has its drawbacks, most notably cost. Operating microscopes are incredibly expensive as are mannequins.

The second type of microsurgical training that is performed is training conducted in a virtual environment. This is the type of training method our project is aimed at addressing. While various virtual training simulators have been developed in the past, they have suffered from some significant flaws that have limited their effectiveness--flaws we we feel the new VR technology can overcome. The first and most significant flaw these existing technologies have is a lack of realism in their simulation. Until very recently, the technology did not exist that could effectively render a realistic surgical environment while being able to effectively receive input from a user and provide realistic feedback. The second problem many previous attempts at virtual reality simulations faced was cost. Until recently programmable virtual reality devices were not commercially available at a large scale. Developers were frequently forced to develop

their own tools, adding to their cost. In the last year or two VR has made great strides however. Working with fine detail in a virtual reality environment is still being developed and major advances are being made daily.

We feel we are in a unique position to overcome many of the issues other forms of microsurgical training face. For starters, with the advent of programmable virtual reality headsets (like the HTC Vive), the cost of creating a virtual training simulator has dropped dramatically just in the last few years. On top of this, huge strides have been made recently in the ability to create and render believable and engaging environments in a virtual space, and the computational power has increased to the point that interacting with these environments requires less specialized equipment.

### 1.3 Our Solution

Our solution is to research and create the basis for a system that can be improved upon by more experienced game designers. The overall goal is to develop the major aspects of a system to train microsurgeons using a virtual reality environment. This allows trainees to attempt microsurgery as many times as necessary without the cost of tools and parts, giving them more practice and a stronger belief of their abilities going into real surgery. Our work is meant to serve as research into these technologies and methodologies and meant to serve as a basis for further development with the advent of virtual reality systems and their capabilities.

## 2. Design

### 2.1 System Specification

The system we are designing is an attempt to create a virtual environment capable of assisting in the training of microsurgical procedures. For our project, we have implemented solutions for the most common activities performed during microsurgical procedures. The activities we have implemented are creating incisions, injecting/extracting gels and fluids, and suturing. On top of creating solutions for these common microsurgical activities, we have developed an operating room simulation and simulated the feel of a microsurgical microscope in order to create a realistic simulation of how a microsurgical procedure is performed.

### 2.2 Activity Solutions

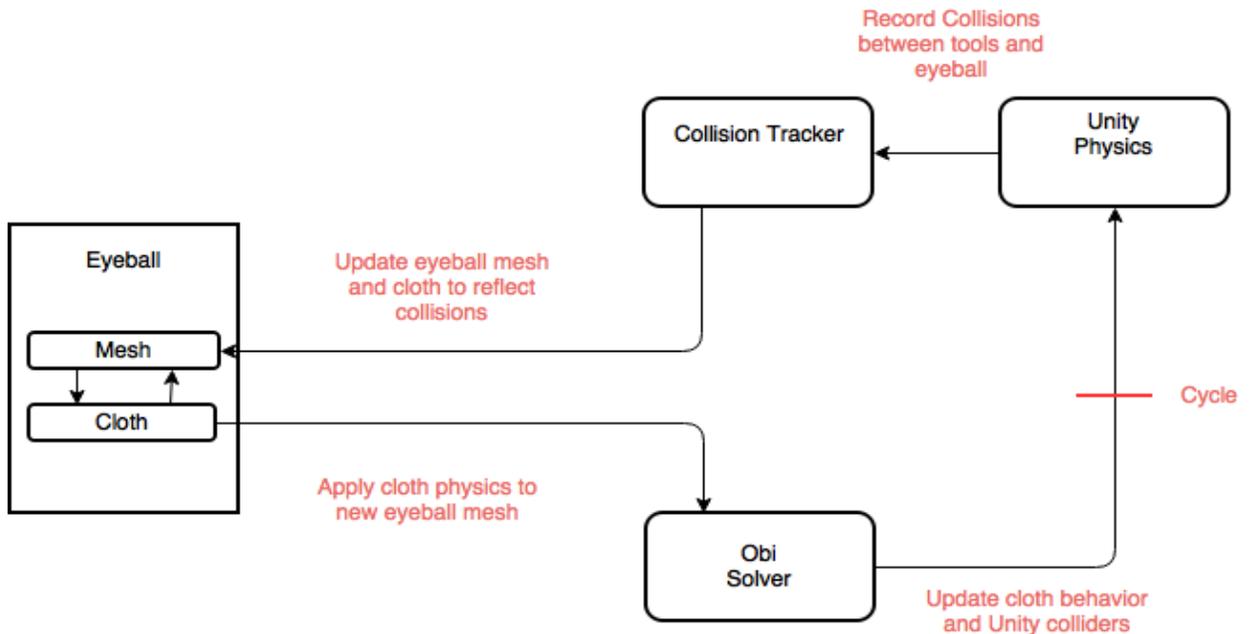
#### 2.2.1 Incisions

**Description:** Microsurgery on the eye requires extremely precise incisions. To simulate this process, users are able to make small, detailed cuts on the eye with realistic tools.

