

A VIRTUAL REALITY BASED ENVIRONMENT FOR  
ORTHOPEDIC SURGERY (VEOS)

By

MURUGAPPAN BALAKRISHNAN, BHARATHI RAJ

KUMAR

Bachelor of Science in Automobile Engineering

Bharathidasan University

Tiruchirappalli, Tamil Nadu

2008

Submitted to the Faculty of the  
Graduate College of the  
Oklahoma State University  
in partial fulfillment of  
the requirements for  
the Degree of  
MASTER OF SCIENCE  
May, 2015

A VIRTUAL REALITY BASED ENVIRONMENT FOR  
ORTHOPEDIC SURGERY (VEOS)

Thesis Approved:

Dr. Joe Cecil

---

Thesis Adviser

Dr. Terry Collins

---

Dr. Damon Chandler

---

## ACKNOWLEDGEMENTS

I am heartily thankful to my advisor, Dr. Joe Cecil, Associate Professor, Department of Industrial Engineering and Management, whose guidance stimulated my interests in knowing the various aspects of the project. I also thank him for his constant guidance which was a strong support to me. I also thank my committee members Dr. Terry Collins and Dr. Damon Chandler for providing me with invaluable support and guidance.

I thank my friends and CICE members Yajun Lu, Zak Zafar, and Rahul Chidurala for helping me in the various stages of my thesis completion. I am extremely thankful to Dr. Miguel A. Pirela-cruz, Head of Orthopedic Surgery and Rehabilitation, Texas Tech University Health Sciences Center, for lending his valuable support and guidance.

I would like to acknowledge the funding received from the National Science Foundation through grant number CNS 1257803 and the Interdisciplinary Planning Grant program from Oklahoma State University (Office of the Provost).

I wish to express my sincere and grateful thanks to my mother Mrs. B. Krishnaveni and Sister B. Indumathi for their constant support and encouragement. I appreciate the support I received from faculty and staff from School of Industrial Engineering and Management during my graduate study.

Name: MURUGAPPAN BALAKRISHNAN, BHARATHI RAJ KUMAR

Date of Degree: MAY, 2015

Title of Study: A VIRTUAL REALITY BASED ENVIRONMENT FOR ORTHOPEDIC SURGERY (VEOS)

Major Field: INDUSTRIAL ENGINEERING AND MANAGEMENT

Abstract:

The traditional way of teaching surgery involves students observing a 'live' surgery and then gradually assisting experienced surgeons. The creation of a Virtual Reality environment for orthopedic surgery (VEOS) can be beneficial in improving the quality of training while decreasing the time needed for training. Developing such virtual environments for educational and training purposes can supplement existing approaches.

In this research, the design and development of a virtual reality based environment for orthopedic surgery is described. The scope of the simulation environment is restricted to an orthopedic surgery process known as Less Invasive Stabilization System (LISS) surgery. The primary knowledge source for the LISS surgical process was Dr. Miguel A. Pirela-Cruz (Head of Orthopedic Surgery and Rehabilitation, Texas Tech University Health Sciences Center (TTHSC)). The VEOS was designed and developed on a PC based platform.

The developed VEOS was validated through interactions with surgical residents at TTHSC. Feedback from residents and our collaborator Dr. Miguel A. Pirela-Cruz was used to make necessary modifications to the surgical environment.

## TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Problem Statement.....	2
1.3 Goals and Objectives.....	2
1.4 Conclusion.....	3
II. LITERATURE REVIEW.....	4
2.1 Introduction.....	4
2.2 Orthopedic Surgery.....	4
2.3 Laproscopic Surgery.....	19
2.4 Heart Surgery.....	21
2.5 Other Surgery.....	26
2.6 Haptic Devices and Tracking Technology.....	28
2.7 Other Challenges.....	31
2.8 Conclusion.....	32
III. VIRTUAL REALITY FRAMEWORK.....	33
3.1 Introduction.....	33
3.2 Less Invasive Stabilization System.....	33
3.3 Architecture of the Virtual Training Environments.....	35
3.3.1 Visualization Module.....	35
3.3.2 Collision Detection Module.....	36
3.3.3 Position Training Module.....	36
3.3.4 Avatar Interaction Module.....	38
3.3.5 External File Module.....	39
3.3.6 Haptic Module.....	40
3.3.7 Surgical Manager.....	41
3.3.8 User Interaction Manager.....	41
3.3.9 Orthopedic Scenario Manager.....	42
3.3.10 Sensing and Feedback.....	42
3.3.11 Hardware and Software Interface.....	42
3.4 Collaboration Diagram.....	42

Chapter	Page
3.4.1 Class diagram.....	45
3.4.2 Keyboard Control.....	46
3.4.3 Mouse Control .....	47
3.4.4 Haptic Control.....	48
3.4.5 Avatar Control .....	49
3.4.6 External File Interaction .....	51
3.4.7 Color Indication .....	53
3.4.8 Display Control.....	54
3.5 The Unity based Non-Haptic Virtual Environment .....	54
3.5.1 Module 1: Learning to assembly the LISS Plate .....	55
3.5.2 Module 2: Virtual Surgery with use of assembled LISS Plate .....	57
3.5.3 Hierarchy in Simulation.....	57
3.6 The Unity based Haptic Environment.....	59
3.7 Creation of Semi-Immersive Environment.....	60
3.8 Description of the various functional module.....	61
3.8.1 LISS assembly module .....	61
3.8.2 LISS insertion module .....	62
3.8.3 Fracture reduction module.....	62
3.8.4 Screw insertion module.....	62
3.8.5 Guide removal module.....	62
3.8.6 Color indication module .....	63
3.9 Conclusion .....	63
 IV. TEST CASES .....	 64
4.1 Introduction.....	64
4.2 Methodology .....	65
4.3 Pre-Test .....	65
4.4 Post-Test .....	67
4.5 Results.....	67
4.6 Conclusion .....	69
 V. CONCLUSION.....	 70
5.1 Introduction.....	70
5.2 Summary of approach .....	70
5.3 Completed work.....	71
5.4 Future Research .....	72
5.5 Conclusion .....	72

REFERENCES .....	74
APPENDICES .....	79

## LIST OF TABLES

Table	Page
1. Results August 2014 .....	66
2. Results February 2015 .....	68



## LIST OF FIGURES

Figure	Page
1. View of an inserted LISS plate .....	34
2. The key steps in LISS Plating surgery .....	35
3. Architecture of Virtual Environment for orthopedic surgery (VEOS) .....	35
4. Flow chart for position training .....	38
5. Collaboration diagram .....	44
6. Exploded view of manual surgery simulation class diagram.....	45
7. Exploded view of automatic surgery simulation class diagram.....	46
8. Animator controller for the avatar in the simulation .....	50
9. Animation flow diagram .....	51
10. Sample external file coordinates .....	53
11. LISS components in VEOS .....	55
12. VEOS with some of the LISS components being assembled .....	56
13. Virtual surgery with use of assembled LISS plate.....	57
14. a. Parent and Child attach function, b. Parent and Child detach function .....	58
15. Resident interacting with VEOS using Phantom Haptic interface .....	59
16. Semi-immersive VR environment for VEOS .....	60
17. Test performance during August 2014 .....	68
18. Test performance during February 2015.....	69

## CHAPTER I

### INTRODUCTION

#### 1.1 INTRODUCTION

Virtual Reality (VR) environment is being increasingly used to train medical residents and surgeons. In earlier days, animals and cadavers were widely used to train the surgeons. The current technological circumstance provides various opportunities in these research areas in order to develop cyber training systems efficiently. Alternative techniques and strategies used in the surgical training are growing rapidly. Virtual reality is one of these alternative training technologies in medical surgical training. VR simulation environment enhances the surgery procedures in the planning, surgeon performances, reduced operating time and cost. VR-based simulation tools are being used in heart surgery, laparoscopic surgery, and endoscopic surgery, among others.

Various papers and articles have also highlighted the benefits of using VR-based simulators in medical training and education [2], [3], [6], [7]. These simulators can provide a low risk environment for especially training beginning surgeons or medical residents as they provide an environment for planning and performing complex surgical processes repetitively to improve skill levels.

## 1.2 PROBLEM STATEMENT

The traditional way of surgical teaching involves students first merely observing a “live” surgery and then gradually progressing to assisting experienced surgeons [1-6]. Medical residents currently also acquire their skills by performing surgeries on cadavers; however, there are some inherent limitations with this approach such as availability, cost and the possibility of infections, which limit their usefulness. The use of virtual surgical environments will enable students to learn the appropriate way of managing various conditions with safety considerations to the students and patients. It will enable residents and surgeons to assess as well as proposed alternative ways surgically respond to a specific medical condition. Such virtual environments can be used not only to acquire new skills young or budding surgeons, but also enable experienced surgeons to continue practicing and maintain their skill levels.

Orthopedic surgery is one of the most difficult surgical domain. The creation of virtual training simulators in this domain will support better training alternatives. There is a need to design and develop a simulator for LISS plating surgery. Such a simulator will provide an effective training environment.

## 1.2 GOAL AND OBJECTIVES

Overall, the research goal is to demonstrate the benefits of using virtual reality based simulation in orthopedic surgical training. To achieve this goal, the following objectives were identified:

- i. Design and develop a VR based environment for orthopedic surgery pertaining to the LISS surgery process.
  - This VR environment was built using the Unity VR tool.
- ii. Incorporate force feedback using haptic technology
- iii. Validate the environment through interactions with an expert orthopedic surgeon.

- The validation of the developed virtual surgical environment was coordinated with Dr. Miguel-Pirela Cruz at Texas Tech Health Science Center (TTHSC).

#### 1.4 CONCLUSION

In this chapter, an introduction to the importance of virtual surgical simulations was provided. The problem statement was outlined. The goal and objectives were also discussed. The next chapter provides an overview of all the previously conducted research on virtual surgery methods and simulations.

## CHAPTER II

### LITERATURE REVIEW

#### 2.1. Introduction

Virtual Environments have been used to train medical surgeons in orthopedic surgery. Using such simulators, residents and surgeons can develop or improve their skills in specific orthopedic surgical processes.

#### 2.2. Orthopedic Surgery

To have an accurate model of human tissue deformation, there was a need to model this realistically according to the layered nature of real human tissues. Qin et al. [9] developed a framework for simulating the soft tissue deformation in an orthopedic surgery. In their work, they first constructed the multi-layered soft tissue model according to the segmented Chinese Visible Human (CVH) datasets, which can provide more details compared to an MRI or CT images. Then the non-linear stress-strain behavior of soft tissues was modeled employing a bilinear 3D mass-spring model and later simulated annealing was used to modify the spring parameters of the model. They used a Physics Processing Unit (PPU) for the first time (as claimed) to perform efficiently the computational analysis of dynamic motions and interactions. They showed that PPU can improve the simulation performance to a great degree. The experimental

results discussed highlight the practical use of their model both in providing interactive and realistic models of human tissue deformations.

Delp and Loan [10] developed a graphical tool called SIMM (Software for Interactive Musculoskeletal Modeling), which can be used to develop and analyze musculoskeletal models. Once a model was defined by the user, the functions of each muscle was analyzed by computing its length, forces, moment arms, and joint moments. The user can develop, alter, and evaluate models of many different musculoskeletal structures, which quantify the effects of musculoskeletal geometry, joint kinematics, and muscle-tendon parameters on muscle-tendon lengths, moment arms, muscle forces, and joint moments. A model in SIMM consists of a set of rigid body segments which were connected via joints. The joints were spanned by muscle-tendons actuators which result in generating forces, and hence, joint moments in them. The software was beneficial for the analysis of 3D images alone (such as what we get from CT scanners) doesn't give much information about the function of the muscles and joints because the image itself does not quantify a muscle. SIMM would be of great value for many purposes, such as exploring the effects of musculoskeletal surgeries on the moment-generating capacities of a muscle. The software was the first to use an interactive graphical environment, which gives many musculoskeletal models with no need to program. It was also used in approximately 25 centers worldwide.

Vankipuram et al. [11] outlined a drilling simulation environment which can be utilized to train orthopedic surgeons. They used a 3-DOF haptic device interfaced with this environment. For volume rendering, there were various algorithms in the literature, generally divided into two categories: the Direct Volume Rendering (DVR) and the Surface Fitting (SF) methods. The Marching Cubes algorithm was used in [11] for volume rendering tasks in their outlined approach. To create a realistic drilling simulation, two key components were taken into consideration: one was the appropriate collision detection algorithm and the other was employing

the proper voxel removal method. The collision detection algorithm enables the detection of virtual target objects coming into contact with the simulated drilling tool; this was achieved by a hierarchical volume bounding method.

Modeling the drilling procedure in the orthopedic surgery simulators plays a key role in the development of orthopedic surgery simulators. To obtain a realistic feel of the bone drilling process, there was a need to calculate the drilling forces coming into play, which was then communicated to the haptic device that interfaces with the virtual simulation environment. Tsai et al [12] presented a model to calculate these forces when modeling a real-like drilling procedure for building an orthopedic surgery simulator. Their model provides an environment capable of choosing the appropriate drill, plate and screw which best fits the patient-specific bone. In their work, haptic functions which can be added to a volume based surgical simulator were introduced. These functions calculate the loads on cutting tips and chisel edge based on the machining theorem method to acquire forces and torques along the thrust, tangential, and radial directions. A 6DoF PHANTOM haptic device was utilized to let the user control the drill angle, position, and speed. They also validated the effectiveness of their model by simulating a plate-and-screw surgery using the developed haptic functions.

VR simulators can be beneficial in increasing the quality of surgical training while decreasing the long training time needed for the education. Blyth et al [13] carried out a survey regarding the attitude of the surgeon community toward the need for VR simulators. Two groups of surgeons were examined during this survey; the first group includes the surgeons qualified before 1990 and the second group consists of the ones qualified after 1990. In their work, two major key points were studied: (i) the participants were asked about the perceived requirements of a surgical simulator and (ii) the type of task for which the surgical simulator would be useful. In addition, they discussed the degree of acceptance of this kind of simulators by the surgeon's society. The surgical simulators were shown to be beneficial in particular for novice students. The

surgeons believed that a realistic simulation of the operation was the most essential need of the VR simulators. In addition, both groups of surgeons concluded that the VR simulators would be helpful, especially for applications in orthopedics, particularly in practicing some techniques such as guide wire placement and minimally invasive surgery, which can provide useful feedback to surgeons in a non-threatening way. In general, they were supportive of the simulators, but they did not think they would have a key impact on the near future of surgical training.

Tsai et al [14] developed a VR orthopedic simulator, which can simulate different orthopedic procedures, including arthroplasty, corrective and open osteotomy, fusion, and open reduction. The simulator was developed on C++ using the OpenGL libraries, and employed computed tomography (CT) or magnetic resonance imaging (MRI) to build the volume data of the orthopedic target and simulate the surgical procedure by providing stereoscopic images of the orthopedic scene. The surgeon wears stereoscopic glasses and uses a surgical instrument attached to a six dimensional-degree tracker to perform the surgical operation. The interface module of the software provided different menus, which allowed the surgeon to select different surgical tools such as bone saws, virtual plates, staples, dissectors, etc., And used different functions including fusion, sectioning surfaces, and determining the collision in order to perform different surgical procedures on solid orthopedic objects such as bone, prosthesis, and bone grafts. The simulator was designed only for performing surgical procedures on rigid orthopedic objects and was not able to model soft tissues and does not provide any force feedback to the surgeon.

During most orthopedic surgery training activities, the medical residents were introduced to correcting the fractures on artificial bones. There were two main problems: (i) synthetic bones were expensive and (ii) for certain types of the bones, the synthetic form might not be available. Further, there were benefits to training using virtual models when they were based on real fracture images (obtained from CT and MRI images) [15]. While VR based systems require advanced processors and equipment, some researchers explored the creation of VSEs, which can



be run on typical personal computers (PCs). For example, Sourina et al [15] developed a virtual orthopedic surgery simulator, called Virtual Bone Setter, capable of running on typical PCs, which can be used to train the surgery residents in correcting bone fractures. The software they developed can be run on ordinary PCs without any need for external VR hardware. They also built and used a geometric database of different fractures, as well as different orthopedic objects, such as screws, plates, nails, wires, and locking bolts, which can be selected by the surgeon.

In [16], a VR-based training system for arthroscopic surgery (diagnosis of joint irregularities) was elaborated. The authors investigated the mechanical design, kinematics, dexterity measure and control loop of haptic devices. They also discussed the organ mesh generation, tissue deformation simulation and collision detection techniques. The manipulator mechanism with four degrees of freedom was developed and kinematic analysis of the manipulator was performed using direct and inverse kinematics analysis. The usable workspace of the manipulator was identified using Jacobian matrix and dexterity measures. A control system for the haptic device was developed for tracking the arthroscopic and the surgical instrument. They also further developed a knee model from the visible human project (VHP) color images and 3D meshes for finite element analysis to simulate a series of actions in the surgery. They also used a complex Finite Element Model (FEM) to deal with non-linear tissue deformation and topology changes. An advanced collision detection algorithm was used to detect virtual collisions between arthroscopic instruments and organs.

Kong et al., [17] developed a prototype of a telesurgical system for bone-setting in order to maintain a safe and accurate operation procedure and reduce surgeon's effort using remotely accessible robots. The robotic system can perform slave/master operation, and in semi-automatic and autonomic operation modes. The system also utilized image guidance technology to receive visual feedback for the positioning and trajectory control. Haptic devices were used in the master devices to get user force feedback on tools. In the slave manipulator, a parallel robot with 6 DOF

was adopted for its compactness, high stiffness and load capacity in order to maintain safety and loading abilities. VR and AR techniques adopted in this simulation for the guidance of operational processes and the 3D models of the system were created using JAVA 3D techniques. Some of the benefits of the teleoperating system were described and the benefits of these systems were reducing the irradiation damage to the surgeon, improving the reposition and locking nail quality, and relieving suffering of patients. The present system communication was based on TCP/IP networks and time delay was ignored. The force, including position loop control system was implemented to make the position and orientation of the tools precise. The doctors can predefine the surgical procedures using the control and graphical interfaces developed in the human machine interface. Finally, the teleoperation test with a broken bone with the information about images and forces to complete the bone-setting were discussed.

Vanicatte et al., [18] implemented interoperability features in a robot-assisted orthopedic surgery system with smart interaction between medical devices during operations. The authors addressed the current lack of reliable communication between the robot and medical devices which can pose a danger to the patient. A central decision making unit (CDMU) was added to the existing robot assisted surgery systems to monitor and make proper decisions for the patients. Some of the elements that were supervised by the CDMU included the anesthesia machine, imaging devices and the robot assistant. The CDMU also had a telemetry unit interface for transferring and receiving the data from various devices using wireless technology. In this proposed design, a simulation of patients with normal, slightly abnormal, and critical risk states were considered with specific values of physiological data, blood pressures, oxygen saturation, and BIS monitoring. The CDMU collects the physiological state of the patient and the decision support software determines the patient condition and shows the details in the CDMU display. The reaction time from the system was compared with the reaction delay during without interoperability features in robot-assisted surgery environment. It showed the absence of

interoperability lead to dangerous inattention mistakes and may cause a significant impact on the patient condition. The CDMU technique enhanced the safety of the surgery and assisted the surgeon, but this had some issues such as lack of standardization of the medical devices and their proprietary wireless technology.