**Method:** Using a combination of mesh manipulation and particle physics generated by the Obi Cloth plugin, we can simulate the size and texture of the eye. We can also simulate precise incisions and varying tool shapes with multiple collision detection methods.

# Incisions

Fixed-step update cycle



## 2.2.2 Injections

**Description:** Microsurgery requires a lot of gel and fluid insertion and extraction. In order to most accurately model microsurgery, we need this aspect to be highly configurable for the different gels/fluids.

**Method:** Leveraging the technologies of the uFlex plugin for Unity, we are able to produce liquid in the form of particles. We then add configurations to these particles to influence the way they collide, stick together, and bounce off surfaces using some mathematical calculations. This allows for different forms of liquids. Combined with removal upon collision, we can simulate this aspect of microsurgery.

## 2.2.3 Suturing

**Description:** Manipulation of cloth particles created using the Obi Cloth plugin to bring two sides of a cut mesh together.

**Method:** Detecting where the tip of the suturing needle has collided, we get the closest particles

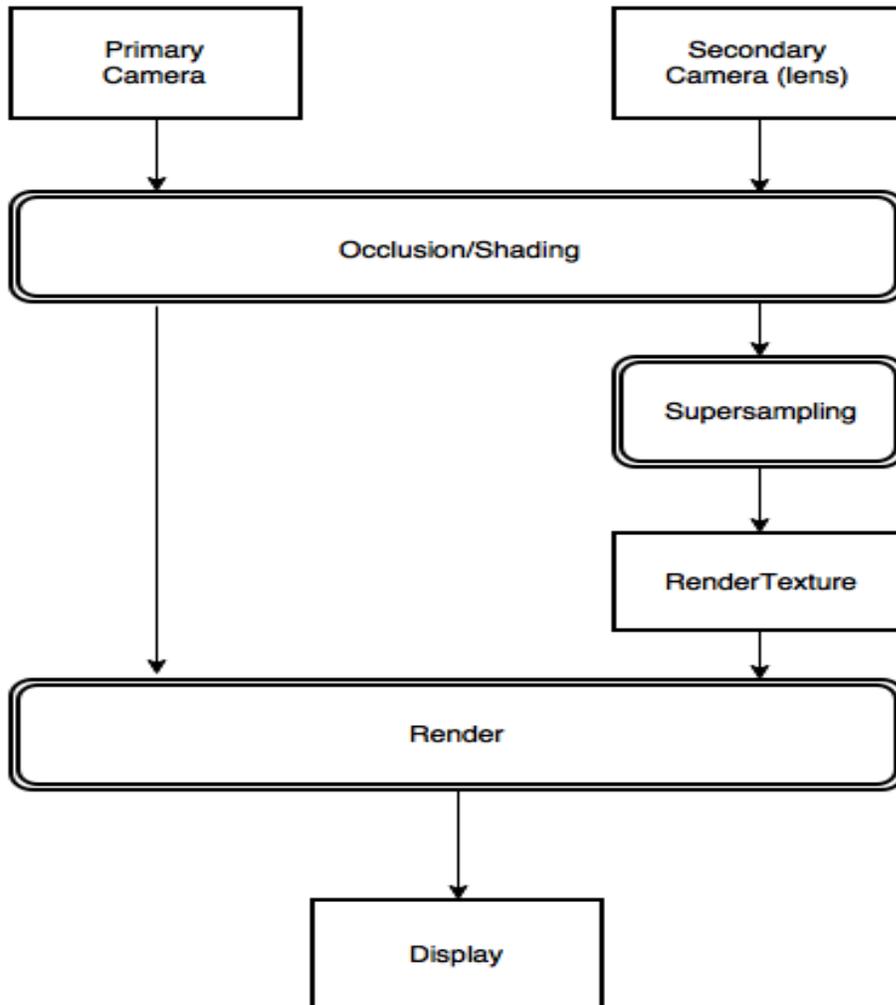
(vertices of the object mesh) to that point. After colliding with the other side, using the same methods, we interpolate particle positions to pull the sides of the “cloth” together.

### 2.2.4 Microscopes

**Description:** A microscope is needed to provide a high level of visual detail for the surgeon. To make the scene more realistic, a microscope is provided in the scene where users can look into it and see a blown-up view of the eye model they will be operating on.

**Method:** Combine multiple camera setups with varying FOVs to achieve variable zoom levels, supersample rendered images for high resolution.

## Microscope Render Pipeline



## 2.3 Requirements

### 2.3.1 Functional Requirements

- Simulate operating environment
- Provide recording/playback tools
- Create basis for microsurgery to serve as proof-of-concept
  - Incisions of the eye
  - Injection of fluids
  - Extraction of fluids/lenses
  - Suturing eye back together

### 2.3.2 Non-Functional Requirements

- Minimize motion sickness
  - Consistent 90 fps or greater
- Provide comfortable interface
- Simulate believable environment

## 2.4 Implementation

For our system, we have developed a number of solutions for the most common activities performed in microsurgical procedures. The idea is that future developers will build on top of these solutions to create full simulations of any number of microsurgical procedures. As such, the solutions we have implemented are each stand alone modules that can be added to a scene to allow for the chosen activity to be performed.

### 2.4.1 Demonstration

For demonstration purposes, we have composed a scene which implements each of our developed activities. When entering the scene, a user is placed in the operation room we designed. In the center of the operating room there is a surgical table and microsurgical microscope looking over the table. When the user approaches the microscope and places his or her eyes up to the eyepiece the scene is switched to one implementing our microscope feature. In this scene, instead of the view rendered to the Vive being a simulation of the user's position and movements in 3D space, the scene is meant to simulate a user looking through the eyepiece of an overhead microscope. This means user will appear to be looking down at the objects he or she is operation on even though his or her head is facing forward. If a user moves his or her head slightly the objects being viewed will appear to remain in the same spot relative to the user's view. When in this scene, a user is able to perform any of the three major activities we have implemented. Stationed in the center of this view is a sphere which the user can make precise incisions on, inject fluids into, and suture the sphere back together. The user is

able to control which activity is being performed by selected the corresponding tool on a menu permanently affixed to the users controller. These tools are a scalpel for making incisions, a syringe for injecting fluids, and a needle for suturing incisions back together.

### 3. Testing Process

As our system was developed as virtual reality simulation in the Unity game engine, we were limited in the forms of testing that could be performed. While there are some frameworks that allow a developer to create unit tests in Unity, ultimately we decided that attempting to implement these would be cumbersome and provide little use feedback about the legitimacy of our system. Instead we decided to rely on user testing to determine if our system was handling the procedures correctly. Our client, Dr. Ben-Shlomo, performs these microsurgeries in the Iowa State College of Veterinary Medicine. He tested our product a few times to view our progress and gave us feedback on what needed to be improved/changed. We were also given the ability to view a live surgery and record the procedure to better understand what we needed to mimic in the virtual space.

## A. Appendix - Operation Manual

### A.1 Getting Started

Using the surgery tools to perform simple operations requires an HTC Vive headset and two HTC Vive controllers. To run the MedVR program, build from Unity to an executable on your desired platform\*. Ensure that the Vive and controllers are connected to your PC via the SteamVR utility. Once all components are detected by SteamVR and the executable is ready, your system is ready.

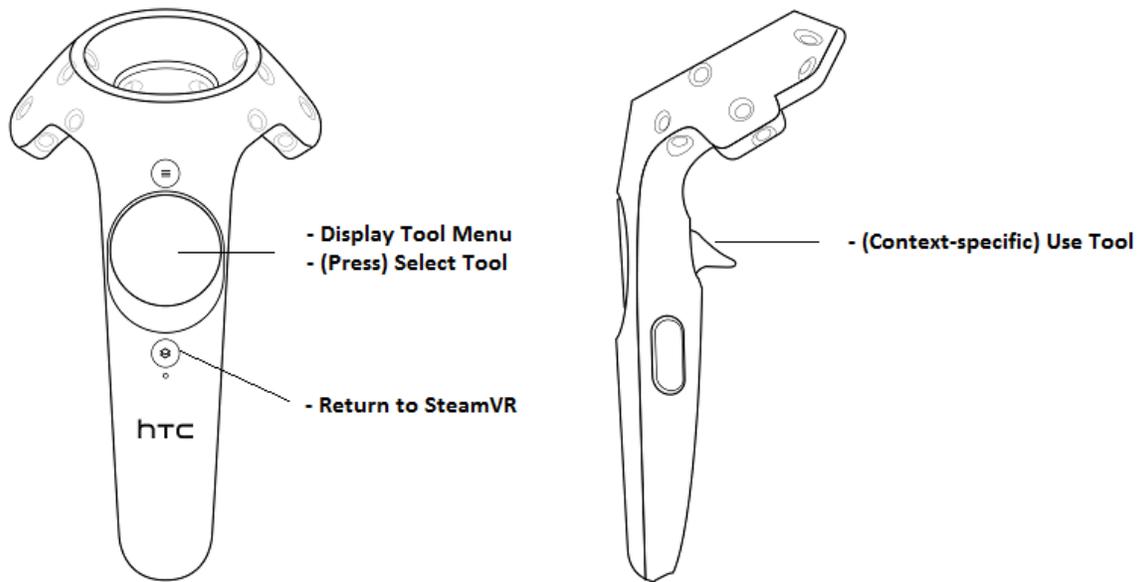
Start the MedVR program by running the executable. Your view in the headset will be replaced with a scene containing a menu allowing you to choose whether you want to drop directly into the surgery scene or if you'd like to get some more info about the eye there is an interactive scene allowing you to inspect all aspects of the eye and learn more about the biology. When you choose the surgery room however you'll be dropped into an office with a surgery table and microscope where you can begin your surgery practice.