Zhijiang et al., [19] proposed a teleprogramming scheme combined with semi-autonomous control for telesurgery contexts to solve the Internet induced time delay, operation fatigues. Teleprogramming was utilized in the telesurgery systems to control the time delay by using a concise robot control language and semi-autonomous control, which helps the user to operate simultaneously. The present work was applied on the long bone fracture therapies for precise repositioning and locking of intramedullary nails. Parallel and serial robots HIT-RAOS were used to fix the broken bones in addition to a C-Arm X-Ray machine used to collect the images. In order to avoid X-Ray exposure, which could have harmed the surgeon and patient, a virtual surgical simulation system was developed for interaction using JAVA 3D techniques. The teleprogramming scheme utilized the robot languages and a background program to reduce the time delay and it helped the operator performs accurately even when the condition of the Internet was deteriorating. Client and the server part of the Semi-autonomous controls were described. An experiment was conducted for locking of intramedullary nails utilizing the teleprogramming scheme combined with semi-autonomous control. The client and server were 4kms apart and the experiment was repeated 10 times in the following modes: video assisted teleoperation and virtual surgical environment. The results showed that the performance of the virtual surgical environment improved by decreasing the time delay in telesurgery.

Monan [20] developed a finite element model (FEM) of the leg and tissue from the actual 3D geometry and investigated the biomechanical characteristics of the model during the orthopedic surgery. The geometry of the FEM was reconstructed from the CT images and the boundary surfaces were processed using Solidworks. Initially key points were created in the FEM

and those key points were used to generate the closed solid models. The FEM were defined as linear elastic and hyper elastic. The bone structures and tendons were called as linear elastic and the soft tissues and muscles were called as hyper elastic models. The Mooney-Relvin model and its functions were used for the finite element models. An experiment was also conducted to observe the force required to separate the two broken bones and distance among them. These forces were obtained by using a force sensor and the ruler made from steel balls. In the experiment, CT images were imposed to see the condition of the patient and from the unloaded condition, every 10 seconds, a parallel robot was moved 1mm forward to conduct the force calibrations. The experimental results were compared with the computational results, which demonstrated the finite element data was reliable. The simulated model was shown to have the ability to predict the force required in the reposition procedure.

Zhang et al., [21] proposed a novel virtual simulation system for robot assisted orthopedic surgery system (HIT-RAOS) to help the surgeon to develop operative planning, surgery rehearsal, telesurgery and training. The hardware of HIT-RAOS contained parallel and serial robots with c-arm X-ray machine for the treatment of long bone fracture reposition operation and locking the intramedullary nails. The surgical environment was divided into three different models and those models were human, auxiliary devices and robot models. The virtual reality (VR) environment of the surgery process was developed using Java3D and VRML techniques. The robot model was created in the pro/engineering geometry models and this embedded with the kinematic details to demonstrate the real time motions in the VRE. Finite element method was used for the leg model and it gave accurate results with real elastic behavior of soft tissues. The geometrical models of the muscles and bones were reconstructed from the CT images of the leg. In the experiment, these models were integrated into the virtual environment and a telesurgery experiment was conducted. The surgery environment contained a real time six degrees of freedom robot on the client side and the virtual system on the server side. These

servers and clients were connected via LAN and every 15ms the data was transported to the virtual simulation system and then the virtual parallel robot move according to the real time movement from the client end. The system also contained force and vision feedback to make the perfect movements of the robot. In conclusion, the system proved that the virtual environment was useful and valuable for surgery practice.

Rambani et al., [22] conducted a study of computer assisted orthopedic training system (CAOS) for fracture fixation and validation of its effectiveness among the junior orthopedic trainees. In this study, the CAOS system developed by their simulation and visualization research group was used. It included a fluoroscopy-based navigation system that combines intraoperative fluoroscopy based imaging using c-arm techniques with surgical navigation concepts. This system was especially helpful for the trainees that had less experience in 3D navigation with 2D images. An experimental study was conducted with the groups that had experience in 3D navigation and was compared with the groups that had no exposure to 3D navigation. The results were assessed based on the amount of time taken, accuracy of guide wire placement, and the number of exposures required to complete the process. The scores were analyzed using statistical applications. In the study, the comparison showed a significant decrease in all parameters of the first group that was greater than the second group. Finally, the computer navigated training system improved the accuracy and time taken to complete the surgical procedures.

Rosenblum and Macedonia [23] gave an overview of surgical procedures followed in the common orthopedic surgery training. The most common orthopedic surgery-training students used the synthetic plastic bones to fix the fractured bones using surgical tools and implants. Afterwards, they used the cadaver for practice before moving to a real surgical environment. The potential drawback of synthetic plastic bones was that they easily break before the student can demonstrate surgical procedures and the better synthetic models were quite expensive. In order to learn the techniques, use the respective tools, place the implants, and get muscle memories, the

PC enabled VR techniques were used. This technique included the realistic 3D geometry of the patient from CT or MRI images, collision detection and sounds, real time 3D rendering, and input techniques. A geometric database was developed to store the different type of surgical tools and broken bone models using polygonal mesh with functional descriptions. To detect the collision a pseudo-physical collision detection method was implemented in the model. The environment guided the surgeon to complete the entire surgical procedures and also enabled the realistic sounds during the use of the instruments. In the experiment, the surgeons were requested to practice fixing the inter-trochanteric fracture by inserting the guide wires, pins, and remove the wire. It guides to identify the appropriate DHS screw, plate, and cortex screw sizes. In conclusion, the implemented tool used more sophisticated virtual tools and increases the realism, which gave a better immersion.

Burdea et al., [24] investigated the telerehabilitation system using VR and haptic interfaces in the networked system. The rehabilitation therapy was arranged for remote, rural located patients with recent orthopedic impairments. To monitor the improvement of the patient, they used haptic and network interactions. The system had two personal computers connected to the Internet from home to clinics. The home computer had a mouse/keyboard input, interactive sound generator, InsideTrack 3D magnetic tracker, and a multipurpose control system. The tracker measured position and orientation of the fingertip, and a camera provided the patient a support to interact people by teleconference. The clinic server monitored the patients exercise data and stored it in a database for later analysis. Rehabilitation software had a 3D graphics environment, patient database and graphical user interface. The 3D graphics environment generated the graphical images for the VR environment; graphical user interface helped the patient to select the required exercise mode from the menu options. There were two types of physical and functional rehabilitation that were analyzed in the VR environment. Physical rehabilitation identified the finger force exertion and range of motion; functional rehabilitation

examined the maximum force exertion level by varying the difficulty of handling the devices. Obtained results were stored in the database by high and low level formats. The high level stored the specific finger forces and the low level stored the finger forces during exercise. Experimental results of the rehabilitation showed a significant change in the grasping forces and also improved the hand-eye coordination function.

Heng et al., [25] developed a VR training system for knee arthroscopic surgery with a tailor-made force feedback device. The existing haptic devices had a limitation in the positional forces and it did not provide the torque feedback. The developed feedback device had a two-hand haptic interface and it enabled the user to navigate and rotate the probe. The device had four degrees of freedom motion mechanism for each handle. The pitch, yaw, and insertion were the three DOF for the arthroscope and the fourth DOF was rotational, which enhances the surgeon's interaction. Three optical encoders were used to track the position and orientation of the probes. The knee-joints models were designed based on the nondeformable and deformable organs. The bones were modeled using surface meshes and the muscle and ligaments were modeled using tetrahedral meshes. These models were obtained from the human project image datasets and typical constraints in modeling meshes were elaborated. The authors also developed the soft tissue deformation model using complex finite element non-linear methods by topological changes in the operational and non-operational regions; cutting algorithm of the soft tissue by subdividing the tetrahedral meshes by tracking the intersection points among the cutting tool and each intersection; the collision detection technique was included during the navigation and cutting. During the experimental studies, the haptic devices achieved a satisfactory feedback and hand-eye coordination. In the end, the developed system provided a mesh generation, real-time soft tissue deformation, and cutting and collision detection to users.

Padilla-Castaneda et al., [26] discussed the integration of a robotic system and VR applications for the orthopedic rehabilitation of the arm representing the strengthening training

and motion recovery. The system allowed exhaustive exercising by motor control, giving visuomotor, haptic feedback and trajectory positioning guidance. The most important part of the system was to assign specific tasks to perform within the virtual environments and helped to evaluate the mobility condition of the patient to personalize the difficulty level of the therapy and provided kineseologic measurements of the patient evaluation. The system used The Robotic rehabilitation device called BRANDO, which simulated the upper extremity of the patient in real time and the physical interaction with the virtual object provided the visual feedback of the patient's arm movements. The system had two VR training applications for the recovery of the elbow and forearm motion and the system Uses Graphical User Interface for managing the system virtually according to the specific patient condition. It included the personalizing the training sessions, tracking the performance of the patients with the managing database of the patient, which consisted the graphical and statistical report. The system took advantage of robotic therapy with task oriented VR at different simulating environments. It also included the patient registration, the personalization of the therapy and the modification of the difficulty level, controlling and monitoring the training sessions, and generating the patient's reports as necessary. The article stated it had conducted pilot experiments with 3 different patients where the patients risked performing more challenging movements and all the patients showed confidence in using the system and many clinical staff were satisfied with these systems.

Eriksson et al., [27] developed surgical training tools using haptic and VR techniques for the milling operation in temporal bone surgery. They utilized the marching cubes algorithm for the visualization of the developed skull bones based on the CT and MRI scanned images. The authors also tried voxel-based haptic rendering techniques for the force feedback. A collision detection method was applied between the probe and VR-object in order to obtain realistic force feedbacks using voxels density values. Material removal modeling of the milling process was developed using the energy-based approach for an accurate detailing in the surgery. H3D API



based haptic devices were designed and developed for these systems. The authors also implemented the graphics and haptic interaction in the surgical environment for the better surgical procedures. In this method an updating frequency of 1000Hz was able to give a realistic experience of the bone milling. In the conclusion, the authors presented the future direction in tele-robotic surgery systems using a VR system for milling operation.

Pappas et al., [28] studied the individual surgeon's progress in a VR simulation for Shoulder Arthroscopy. The study was conducted with 43 surgeons and the results were evaluated based on the time of completion, number of probe collisions with the tissues, average probe velocity, and distance travelled from the simulated probe to optimal computer determined distances. The results were compared with the historical data of experienced users up to three years. The results showed improvement over the historical data and also the performance had improved upon the moderate user groups. At the conclusion, the authors demonstrated techniques that could be used for teaching surgery skills more particular to maintain hand-eye coordination. The authors also explained the future directions for actual surgical procedures using advanced simulators.

Lundstrom et al., [29] developed a multi-touch table system for orthopedic surgical planning. The article focused on the 3D visualization table design for better understanding of the surgical planning and also demonstrated the two novel interaction components in the touch tables. The article also utilized the user interaction study to explain the uses of the device. The developed device contains the following features: a large multi-touch display for visualization of clinical scenarios, interface with rendering for interactive, 6DOF interaction with 3D rendering, free orientation along with the main axes, movable alternator pucks for feature sets with touch gestures, and feasible zooming projections. Finally, the developed device enhanced the orthopedic surgeon clinical task with better understating of the complex human anatomy.

d'Aulignac et al., [30] proposed a mass-spring model of the dynamics of a human thigh based on real data to detect thrombosis illness in the vein. The current state of the vein was deduced based on the pressure applied to the echographic probe on the thigh, which subsequently was used to identify the illness. Two different probes were used to study the behavior of the thigh with respect to force being applied. The measurements were taken over the region of the thigh at multiple locations where the robotic arm advances based on data from end effector. The recorded force was repeated until the upper force limit. In order to simulate the real scenario, a two-layer model of the thigh using both linear and non-linear spring was proposed. Using least-squares minimization method, the parameters of the springs were estimated. Finally, a fully functional simulator coupled with a haptic interface was used to train practitioners for echographic exams.

Measure and Chaillou, [31] simulated dynamic behavior in organs without compromising the real time simulation and generality. The author stated the main problems in current models – lack of the realistic organs and tools, interaction with its environment, and structure modifications in the process. They discussed the need for synchronicity in the dynamic models and the types of interactions between the tool and the organ. With this in consideration, the author proposed a mechanical body model - a spring model with a rigid component in order to attain a compromise between memory, realism in real time. Further, they used Euler's method and integrated the equations so to obtain orientation of the parts. They analyzed the model to list down the pros and cons. The conditions for limitations, synchronicity and deformation, were discussed with a possible way to solve the problem partially. The author accepted that the model was under construction, but asserted that though synchronicity had not been reached, the simulation was interactive. The authors stated that instead of simulating with complex models, the proposed simple model, which works in real time realistically, would make it possible.

Mabrey et al., [32] questioned the reliability and practicality of VR in orthopedics and whether VR should be utilized in all orthopedic practices. To answer their questions they

researched different trials in VR training versus no VR Training. They discussed many separate cases where surgery residents were split up into two groups: one with VR Training in the surgery and one without. The results in all the cases showed improvements to the students that had trained with VR such as faster completion time, better accuracy, and better overall comprehension of the procedure.

Karaliotas., [33] identified and discussed the many challenges in creating accurate and practical VR simulations for Surgery. These challenges included accurately depicting the complex anatomy of surgical subjects as well as making the use of the surgical tools realistic. The authors also discussed the development of their own VR System that improved some of these aspects. To accurately depict complex anatomical structures they used MR and CT scans to generate a realistic 3D view of the structure. To improve realistic tool movement, they used an NDI Polaris Optical Tracker which helped with real time tool orientation during the simulation.

Willaert et al., [34] explored Patient Specific VR Simulators and discussed the practicality of its uses. They review through 12 different PSVR systems. Out of 12, 11 of them were still in prototype stage while only one was actually commercialized. Their conclusion was that PSVR has a lot of potential and was a very significant technological advancement in the field of medicine.

Karaliotas., [35] discussed the short comings of current medical simulations such as lack of a 3D interface or having to work with animals or test patients. The author also discussed the validity in using VR Simulators to train new and upcoming surgeons. The author went through the history and development of VR and asked questions pertaining to the practicality of surgical simulation such as “Were the skills learned in a virtual environment transferable to an operating room?” The author’s conclusion was that Virtual Simulation training was valid in that it decreases

the chances for potential risks by allowing realistic training that was flexible to the certain surgical situation.

### 2.3. Laproscopic Surgery

Laparoscopic Surgery was another surgical field in which VR based simulators were introduced.

Azzie et al [36] developed and validated a pediatric laparoscopy simulator (PLS) subsequent to its adult laparoscopies counterpart. This simulator was designed to teach and assess the pediatric surgical skills by enhancing the adult version so that it represents the unique characteristics for the pediatric surgeries. As detailed in [36], while the simulator inherited many common features of the adult version, it had addressed the previous limitations, including the need for notably reduced working space, the smaller field of view, the more delicate children tissues, and the more precise and sensitive movement required for the surgery. The intra-corporeal suturing was reported as the task which has the greatest difference in performance between the adult and PLS simulators. Subsequently, the simulator was tested with a group of candidates to assess their level of expertise in pediatric surgery. They showed that the PLS simulator was able to differentiate between novice, intermediate, and expert surgeons. Moreover, the results show that the PLS simulator's tasks were more difficult to perform than the adult simulators, which agreed with the initial hypothesis of the authors.

In the world of robot-assisted surgical training, there were still many obstacles, including the high cost associated with the surgical instruments. One of the major technological innovations in the field of robot assisted surgery was the DA Vinci surgical robot (DVSS). For such surgical contexts, there was a need to train surgeons in using the surgical robots; VR based simulation can enable such virtual training activities. Robotic Surgery Simulator (RoSS) was a novel VR based simulator for the DVSS. RoSS was developed in collaboration between the University at Buffalo

and Roswell Park Cancer Institute. Before a surgical simulator could be used, there must be a way to evaluate its validity and as the first step, the face validity of the simulator must be tested.

Sexias-Mikelus et al. [37] discussed the face validity of RoSS by performing a study between thirty surgeons and novices. The result showed that RoSS was real close to the DVSS console in terms of virtual simulation and instrumentation. The results of this study had been categorized in different aspects such as: closeness of RoSS to DVSS, feel of the pinch device, movements of the arms, clutch and camera movement, and visual display.

Makiyama et al [38] developed a virtual surgery simulator with the use in laparoscopic renal surgery. The simulator was patient specific; using the CT images of each individual person to build the volume model of the organ under consideration, the simulator enables the surgeon to perform the pre-operative rehearsal process. The physical simulator utilized two forceps, one foot pedal and one scope for the assistants. The pedal made the surgeon perform an electrical cut by pressing it. Different surgical tools were also available and could be changed using forceps and foot pedal by the surgeon. There was also a haptic device which gave the surgeon a much needed tactile feedback of the interactions between the simulated body organ and the surgical tool. The simulation process started by first using the CT data to build a volume model of the live organ using an extended region-growing method [38]. The soft-tissue deformations were modeled using Finite Element Methods (FEM). The simulator discussed in [38] had been used by surgeons in ten cases in clinic centers and reported as a useful tool for performing pre-operative rehearsal. In addition to beginners' training and performing the pre-operative rehearsal, the simulator could also be used by expert surgeons to propose creative techniques and non-routine approaches in laparoscopic surgery, which could enhance the planning process and facilitate a safer operation.

Grantcharov et al [39] examined the impact of VR surgical simulation for improvement of psychomotor skills relevant to the performance of laparoscopic cholecystectomy. Their research objective was to validate the role of VR simulation in surgical training by assessing the

possibility of improvement and revision of surgical skills obtained through training in a VR simulation module and its replication in the physical operations. This study was carried out in three departments of abdominal surgery in teaching hospitals using the Minimally Invasive Surgical Trainer – Virtual Reality (MIST-VR). In this study, twenty surgeons with limited experience in laparoscopic surgery participated and they were assigned to do six complex tasks which were designed to simulate the same techniques used during laparoscopy to cholecystectomy. Comparison of performance of the two groups (group 1 trained using VR and the second group which served as the control group) was analyzed using statistical analysis; the authors concluded that surgeons who received the VR training performed laparoscopic cholecystectomy significantly faster than those in the control group; moreover, the VR trained group showed greater improvement in their error and economy of movement.

Aggarwal et al., [40] performed a survey about the attitude and willingness of the senior and junior surgeons toward the VR laparoscopies simulators. They interacted with 245 consultants and their corresponding specialist registrar (SpR) to fill in their questionnaire and express their opinion toward this emerging field of technology. Among these surgeons, 81 percent agreed that VR simulators can help the training community to have a better training in laparoscopic techniques. 90 percent of junior SpRs and 67 percent of senior SpRs were interested in using VR in their training system.

#### 2.4 Heart surgery

Another important surgical area where virtual environments were used was heart surgery. Some of these environments aided in training surgeons for minimally invasive cardiac surgical procedures; others were used for planning cardiac surgical processes.

Peters [41] discussed the development of a VR environment to assist surgeons in minimally invasive cardiac surgery. This system uses VR techniques to integrate anatomical

models, intra-operative imaging and models of magnetically tracked surgical tools. Preoperative anatomical models were constructed using an MRI scanner (where various anatomical features were identified). Intra-operative images were acquired using a 2D-trans-esophageal echocardiography (TEE) transducer. An intra-operative VR environment was developed with the pre-operative cardiac models. The virtual environment was complemented with the tracking surgical instruments in the virtual environment. The researchers also described preliminary efforts by using their system for guidance and planning of typical intracardiac procedures. Two of the applications highlighted include a Mitra valve implantation procedure, an Atrial Septal defect (ASD) closure intervention and radio-frequency ablation procedure. The virtual space used in this approach reflected the actual surgical space during heart surgery and was based on images of the heart before and during the surgery itself. Using this environment, the surgeon had access to a global 3D view of the heart, a detailed view of the surgical target, and information on the position of the surgical instruments. The results from this study indicated that the model-to-subject registration technique used to augment the intra-procedure images with the pre-operative models ensures a ~5 mm accuracy for the cardiac structures which were within 10 mm from the valvular region (this was acceptable from a surgical perspective). The accuracy of the VR enhanced ultrasound guidance system was also assessed from a surgeon's point of view. The authors concluded that the use of the VR system helped in the performance of the intra-cardiac surgical process. This approach also supported target visualization, planning and guidance for the surgery.

Par et al., [42] a virtual fixture technique was developed to support robotic cardiac catheter manipulation tasks. The authors classified the haptic virtual fixtures into guidance virtual fixtures and forbidden-region virtual fixtures. The concept of virtual fixtures was used in various fields, including rehabilitation exercise assist system and minimally invasive surgery [43]. Images from X-Ray fluoroscopy were used along with vision-assisted control methods based on the forbidden-region virtual fixture (FRVF) technique were used to prevent collision of

the catheter tip and the vessel wall. A master-slave robotic platform was developed for this approach. The master handles provided haptic rendering to the user. An algorithm was developed to support the virtual force generation task, which fed the signal back to the user when the catheter tip penetrated the forbidden region. The overall benefit of this approach was that it could provide additional information to clinicians to safely manipulate the catheters in cardiac procedures.

In cases involving congenital heart diseases, operation planning was vital to the success of the overall surgery as it provides a better pre-operative understanding which could minimize surgical explorations. Sorensen et al., [44] a simulation based approach was outlined to assist in pre-operative planning related to invasive cardiology. The overall emphasis of their research was to translate 2D imaging data into 3D visualization based environments which in turn could promote better understanding of cardiac morphology. The overall approach involved four major steps including data acquisition, segmentation, model generation and VR based visualization. The input data were MRI scans, which provided fairly accurate cardiac dimensions for imaging of cardiac chamber volumes and myocardial masses. Segmentation involved (in general) the classification of pixels into regions. Algorithms were developed to perform segmentation of the 3D cardiac MRI data sets. The VR environment included both hardware and software. These included dual displays in an 'L' configuration, shutter glasses (Crystal Eyes 3D eyewear for stereo viewing), tracking devices (from Polhemus) and an advanced computing processor (Onyx2). For interactions with the VR environment, a scene interaction library was used. A user could perform basic navigation including zooming, rotation, etc by simply moving the hands when holding and activating the stylus trackers.

Berlage et al., [45] outlined the first part in the development of a simulation and planning system for minimally invasive coronary artery bypass surgery. As the authors indicate, though minimally invasive surgical procedures were beneficial to the patients, they were a number of



difficulties associated with it. The objective was to create a simulation system that helps the surgeon plan and choose appropriate incisions which can also provide an optimal operation range of the instruments. Their long term interest was to use this system and approach for education and training. The simulation was based on multimodal image data registered to a virtual heart model. The main objective to develop such a system was to help the surgical community plan for such operations and thereby further reduce the complications associated with minimally invasive surgeries. The approach addressed the three primary problems associated with such surgeries: (i) port or incision identification, (ii) training and (iii) spatial orientation for proper and precise motion (during surgical procedures). The advantage of using such a virtual model over a 3D reconstruction was that it incorporated dynamic predictions of the heart motion. The heart model with the ribs and the chest surface had an accompanying set of virtual instruments and a simulated endoscope. These virtual instruments could be moved using an electromagnetic tracker, while the endoscope could be controlled using menu operations or voice activated commands. The endoscope view was simulated by a virtual camera in the scene. This enabled assessment of the spatial accessibility of the operation field in a VR environment. The primary outcomes of this earlier work demonstrated that simulation of minimally invasive cardiac procedures was possible using a virtual heart model (and VR technology).

Thanh et al., [46] discussed the use of a virtual environment with haptic interface to train new surgeons in heart myoblast processes. In the heart myoblast process, myoblast cells were injected into the heart to restore muscular function. The use of a robotic surgical assistant system helped in the process, but the lack of experience on the part of the surgeon caused damage to the surrounding tissue. In [46], a haptic enhanced virtual environment was outlined which can be used to train surgeons to improve hand-eye coordination as well as used to enhance a surgeon's teleoperation robot-assisted surgery skills. The outcomes of experiments on human performance were also discussed (when using and interacting with haptic feedback based virtual

environments). This study involved 10 operators repeating the needle insertion and injection ten times; the experiments demonstrated a training success rate of 84.00% and 75% respectively for static and dynamic motion heart scenarios; it was observed that some operators improved their times by 300 % when compared to the training using a static heart scenario.

Ren et al., [47] the use of 3D virtual fixtures to augment the visual guidance system with haptic feedback during minimally invasive heart surgery was detailed. While minimally invasive surgery (MIS) had many benefits over conventional procedures, it had several drawbacks. These include restricted maneuverability, limited field-of-view, and the lack of tactile feedback. The virtual fixtures can be used to provide the surgeon with more helpful guidance by constraining the surgeon's hand motions thereby protecting sensitive structures. VFs can be described as computer-generated forces that were reflected back to the operator as feedback during a surgical procedure. Two categories of VFs were outlined: forbidden-region virtual fixtures (FRVFs), and guidance virtual fixtures (GVFs). GVFs were used to guide the surgeon toward a target and can help an operator or robot to move along desired paths or organ surfaces, FRVFs restricted access to "forbidden" regions and were viewed as hard constraints which prevented an operator/robot from entering forbidden regions. The proposed dynamic virtual fixtures were applicable to many intracardiac procedures, including atrial-septal defect repair, valve repair and replacement, and ablation for atrial fibrillation (among others).

Yu et al., [48] an interactive simulator was described in real-time and tactile catheter navigation. The authors acknowledged that realistic simulation of cardiac intervention was still very challenging because of the complex and dynamic nature of the cardiac intervention process. The myocardium was represented using tetrahedral volumetric mesh which presents the main anatomical information, including details of valve and valve gaps. Different parts of the adjacent tissues were identified by different colors. The cardiac intervention was modeled as a two-body interactive problem; the catheter and the heart were independently modeled. While the heart was

modeled using mesh geometry, the catheter was modeled by a number of cylinder rods. The overall simulation environment comprised of a stereographic visual panel, a tactile catheter device and a virtual patient model. The 3 D tracking system used was the Polhemus PATRIOT stylus system. The simulator worked with on a Windows PC workstation with an NVIDIA graphics card; the graphical environment was created using OpenGL Graphic Library.

## 2.5. Other Surgery

VR based environments and tools were used in a wide range of other surgical domains. A review of environments from various surgical domains was discussed in this section.

Montgomery et al [49] discussed the use of Virtual Environments for Reconstructive Surgery (VERS) from the removal of a soft-tissue tumor; in this paper, the authors review the case of a 17 year-old boy with a severe facial defect arising from the removal of a soft-tissue tumor. They implemented a number of virtual tools and interfaces, including (a) selecting surgical tools to pick up an object in the environment (b) a marker tool to ‘mark’ on the surface of a target object, and (c) a lighting tool for more precise localization of lighting. They visualized the high-resolution data of the patient and produced color prints from various views; subsequently, the VERS system was used to interact with these data. The virtual system (VERS) can be used to quickly interact with the meshes representing the skull and soft tissue. With the available data, a surgeon was able to virtually cut the bone and examine the fit of a new bone to be placed into the (target) defect area. In the end the system which they created integrates 3D reconstruction, visualization, quantification and manipulation of multi-model patient data for the purpose of surgical planning. The system was found to be instrumental in the preparation and correction of severe craniofacial defect and was well received by the surgical community.

Suzuki et al., [50] the development of a surgery planning system using VR techniques was detailed for the incision of skin and organs. A force feedback device was developed which

responds to the pressure of the virtual operator's hand. In this study, they used reconstructed sphere-filled model of the liver and the surface of the liver model was wrapped with the surface image of autopsied liver tissue (using a texture mapping method to obtain a realistic appearance for the simulation). A haptic device was developed with force and motion control manipulators which were attached to the thumb, forefinger and the middle finger of the operator; the forces coming into play was calculated using a 'sphere-filled model' proposed by the authors. The performance of the virtual model was more accurate and the model generated the images virtually without any delay when using a surgical tool. The force feedback device performed surgical maneuvers with the sense of touch more accurately in the real time environment. The developed system was used for surgeries in the abdominal region.

Cecil et al., [5] An information model based framework for the development of microsurgery virtual environment was outlined. Microsurgery involves the sewing together of blood vessels, nerves or tendons, or blood vessels to correct an injury, or congenital defects. In surgical operations, small arteries and veins were reconnected within the operated area; the surgeon uses an extremely thin thread and completes the surgery with the help of microscopes. Cecil et al., [5] The information framework proposed involved the use of information models to capture the complex relationships within a surgical process and a haptic interface based environment to train medical residents in microsurgery. The information model was built using an engineering Enterprise Modeling Language (eEML) where the functional relationships and temporal precedence constraints were explicitly modeled; this model was developed in close interaction with expert micro surgeons. The role of such an information intensive process model in providing a foundation to build VR based simulation environments to train surgeons was also discussed.

Both semi and fully immersive systems require the use of VR sensors such as motion trackers (such as the Flock of Birds® unit from Ascension Technologies) and immersion

supporting devices such as interactive ‘wands (such as WorkWand®). Stereo eyewear was typically used to view active stereo in semi or fully immersive environments.

The original CAVE® (Cave Automatic Virtual Environment) was developed in 1992 at the University of Illinois and was an example of a fully immersive system; it had four projection screens which allow users to immerse themselves using sensors and stereovision eye wear. Johansson and Ynnerman [51] discussed different immersive visual interface’s ability to support engineering designers during product design. An analysis was performed to find induced errors in a mechanical product using different display solutions: an immersive workbench (67” screen, Barco projector, CrystalEyes®, head tracking, magic wand® and keyboard), a desktop-VR (21” monitor, CrystalEyes®, keyboard and mouse), and a desktop system (21” monitor, keyboard and mouse). Based on the results, the authors concluded that the immersive displays supported the product designers the best.

## 2.6 Haptic Devices and Tracking Technology

In this section, discussed some of the haptic and tracking technology from various vendors.

Coutee et al., [52] described the development of an application known as HIDRA (Haptic Integrated Dis/Reassembly Analysis), which provides a haptic feedback through a haptic interface (PHANTOM™). A haptic interface was a peripheral device that measures the forces applied by the user’s avatar (user’s representation in a virtual environment) in 3D space and exerts those forces of the actual user, thereby providing haptic feedback. Providing haptic feedback allowed the user to feel the friction, wiggle and touch. The HIDRA test bed was a two-loop simulation, which provided haptic and graphic rendering. When a user touched or applied force to the virtual object through a PHANTOM™, HIDRA monitored and responded to such forces. The haptic simulation loop of HIDRA continuously calculated the user’s fingertip position and checks for

collision of fingertip and the virtual object. If a collision was detected, then equivalent force was exerted on the fingertip through the PHANTOM™.

Springer and Ferrier [53] described the design of a multi-finger force-reflecting haptic interface device for teleoperational grasping. The entire system consisted of a master or haptic interface and a slave or a remote manipulator. The Master tracked the positions of the user and used these positions to manipulate the motion of the slave. The hand mounted mechanism was capable of representing the fingertip position throughout a wide range of grasping motions.

Balijepalli and Kesavdas [54] described the design, implementation and evaluation of a haptic simulator based on a force model that renders precise crisp force feedback using an abrasive hand-grinding tool. Based on a force model for grinding tools, a haptic interface to machining a work piece was developed; 3D Terrain modeling and dynamic texture modification algorithm were developed to simulate the polishing or grinding process. For haptic interaction, a framework for fast and accurate collision detection was implemented by Gregory et al. [55]. The algorithms used polygonal models with real-time hybrid hierarchical representation and exploit frame-to-frame coherence for fast proximity queries.

Magnetic Levitation devices (Maglev) were a haptic technology introduced first at IBM Research and later commercialized by Butterfly Haptics under the NSF Major Research Instrumentation grant with a license from Carnegie Mellon University [56], [57]. Maglev devices utilized a different technology compared to other haptic devices; rather than using mechanical elements such as linkages, motors, cables, and bearings (found in most haptic devices), they interact with the user through a handle which was levitated by magnetic fields. The Maglev device was intended to provide interaction with a high degree of fidelity, position/force bandwidth, and position resolution, and a wider range of possible stiffness. During the past decade, many researchers have worked with haptic technology, which led to Lorentz Levitation

devices, including the IBM Magic Wrist, UBC Wrist, the UBC Maglev Joystick, and the CMU Magnetic Levitation Haptic Interface [56, 57].

Maglev haptic devices provide 6 degrees of freedom through a single moving part levitated in a magnetic field. The handle was attached to a “flotor” which floats between stators. The flotor was tracked by optical sensors to determine its position and orientation, and then this information was sent for further processing and then back to the user through the handle [56].

Gurocak et al [58] described the design and implementation of a force feedback hand master haptic device called AirGlove. In a VR environment, touch and force feedback provide realism by considering the physical properties like object rigidity, weight, friction and dynamics of the objects. The AirGlove consisted of an air jet block with 5 ports, a remote box having 6 pressure sensors and a control software. Compressed air was exhausted through the ports of the air jet block to apply thrust force on the user’s hand. AirGlove provided gravitational force feedback to the user as he or she manipulates objects in the virtual environment.

Electromagnetic tracking systems used for immersive systems require a calibrating system as they suffer degradation in accuracy due to stray electromagnetic fields in the vicinity Jayaram et al., [59]. There were two types of tracking systems: one with DC magnetic fields (Flock of Birds) and the other with AC magnetic fields (Polhemus Fastrak). A calibration system called COVE (Calibration of Virtual Environment) was used for measuring static errors of DC magnetic trackers and automatically correcting them and several interpolation techniques. The COVE calibration process consists of five steps: discretization, grid data collection, interpolation, incorporation, and evaluation.

The use of Virtual Surgical Environments (VSE) in daily practice was not yet widespread for various reasons; in Aggarwal et al., [40], some of these reasons were outlined which include lack of familiarity with VR technology, high-priced cost, poor validation of effectiveness as well

as the reluctance of surgeons and faculty to invest their time in this emerging area and technology.

## 2.7 Other Challenges

In this section, a discussion of the challenges involved in the development and evolution of virtual surgical environments will be provided.

One of the major challenges was that the creation of virtual environments was a complex time consuming process. Most of the software tools available on the market today require long periods of training to acquire and hone the software skills necessary to build such environments. In addition, very few of the haptic technology vendors provide training services to teams interested in building simulation tools. Currently, a new generation of software tools which are more user friendly (such as Unity 3D™) holds the potential to change this trend.

The design and development of virtual environments require the collaboration of medical surgeons, software specialists and engineering / science experts. While the process expertise (for a given surgical domain) lies with surgeons, this understanding of a specialized surgical process needs to be studied and modeled prior to the design of a target surgical environment Cecil et al., [3, 4]. The importance of creating information models (which was more common place in the development of engineering simulation environments) prior to the design of virtual environments will enable the implementation of more realistic and detailed simulation environments.

Another key problem is the cost involved in the acquisition of VR equipment (which varies depending on the level of immersion) including trackers / cameras, 3D eyewear, sensors and other peripherals such as controllers, joystick, wands, etc. (Which was needed to interact with the target simulation environment). While the cost of computing processors, 3 D eyewear and trackers came down significantly, the cost of fully or semi immersive technologies is still high; today, semi immersive technologies (including trackers, projectors and software) can range from



\$25,000 to \$ 200,000. Fully immersive technology (such as a CAVE system) is more expensive and can cost upwards of \$500,000.

There is also a need to conduct more comprehensive assessment of the impact of using VSEs; while several studies indicated their benefit in terms of developing surgical skill for surgeons, more studies are necessary which focus rigorously on the development/technology cost versus benefit/impact in surgical fields and medical training.

The current state of haptic technology also needs to improve if more realistic environments are to be developed. While most of the existing haptic products provide basic force feedback and modeling capabilities, there is a need to develop more advanced technology that can provide more sensitive and realistic force feedback which is crucial for this emerging application area involving training of surgeons. For example, when simulating micro surgical processes, the force feedback should be sensitive enough to be able to simulate a surgical needle piercing the wall of a blood vessel. Development of such next generation haptic technology that are more innovative and low cost is key to wider adoption of virtual surgical environments.

## 2.8 Conclusion

Initially, the Literature review had been carried out in the orthopedic surgery, heart surgery, laparoscopic surgery and other techniques on the basis of virtual environments. The potential issues and problems on these surgical methods were identified. The following chapter gives the overall design approaches and methods for creating the VR environment for the orthopedic surgery specifically Less Invasive Stabilization System.

## CHAPTER III

### VIRTUAL REALITY FRAMEWORK

#### 3.1. Introduction

This chapter outlines a virtual reality framework for the orthopedic surgical environment with a collaboration diagram and architecture of the overall surgical process design and gives a brief view of a simulated surgical process developed using Unity environment.

#### 3.2. Less Invasive Stabilization System (LISS)

The initial scope of the Virtual Environment for Orthopedic Surgery (VEOS) is limited to a process called Less Invasive Stabilization System (LISS) surgery. LISS surgery is a specific orthopedic surgery process which is used for patients with fractures of the human femur. Figure 1 shows a view of the LISS components used for such a surgery.

The LISS plating process is a complex activity involving a number of steps (Figure 2). After performing the preoperative steps and determining the proper implants for the observed fracture, the surgeon has to assemble the main components for a LISS insertion guide with the fixation bolt, stabilization bolt, an insertion sleeve, and the LISS plate. Once the units are assembled, a LISS plate is inserted between the vastus lateralis muscle and periosteum. Again the plates are properly checked for orientation and position. Fracture reduction can be obtained using

K-wire and a pull reduction instrument by aligning the broken bones back to their proper positions. This procedure is usually performed using hands or other specific reduction methods. The important step in the surgery is the insertion of the LISS plate into its proper position under the skin and in proper contact with the femur. The next and most important step is to screw the LISS plate to the bone using locking and diaphyseal surgical screws. When the LISS plate is placed properly over the femur, the insertion guide can be detached from the LISS plate (Figure 1).

The VEOS is being developed from three different perspectives: a haptic-based environment, a non-haptic environment, and a semi-immersive environment. These virtual environments can be used in different modules such as training or in evaluating the performance of surgery students. The overall computer architecture for the VEOS is as shown in Figure 3.