\*currently, only the Windows OS is supported.

### A.2 Tools and Navigation

Once inside the operating theater, navigation will be mapped to your real-world position relative to the Vive sensors. If you experience trouble with navigation, ensure that your HTC Vive sensors have been placed far enough apart and within your line of sight. Note that a room-scale (minimum 6' x 8' area) VR setup is required to use the MedVR program.

Your Vive controllers are used for interaction with the environment. By placing your thumb on the Vive controller touchpad, a radial menu will be displayed with a number of usable tools. To select a tool, move your thumb so that the correct segment of the menu is highlighted. Press down on the touchpad with your thumb to select a tool.



### A.3 Microscopes

The MedVR program uses a combination of multiple cameras and a modified rendering pipeline with supersampling to achieve high-detail images of small objects. The zoom level, positioning, and supersampling factors can be adjusted within the Unity Editor prior to building an executable. Manipulation of the microscope can be achieved by repositioning the **ScopeCam** object in the Unity Editor.

In the standard operating theater, the Microscope has been set up to automatically handle transitions between scoped and unscoped views. To look through the microscope, simply move your head towards the microscope's eyepiece. Once your perspective has changed, you can tilt your head away to return to the operating theater view. Your hand positions and tools will persist between views.

### A.4 Operating

The MedVR program contains a sample operation scene with a single operable object. By using the microscope, this object can be manipulated with the tools accessed via the Vive controller. This object is a sphere designed to emulate the texture and relative size of an eyeball. A number of operations can be performed on this sphere.

#### **A.4.1 Incisions**

To perform an incision on the sphere, use your Vive controller to spawn the Scalpel tool.

The eye can be deformed by any part of the scalpel, and can be cut open using the scalpel's blade.

Incisions can be adjusted, to a slight degree. In the Unity Editor, the scalpel can be resized by adjusting the model's scale. Different sized scalpels produce different cuts. The scalpel's cut distance can also be adjusted (though this is not advised) by changing the Triangle Distance Tolerance value on the sphere object.

#### **A.4.2 Sutures**

Note: Suturing is relatively incomplete and may result in frame drops or strange subject behavior.

To perform a suture, first make an incision on the sphere. Once a cut has been opened, use your Vive controller to spawn the Needle tool. Pushing the needle through two points on opposite sides of the cut will draw those points together. Often, a cut will require multiple sutures. To finish a suture and begin a new one, press the trigger on the Vive Controller.

#### **A.4.3 Injections**

A simple fluid physics simulation has been provided in the form of the Syringe tool. Use your Vive controller to spawn the Syringe tool, then press the trigger on the Vive Controller to inject a basic liquid. Liquid properties (particle size, viscosity, etc.) can be adjusted in the Unity Editor.

## B. Appendix - Alternate Designs

### B.1 Initial Design & Plan

Our goal for this project was to provide the most instructional and useful application to train microsurgions. This involved a lot of additional features that are not found in our final design. We strived to create a few full surgical operations and combine it with a recording and performance metrics system. We took time to research the possibility of creating volumetric (rather than mesh-based) realistic organs given MRI and CAT scans of the body.

#### B.1.1 Pros

If we had pursued this design, we could possibly have an application with more detail and a more accurate feel towards an actual surgery. By adding the feature to create the volumetric organs, we could also have been able to create models that could be used for many different surgeries and having the volumetric base vs the mesh base we could possibly have done more operations on the organ itself.

#### B.1.2 Cons

The technology currently does not exist to make many of these features possible. As such, for many of these things, we had to find other ways to get the same effect. In order to have a project that can advance as the technology develops while demonstrating the need and prospect of this system, we had to change our design and approach. This meant scrapping many features found in this original design and focusing more on developing the basic structure for other main features and aspects of microsurgery.