Figure 1. View of an inserted LISS plate (as shown in [60])

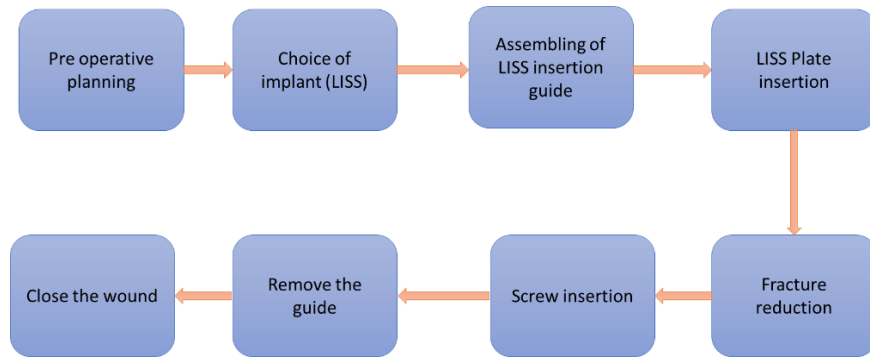


Figure 2. The key steps involved in LISS plating surgery

### 3.3. Architecture of the Virtual Training Environments

The overall architecture and functional responsibilities of the various modules is discussed in this section.

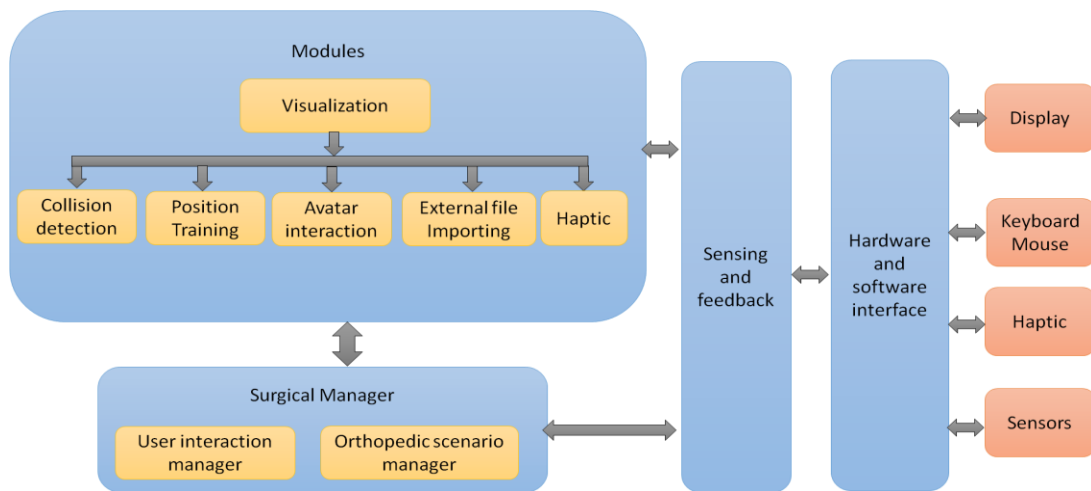


Figure 3. Architecture of Virtual environment for orthopedic surgery (VEOS)

#### 3.3.1 Visualization Module

The purpose of the visualization module is to display different stages of the simulation from the initial position of LISS parts to the fracture reduction of the femur using various cannulated screws and positioning them properly during the surgery. In this module, the CAD

models of the surgical instruments, the femur and all other objects are interfaced. It also provides the information and the sequence of the surgical process.

### 3.3.2 Collision Detection Module

A collision detection module contains a collision detection algorithm, which is used to find the collision between the surgical tool and the bone. The algorithm provides a force feedback/color indication when the surgeon moves the tool and collides & disorients the bone. This algorithm prevents the penetration of bone and keeps the hand-eye coordination of the surgeon in order.

### 3.3.3 Position training module

The position training module helps residents to predict the exact location of the surgical tools in the non-immersive environment. The position training module measures the distance between surgical implants and bone and also the lower and the upper proximity of the LISS plate. In this module, the indicator ensures the exact place of the implant by turning on the red color. In case the surgeon placed the implant in the wrong direction it indicates the changes in the form of green and yellow light. Thus, the surgeon can realize the mistakes in the placement of the implant and correct its position.

Placing the plate between the vastus lateralis muscle and the periosteum is difficult. Orienting and positioning the plate is very significant in the LISS surgery process. Further, a novice surgeon requires practice before operating in the real surgical environment in order to reduce the fracture of femur. In this surgical simulation, a position training module is developed to indicate the misplacements of the plate during the surgical process.

This is a very useful tool that the experienced surgeons can utilize to show the residents the proper directional placing of the LISS plate. The simulation environment incorporates a color

indicator module and provides the feedback to the user by changing to a specific color depending on the orientation of the plate. This feedback is mainly focused on the upper and lower proximity of the femur.

Two different methods were developed in the module to indicate the position of the plates at the upper and lower ends of the femur. Each indication method has the same features to show the changes in the orientation of the plate. These methods have red, yellow, and green indicator lights to indicate the changes in the environment whenever there is a misplacement of the plate. In order to maintain accuracy in the system, each indicator in the module is designed with varying dimensions. These dimensions are the primary resource for the color indicator to post the feedback for the resident.

The light changes to red when the surgeon places the plate with proper orientation, in correct position, and with accurate placement. The light turns yellow when the surgeon slightly changes/adjusts the plate. The yellow light continues until the surgeon places the plate in the correct orientation. The green light shows when the surgeon exceeds the accepted boundary and places the plate in some random directions; the light acts as an emergency tool to warn the surgeon. The dimension maintained for the red color indicator is a 0.75cm diameter region, 1cm diameter for the yellow color indicator, and the green color indicator has a 3cm diameter region. A detailed flowchart is shown in Figure 4.

The color changes are used as an alert system for the surgeon to understand the complexity, difficulties, and the importance of the LISS surgery process. The only allowable dimension is 0.75cm, and is exhibited by a circular indicator in the color of red. While the only allowable dimension is 0.75cm, 1 cm is very close. That is, if a yellow light is indicated the resident will know that he or she is close to reaching the ideal dimension. A green light indicates that the resident is far from being within the allowable dimension. The developed position

training module provides quick response and feedback through the indicators and helps the residents learn quickly and efficiently.

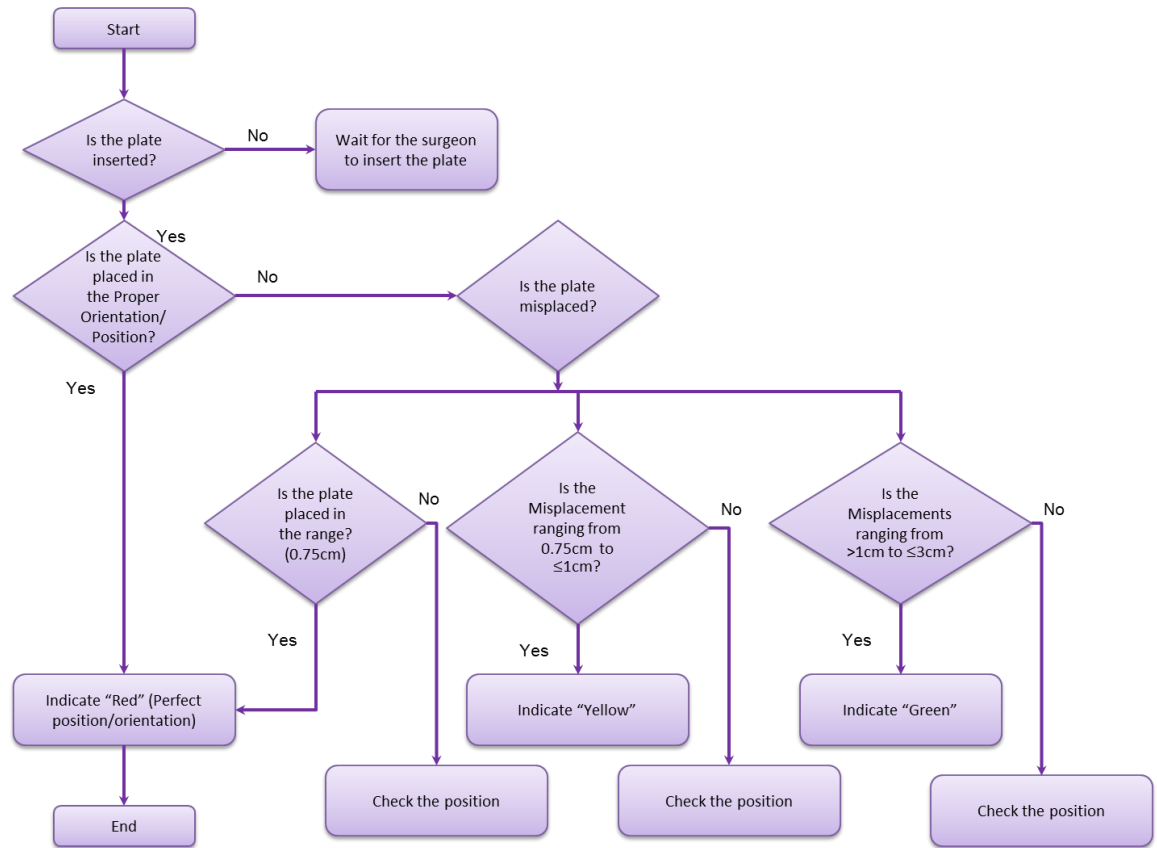


Figure 4. Flow chart for position training

### 3.3.4 Avatar interaction module

An Avatar module is created to assist the user to interact with simulation environment. The user can empathize and develop the learning skills of the orthopedic surgery steps with the use of the Avatar. The Avatar model developed with audio features enhances the learning capability of the user.

In the virtual surgery, there are many parts and steps that need to be explained to the user. Reading from the screen would distract the user from watching a certain step or examining a part, instead a user can simply listen to the description or step as they work through the virtual surgery

through an audio arrangement. To make the simulation more realistic, an Avatar is added into the scene to read this description and steps.

While there is audio playing in the virtual surgical environment, the mouth of the Avatar begins to move simulating someone talking to the user. While there is no audio being played, the Avatar still moves minimally so that the Avatar still seems life like.

The user can click on any surgical part on the table, and the Avatar tells the user the name of the equipment as well as the purpose it serves in the surgical process. When the user is watching or practicing on the virtual surgery, the Avatar provides explanations of the current step that is being performed. Any of these actions can be repeated so that the user can listen to the Avatar explain the descriptions or steps until the user feels comfortable.

This feature helps the user acclimate the process in a much better way. While they are in other medical training, there is a doctor there explaining the function of the medical tools and walking the trainees through the medical process. Since the user is used to this method of instruction, the user will feel more comfortable in the virtual surgery environment.

### 3.3.5 External File Module

In this module the residents can decide the path of the surgery using an external coordinate file. It eventually supports the surgeon/resident to plan the path in advance. This module controls the movement of the surgical tools and their positions based on the given coordinates.

In the virtual surgery simulation, there is a default starting position for each object in the scene. If a user desires to change these starting positions of the objects in the scene, they can utilize the external coordinate file. To use this feature, the user creates a text document constructed of rows and columns and using an external code in Unity, the scene will import the



document and initialize with new starting positions. This feature allows an instructor or medical doctor to customize the position and orientation of any object in the virtual surgery environment.

The instructor is able to choose any object in the scene and change the starting position and rotation in the x y z coordinate system. The instructor can also choose to move a part slowly to the position desired. For this, the instructor simply inputs the time in seconds that they want the part to move, and the part will move from the starting position to the position indicated by the instructor in the time provided.

This is a useful feature as it allows instructors or medical doctor to construct the virtual surgery environment as they see fit. This is also useful because the instructor can mix up the parts or move the scene around to make the simulation different for each student, or simply test one's knowledge of the process and the tools used.

### 3.3.6 Haptic Module

The Haptic module is created to provide the real-time surgical force feedback during the surgery. The Geomagic touch haptic device is used in this module. The device is programmed for the LISS surgery environment. This external device is used to interact with the simulation developed in the Unity environment.

The haptic device utilized is called the Geomagic Touch. A custom Unity plugin is used to relay information between the Unity and the device. The plugin utilizes the libraries in the OpenHaptic Toolkit and Geomagic's Phantom driver library. The plugin also allows us to change the mass, static and dynamic friction, stiffness, and other properties by changing variables in a C# script attached to the object of interest. By being able to treat the haptic device as just another GameObject in Unity it is integrated with the existing simulation of the surgery.

With haptic feedback module, the surgery environment emulates the physical forces experienced during a real surgery. Some of these crucial feedback forces include placing the LISS plate against a distal femur, feeling the LISS plate position, measuring the required depths with Kirschner wires, and drilling the bone in a stable manner.

The haptic device consists of two buttons on the stylus. The surgeon picks up the objects in the surgery environment by making virtual contact with the haptic object and the object to be moved by pressing and holding the button on the stylus, and then simply moving the stylus to the desired position to obtain the desired position of the object in the surgery environment.

### 3.3.7 Surgical Manager

This module coordinates interactions between the various modules and the user. Surgical manager contains different surgical scenarios. In this module, the step by step procedure of LISS plating surgery is listed as one scenario.

### 3.3.8 User interaction manager

The interaction between the user and computer is synchronized with the use of user interaction Manager. It enables the user to identify and load different orthopedic scenarios for simulation as well as communicate outcomes of a given simulation session.

- Cad files loaded
- Conversion of cad files from different formats (.obj, etc)
- Splitting the windows to be able to visualize with top view, side view and other views
- Conveying problems encountered during simulation to the user
- User can also ask questions about how to interact with the simulation environment

### 3.3.9 Orthopedic scenario manager

The scenario manager refers the modules and step by step processes involved in the VEOS. The VEOS scenario manager provides the LISS assembly, insertion of LISS plate, fracture reduction, screw insertion, and guide removal in a sequence. It also provides the control modules where user can interact the surgical environment using some external devices.

### 3.3.10 Sensing and Feedback

This module assists the user to obtain the feedback from the simulation environment based on the external hardware interactions. The external devices control the simulation and modify the movements of the surgical tools as well as positions of the tools. The user is able to see the changes in display or gets the force feedback from the haptic devices.

### 3.3.11 Hardware and Software Interface

The external devices such as keyboard, mouse, and haptic can be utilized to provide some extra inputs. They are assigned to rotating the camera, picking up/ dropping a surgical tool, changing/tuning the orientation of the surgical tool, placing the implants in their proper positions, and a couple of other surgical functions needed during the surgery.

## 3.4. Collaboration diagram

The surgery simulator manager is the primary segment where the information exchanges among the various modules happen. The menu control module allows selecting various modules in the collaborative model. Automatic and manual are two common modules which are mainly used for user information exchange. Automatic module initiates the simulation without any user interaction. It provides the surgical simulation in a step by step manner. The manual module runs the simulation based upon the user inputs. The user input is obtained from the keyboard control module, mouse control module, external file control module, and haptic control module. There

are some more advanced modules involved in this collaboration diagram. They are display control module, network control module, collision control module, and avatar control module. The network control module exchanges the information from the user to simulator at different locations. Display control module updates the frame rate based on the graphics and enhances the simulation speed. The avatar control module enables the effective interaction between user and simulation using avatars.

The collaboration diagrams (Figure.5) are used to show how the objects are interacting in the surgical environment. The user has an option to select three different modes of operation in the manual module. The three modes are triggered by *getKeyboard*, *getMouse* and *getHaptic* user defined functions. These options are activated based on the user selections on the surgery simulator manager. Once the functions are selected, it sends the user feedback to the appropriate control environment. These environments feed the information to the next level of interaction. There are three different options provided for interacting objects in the manual module: *select*, *move* and *generate* a sequence. The *select object* class assists to select particular components and move object class according to the surgical sequences. The surgical sequence selector generates the desired positions of each component on the simulation. The user also gets the movements of the components in the simulation by the display function, which gives the on time feedback of the part movements.

In addition to the user interaction, the simulation also has an alert system to monitor the progress of the surgery. These alerts can be obtained from the Avatar and the collision detection module. The Avatar module is a user friendly method to assist the user to learn the surgical steps and assembly steps based on a graphical Avatar audio interaction, whereas the collision detection module shows the collision between the femur and LISS plate during the surgical steps by color indication. External text file interaction is utilized for path planning of objects in the simulation.

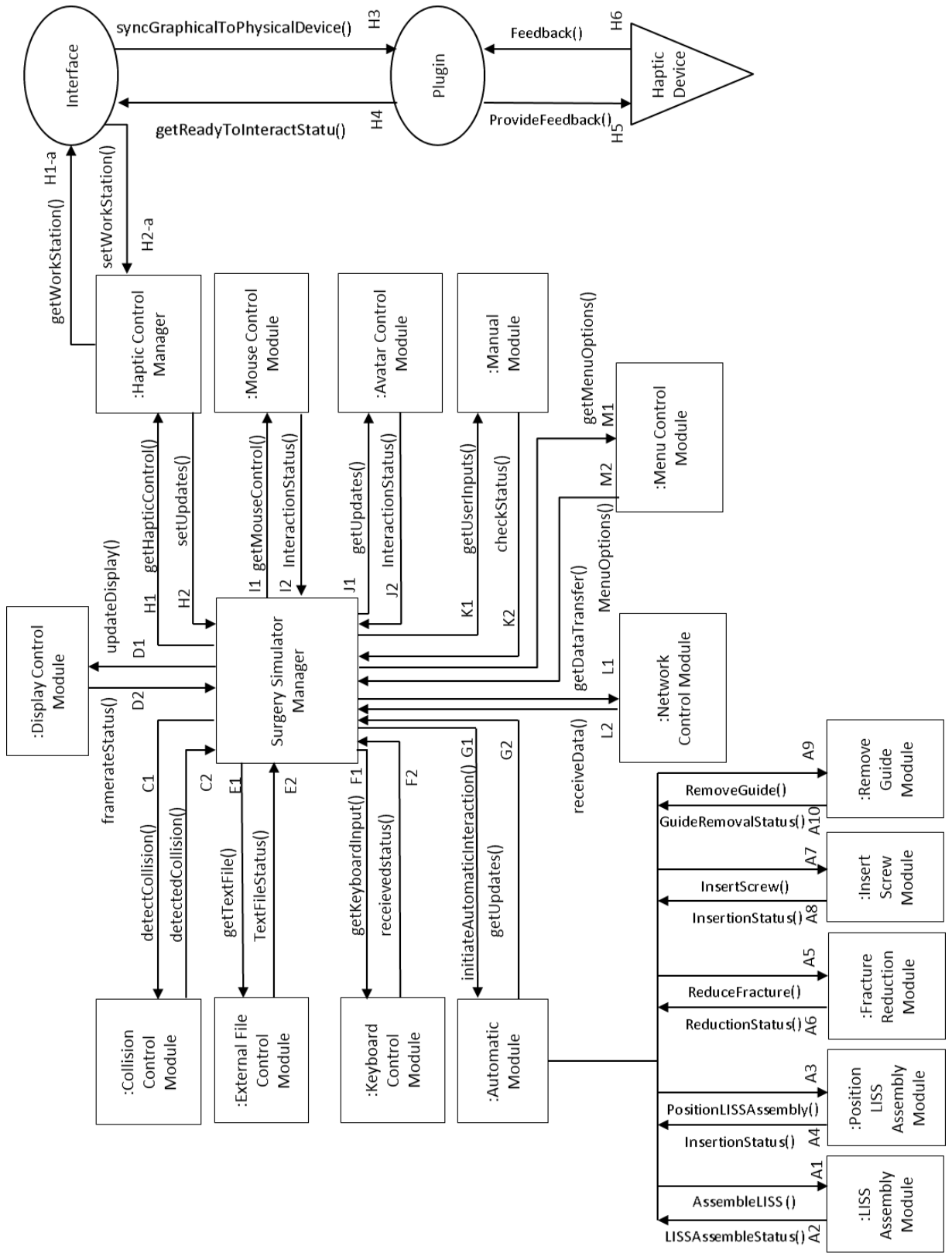


Figure 5. Collaboration diagram

Haptic control manager is a software component which manages the haptic plugin and interfaces the haptic device with the simulation environment. The getworkstation function detects the position and size for the haptic interaction and defines the workstation using setworkstation function.

### 3.4.1 Class Diagram

Class diagram provides the information of the developed system by describing the system structure on the basis of the system's classes, attributes associated with the classes, essential operational functions, and the relationship between the objects [61].

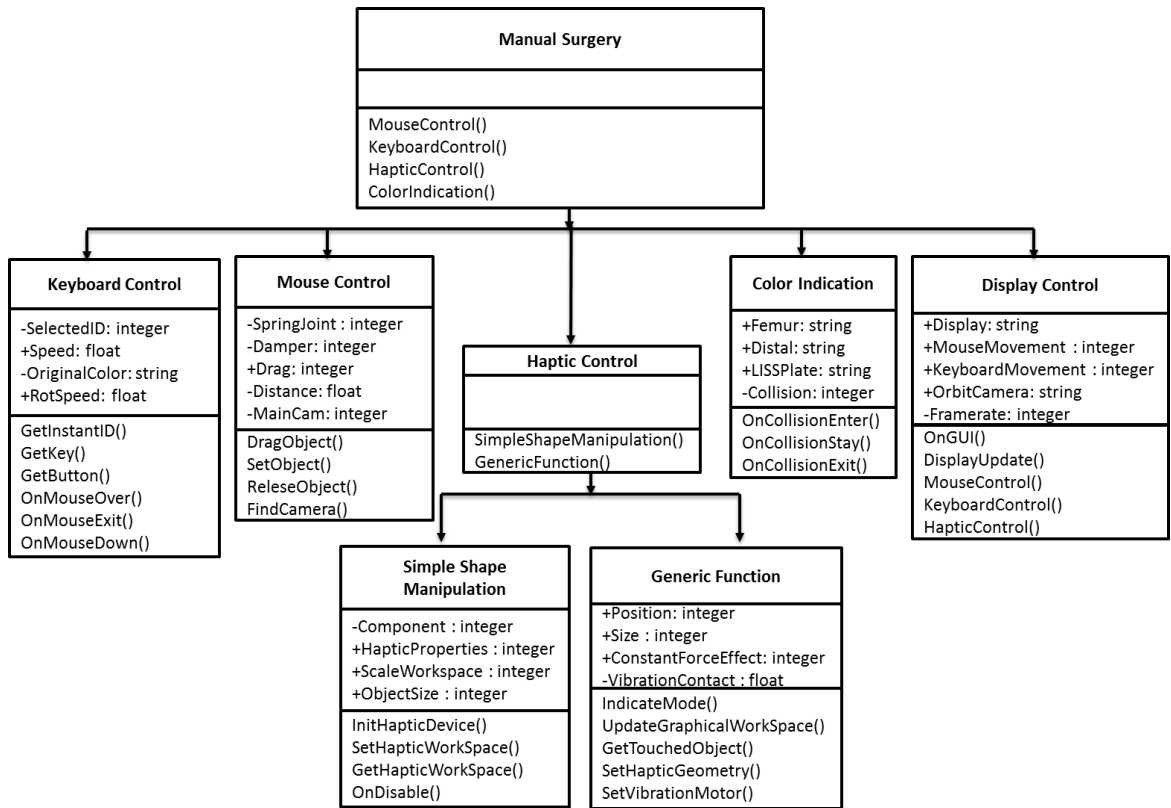


Figure 6. Exploded view of manual surgery simulation class diagram

The manual surgery simulation is the training module where the surgeon uses the control methods for the user interaction. A detailed class diagram (Figure 6) for the manual surgery simulation presents the overview of the various classes and their functions.

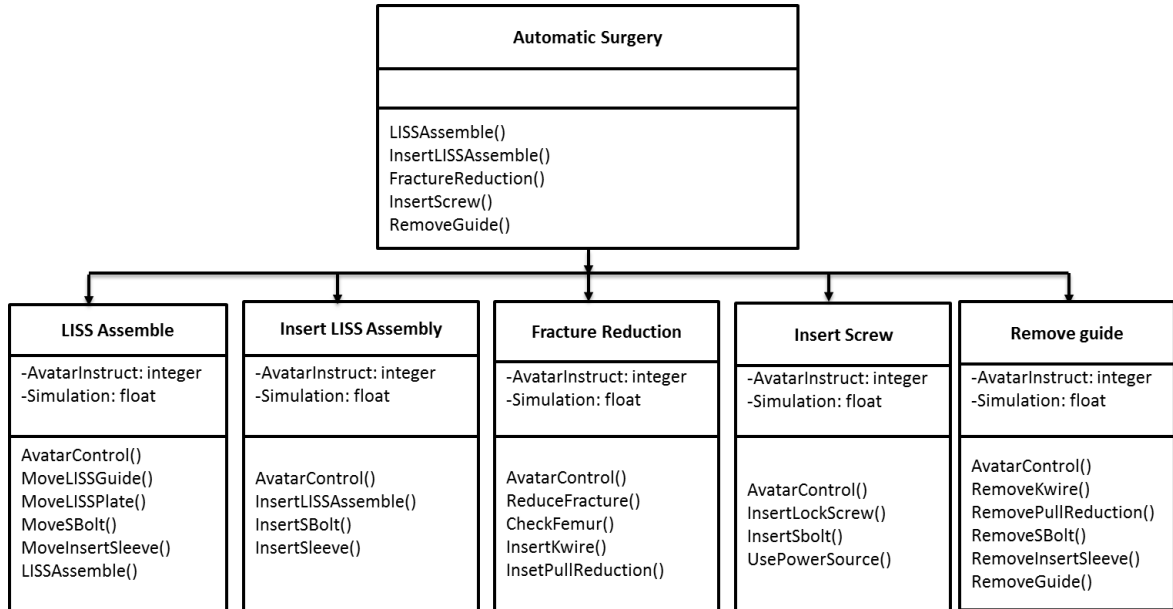


Figure 7. Exploded view of automatic surgery simulation class diagram

Automatic surgery simulation class diagram (Figure 7) specifies the classes involved in the system generated training module. Essential variables called for in the automatic mode are the instruction and simulation variables. The remaining functions in each class set the assembly, insertion, and removal of surgical tools for surgical steps. The following sections give a detailed explanation of the class diagrams from Figures 6 and 7.

### 3.4.2 Keyboard control

The keyboard control function is developed using javascript and it uses the update function of moving objects in the simulation environment. Private and public variables are defined for this function and *SelectID* class is utilized for a particular component update. Mouse functions *OnMouseOver* (), *OnMouseExit*, and *OnMouseDown* () are used to select objects in the

Unity environment. Color rendering options are initiated on the materials when the objects are selected. For moving the object, *GetKey ()* and *GetButton ()* functions utilizes six different key parameters and these are assigned to the update function. It enables the transform class for translation and rotation of objects with a default variable speed limit of 0.5Sec. Movements of the objects are visible at this speed limit. The keys are assigned in a way that the user can move the objects in all directions in the environment.

In the original simulation the user also has an option to pause and continue the play option using the keys “P” and “Space”. These types of changes are done by utilizing the time scale classes in any update function. A summary of the main functions in this module is provided below.

*OnMouseOver ()*: Detects the objects using the mouse cursor

*OnMouseExit ()*: Exits the object selected

*OnMouseDown ()*: Used to select objects

*GetKey ()*, *GetButton ()*: Used for keyboard key assignment

### 3.4.3 Mouse control

Class diagrams are used for the mouse control option in VEOS and the movements are frequently updated with the camera position. The private variables of mass, spring, damper, drag, angular drag, and distance are specified for control functions. A user defined function is prepared to pick and move the objects using *springjoint* classes. *Springjoint* defines the body mass, angular velocity and mouse orientation update for moving objects on the desired location. It also considers the center of mass of each object facing changes in position. Camera is also updated using user defined function *FindCamera()* for frequent visibility of the object position during motion. Mouse control function is activated by selecting certain objects in the environment and drags the object to final position using simple left mouse buttons. Material rendering option



indicates the picked object by changing the material color. The function *SetObject ()* sets the current location and the final location of the object. The *DragObject()* function picks the objects in one location and places it in a different location. The *ReleaseObject ()* function releases the set objects during the dragging function. A summary of the main functions in this module is provided below.

*SetObject ()*: Sets the location of objects

*ReleaseObject ()*: Releases the object in specified location

*DragObject ()*: Picks object from different locations

#### 3.4.4 Haptic control

Haptic control function uses the custom Unity plugin to relay information between the Unity and the (Geomagic Phantom Omni Haptic) device. The plugin utilizes the libraries in the OpenHaptic Toolkit and Geomagic's Phantom driver library. The plugin also allows to change the mass, static and dynamic friction, stiffness, and other properties by changing variables in a C# script attached to the objects.

The plugin is implemented by multiple scripts and it is used for relaying the Haptic's position and force feedback in simulation. The Plugin Import loads the functions from the Unity-Haptic plugin and it can be accessed by other C# scripts within VEOS. A generic function provides abstracted methods for the setup of the Haptic device and its properties. Simple shape manipulation scripts utilize the methods of the generic function classes in order to set up the Haptic device. It also communicates Haptic's position, objects touched, and so on to the update function and the force feedback is the result sent back to the Haptic device.

The Haptic Control class is an aggregate of the Simple Shape Manipulation and Generic Functions class. The Generic Functions class, which is dependent on the haptic plugin, provides the essential haptic functions that can be used by other scripts. Some of the essential haptic

functions are *UpdateGraphicalWorkspace ()*, *GetTouchedObject ()*, etc. There are also essential variables that are in the Generic Functions class that affect the ultimate haptic force feedback and these variables are passed through the Haptic Properties class. Through these variables, the properties of a haptic object and its interactions are easily changed, such as the rigidity of the object can be set on a scale of 0.0 to 1.0. The abstraction of the variables and functions makes it quick and easy to setup the haptic in the simulation and also makes it to change the entire haptic interaction with an object in the simulation. A summary of the main functions in this module is provided below.

*InitHapticDevice ()*: Initiates haptic device functions

*SetHapticWorkSpace ()*: Initiates the working space of the haptic stylus probe

*GetHapticWorkSpace ()*: Provides the update of the work space of the haptic

*UpdateGraphicalWorkSpace ()*: Updates the simulation environment based haptic feedback

*GetTouchedObject ()*: Initiates the object selection using haptic stylus probe

#### 3.4.5 Avatar control

In the VEOS, the Avatar movement is controlled by two components. The first component is an Animator Controller that is constructed in the Unity Engine made up of the individual animations of the Avatar. The second component is an external control function that activates the transitions in the Animator Controller between the different animations. The Figure.8 shows the Animator Controller for the Avatar in the Simulation.

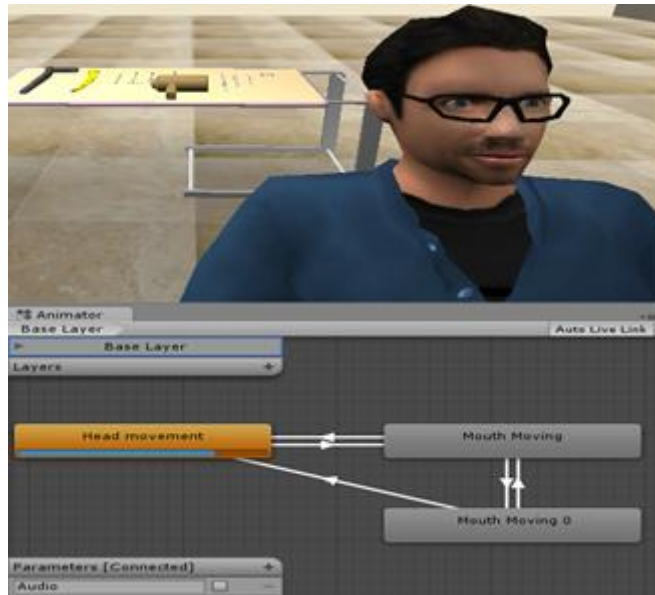


Figure 8. Animator Controller for the Avatar in the Simulation

The ‘Head Movement’ animation is the default state of this animator controller. That is, the avatar will remain at this animation until the transition is activated. From Figure.9, A Boolean parameter named ‘Audio’ was created to control these transitions. This Boolean parameter is initialized as false and is changed by the external Java Script. When the user activates any audio in the scene, the external script switches the ‘Audio’ parameter to true, which activates the transition to the ‘Mouth Moving’ animation. The Avatar remains in this state while the ‘Audio’ parameter is still true. Once the audio in the scene is finished playing, the external Java Script changes the ‘Audio’ parameter back to false and the avatar returns to the default animation.

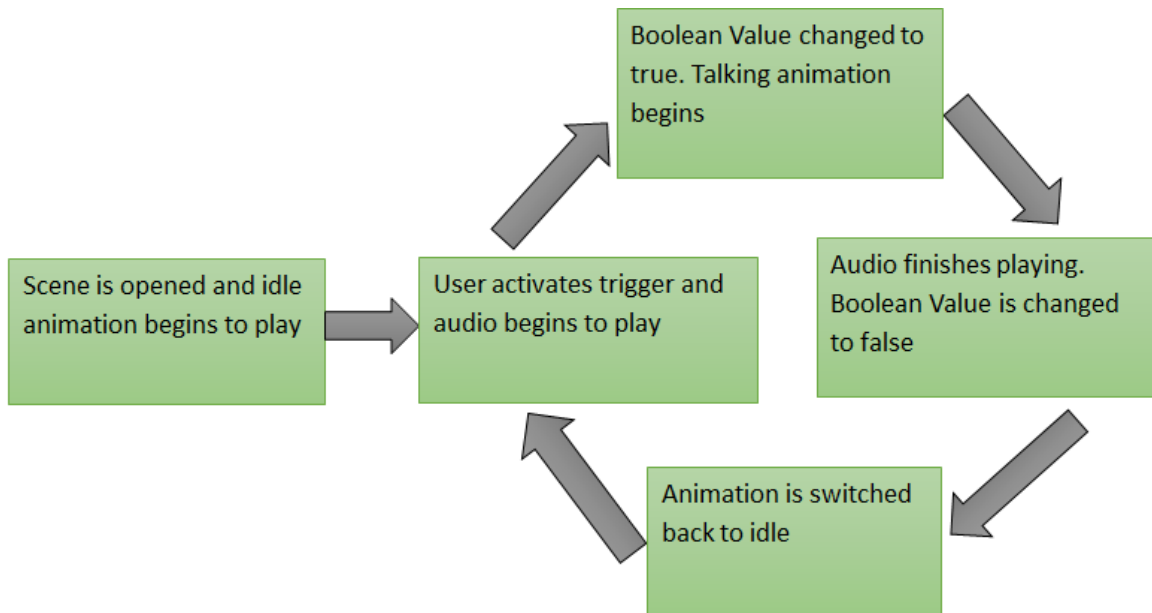


Figure 9. Animation flow diagram

A summary of the main functions in this module is provided below.

*ActivateAvatar ()*: Activates the avatar for interaction

*GetKeyDown ()*: Gets the user input for further updates

*PlayAudio ()*: Initiates the audio feedback

#### 3.4.6 External file interaction

The External File feature is a C# script that takes advantage of Unity's *Keyframe* class, and Animation Curve class. *Keyframes* are a simple one to one function that merely returns a value *y* for a given number *x* (time). An array of *Keyframes* can be created to represent the movements of an object (*y*) over time (*x*). The Animation Curve requires the parameters of an array of *Keyframes* and time. The Animation Curve's *Evaluate ()* method takes in a number *x* which is the time, and it will return the return value *y* from the appropriate *Keyframe* object that was assigned the same *x* time. Using six of these Animation Curves (with six *Keyframe* arrays), the movements (position and rotation) of an object can be represented in Unity throughout time.

There is a method called *makeCoordsToKeys ()*, which converts a comma separated text file that contains the name of the object, position ( $x, y, z$ ), and rotation ( $xI, yI, zI$ ), and the time at which the object needs to have the given position and rotation, to the six arrays of *Keyframes* and saves it in an Animation Curve object. Each line can only have one object and its required information. The user can provide this comma separated value text file with multiple objects (they do NOT need to be in any order; just need the required formatting), and another script called *SeparateTextFiles* separates and creates new text files that only houses one object and all its positions, rotations, and times.

The *ExternalFile* script then calls the *makeCoordsToKeys ()* method on each of the newly created text files that were created with the *SeparateTextFiles* script. The *makeCoordsToKeys ()* method converts the text files into the six arrays of *Keyframes*, and saves it to an Animation Curve. When the simulation is running, Unity has a clock that increments on every frame, and this changing time is the input for the Animation Curve's *Evaluate ()* method.

These scripts and their methods essentially read in paths from the text file to multiple objects, and then these paths are executed in the simulation for their respective objects. The speed of the objects' paths is linear and determined by the difference between the two points in position and rotation, and the time. The following figure provides a sample external file coordinates of multiple objects. A summary of the main functions in this module is provided below.

*makeCoordsToKeys ()*: Calls the coordinate text file

*Evaluate ()*: Evaluates the animation curve

```

//      Object Name;      Time;      Position(X,Y,Z);      Rotation(X,Y,Z),
LISS Guide;      5;      2;2;2;      0;0;0;
LISS Guide;      10;      10;10;10;      90;0;0;
Fixation Bolt;      5;      1;2;3;      0;90;0;
Fixation Bolt;      10;      2;10;10;      90;0;90;
Insertion Sleeve;      5;      5;5;5;      90;90;90;
Insertion Sleeve;      10;      10;5;10;      90;0;0;
LISS Plate;      5;      7;8;4;      0;45;0;
LISS Plate;      10;      7;10;8;      0;0;45;

```

Figure 10. Sample external file coordinates

### 3.4.7 Color indication

Collision detection control function utilizes the *OnCollisionEnter*, *OnCollisionStay* and *OnCollisionExit* classes for color indication. Private variables for each color indication are applied to this control function. Material rendering and collision detection control are applied on the objects from LISS plate to femur models. The function detects the collision every time when the object in place of the allowable distances. C script is developed in a way that the models are frequently updated based on the position. A trigger function is used to implement the color indication over the sphere objects. The triggers are activated based on the collision status of LISS objects and femur model. A summary of the main functions in this module is provided below.

*OnCollisionEnter* (): Activates Collider during the collision between objects

*OnCollisionStay* (): Initiates the collision status after collision between objects

*OnCollisionExit*(): Exit the collider when the objects are not in collision.

### 3.4.8 Display control

The display control function combines mouse and keyboard controls for the navigation purposes. Forward, turning and movement variables are defined for the target position of the camera movement which also enables the *DisplayUpdate()* function. The display update function updates the framerate of the camera based on the keyboard movement. The *MouseControl ()* function utilizes the orbit camera class for rotating the camera and frequently updates the display based on the move in VEOS. The left mouse button and arrow keys are assigned for the movements in the VEOS to check the positions of LISS components. *OnGUI ()* function enhances the display properties of the VEOS. *KeyboardControl ()* function renders the updates of the object position using keyboard keys. A summary of the main functions in this module is provided below.

*KeyboardControl ()*: Updates the object position with keyboard keys

*DisplayControl ()*: Monitors the and updates the framerate.

*MouseControl ()*: Uses the movement control using mouse buttons

*OnGUI ()*: Initiates the graphical user interface and enhances the display

### 3.5 The Unity based Non-Haptic Virtual environment

The Virtual Environment for Orthopedic Surgery (VEOS) is a non-haptic environment which has been created using the Unity software engine. The current implementation is on a PC based platform. The CAD models of the various objects of interest (OOI) including the femur, LISS components, surgical tools, etc. were built using Solidworks CAD software. The simulation environment was created using the Unity 3D graphics engine along with accompanying ‘application’ related programs written in C++ and Java. A discussion of the Unity based Non-Haptic environment follows.

The Unity 3D based system provides two basic environments or views for interaction: the first environment consists of scene, hierarchy, project and inspector windows where the actual simulation model is created; the second environment is a game window where the user can visualize and interact with the ‘built’ environment.

The Simulation based Environment has 2 segments for LISS plating related training: In this environment, there are 2 modules: (i) Module 1 allows users to understand the assembly of the LISS plate and then facilitates the practice of such an assembly virtually (ii) Module 2 begins with the assembled LISS plate which is then used to complete the surgical process steps discussed earlier

### 3.5.1 Module 1: Learning to assemble the LISS plate

The focus of module 1 is to ensure that medical residents and students acquire a basic vocabulary and knowledge of the various components as well as knowing how to assemble this LISS plate. In the first phase of using module 1, a resident become familiar with the components using both voice and text cues (Figure.11). The second phase of the training involves becoming familiar with the steps involved in assembling the LISS plate.

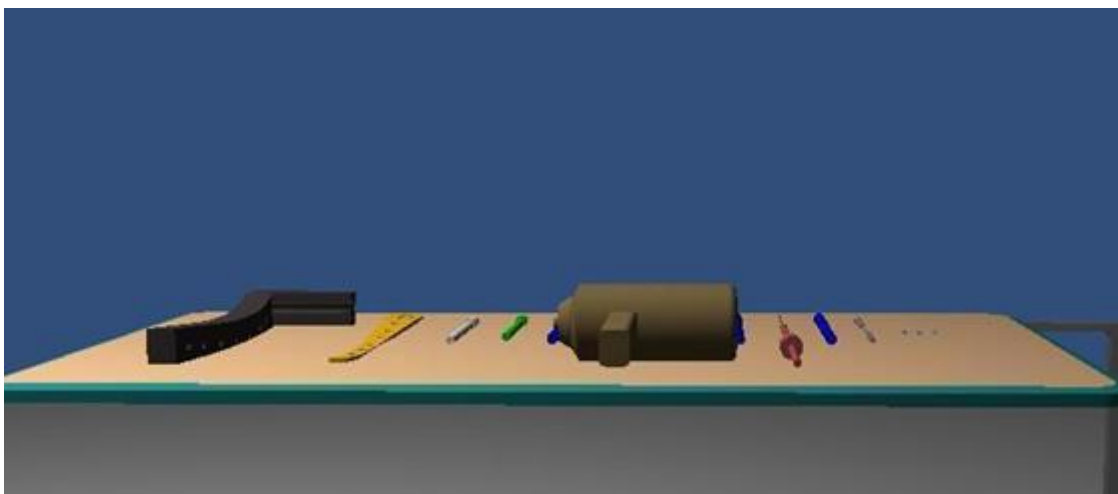


Figure 11. LISS components in VEOS



Initially the user runs the complete simulation of the entire process of LISS plating surgery. The key steps include assemble of insertion guide, insertion of fixation bolt, identifying screw and nut with the appropriate length, screwing the Fixation bolt with the selected screw & nut, and insertion of the stabilization bolt onto the insertion guide. Figure.12 provides a view of this virtual environment with some of the LISS components being assembled. The user has the capability to stop, pause and continue the simulation or repeat a specific step if necessary. Using keyboard and mouse, users zoom in, turn around and explore the intricacies of any of the steps involved.

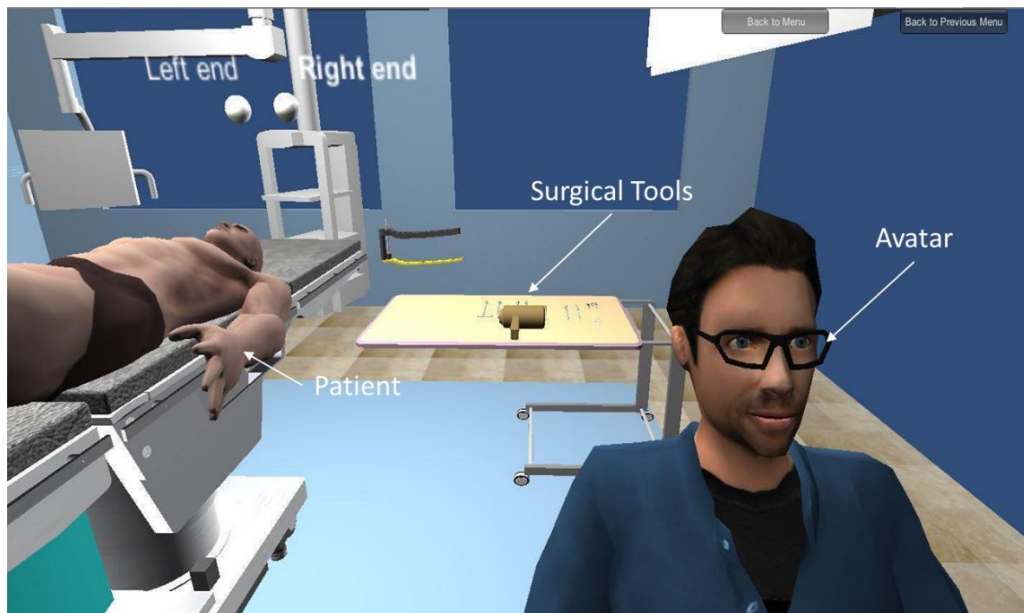


Figure 12. VEOS with some of the LISS components being assembled

Unity software has a capability to accommodate various types of scripts to unite the assembling sequences in the real time environment. Some of the special scripts are used in the creation of the VEOS include textures, colors, appearance, and audio scripts. These scripts used to improve the appearance and quality of the design as well as improve the interactive capabilities. Control scripts are used to facilitate user friendly interactions with the virtual environment.

### 3.5.2 Module 2: Virtual Surgery with use of assembled LISS plate

The second module is developed for medical residents to begin practicing the surgical procedure involving the correction of a distal femur fracture. The simulation allows students to complete the step by step procedures along with providing both voice and text based descriptions of each of the surgical steps. The user is also given the option to pause the simulation at a specific stage or step, and then attempt to complete it or study the angle or orientation of a specific component or surgical tool. A view of this second module is shown in Figure 13.



Figure 13. Virtual Surgery with use of assembled LISS plate

### 3.5.3 Hierarchy in Simulation

The hierarchy in the Unity environment contains custom objects, 3D models, prefabs, lights, cameras, UI's, and audios. The hierarchy uses the parenting concept for any object to make a child of another object. It is accomplished by dragging the desired object into the parent object. The child object inherits the same movement and rotation of its parent object. The nested child option is also possible in the hierarchy where the child can have another object.

The surgery simulation uses the similar situation for the LISS assembly. In the simulation, LISS guide acts as a parent object to the LISS plate, fixation bolt, stabilization bolt, and insertion sleeve plays. The user moves the components and sees the movement and rotation of the child objects when the parent objects are moved in a similar way. Unity platform creates the hierarchy of various objects and interfaces them in the simulation. The sample code in figure 14 (a & b) illustrates the sample attach and detach function from a parent to child objects of surgical components in LISS surgery.

```
function LISSPlateParentFromTranslationtoFixture()
{
    fixture.AttachChildren();
    LISSPlate.transform.parent= LISSPlateAttach;
};
function LISSPlateParentFromRotationtoFixture()
{
    fixture.AttachChildren();
    LISSPlate.transform.parent= LISSPlateAttach;
};
```

a

```
function LISSPlateParentFromTranslationtoFixture()
{
    fixture.DetachChildren();
    LISSPlate.transform.parent= LISSPlateDetach;
};
function LISSPlateParentFromRotationtoFixture()
{
    fixture.DetachChildren();
    LISSPlate.transform.parent= LISSPlateDetach;
};
```

b

Figure 14. (a) Parent and Child attach function, (b) Parent and Child detach function

### 3.6 The Unity based Haptic Virtual environment

The module is developed to assist medical residents to get a practice of surgical methods with real-time force feedbacks using haptic devices. Physical feedback is a determining factor for the success of the LISS distal femur surgery. A successful LISS distal femur surgery requires exceptional hand eye coordination and wariness of minute yet significant physical details. Traditional virtual simulations does not give the user such experiences, thus, the novice surgeon lacks training in this entire degree of perception that is crucial to the surgery. The haptic feedback module allows the novice surgeon to train themselves to notice these physical signs and take action accordingly. No explicit modeling of force feedback approach is provided. The haptic interface primarily is provided so that the resident can get some ‘feel’ of picking up and interacting with the simulation environment. The following figure (15) shows the resident interacting with VEOS using the Phantom haptic interface.



Figure 15. Resident interacting with VEOS using a Phantom haptic interface (Non-immersive virtual surgical environment)

### 3.7. Creation of Semi – Immersive environment for VEOS

Mechdyne provides a getReal3D plug-in for Unity software to support the immersive environments. The plug-in incorporates 3D Stereo, head and hand tracking for interaction with the 3D environment. It utilizes the motion tracking system to pick up the objects in the virtual world, and it gives an astonishing sense of reality. This plug-in is compatible with the environments such as CAVE, head-mounted displays, and virtual reality equipment [62]. The getReal3D plug-in supports the functions of 3D stereo based semi immersive environments and tracker interaction. With the getReal3D plug-in the user can change the non-immersive 2D environment (without stereo eyewear and tracker) into the 3D environment (with stereo eyewear and tracker functionalities). A summary of the getReal3D in this module is provided below.

- getReal3D plugin converts the 2D environment into a 3D environment
- It assists the user to navigate and interact with the 3D environment using wand manager
- getReal3DController monitors the user position with the help of trackers and generates the stereo vision (3D) based on the user location

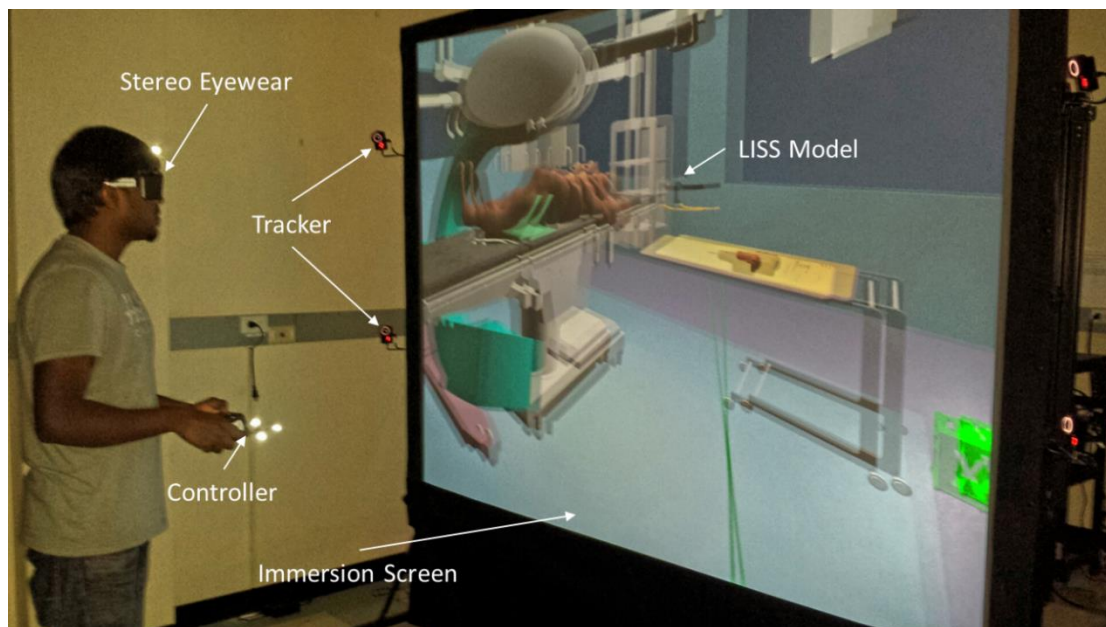


Figure 16. Semi-immersive VR environment for VEOS

Semi immersive virtual environment (Figure 16) for the VEOS is implemented using the getReal3D plug-in in Unity platform. It is designed to assist the surgical steps and assembly procedures in orthopedic surgery. The user can navigate and control the 3D environment using the wand and eyewear. getReal3D plug-in enables the user to immerse in the virtual world using the motion sensors/trackers (mounted on the powerwall), wand (joystick with tracking devices) and stereo eyewear (3D eyewear with tracking devices). It allows the user to interact and create alternative approaches for the surgical procedures.

### 3.8 Description of the various surgery functional modules

In the VEOS, the orthopedic surgeon/resident deals with the surgical tools, human and bone model, implants, surgery table and patient positions. Fracture fixation requires specific virtual fixation techniques and it consists of complex simulation techniques. The virtual environment provides a realistic 3D model with the same behavior and restraints. The real time environment considers all the facts, including the collision between tools and bone, actual surgical procedures, and obtain end results of the surgery. VEOS considers all the required training situations and provided an overview of the surgical procedures in a systematic way. Some of the modules utilized in the VEOS are explained in detail in the following paragraphs.

#### 3.8.1 LISS assembly module

In this module, an automatic simulation environment is developed to show the user to learn the following assembly steps: assembly of LISS insertion guide main components and radiolucent extension, insertion of fixation bolt through hole A of the insertion guide (A-G were used to for distal holes and 1-13 used for diaphyseal holes), attachment of the LISS plate with the fixation bolt using fastener, and insertion of stabilization bolt with insertion sleeve (for more stabilization of LISS plate and guide).

### 3.8.2 LISS Insertion module

In this module, the user learns the insertion techniques of LISS plate assembly. The assembled plate is inserted between the vastus lateralis and the periosteum with the assistance of the LISS insertion guide. The module also demonstrates the major steps that have to be followed during the insertion which is plate orientation, plate position and collision of the plate and distal femur. The plates are most commonly inserted using minimal invasive techniques and the incisions are made in the proximal and distal side of the bone.

### 3.8.3 Fracture reduction module

The fracture reduction module demonstrates the user for reduction of fracture using several steps to achieve the final reduction. The reduction is the most important step in the orthopedic surgery. The simulation gives the preferred approach to resolve the fracture using K-wire and pull reduction instrument. The K-wire is placed in the hole A and E which helps to check the fracture reduction in between proximal and distal location.

### 3.8.4 Screw insertion module

This module demonstrates the screw placement on the femur. In any fracture minimum of four screw insertion are recommended and those screws are placed away from the fracture. The module is designed in a way that the user can check the placement of locking screw and diaphyseal screws. Appropriate screw diameter and length are selected and inserted into the LISS plate through insertion sleeves and tightened using power tools and torque limited screw drivers.

### 3.8.5 Guide removal module

In this final module the removal of guide and other components are present. Initially, insertion sleeve and stabilization bolt in the LISS insertion guide are detached. The guide with

fixation bolt is detached at the end. The virtual reality environment demonstrates these steps in this final module.

#### 3.8.6 Color indication module

The positioning of the plate in between the distal and proximal is an important step for fracture reduction of the femur. The VEOS color indication module is developed to train the surgeon for placing the LISS plate in proper direction using different color indication. In this module the LISS assembly is moved using the keyboard controls and the user has the access to see the visible changes happening around the femur and plate. This supports the surgeons to enhance the surgical skills.

#### 3.9. Conclusion

The methods discussed in this chapter are the relevant approaches for creating a virtual reality environment for VEOS. The architecture and collaboration diagrams are used to create the simulation environment which is very user-friendly. Subsequently, the surgical simulation is performed based on the user requirements. The developed Unity environment enhances the active learning experience and its user-friendly environment keeps the system accurately. The next chapter would discuss about the results and observations obtained after testing the Virtual Environment with various a surgeon and residents.



## CHAPTER IV

### TEST CASES

#### 4.1. Introduction

There were major advances in less invasive stabilization system (LISS) over the past decades. Orthopedic procedures now represent the standard LISS. The advantages of LISS over conventional methods were demonstrated for other operations. To take advantage of the LISS approach, the skilled and trained surgeon had to perform the training procedures. In this virtual reality environment, surgeons were assessed by means of training. Such a training system allowed the repeated procedures and practice of standard tasks. It also provided a hand on experience to the surgeon subsequently measures the performance.

The developed simulation environment was evaluated by Dr. Migul-Pirela Cruz and residents/students from Texas Tech University Health Sciences Center. Dr. Cruz checked each module and provided a detailed feedback on the VEOS. The residents had an experience to practice the module. The feedback from the residents was noted down. Most of the residents felt the simulation environment helped them to learn the surgical steps and process quickly.

## 4.2. Methodology

The study was conducted in the department of orthopedic surgery and rehabilitation in Texas Tech University Health Sciences Center. Six residents/surgeons with limited experience in orthopedic surgery participated in this study. The study was carried out during August 2014 and February 2015. All participants were asked to run the simulation of orthopedic surgery and supervised by an experienced surgeon. The first set of evaluation (August 2014) contained the following training modules that included five tasks of LISS in VEOS (Assemble LISS Plate, Insertion of LISS Plate, Fracture reduction, Insertion of screws, and Removal of the guide). The five tasks were designed to simulate the same techniques used during the orthopedic surgery. The second set of evaluation (February 2015) was modified and improved on the basis of suggestions provided during the first set of evaluation. The second set of evaluation contained five control tasks that can be accessed by the user. In the simulation the user could interact with the virtual environment with the use of keyboard, mouse, Haptic devices, position control and external file uploading. The procedures performed by the residents were assessed by the senior surgeon using pre-test and post-test methods. The performance rating scale was developed in the assessment of the skills demonstrated by residents/surgeons during the training.

## 4.3. Pre-Test

The pre-test was used to gather the information about the residents and their knowledge about the LISS in general. The questions were prepared in a way that the resident can easily answer some of the basic concepts on LISS. The test was conducted for the six residents and the results were used to analyze the post-test.

The following questions were given to the residents during the first set of evaluation,

1. Name 3 components used for LISS plating?
2. Which hole is used for attaching the plate to the jig?

3. Which hole is used to determine that the plate is aligned correctly on the jig?
4. Which portion of the plate is applied on the condyles of the femur?
5. Which portion of the plate is applied on the shaft of the femur?
6. How is the plate applied to the jig?
7. What instrument or tool is used to apply the plate to the jig?
8. What does the acronym LISS mean?
9. How many critical radiographic views are needed for LISS plating of the distal femur?
10. How is the jig attached to the LISS plate?
11. The most distal hole on the LISS plate (where the plate is held to the jig) is applied to which portion of the femur?
12. Which hole(s) is the most important to determine that the plate is aligned correctly on the jig?

**Table 1: Results August 2014**

<b>Title</b>	<b>Pre-test</b>	<b>Post-test</b>
Subject A	50	50
Subject B	50	50
Subject C	16	33
Subject D	16	83
Subject E	50	50
Subject F	0	33

The following questions were given to the residents during the second set of evaluations after the development of the improved VEOS (February 2015). The developed environment incorporated the Haptic device for force feedback. The residents used these haptic techniques to assemble the LISS components in the VEOS.

1. What does the initials LISS represent?
2. What is the basic intent of this type of surgery?
3. What holds the plate?
4. How does one ensure alignment of the plate with the holding apparatus?
5. How are the holes numbered? (Proximal to distal or distal to proximal)
6. What direction should the image (C-arm) come in from?
7. What device positions the extremity correctly on the operating room table?
8. What happens if the extremity positioning device is not used?
9. What x-ray projections are most important to obtain during surgery?
10. What potential complication can occur if the plate is positioned too distal?
11. What can eventually develop if the articular surface of the joint is not reduced properly?
12. What size screws are being used to fix the plate?
13. What device is used for final tightening of the screws?
14. What device is used for the initial insertion of the screws?
15. Which is the last bolt to be removed? and why?

#### 4.4. Post-Test

The post-test was conducted after residents used the simulation environment. The questions which were posed during the pre-test were repeated during the post-test evaluation. The results were compared to the pre-test results.

#### 4.5 Results

All participants performed the training module in the VEOS and supervised by an experienced surgeon. The rating scale was designed for the pre-test and post-test. The surgeon reviewed the training status of the residents and performance improvement individually. The obtained results were tabulated (Tables 1 & 2) on the basis of pre-test and post-test. Analysis of

six subjects and their learning skills on the orthopedic surgery were recorded. The rating scale of the each subject was designed based on the number of questions correctly addressed by the residents.

**Table 2: Results February 2015**

Title	Pre Test	Post Test
Subject A	33.33	86.67
Subject B	13.33	26.67
Subject C	26.67	53.33
Subject D	40	80
Subject E	0	66.67

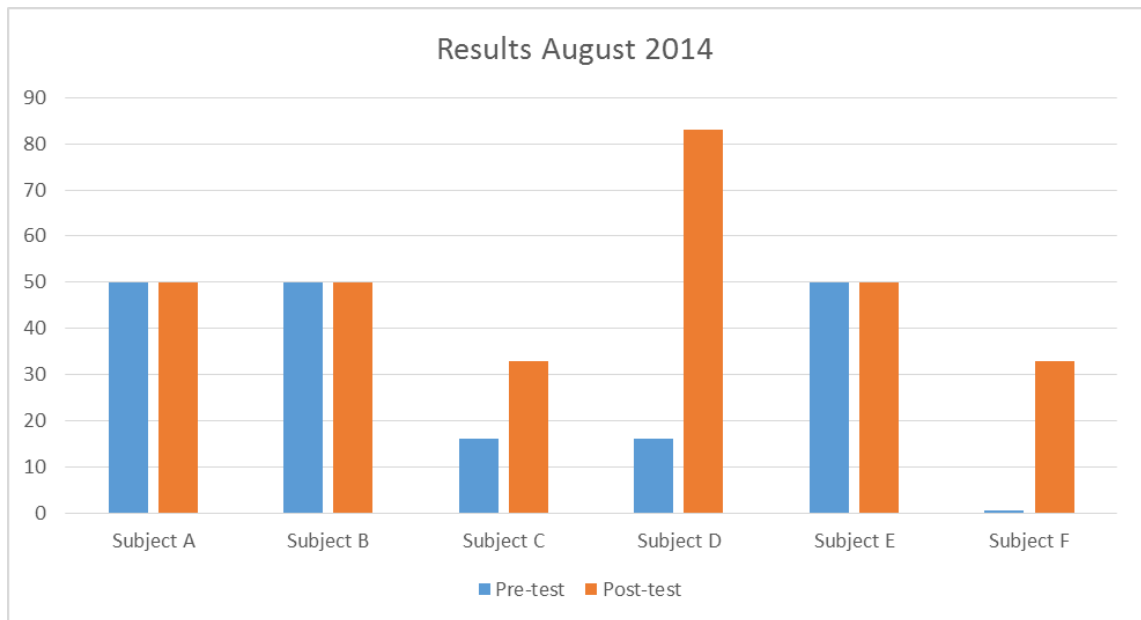


Figure 17. Test performance during August 2014

During the first round of testing (Figure. 17), only three of the six residents showed improvement on their test performance. However, in the second round of testing (Figure. 18) all

participating residents showed improvement on their test performance. The results showed that VEOS training study improved the learning skills as compared with the normal learning methodologies.

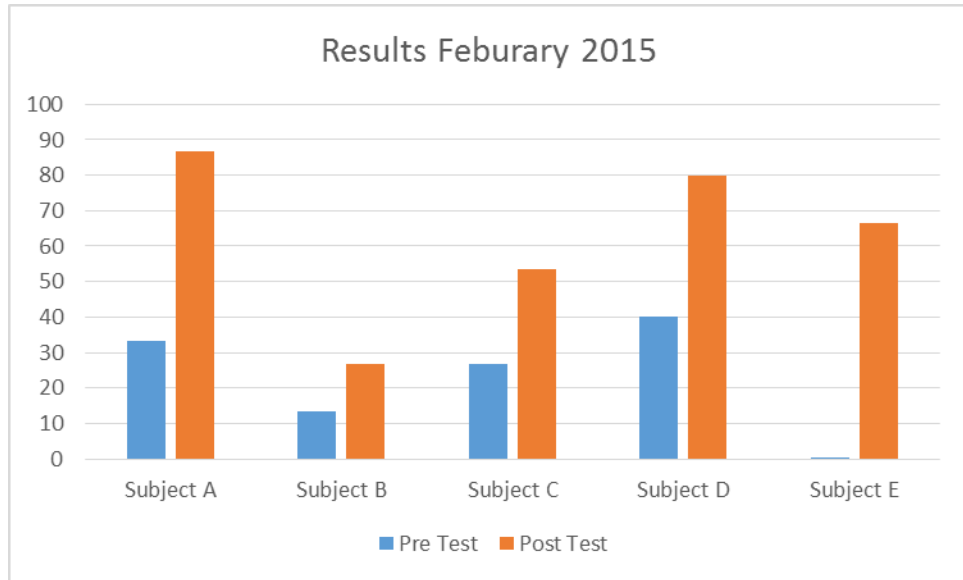


Figure 18. Test performance during February 2015

#### 4.6. Conclusion

The study provided evidence that training with the VEOS improved surgical learning skills and enhanced the performance of the residents. This training in VEOS improved the technical skills relevant to the orthopedic surgery. The results highlighted the importance of training and evaluation of surgical procedures in a virtual environment. The computer developed model allowed repeated practice and training of realistic surgical models using simple controls. The participants in the experimental study showed improvement in the surgery process. The rating scale of the simulation assessment reflected performance improvement and validated the results with the senior surgeons. The conclusions of the whole simulation model and study method are discussed in the next chapter.

## CHAPTER V

### CONCLUSION

#### 5.1. Introduction

Virtual reality simulators were beneficial in increasing the quality of training while decreasing the time needed for training a specific skill. These simulators played a key role in the training of medical residents in surgery and other fields.

An information model was developed interacting with expert surgeons to understand and detail a specific orthopedic surgery process referred to as LISS plating surgery. Subsequently, the virtual environment was designed and developed on a PC based platform. Such simulators can be used to train medical residents in orthopedic surgery procedures.

A multi module simulation environment for orthopedic surgery was developed using Virtual Reality technology and was useful in educating and training medical residents intending to specialize in orthopedic surgery.

#### 5.2. Summary of the Approach

The surgical bones and instruments were created using CAD models. These CAD models were imported into the unity software which helped to develop the virtual environment. VEOS

The environment was developed on the basis of collaborative diagram.

VEOS environment generated the following simulation: LISS assembly, insertion of LISS plate, fracture reduction, screw insertion, and removal of the guide. Each module indicated surgical procedures in a gradual manner. It also had control methods that provide the user interaction for surgery.

### 5.3. Completed work

- Literature review of the virtual reality surgical environments: First a literature review of existing virtual environments for orthopedic surgery, laproscopic surgery and heart surgery was conducted. Secondly the haptic device and tracking techniques used in the virtual environments were discussed. The review of the existing literature presented the challenges and lack of research involved in the less invasive stabilization system for orthopedic surgery.
- Development of a virtual reality framework for orthopedic surgery: Steps involved in the LISS study were identified. This approach decomposed the complex surgical activities in a simple way. Collaboration diagram was created to demonstrate the required programming methods for simulation. This approach united the necessary program classes and objects in a separate diagram. Architecture diagram was developed to show the functional responsibilities of the virtual environment.
- The virtual environment was created using Unity software environment: CAD models were created and imported into unity software. Scripting languages (C&Java Scripts) were applied to the environment to present various surgical steps.
- Developed modules for training: The automatic and manual surgical modules were created. This approach enhanced the user interaction. Color changing module and avatar



module were created to show a detailed surgical process. Control modules were developed to move the objects in the environment using keyboard, mouse and haptic.

- Conducted the test cases: Surgical simulation environment was evaluated by Dr. Cruz. Two sets of test cases for simulation were conducted on the basis of pre and post-test.

#### 5.4. Future Research

Virtual reality environment for medical training is an emerging field of research. Various medical training virtual environments have been discovered and implemented in medical education. The potential issue with this technology is the need for an active interaction with experts prior to the beginning of the field training. Collaborative education is one of the future directions where an expert surgeon at different locations can teach the students using networks. Validation using the test cases has to be increased to a larger group of participants. Further an in-depth research work can be carried out on Haptic force feedback in the collaborative environment. The proposed surgical planning can be integrated with the UNITY environment to assist in simulation. This surgical planning can further be linked to physical surgery. Cloud computing techniques can be used to improve the computing speed and efficiency. In the current scenario, the simulation is limited to LISS, but it can be further expanded to other orthopedic surgical scenarios.

#### 5.5. Conclusion

This chapter discussed the development of a virtual reality based simulation environment for orthopedic surgery. It also discussed the future research which could be undertaken in the same field.

The aim of Virtual Environment Orthopedic Surgery was segregated into following objectives:

- Initial work on Literature review for Virtual reality based surgery environments.

- Design and development of VR based orthopedic surgery using UNITY VR tools.
- C script and JAVA script have been developed to generate simulation for VEOS.
- VEOS can be validated through interaction with surgeons and medical interns.

These objectives are explained in a detail way in the following chapters.

Chapter two gave a detailed survey of virtual environments developed for orthopedic surgery, heart surgery and laparoscopic surgery. It also dealt with the other challenges involved in the development of virtual surgical environments.

Chapter three provided an overall architecture and approach of the LISS. Collaboration diagram was developed to create a complete control function for VEOS. UNITY based Haptic and non-Haptic environments were created to demonstrate about the procedures of LISS.

Chapter four described about the evaluation procedures and techniques used for the VEOS. It gave a detailed description about the methodologies for analyzing the performance of residents. This analysis involved pre and post-test methods.

Chapter five provided a brief summary of the work accomplished in this project report. This chapter explained the importance of the problem and solution to the problems. Future directions of research work were also discussed.

## REFERENCES

1. Cecil, J. (2013), Cyber and Cloud Technologies: Next Generation Technologies for Engineering Education. World Innovations in Engineering Education and Research, pp 41-51.
2. Cecil, J., Cruz, M. (2013), An information model for designing virtual environments for orthopedic surgery. Proceedings of the 2013 EI2N workshop, OTM Conference, Graz, Austria.
3. Cecil, J., Ramanathan, P., et al. (2013), Collaborative Virtual Environments for Orthopedic Surgery. Proceedings of the 9th annual IEEE International Conference on Automation Science and Engineering (IEEE CASE 2013), Madison, WI, August 17 to 21.
4. Cecil, J. (2012), Development of Virtual Learning Environments in Engineering Education. Innovations 2012, World Innovations in Engineering Education and Research, pp 263-275.
5. Cecil, J., Cruz, M. (2011), An Information Model based framework for Virtual Surgery. International Journal of Virtual Reality, International Journal of Virtual Reality, Vol 10 (2), pp 17-31.
6. Cecil, J., Huber, J. (2010), Virtual Prototyping in Engineering. Virtual Engineering, Momentum Press, New Jersey, pp 1- 15.
7. Cecil, J., Cruz, M. (2010), Development of an Information Model for Virtual Surgical Environments. Proceedings of the 2010 Tools and Methods in Competitive Engineering, Ancona, Italy, April 12-15.
8. Cecil, J., Gobinath, N. (2009), Virtual Reality based Prototyping Techniques in Product and Process design. Proceedings of 2009 Computer Aided Design conference, Reno, Nevada, June 8-12.
9. Qin, J., Pang, W, M., Chui, Y, P., Wong, T, T., Heng, P, A. (2010), A Novel Modeling Framework for Multilayered Soft Tissue Deformation in Virtual Orthopedic Surgery. Journal of Medical Systems, Vol. 34, pp 261–271.

10. Delp, S, L., Loan, J, P. (1995), A Graphics-Based Software System To Develop And Analyze Models Of Musculoskeletal Structures. *Computers in Biology and Medicine*, Vol. 25, No.1, pp 21-34.
11. Vankipuram, M., Kahol, K., McLaren, A., Panchanathan, S. (2010), A virtual reality simulator for orthopedic basic skills: A design and validation study. *Journal of Biomedical Informatics*. Vol.43(5), pp 661-668.
12. Tsai, M, D., Hsieh, M, S., Tsai, C, H. (2007), Bone drilling haptic interaction for orthopedic surgical simulator. *Computers in Biology and Medicine*, Vol.37, pp 1709-1718.
13. Blyth, P, Anderson, I, A., Stott, N, S. (2006), Virtual Reality Simulators in Orthopedic Surgery: What Do the Surgeons Think?. *Journal of Surgical Research*, Vol. 131, pp 133-139.
14. Tsai, M, D., Hsieh, M, S., Jou, S, B. (2001), Virtual reality orthopedic surgery simulator. *Computers in Biology and Medicine*, Vol. 31, pp 333-351.
15. Sourina, O., Sourin, A., Howe, T, S. (2000), Virtual Orthopedic Surgery Training on Personal Computer. *International Journal of Information Technology*, Vol. 6, no. 1, pp 16-29.
16. Zhang, G., Zhao, S., Xu, Y. (2003), A virtual reality based arthroscopic surgery simulator. *Proceedings of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, Vol. 1, pp 272-277.
17. Kong, M, X., Du, Z, J., Sun, L, N., Fu L, X., Jia, Z, H., Wu, D, M. (2005), A Robot-Assisted Orthopedic Telesurgery System. *Proceedings of IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4, 2005*, pp 97-101.
18. Vanicatte, M., Krishnan, S, M., Arney, D., Lee, I., Goldman, J, M. (2009), Incorporation of Interoperability in a Robot-Assisted Orthopedic Surgery System. *IFMBE Proceedings of 25th Southern Biomedical Engineering Conference*, Vol. 24, pp 13-16.
19. Zhijiang, D., Zhiheng, J., Minxiu, K. (2005), Virtual Reality-based Telesurgery via Teleprogramming Scheme Combined with Semi-autonomous Control. *Proceedings of IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4*, pp 2153-2156.
20. Monan, W. (2009), Development and validity of Tissue Biomechanics Modeling for Virtual Robot Assisted Orthopedic Surgery System. *3rd International Conference on Bioinformatics and Biomedical Engineering*, pp1-4.
21. Zhang, F., Zhijiang, D., Lining, S., and Zhiheng, J. (2007), A new novel virtual simulation system for robot-assisted orthopedic surgery. *Proceedings of IEEE International Conference on Robotics and Biomimetics*, December 15 -18, Sanya, China, pp 366-370.
22. Rambani, R., Viant, W., Ward, J., Mohsen, A. (2013), Computer-Assisted Orthopedic training system for fracture fixation. *Journal of Surgical Education*, Vol 70 (3), May-June 2013, pp 304-308.
23. Rosenblum, L., Macedonia, M. (2000), Virtual orthopedic surgery training. *IEEE Computer Graphics and Applications*, Vol-20(3), pp 6-9.

24. Burdea, G., Popescu, V., Hentz, V., Colbert, K. (2000), Virtual reality-based orthopedic telerehabilitation. *IEEE Transactions on Rehabilitation Engineering*, Vol. 8, no. 3, pp 430-432.
25. Heng, P, A., Cheng, C, Y., Wong, T, T., Xu, Y., Chui, Y, P., Chan, K, M., Tso, S, K. (2004), A Virtual-reality training system for knee arthroscopic surgery. *IEEE Transactions on Information Technology in Biomedicine*, Vol. 8 (2), pp 217-227.
26. Padilla-Castaneda, M, A., Sotgiu, E., Frisoli, A., Bergamasco, M., Orsini, P, Martiradonna, A., Olivieri, S., Mazzinghi, G., Laddaga, C. (2013), A virtual reality system for robotic-assisted orthopedic rehabilitation of forearm and elbow fractures. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* November 3-7, Tokyo, Japan, pp 1506-1511.
27. Eriksson, M., Flemmer, H., Wikander, J. A Haptic and Virtual Reality Temporal Bone Surgery Simulator. *Mechatronics Lab, The Royal Institute of Technology Stockholm, Sweden*, pp 1-17.
28. Pappas, G., Gomoll, A, H., Forsythe, B., Warner, J, J, P. Virtual Reality Simulation for Shoulder Arthroscopy: A Three-Year Follow-Up Study of Individual Skill Progression, Harvard Shoulder Service, Brigham and Women's Hospital and Massachusetts General Hospital, Department of Orthopedic Surgery Brigham and Women's Hospital, 75 Francis Street, PBB A-2 Boston, MA 02115, pp 95-98.
29. Lundstrom, C., Rydell, T., Forsell, C., Persson, A., Ynnerman, A., (2011), Multi-Touch Table System for Medical Visualization: Application to Orthopedic Surgery Planning. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, No. 12, pp 1775-1784.
30. d'Aulignac, D., Cavusoglu, M, C., Laugier, C. Modeling the Dynamics of the Human Thigh for a Realistic Echographic Simulator with Force Feedback. *GRAVIR Laboratory, France, Robotics and Intelligent Machines Laboratory, University of California, Berkeley, C. Taylor, A. Colchester (Eds.): MICCAI'99, LNCS 1679, pp 1191-1198.*
31. Meseure, P., Chaillou, C. (1997), Deformable Body Simulation with adaptive subdivision and cuttings. *Proceedings of the WSCG'97 Conference, Plzen, 10-14 February*, pp 361-370.
32. Mabrey, J, D., Reinig, K, D., Cannon, W, D. (2010), Virtual Reality in Orthopaedics Is It a Reality?. *Clinical Orthopaedics and Related Research*, Vol.468, pp 2586-2591.
33. Seth, T., Chaudhary, V., Buyea, C., Bone, L.(2011), A Virtual Interactive Navigation System for Orthopaedic Surgical Interventions. *ISABEL '11, October 26-29, Barcelona, Spain.*
34. Willaert, W, I., et al. (2012), Recent Advancements in Medical Simulation: Patient-Specific Virtual Reality Simulation. *World journal of surgery*, Vol. 36 (7), pp 1703-12.
35. Karaliotas, C. (2011), When Simulation in Surgical Training Meets Virtual Reality. *Hellenic Journal of Surgery*, Vol. 83(6), pp 303-16.
36. Azzie, G., Gerstle, J, T., Nasr, A., Lasko, D., Green, J., Henao, O., Farcas, M., Okraïneç, A. (2011), Development and validation of a pediatric laparoscopic surgery simulator. *Journal of Pediatric Surgery*, Vol. 46, pp 897-903.

37. Seixas-Mikelus S, A., Kesavadas, T., Srimathveeravalli, G., Chandrasekhar, R., Wilding, G.E., Guru, K,A. (2010), Face Validation of a Novel Robotic Surgical Simulator. *Laparoscopy and Robotic*, Vol. 76 (2), pp 357-360.
38. Makiyama, K., Nagasaka, M., Inuiya, T., Takanami, K., Ogata, M., Kubota, Y. (2012), Development of a patient-specific simulator for laparoscopic renal surgery. *International Journal of Urology*, Vol. 19, pp 829-835.
39. Grantcharov, T, P., Kristiansen, V, B., Bendix, J., Bardram, L., Rosenberg, J., Funch-Jensen, P. (2004), Randomized Clinical Trial of Virtual Reality Simulation for Laparoscopic Skills Training. *British Journal of Surgery*; Vol. 91, pp 146-150.
40. Aggarwal, R., Balasundaram, I., Darzi, A. (2008), Training Opportunities and the Role of Virtual Reality Simulation in Acquisition of Basic Laparoscopic Skills. *Journal of Surgical Research*, Vol. 145, pp 80–86.
41. Peters, T,M., Linte, C,A., More, J., Bainbridge, D., Jones, D,L., Guiraudon, G,M. (2008), Towards a medical virtual reality environment for minimally invasive cardiac surgery. *Miar, Lncs 5128*, pp 1-11.
42. Par, J, W., Choi, J., Park, Y., Sun, K., (2001), Haptic Virtual Fixture for Robotic Cardiac Catheter Navigation. *Artificial Organs*, ISSN 0160-564X, Vol. 35, Issue 11, pp 1127-1131.
43. Li, M., Ishii, M., Taylor, R, H. (2007), Spatial motion constraints using virtual fixtures generated by anatomy. *IEEE T Robot*, Vol. 23, pp 4–19.
44. Sorensen, T, S., Therkildsen, S, V., Makowski, P., Knudsen, J, L., Pedersen, E, M. (2001), A new virtual reality approach for planning of cardiac interventions. *Artificial Intelligence in Medicine*, Vol. 22, Issue 3, pp 193-214.
45. Berlage, T., Schmitgen, A., Schmitz, C., Welz, A. (2001), Simulation and Planning of Minimally Invasive Coronary Artery Bypass Surgery. *International Congress Series Vol. 1230*, pp 68-72.
46. Thanh, L, V., Nahavandi, S. (2010), A Haptic Training Environment for the Heart Myoblast Cell Injection Procedure. *11th Int. Conf. Control, Automation, Robotics and Vision*, Singapore.
47. Ren, J., Patel, R, V., McIsaac, K, A., Guiraudon, G., Peters, T, M. (2008), Dynamic 3-D Virtual Fixtures for Minimally Invasive Beating Heart Procedures. *IEEE Transactions on Medical Imaging*, Vol. 27 (8), pp 1061- 1070.
48. Yu, R., Zhang, S., Chiang, P., Cai, Y., Zheng, J. (2010), Real-Time and Realistic Simulation for Cardiac Intervention with GPU. *Second International Conference on Computer Modeling and Simulation*, pp 1-10.
49. Montgomery, K., Stephanides, M., Schendel, S., Ross, M. (1998), A case study using the virtual environment for reconstructive surgery. *Proceeding of Visualization '98 conference*, pp 431 – 434.
50. Suzuki, N., Hattori, A., Takatsu, A., Uchiyama, A., kumano, T., Ikemoto, A., Adachi, Y., (1998), Virtual surgery simulator with force feedback function. *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vol. 20(3), pp 227-33.

51. Johansson, P., Ynnerman, A. (2004), Immersive visual interfaces-assessing usability by the effects of learning/results from an empirical study. *Journal of Computing and Information Science in Engineering*, Vol. 4, pp 124-131.
52. Coutee, A.S., McDermott, S.D., Bras, B. (2002), A haptic assembly and disassembly simulation environment and associated computational load optimization techniques. *Journal of Computing and Information Science in Engineering*, Vol. 1, pp 113-122.
53. Springer, S, L., Ferrier, N, J. (2002), Design and control of a force-reflecting haptic interface for teleoperational grasping. *Journal of Mechanical design*, Transactions of ASME 124, pp 277-283.
54. Balijepalli, A., Kesavdas, T. (2004), Value-addition of haptics in operator training for complex machining tasks. *Journal of Computing and Information Science in Engineering*, Vol. 4, pp 91-97.
55. Gregory, A., Lin, M., Gottschalk, S., Taylor, R. (1999), A Framework for Fast and Accurate Collision Detection for Haptic Interaction. *IEEE Annual VR International Symposium*, pp 38-45.
56. Magnetic Levitation (MagLev) Haptic Consortium web site, <http://maglevhaptics.org/> Accessed 12 August 2014.
57. Magnetic Levitation Haptic Consortium – Butterfly Haptics web site, <http://butterflyhaptics.com/maglev-haptics/> Accessed 12 August 2014.
58. Gurocak, H., Jayaram, S., Parrish, B., Jayaram, U. (2003), Weight sensation in virtual environments by using a haptic device with air jets. *ASME Journal of Computing and Information Science in Engineering*, Vol. 3, pp 130-134.
59. Jayaram, U., Repp, R. (2002), Integrated real-time calibration of electromagnetic tracking of user motions for engineering applications in virtual environments. *Journal of Mechanical design*, Vol. 124, pp 623-632.
60. Less Invasive Stabilization System Technique Guide, SYNTHES, <http://synthes.vo.llnwd.net/o16/LLNWMB8/US%20Mobile/Synthes%20North%20America/Product%20Support%20Materials/Technique%20Guides/SUTGLISSDistFemurJ2892H.pdf>, Accessed 25 August 2014.
61. Craig Larman (2005), *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. ISBN-13: 978-0131489066
62. Mechdyne getRead3D plug-in, <http://www.mechdyne.com/software.aspx?name=getReal3D+for+Unity>, Accessed 01 March 2015.

## APPENDICES

A sample class program is shown in the following page,

```
using UnityEngine;
using System.Collections;

public class Example2 : MonoBehaviour
{
    FunctionClass myClass;
    public Vector3 pointAFix = new Vector3(11.619f,-3.695f,3.7702f);
    public Vector3 pointBFix = new Vector3(10.074f,1.331f,4.8505f);
    public Vector3 RotAFix = new Vector3(90,0,0);
    public Vector3 RotBFix = new Vector3(0,-90,0);
    public Vector3 pointAFixationBolt = new Vector3(9.54503f,-2.8118f,-0.0070f);
    public Vector3 pointBFixationBolt = new Vector3(4.952134f,3.0634f,4.6913f);
    public Vector3 pointCFixationBolt = new Vector3(11.2263f,3.0634f,5.28545f);
    public Vector3 pointDFixationBolt = new Vector3(11.2263f,-0.4133f,5.28545f);
    public Vector3 RotAFixationBolt = new Vector3(0,0,270);
    public Vector3 RotBFixationBolt = new Vector3(0,180,360);
    public Vector3 pointALISSPlate = new Vector3(9.3202f,-2.617f,1.9476f);
    public Vector3 pointBLISSPlate = new Vector3(10.705f,-0.125f,2.8174f);
    public Vector3 RotALISSPlate = new Vector3(0,0,0);
    public Vector3 RotBLISSPlate = new Vector3(0,-90,0);
    public Vector3 pointAInsertionSleeve1 = new Vector3(10.78f,-2.343f,-1.995f);
    public Vector3 pointBInsertionSleeve1 = new Vector3(10.687f,3.1444f,-0.274f);
    public Vector3 pointCInsertionSleeve1 = new Vector3(10.687f,3.1444f,4.4632f);
    public Vector3 pointDInsertionSleeve1 = new Vector3(10.687f,0.3156f,4.4632f);
    public Vector3 RotAInsertionSleeve1 = new Vector3(90,180,90);
    public Vector3 RotBInsertionSleeve1 = new Vector3(0,90,0);
    public Vector3 pointAStabilizationBolt = new Vector3(9.9422f,-2.739f,-0.884f);
    public Vector3 pointBStabilizationBolt = new Vector3(9.9422f,2.2445f,-0.884f);
    public Vector3 pointCStabilizationBolt = new Vector3(11.118f,2.2445f,4.7188f);
    public Vector3 pointDStabilizationBolt = new Vector3(11.118f,-0.29f,4.7188f);
    public Vector3 RotAStabilizationBolt = new Vector3(0,0,-90);
    public Vector3 RotBStabilizationBolt = new Vector3(0,0,0);
}
```



```

void Start ()
{
    GameObject Fix;
    GameObject FixationBolt;
    GameObject LISSPlate;
    GameObject InsertionSleeve1;
    GameObject StabilizationBolt;
    Fix = GameObject.Find("Left LISS Insertion Guide");
    FixationBolt = GameObject.Find("Fixation bolt with hole- for Liss insertion guides");
    LISSPlate = GameObject.Find("Left Liss Plate 13 Holes aligned");
    StabilizationBolt = GameObject.Find("Stabilization bolt with hole- for Liss insertion
guides edited");
    InsertionSleeve1 = GameObject.Find("Insertion Sleeve with Screw part edited");
    myClass = new
    FunctionClass(Fix,FixationBolt,LISSPlate,InsertionSleeve1,StabilizationBolt);
}

void OnMouseDown()
{
    myClass.MoveFixAtoB(transform, pointAFix, pointBFix,RotAFix ,RotBFix ,2.0f);
    yieldFunction(1.0f);

    myClass.MoveFixationBoltAtoB(transform, pointAFixationBolt,
pointBFixationBolt,pointCFixationBolt,pointDFixationBolt, RotAFixationBolt
,RotBFixationBolt ,2.0f);
    yieldFunction(6.5f);
    myClass.MoveLISSAAtoB(transform, pointALISSPlate,
pointBLISSPlate,RotALISSPlate,RotBLISSPlate, 2.0f);
    yieldFunction(8.0f);
    myClass.MoveInsertionSleeve1AtoB(transform, pointAInsertionSleeve1,
pointDInsertionSleeve1,RotAInsertionSleeve1,RotBInsertionSleeve1, 2.0f);
    yieldFunction(2.5f);
    myClass.MoveStabilizationBoltAtoB(transform, pointAStabilizationBolt,
pointBStabilizationBolt, pointCStabilizationBolt,
pointDStabilizationBolt,RotAStabilizationBolt,RotBStabilizationBolt, 2.0f);
    yieldFunction(6.5f);
}

IEnumerator yieldFunction(float i)
{
    yield return new WaitForSeconds (i);
}

```

```

class FunctionClass
{
    IEnumerator yieldFunction(float i)
    {
        yield return new WaitForSeconds (i);
    }
    GameObject Fix;
    GameObject FixationBolt;
    GameObject LISSPlate;
    GameObject InsertionSleeve1;
    GameObject StabilizationBolt;
    public FunctionClass(GameObject Fix1, GameObject FixationBolt1, GameObject
    LISSPlate1,GameObject InsertionSleeve11, GameObject StabilizationBolt1 )
    {
        Fix = Fix1;
        FixationBolt = FixationBolt1;
        LISSPlate = LISSPlate1;
        InsertionSleeve1 = InsertionSleeve11;
        StabilizationBolt = StabilizationBolt1;
    }
    public void MoveFixAtoB (Transform thisTransform , Vector3 startPos, Vector3
    endPos,Vector3 startRot , Vector3 endRot,float time)
    {
        float i = 0.0f;
        float rate = 1.0f/time;
        while (i < 1.0f)
        {
            i += Time.deltaTime * rate;
            Fix.transform.localPosition = Vector3.Lerp(startPos, endPos, i);
            Fix.transform.localEulerAngles = Vector3.Lerp (startRot,endRot,i);
            yieldFunction(0f);
        }
    }
    public void MoveFixationBoltAtoB (Transform thisTransform, Vector3 startPos , Vector3
    endPos,Vector3 Pos3,Vector3 Pos4, Vector3 startRot,Vector3 endRot, float time)

```

```

{
    float i = 0.0f;
    float rate = 1.0f/time;
    while (i < 1.0f)
    {
        i += Time.deltaTime * rate;
        FixationBolt.transform.localPosition = Vector3.Lerp(startPos, endPos, i);
        FixationBolt.transform.localEulerAngles = Vector3.Lerp (startRot,endRot,i);
        yieldFunction(0f);
    };

    float i2 = 0.0f;
    float rate2 = 1.0f/time;
    while (i2 < 1.0f)
    {
        i2 += Time.deltaTime * rate2;
        FixationBolt.transform.localPosition = Vector3.Lerp(endPos, Pos3, i2);
        yieldFunction(0f);
    };

    float i3 = 0.0f;
    float rate3 = 1.0f/time;
    while (i3 < 1.0f)
    {
        i3 += Time.deltaTime * rate3;
        FixationBolt.transform.localPosition = Vector3.Lerp(Pos3, Pos4, i3);
        yieldFunction(0f);
    }
}

public void MoveLISSAtoB (Transform thisTransform,Vector3 startPos,Vector3
endPos,Vector3 startRot,Vector3 endRot,float time)
{
    float i = 0.0f;
    float rate = 1.0f/time;
    while (i < 1.0f)
    {
        i += Time.deltaTime * rate;
        LISSPlate.transform.localPosition = Vector3.Lerp(startPos, endPos, i);
        LISSPlate.transform.localEulerAngles = Vector3.Lerp (startRot,endRot,i);
        yieldFunction(0f);
    }
}

public void MoveInsertionSleeve1AtoB (Transform thisTransform,Vector3
startPos,Vector3 endPos,Vector3 Pos3,Vector3 Pos4,Vector3 startRot,Vector3
endRot,float time)
{
    float i = 0.0f;
    float rate = 1.0f/time;
    while (i < 1.0f)
    {
        i += Time.deltaTime * rate;
        InsertionSleeve1.transform.localPosition = Vector3.Lerp(startPos, endPos, i);
        InsertionSleeve1.transform.localEulerAngles = Vector3.Lerp (startRot,endRot,i);
        yieldFunction(0f);
    };

    float i2 = 0.0f;
    float rate2 = 1.0f/time;
    while (i2 < 1.0f)

```

```

{
    i2 += Time.deltaTime * rate2;
    InsertionSleeve1.transform.localPosition = Vector3.Lerp(endPos, Pos3, i2);
    yieldFunction(0f);
}

float i3 = 0.0f;
float rate3 = 1.0f/time;
while (i3 < 1.0f)
{
    i3 += Time.deltaTime * rate3;
    InsertionSleeve1.transform.localPosition = Vector3.Lerp(Pos3, Pos4, i3);
    yieldFunction(0f);
}

}

public void MoveStabilizationBoltAtoB (Transform thisTransform,Vector3
startPos,Vector3 endPos,Vector3 Pos3,Vector3 Pos4,Vector3 startRot,Vector3
endRot,float time)
{
    float i = 0.0f;
    float rate = 1.0f/time;
    while (i < 1.0f)
    {
        i += Time.deltaTime * rate;
        StabilizationBolt.transform.localPosition = Vector3.Lerp(startPos, endPos, i);
        StabilizationBolt.transform.localEulerAngles = Vector3.Lerp (startRot,endRot,i);
        yieldFunction(0f);
    };

    float i2 = 0.0f;
    float rate2 = 1.0f/time;
    while (i2 < 1.0f)
    {
        i2 += Time.deltaTime * rate2;
        StabilizationBolt.transform.localPosition = Vector3.Lerp(endPos, Pos3, i2);
        yieldFunction(0f);
    }

    float i3 = 0.0f;
    float rate3 = 1.0f/time;
    while (i3 < 1.0f)
    {
        i3 += Time.deltaTime * rate3;
        StabilizationBolt.transform.localPosition = Vector3.Lerp(Pos3, Pos4, i3);
        yieldFunction(0f);
    }
}
}
}
}
}

```

## Nomenclature

Symbol	Description
A1	Assemble LISS function input to LISS assembly module
A2	LISS Assemble Status function output from LISS assembly module
A3	Position LISS Assembly function input to Position LISS Assembly module
A4	Insertion status function output from Position LISS Assembly module
A5	Reduce fracture function input to Fraction reduction module
A6	Reduction status function output from Fraction reduction module
A7	Insert screw function input to Insert screw module
A8	Insertion status function output from Insert screw module
A9	Remove guide function input to Remove guide module
A10	Guide removal status function output from Remove guide module
C1	Collision detection function input to Collision control module
C2	Detected collision status function output from Collision control module
D1	Display update function input to Display control module
D2	Frame rate status function output from Display control module
E1	Text file coordination function input to External file control module
E2	Text file coordination status function output from External file control module
F1	Keyboard input function to Keyboard control module
F2	Received status of keyboard function output from Keyboard control module
G1	Initiate automatic interaction/surgery function input to Automatic module
G2	Update of the simulation function output from Automatic module
H1	Haptic control initiation function input to Haptic control manager
H2	Updates function from Haptic control manager
H1-a	Gets the workstation function to Interface
H2-a	Sets the workstation function from interface
H3	Synchronize the graphical station to physical station device function to Plugin
H4	Gets ready to interact status function from plugin
H5	Feedback function to haptic device
H6	Provides feedback from haptic device
I1	Gets the mouse control function to Mouse control module
I2	Provides the interaction status from Mouse control module
J1	Gets the updates to Avatar control module
J2	Provides the interaction status from Avatar control module
K1	Gets the user inputs function to Manual module
K2	Checks the status from Manual module
L1	Transfers the data to Network control module
L2	Receives the data from Network control module
M1	Gets the menu options function to Menu control module
M2	Updates the menu option functions from Menu Control module

VITA

MURUGAPPAN BALAKRISHNAN, BHARATHI RAJ KUMAR

Candidate for the Degree of

Master of Science

Thesis: A VIRTUAL REALITY BASED ENVIRONMENT FOR ORTHOPEDIC SURGERY (VEOS)

Major Field: Industrial Engineering and Management

Biographical:

Education:

Completed the requirements for the Master of Science in Industrial Engineering and Management at Oklahoma State University, Stillwater, Oklahoma in May, 2015.

Completed the requirements for the Master of Science in Engineering Design at PSG College of Technology, Coimbatore, Tamil Nadu, India in May, 2012.

Completed the requirements for the Bachelor of Science in Automobile Engineering at Bharathidasan University, Tiruchirappalli, Tamil Nadu, India in May, 2008.

Experience: Graduate Research and Teaching Assistant (October 2013 to present)

Professional Memberships: Alpha Pi Mu, Phi Kappa Phi